# **Udajuicer**:
# Threat Report

**YOUR NAME: Mohammed Alnajrani**

*8/31/2023*

# Purpose of this Report:

This is a threat model report for **Udajuicer**. The report will describe the threats facing Udajuicer. The model will cover the following:

- Threat Assessment
    - Scoping out Asset Inventory
    - Architecture Audit
    - Threat Model Diagram
    - Threats to the Organization
    - Identifying Threat Actors
- Vulnerability Analysis
- Risk Analysis
- Mitigation Plan

# Section 1

## Threat Assessment

# 1.1: Asset Inventory

**Components and Functions**

- ***Web server:*** is used to respond to client requests made over the World Wide Web. And the main job of it is to display website content to users by storing, processing, and delivering data
- ***Application server:*** is used to deliver  a dynamic, customized response to a client's requests
- ***Database:*** is used to create, edit, and maintain database files and records

# 1.2 Architecture Audit

**Flaws**

- *Lack of Demilitarized Zone (DMZ)*
- *A firewall has not been used*
- *Lack of Content Delivery Network (CDN)*
- *Lack of Web Application Firewall (WAF)*
- Load Balancing is not been used

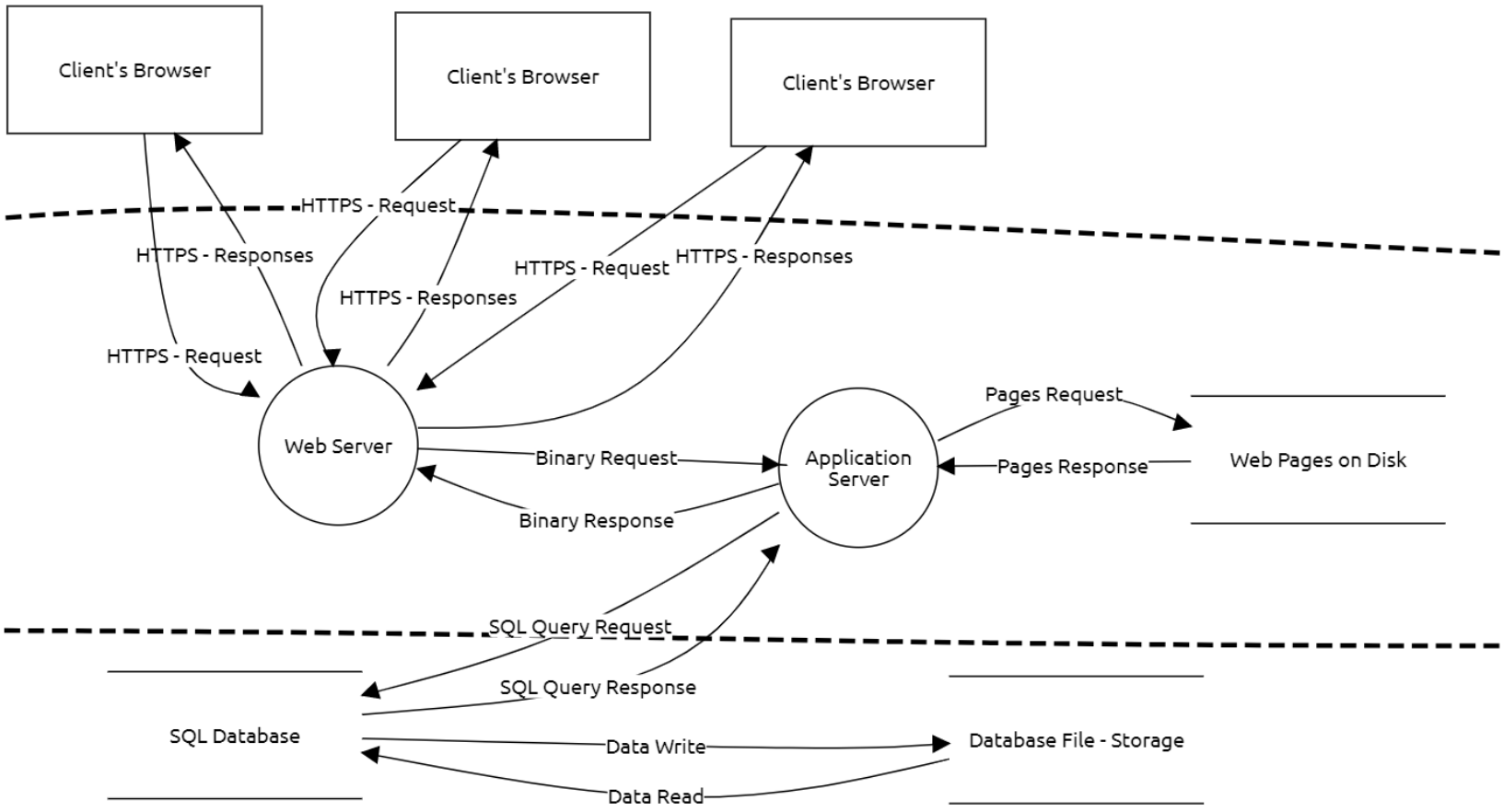
# 1.3 Threat Model Diagram

**Using OWASP Threat Dragon, build a diagram showing the flow of data in the Juice Shop application and identify 3 possible threats to the Juice Shop. Make sure to include the following components:**

- **Client**
- **Web Server**
- **Application Server**
- **Database**

# 1.3 Threat Model Diagram

**Insert the threat Model Diagram Here:**




- *SQL Injection*
- *Cross Site Scripting (XSS)*
- *Broken Authentication*

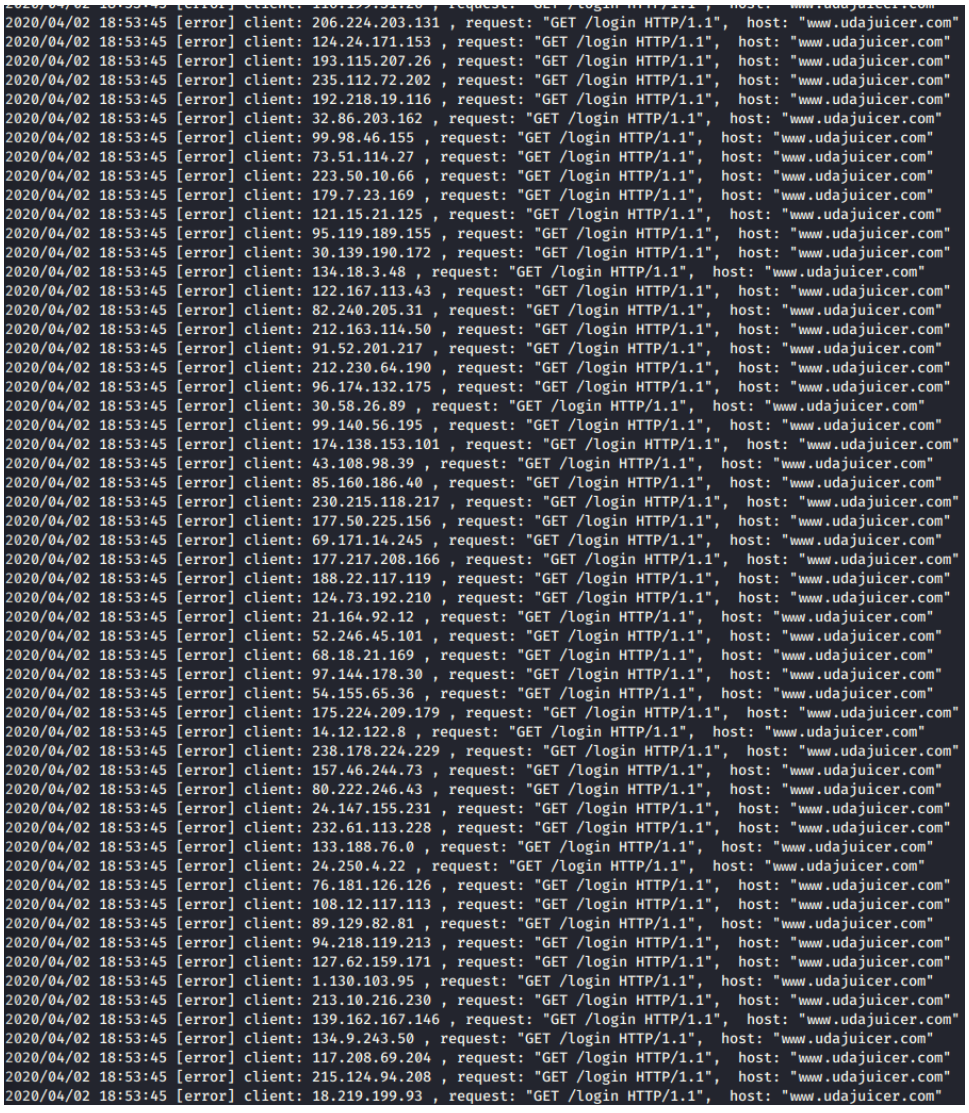
# 1.4 Threat Analysis

## What Type of Attack Caused the Crash?

Denial of Service (DoS)

## What in the Logs Proves Your Theory?

There are huge amounts of requests sent at the same time from different IPs to overwhelming the site and causing it to crash



# 1.5 Threat Actor Analysis

**Who is the Most Likely Threat Actor?**

Script Kiddies

**What Proves Your Theory?**

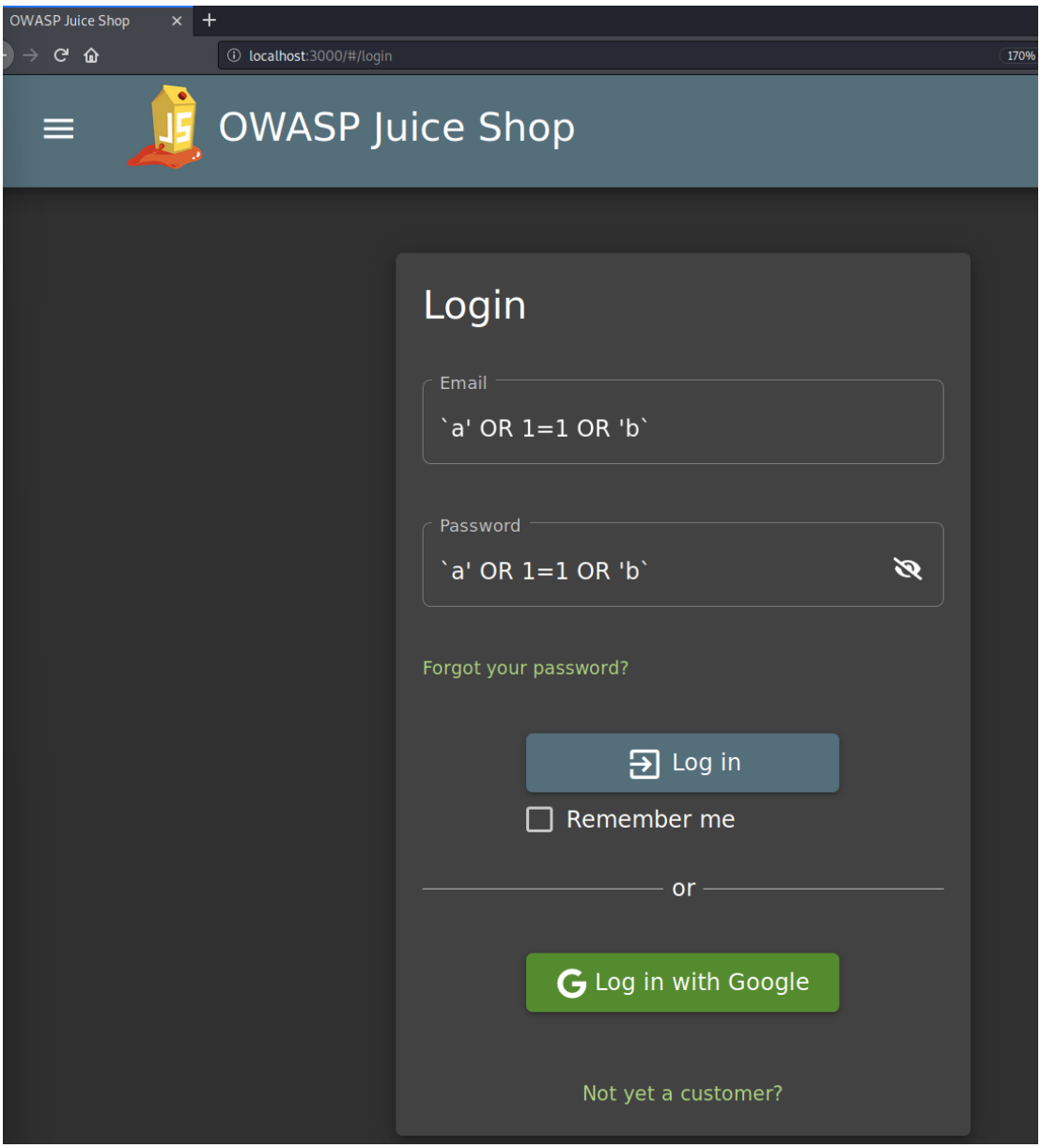
At first glance, it comes to mind that the threat actor responsible for the attack is Hacktivists, but Since the attack does not carry any political motives or ideological goals, because the attack targeted a site that sells juices, which is not related to any political or ideological goals. So, the threat actor should be Script Kiddies which are amateur attackers but there are prebuilt scripts that help in DoS attacks, and they use prebuilt scripts to perform this attack.

# Section 2

Vulnerability Analysis

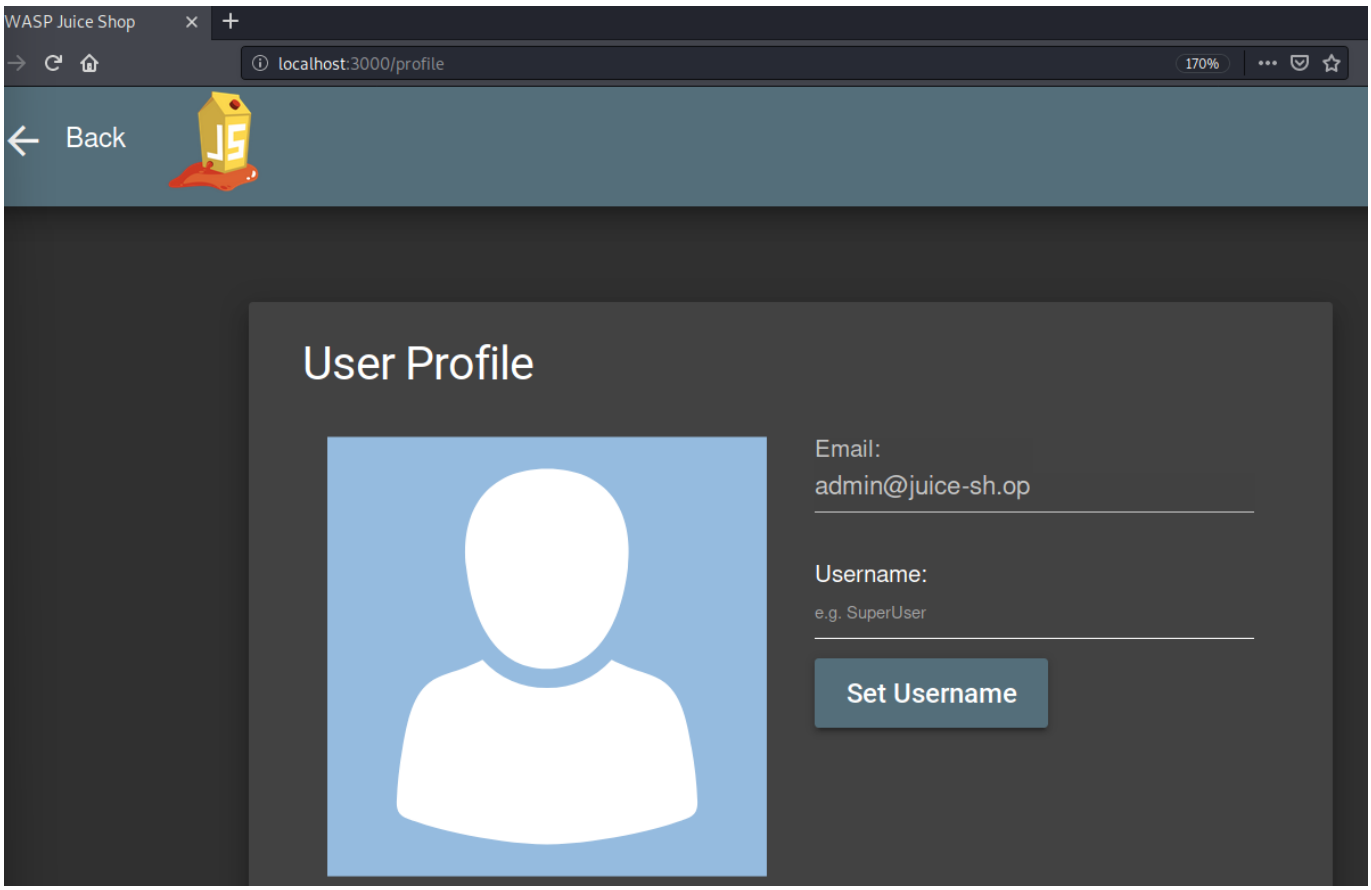
# 2.1 SQL Injection

## Insert Screenshot of Your Commands Here:



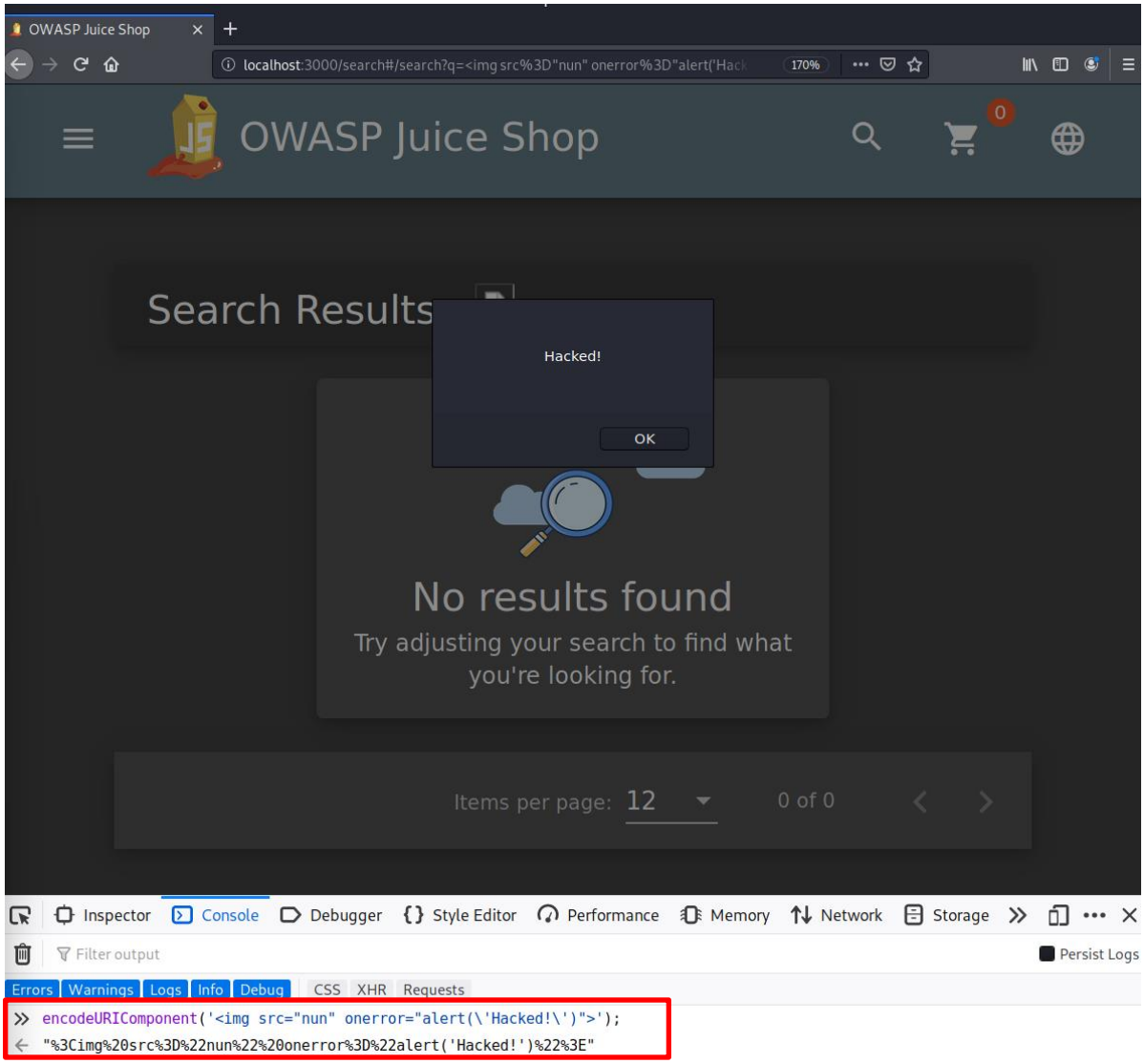
# 2.1 SQL Injection

## Insert Screenshot of Account Settings Showing You as Admin Here:



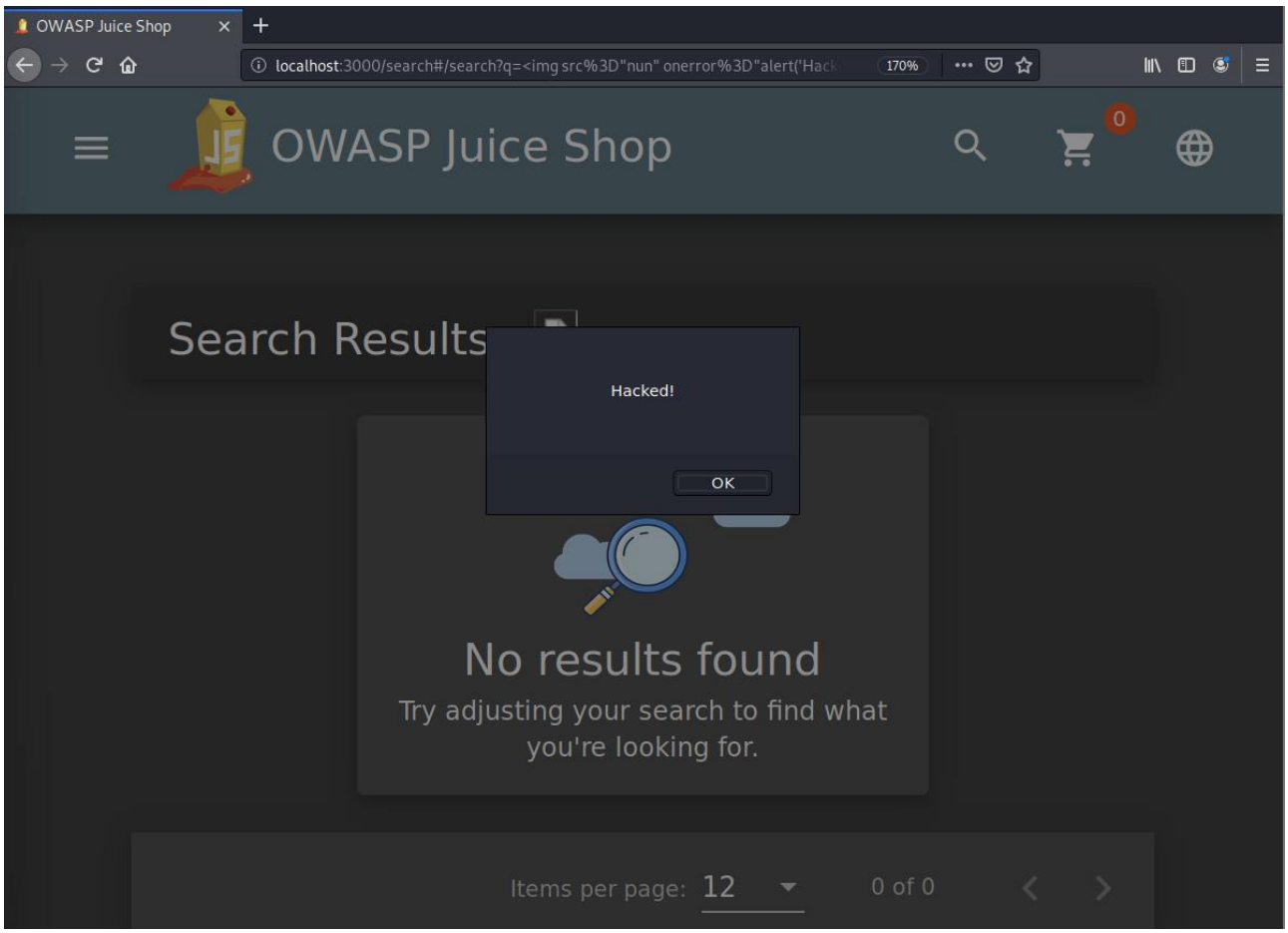
# 2.2 XSS

## Insert Screenshot of Your Commands Here:



# 2.2 XSS

**Insert Screenshot of `alert()` popup saying "Hacked!" Here:**



# Optional Task:

**Extra Vulnerabilities**

- *Insecure Design*
- *Sensitive Data Exposure*
- *Cryptographic Failures*

# Section 3

Risk Analysis

# 3.1 Scoring Risks

| **Risk** | **Score**<br>***(1 is most dangerous, 4 is least dangerous)*** |
|---|---|
| *Denial of Service (DoS)* | 1 |
| Insecure Architecture | 2 |
| SQL Injection | 1 |
| XSS Vulnerability | 3 |

# 3.2 Risk Rationale

**Why Did You Choose That Ranking?**

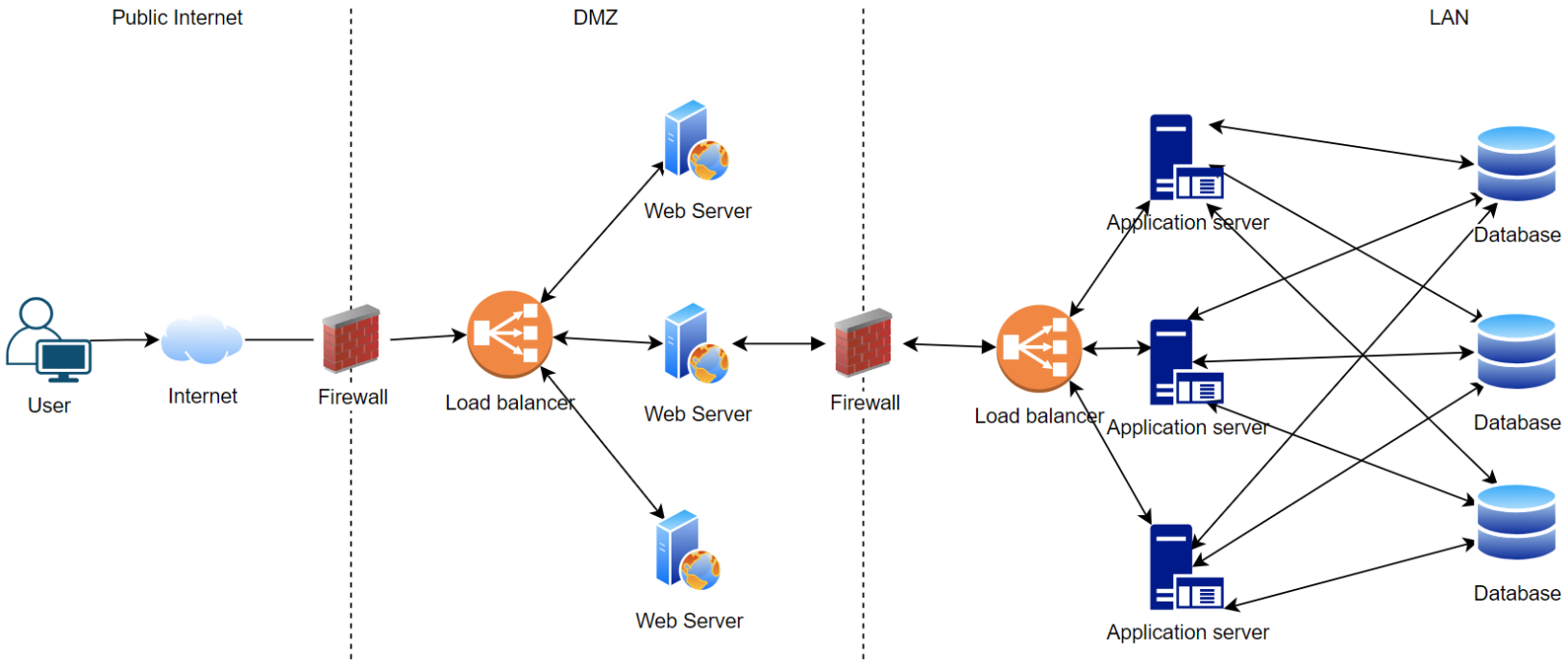
- **Denial of Service (DoS)** Score is 1 because an amateur hacker may do it and making a resource unavailable has a strong impact in terms of reputational and financial loss.
- ***Insecure Architecture*** Score is 2 since it may lead to other attacks such as DoS, SQL injection, and data breach.
- ***SQL Injection*** Score is 1 because it causes severe damage to the company's reputation by leaking data or spoofing identity and tampering with existing data.
- ***XSS Vulnerability*** The score is 3 because it is easy to avoid by either users or website owners and needs a professional hacker.

# Section 4

## Mitigation Plan

# 4.1 Secure Architecture

## Insert Image of Your Secure Architecture Here:



# 4.2 Mystery Attack Mitigation

**What is Your Mitigation Plan?**

We can mitigate Denial of service (DoS) by implementing a secure architecture by using

- Content Delivery Network (CDN) which provides a group of edge servers that serve cached content from the origin.
- Load balancers to distribute incoming traffic evenly from client devices to servers
- Firewalls to filter requests upstream long before they reach the target network.
- Network segmentation technic to mitigate the attack.

# 4.3 SQL Injection Mitigation

**What is Your Mitigation Plan?**

We can mitigate SQL Injection attacks by using

- Input Sanitization to clean user input,
- Input Validation to match user input against expected input,
- Use Prepared Statements with Parameterized Queries which will be waiting for parameters in the form of user input.
- Escaping process which will convert characters in code, so they don't get interpreted.

# 4.4 XSS Mitigation

**What is Your Mitigation Plan?**

We can mitigate XSS attacks by using

- Input Sanitization to clean user input,
- Input Validation to match user input against expected input
- Escaping process which will convert characters in code, so they don't get interpreted.