**Project Parts:**

**Part 1: Vulnerability Identification and Assessment**

**1. Task:** Conduct a vulnerability assessment on OWASP Juice Shop. Use tools discussed in the course (e.g., OWASP ZAP, Burp Suite) to scan for vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication weaknesses.

This project involves using OWASP Juice Shop, a deliberately insecure web application, to identify and assess vulnerabilities. Tools like OWASP ZAP and Burp Suite are employed to uncover security issues such as SQL injection, XSS, and authentication weaknesses.

### **Using OWASP ZAP to scan for vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication weaknesses.**

OWASP ZAP is an open-source tool designed for scanning and testing web application security.

#### **Steps for Vulnerability Scanning Are Below:**

1. **Setup and Proxy Configuration:**
   * Launch OWASP ZAP and configure your browser to use ZAP as a proxy (usually localhost:8080).
   * Use ZAP’s browser or configure your preferred browser for testing.
2. **Crawling the Application:**
   * Enter the URL of the target application (e.g., OWASP Juice Shop) in the “Quick Start” tab and click **Attack**.
   * ZAP will automatically crawl the website, identifying pages and forms.
3. **Active Scan for Vulnerabilities:**
   * After the crawl, select the target site from the "Sites" tab.
   * Right-click and select **Attack > Active Scan**.
   * OWASP ZAP will perform active tests for vulnerabilities, including SQL injection and XSS.
4. **Reviewing Results:**
   * Go to the “Alerts” tab to view identified vulnerabilities.
   * Click on each alert to see details like the affected URL, the vulnerability type, and potential remediation steps.
5. **Authentication Weakness Testing:**
   * Use the "Forced Browse" feature to identify unprotected resources.
   * Employ the "Fuzzer" to test login forms with brute force attacks or weak credentials.

### **Using Burp Suite to scan for vulnerabilities such as SQL injection, cross-site scripting (XSS), and authentication weaknesses.**

Burp Suite is a professional-grade tool with advanced features for manual and automated security testing.

#### **Steps for Vulnerability Scanning Are Below:**

1. **Setup and Proxy Configuration:**
   * Launch Burp Suite and configure your browser to route traffic through Burp (default: localhost:8080).
2. **Intercept and Explore Traffic:**
   * Use the browser to interact with the target application while Burp’s **Proxy** tool captures traffic.
   * Identify points of interest such as login forms, search fields, or parameters in requests.
3. **Scanning for Vulnerabilities:**
   * Send requests to the **Scanner** tool by right-clicking intercepted requests in the Proxy tab and selecting **Send to Scanner**.
   * Start the scan, focusing on identified endpoints.
4. **Manual Testing:**
   * Use the **Repeater** tool to craft and modify HTTP requests.
     + For SQL injection: Test inputs like ' OR 1=1-- to observe the application’s response.
     + For XSS: Test inputs like **<iframe src = javascript:alert(document.cookie);>** to check for reflected output.
   * Use the **Intruder** tool to test authentication weaknesses by performing credential stuffing or brute force attacks.

**2. Deliverables:**

**○** A vulnerability report listing each identified issue, including type, impact, and affected areas.

○ Screenshots or brief notes explaining how each vulnerability was identified.

**Vulnerability Report for OWASP Juice Shop are as follows:**

This report outlines vulnerabilities identified during a security assessment of OWASP Juice Shop using **OWASP ZAP** and **Burp Suite**. Each issue includes details about its type, impact, affected areas, and how it was identified.

### **1. SQL Injection**

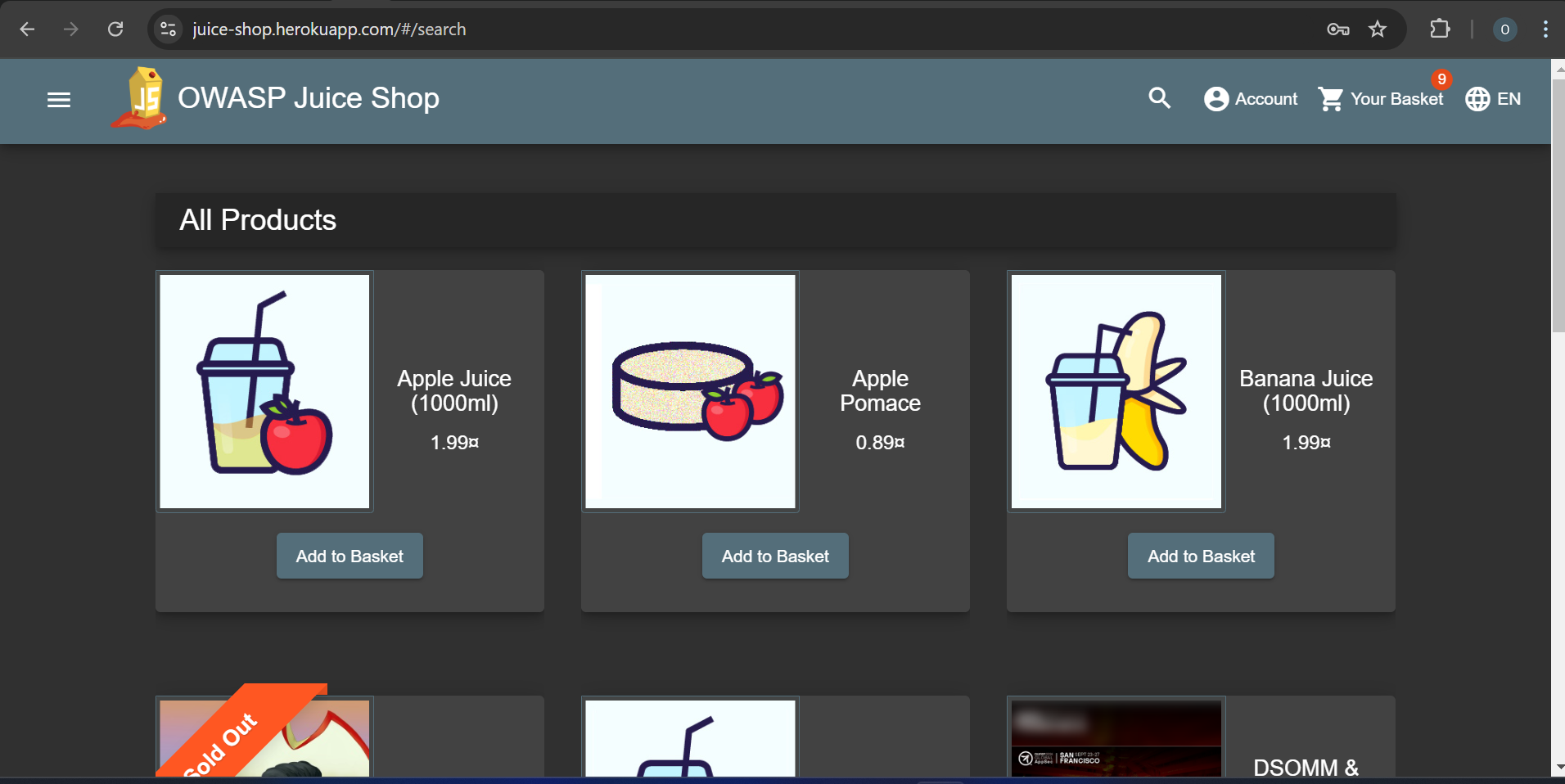
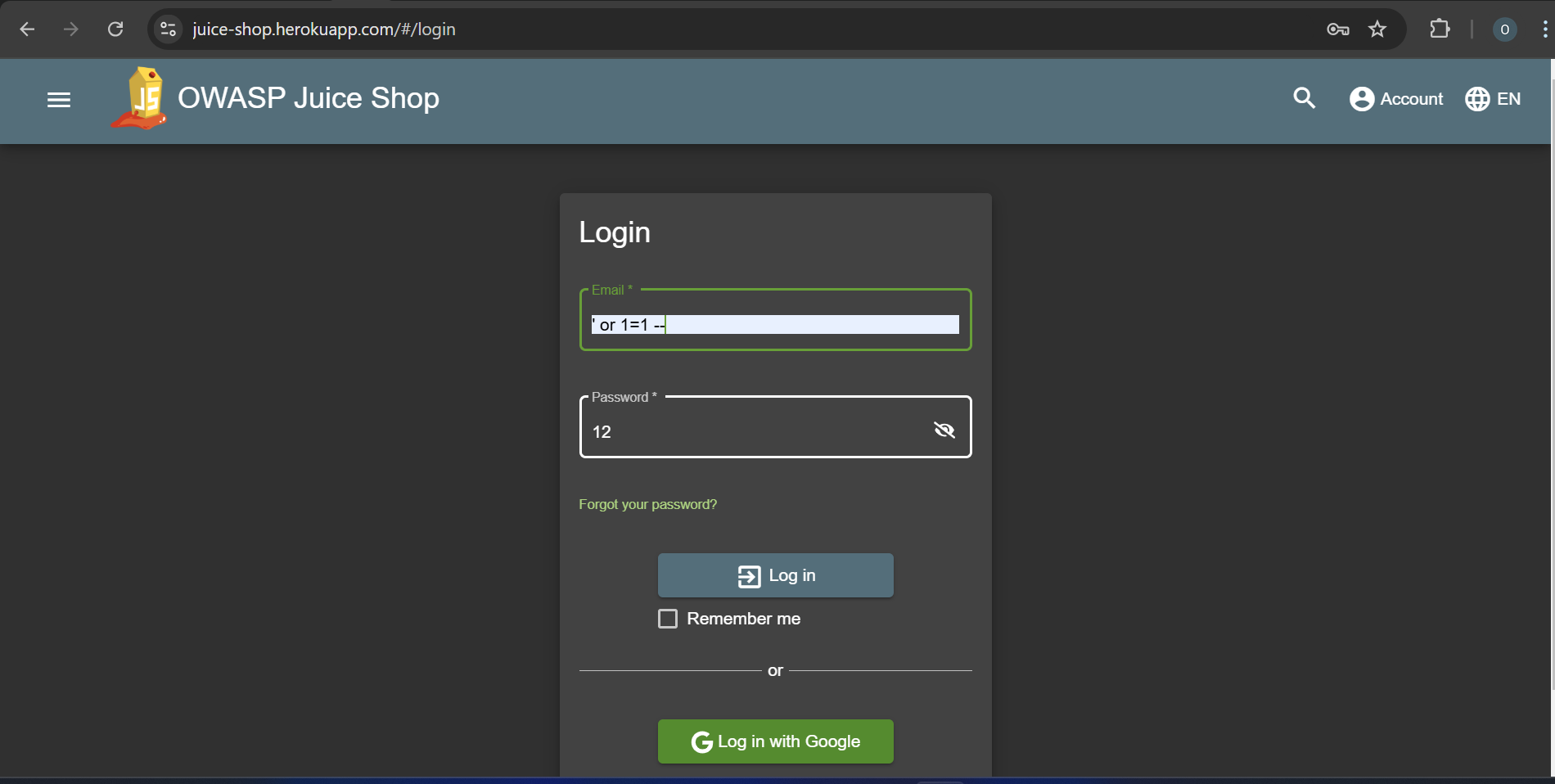
**Impact:** High

Exploiting this vulnerability allows attackers to manipulate backend database queries, potentially leading to data theft, unauthorized data modification, or full database compromise.

**Type:** SQL Injection  
**Affected Areas:**

* Login page: The **email** and **password** fields are vulnerable to SQL injection.

**Identified Issue:**

* Using **Burp Suite Repeater**, a payload like ' OR '1'='1 was submitted in the login field.
* The server responded with a valid session token, confirming that the input was not properly sanitized.

### **2. Cross-Site Scripting (XSS)**

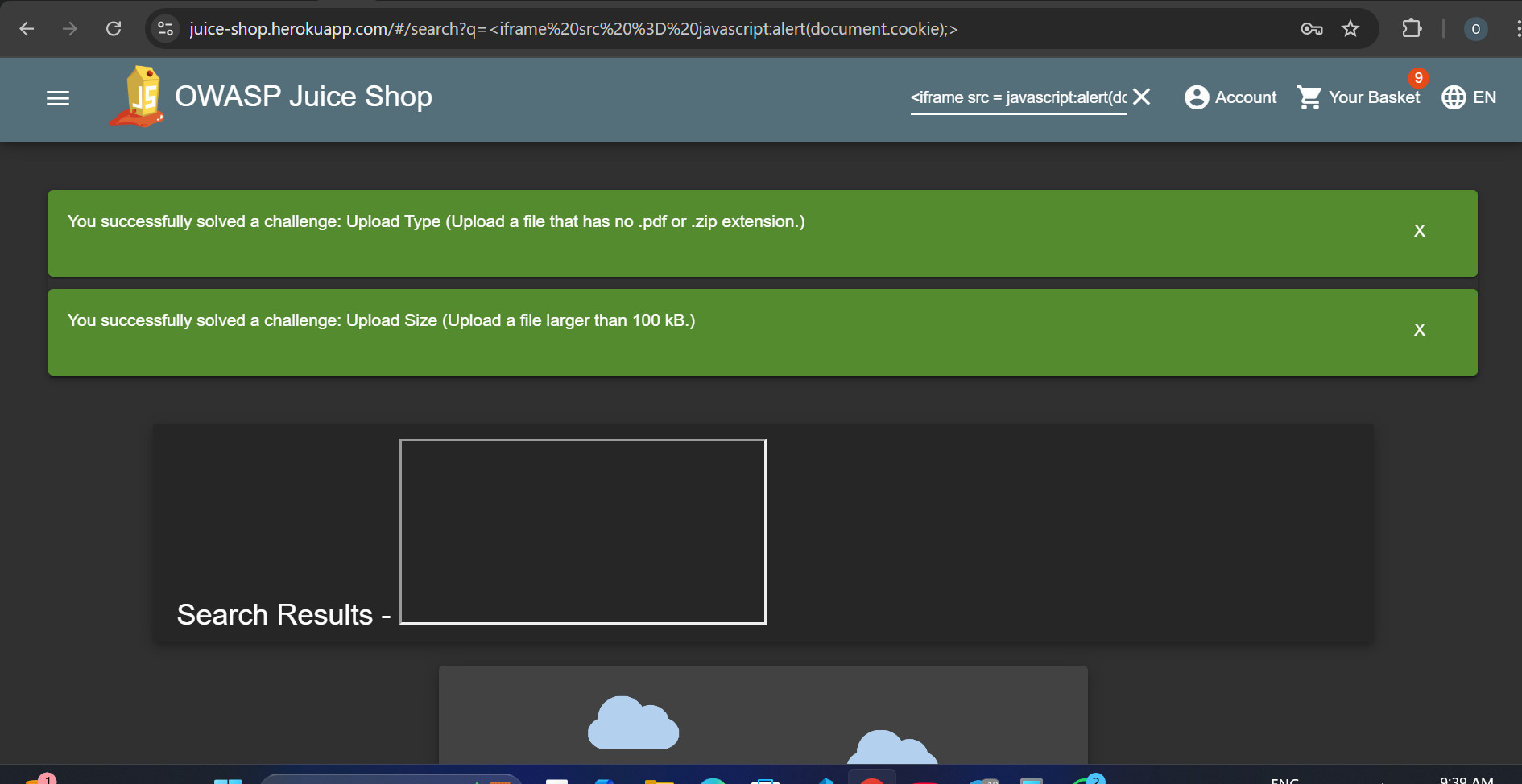
**Type:** Reflected XSS  
**Impact:** Medium

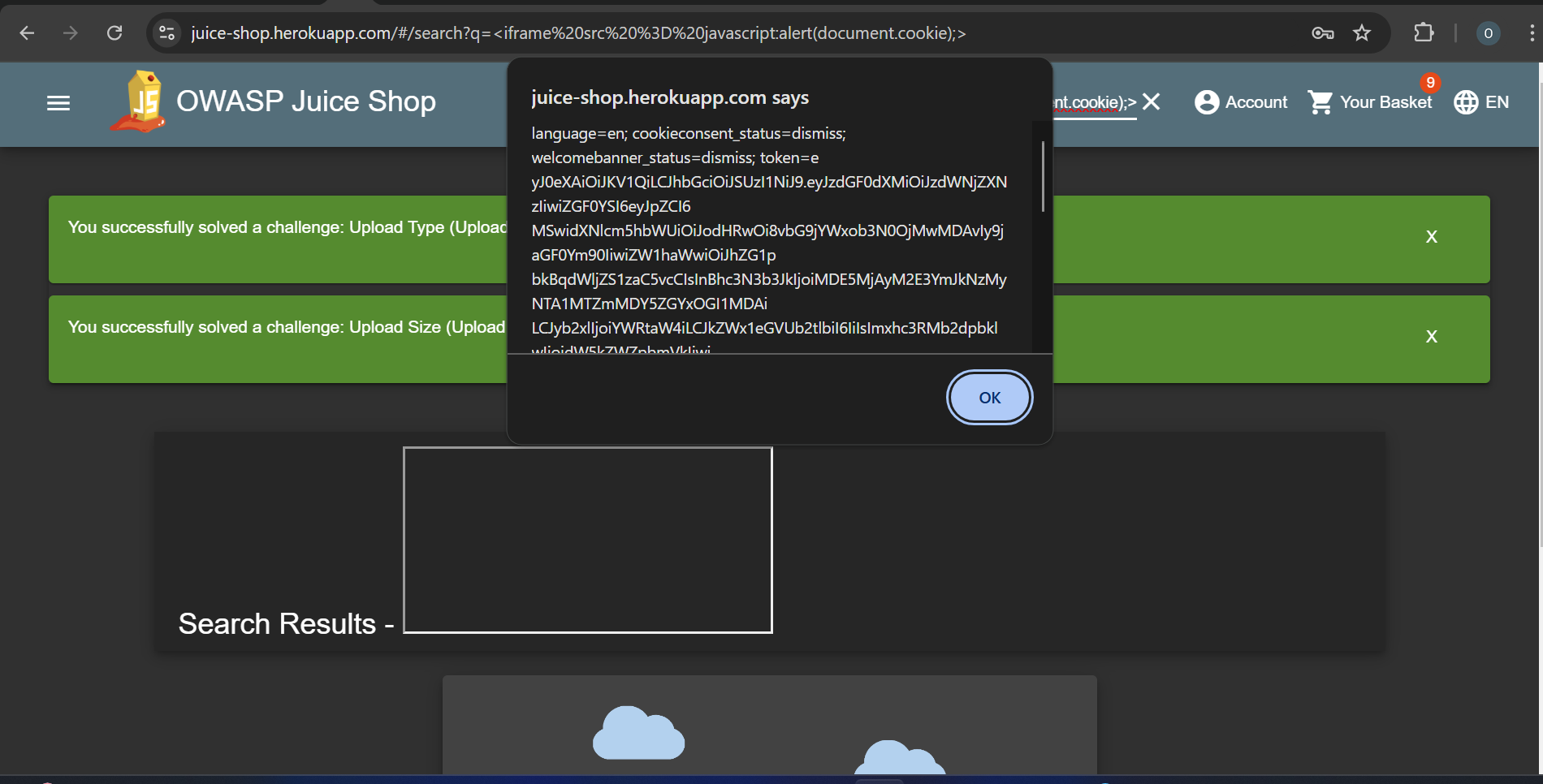
* This vulnerability allows an attacker to inject malicious scripts into the application, potentially stealing session cookies, redirecting users, or performing malicious actions on behalf of the victim.

**Affected Areas:**

* Search functionality: The **q** parameter in the search bar.

**Identified Issue:**

* Using **OWASP ZAP Active Scan**, the scanner flagged the **q** parameter as injectable.
* Manually confirmed by submitting **<iframe src = javascript:alert(document.cookie);>** in the search field. The alert box executed, confirming the vulnerability.



### **3. Authentication Weakness**

**Type:** Weak Password Policy  
**Impact:** High

* The application allows the use of weak passwords, increasing the likelihood of brute force or credential stuffing attacks.

**Affected Areas:**

* Registration page and password reset functionality.

**Identified Issues:**

* Using **Burp Suite Intruder**, a dictionary attack was performed on the login page. Common passwords like **12** and **password** were accepted.
* OWASP ZAP flagged missing complexity requirements for the password field during its active scan.

**3. Skills Applied:**

**○ Threat identification and vulnerability scanning.**

#### **1. Threat Identification and Vulnerability Scanning Are Below:**

* **Techniques Used:**
  + Conducted automated scans using **OWASP ZAP** to identify vulnerabilities like SQL injection and XSS.
  + Employed **Burp Suite** for manual testing, including intercepting traffic, fuzzing inputs, and crafting custom payloads.
* **Key Accomplishments:**
  + Successfully identified SQL injection vulnerabilities by analyzing server responses to malicious inputs.
  + Detected cross-site scripting (XSS) issues through crafted payloads in user input fields.
  + Highlighted authentication weaknesses using brute force attacks and weak credential detection.
* **Tools Mastered:**
  + OWASP ZAP’s automated scanning capabilities and manual tools like the Fuzzer and Spider.
  + Burp Suite’s Repeater for crafting specific requests and Intruder for brute force testing.

**○ Assessment of risks based on potential impact.**

#### **2. Assessment of Risks Based on Potential Impact Are Below:**

* **Techniques Used:**
  + Categorized each vulnerability by type, affected areas, and its potential impact on the application.
  + Evaluated risks using a combination of **likelihood of exploitation** and **severity of impact.**
* **Key Accomplishments:**
  + Assessed SQL injection as a **High-Risk Vulnerability** due to its ability to compromise the database entirely.
  + Rated XSS as **Medium-Risk**, considering its potential to hijack user sessions or deliver malicious payloads.
  + Identified weak password policies as a **High-Risk** issue because of the increased likelihood of brute force attacks.
* **Risk Framework Utilized:**
  + Applied qualitative measures (High, Medium, Low)

**Part 2: Secure Network Design**

**1.** **Task:** Based on the vulnerabilities identified in Part 1, design a secure network architecture. Consider best practices such as segmentation, firewall implementation, and intrusion prevention to mitigate the identified risks.

**Internet**

**|**

**Router**

**|**

**Perimeter Firewall**

**|**

**+--------------------------+**

**| |**

**DMZ Internal Network**

**| |**

**Web App Server VLAN Segmentation**

**| | |**

**WAF + IPS User VLAN Admin VLAN**

**| |**

**Database Server File Server/Logs**

**|**

**Encryption Layer**

### **Network Components and Security Controls**

#### **1. Perimeter Firewall**

* **Role:** Filters traffic entering or leaving the network.
* **Mitigation:**
  + Prevents external attackers from accessing the internal network directly.
  + Blocks malicious traffic, such as payloads targeting SQL injection vulnerabilities.

#### 

#### 

#### **2. DMZ (Demilitarized Zone)**

* **Role:** Hosts public-facing services like the web application server.
* **Mitigation:**
  + Isolates the application server from the internal network.
  + Contains potential breaches, preventing attackers from accessing sensitive internal systems.

#### **3. Web Application Firewall (WAF)**

* **Role:** Inspects HTTP traffic to detect and block malicious payloads targeting application vulnerabilities like SQL injection and XSS.
* **Mitigation:**
  + Filters harmful inputs before they reach the application server.
  + Adds a dedicated layer of protection for the web app server.

#### **4. Intrusion Prevention System (IPS)**

* **Role:** Monitors network traffic for suspicious patterns, such as brute force attempts.
* **Mitigation:**
  + Identifies and blocks attacks in real-time.
  + Reduces risks of exploitation from authentication weaknesses.

#### **5. VLAN Segmentation**

* **Role:** Divides the internal network into logical segments for better control and isolation.
  + **User VLAN:** For general employee access.
  + **Admin VLAN:** Restricted to administrators managing sensitive systems.
* **Mitigation:**
  + Limits lateral movement within the network, containing breaches.
  + Applies least privilege principles, restricting access to critical systems.

#### **6. Database Server**

* **Role:** Stores application data securely, with direct access restricted to the application server.
* **Mitigation:**
  + Enforces encryption at rest and in transit (TLS/SSL).
  + Reduces SQL injection risks by isolating the database server from user access.

#### **7. Encryption Layer**

* **Role:** Ensures data confidentiality for sensitive communications.
* **Mitigation:**
  + Prevents interception of credentials or sensitive information during transmission.

**2. Deliverables:**

**○** A network design proposal document or diagram with detailed explanations for each security control included.

○ Brief descriptions of each network component and how it strengthens security.

**Secure Network Design Proposal Document:**

#### **1. Introduction**

This document outlines a secure network architecture designed to mitigate vulnerabilities identified in OWASP Juice Shop, including SQL injection, cross-site scripting (XSS), and weak authentication. The design integrates industry best practices, such as segmentation, firewalls, and intrusion prevention systems, to ensure robust security and resilience.

#### **2. Objectives**

* Mitigate vulnerabilities that could lead to unauthorized access, data breaches, or system compromise.
* Implement a defense-in-depth strategy with multiple layers of protection.
* Ensure compliance with best practices for secure network architecture.

#### **3. Network Architecture**

**Diagram Description:**The architecture is structured around the following layers:

* **Internet-facing Components:** Includes a perimeter firewall and a DMZ to host public-facing services.
* **Internal Network:** Divided into logical VLANs for users, administrators, and critical servers, with controlled access and monitoring.

#### **4. Key Components and Controls**

**Perimeter Firewall:**

* **Purpose:** Filters incoming and outgoing traffic to block unauthorized access.
* **Justification:** Prevents network reconnaissance and attacks like port scanning and denial-of-service (DoS).

**DMZ (Demilitarized Zone):**

* **Purpose:** Hosts the web application server, isolating it from the internal network.
* **Justification:** Reduces the risk of attackers reaching sensitive internal resources if the web server is compromised.

**Web Application Firewall (WAF):**

* **Purpose:** Inspects HTTP/S traffic for malicious payloads targeting vulnerabilities like SQL injection and XSS.
* **Justification:** Filters harmful inputs and enhances application security.

**Intrusion Prevention System (IPS):**

* **Purpose:** Monitors and blocks suspicious network activity.
* **Justification:** Detects and prevents brute force attacks and exploitation of weak authentication mechanisms.

**VLAN Segmentation:**

* **Purpose:** Creates logical network segments for better access control.
  + User VLAN: For general employee activities.
  + Admin VLAN: For system administrators managing critical resources.
* **Justification:** Limits lateral movement in the network and enforces the principle of least privilege.

**Database Server with Encryption:**

* **Purpose:** Stores data securely, accessible only by the web application server.
* **Justification:** Reduces SQL injection risks and ensures data confidentiality and integrity with encryption protocols (e.g., TLS/SSL).

**Encryption Layer:**

* **Purpose:** Protects sensitive data in transit and at rest.
* **Justification:** Prevents interception of credentials, sessions, and sensitive information.

#### **5. Risk Mitigation Highlights**

* **SQL Injection:** Mitigated by database server isolation, WAF rules, and strict input validation.
* **XSS:** Addressed with a WAF to sanitize input and block malicious scripts.
* **Authentication Weaknesses:** Managed with an IPS to detect brute force attempts and by enforcing strong password policies.

#### **6. Conclusion**

The proposed architecture implements a secure, layered defense to address vulnerabilities identified in OWASP Juice Shop. By leveraging segmentation, proactive monitoring, and robust encryption, this design ensures the application and network are resilient to common threats while adhering to industry best practices.

**3. Skills Applied:**

○ Network security fundamentals and secure design principles.

○ Application of risk mitigation strategies.

#### **1. Network Security Fundamentals and Secure Design Principles Are Below:**

* **Techniques Used:**
  + Applied best practices such as **network segmentation**, **firewalls**, and **DMZs** to isolate critical resources.
  + Integrated tools like **WAFs** and **IPS** for proactive monitoring and protection.
  + Implemented encryption standards to ensure secure data transmission and storage.
* **Key Accomplishments:**
  + Designed a layered defense-in-depth architecture to protect against identified vulnerabilities, including SQL injection, XSS, and weak authentication.
  + Ensured compliance with principles like **least privilege** and **zero trust**, reducing potential attack surfaces.
* **Core Concepts Mastered:**
  + Firewall rule configuration, VLAN segmentation, and DMZ implementation.
  + Encryption protocols (TLS/SSL) for safeguarding sensitive communications.

#### 

#### **2. Application of Risk Mitigation Strategies:**

* **Techniques Used:**
  + Mapped vulnerabilities from Part 1 (e.g., SQL injection) to specific mitigations like database isolation and WAFs.
  + Employed proactive measures, such as IPS and password policies, to reduce the likelihood of successful brute force or credential attacks.
* **Key Accomplishments:**
  + Reduced the impact of vulnerabilities by aligning design choices with **identified risks and their potential impact**.
  + Enhanced the network’s overall resilience by focusing on containment and prevention through segmentation and monitoring.
* **Risk Framework Utilized:**
  + Prioritized mitigations based on the severity of identified vulnerabilities, ensuring high-impact risks (e.g., SQL injection) were addressed first.