- Primero, con el comando ping, revisaremos si existe conexión con la máquina vulnerable por medio de envio de paquetes ICMP

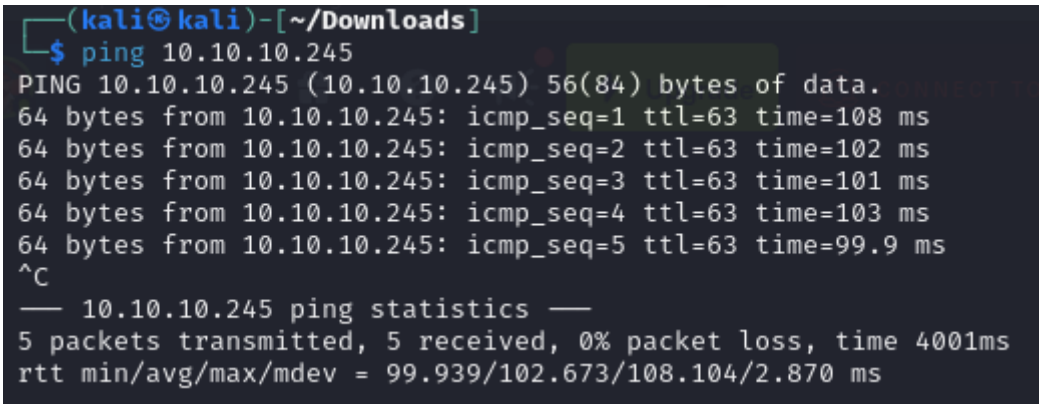


Figura 01. ping

Podemos observar que 5 paquetes son enviados y 5 recibidos, lo que quiere decir que la máquina vulnerable está activamente recibiendo conexiones, tambien nos damos cuenta que nos enfrentamos a una maquina linux

- Ahora, por medio del uso de la herramienta nmap, realizaremos una enumeración de puertos y servicios para identificar algunos vectores de ataque

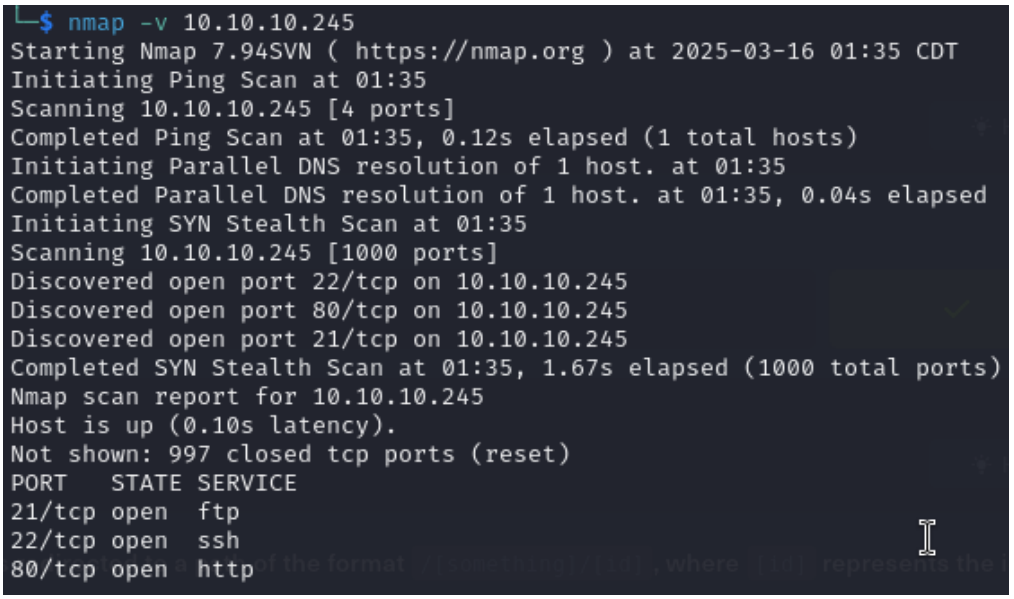


Figura 02. Escaneo inicial

Acá, realizamos un escaneo básico al host 10.10.10.245, en el que podemos identificar 3 puertos abiertos con el servicio que están utilizando. Me genera curiosidad el puerto 80, asi que introduciremos su dirección ip en nuestro navegador para ver de qué se trata

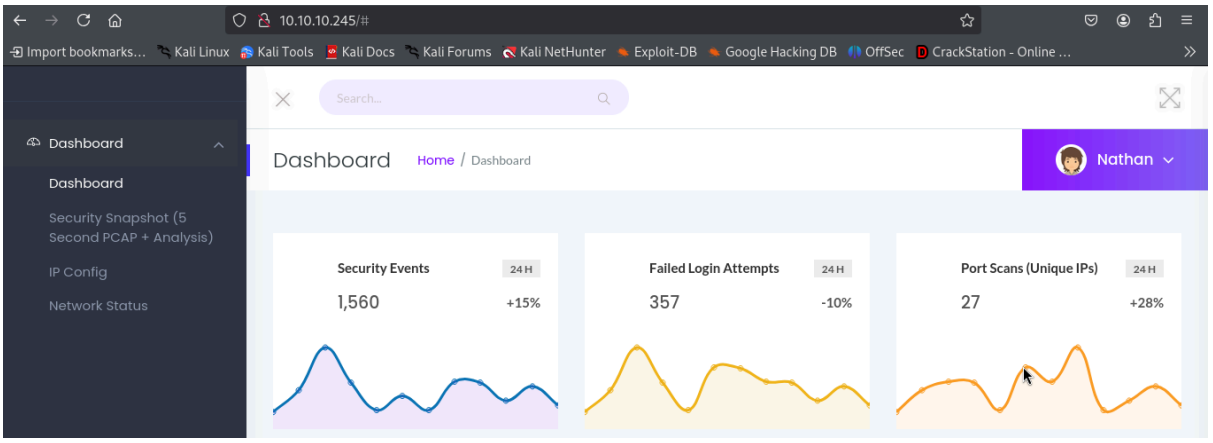


Figura 03. Revisión puerto 80

Podemos ver que se trata de un dashboard que muestra distintos eventos de seguridad por medio del monitoreo de resultados

- Si nos dirigimos al apartado de Security Snapshot, esta es una opción que ofrece la descarga de archivos PCAP. PCAP es un formato común para almacenar captura de paquetes

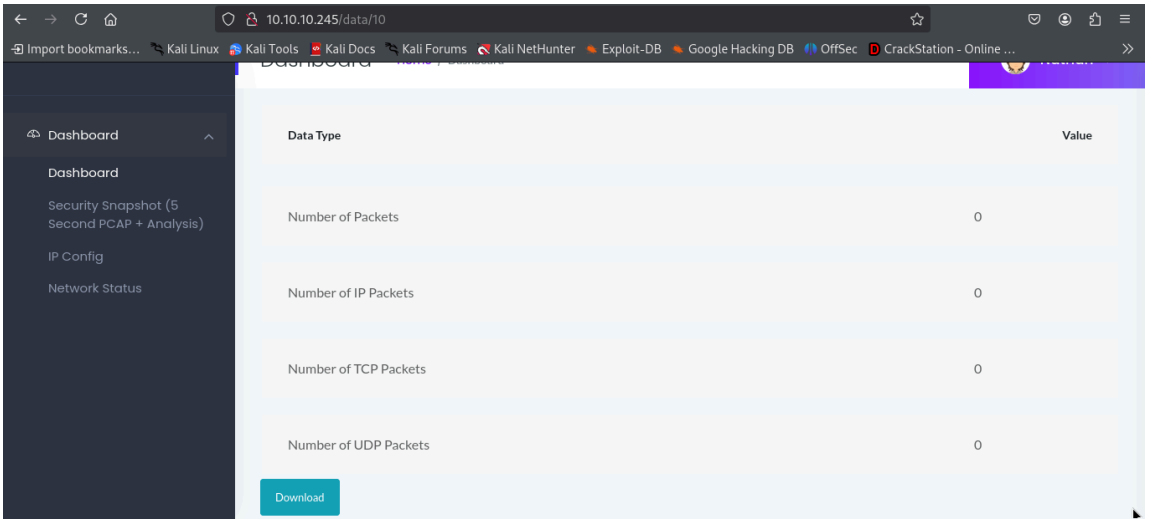


Figura 04. PCAP files

- También contamos con el apartado de ip, en el que por debajo corre el comando **ifconfig** y nos muestra información de la interfaz de red

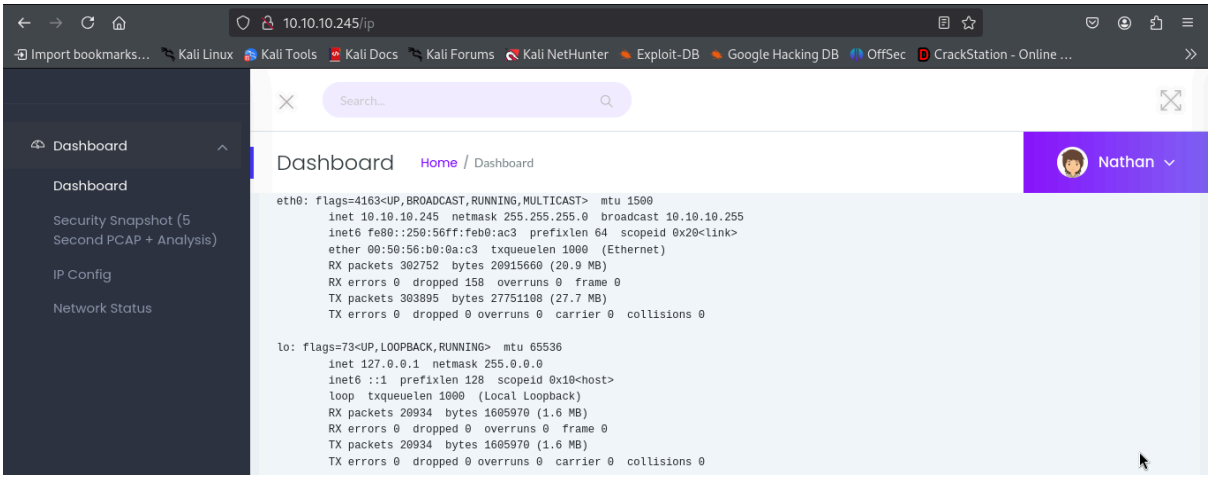


Figura 05. ip-ifconfig

- También se puede ver el estado de red, que nos muestra información del comando **netstat,** revelando así, detalles de conexiones y puertos activos

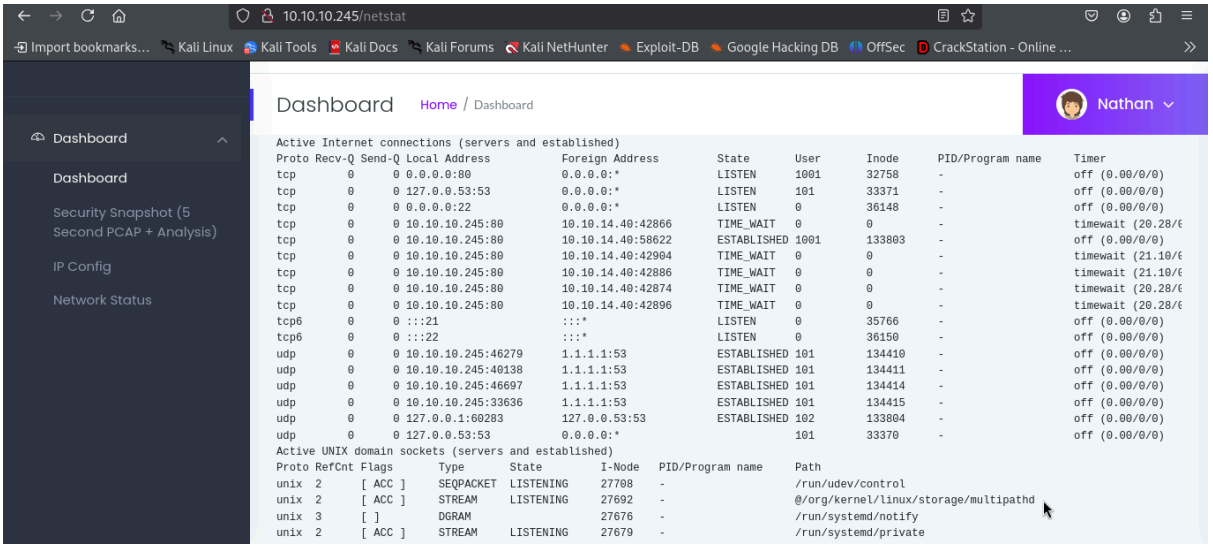


Figura 06. netstat

- Ahora, volviendo al tema de los archivos PCAP, vemos que al Descargar un archivo PCAP, se realiza una petición GET al endpoint /download

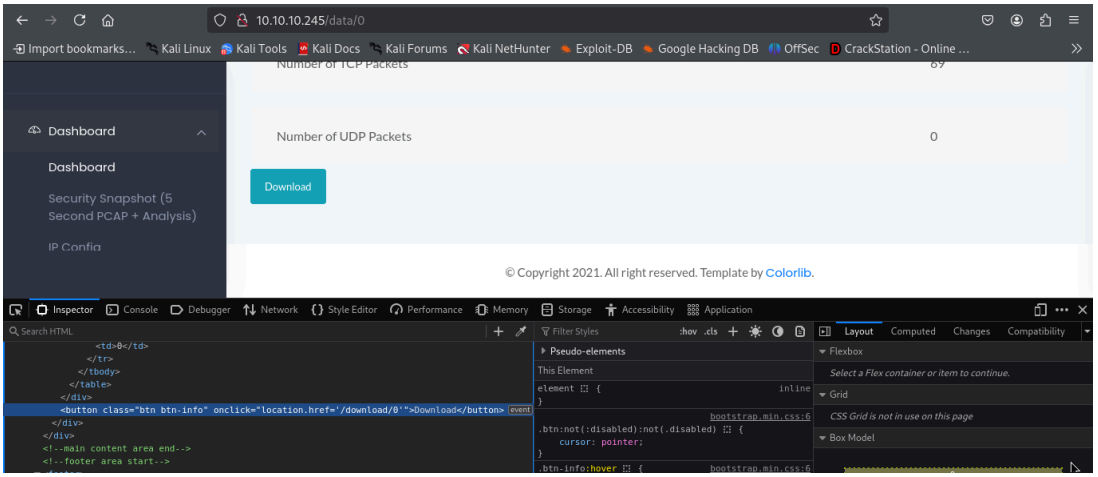


Figura 07. endpoint /download

- Descargamos el archivo PCAP y lo abriremos con nuestra herramienta wireshark para analizarlo un poco

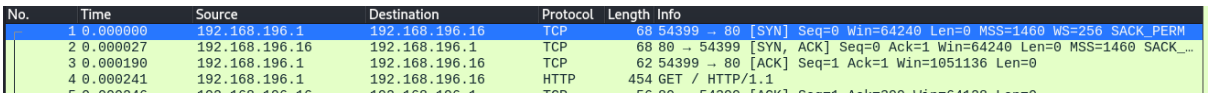


Figura 08. 3-way handshake

Podemos observar un proceso de transmisión de paquetes 3-way handshake entre las direcciones **192.168.196.1 y 192.168.196.16**

- Realizaremos una búsqueda de la palabra “password”

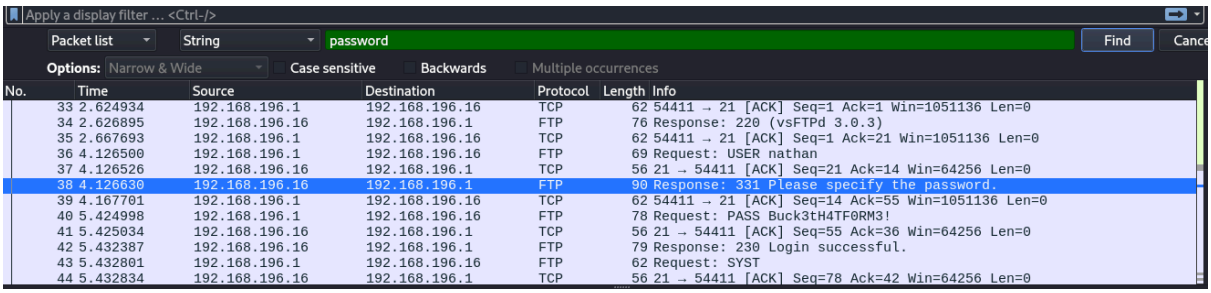
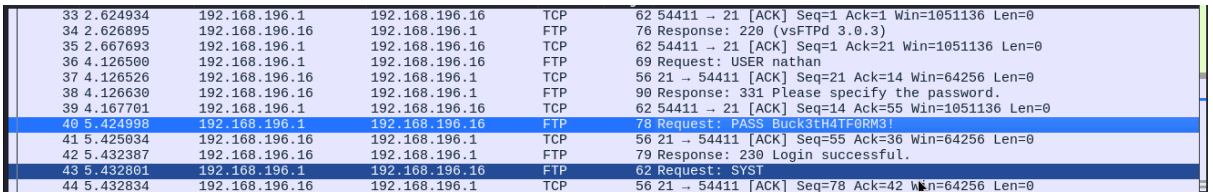


Figura 09. “password”

Podemos ver que hay una coincidencia por medio de una conexión FTP

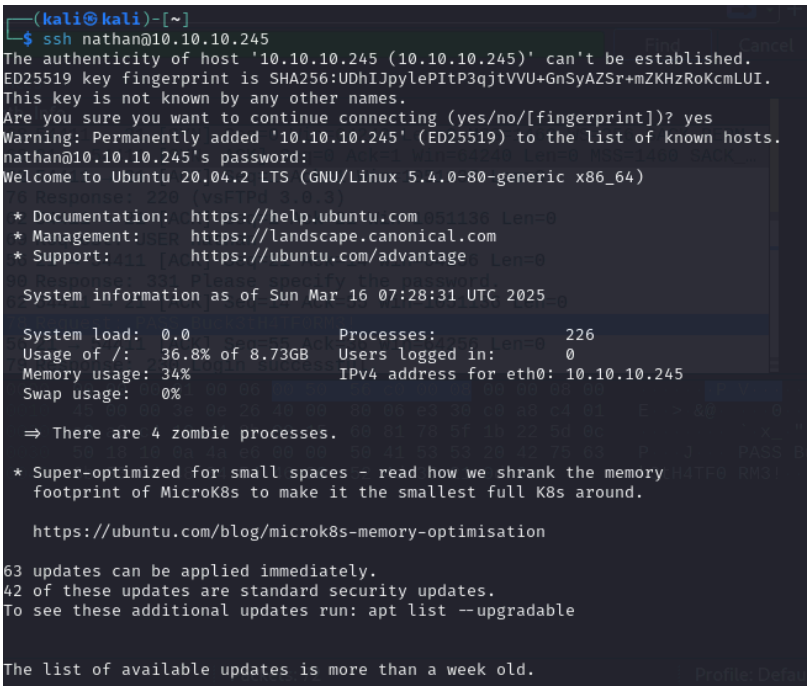
- Si analizamos un poco más, podremos observar la línea “PASS Buck3tH4TF0RM3!”, donde posiblemente sea la contraseña del usuario Nathan



33	2.624934	192.168.196.1	192.168.196.16	TCP	62 54411 → 21 [ACK] Seq=1 Ack=1 Win=1051136 Len=0
34	2.626895	192.168.196.16	192.168.196.1	FTP	76 Response: 220 (vsFTPd 3.0.3)
35	2.667693	192.168.196.1	192.168.196.16	TCP	62 54411 → 21 [ACK] Seq=1 Ack=21 Win=1051136 Len=0
36	4.126500	192.168.196.1	192.168.196.16	FTP	69 Request: USER nathan
37	4.126526	192.168.196.16	192.168.196.1	TCP	56 21 → 54411 [ACK] Seq=21 Ack=14 Win=64256 Len=0
38	4.126630	192.168.196.16	192.168.196.1	FTP	90 Response: 331 Please specify the password.
39	4.167701	192.168.196.1	192.168.196.16	TCP	62 54411 → 21 [ACK] Seq=14 Ack=55 Win=1051136 Len=0
40	5.424998	192.168.196.1	192.168.196.16	FTP	78 Request: PASS Buck3tH4TF0RM3!
41	5.425034	192.168.196.16	192.168.196.1	TCP	56 21 → 54411 [ACK] Seq=55 Ack=36 Win=64256 Len=0
42	5.432387	192.168.196.16	192.168.196.1	FTP	79 Response: 230 Login successful.
43	5.432801	192.168.196.1	192.168.196.16	FTP	62 Request: SYST
44	5.432834	192.168.196.16	192.168.196.1	TCP	56 21 → 54411 [ACK] Seq=78 Ack=42 Win=64256 Len=0

Figura 10. Credencial de acceso

- Ahora, estableceremos una conexión ssh con las credenciales de acceso encontradas para ver si tenemos éxito o no



```
┌──(kali㉿kali)-[~]
└─$ ssh nathan@10.10.10.245
The authenticity of host '10.10.10.245 (10.10.10.245)' can't be established.
ED25519 key fingerprint is SHA256:UDhIJpylePItP3qjtVVU+GnSyAZSr+mZKHzRoKcmLUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.245' (ED25519) to the list of known hosts.
nathan@10.10.10.245's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Mar 16 07:28:31 UTC 2025

  System load:  0.0               Processes:             226
  Usage of /:   36.8% of 8.73GB   Users logged in:       0
  Memory usage: 34%               IPv4 address for eth0: 10.10.10.245
  Swap usage:   0%

  ⇒ There are 4 zombie processes.

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

63 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
```

Figura 11. conexión ssh

Al brindar las credenciales de acceso, logramos tener acceso al sistema

- Ahora, usaremos un script llamado LinPEAS, para descubrir posibles rutas que nos permitan escalar privilegios

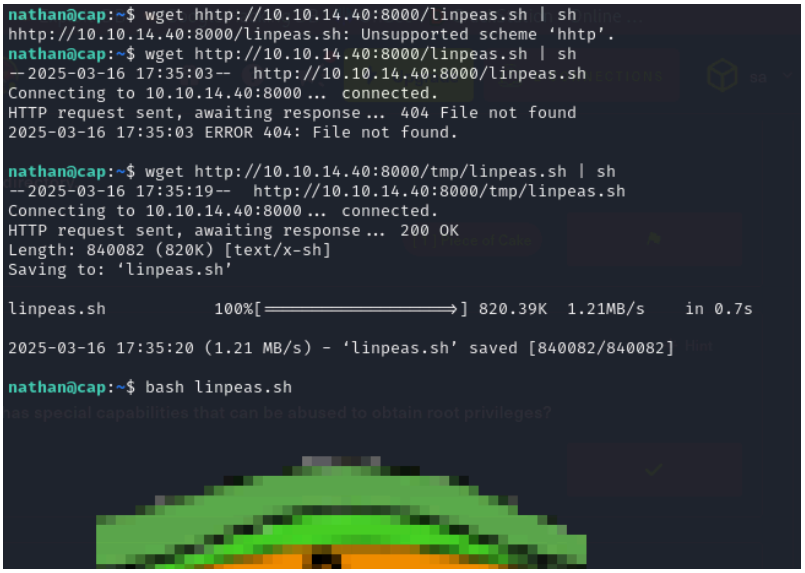


Figura 12. Ejecucion linpeas.sh

- El escaneo con Lineas nos muestra que la ruta /usr/bin/python3.8 tiene el cap_setuid con la capacidad habilidad para escalar privilegios

FIgura 13. Ruta con capacidades de escalar priv

- Buscaremos en GTFOBins si se puede ser explotada esta vulnerabilidad

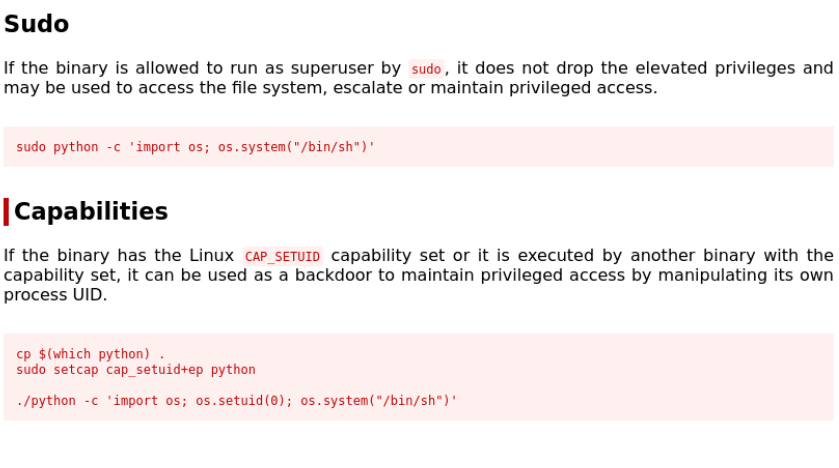


Figura 14. GTFOBins

Podemos ver, que podremos utilizar el setuid como un backdoor para mantener el acceso privilegiado, manipulando su propio proceso

- Ahora ejecutaremos el comando con python3 ya que es el que se encuentra disponible

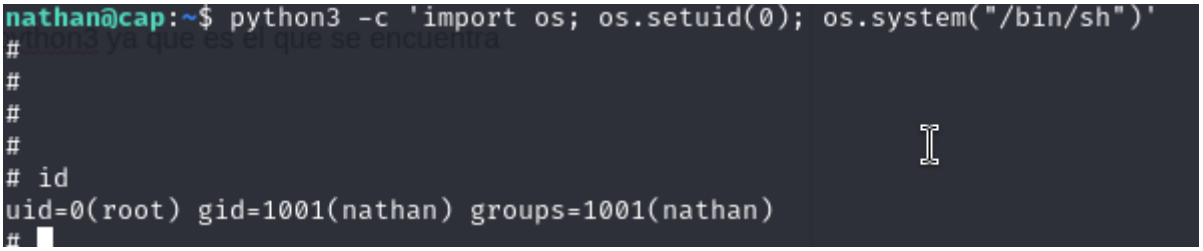


Figura 15. privilegios root

y así, ya tendremos acceso a privilegios root