CSC 515: Spring 2023 – Workshop 5: Insufficient Authentication and Access Controls

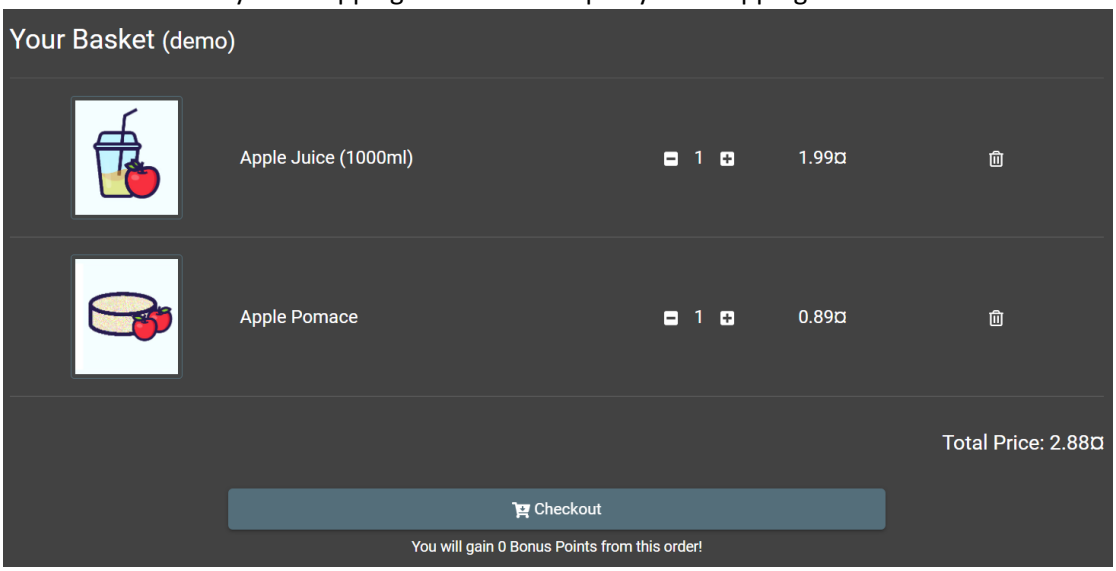
In this workshop, we will exploit a vulnerable web application by exploiting insufficient authentication and access controls

For this workshop, you'll need to complete the following tasks and then answer questions in Gradescope.

#	Task	Description
1	Other Shopping Baskets	Access another customer's shopping basket
2	Submit Unauthorized Feedback	Submit feedback that appears to be provided by other customers
4	On Your Own	Perform additional attacks and report on your progress

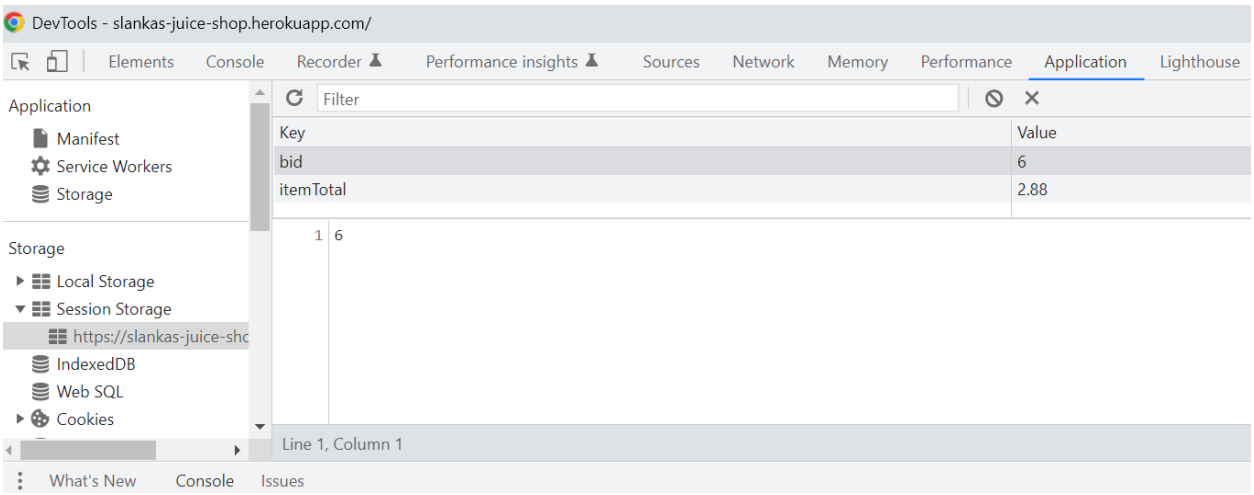
Access Another Customer's Basket

1. Open the OWASP Juice Shop application. Login as any user (e.g., demo/demo) Make sure your browser's developer tools are open and currently monitoring network requests.
2. Add some items to your shopping basket. Then open your shopping basket.



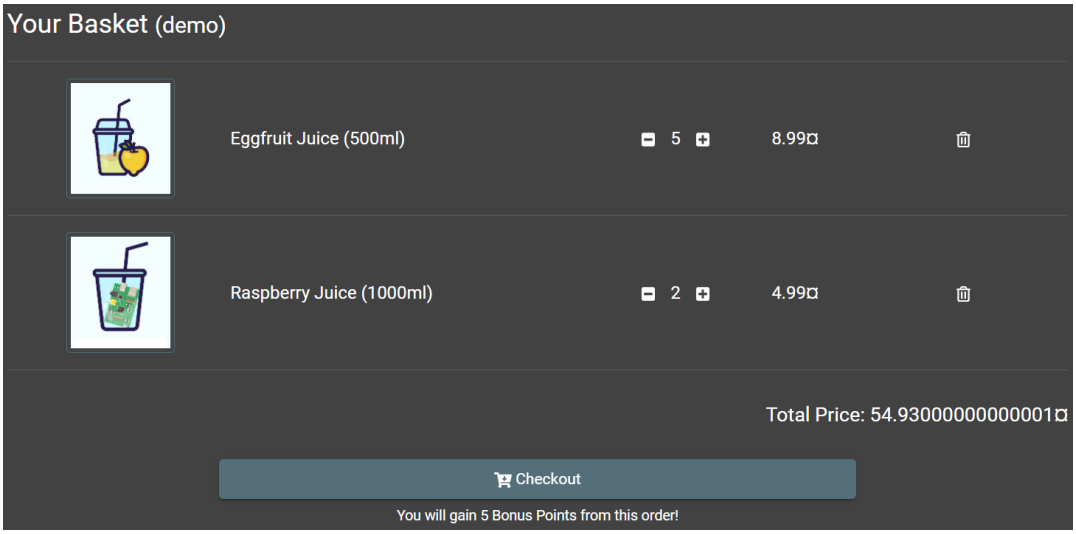
Shopping basket with some products

3. Check the session storage in your browser's developer tools (*note: in Chrome this is under Applications - Storage*). Notice there's a bid with some value.



Session Storage Information

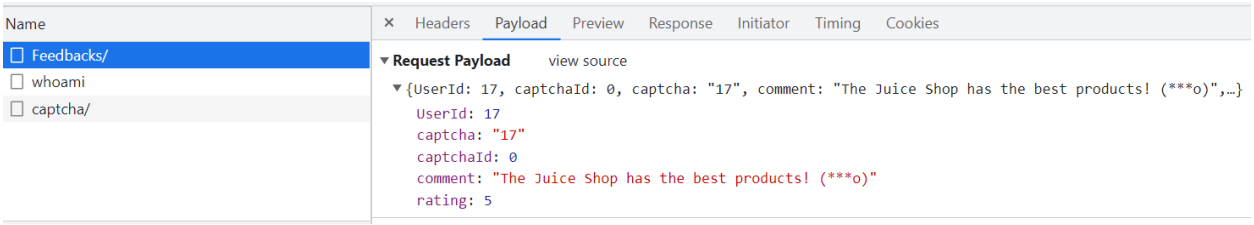
4. Edit the bid value to change to some other value (like 5). Reload the shopping basket page.



Viewing another customer's basket

Submit Unauthorized Feedback

1. Login to the OWASP Juice Shop as any user (for example: username demo and password demo). Make sure your browser's developer tools are open and is currently monitoring network requests.
2. Open the Customer Feedback page:
3. Fill-out the form with values and submit. In the browser's developer tools, notice the request that was submitted:



Feedback request submitted

Notice the information in the request payload for the request.

4. Let's create a custom request to try to submit feedback with a different customer's id. In the developer's console, submit the following script:

```
fetch('<YOUR_SERVER_INFO>/api/Feedbacks/', {
  method: 'POST',
  body: JSON.stringify({
        "UserId":<ANY USER ID>,
        "captchaId":<CAPTCHA ID>,
        "captcha":"<CAPTCHA ANSWER>",
        "comment":"<COMMENT TEXT>",
        "rating":<RATING>
  }),
  headers: {
    'Content-type': 'application/json; charset=UTF-8'
   }
})
.then(console.log)
```

where the YOUR_SERVER_INFO is the URL of your Juice Shop instance, CAPTCHA ID is the id of the captcha used when you submitted the original form, and CAPTCHA ANSWER was the answer to the captcha when you submitted the form. Fill-in the values for UserId, comment, and rating as you would like.

5. What happened when you submitted the request?

```
> fetch('https://slankas-juice-shop.herokuapp.com//api/Feedbacks/', {
    method: 'POST',
    body: JSON.stringify({
      "UserId":15,
      "captchaId":0,
      "captcha":"17",
      "comment":"For those who are about to rock, we salute you...",
      "rating":5
    }),
    headers: {
      'Content-type': 'application/json; charset=UTF-8'
     }
  })
  .then(console.log)
▶ r {__zone_symbol__state: null, __zone_symbol__value: Array(0)}
▼ Response {type: 'basic', url: 'https://slankas-juice-shop.herokuapp.com//api/Feedbacks/', redirected: false, status: 201, ok: true, …}
    body: (...)
    bodyUsed: false
  ▶ headers: Headers {}
    ok: true
    redirected: false
    status: 201
    statusText: "Created"
    type: "basic"
    url: "https://slankas-juice-shop.herokuapp.com//api/Feedbacks/"
  ▶ [[Prototype]]: Response
```

Feedback Sent

Notice that we didn't have to include any Authorization header with the request! That means you could logout of the software and still be able to submit feedback without being logged in and by providing random customer IDs.

On Your Own

In Moodle, answer the questions in the Workshop 5 activity by performing attacks in which the application does not appropriately validate authentication and access control:

- Use the product rating form (on the shop homepage, click a product's image to open information about the product, which includes the review form) to submit a review that appears to be submitted by a different customer.
- Update the "Apple Pomace" product to change the "sent back to us" URL to instead link to https://www.csc.ncsu.edu.