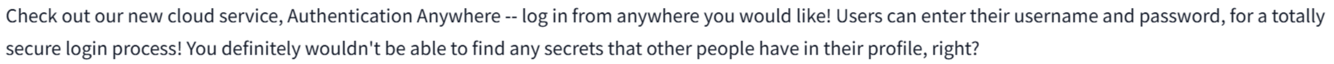
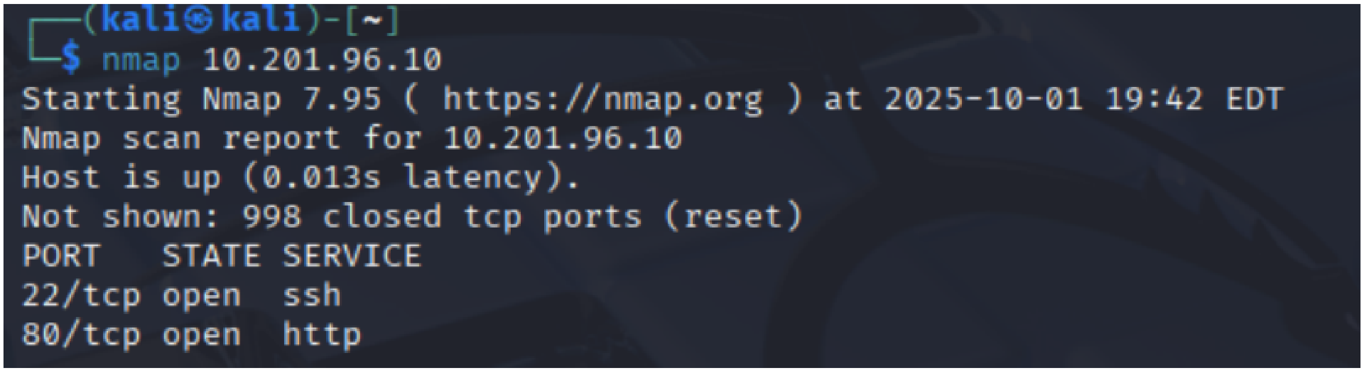
This room is a basic example of how to exploit an IDOR (Insecure Direct Object Reference) vulnerability. This is a common type of vulnerability where an application uses an input supplied by the user to directly reference an object without actually validating that the user has the authorization to do so.



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?

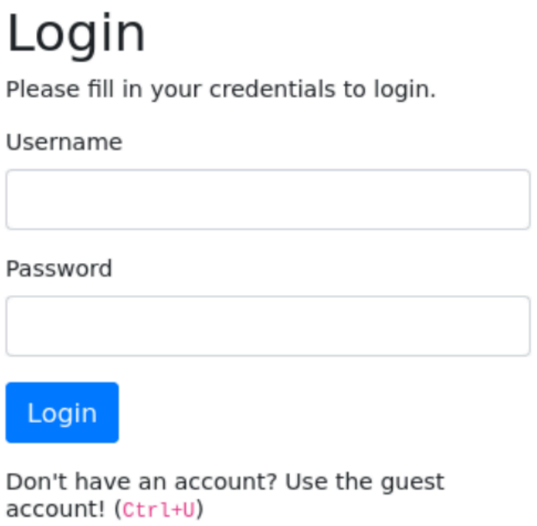
Recon

The first thing I'll do is run a quick nmap scan to see what services are running on the target machine.



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.201.96.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-01 19:42 EDT
Nmap scan report for 10.201.96.10
Host is up (0.013s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Just two services are running, ssh and http. Next, I'll navigate to the target IP in my browser.



Login

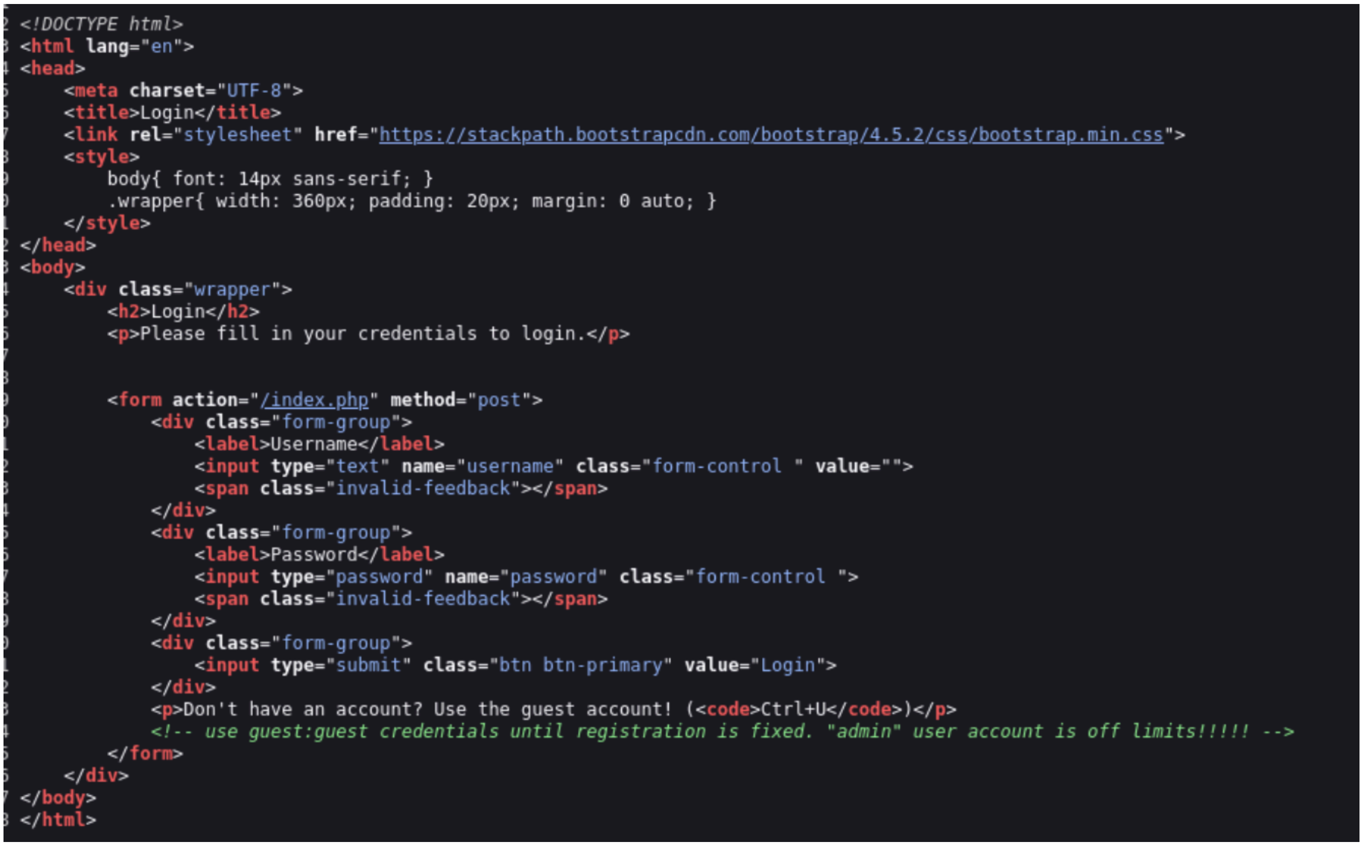
Please fill in your credentials to login.

Username

Password

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

I'm greeted by a pretty simplistic login page. Obviously, I don't have login credentials, so let's see what we can do with the guest account mentioned.



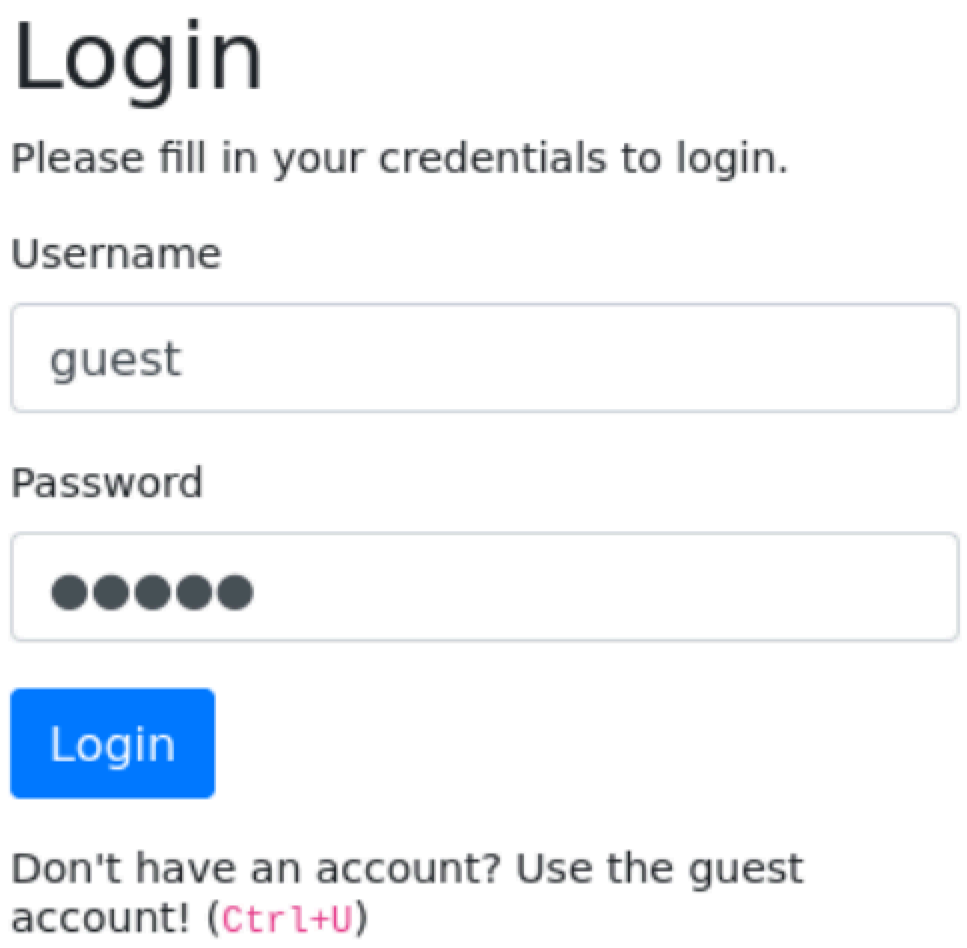
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body{ font: 14px sans-serif; }
        .wrapper{ width: 360px; padding: 20px; margin: 0 auto; }
    </style>
</head>
<body>
    <div class="wrapper">
        <h2>Login</h2>
        <p>Please fill in your credentials to login.</p>


        <form action="/index.php" method="post">
            <div class="form-group">
                <label>Username</label>
                <input type="text" name="username" class="form-control " value="">
                <span class="invalid-feedback"></span>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input type="password" name="password" class="form-control ">
                <span class="invalid-feedback"></span>
            </div>
            <div class="form-group">
                <input type="submit" class="btn btn-primary" value="Login">
            </div>
            <p>Don't have an account? Use the guest account! (<code>Ctrl+U</code>)</p>
            <!-- use guest:guest credentials until registration is fixed. "admin" user account is off limits!!!!! -->
        </form>
    </div>
</body>
</html>
```

Pressing Ctrl + U opens a new tab to inspect the source code of the login page. In the comment at the bottom, we can see guest credentials are "guest:guest". Not very secure, if you ask me. The comment also makes a note not to use the "**admin**" account. That gives us the username for the account we are trying to access.

Find Flag

Let's see how far we can get with the guest account for now.



Login

Please fill in your credentials to login.

Username

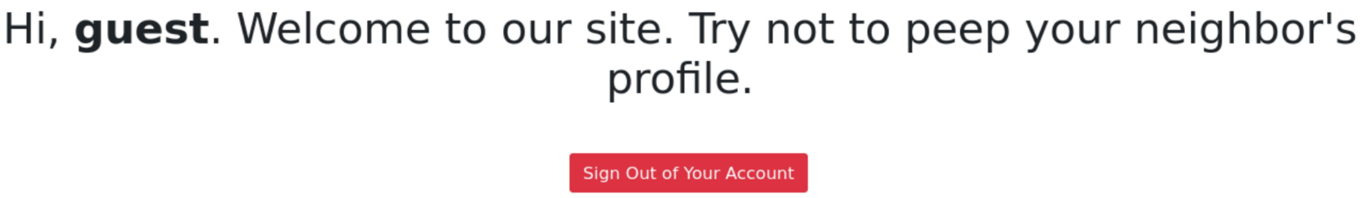
guest

Password

●●●●●

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

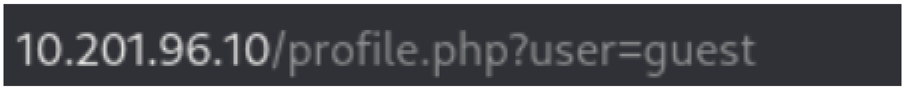
Hitting the login button yields the following page:



Hi, **guest**. Welcome to our site. Try not to peep your neighbor's profile.

Unfortunately for the person who wrote this message, I will 100% be peeping my neighbor's profile.

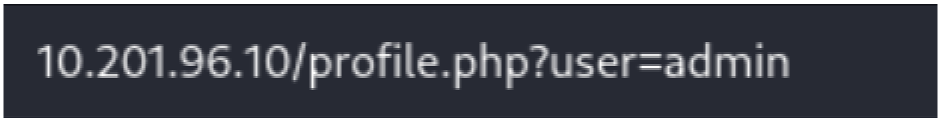
Time to exploit the IDOR vulnerability. Let's take a look at the current URL.



```
10.201.96.10/profile.php?user=guest
```

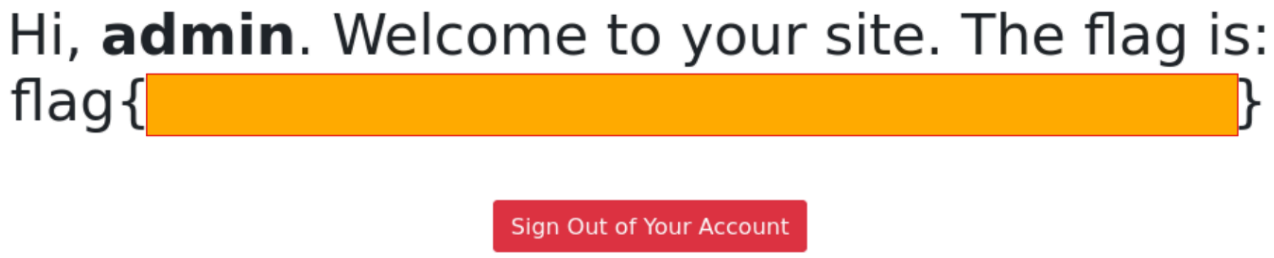
We can see that our current user is detailed plainly in the URL, displaying "user=guest". This matches the username we entered when we logged in. What if we were able to change this value to display another user's equivalent of this page? All we would need would be the username of the target. Speaking of which, remember that "admin" account mentioned earlier? That's a username we can try accessing!

Just change the "guest" part of the URL to "admin".



```
10.201.96.10/profile.php?user=admin
```

Hit enter!



Hi, **admin**. Welcome to your site. The flag is: flag{[redacted]}

Perfect! We were able to exploit the IDOR vulnerability to display the admin page because the application did not validate if we had authorization to view the page.

Conclusion

This room was a quick and easy example of how to exploit an IDOR vulnerability on a web application. It's clear how important it is to verify user authorization! Also, it's important not to leave login credentials in easy to find locations, such as comments on page source.