Security Testing

Requirement 1. Setup your security testing environment

1. Deploy OWASP Juice Shop virtual machine.

* **Use Case:** OWASP Juice Shop server
* **CPUs:** 2
* **RAM:** 2 GB
* **Operating System:** Ubuntu 22.04
* **Network Adaptor 1:** NAT
* **Network Adaptor 2:** 192.168.123.1 (Internal Network)
* **Source:** Pre-built, cloned, or ISO
* **Install:** Node.js version 16, npm, git
* **Source for OWASP Juice Shop:** **git clone** [**https://github.com/juice-shop/juice-shop.git**](https://github.com/juice-shop/juice-shop.git) **–depth 1**
* **Credentials:** User: student, Password: Student1

1. Deploy Kali Linux (or ParrotOS) virtual machine for security testing.

* **Use Case:** Security Testing workstation
* **CPUs:** 2
* **RAM:** 4 GB
* **Operating System:** Kali Linux
* **Network Adaptor 1:** NAT
* **Network Adaptor 2:** 192.168.123.10 (Internal Network)
* **Source:** Pre-built, cloned, or ISO
* **Credentials:** User: kali, Password: kali

1. Configure virtual machines with specified specifications.
   * OWASP Juice Shop: 2 CPUs, 2 GB RAM, Ubuntu 22.04, NAT, Internal Network, NodeJS, npm, git.
   * Security Testing Workstation: 2 CPUs, 4 GB RAM, Kali Linux, NAT, Internal Network.

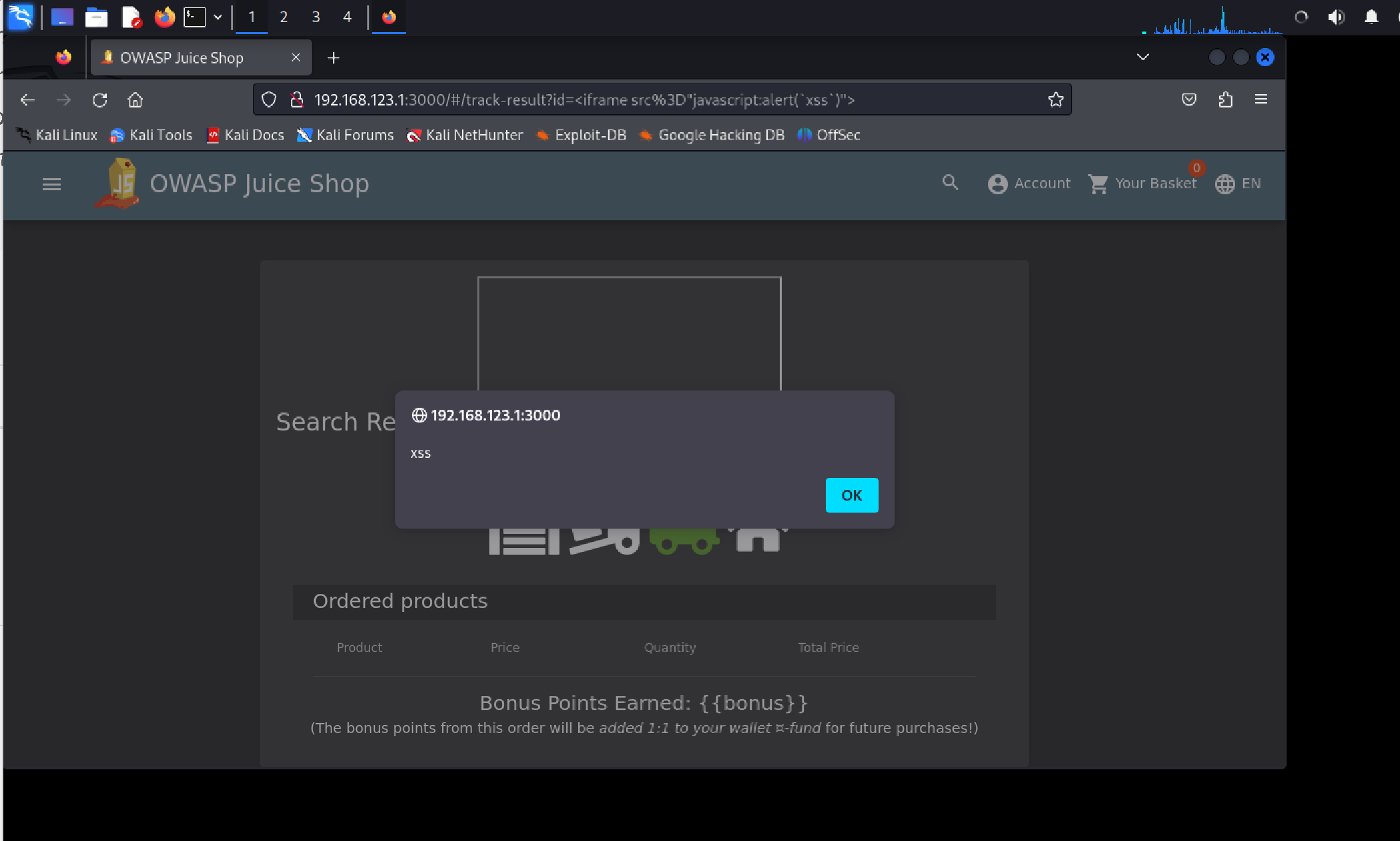
Requirement 2. OWASP Juice Shop Functionality Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Description** | **Inputs** | **Outputs** | **Expected Behavior** |
| User Registration | Allows users to create an account on the OWASP Juice Shop. | Username, Email, Password | Success/Failure messages | Successful registration redirects to the login page. |
| Login | Enables users to log in to their accounts. | Username, Password | Success/Failure messages | Successful login redirects to the user dashboard. |
| Product Listing | Displays a list of available products for purchase. | N/A | List of products | Products are displayed with details and images. |
| Product Details | Shows detailed information about a selected product. | Click on a product | Detailed product information | Displays details like price, description, etc. |
| Add to Basket | Allows users to add products to Your Basket. | Select a product | Success/Failure messages | Product is added to the basket. |
| Your Basket | Shows the contents of the user's shopping cart. | Specify quantity | List of items in the cart | Displays items with quantity and total cost. |
| Checkout | Initiates the select an address process from the pre-existing addresses or add new address. | Confirm the selected address. | List of existing addresses | Proceeding successfully redirects to adding a new address or continuing to delivery options. |
| Add New Address | Initiates the process of adding a n address | Country, Name, Mobile Number, ZIP Code, Address, City, State | Success/Failure messages | Successful registration redirects to the address page. |
| Delivery | Displays a list of available delivery speed options. | Click on a delivery option. | List of delivery options. | Displays details like price, expected delivery etc. |
| Payment | Handles payment processing for the user's order. | Payment details | Order confirmation | Successful payment redirects to order confirmation. |
| User Profile | Allows users to view and edit their profile information. | Click on the user profile icon | User details, Edit options | Edits are saved, and updated details are displayed. |
| Security Dashboard | Displays security-related statistics and information. | Access through the menu | Security metrics, Recent activities | Provides insights into security status. |

Requirement 3. Exploit a Cross-Site Scripting vulnerability

API-only XSS

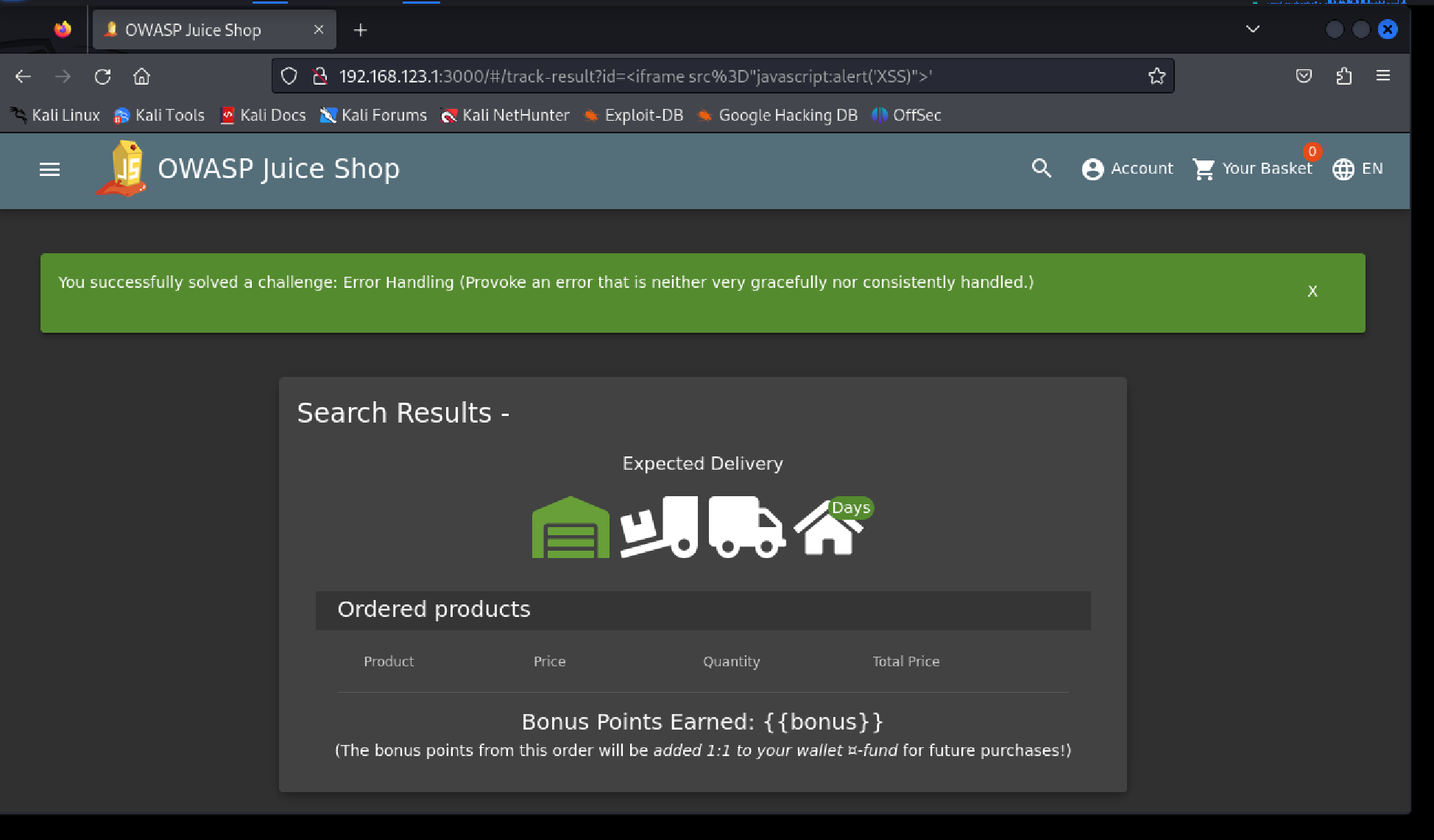
Perform a *persisted* XSS attack with <iframe src="javascript:alert(`xss)">` by running it in the API.



Payload executed successfully in the API, triggering an alert.

- Output is :

XSS



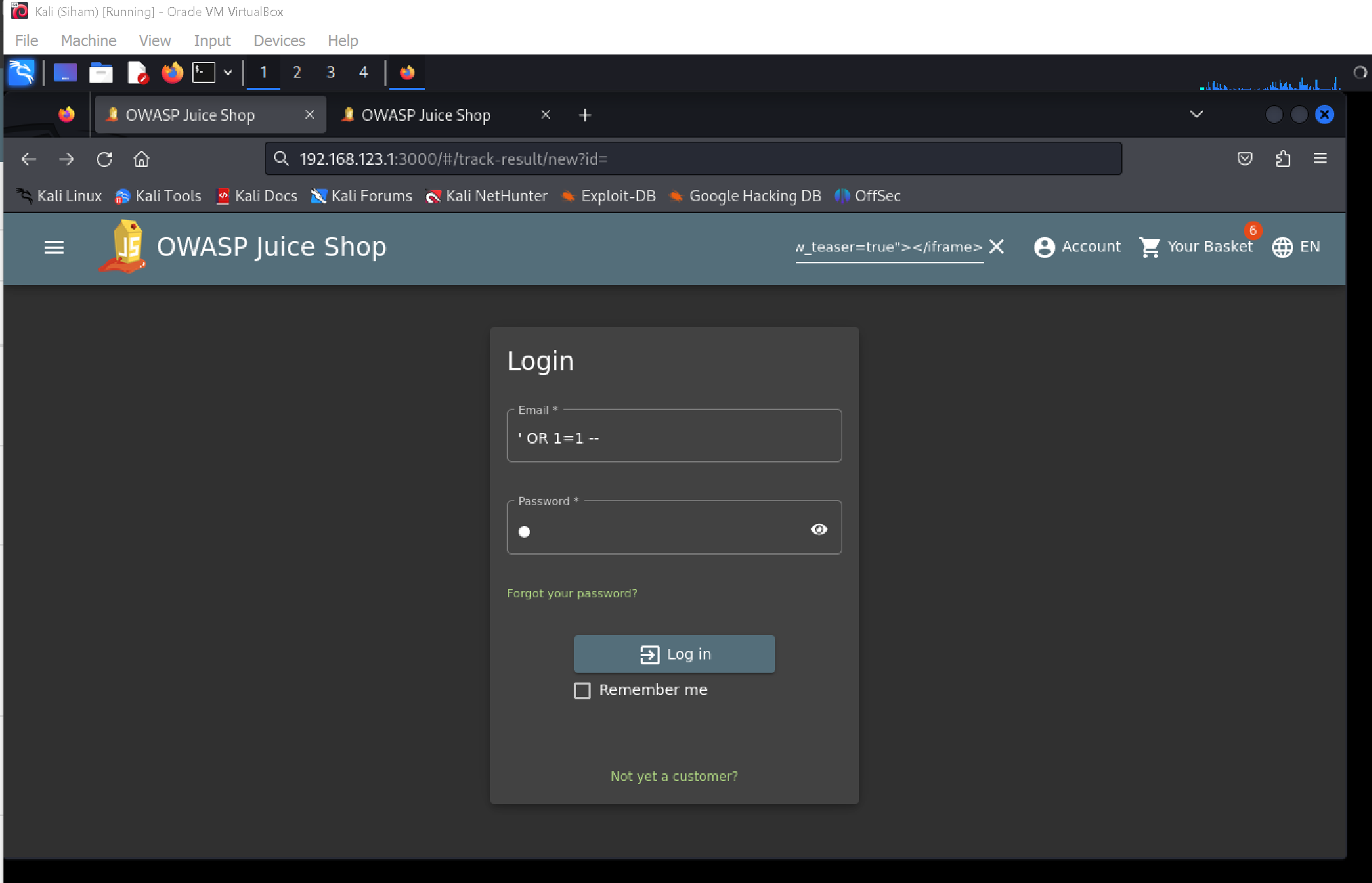
Bonus Payload

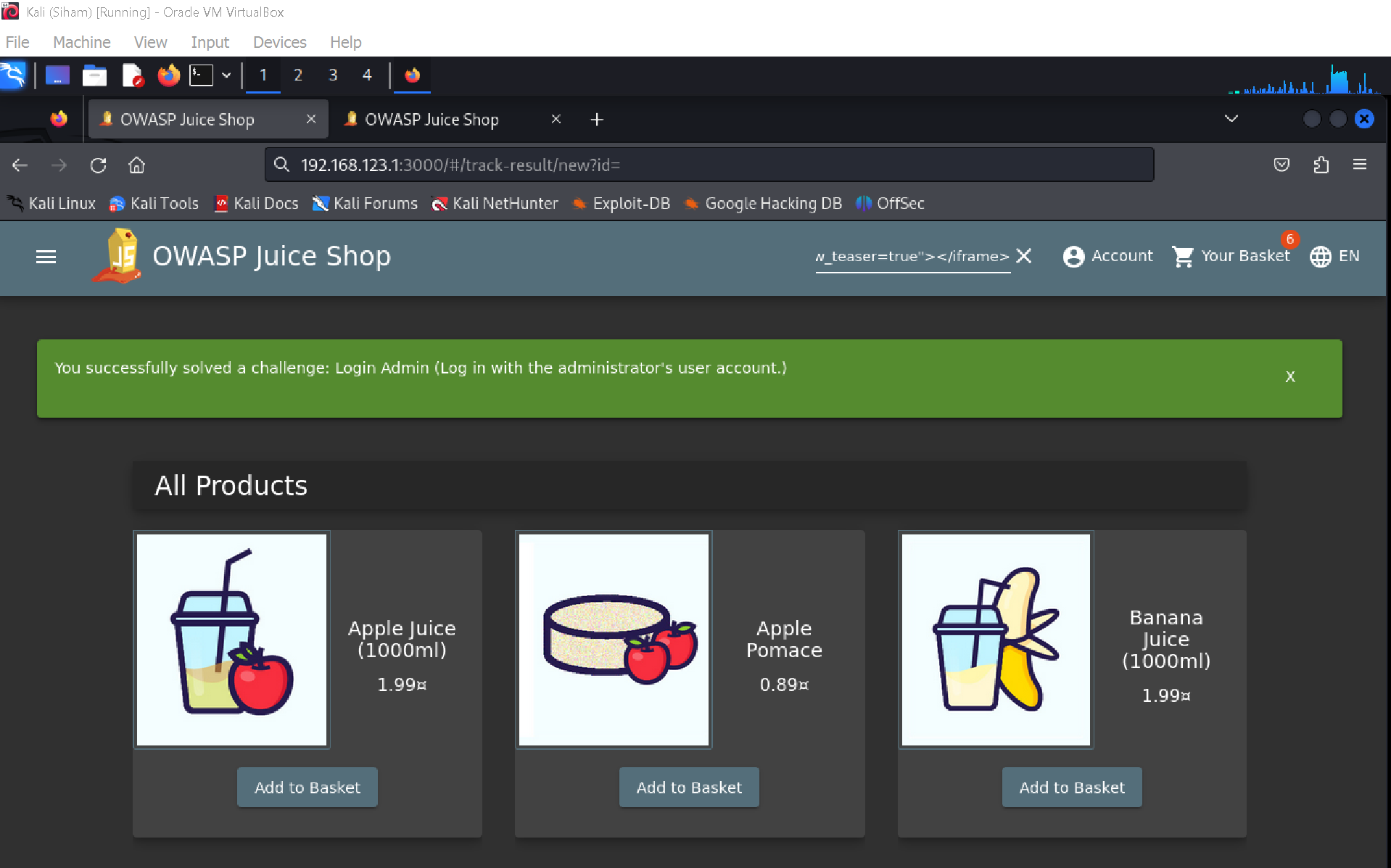
Use the bonus payload <iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto\_play=true&hide\_related=false&show\_comments=true&show\_user=true&show\_reposts=false&show\_teaser=true"></iframe> in the *DOM XSS* challenge.

Requirement 4. Exploit a SQL injection vulnerability.

Login Admin

Log in with the administrator’s user account.



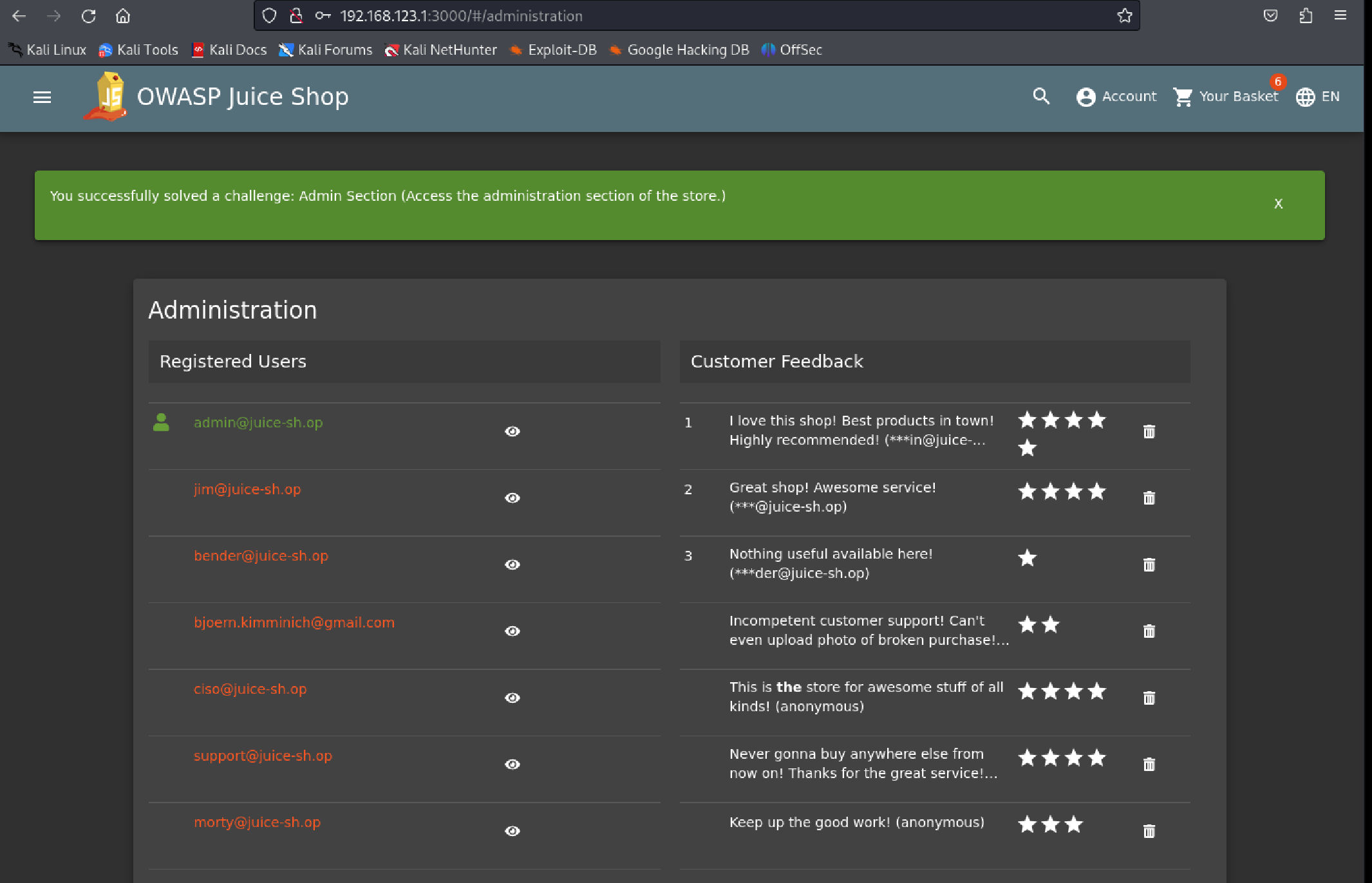


Requirement 5. Exploit a Broken Access Control vulnerability.

Admin Section

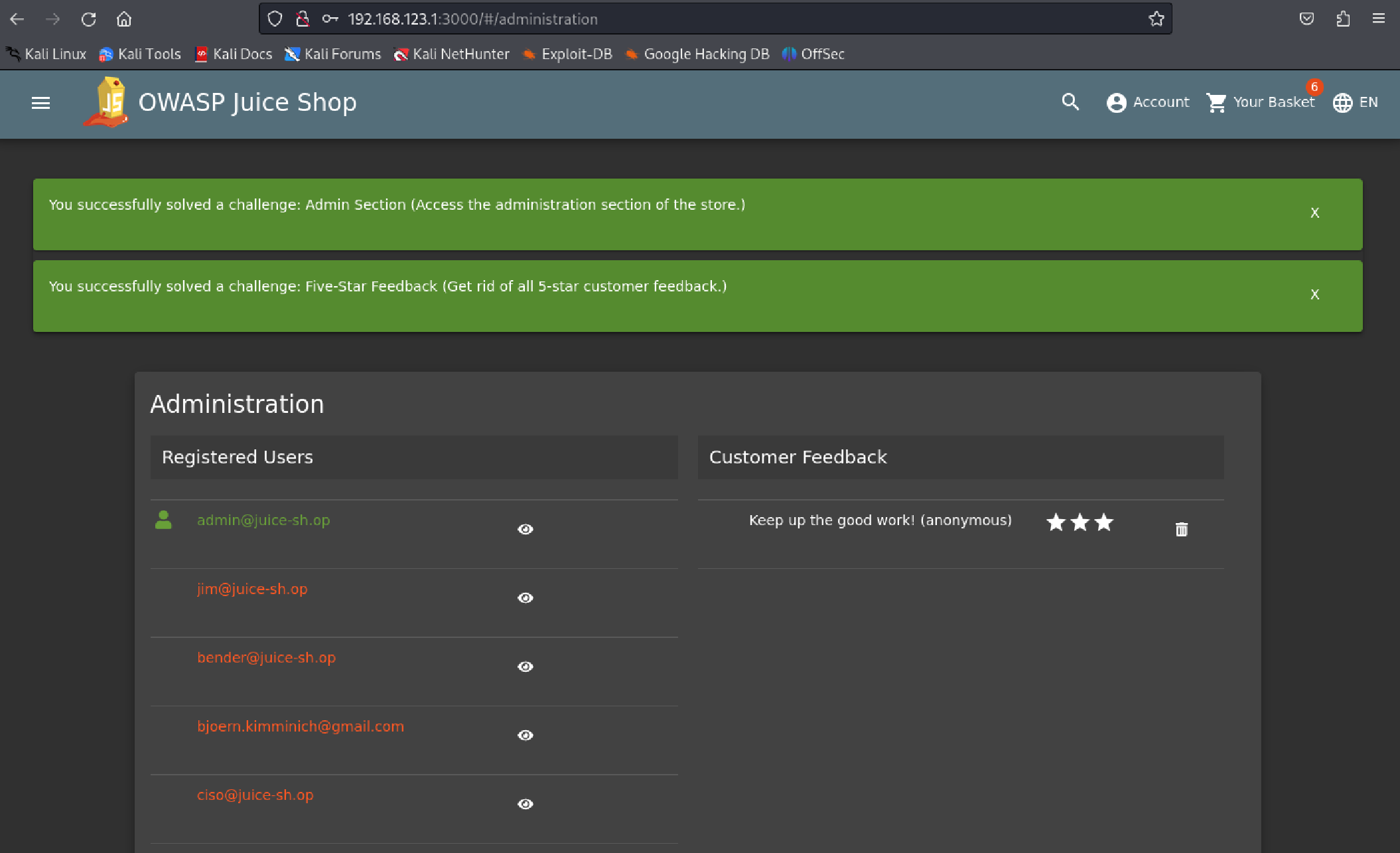
Access the administration section of the store.

* By adding the word administration into the url.



Five-Star Feedback

Get rid of all 5-star customer feedback.

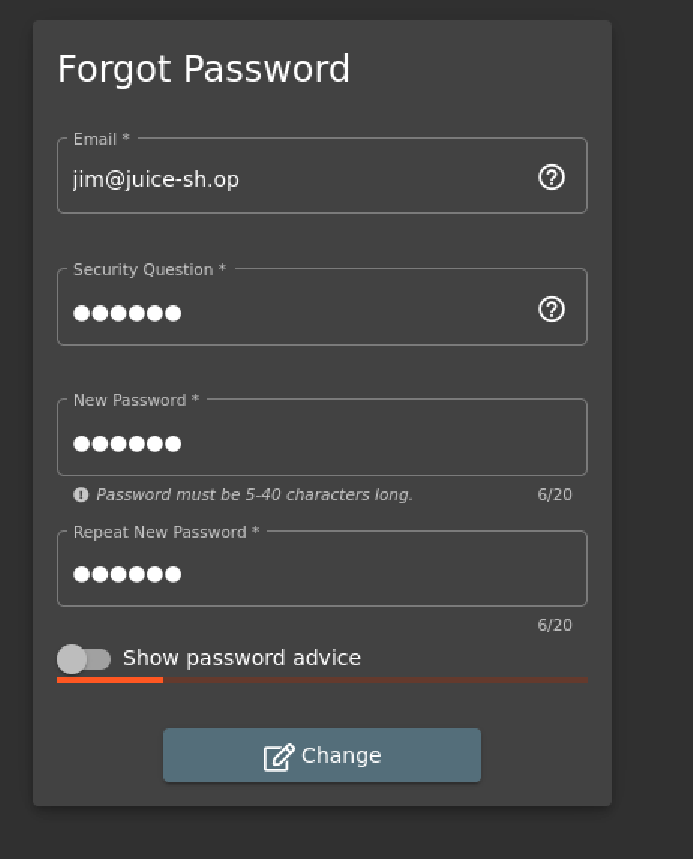


Requirement 6. Exploit an Authentication Bypass vulnerability.

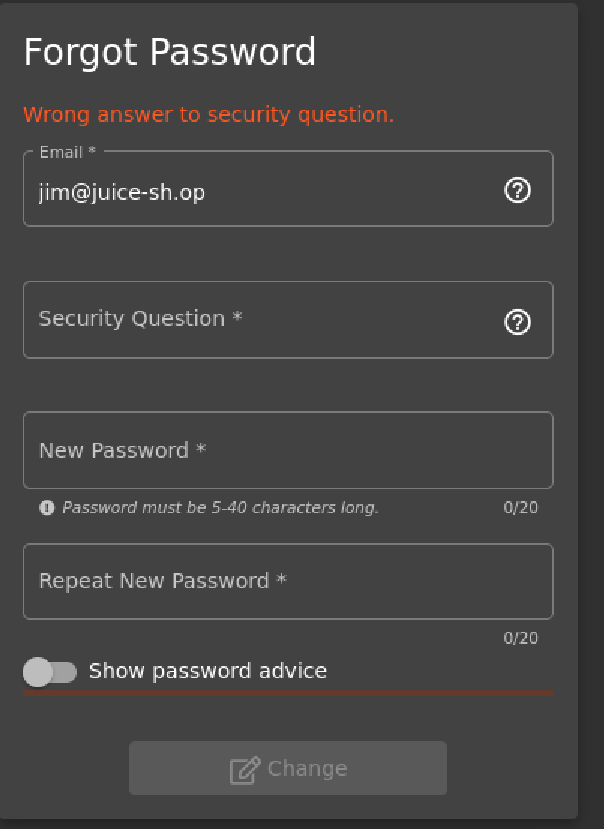
Reset Jim’s Password

Reset Jim’s password via the Forgot Password mechanism with *the truthful answer* to his security question.

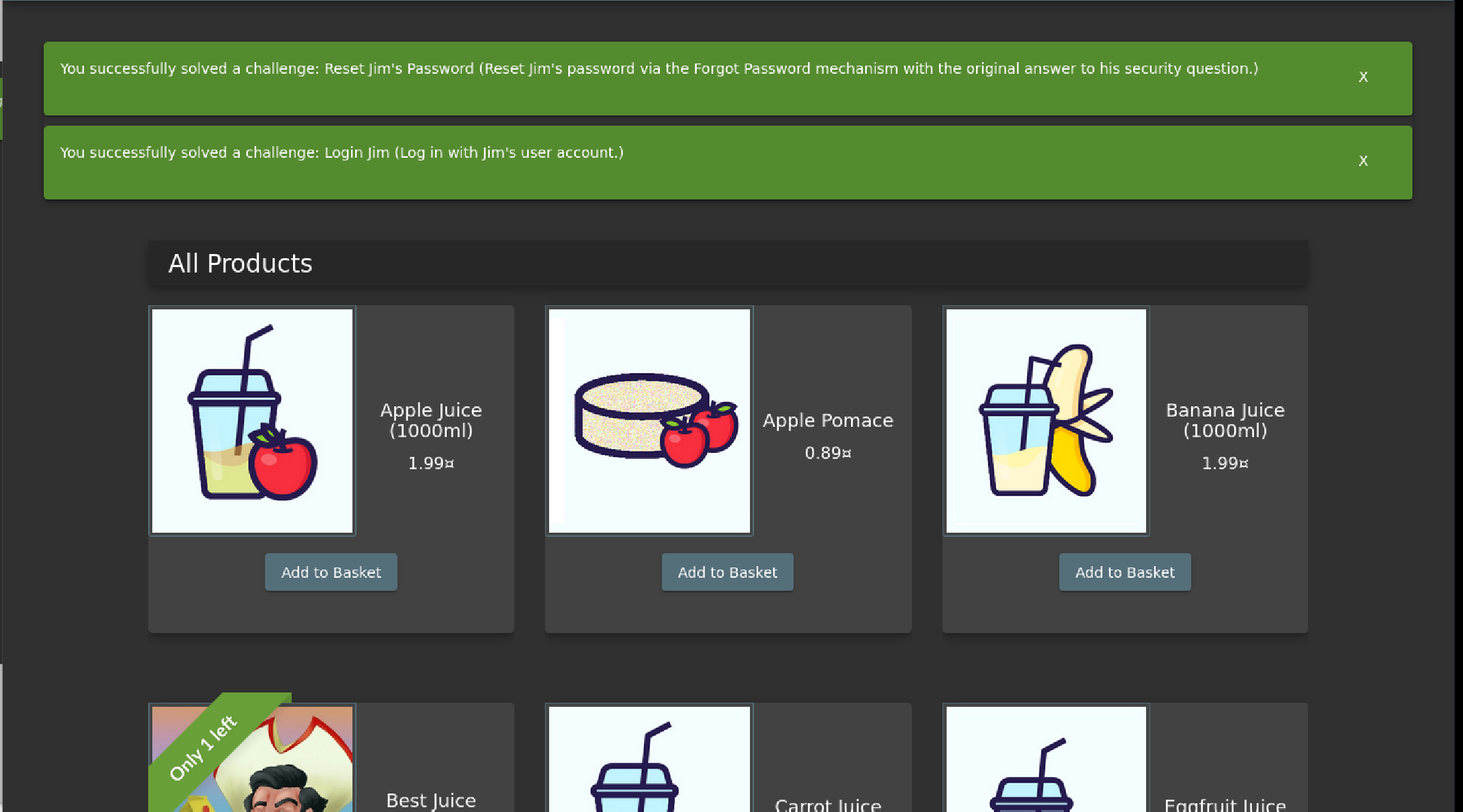
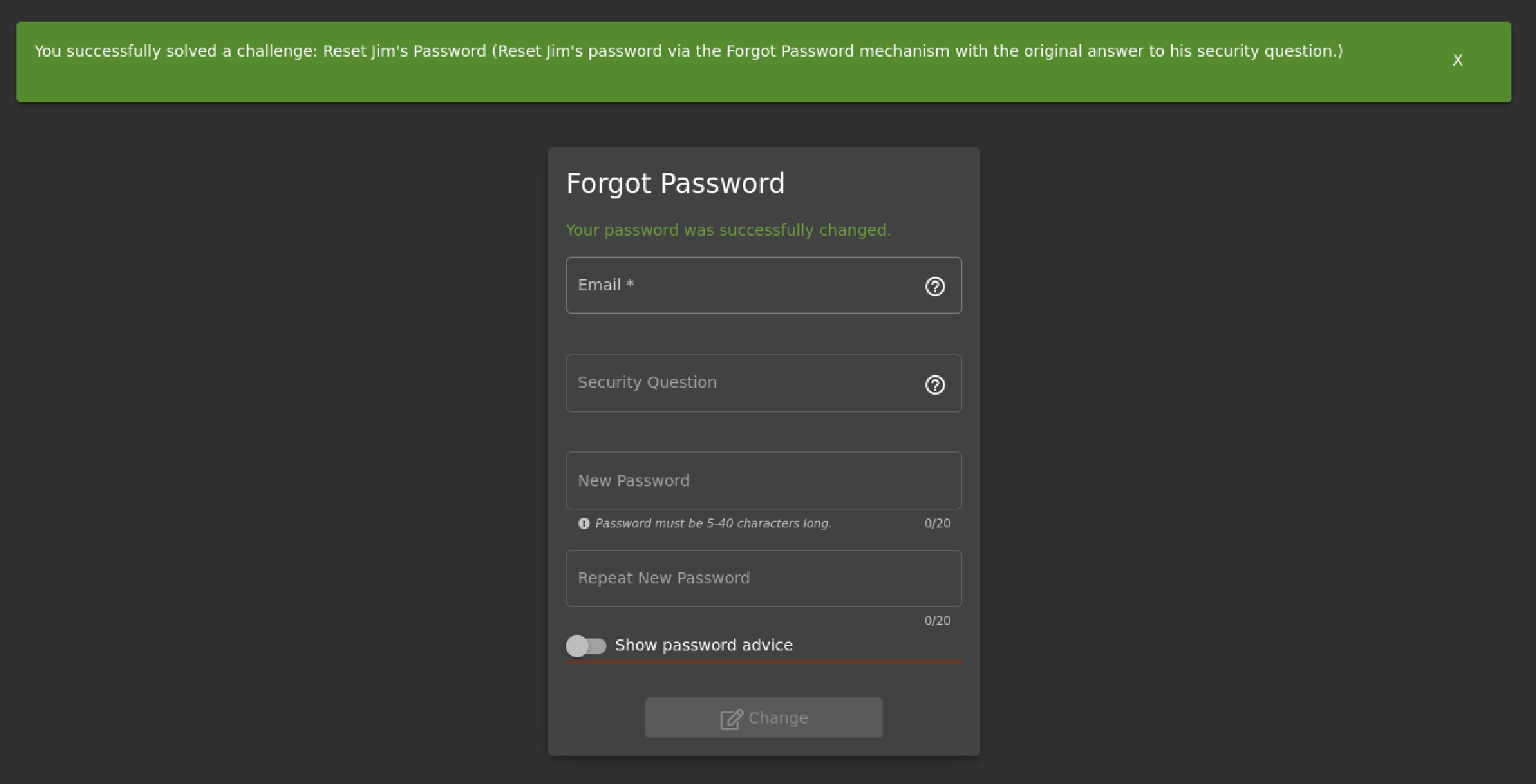
By googling jim I was able to find out that jim is James Tiberius Kirk and that the name of his older brother is George Samuel Kirk.



After misspelling the name Samuel I got an alert stating it was the wrong response.



After the correct spelling, I changed the password then login to Jim’s account.

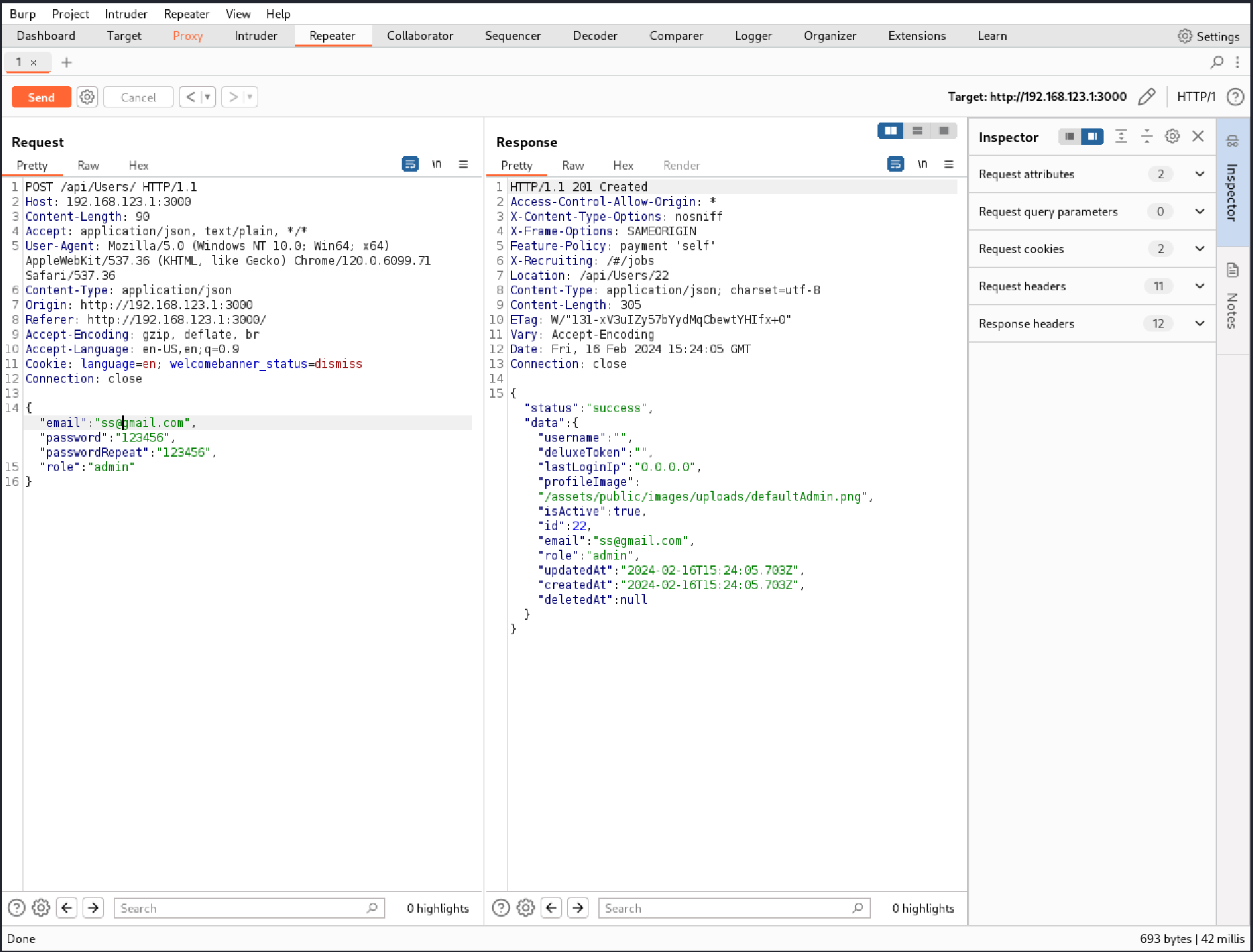


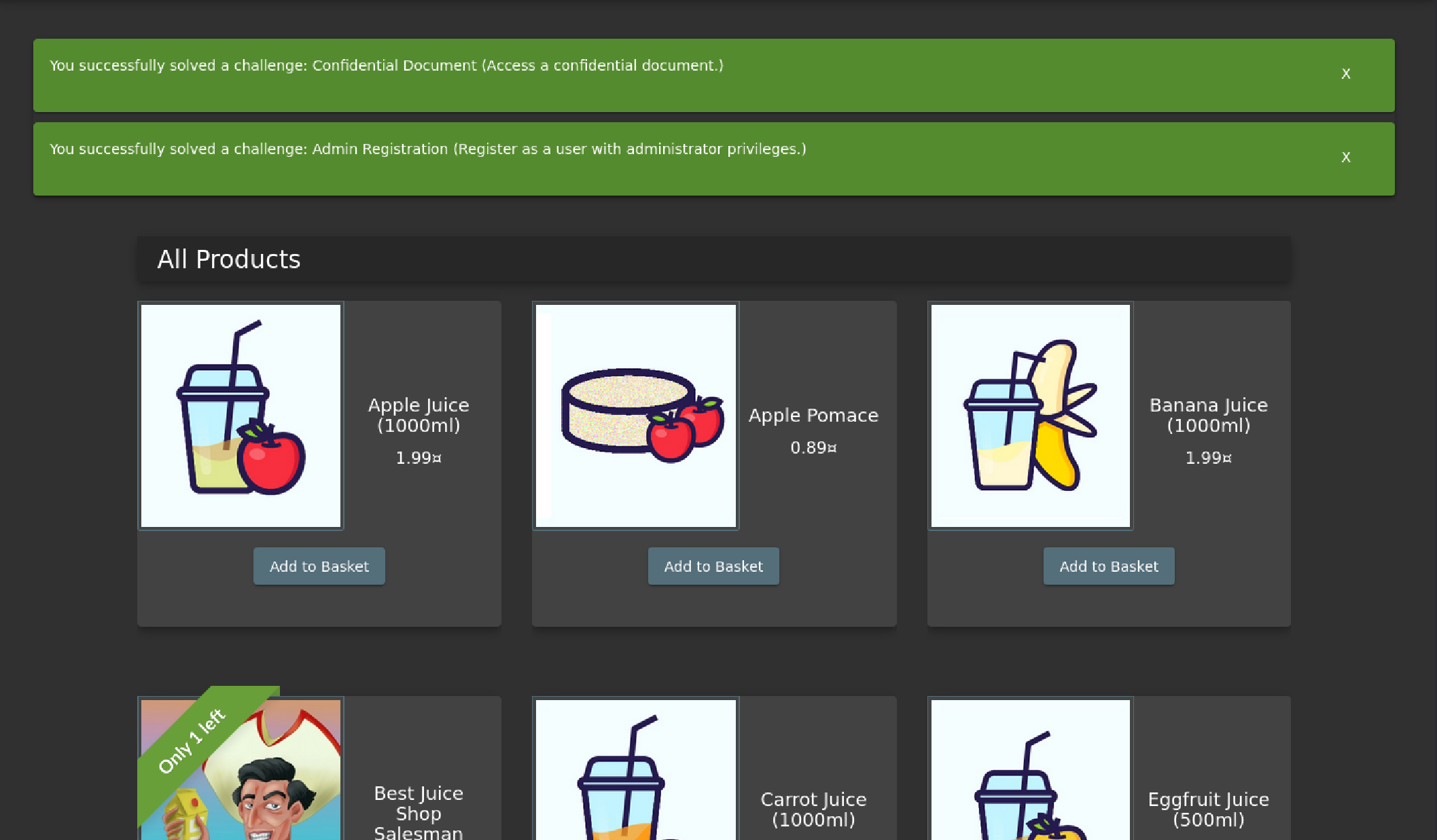
Requirement 7. Exploit an Improper Input Validation vulnerability.

Admin Registration

Register as a user with administrator privileges. Open Burp suite application on kali click next without entering any data then click target where you will click the open browser button.

When the new browser opens type your localhost:3000 to open the owasp web page then create a new account. After creating new account go back to the burp site and under Proxy click intercept is on then under repeat add "role": "admin" to the code which will successfully give it the required administrative role.





Requirement 8. Exploit a Sensitive Data Exposure vulnerability

Confidential Document

Access a confidential document by using this URL <http://192.168.123.1:3000/#/ftp>

