# Week 3 Practical - Hacking Practical

# OWASP Juice Shop

This Lab will use the [OWASP Juice Shop](https://owasp.org/www-project-juice-shop/). *“the most modern and sophisticated insecure web application! It can be used in security trainings, awareness demos, CTFs and as a guinea pig for security tools! Juice Shop encompasses vulnerabilities from the entire*[*OWASP Top Ten*](https://owasp.org/www-project-top-ten)*along with many other security flaws found in real-world applications!” (OWASP, 2020)*

**Note :** These activities are taken from : <https://pwning.owasp-juice.shop/>

# Step 1 – Install the Juice Shop

Go to : <https://github.com/juice-shop/juice-shop>

Scroll down, under setup you will see that there are multiple was to get an instance of the Juice Shop running, if using your own device, you can choose any method,

**If using the PC’s in the labs use the NodeJS method (see below):**

* **Open a Command Prompt and clone the Juice Shop from GitHub:**

git clone https://github.com/juice-shop/juice-shop.git --depth 1

**A computer screen with white text

Description automatically generated**

* **Go into the cloned folder using the command : cd juice-shop**
* **Install the Juice shop using the command : npm install**
* **Once the install command has completed**
* **Start the Juice shop using the command : npm start**
* **Open a Browser :** [**http://localhost:3000**](http://localhost:3000/)

**Now we are ready to start!**

A screenshot of a computer

Description automatically generated

**NOTE:** This is your own instance of the Juice-Shop that is running. Create a dummy account for yourself, you do not need to use your real details, just the correct format, for example for your email address could be: [myname@myemail.me](mailto:myname@myemail.me)

**Complete the Steps 1-5 below**

### **Perform a reflected XSS attack**

Explain what Reflected Cross-site Scripting (XSS) is and how it occurs. Complete the steps below:

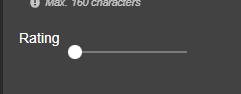
1. Log in as any user.
2. Paste the attack string <iframe src="javascript:alert(`You Have Just Created an XXS Attack!`)"> into the Search field.
3. Click the ***Search*** button.

### **Give a zero-star feedback to the store**

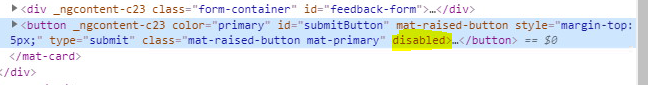
Make sure you are logged in, then place an order (use dummy payment info). Go to the *Customer Feedback* form and add a c*omment* text, solve the CAPTCHA at the bottom of the form, **do not give a rating!**

DO NOT TOUCH!

Do not give a rating!



1. The *Submit* button is still **disabled** because you did not select a *Rating* yet.
2. Inspect the *Submit* button with your DevTools (F12) and find the disabled attribute of the <button>HTML tag.
3. What happens if you delete it from the tag.



### **SQL Injection**

### **3a. Access the administration section of the store**

1. Do some detective work to find out the name of the admin page..
2. Open the **main.js** in your browser's developer tools (F12 in Chrome) and search for "admin".
3. One of the matches will be a route mapping to path to the "administration" page.
4. In your address bar, add on the name of the admin page to the URL, see if you can navigate to that page, will give a 403 Forbidden error.

([**http://localhost:3000/#/administration**](http://localhost:3000/#/administration))

### **3b. Use SQL Injection to log in with the administrator's user account**

1. Now lets use SQL injecton to log in as an administrator so that we can access the admin page!
2. Log in with:  **AnyEmail ' or 1=1--**  and any **Password** which will authenticate the first entry in the Users table which coincidentally happens to be the administrator!!
3. Now change the URL again to the administrator page and you should get access!

**Read this to learn another way to** [Retrieve a list of all user credentials via SQL Injection](https://bkimminich.gitbooks.io/pwning-owasp-juice-shop/content/appendix/solutions.html#retrieve-a-list-of-all-user-credentials-via-sql-injection)

### **Access the session cookie**

1. Log in as any user.
2. In the search field type :

**<iframe src="javascript:alert(document.cookie)">**

1. An alert box with the Session details should appear.
2. See how simple it is to get the webpage server to run **your** JavaScript code. Imagine what a hacker might do!
3. Try this code in the search field: (Sound on!)

**<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto\_play=true&hide\_related=false&show\_comments=true&show\_user=true&show\_reposts=false&show\_teaser=true"></iframe>**

1. Can you change the URL to play something else?

### **Add a script to re-direct the user to another webpage ?**

1. Try out the following commands in the search box to see what happens.
2. Remember to allow PopUps in your browser

**<iframe src=window.location.href="https://www.google.com/search?q=what+is+XSS">**

**<iframe src=javascript:window.location.href="https://demo.testfire.net">**

**<iframe src=javascript:window.open("https://www.google.ie")>**

**<iframe src=javascript:window.open("https://www.google.com/search?q=what+is+xxs")>**

### **Extra: – JUST FOR FUN**

* You should also have a go at the other XXS Challenges on the OWASP Juice shop:
* <https://pwning.owasp-juice.shop/companion-guide/latest/part2/xss.html>