Cybersecurity 701

XSS Juice Shop Lab

XSS Juice Shop Materials

- Materials needed
  - Kali Virtual Machine (with Juice Shop)

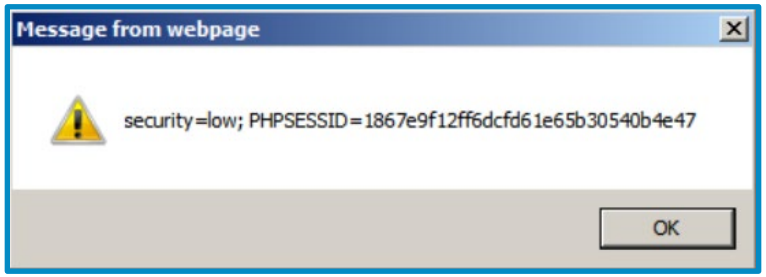
- Software Tool used
  - Juice Shop
    - Follow the Juice Shop Setup Lab if not previously installed/available on your VM

Objectives Covered

- Security+ Objectives (SY0-701)
  - Objective 2.3 - Explain various types of vulnerabilities.
    - Web-based
      - Cross-site scripting (XSS)

What is a Cross-Site Scripting Attack?

- Inserting scripts (usually JavaScript) into a pages' HTML to bypass server access controls
- Can be used to access data that should be hidden on a webpage
  - Why is this dangerous if the user is privileged?



XSS Juice Shop Lab Overview

1. Set up environment
2. Access Juice Shop website
3. XSS Tier 1
4. XSS Tier 2

Set up Environment

- Log into your range
- Open the Kali Linux Environment
  - You should be on your Kali Linux Desktop
  - Open a Terminal

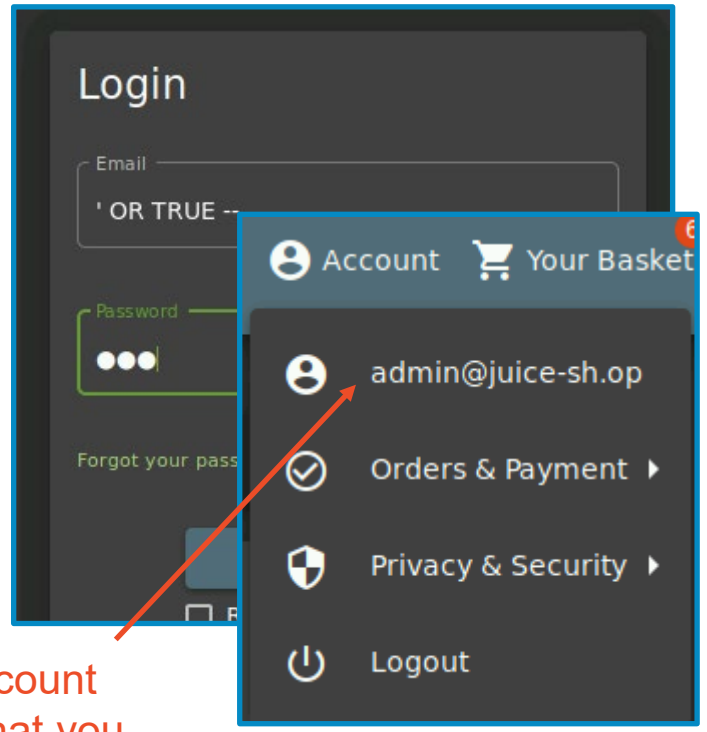
Access Juice Shop Website

- Navigate to the juice-shop directory and start npm
  - `cd juice-shop`
  - `npm start`
- Open Firefox and navigate to localhost:3000

Access Juice Shop Website

- Click on **Account** and then **Login**
- Log in as the admin user
  - Username: `` ` OR TRUE -- ``
  - Password: `111`

Please Note: This is a SQL Injection to log in as the admin account. The SQL Injection lab goes into more details on how this works



Click on the Account option to verify that you are logged in as the admin account

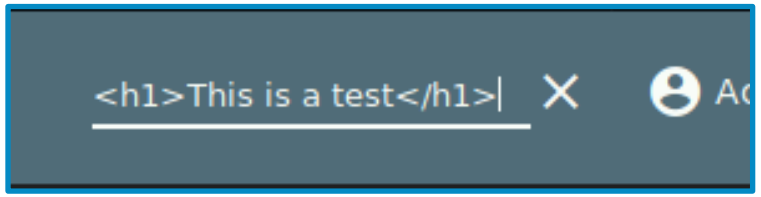
XSS Tier 1

- Here we want to test if we can inject HTML into the web page. Click the magnifying glass and search for the following:

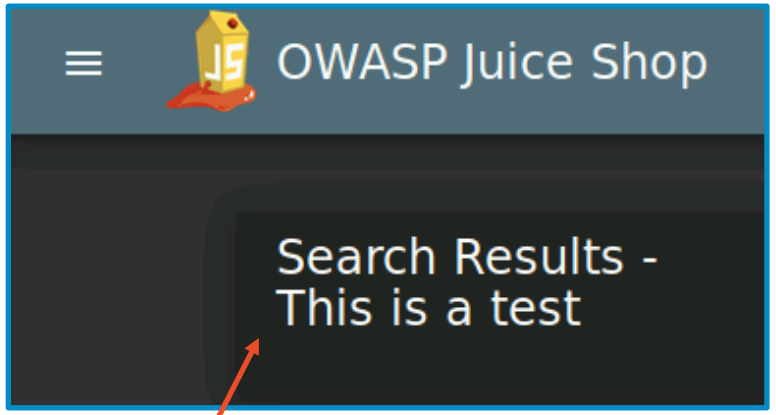
  pineapple juice

- Then, enter a line of HTML:

  `<h1>This is a test</h1>`



Please Note: This verifies that we can inject HTML scripts into the webpage

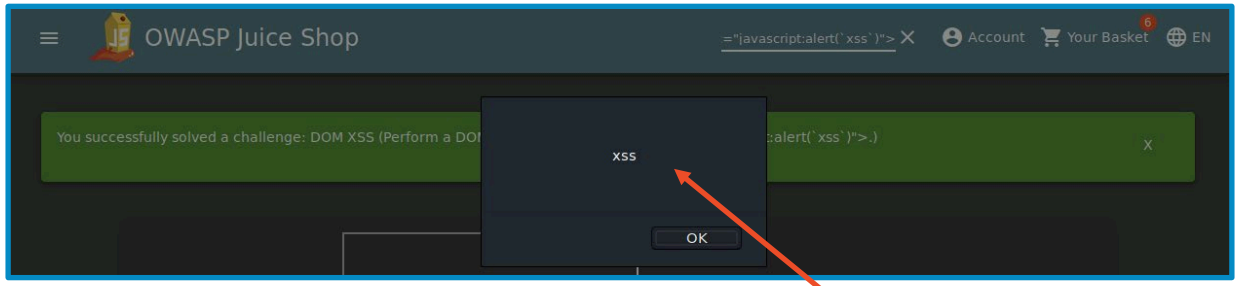


Notice, that the HTML code appears as the header

XSS Tier 1

- Insert the following code into the Juice Shop search bar:

```
<iframe src="javascript:alert(`xss`)">
```

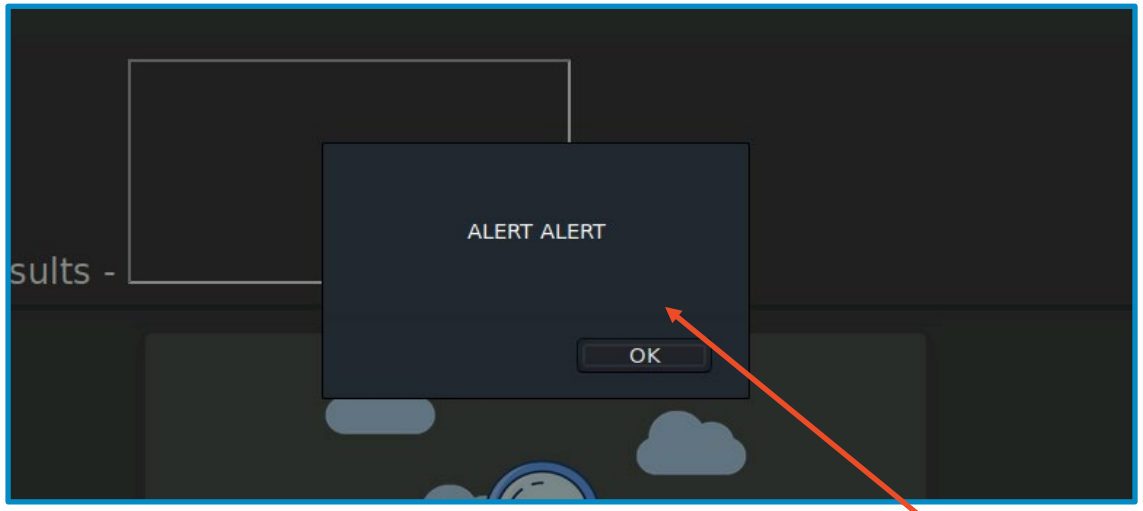


Verify that the alert ‘xss’ appeared on the screen

XSS Tier 1

- Modify the alert that appears on the screen

```
<iframe src="javascript:alert(‘ALERT ALERT’)">
```

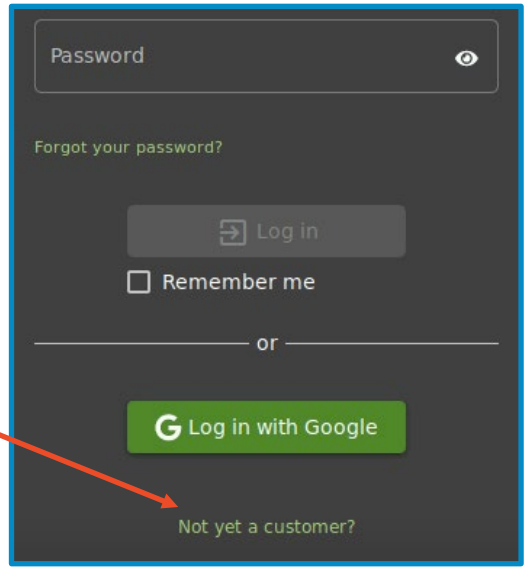


Verify that the new alert ‘ALERT ALERT’ appeared on the screen

XSS Tier 2

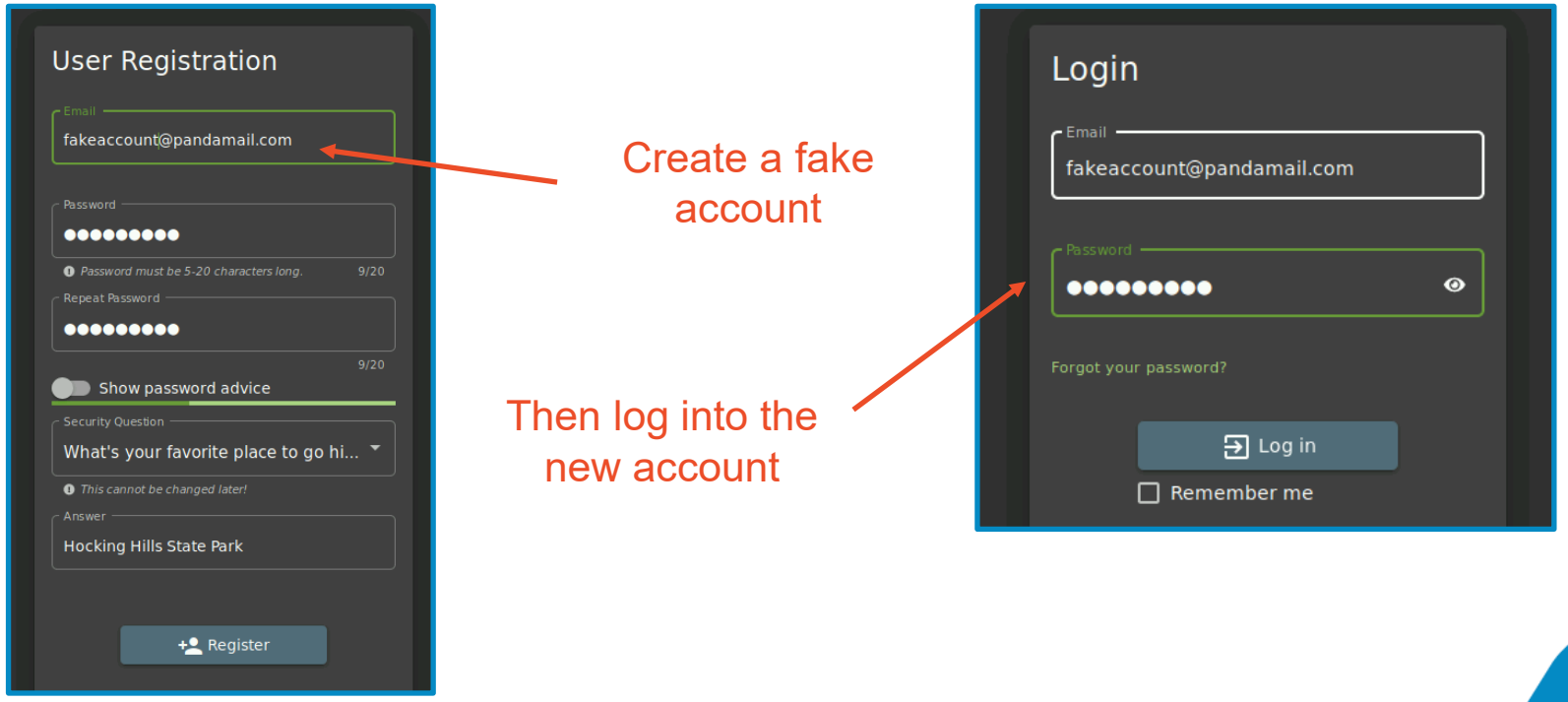
- Create a new user
- Log out of the admin account
- Click **Account** and then **Login**
- Scroll to the bottom and select “**Not yet a customer?**”

Click on “Not yet a customer?” to create a new account



XSS Tier 2

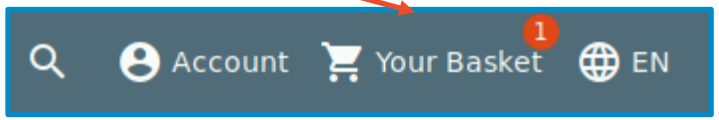
- Fill out the form with any credentials
  - It is recommended not to use a real email account
- Then log-in using this new account



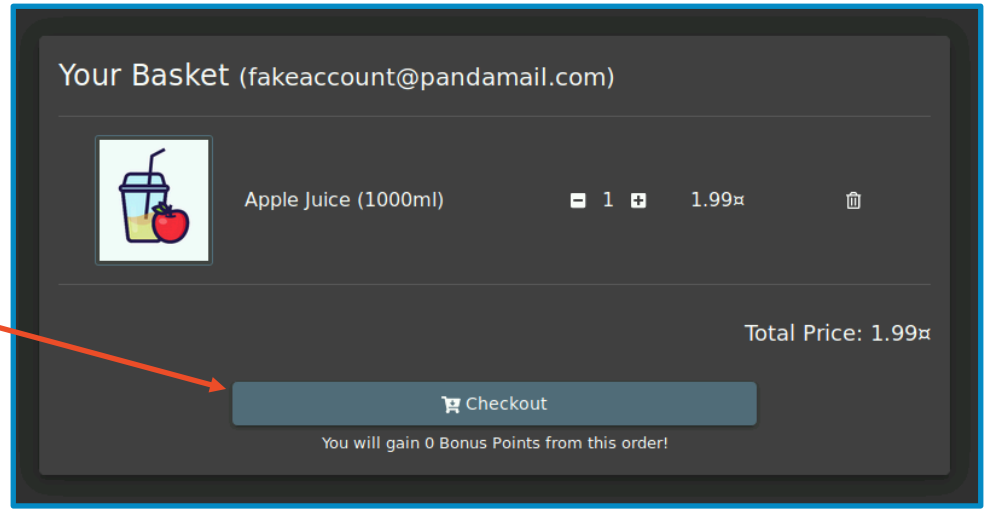
XSS Tier 2

- Add Apple Juice to the basket
- Then, scroll to the top and click Your Basket
- Then click Checkout

You should notice an item has been added to your basket

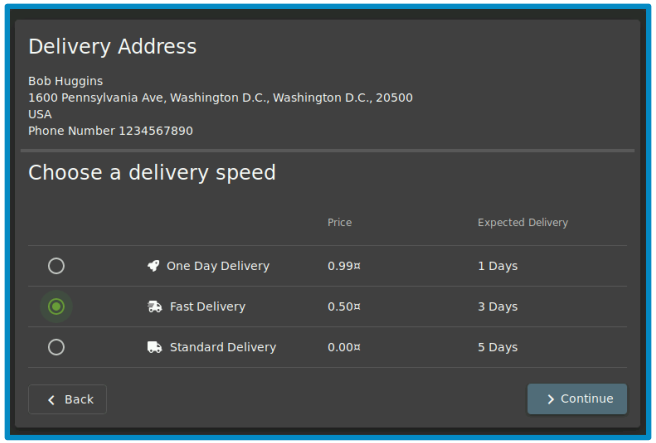


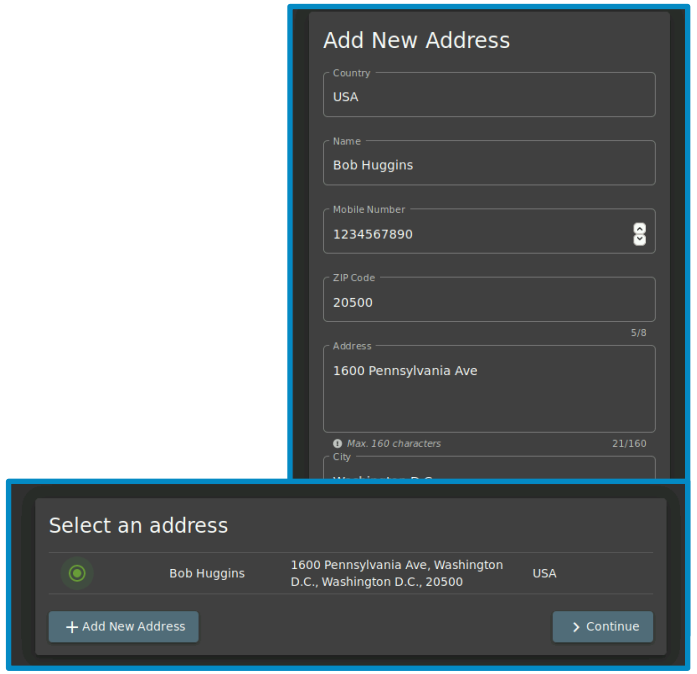
With the apple juice in the basket, select ‘Checkout’



XSS Tier 2

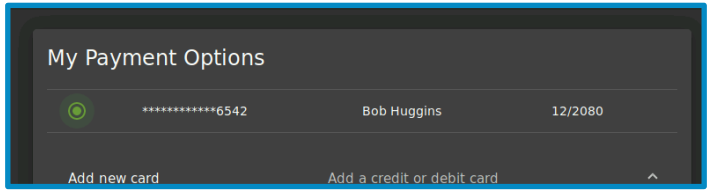
- Click Add New Address
- Enter a fake address
- Click **Submit**
- Select the address and then click **Continue**
- Choose any delivery speed, then **Continue**.



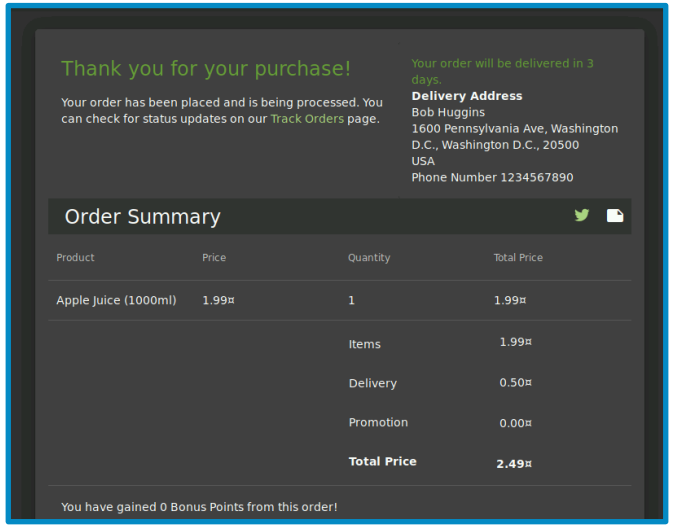


XSS Tier 2

- Click **Add New Card**
- Add a (fake) credit card, select it, then click **Continue**



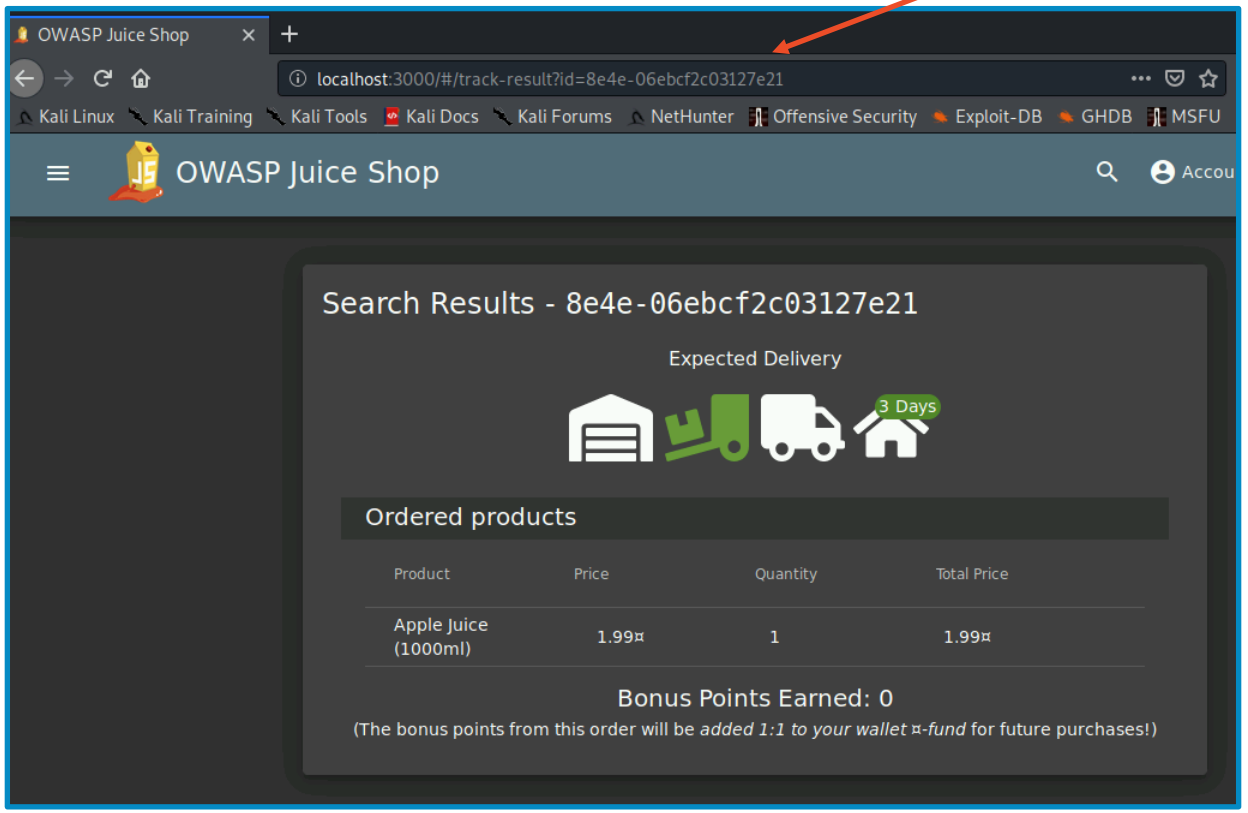
- Place your fake order



XSS Tier 2

- Click Track Orders

You are going to alter the URL



You should see something similar

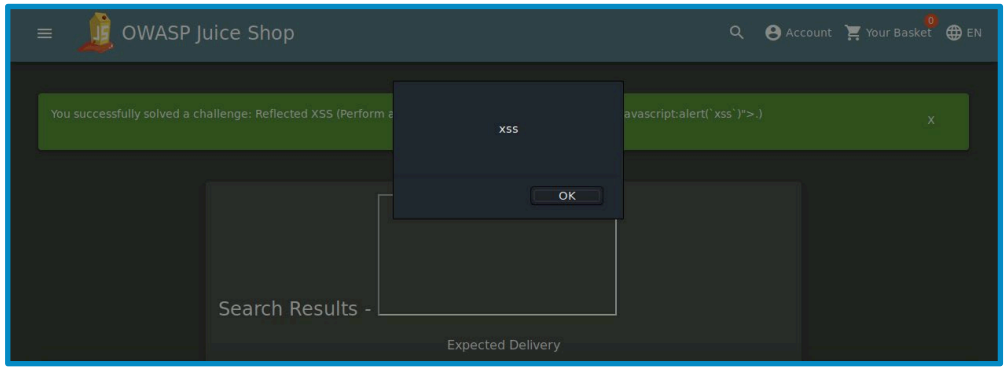
XSS Tier 2

- In the URL bar, replace everything after id= with

```
<iframe src="javascript:alert(`xss`)">
```

These are ticks and not apostrophes. The tick button is located to the left of the “!” and “1” button on the keyboard

- Hit **ENTER** and then refresh the webpage



You should see the alert appear.
You might have to refresh the page!

Defending Against a Cross-Site Scripting Attack

- Sanitize the inputs!
  - Reject inputs that are not what the search was meant for
  - NEVER trust user input – check it
  - “Escape” the user’s input
    - Escaping the user’s input will not run the data as HTML
    - Will not interpret the HTML
    - This will just display whatever was typed as is
    - Will display characters as they are
      - What is the importance of these characters: “<“ and “>” in a XSS attack?
- What are some other ways of defending against a Cross-Site Scripting attack?