

Write-up DBS401: Group 6

👋 Welcome to my write-up!

🔧 Tools: Burp Suite

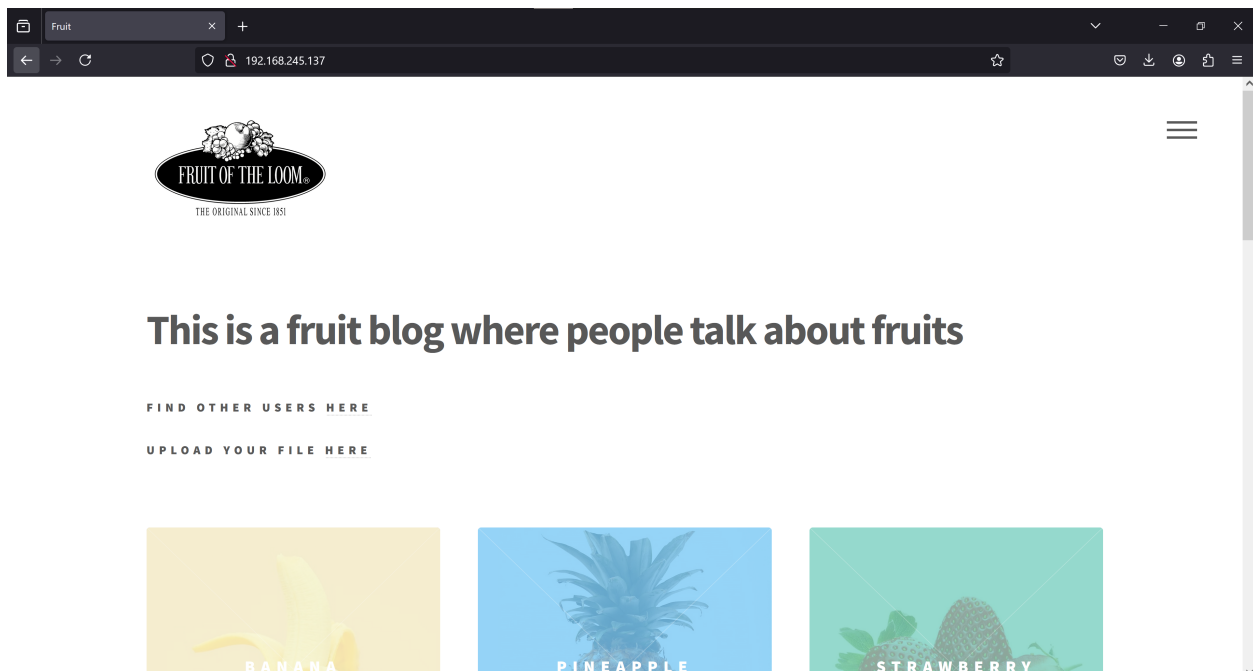
📁 Flags' Completion Table

Flag Number	Vulnerability's Name	Done	Flag
1	Path Traversal	Got Flag Only	{flag1_path_traversal}
2	File Upload	Got Flag Only	{flag2_file_upload_priv_esc}
3	IDOR	yes	flag3



Description

- A fruit blog website



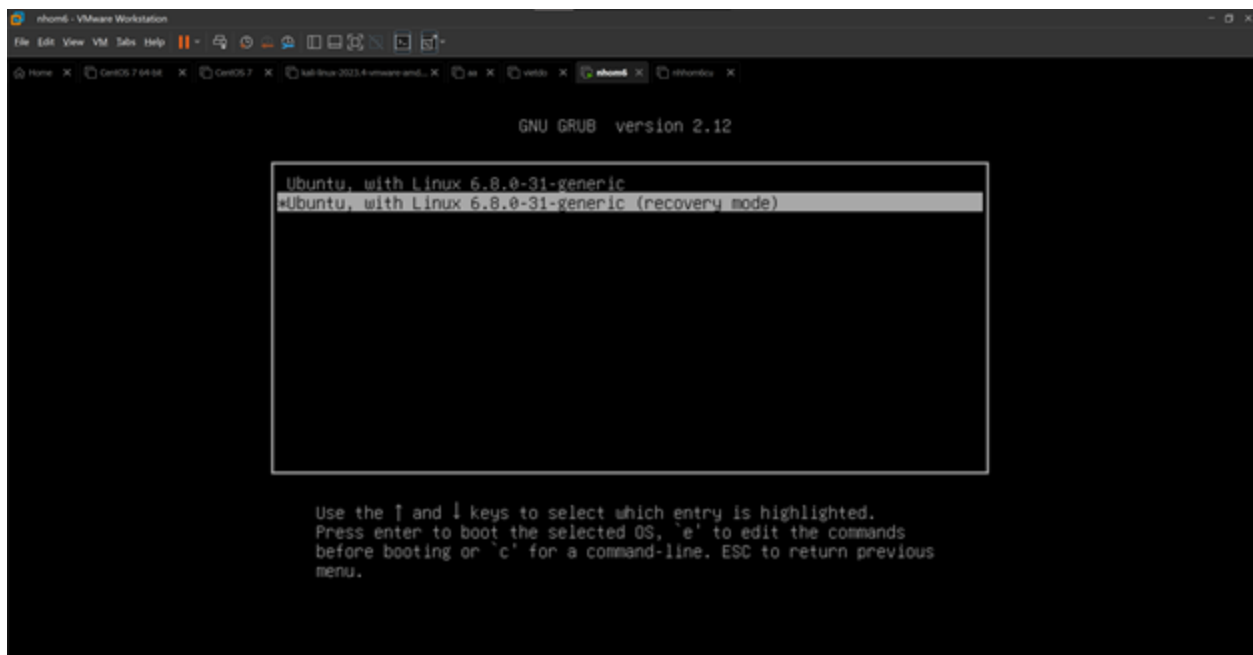
✅ Detailed Analysis

Vulnerability 1: OS Misconfiguration Leads To Unauthorized Root Access

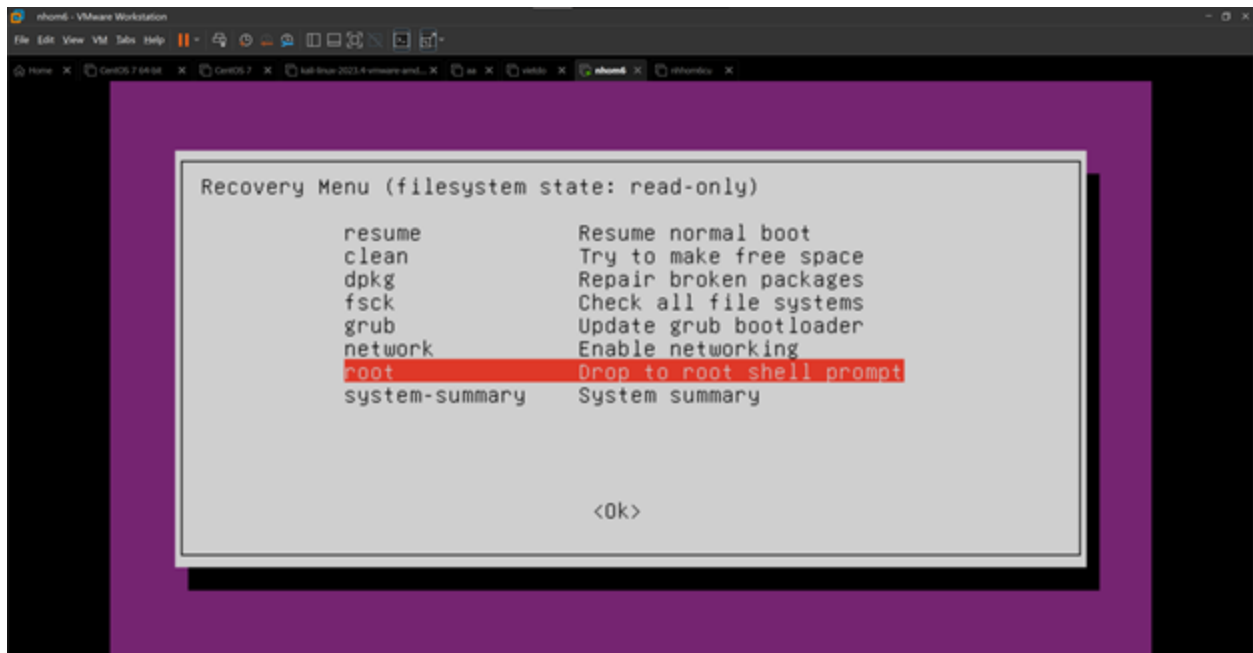
By accessing the recovery mode, an unauthorized person can gain root access without needing the root password because the owner of this machine forgot to set **GRUB Password**. This allows unauthorized person to change any system settings, view or modify all files, and potentially compromise any data stored on the machine.

Step to reproduce

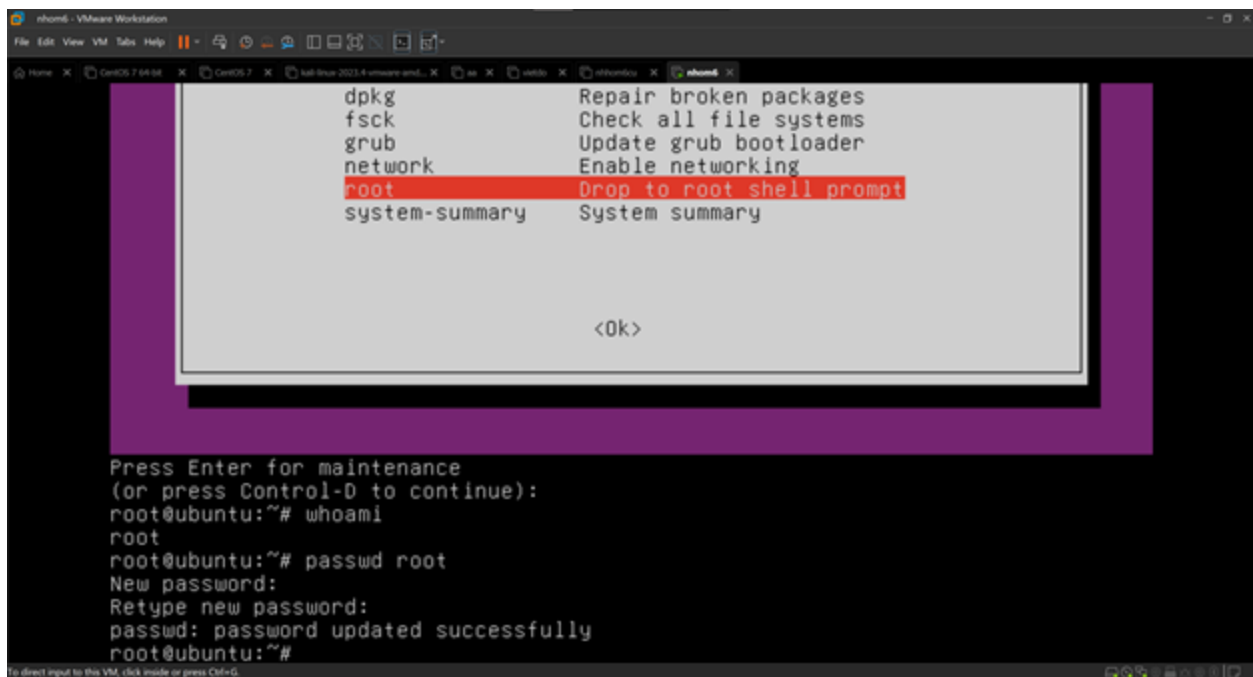
- **Reboot Your System:** Restart your computer.
- **Access GRUB Menu:** Hold down the **Shift** key during boot to access the GRUB menu. If you're using UEFI, you might need to press **Esc** instead.
- **Select Recovery Mode**



- **Select Root Shell Prompt:** After booting into recovery mode, you will be presented with a menu. Select the option "Drop to root shell prompt" or "root".



- Change password of root and we can get fully access to the machine



- According to hint that Group 6 provided, we can know exactly where the flags are

CTF: Câu tạo máy sẽ giống 1 bài CTF cơ bản flag{....}

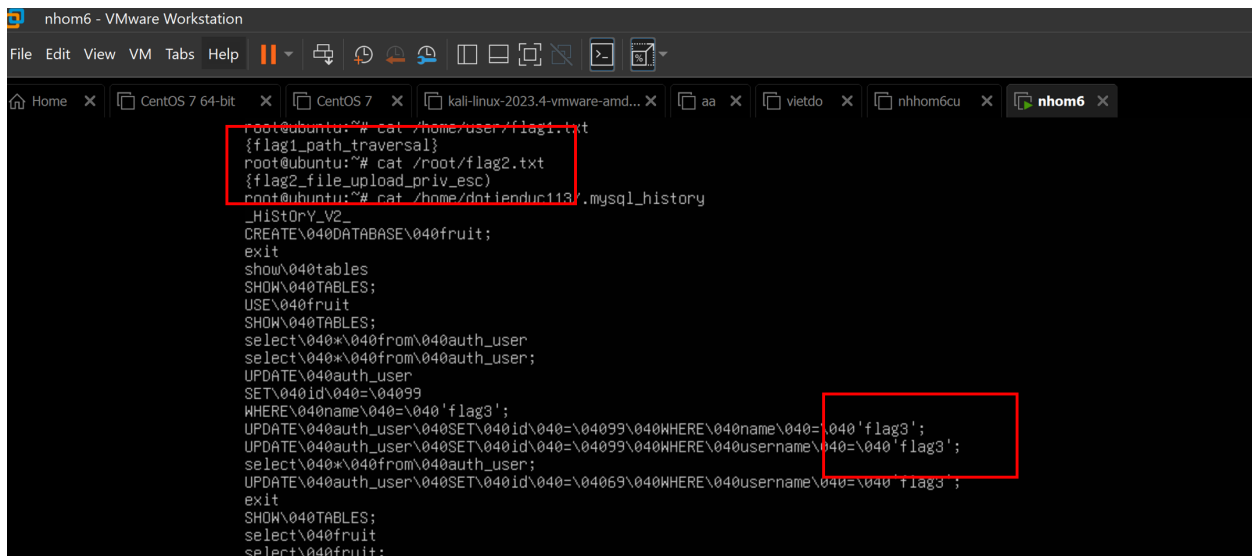
Flag1: ở /home/user

Flag2: ở /root

Flag3: tự tìm

Note: flag3 sẽ ko theo định dạng flag{...} mà chỉ là flag3

- So we can get all the flags at ease



```
root@ubuntu:~# cat /home/user/flag1.txt
{flag1_path_traversal}
root@ubuntu:~# cat /root/flag2.txt
{flag2_file_upload_priv_esc}
root@ubuntu:~# cat /home/.dotfileduc113/.mysql_history
_HISTOry_V2_
CREATE\040DATABASE\040fruit;
exit
show\040tables
SHOW\040TABLES;
USE\040fruit
SHOW\040TABLES;
select\040*\040from\040auth_user
select\040*\040from\040auth_user;
UPDATE\040auth_user
SET\040id\040=\04099
WHERE\040name\040=\040'flag3';
UPDATE\040auth_user\040SET\040id\040=\04099\040WHERE\040name\040=\040'flag3';
UPDATE\040auth_user\040SET\040id\040=\04099\040WHERE\040username\040=\040'flag3';
select\040*\040from\040auth_user;
UPDATE\040auth_user\040SET\040id\040=\04069\040WHERE\040username\040=\040'flag3';
exit
SHOW\040TABLES;
select\040fruit
select\040fruit;
```

- Flag 1: {flag1_path_traversal}
- Flag 2: {flag2_file_upload_priv_esc}
- Flag 3: flag3

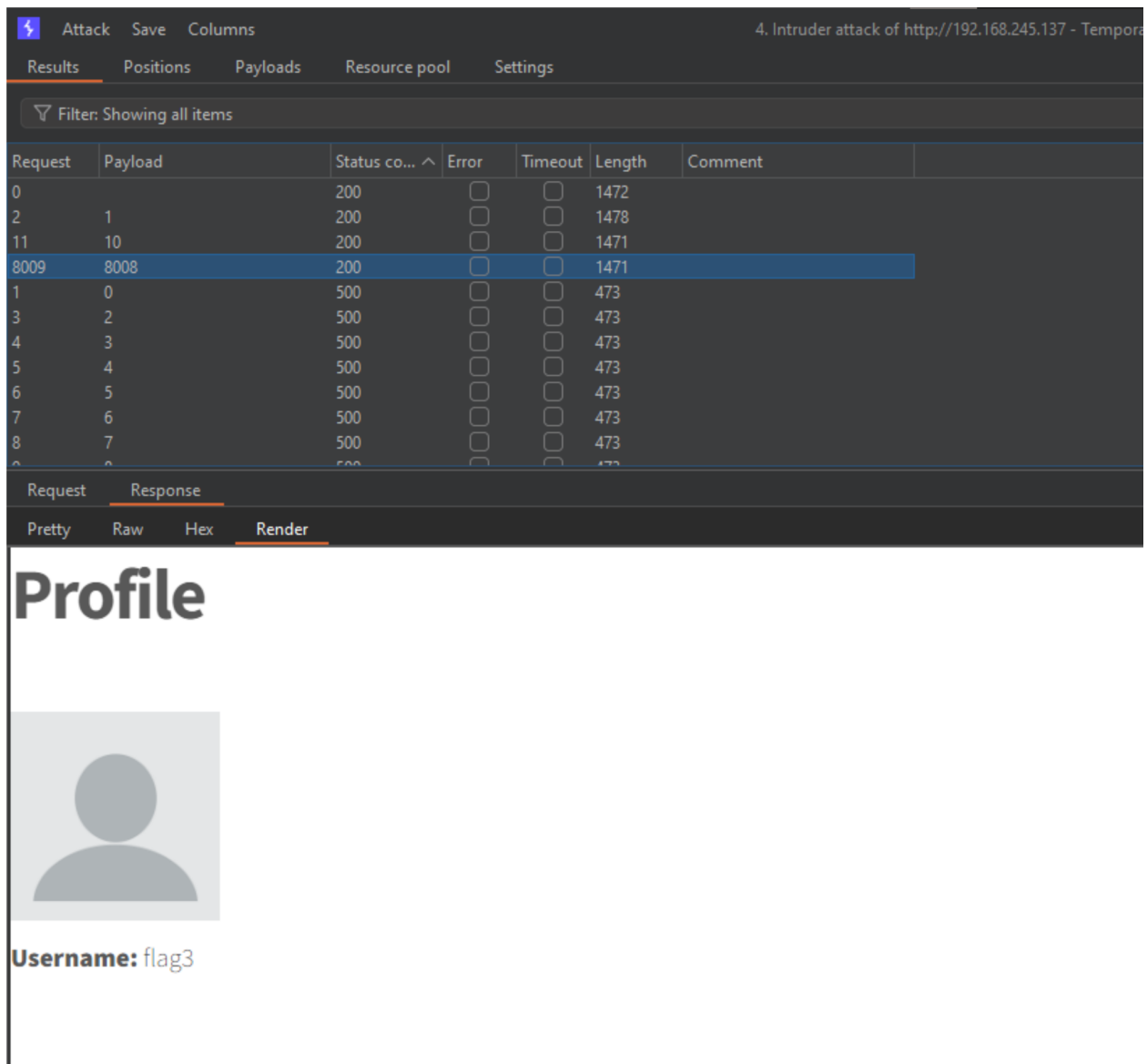
✓ Normal Solution

Vulnerability 2: IDOR

During the registration and login process, we examined the user information function. It was observed that the web application manages user IDs in a

predictable manner. For instance, accessing a user's details can be done via a URL like `http://192.168.245.137/user/id=8008`, which suggests the presence of an IDOR vulnerability.

- To confirm this, we utilized the intruder feature of the Burp Suite tool to test. Our tests revealed that the application indeed suffers from an IDOR vulnerability, as it allowed us to view the information of a few users. One of them is flag.




The screenshot shows the Burp Suite Intruder tool interface. The top menu includes 'Attack', 'Save', and 'Columns'. The main window displays a table of requests and their corresponding responses. The table has columns for 'Request', 'Payload', 'Status code', 'Error', 'Timeout', 'Length', and 'Comment'. The request with ID 8009 is highlighted, showing a status code of 200 and a length of 1471. Below the table, the 'Response' tab is selected, showing a rendered HTML page titled 'Profile'. The page contains a placeholder for a user profile picture and the text 'Username: flag3'.

Request	Payload	Status code	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1472	
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	1478	
11	10	200	<input type="checkbox"/>	<input type="checkbox"/>	1471	
8009	8008	200	<input type="checkbox"/>	<input type="checkbox"/>	1471	
1	0	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
3	2	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
4	3	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
5	4	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
6	5	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
7	6	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
8	7	500	<input type="checkbox"/>	<input type="checkbox"/>	473	
9	8	500	<input type="checkbox"/>	<input type="checkbox"/>	473	

Request Response

Pretty Raw Hex Render

Profile



Username: flag3

► **Flag 3:** flag3