OWASP Top 10 Part 1

Module 9 | Activity 1

Introduction

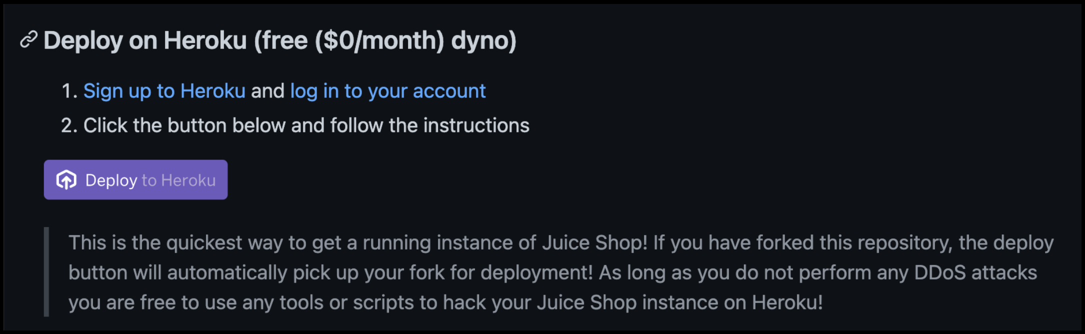
In order to help you practice identifying the OWASP Top 10 in the real world. The OWASP Foundation hosts a project named "OWASP Juice Shop" which is basically an insecure web application with all types of vulnerabilities. In this activity you will put your bug-hunting skills to the test on a virtual marketplace.

As a beginner it might be difficult to figure these out on your own but do not be discouraged. After each of the following sections, you will be able to follow a walkthrough of the exercise. These will be especially useful for you to remember during the Spring Cyber Defense Competition.

Running Juice Shop

Since Juice Shop is a web application we need to find a way to have our own copy of it on our computer and not affect anybody else's work. There are many ways to do this. The fastest way to get started is to use the cloud application platform, Heroku.

- First, navigate to the Juice Shop github repository, by following the link...https://github.com/bkimminich/juice-shop this is basically where the Juice Shop developer posts the project components.
- Scroll down and right under the table of contents you will find the instructions to deploy the application using Heroku. Click on the purple button to create an account, it's free.

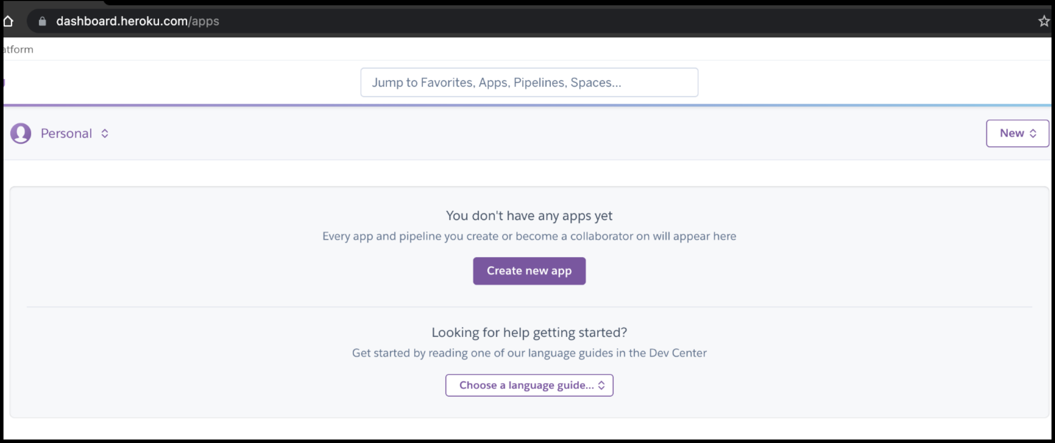


Deploy on Heroku (free ($0/month) dyno)

1. Sign up to Heroku and log in to your account
2. Click the button below and follow the instructions

> This is the quickest way to get a running instance of Juice Shop! If you have forked this repository, the deploy button will automatically pick up your fork for deployment! As long as you do not perform any DDoS attacks you are free to use any tools or scripts to hack your Juice Shop instance on Heroku!

- Logging in should take you to the Heroku main page...



dashboard.heroku.com/apps

You don't have any apps yet

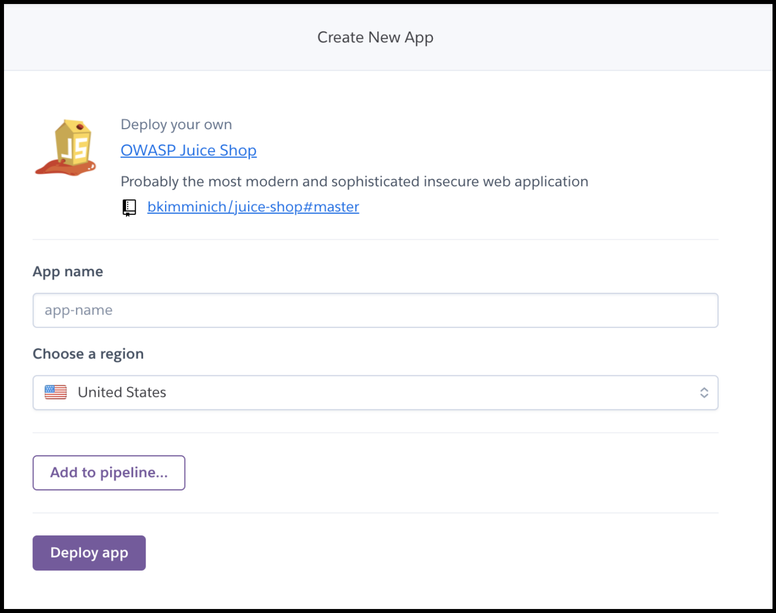
Every app and pipeline you create or become a collaborator on will appear here

Looking for help getting started?

Get started by reading one of our language guides in the Dev Center

Jump to Favorites, Apps, Pipelines, Spaces...
Personal
New
Create new app
Choose a language guide...

- After creating your account you need to go back to the OWASP github and deploy an instance of the application by clicking the purple button from the second bullet point. Name your app and deploy it.



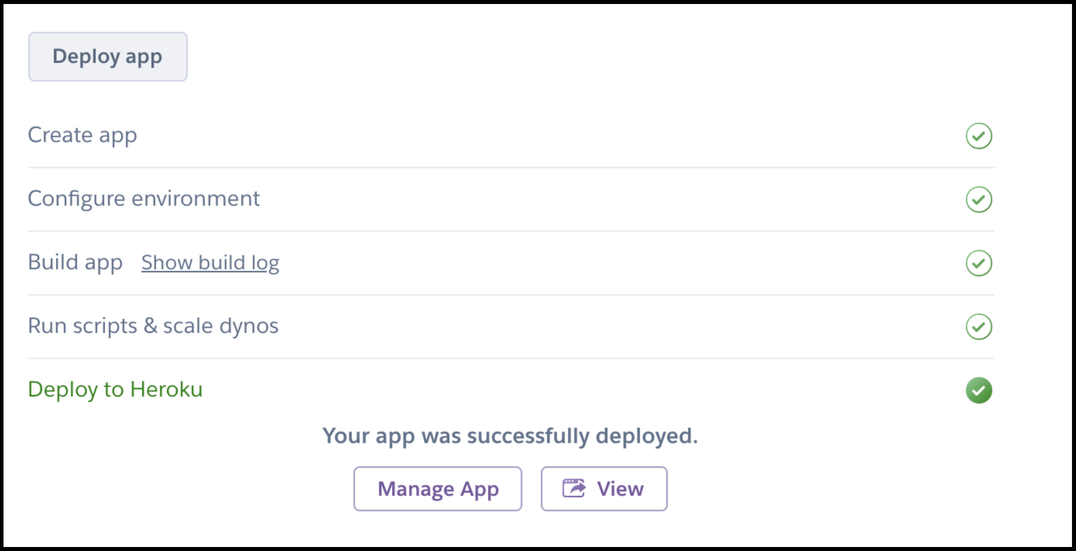
Create New App

Deploy your own
OWASP Juice Shop
Probably the most modern and sophisticated insecure web application
bkimminich/juice-shop#master

App name: app-name

Choose a region: United States

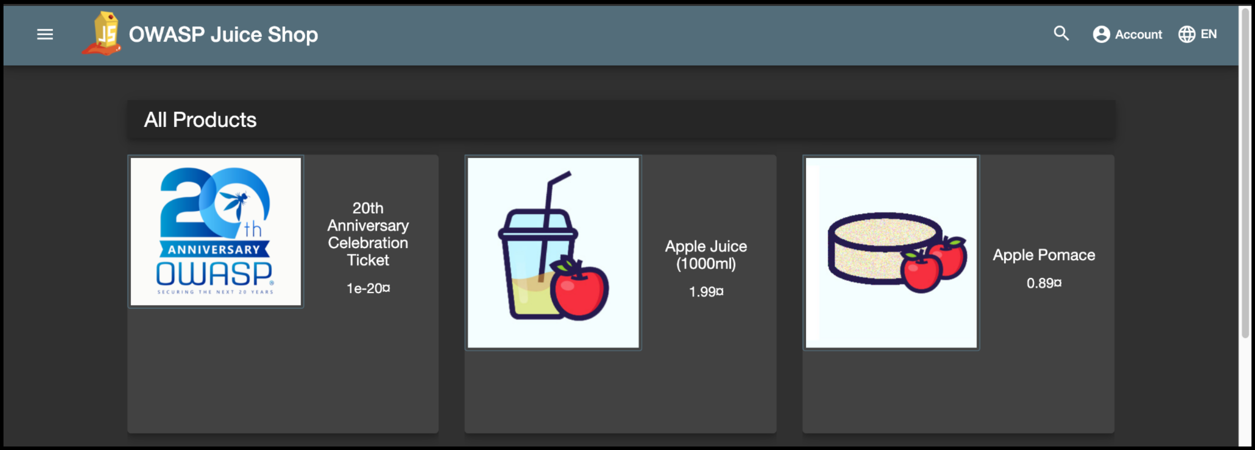
- Wait a few minutes while the app loads and does its thing. You might have to click "View", just in case the app doesn't automatically open.



Create app ✓
Configure environment ✓
Build app Show build log ✓
Run scripts & scale dynos ✓
Deploy to Heroku ✓

Your app was successfully deployed.

- Once you launch Juice Shop, you should see something resembling an online marketplace environment.



- Feel free to use the "[Pwning OWASP Juice Shop]" handbook as a resource while you search for The Top 10.

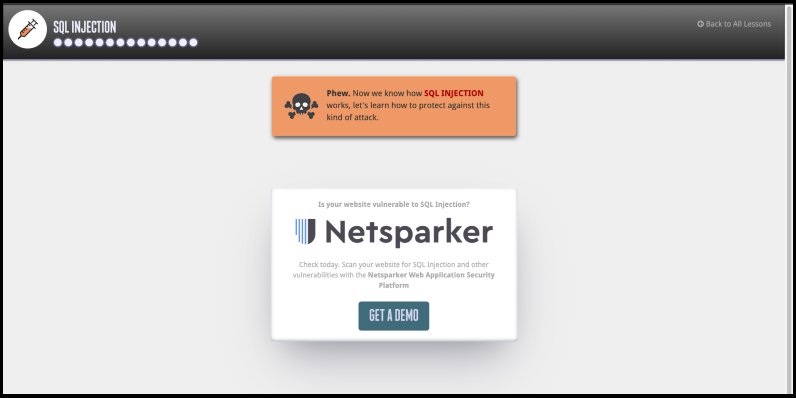
SQL Injection

Injections, and in particular SQL injections are one of the most common security vulnerabilities on web applications across the internet today.

SQL which stands for "Structured Query Language" is used to communicate with a database to manage the information stored inside. When you use an input field within a website it is also possible to trick the database using SQL (which is basically just a language) into showing you information you are not supposed to access. That is the essence of a SQL injection.

- Navigate to the following website [link], run through the introductory activity with your team on hacksplaining.com
  - As you run through the activity, think about the command you used in order to gain access to the bank account without having the real password. Why does this work?
  - Once you complete the activity move on to part two. Which covers some remediation strategies for the injection vulnerability. Click on the orange

rectangle. If the page is not available for whatever reason, use a search engine to look up some of these methods on your own.



Phew. Now we know how **SQL INJECTION** works, let's learn how to protect against this kind of attack.

Is your website vulnerable to SQL Injection?

Netsparker

Check today. Scan your website for SQL Injection and other vulnerabilities with the **Netsparker Web Application Security Platform**

- Now that you have some exposure to this type of vulnerability we will look for that same flaw in Juice Shop. Hint: Go to the sign-in page...
- Once you make it to the sign in part of Juice Shop take a few minutes to look up some SQL injection commands that might be able to help you. Maybe even the one from the "Modern Rogue" video.
- You know you have been successful at exploiting this type of vulnerability when you are able to sign in the admin or any other user.

Broken Authentication

When we think about what a broken authentication vulnerability entails, we can remember that it basically revolves around the idea of poor authentication practices. Such as NOT implementing two-factor authentication or the existence of rogue accounts with weak passwords, poor session management practices etc.

Go back to the Juice Shop app and try to identify a B.A vulnerability.

Hint: Think of a few easy to guess passwords, look some up, there are tons of lists of unsafe credentials people use all the time. Try to login as the admin without using an injection.

Sensitive Data Exposure

For this type of vulnerability you will have to find some type of sensitive item. Which could take the form of an email, document or log.

To complete this section you will need to look around Juice Shop's page for something secret or classified. If you haven't already noticed. You will receive a challenge complete green banner when you have successfully completed a challenge.

Hint: Open the side menu and navigate to the About Us tab. What you will see is a bunch of latin words which are just meant as fluff. But what do you see in that sea of unknown words? This link will take you to the company's fake Terms of Service. And this page honestly looks like a dead end. But if you look closely at the url bar you will notice that the document's path is actually listed on the link. The directory containing this file is the ftp directory. (FTP is a file management system of sorts for websites) Now remove the last part of the link and hit enter, where does this take you? Why is it bad that a hacker could know about future company acquisitions?

XML External Entities XXE

This might be the most confusing security risk out of the top ten and there are only two examples in Juice Shop of this type of vulnerability. Both of which have a medium to high difficulty rating so we will skip this one.

The essence of this entry is that an XXE attack, if successful, would allow a hacker to probe your file system. XML is a type of markup language useful for both humans and computers that provides a universal set of rules for documents on the internet. More often than not an external entity will be consulted to validate the correctness of files on a web application. If an attacker manages to play their cards properly they will be able to assume the role of an external entity and demand files from your servers. Including sensitive password files.

Broken Access Control

In a broken access control misconfiguration, authenticated users are not properly assigned their privileges. User privileges in the realm of web applications can be defined as the actions a user is allowed to perform once signed in.

A manager will have a lot more control over their company website than the average-Joe employee. In this section you will attempt to find an instance of this vulnerability.

Wouldn't it be weird if you could see what your neighbor is placing in their cart on Amazon?

Let's see if we can do something similar on Juice Shop, shall we?

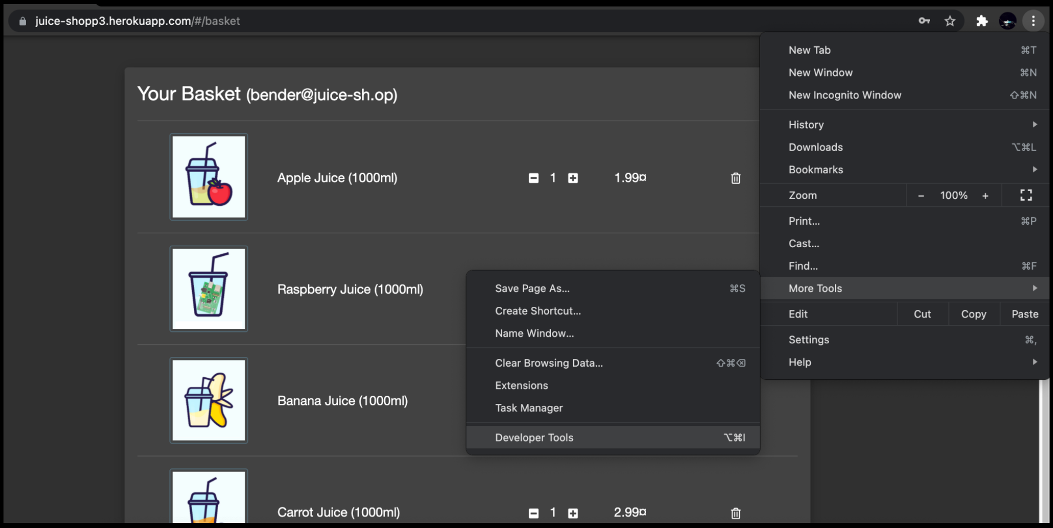
Log in as Bender. Using an injection.

Email: bender@juice-sh.op'--

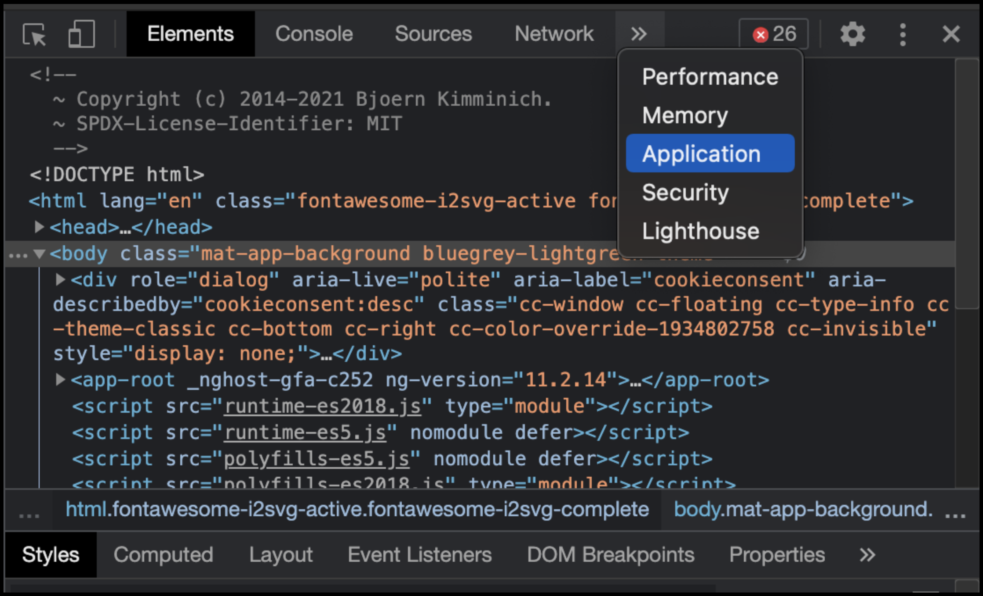
Password: *anything*

Add a few items to your cart.

While still on the Juice Shop Basket page go to your browser settings and look for Developer Tools...



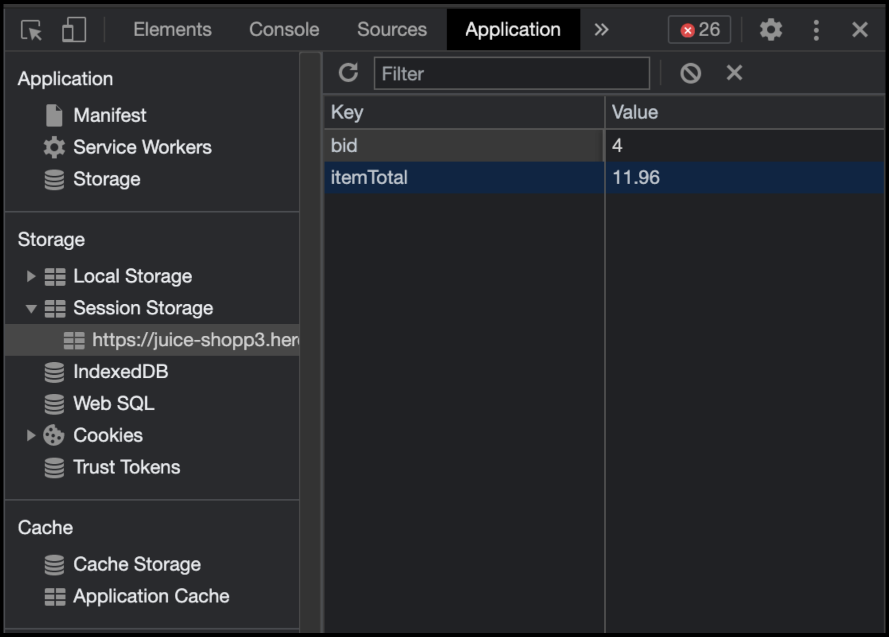
Once you have made it to developer tools look for the Application tab and click on it.



```
<!--
  ~ Copyright (c) 2014-2021 Bjoern Kimminich.
  ~ SPDX-License-Identifier: MIT
  -->
<!DOCTYPE html>
<html lang="en" class="fontawesome-i2svg-active fo                    complete">
 ▶<head>…</head>
…▼<body class="mat-app-background bluegrey-lightgre
   ▶<div role="dialog" aria-live="polite" aria-label="cookieconsent" aria-
   describedby="cookieconsent:desc" class="cc-window cc-floating cc-type-info cc
   -theme-classic cc-bottom cc-right cc-color-override-1934802758 cc-invisible"
   style="display: none;">…</div>
   ▶<app-root _nghost-gfa-c252 ng-version="11.2.14">…</app-root>
    <script src="runtime-es2018.js" type="module"></script>
    <script src="runtime-es5.js" nomodule defer></script>
    <script src="polyfills-es5.js" nomodule defer></script>
```

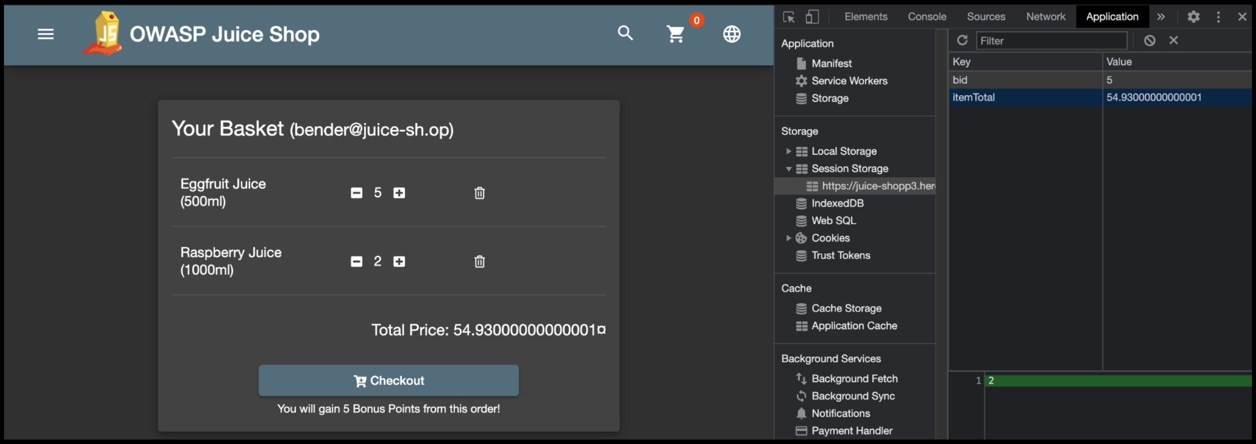
```
…  html.fontawesome-i2svg-active.fontawesome-i2svg-complete   body.mat-app-background. …
```

Now look for the bid value and change it by clicking on the number, in this case “4” and change it by adding the value of one to it. The bid value could be interpreted as UserID, so if you change it you also change the user.



Key	Value
bid	4
itemTotal	11.96

After you change “bid’s” value you may reload the page by simply hitting refresh. You will notice that your cart is now different. I had four different items in my original cart but now I only have two pairs of two items. Clearly we exploited a B.A.C vulnerability.



Your Basket (bender@juice-sh.op)

Eggfruit Juice (500ml) 5

Raspberry Juice (1000ml) 2

Total Price: 54.93000000000001¤

You will gain 5 Bonus Points from this order!

Key	Value
bid	5
itemTotal	54.93000000000001

```
2
```

KEY

As always, the solutions to almost everything can be found on the internet hidden behind a paywall. Thanks to the generosity of the people behind the OWASP Juice Shop project the challenge answers exist in the appendix of the manual listed above. But here is the link once again: https://bkimminich.gitbooks.io/pwning-owasp-juice-shop/content/appendix/solutions.html

They are organized by difficulty.

Before you take a look at them however, put a lil more thought into it. And remember, the answers were within you all along...