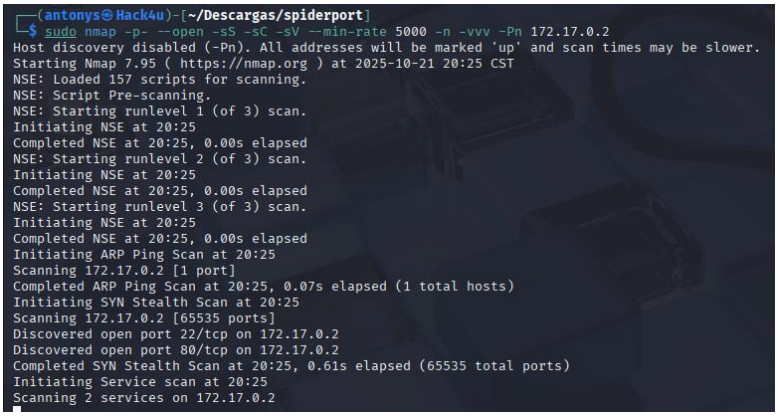
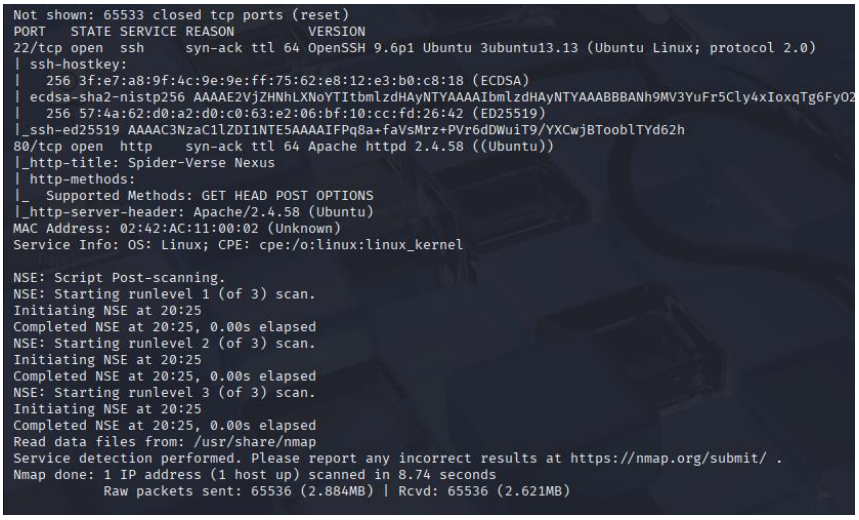
Maquina SpiderPort

Comanezamos montando la maquina.

Con nmap, realizamos un escaneo de puertos de nuestra maquina victima.

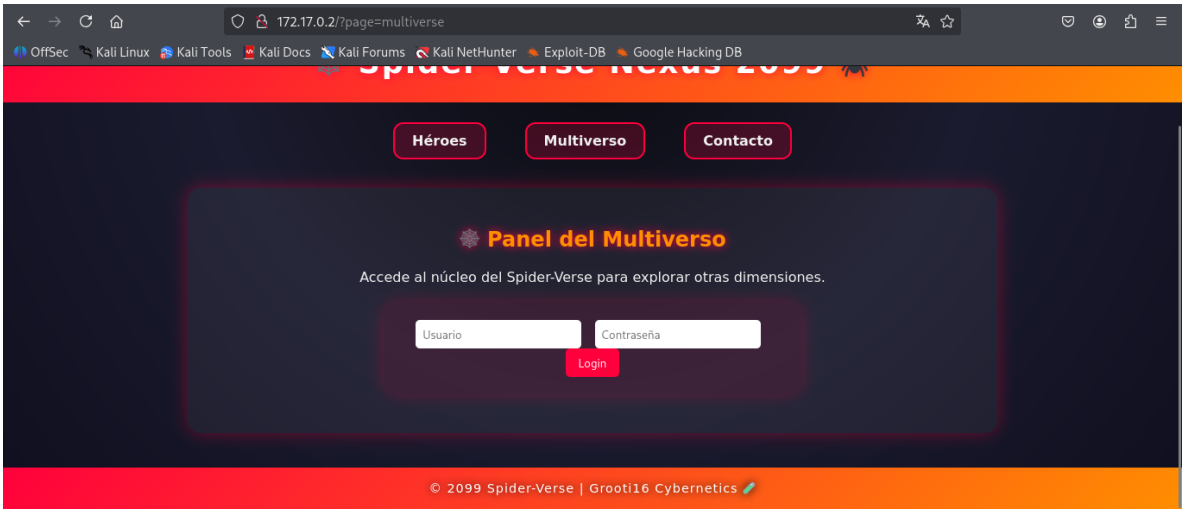


Aquí podemos ver que los puertos 80 para servicios web y 22 para conexiones remotas se encuentran abiertos.



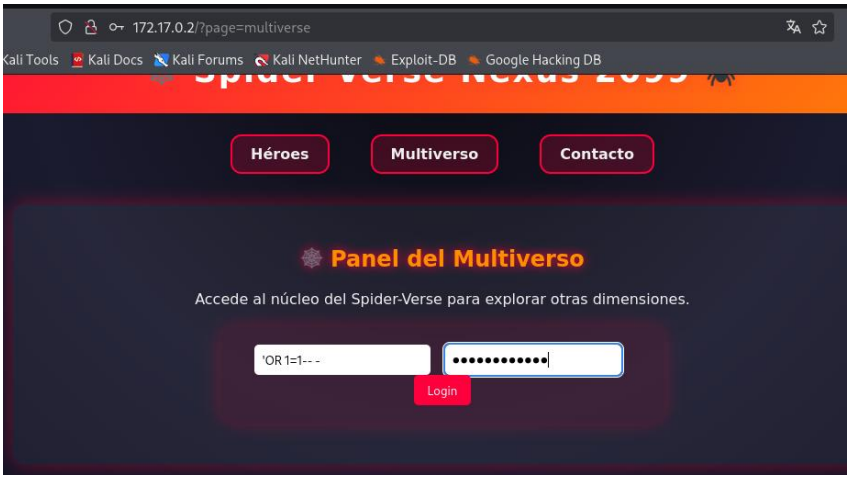
Si accedemos al puerto 80 nos muestra este panel.

Explorando el panel, podemos ver que hay un Login.



Y del otro lado podemos ver información.

Viendo este panel, lo primero que se nos ocurre es una inyeccion SQL.



Pero podemos ver que el WAF nos bloquea el comando.

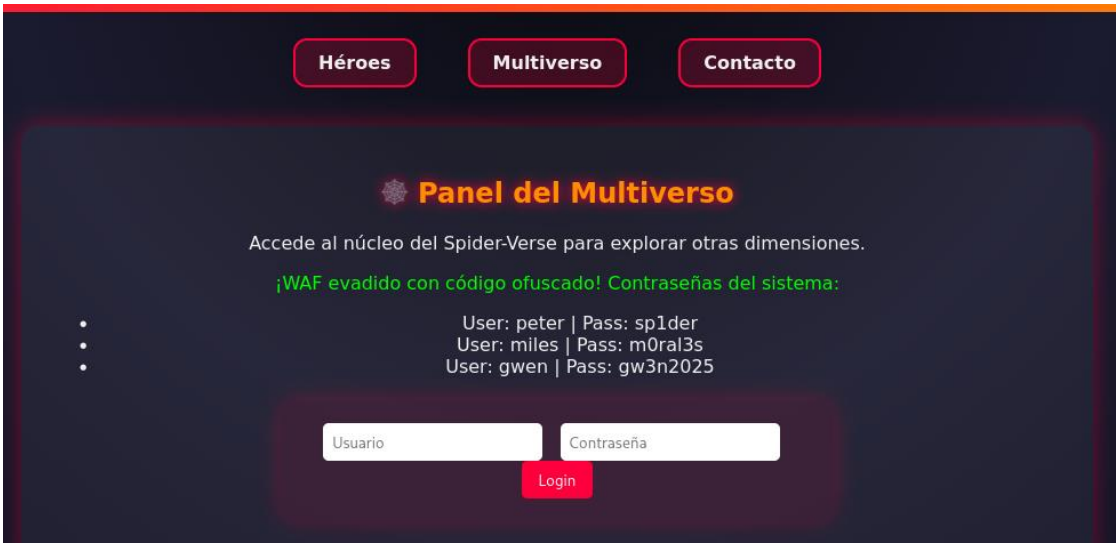
En seguridad informática, un payload es el contenido útil que se envía a una aplicación web con un propósito específico.

Un WAF (Web Application Firewall) detecta patrones comunes de ataque (como ' OR 1=1 --) y los bloquea.
Entonces, para que el payload pase sin ser detectado, podemos modificarlo con mayúsculas, minúsculas, espacios y evitar los caracteres para que el WAF no los pueda detectar.

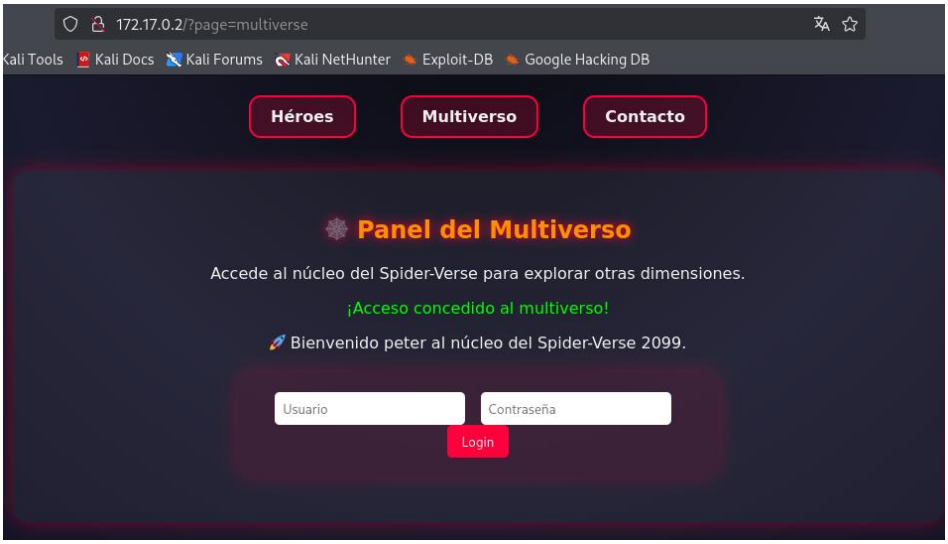
Aquí podemos ver un or 1= 1

Esta completamente modificado y el WAF no lo detecta como un script malicioso común.

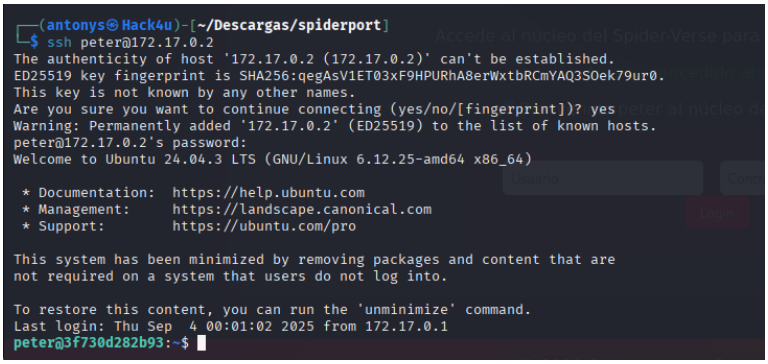
Y vemos que nos puede dar acceso pero nos arroja los usuarios 😊



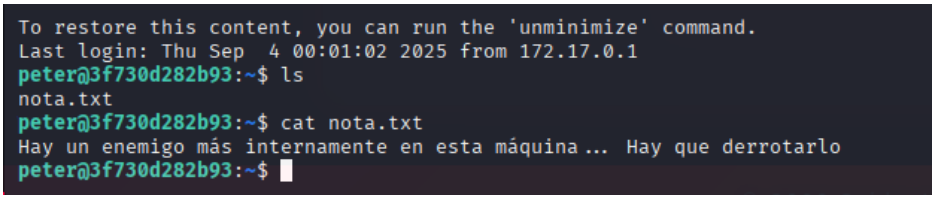
Probando con los diferentes usuarios vemos que nos lleva al mismo sitio y no da para algo más.



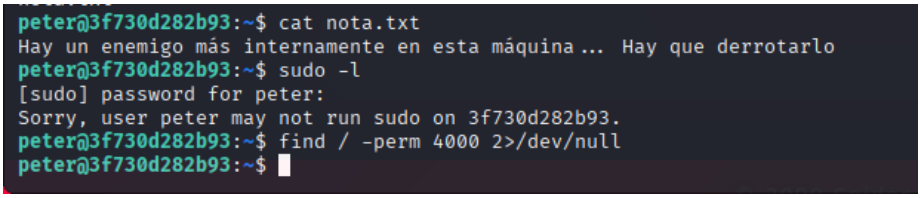
Procedemos a realizar la conexión remota por SSH, con el usuario Peter.



Una vez dentro, podemos ver que hay un archivo de texto que nos puede dar una pista.

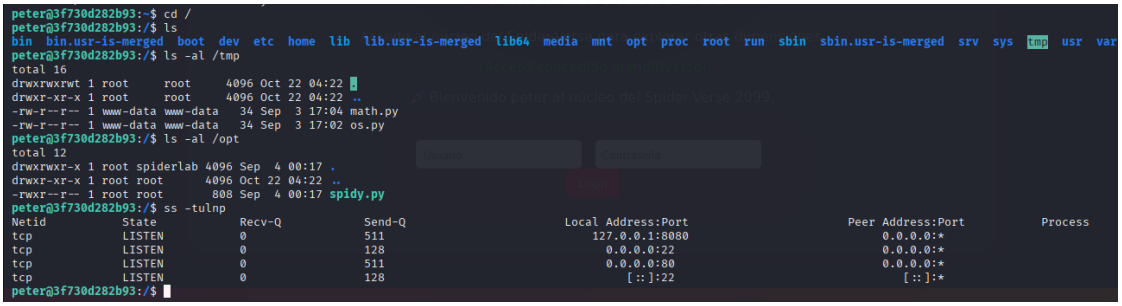


Explorando el directorio vemos que no encontramos nada útil.



Analizamos los directorios mas comunes para checar archivos importantes en este caso TMP y OPT, logramos encontrar archivos pero solo los puede ejecutar root y nosotros no podemos ejecutar nada siendo root por permisos mas elevados ☹

Así que procedemos a analizar si hay servicios corriendo dentro del usuario y podemos ver que si hay.



Vemos que desde la maquina remota hay un servicio local en el puerto 8080 corriendo, pero solo es accedido desde la maquina víctima.

Con la herramienta curl procedemos a extraer lo que corre ese servicio y vemos que es una pagina web.

Y por intuición podemos darnos la idea de que es un sistema de inserción de comandos.

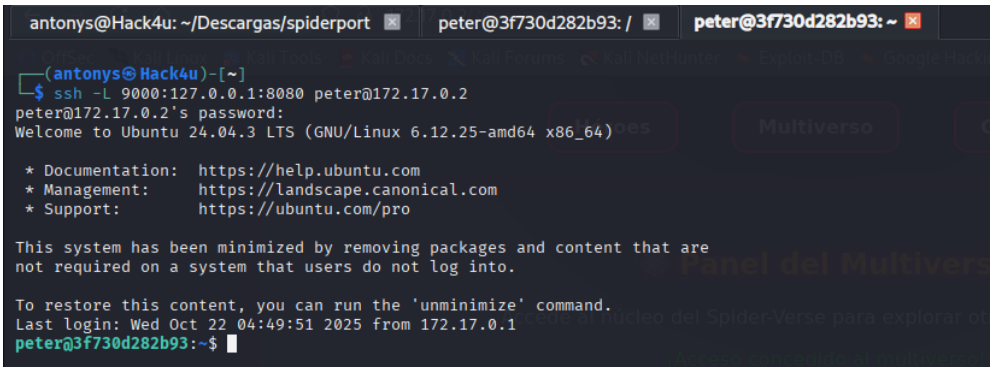
Asi que tenemos que hacer port forwarding para llegar a ese puerto abierto como sea.

El port forwarding (en español, reenvío de puertos o redirección de puertos) es una técnica de red que permite enviar conexiones que llegan a un puerto específico de un dispositivo (como un router o firewall) hacia otro dispositivo dentro de una red local.

Lo hacemos con este comando: ssh -L 9000:127.0.0.1:8080 peter@172.17.0.2

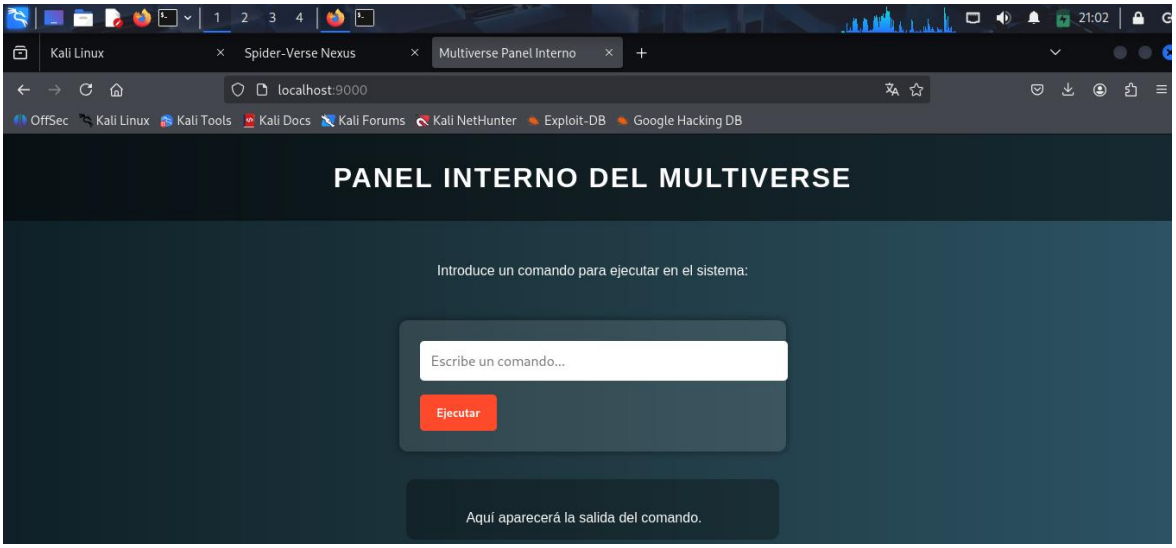
- ssh-->protocolo
- -L --> port forwarding
- 9000:127.0.0.1:8080 --> mi puerto 9000 es el 8080 del localhost (victima)
- peter@172.17.0.2 --> como quien nos conectamos por ssh
- puse el puerto 9000 en vez del mismo de la máquina (8080) por si tengo que usar burp suite que utiliza ese puerto y crea conflicto

y vemos que nos da otra dirección a donde mandar el trafico 😊



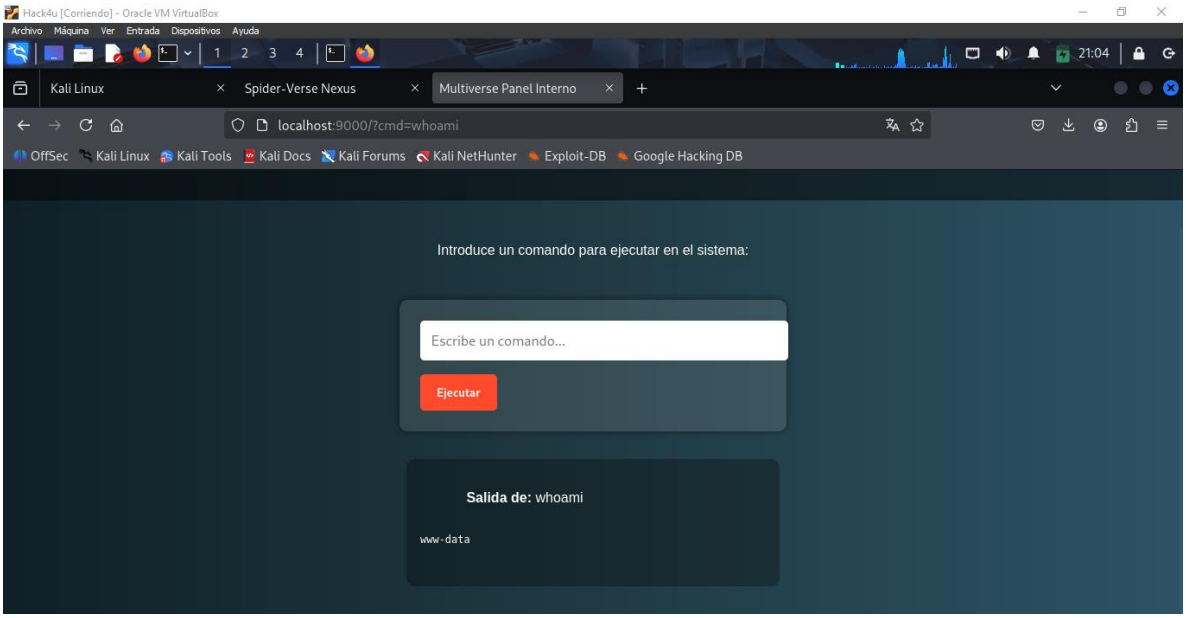
Nos conectamos con localhost en firefox desde nuestro puerto que abrimos en escucha en este caso el mío fue el 9000

Ya que recibimos el port forwarding 😊

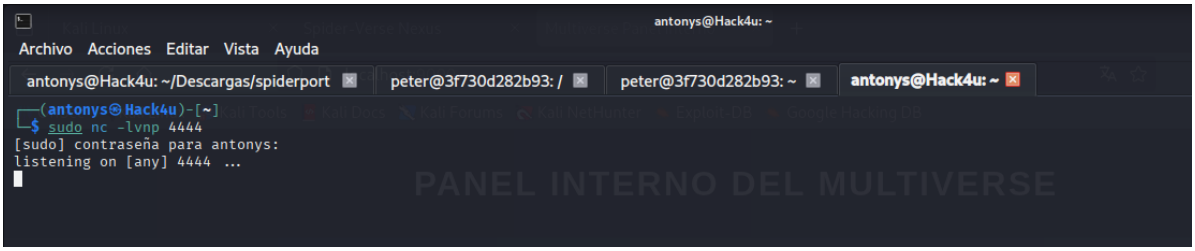


Ejecutamos el comando whoami y vemos que nos lo regresa 😊

Así que procedemos a hacer una reverse Shell.

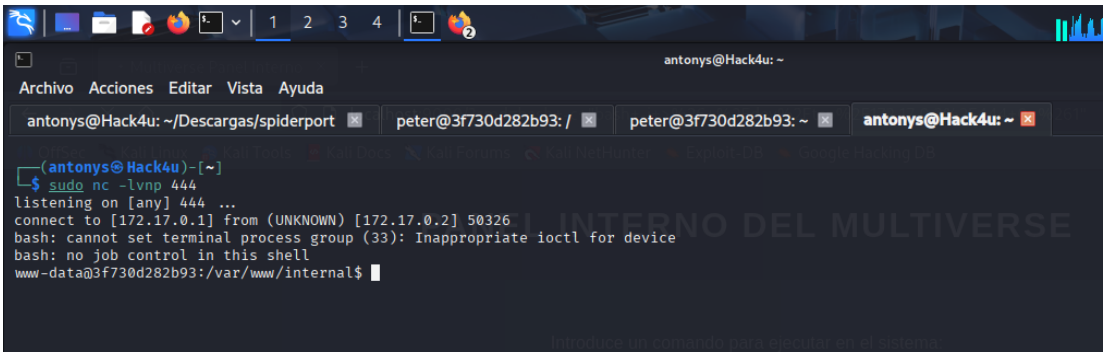


Nos ponemos en escucha con netcat, en el puerto que deseemos.

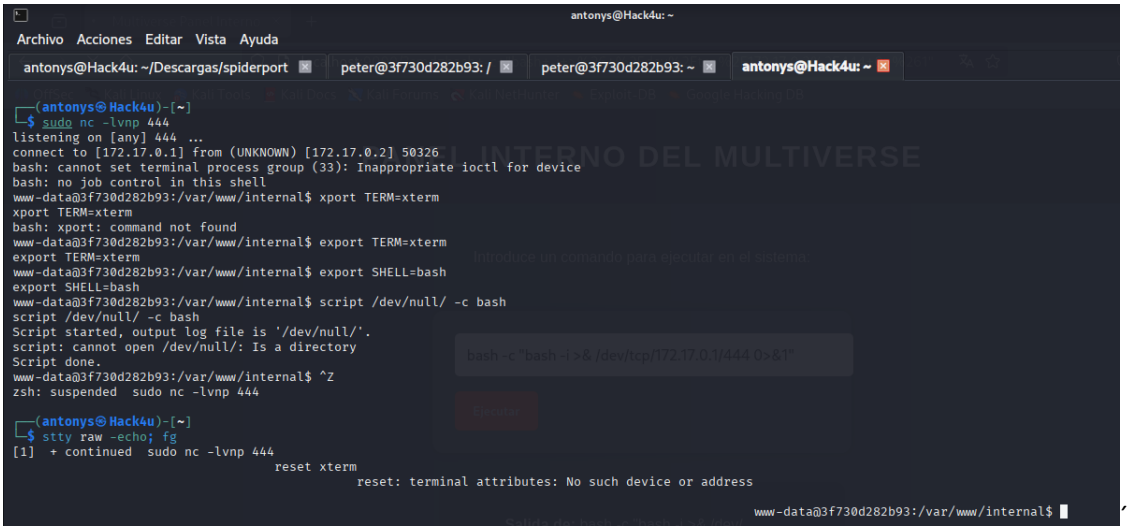


Y le insertamos una revershell, en el panel, con la nueva IP que nos dio.

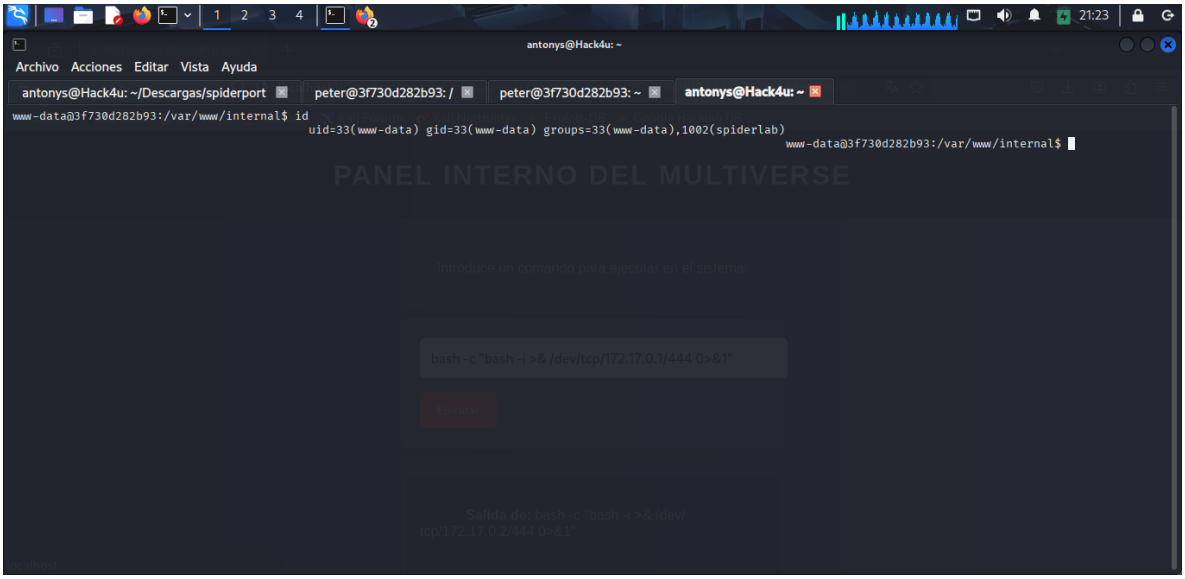
Ya nos dio nuestra Shell y podemos empezar a realizar la escalada de privilegios.



Realizamos el tratamiento de la TTY.



Y el primer comando que ejecutamos es id, para saber donde estamos y quienes somos.



Asi que procedemos a ver si hay archivos que podamos ejecutar, con sudo -l

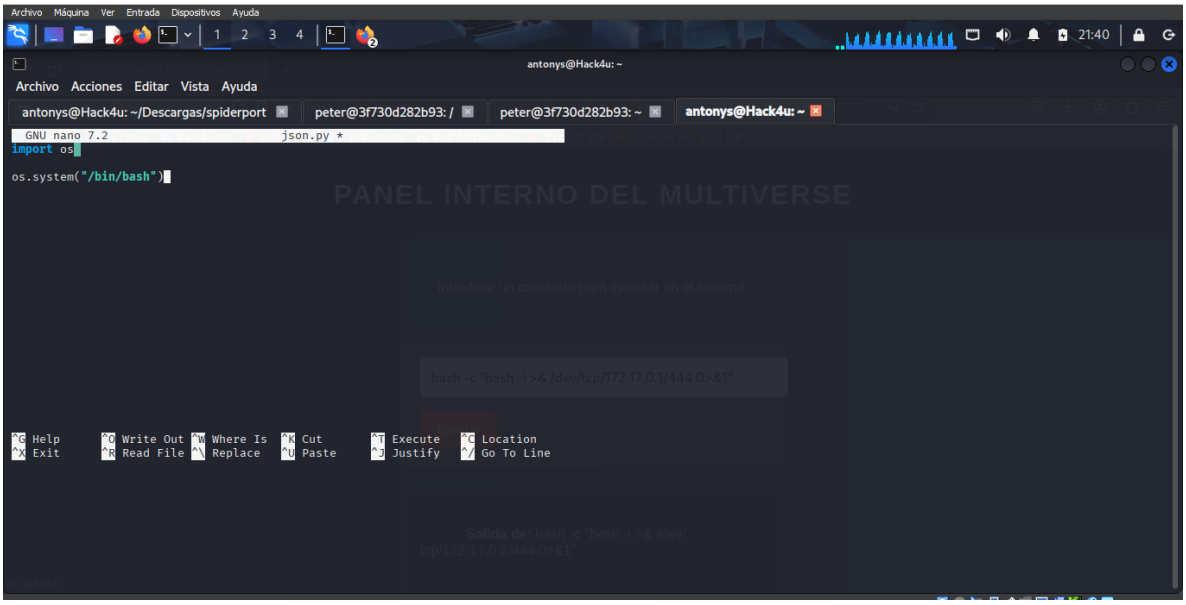
```
www-data@3f730d282b93:/$ sudo -l
Matching Defaults entries for www-data on 3f730d282b93:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User www-data may run the following commands on 3f730d282b93:
    (ALL) NOPASSWD: /usr/bin/python3 /opt/spidy.py
www-data@3f730d282b93:/$
```

Y explorando ese archivo de Python solo son impresiones de personajes, pero aun así no lo podemos modificar y vemos que le podemos modificar las librerías, en este caso la json.

```
import json
import math
def web_swing():
    print("🕸️ Spider-Man se balancea por la ciudad.")
    print("Explorando los tejados y vigilando la ciudad...")

def run_tasks():
    print("🕷️ Ejecutando tareas del día...")
    print("Saltos calculados:", math.sqrt(225))
    data = {"hero": "Spider-Man", "city": "New York"}
    print("Registro de datos:", json.dumps(data))

def fight_villains():
    villains = ["Green Goblin", "Doctor Octopus", "Venom"]
    print("Villanos en la ciudad:", ", ".join(villains))
    for v in villains:
        print(f"🕸️ Enfrentando a {v}...")

if __name__ == "__main__":
    web_swing()
    run_tasks()
    fight_villains()
    print("🕸️✨ Spider-Man ha terminado su ronda.")
www-data@3f730d282b93:/$
```

Asi que creamos un archivo con nano llamado json.py con un script para que nos dé una bash.

Para que cuando ejecutemos el archivo py pasado, ejecute primero la librería modificada y nos de la bash.



Una vez ejecutado podemos ver que nos hizo root.

```
www-data@3f730d282b93:/opt$ nano json.py
Unable to create directory /var/www/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

www-data@3f730d282b93:/opt$ sudo /usr/bin/python3 /opt/spidy.py
root@3f730d282b93:/opt#
```

Y podemos ver la flag 😊