

Vulnerability Name: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

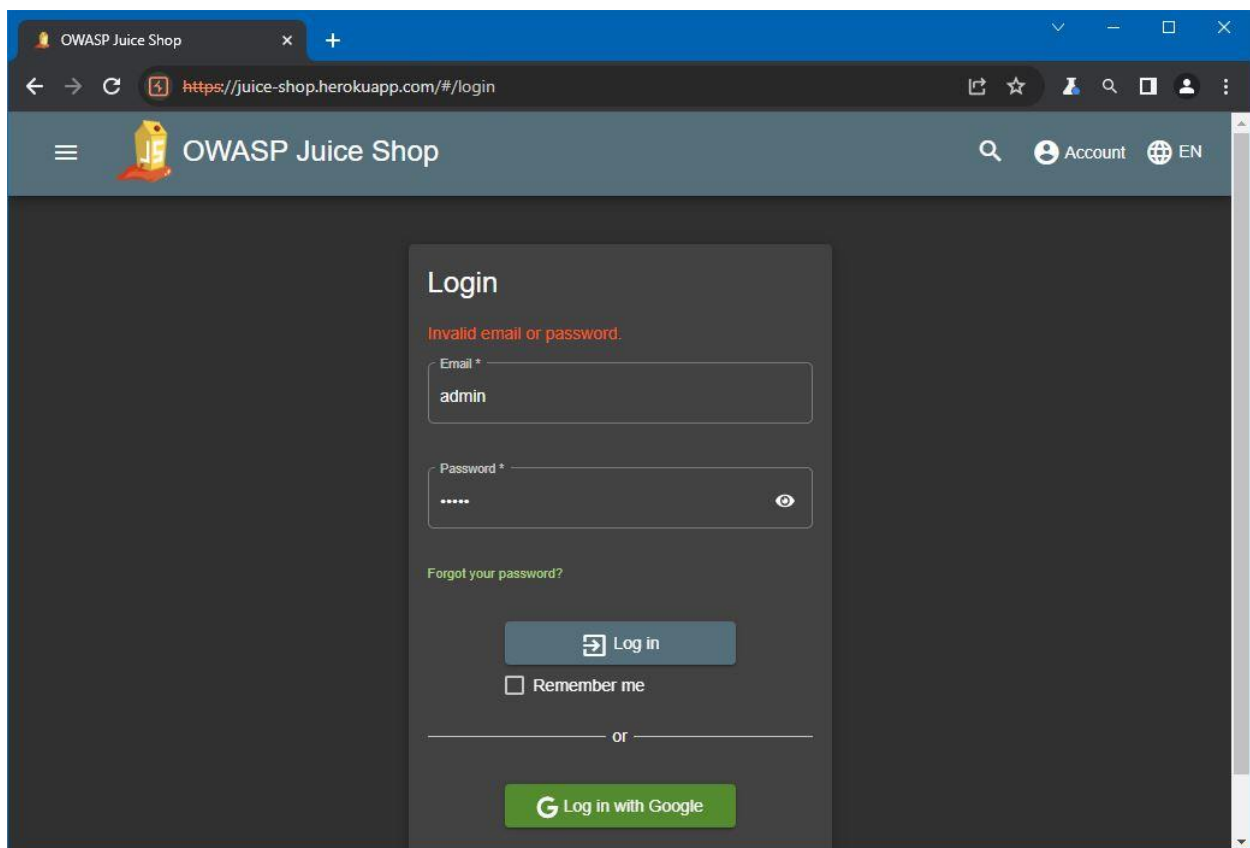
CWE: CWE-89

Description:

CWE-89, titled "Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')," is a common software vulnerability that occurs when an application does not properly validate or sanitize user inputs before including them in SQL queries.

Business Impact:

CWE-89, or SQL injection, poses a severe business impact. This vulnerability can lead to data breaches, causing financial losses, legal repercussions, and reputational damage. Data theft compromises sensitive customer information and intellectual property, eroding trust and potentially triggering costly legal actions



## TEAM 1.1

[illegible]

The screenshot displays the Burp Suite interface with an intercepted HTTP POST request. The main window shows the raw request data, which is a JSON object containing login credentials. The right-hand Inspector panel provides a structured view of the request, including its attributes, query parameters, cookies, and headers.

**Request Attributes:**

Name	Value
Method	POST
Path	/rest/user/login

**Request query parameters:** It's empty in here

**Request cookies:**

Name	Value
language	en
welcomebanner_status	dismiss
cookieconsent_status	dismiss

**Request headers:** 17

Vulnerability Name: Improper input validation

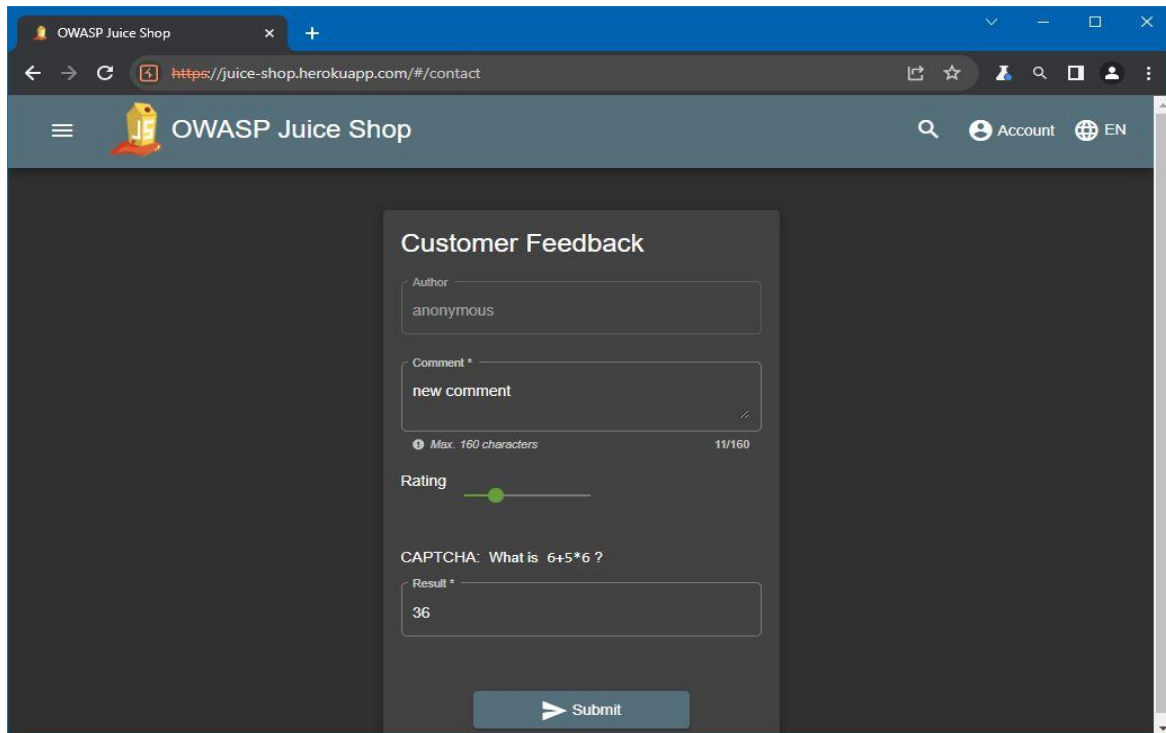
CWE: CWE 20

Description:

CWE-20, also known as "Improper Input Validation," is a software weakness that occurs when a program does not adequately validate and sanitize user inputs. This can lead to security vulnerabilities such as buffer overflows, SQL injection, and cross-site scripting.

Business Impact:

CWE-20 can have a significant business impact, including data breaches, financial losses, reputation damage, and legal liabilities. Vulnerabilities arising from improper input validation can allow attackers to exploit software, leading to unauthorized access, data theft, and service disruptions, potentially resulting in customer trust erosion and costly remediation efforts.

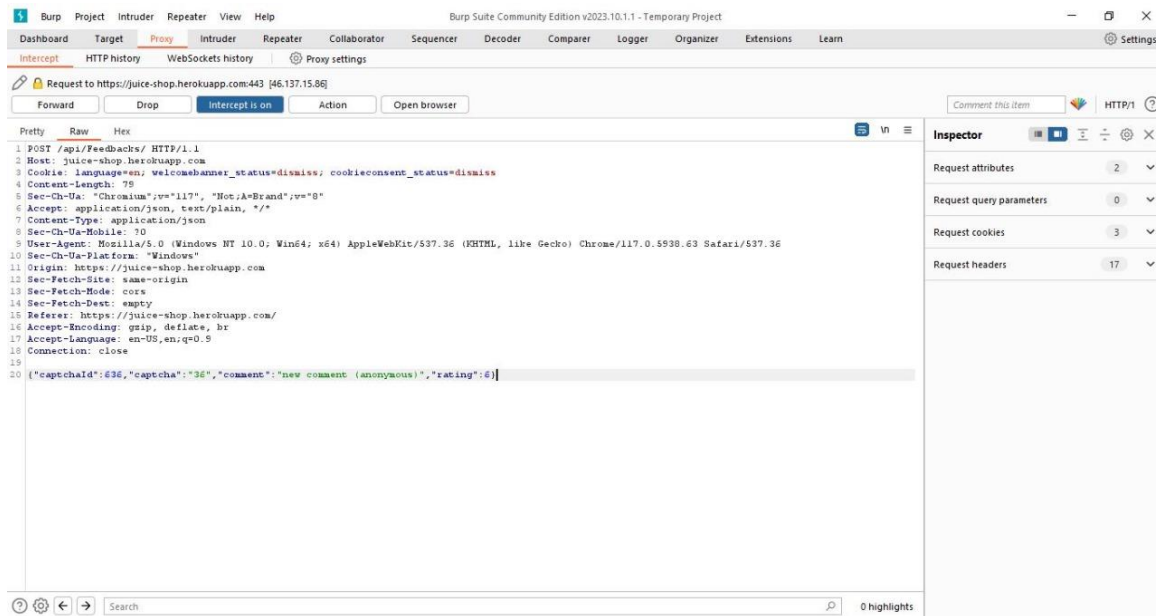
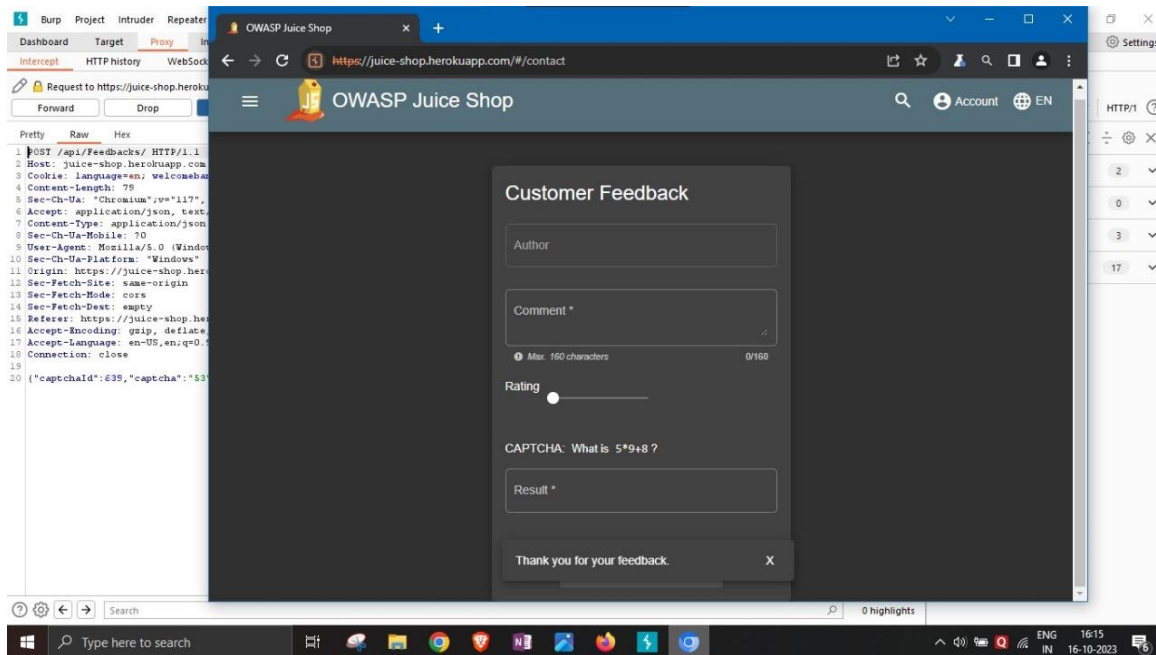


The screenshot shows a web browser window with the URL `https://juice-shop.herokuapp.com/#/contact`. The page title is "OWASP Juice Shop". The main content area displays a "Customer Feedback" form. The form has the following fields and elements:

- Author:** A text input field containing the value "anonymous".
- Comment \*:** A text input field containing the value "new comment". Below the field is a character count: "Max. 160 characters" and "11/160".
- Rating:** A horizontal slider control with a green dot indicating a rating of 1 out of 5.
- CAPTCHA:** A section titled "CAPTCHA: What is  $6+5*6$  ?". It includes a "Result \*" input field containing the value "36".
- Submit:** A button with a right-pointing arrow and the text "Submit".

The form is set against a dark background, and the browser's address bar and navigation icons are visible at the top.

## TEAM 1.1



Vulnerability Name: Sensitive data exposure

CWE: CWE 200

Description:

CWE-200, known as "Exposure of Sensitive Information to an Unauthorized Actor," represents a security weakness where sensitive data, like passwords, encryption keys, or personal information, is improperly disclosed to unauthorized individuals or systems. This vulnerability can lead to serious breaches, compromising privacy and security.

### Business Impact:

CWE-200 can result in severe business consequences, including reputational damage, loss of customer trust, legal consequences, and financial losses. Exposing sensitive information to unauthorized actors can lead to data breaches and regulatory fines, impacting an organization's bottom line and causing long-term damage to its brand and operations.

The screenshot displays the Burp Suite interface with the 'History' tab selected. The top toolbar shows various tools like Burp, Project, Intruder, Repeater, View, Help, Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and a search bar. The main window is divided into three panes: Request, Response, and Inspector.

**Request Pane:** Shows a list of HTTP history items. The selected item is a GET request to `https://juice-shop.herokuapp.com` with a status of 200 OK. The request details are visible in the right pane.

**Response Pane:** Displays the response body, which is a 200 OK status with a 'Content-Type: text/html' header. The response content is a placeholder for a document, indicating that the document is confidential and should not be distributed.

**Inspector Pane:** Shows the request and response headers, cookies, and other metadata. The request headers include 'Host: juice-shop.herokuapp.com', 'Cookie: language=en; welcome\_banner\_status=dismiss; cookieconsent\_status=dismiss', and 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5930.63 Safari/537.36'.

**Target Pane:** Shows the target URL, `https://juice-shop.herokuapp.com`, and a list of other targets in the history.

**Filter:** A filter is applied to the history list, showing only items with a status of 200 OK.

# TEAM 1.1

Burp Suite Community Edition v2023.10.1.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings

Site map Issue definitions Scope settings

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time requ...
https://juice-shop.h...	GET	/socket.io/?EIO=4&...		✓ 101	145				16:27:35 16...
https://juice-shop.h...	GET	/socket.io/?EIO=4&...		✓ 101	145				16:29:32 16...
https://juice-shop.h...	GET	/		✓ 200	4723	HTML	OWASP Juice Shop		16:28:56 16...
https://juice-shop.h...	GET	/api/challenges/na...		200	1542	JSON			16:28:34 16...
https://juice-shop.h...	GET	/api/feedbacks/		200	9078	JSON			16:29:26 16...
https://juice-shop.h...	GET	/api/quantities/		200	6888	JSON			16:12:22 16...
https://juice-shop.h...	GET	/assets/i18n/en.js...		200	20728	JSON			16:28:09 16...
https://juice-shop.h...	GET	/favicon.ico		200	4218	HTML	OWASP Juice Shop		16:28:31 16...
https://juice-shop.h...	GET	/ftp/legal.md		200	3521	text			16:28:30 16...
https://juice-shop.h...	GET	/main.js		200	492446	script			16:28:58 16...

Request

1 GET /ftp/legal.md HTTP/1.1

2 Host: juice-shop.herokuapp.com

3 Cookie: language=en; welcomeBanner\_status=dismiss; cookieconsent\_status=dismiss

4 Sec-CH-UA: "Chromium",v="117", "Not\_A=Brand",v="0"

5 Sec-CH-UA-Mobile: ?0

6 Sec-CH-UA-Platform: "Windows"

7 Upgrade-Insecure-Requests: 1

8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.5938.63 Safari/537.36

9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7

10 Sec-Fetch-Site: same-origin

11 Sec-Fetch-Mode: navigate

12 Sec-Fetch-User: ?1

13 Sec-Fetch-Dest: document

14 Referer: https://juice-shop.herokuapp.com/

15 Accept-Encoding: gzip, deflate, br

16 Accept-Language: en-US,en;q=0.9

17 Connection: close

Response

1 HTTP/1.1 200 OK

2 Server: Cowboy

3 Connection: close

4 Access-Control-Allow-Origin: \*

5 X-Content-Type-Options: nosniff

6 X-Frame-Options: SAMEORIGIN

7 Feature-Policy: payment 'self'

8 X-Recruiting: /#/jobs

9 Accept-Ranges: bytes

10 Cache-Control: public, max-age=0

11 Last-Modified: Sun, 15 Oct 2023 20:23:28 GMT

12 Etag: W/"b47-1b33502b77"

13 Content-Type: text/markdown; charset=UTF-8

14 Vary: Accept-Encoding

15 Date: Mon, 16 Oct 2023 10:50:27 GMT

16 Via: 1.1 proxy

17 Content-Length: 3047

18

19 # Legal Information

20

21 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy

22 eirmod tempor invidunt ut labore et dolore magna

Inspector

Request attributes 2

Request cookies 3

Request headers 16

Response headers 16

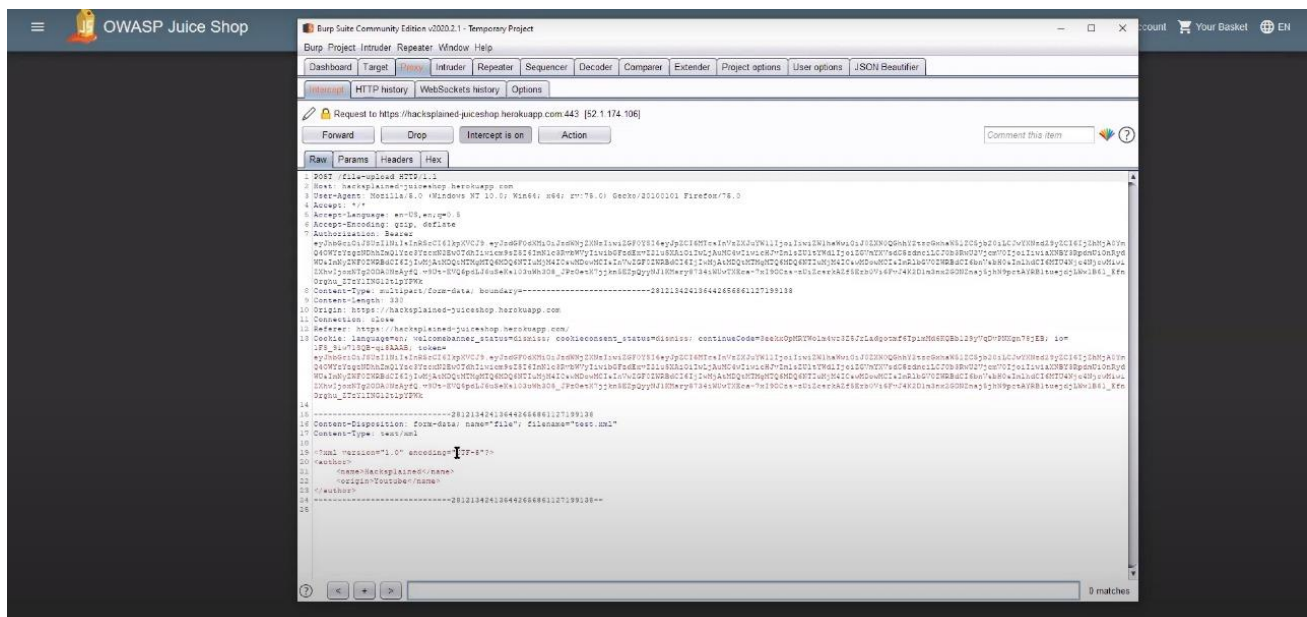
## Vulnerability Name: Security misconfiguration

### Description:

Security misconfiguration vulnerabilities occur when a system, application, or component is improperly set up, leaving it exposed to potential attacks. These weaknesses can lead to unauthorized access, data breaches, or other security incidents due to poorly configured permissions, default settings, or unnecessary features being enabled.

### Business Impact:

Security misconfigurations can have significant business impacts, including data breaches, downtime, regulatory fines, and reputational damage. Improperly configured systems or applications can lead to unauthorized access, data exposure, and service disruptions. These incidents can result in financial losses, eroded customer trust, and legal consequences, affecting an organization's bottom line and market standing. Proper configuration management is essential to mitigate these risks.





## TEAM 1.1

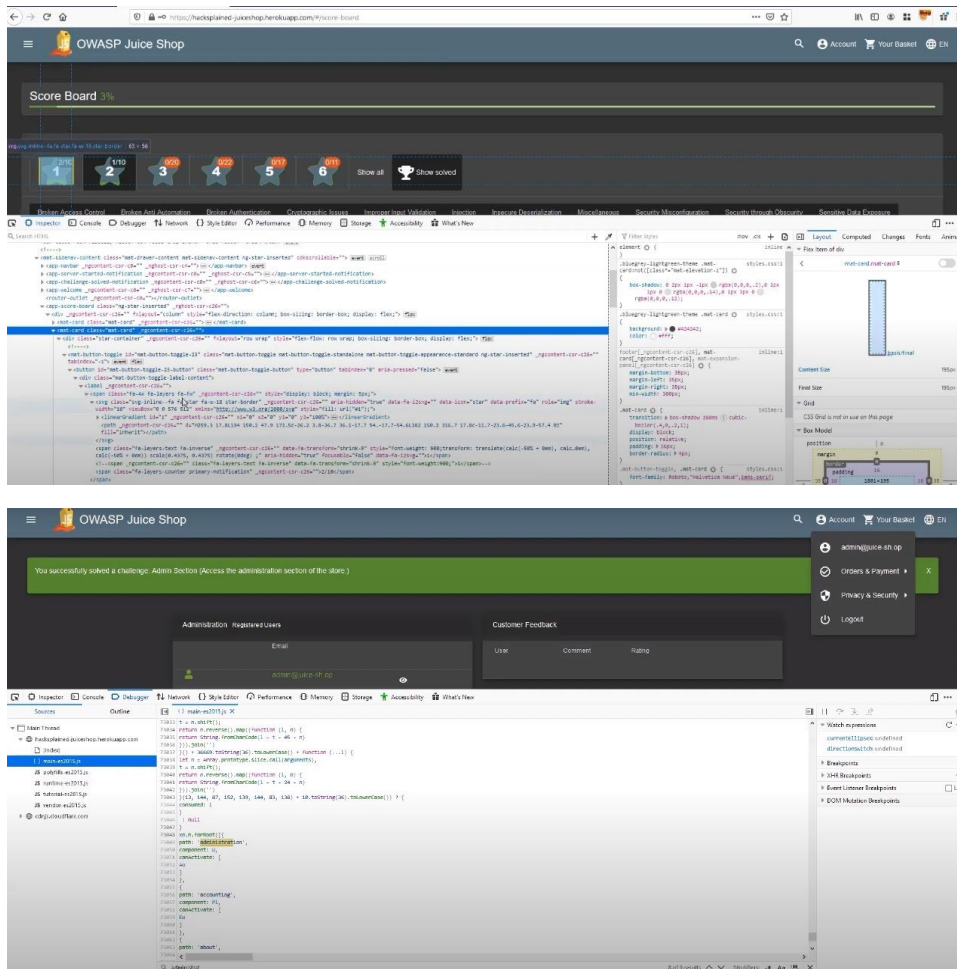
### Vulnerability Name: Broken access control

#### Description:

Broken access control is a security vulnerability that occurs when an application or system fails to enforce proper access restrictions. It allows unauthorized users to access sensitive data, perform actions, or modify resources they should not have permission to access. This issue can lead to unauthorized data exposure, data tampering, and pose significant security risks if not mitigated effectively through access control mechanisms.

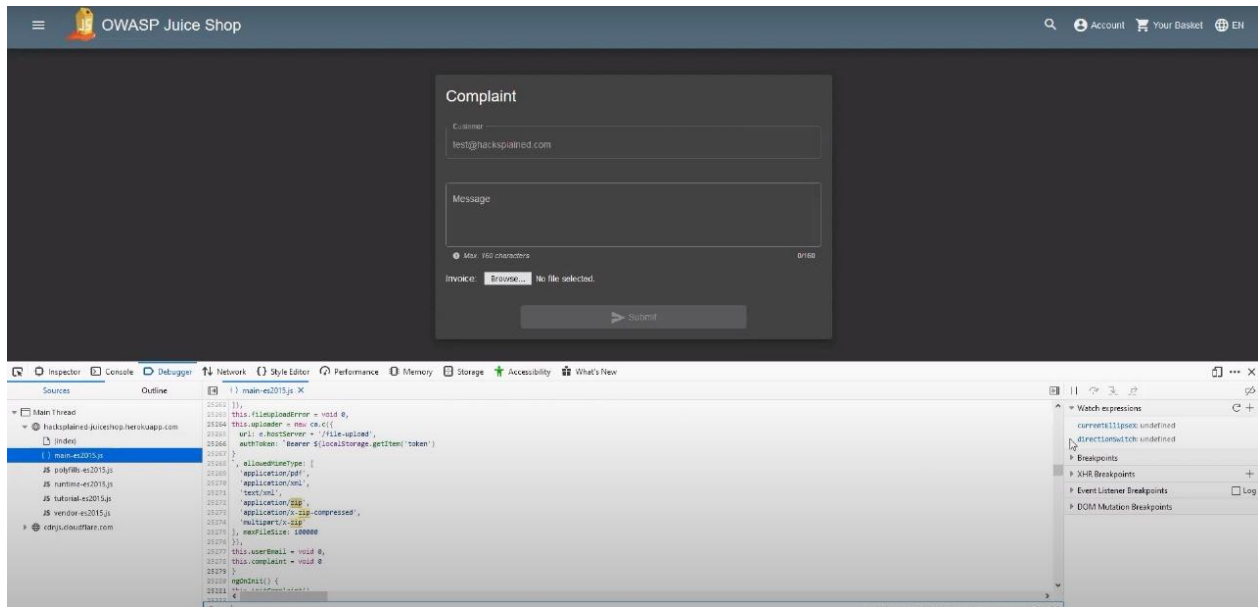
#### Business Impact:

Broken access control can have serious business impacts, including data breaches, compromised privacy, regulatory fines, and damage to reputation. Unauthorized users gaining access to sensitive data or functionality can lead to information theft, legal liabilities, and a loss of customer trust. This can result in financial losses, costs associated with investigations, and remediation efforts, ultimately affecting the organization's financial stability and brand image.





## TEAM 1.1



Vulnerability Name: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

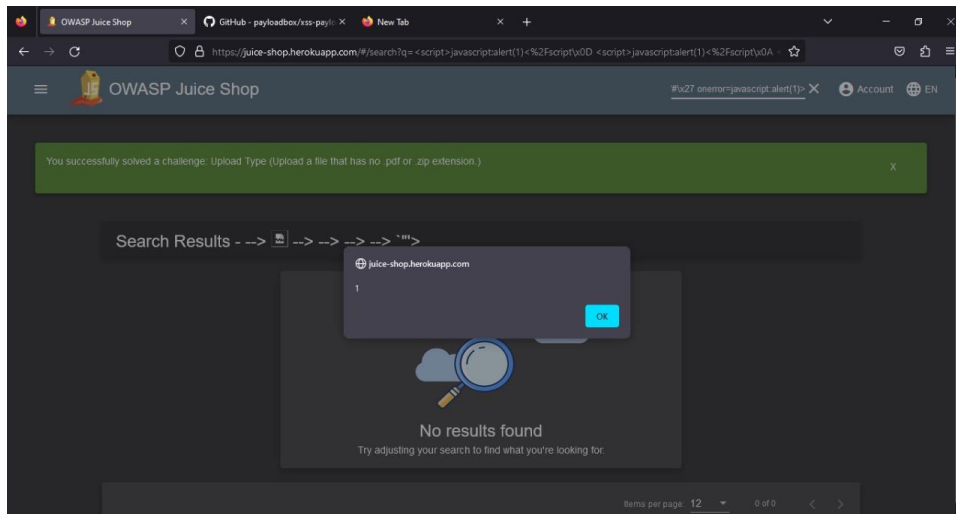
CWE: CWE 79

Description:

CWE-79, also known as "Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')," is a security vulnerability where an application includes untrusted data in web pages without proper validation. Attackers can inject malicious scripts, enabling them to steal data or perform actions on behalf of users, compromising their security and privacy.

Business Impact:

CWE-79 can have significant business impact, including reputation damage, data breaches, and financial losses. Cross-site scripting vulnerabilities allow attackers to steal sensitive data, compromise user accounts, and deface websites, eroding customer trust and potentially leading to regulatory fines. Organizations may also incur costs for incident response, legal liabilities, and remediation, affecting their bottom line and market standing.



## TEAM 1.1

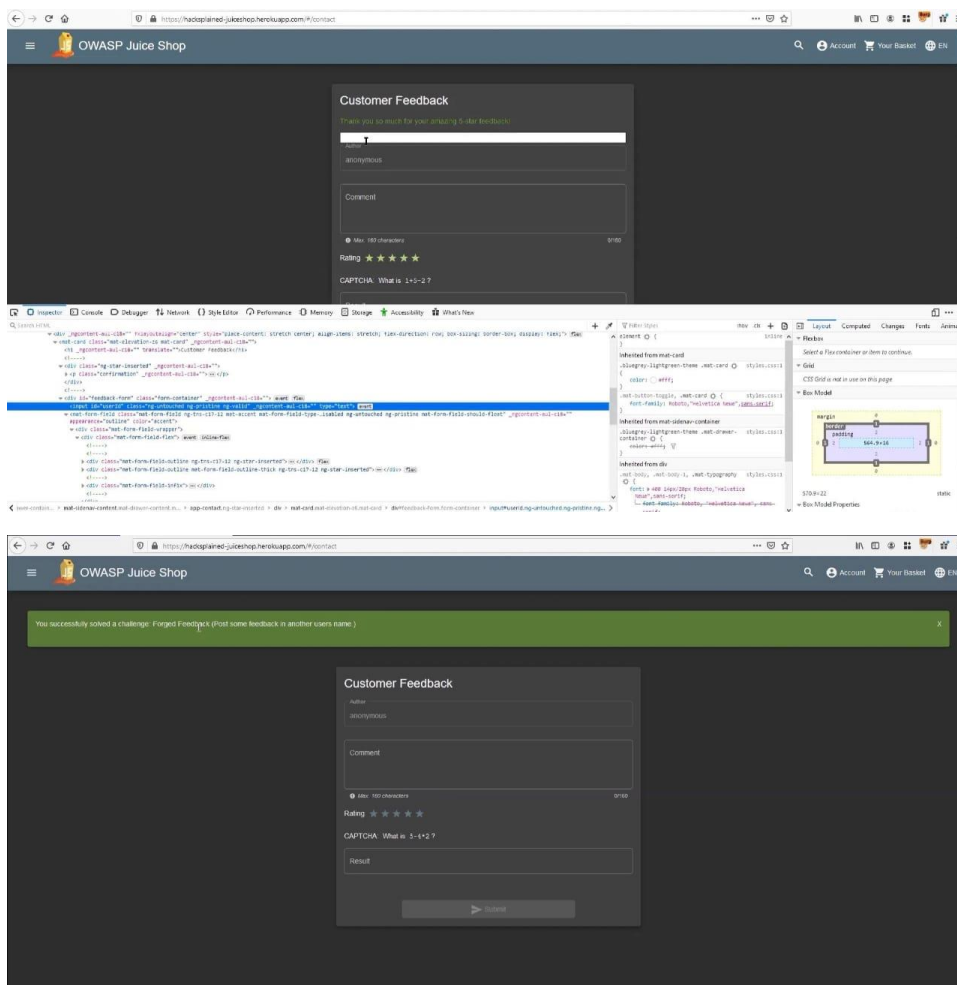
### Vulnerability Name: Forged Feedback

#### Description:

Forged feedback is a security vulnerability where attackers manipulate or counterfeit feedback or responses from a system to deceive users or systems. This can lead to misinformation, trust erosion, and potentially security breaches when users make decisions or take actions based on the fraudulent feedback provided.

#### Business Impact:

Forged feedback vulnerabilities can have a significant business impact, including damage to an organization's reputation, financial losses, and a potential loss of customer trust. Attackers exploiting these weaknesses can deceive users and may lead to actions that compromise security, result in data breaches, or tarnish an organization's image. This can result in costs for incident response, legal liabilities, and long-term harm to the organization's market standing.



Vulnerability Name: Broken Authentication

Description:

Broken authentication is a security issue where flawed or weak authentication and session management in an application enable unauthorized users to gain access to accounts and data. This can lead to data breaches, reputation damage, legal consequences, and financial losses.

Business Impact:

The business impact of broken authentication is significant. It can result in unauthorized access to user accounts and data, leading to data breaches and potentially severe financial losses. It also risks damaging a company's reputation and trust among its customers. Legal consequences and non-compliance with data protection regulations may further compound the impact, resulting in fines and legal actions.

TEAM 1.1

Login

Email

Password

Forgot your password?

Log in

☐ Remember me

or

Log in with Google

Not yet a customer?

OWASP Juice Shop

You successfully solved a challenge: Login Bender (Log in with Bender's user account.)

All Products

Apple Juice (1000ml)  
1.99€

Add to Basket

Apple Pomace  
0.89€

Add to Basket

Banana Juice (1000ml)  
1.99€

Add to Basket

Carrot Juice

Add to Basket

Eggfruit Juice

Add to Basket

Green Juice

Add to Basket

Account

Privacy Policy

Request Data Export

Request Data Erasure

Change Password

2FA Configuration

Last Login IP

Orders & Payment

Privacy & Security

Logout

Project: OWASP Juice Shop

Target: localhost:3000

Intercept: HTTP history

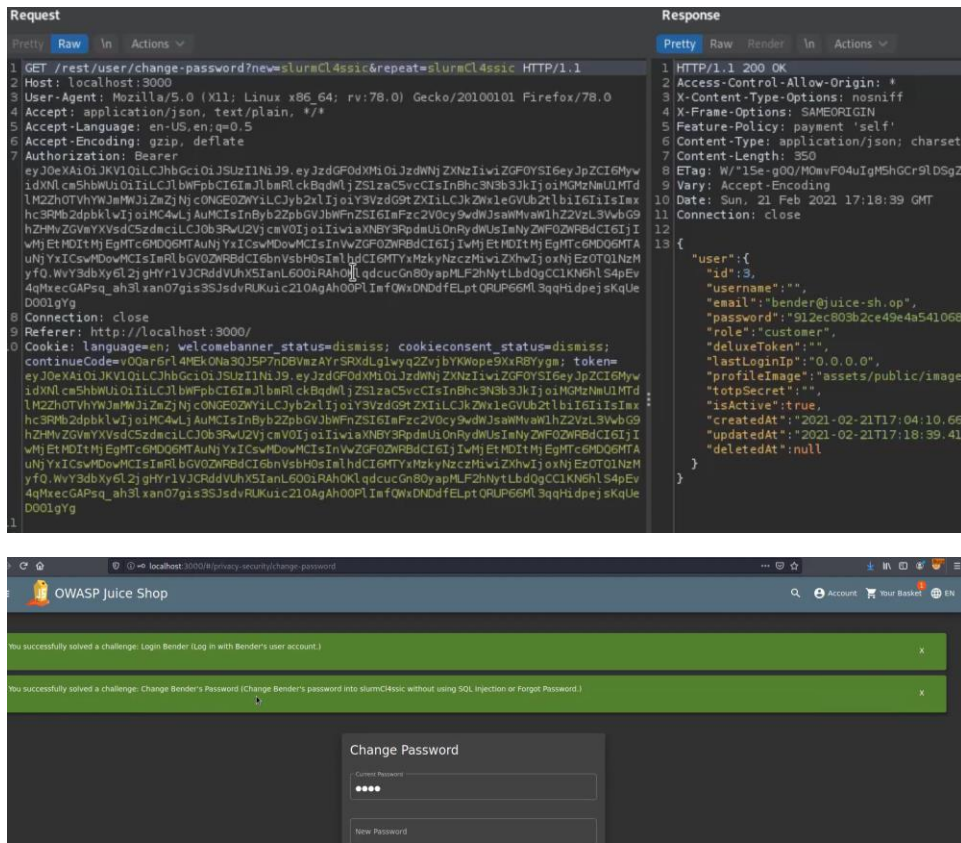
Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
396	http://localhost:3000	GET	/rest/user/change-password?current=a...	✓		401	368	text		
383	http://localhost:3000	GET	/rest/continue-code			200	412	JSON		
382	http://localhost:3000	GET	/rest/products/search?q=	✓		304	255			
381	http://localhost:3000	GET	/api/Quantity/			304	285			
380	http://localhost:3000	GET	/rest/user/whoami			200	462	JSON		
379	http://localhost:3000	GET	/rest/user/whoami			200	462	JSON		
378	http://localhost:3000	GET	/rest/basket/3			200	892	JSON		
377	http://localhost:3000	POST	/rest/user/login	✓		200	1166	JSON		
376	http://localhost:3000	GET	/rest/user/whoami			200	343	JSON		
375	http://localhost:3000	GET	/rest/user/whoami			200	343	JSON		
374	http://localhost:3000	GET	/rest/admin/application-config/			304	255			

Request

Response

## TEAM 1.1



Vulnerability Name: Captcha Bypass(Broken Anti Authentication)

Description:

Captcha Bypass, a type of Broken Authentication, is a security vulnerability where automated scripts or attackers circumvent CAPTCHA challenges designed to prevent unauthorized access. By evading these safeguards, attackers can gain access to protected systems, potentially causing data breaches, fraud, or other malicious activities.

Business Impact:

Captcha Bypass, a form of Broken Authentication, can have serious business consequences. By allowing unauthorized users to evade CAPTCHA challenges, it can lead to unauthorized access, fraud, spam, and account takeovers. This undermines user trust, increases operational costs for dealing with fraudulent

## TEAM 1.1

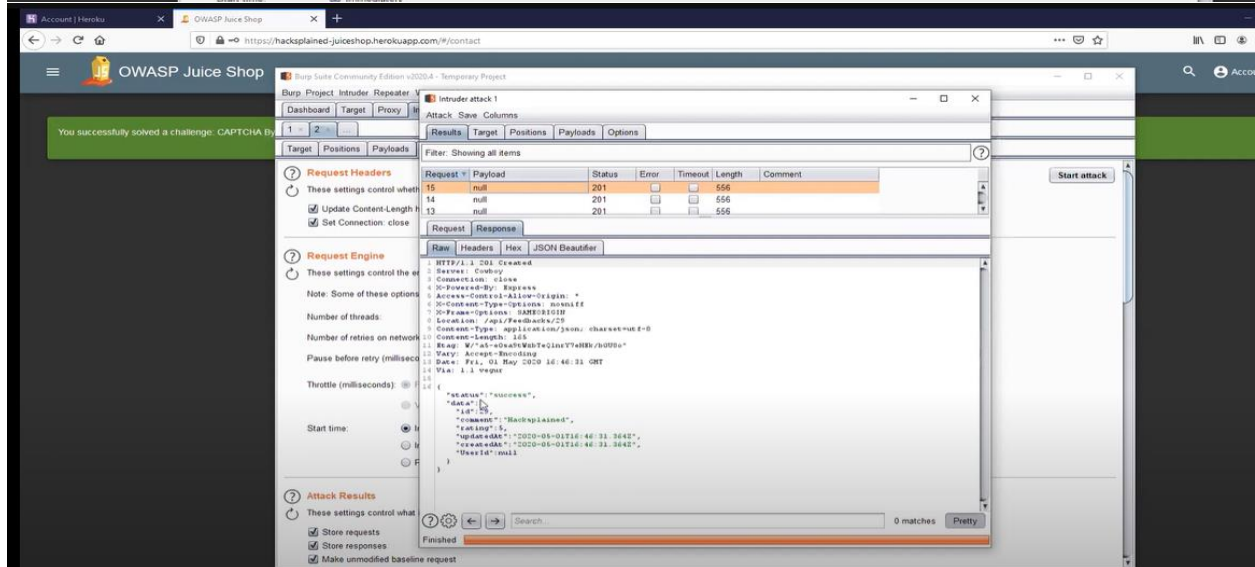
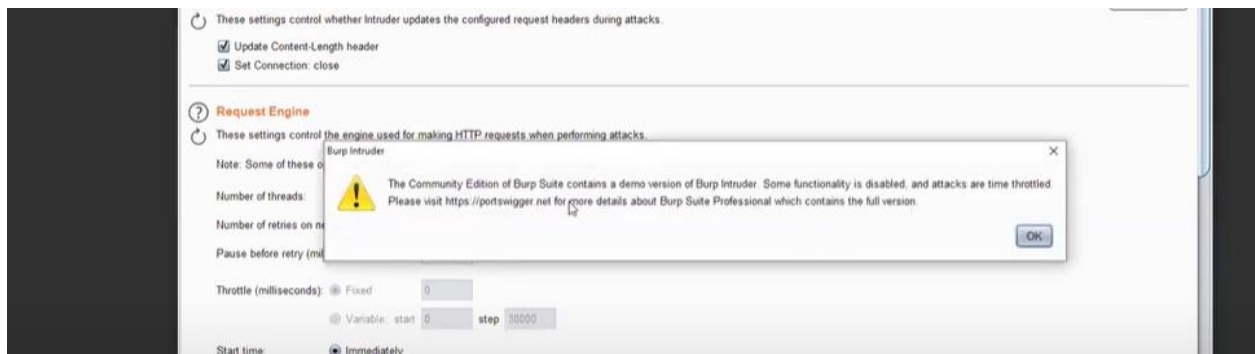
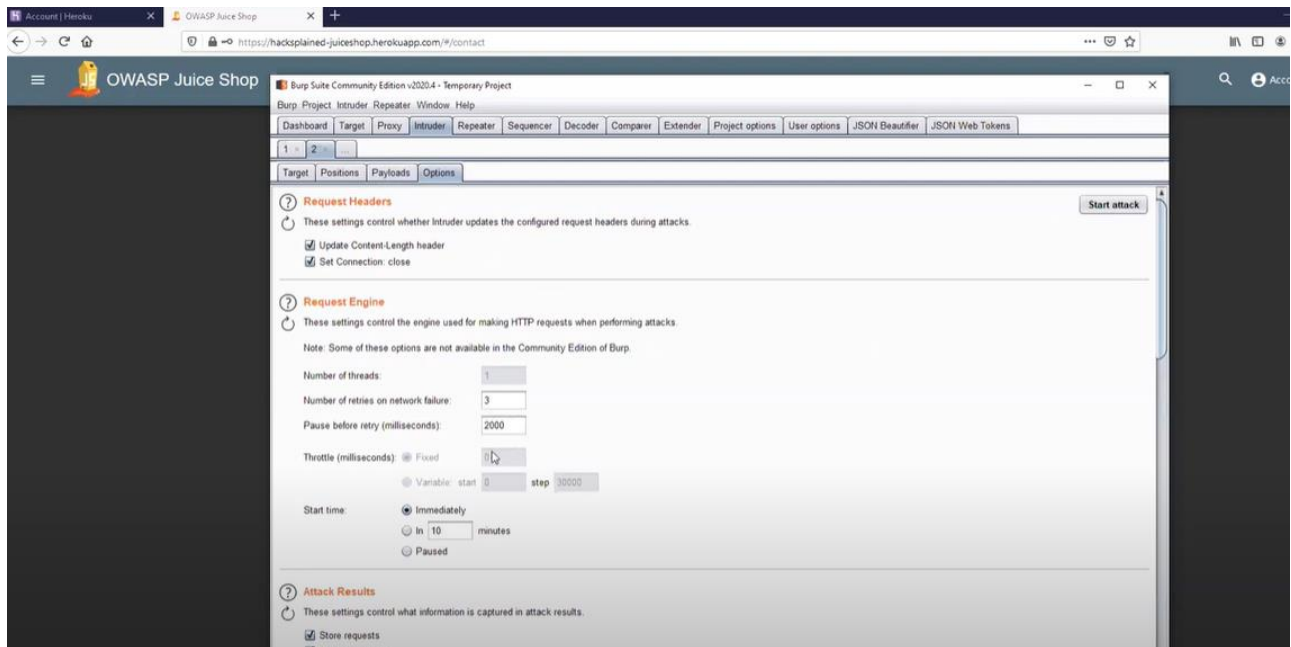
activities, and may result in reputational damage. Companies may also face legal and regulatory issues if they fail to protect user accounts and data adequately.

The screenshot shows the 'Customer Feedback' form in the OWASP Juice Shop application. The form is titled 'Customer Feedback' and has a 'Submit' button. It includes a 'Comment' field with the text 'Hacked!land', a 'Rating' field with five stars, and a 'CAPTCHA' field with the text 'What is 3+2+1?'. The CAPTCHA field contains the answer '-11'. The form is set against a dark background.

The screenshot shows the OWASP Juice Shop application interface. The top navigation bar includes links for 'Home', 'About', 'Contact', 'Privacy', 'Terms', 'Help', and 'Account'. The main content area displays the 'Hacked!land' page, which features a 'Request' and 'Response' section. The 'Request' tab is active, showing a detailed view of the HTTP request, including the method (GET), URL (https://hacked!land.juice-shop.herokuapp.com/), and headers. The 'Response' tab is also visible, showing the response status (200) and headers. The interface is dark-themed and includes a search bar at the bottom.



## TEAM 1.1



Vulnerability Name: Login Admin(Injection)

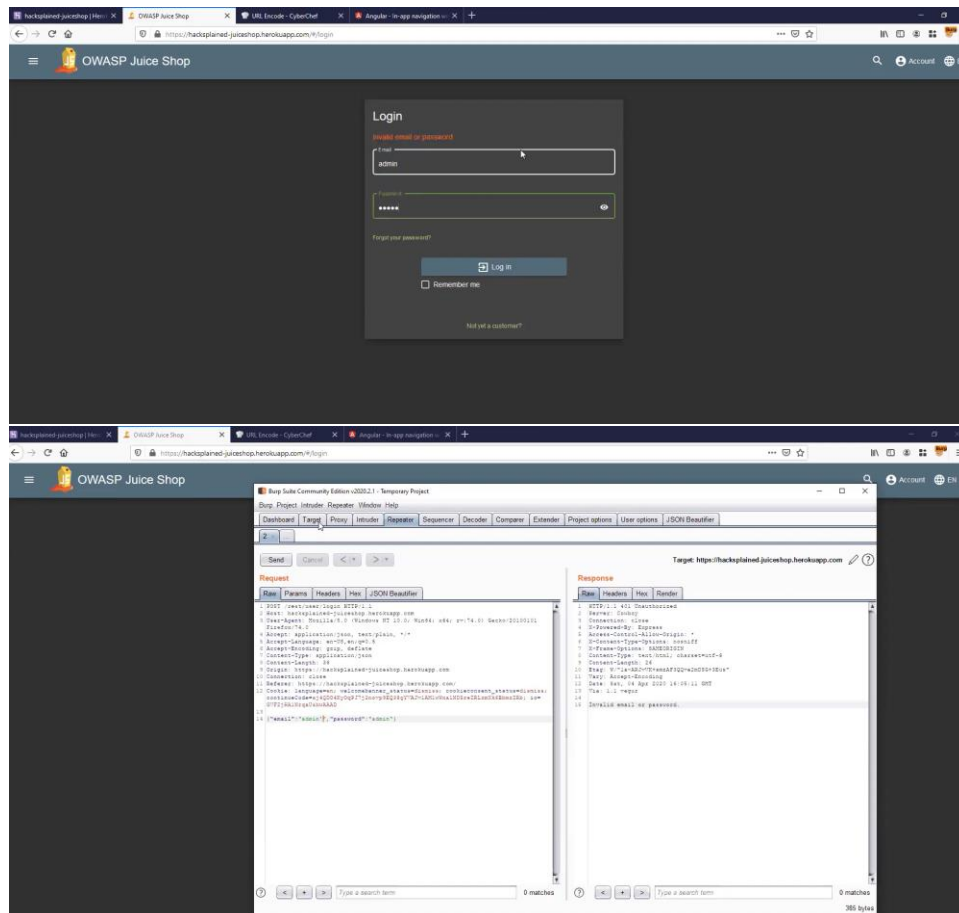
Description:

## TEAM 1.1

A "login admin(injection)" typically refers to a security vulnerability known as SQL injection, where an attacker manipulates input fields to gain unauthorized access to an admin account on a website or application. This technique involves injecting malicious SQL code, potentially leading to data breaches or system compromises.

### Business Impact:

The business impact of a "login admin(injection)" attack can be severe. It can lead to unauthorized access to sensitive data, user accounts, and administrative controls. This can result in data breaches, loss of customer trust, legal repercussions, and financial losses due to legal actions, compliance fines, and costs associated with remediation efforts. It may also damage the company's reputation, affecting future business opportunities and partnerships.



## TEAM 1.1

