

CTF Writeup

This is my submission for task 1- the CTF writeup of the Neighbor room on TryHackMe.

Challenge Information:

Name: Neighbor

Platform: TryHackMe

Difficulty: Easy

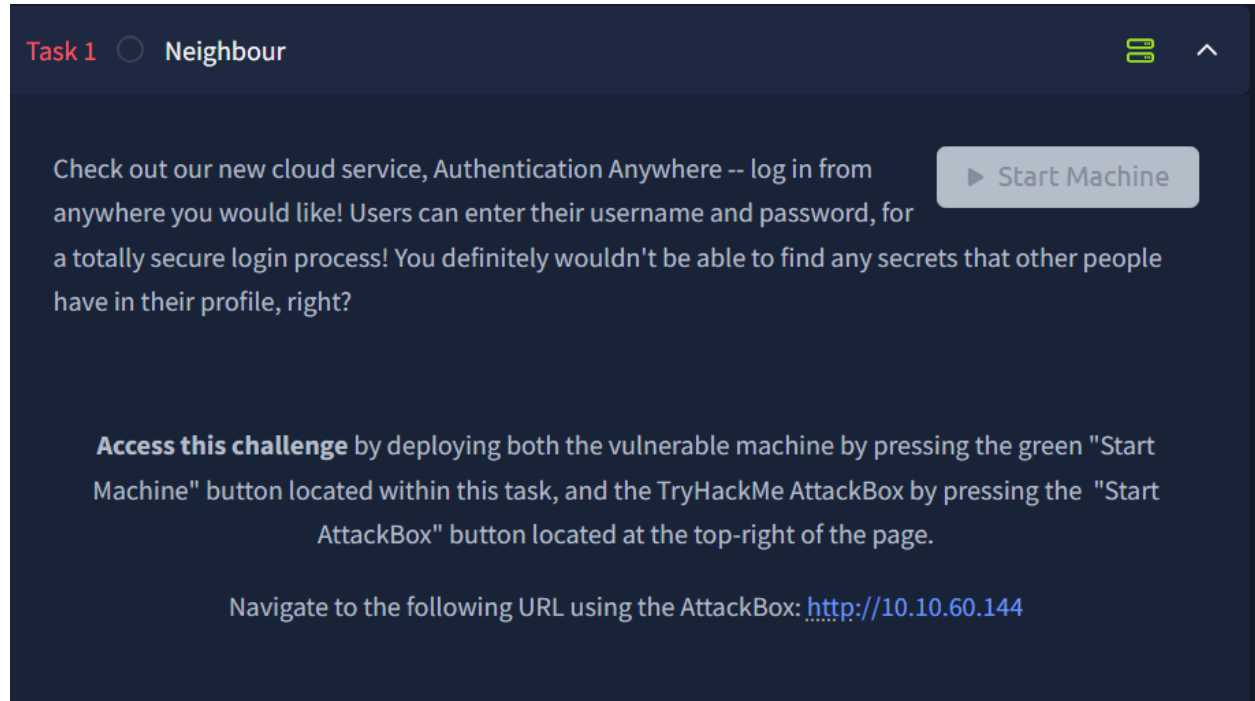
Category: Web Exploitation

Link: <https://tryhackme.com/room/neighbour>

Challenge Description:

You're given access to a networked machine. Your goal is to enumerate the system, find the vulnerable services, exploit them, and escalate privileges to retrieve flags.

Challenge Walkthrough:



The screenshot shows the 'Task 1' interface for the 'Neighbour' challenge on TryHackMe. The header includes 'Task 1' and 'Neighbour' with a progress indicator. A 'Start Machine' button is visible. The main text describes a cloud service called 'Authentication Anywhere' and provides instructions on how to access the challenge by deploying a vulnerable machine and using the TryHackMe AttackBox. A URL is provided for navigation: <http://10.10.60.144>.

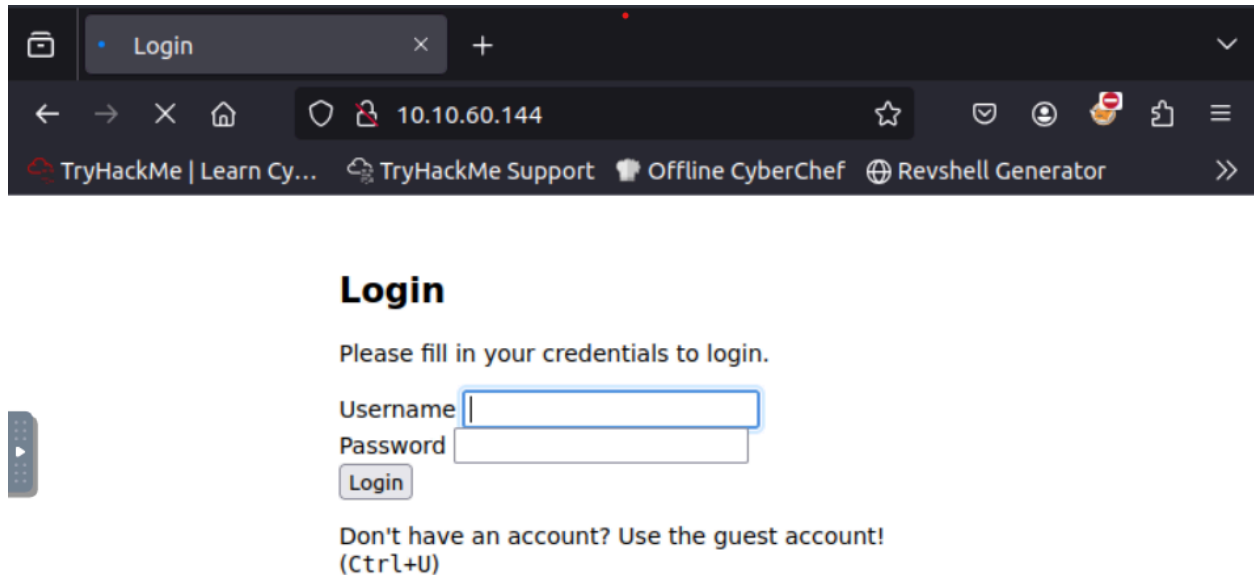
Task 1 ○ Neighbour

Check out our new cloud service, Authentication Anywhere -- log in from anywhere you would like! Users can enter their username and password, for a totally secure login process! You definitely wouldn't be able to find any secrets that other people have in their profile, right?

Access this challenge by deploying both the vulnerable machine by pressing the green "Start Machine" button located within this task, and the TryHackMe AttackBox by pressing the "Start AttackBox" button located at the top-right of the page.

Navigate to the following URL using the AttackBox: <http://10.10.60.144>

First I typed in the target ip address in the browser to go to the site. There a login page was displayed but we don't know the valid credentials to log in yet.



Login

Please fill in your credentials to login.

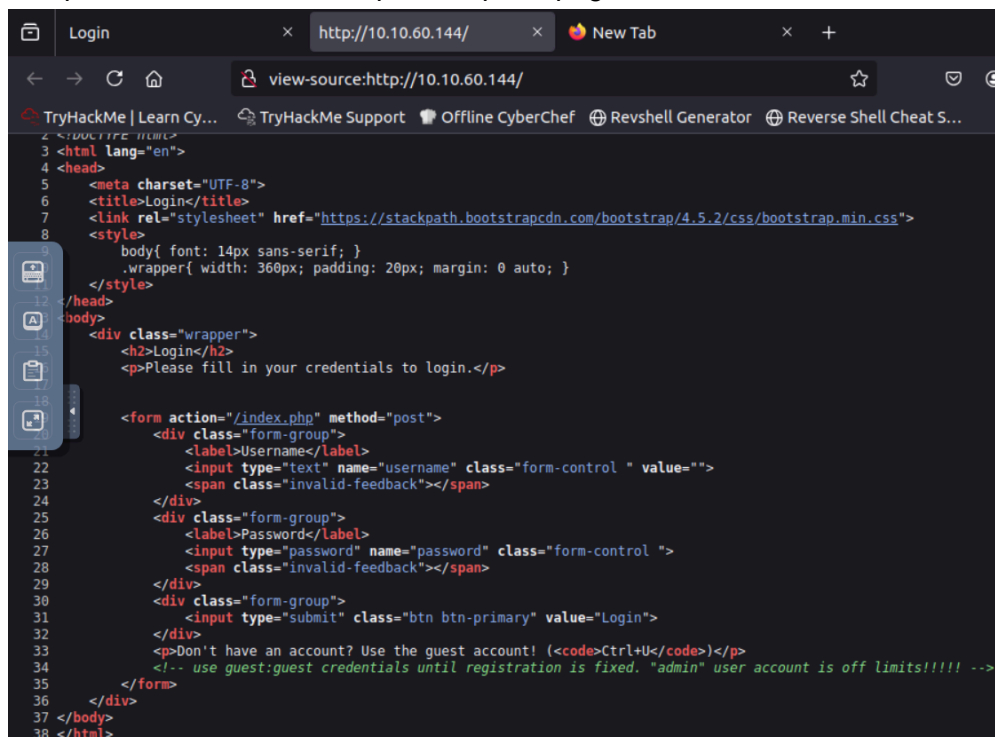
Username

Password

Login

Don't have an account? Use the guest account!
(Ctrl+U)

So I pressed Ctrl+U, which opened up the page source.



```
1 <DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <title>Login</title>
6 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7 <style>
8 body{ font: 14px sans-serif; }
9 .wrapper{ width: 360px; padding: 20px; margin: 0 auto; }
10 </style>
11 </head>
12 <body>
13 <div class="wrapper">
14 <h2>Login</h2>
15 <p>Please fill in your credentials to login.</p>
16
17 <form action="/index.php" method="post">
18 <div class="form-group">
19 <label>Username</label>
20 <input type="text" name="username" class="form-control " value="">
21 <span class="invalid-feedback"></span>
22 </div>
23 <div class="form-group">
24 <label>Password</label>
25 <input type="password" name="password" class="form-control ">
26 <span class="invalid-feedback"></span>
27 </div>
28 <div class="form-group">
29 <input type="submit" class="btn btn-primary" value="Login">
30 </div>
31 <p>Don't have an account? Use the guest account! (<code>Ctrl+U</code>)</p>
32 <!-- use guest:guest credentials until registration is fixed. "admin" user account is off limits!!!! -->
33 </form>
34 </div>
35 </body>
36 </html>
```

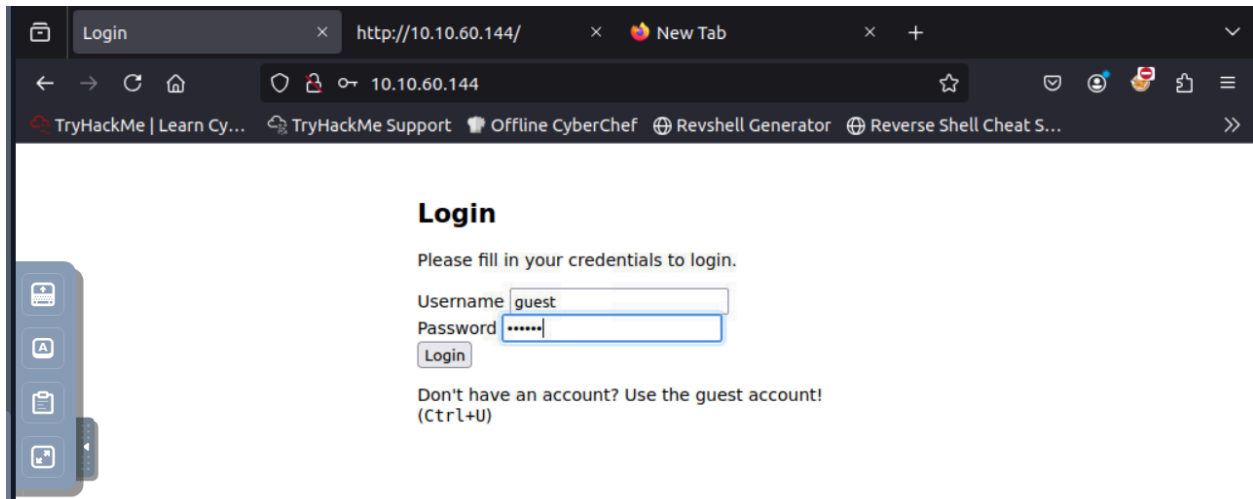
From here I got to know that there seems to be an admin user account as well as the guest.

From here I also got the credentials for the guest profile :

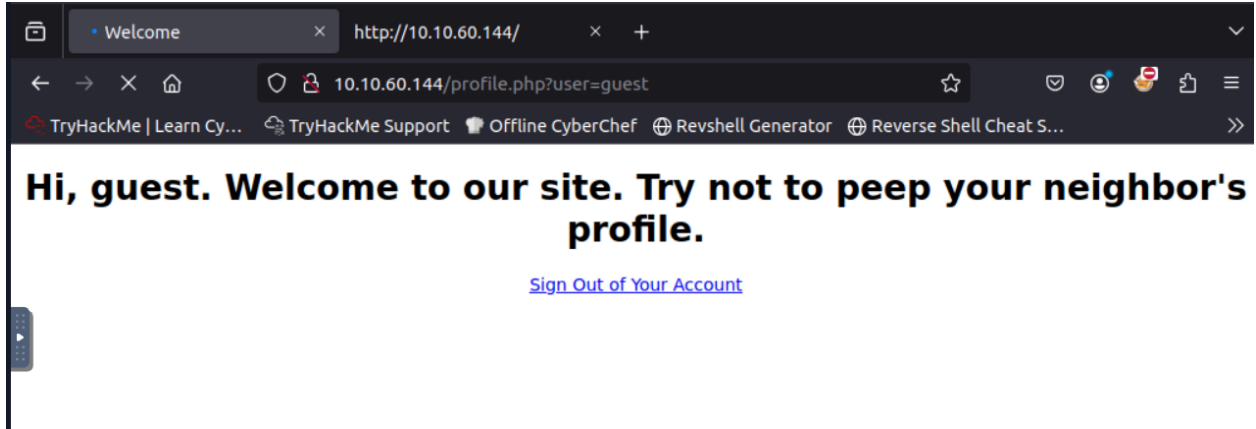
Username: guest

Password: guest

Going back to the login page, I entered the credentials and successfully logged in.

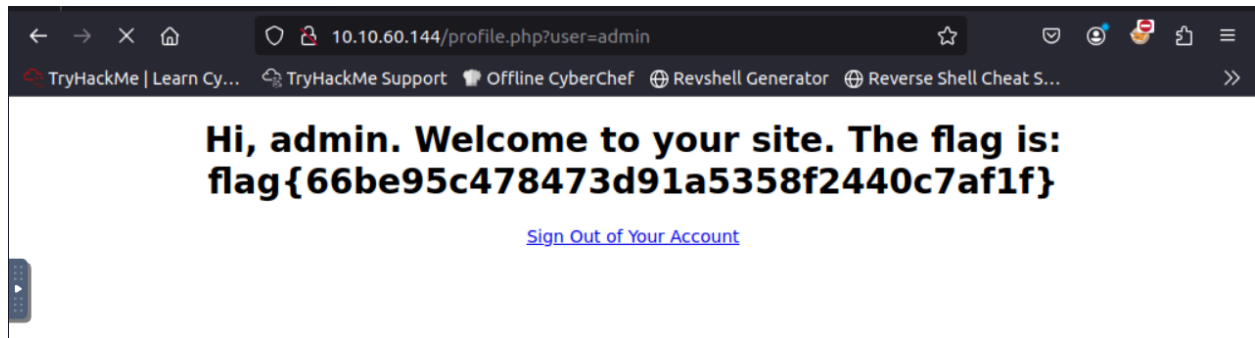


After login, the page is redirected to:

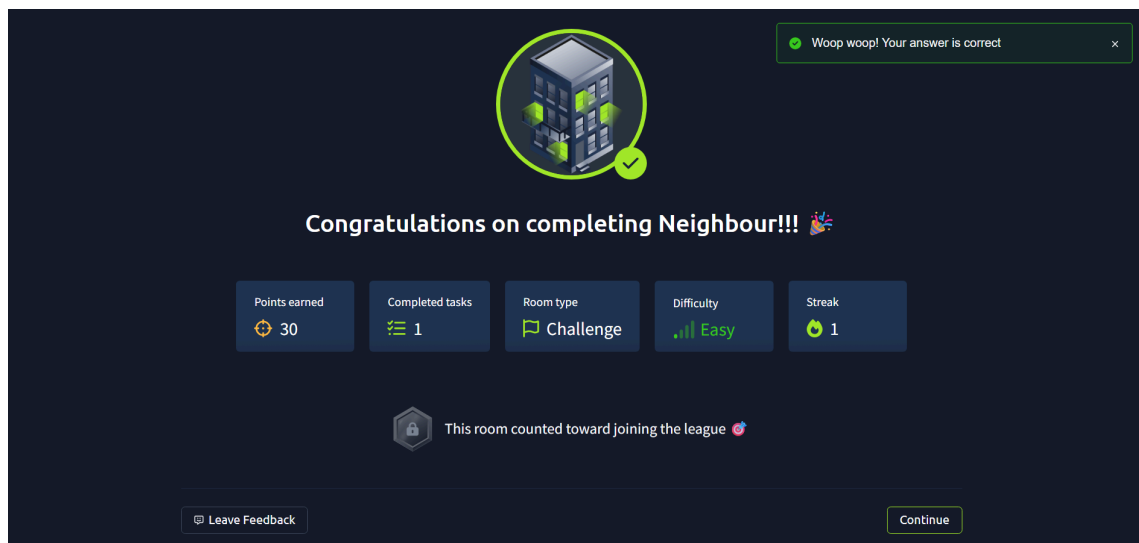
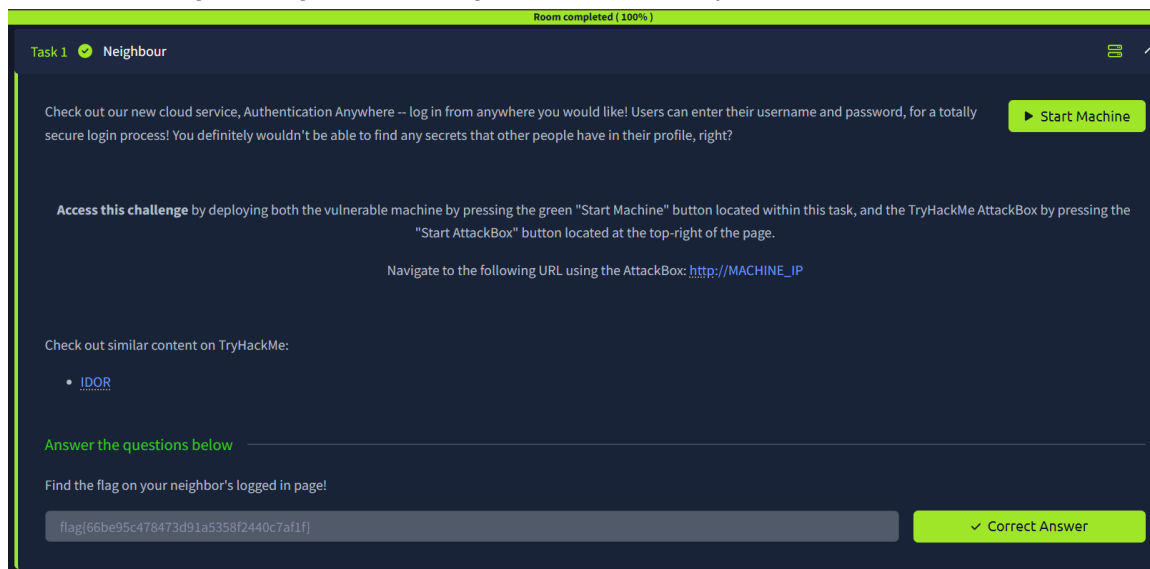


But from the page source earlier, we could tell that there are both guest as well as admin user accounts. So I will try to replace the user=guest part of the url to user=admin.

By doing this, it takes me to the admin page where I got the CTF flag.



After submitting the flag, the challenge was successfully completed.



What I Learned:

- How to find hidden credentials in page source
- Basics of web app enumeration
- Real-world application of IDOR vulnerabilities
- Importance of validating user access on the backend

Core Concept:

Insecure Direct Object Reference (IDOR) is a type of access control vulnerability that occurs when an application exposes internal implementation objects (like files, database records, or keys) through user-supplied input, without proper authorization checks.

IDOR happens when users can access or manipulate resources they shouldn't be able to — just by modifying URL parameters, form data, or API requests.

In this CTF challenge, I was able to easily access an unauthorized page just by modifying the URL parameters.

By Christina George.