Cap

Penetration Testing Writeup

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

Índice

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 escalação 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

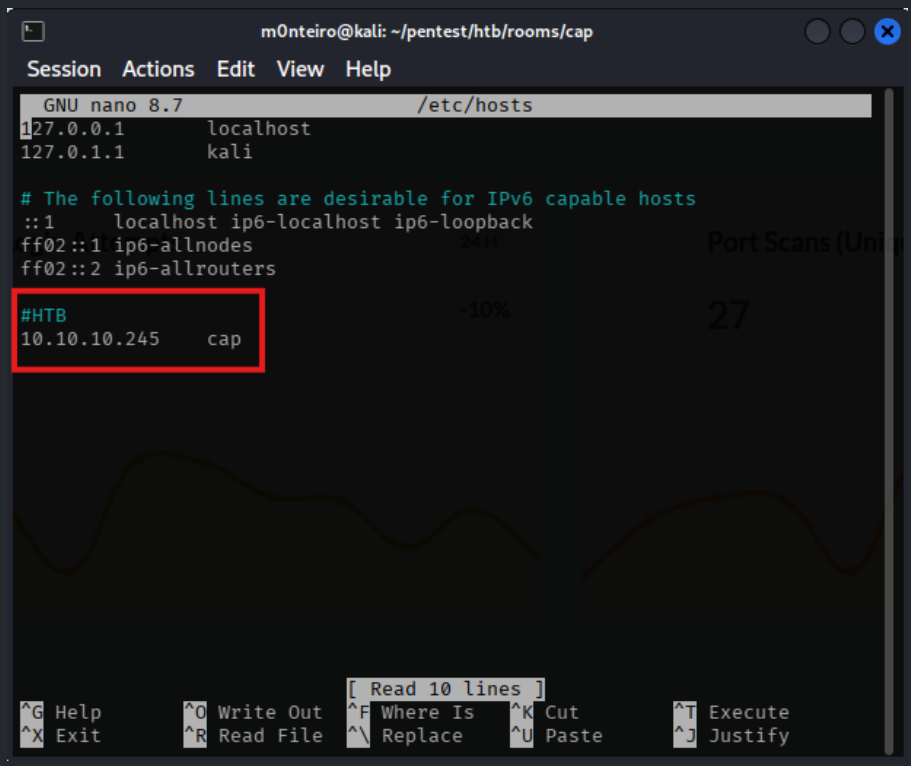
A abordagem seguiu as seguintes fases:

1. Reconhecimento e Enumeração
2. Análise de Vulnerabilidades
3. Exploração
4. Pós-Exploração e Escalação 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.



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

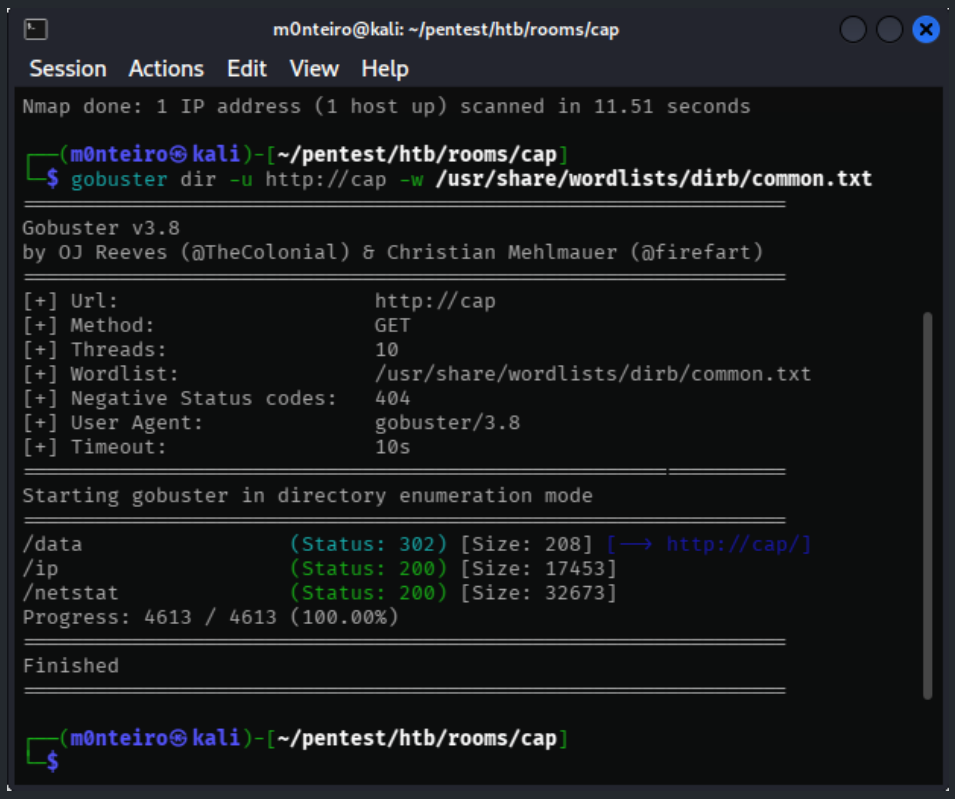
3.2. Enumeração Web (se aplicável)

Ferramentas utilizadas:

- Gobuster para enumeração de diretórios

Ao executar o comando:

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

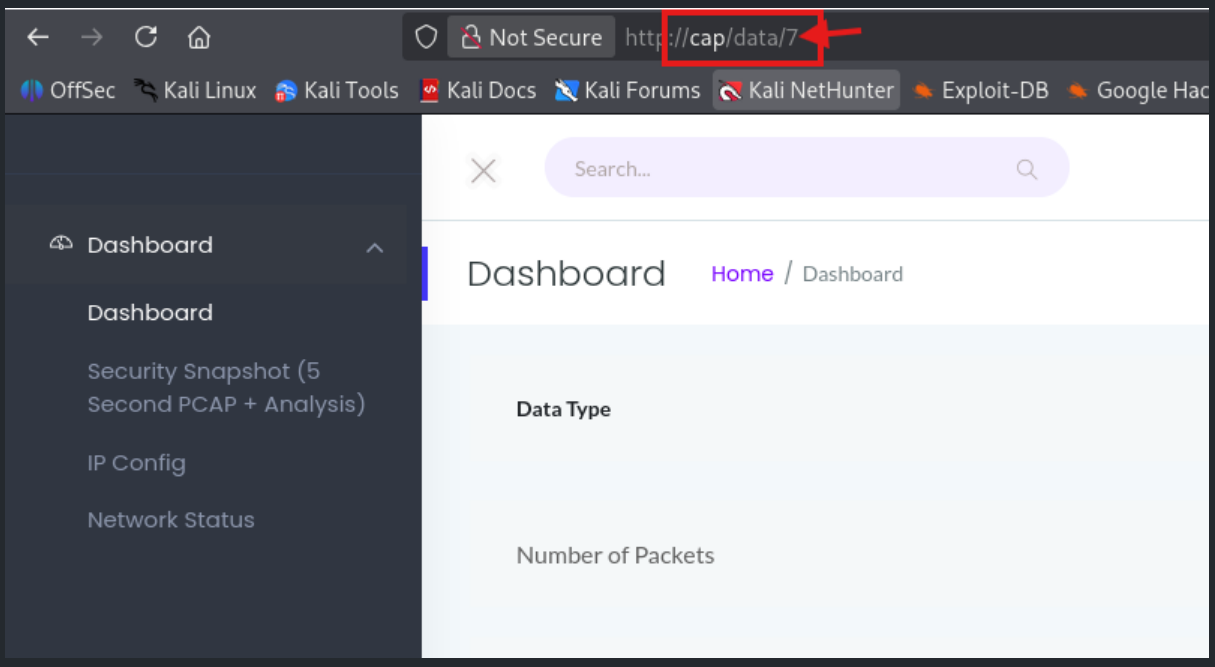


A saída nos retorna nada além do que já está visível no site, o que nos resta explorar a plataforma.

4. Exploração

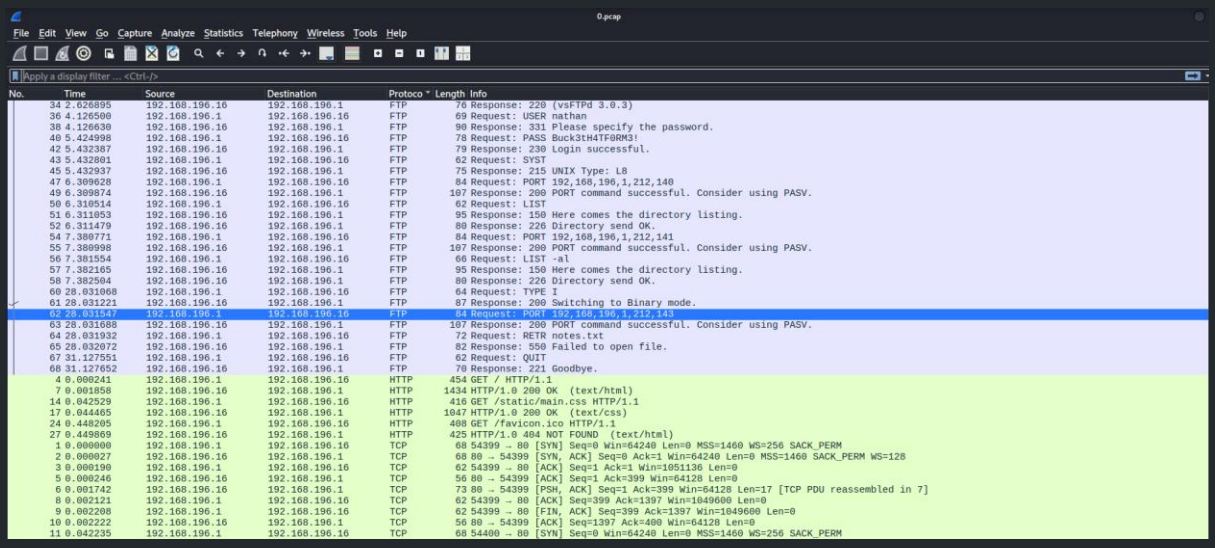
4.1. Acesso Inicial

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.

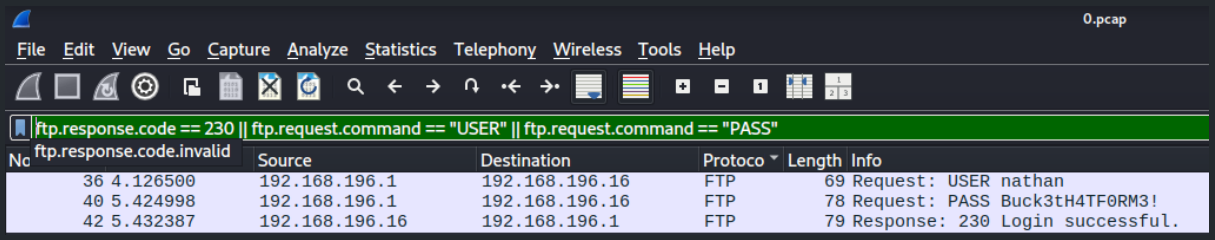
A captura tem extensão ,pcap, usada pelo Wireshark. Acessando o conteúdo por ele podemos ver que um usuário estava mexendo no FTP.



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"
```

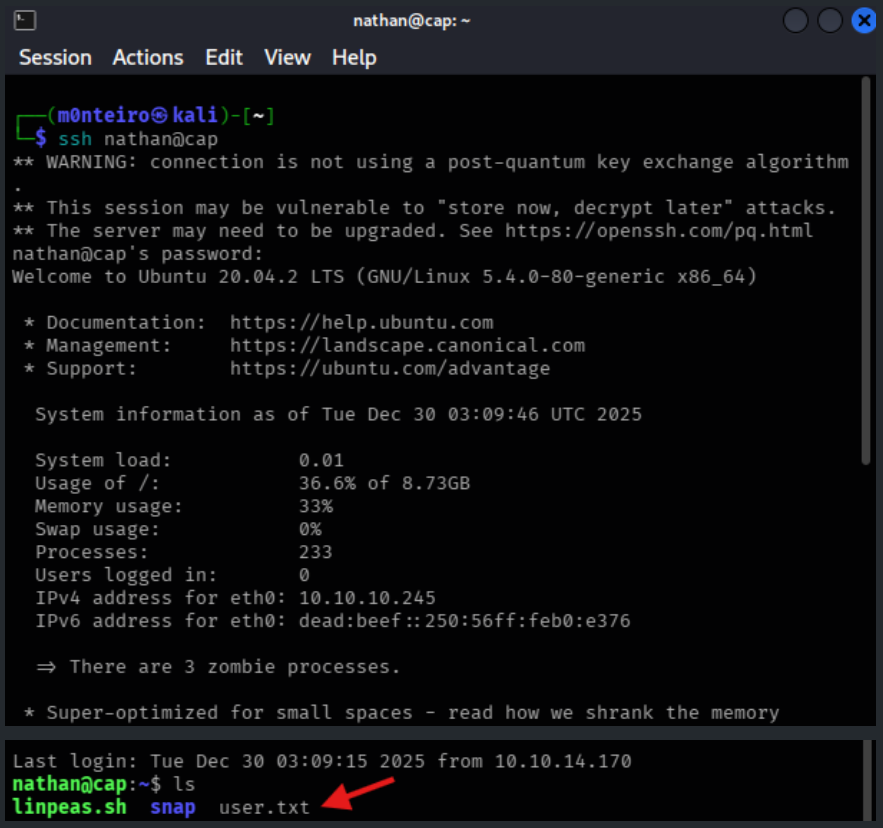
Podemos identificar se houve alguma tentativa bem sucedida, qual usuário foi o usuário usado e senha.



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.



4.2. User Flag

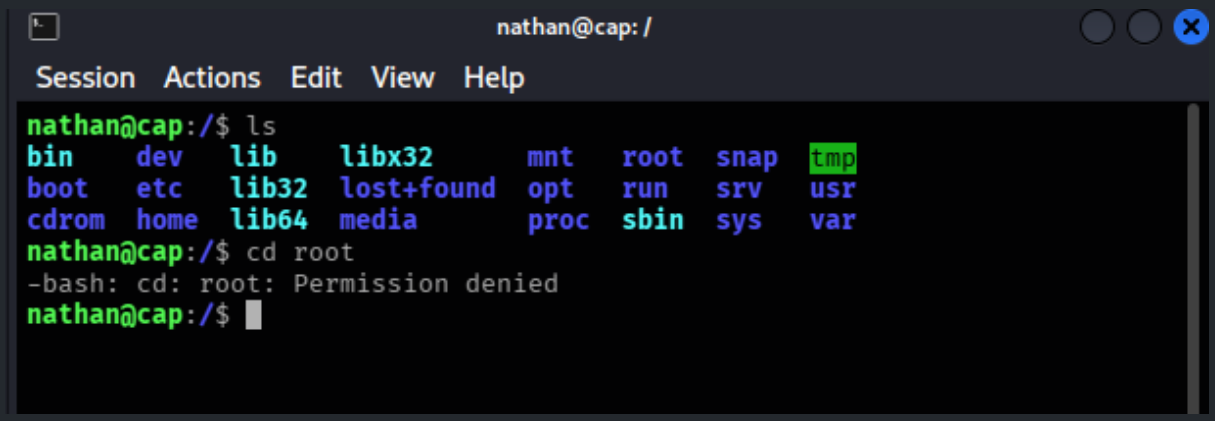
Localização: /home/nathan

`Flag: 782147f8be5746ee8e1425ab9ebf2bc5`

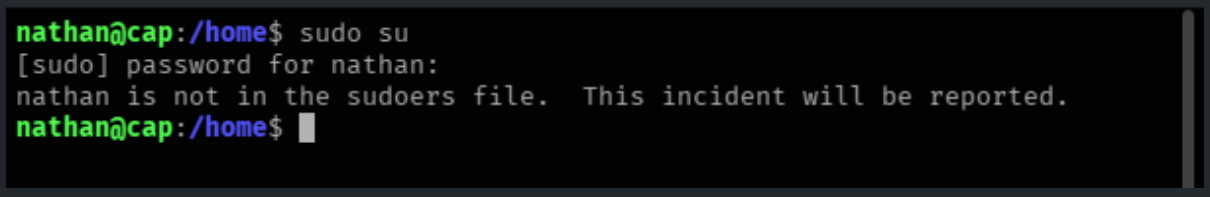
5. Pós-Exploração e Escalação de Privilégios

5.1. Enumeração Interna

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



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

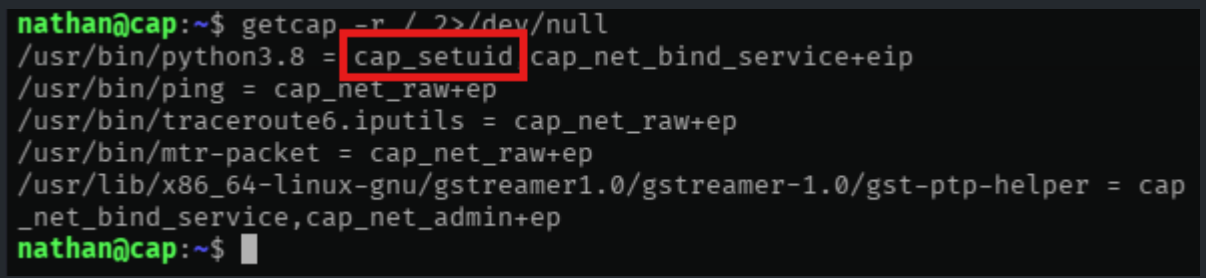


5.2. Vetor de Escalação

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
```

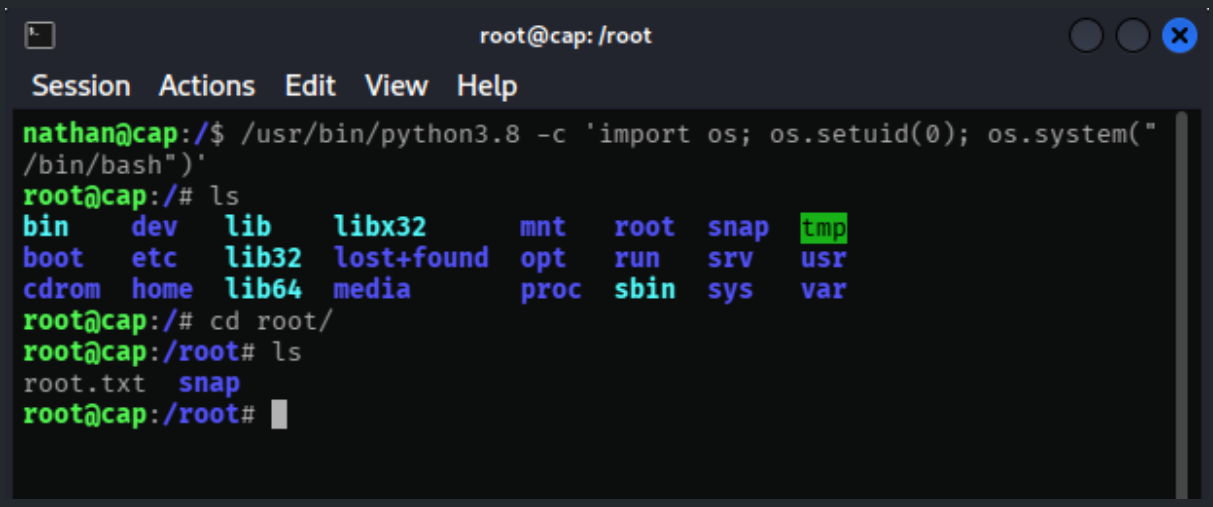
Saída:



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.



5.3. Root/Administrator Flag

Localização: /root

`Flag: fdb2d12d2c627251e76a295b6d11fee4`

6. Análise de Vulnerabilidades

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

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