Kenobi

TRYHACKME | Resolución de la Máquina KENOBI - HACKING ÉTICO [CTF]

Resolución de la máquina kenobi de tryhackme paso a paso, donde vamos a aprender el funcionamiento de distintas vulnerabilidades y a cómo detectarlas. Veremos las monturas dentro de la máquina kenobi y como mover archivos internos

https://youtu.be/7L4T_9G3f-A?si=F8yDTTUbOAR5w93q

Machine: Easy

IP: 10.10.51.163

Antes que todo, lo primero que hacemos siempre es verificar que tengamos conexión con la maquina, para ellos lo hacemos con un ping.

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi]
└─# ping -c 1 10.10.51.163
PING 10.10.51.163 (10.10.51.163) 56(84) bytes of data.
64 bytes from 10.10.51.163: icmp_seq=1 ttl=63 time=71.1 ms

── 10.10.51.163 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 71.090/71.090/71.090/0.000 ms

┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi]
└─#
```

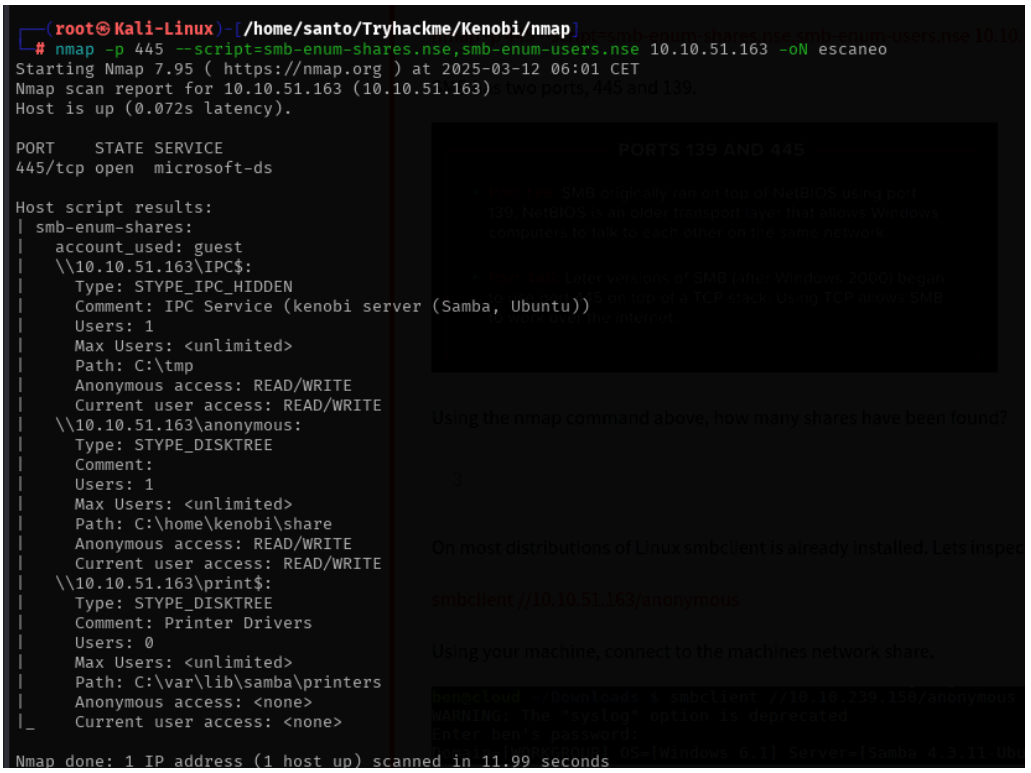
Ahora con nmap vamos hacer un reconocimiento de los puertos, para ver que puertos hay abiertos y que servicios corren de tras de dichos puertos, para esta enumeración usaremos este comando que es muy completo y optimo para lo que necesitamos

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi/nmap]
└─# nmap -p- -sS -Pn -sC --open --min-rate 5000 -n -Pn -vvv 10.10.51.163 -oG allPort
```

Ahora ya tenemos nuestra escaneo exhaustivo con nmap

Una vez ya sepamos que puertos están abiertos con este comando podemos mirar que servicios estas corriendo en el puerto 445 en especifico

`nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.51.163`



Vamos hace un escaneo del puerto 111 donde se esta ejecutando rpcbind, un servicio que gestiona llamadas de procedimiento remoto (RPC), el cual es un programa que utiliza una computadora para ejecutar código en otra maquina remota.

`nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.43.165`

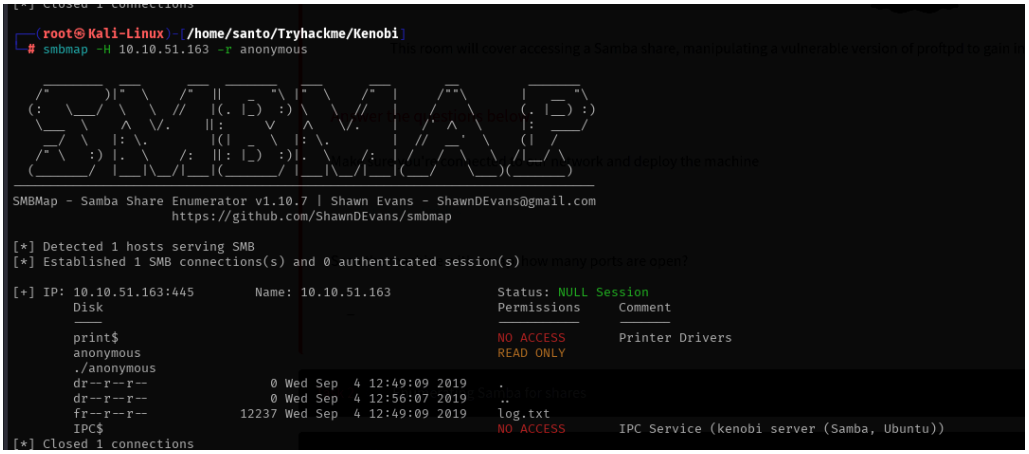
```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi/nmap]
└─# nmap -p 111 --script=nfs-ls,nfs-showmount 10.10.43.165 -oN escaneoPort111
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-12 23:35 CET
Nmap scan report for 10.10.43.165 (10.10.43.165)
Host is up (0.072s latency).

PORT    STATE SERVICE
111/tcp open  rpcbind
| nfs-ls: Volume /var
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID  GID  SIZE  TIME                 FILENAME
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  .
| rwxr-xr-x   0    0    4096  2019-09-04T12:27:33  ..
| rwxr-xr-x   0    0    4096  2019-09-04T12:09:49  backups
| rwxr-xr-x   0    0    4096  2019-09-04T10:37:44  cache
| rwxrwxrwx   0    0    4096  2019-09-04T08:43:56  crash
| rwxrwsr-x   0    50   4096  2016-04-12T20:14:23  local
| rwxrwxrwx   0    0    9     2019-09-04T08:41:33  lock
| rwxrwxr-x   0    108  4096  2019-09-04T10:37:44  log
| rwxr-xr-x   0    0    4096  2019-01-29T23:27:41  snap
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  www
|_
| nfs-showmount:
|_  /var *

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds
```

Bueno como podemos observar hay muchos puertos abiertos en esta maquina, pero en especial vemos el puerto 445 que es el puerto de recursos compartidos de Windows (samba) por lo que con la herramienta `smbmap` vamos a buscar los recursos compartidos que se estén compartiendo en la maquina objetivo.

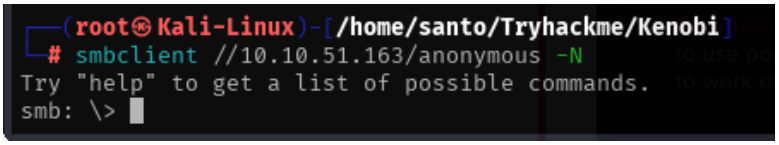
```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi]
└─# smbmap -H 10.10.51.163

SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
                     https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 0 authenticated session(s)

[+] IP: 10.10.51.163:445        Name: 10.10.51.163              Status: NULL Session
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        anonymous                                               READ ONLY
        IPC$                                                    NO ACCESS       IPC Service (kenobi server (Samba, Ubuntu))
[*] Closed 1 connections

┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi]
└─#
```

Y aquí como vemos nos a encontrado varios recursos compartidos, uno de ellos se llama Anonymous

Con este comando podríamos acceder a este recurso compartido, pero aquí lo interesante seria poder acceder por una linea de comandos para así podernos mover por los directorios, entonces para esto es mejor `smbclient`.



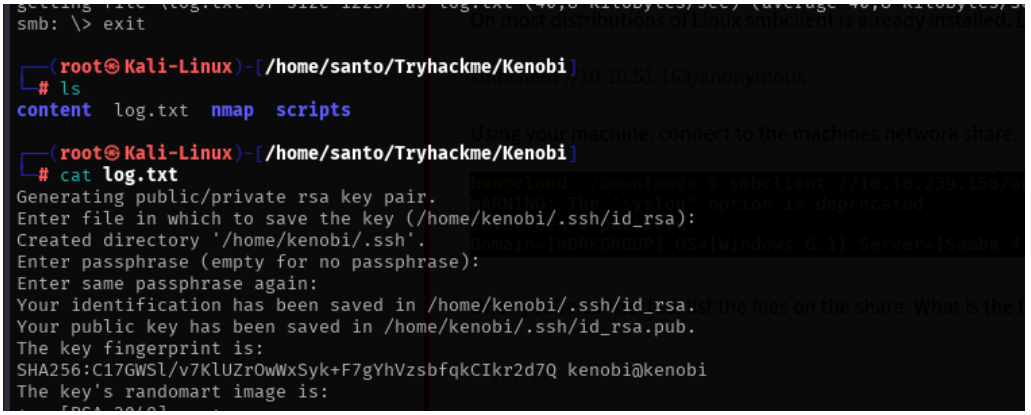
Aquí con smbclient nos intentamos conectar a la maquina por el recurso que hemos visto anteriormente el de Anonymous con el comando (-N) le indicamos que queremos hacer una Null Session



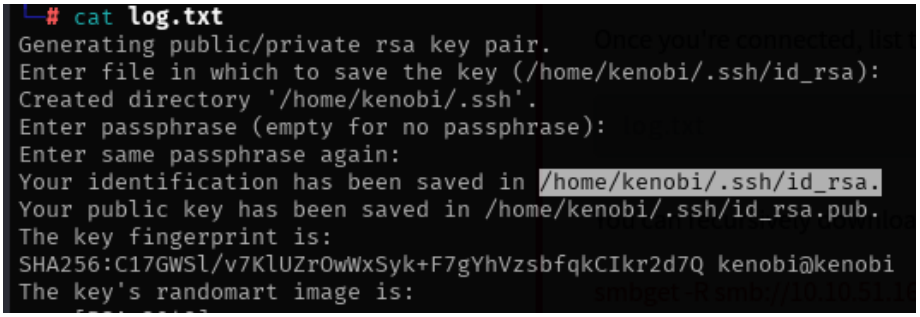
> 💡 Que es una Null Session?
>
> Una Null Session es un tipo de conexión anónima a un sistema Windows que permite a un atacante autenticarse sin necesidad de credenciales validas.

Una vez ya estemos dentro podremos ejecutar comando, como podemos observar hay un archivito .txt así que vamos a bajárnoslo con el siguiente comando

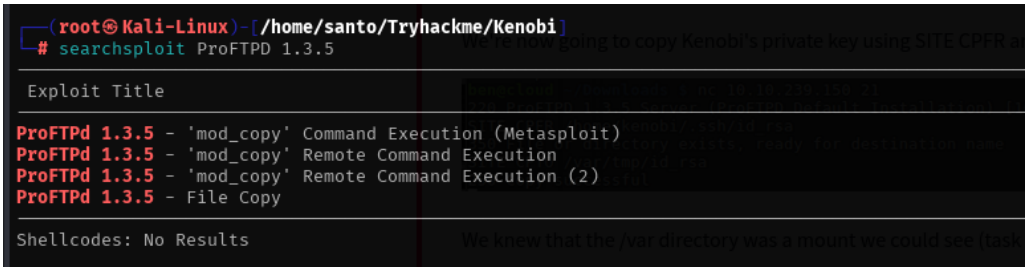
Ahora salgo de la shell y nos procedemos acceder a el archivito



Y al espectar un poco podemos ver que hay un usuario llamado `Kenobi`, y podemos ver que tiene un `id_rsa` .

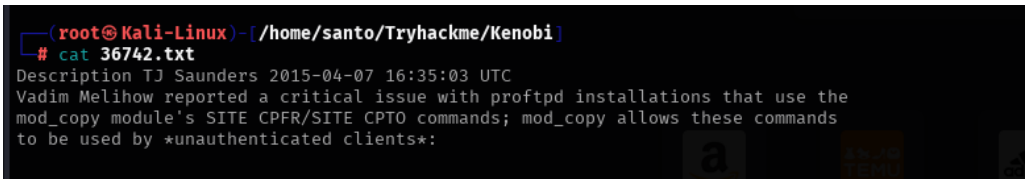


Bien entonces ya sabemos que tenemos un usuario que se llama Kenobi, ahora vamos abrir otra vez el reporte de nmap, y como podemos observar tenemos el puerto 21 abierto con una versión un poco vieja, así que vamos hacer uso de searchsploit y vamos a buscar a ver si hay vulnerabilidades reportadas para estos puertos



De estos exploit que hay vamos a escoger el ultimo, para descargárnoslo solo basta con escribir el parámetro (-m) y ya se nos descargaría, y así es como ya lo tendríamos en nuestra maquina, listo para ser ejecutado

Ahora nos vamos abrir el archivo e intentar entenderlo



> 💡 Explicación de esta vulnerabilidad:
>
> ProFTPD es un servidor FTP muy utilizado en linux. En la versión afectada (`1.3.5rc3`), el modulo `mod_copy` permite a los atacantes no autenticados copiar archivos arbitrarios dentro del sistema de archivos del servidor, lo que puede llevar a una escalada de privilegios o ejecución de código remoto (RCE).

El problema es que los comandos **SITE CPFR (copy from) y SITE CPTO (copy to)** permiten a un atacante:

1. **Leer archivos sensibles**, como `/etc/passwd` (donde se almacenan los usuarios del sistema).
2. **Escribir archivos en directorios accesibles por el servidor web**, lo que puede permitir la ejecución de código malicioso.

Si nos dirigimos a el esca neo que hicimos previamente a el puerto 111 podemos ver que tiene monturas

📌 Explicación de las "Monturas" (Mounts) en NFS

En el contexto de NFS (Network File System), una “montura” es un directorio compartido en un servidor que puede ser accedido por otros dispositivos en la red como si fuera una carpera local.

Básicamente, es un **recurso compartido** que otro sistema puede "montar" (conectar) en su propio sistema de archivos. Esto permite que varios usuarios o sistemas accedan a los mismos archivos de manera remota.

Tenemos un comando que nos dice la raíz de donde sale la montura, y es el siguiente

```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi/nmap]
└─# showmount -e 10.10.43.165
Export list for 10.10.43.165:
/var *
```

En esta maquina existe una montura dentro del directorio /var

---

Entonces tenemos estos dos casos, tenemos por un lado el servicio ProFTPD 1.3.5 que nos permite copiar archivos internos y luego por otro lado tenemos una montura en el puerto 111 dentro del directorio `/var`. Así que si nosotros conseguimos copiarnos el `id_rsa` (que es la clave privada de SSH) Luego podría montármelo en mi sistema y por tanto acceder dentro de la maquina por vía SSH.

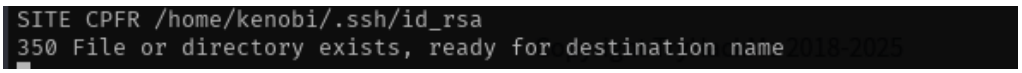
Ahora vamos a copiar la clave privada de Kenobi usando comandos SITE CPFR y SITE CPTO, que vimos anteriormente en el exploit que nos descargamos

```
└─# cat 36742.txt
Description TJ Saunders 2015-04-07 16:35:03 UTC
Vadim Melihow reported a critical issue with proftpd installations that use the
mod_copy module's SITE CPFR/SITE CPTO commands; mod_copy allows these commands
to be used by *unauthenticated clients*:

_____________________________

Trying 80.150.216.115 ...
Connected to 80.150.216.115.
Escape character is '^]'.
220 ProFTPD 1.3.5rc3 Server (Debian) [::ffff:80.150.216.115]
site help
214-The following SITE commands are recognized (* =>'s unimplemented)
214-CPFR <sp> pathname
214-CPTO <sp> pathname
214-UTIME <sp> YYYYMMDDhhmm[ss] <sp> path
214-SYMLINK <sp> source <sp> destination
214-RMDIR <sp> path
```

Así que lo primero que hacemos es con netcat por el puerto 21 en el cual corre el servicio vulnerable, conectarnos a la maquina

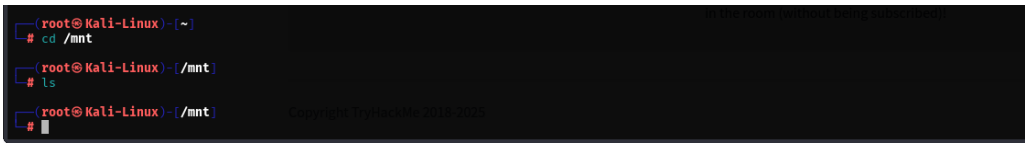
```
┌──(root㉿Kali-Linux)-[/home/santo/Tryhackme/Kenobi]
└─# nc 10.10.43.165 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.43.165]
```

Ahora ponemos el siguiente comando para que nos encuentre el id_rsa (la ubicación del id_rsa la tenemos en el log que nos descargamos anteriormente)

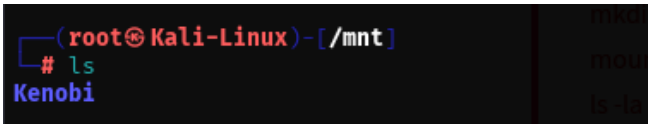


Y lo vamos a pasar a nuestra carpeta de nuestra maquina de la siguiente forma

Ahora vamos a la raíz de el sistema y una vez estemos aquí vamos a ubicarnos en el directorio `/mnt`

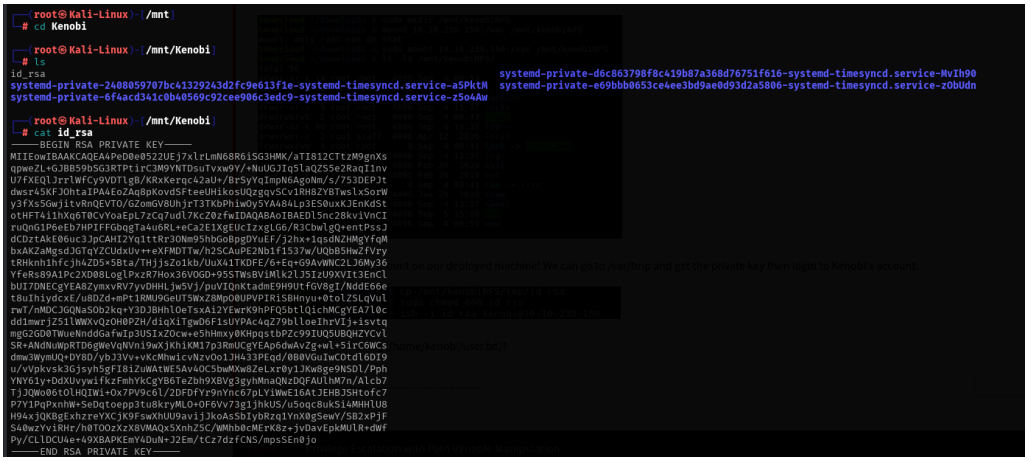


Ahora nos creamos una carpeta, en mi caso yo la llame como el nombre de la maquina Kenobi



Ahora mismo una vez creado el directorio este que acabamos de crear, le vamos a decir a la maquina que con el comando `mount` y la ip de la maquina victima, le decimos “como yo tengo a la maquina desde este directorio ahora móntame todo esto a esta carpeta

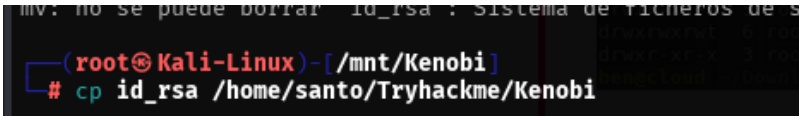
Entonces ahora mismo deberíamos tener dentro de la carpeta el id_rsa aquí ya en nuestro poder, como se puede apreciar aquí en la imagen



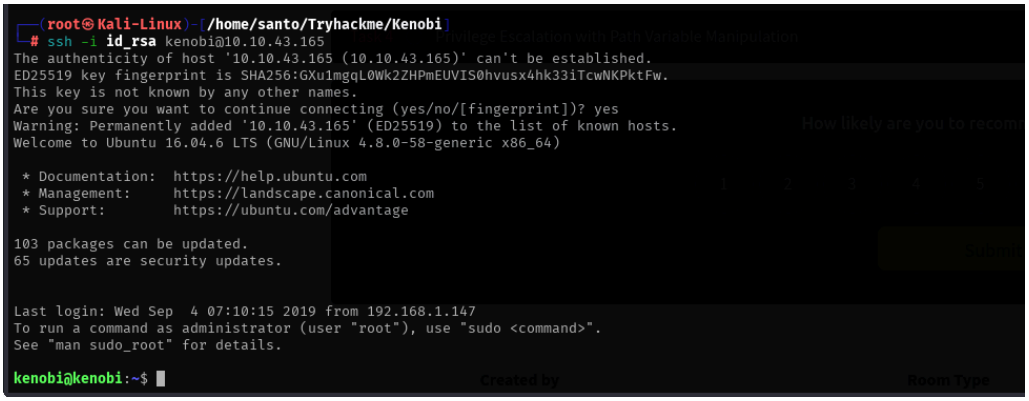
Aquí lo que hicimos fue que básicamente a través de un fallo de seguridad de seguridad en ProFTPD en la versión 1.3.5 pues pudimos copiar el id_rsa a al ubicación que le indicamos, ahora lo que sigue es como ya tenemos copiado este archivo ahora vamos a copiar todo lo que esta en el directorio /var a mi maquina local

---

Vamos a darle permisos a nuestro id_rsa, para ello tenemos que copiarlo de la ubicación que se encuentra ahora mismo ya que aquí esta dentro de la montura, por ende solo tenemos permisos de lectura



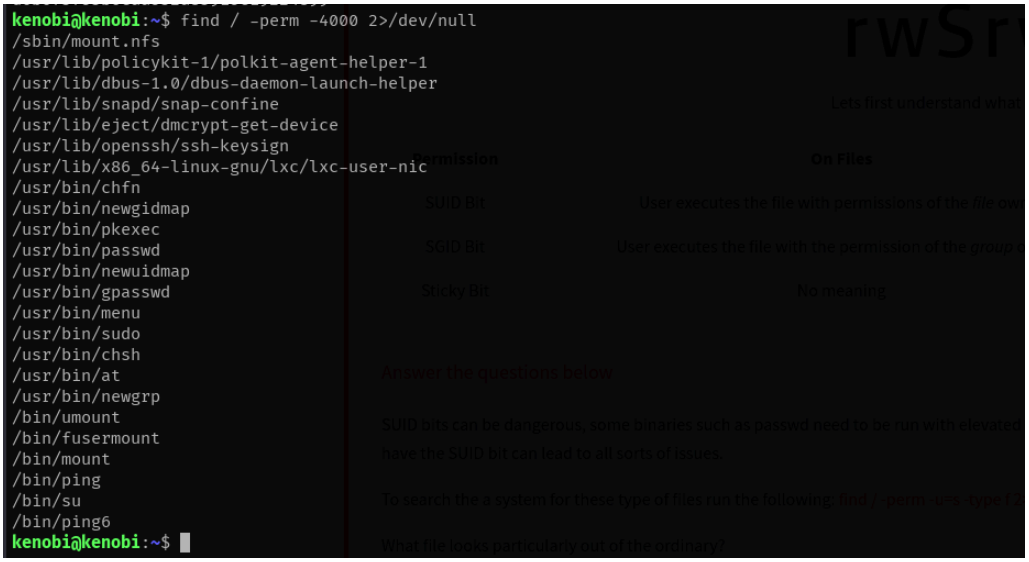
Ahora como ya tenemos el id_rsa que es la clave privada de SSH pues ahora podríamos establecer una conexión por el servicio SSH, esto lo hacemos de la siguiente manera



Ahora ya estamos dentro

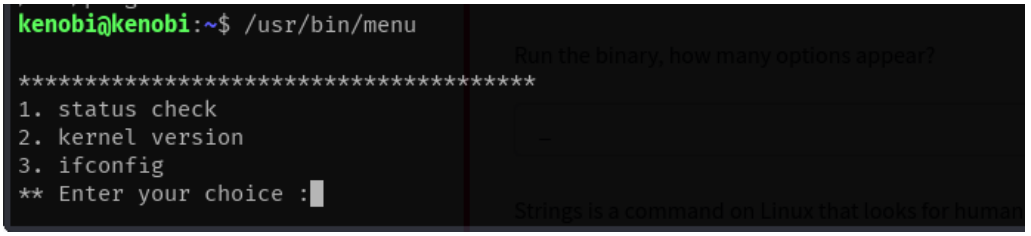
ESCALA DE PRIVILEGIOS o USMEO EN EL SISTEMA

Vamos a buscar en el sistema los binarios que tengan permisos como root o permisos aceptables para poder así escalar privilegios

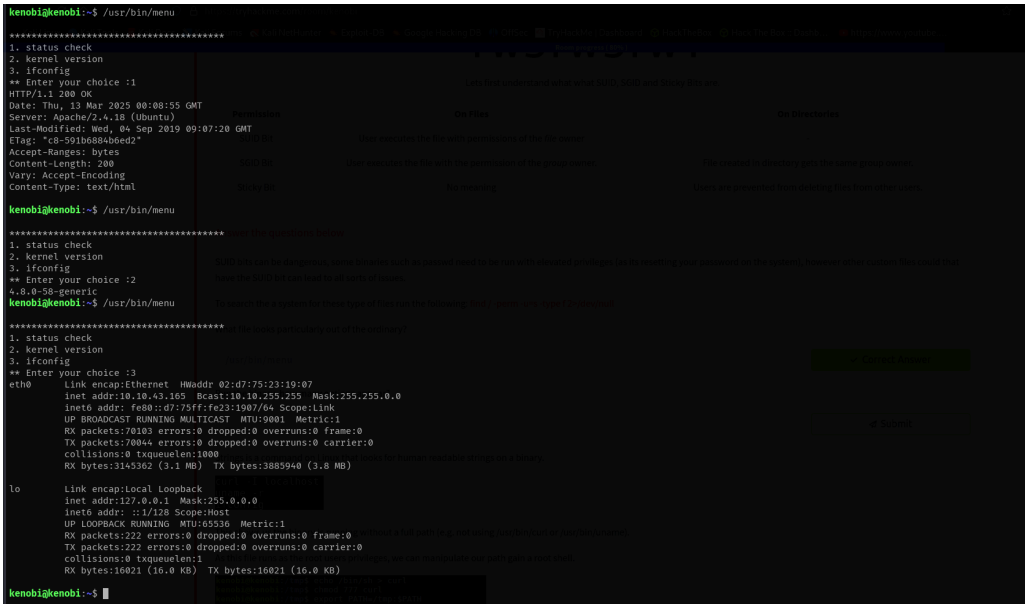


Como podemos observar vemos un binario que tiene privilegios que es muy raro esta fuera de lo común el de (`/usr/bin/menu`)

Asi que vamos a ejecutarlo



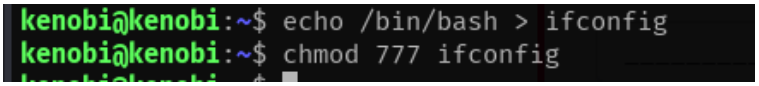
Es como una aplicación que ejecuta comandos



Y fijaros en una cosa, todo esta se esta ejecutando como root.

Así que si nosotros modificamos el programita podríamos ejecutar como root todos los comando que le pongamos, pero en vez de meter un (`ifconfig`) voy a meter un (`bin/bash`)

Así que vamos hacer un echo y todo esto lo vamos a enviar a un archivo, en este caso a program y le damos todo tipo de permisos

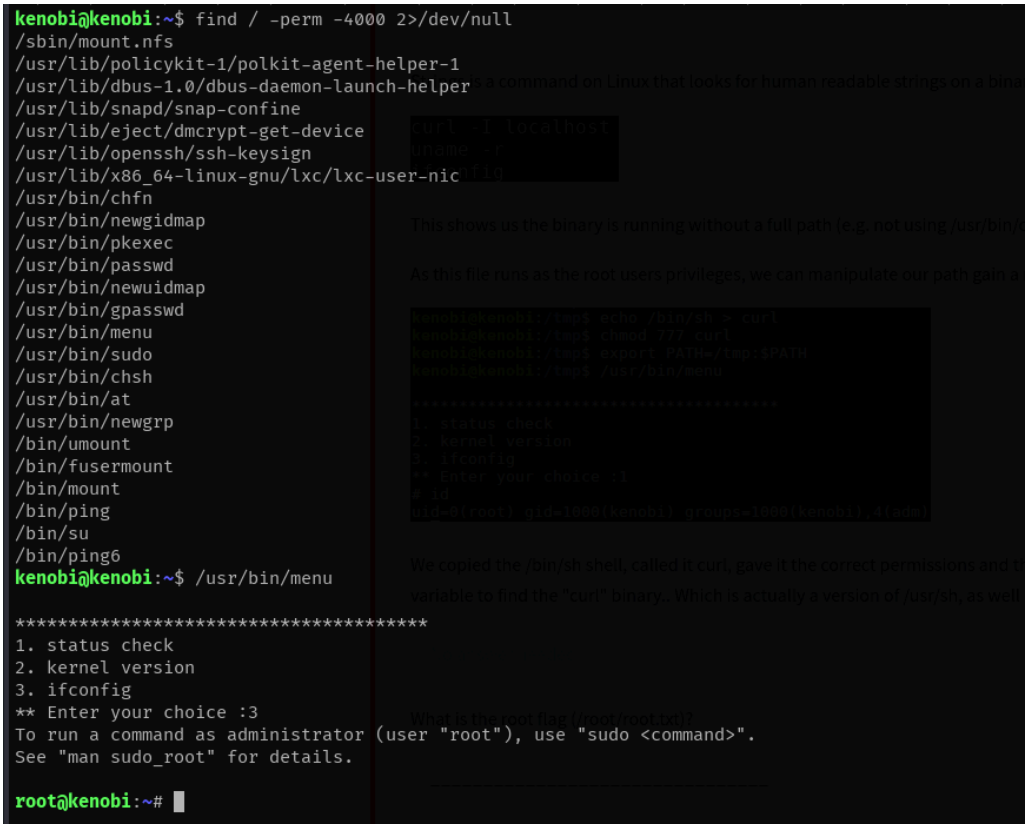


Y lo exportamos a el $PATH de linux

Para no ponerle el directorio en donde lo hemos creado solo con ponerle un . le indicamos que es en el directorio actual donde nos encontramos

Entonces cuando el programa vaya a buscar el comando ifconfig no lo va a encontrar la que hemos creado uno con el mismo nombre y ejecutara el nuestro.

Entonces volvemos a ejecutar el programa



Y así es como ya seriamos usuario root