Udajuicer:
Threat Report

YOUR NAME: Thioro

Fall

DATE: 5/14/2

Section 1

Threat Assessment

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

1.1 Asset Inventory

Components and Functions

- ✓ ***Web Server:*** *process and distributes the information to the end users*
- ✓ ***Application Server:*** *helps to access the informations requested by the end users from the database.*
- ✓ ***Database:*** *Storage of data*

1.1 Asset Inventory (cont)

How a request goes from client to server

- ✓ *When a client makes a request, it establishes a communication between the client's browser and the web server. The client needs an internet connection to place an order at the shop. The internet will connect the client to the shop web server.*

- ✓ *From the web server, a communication request is established with the application server.*

- ✓ *Then from the application server, the request goes to the SQL database.*

1.2 Architecture Audit

Flaws

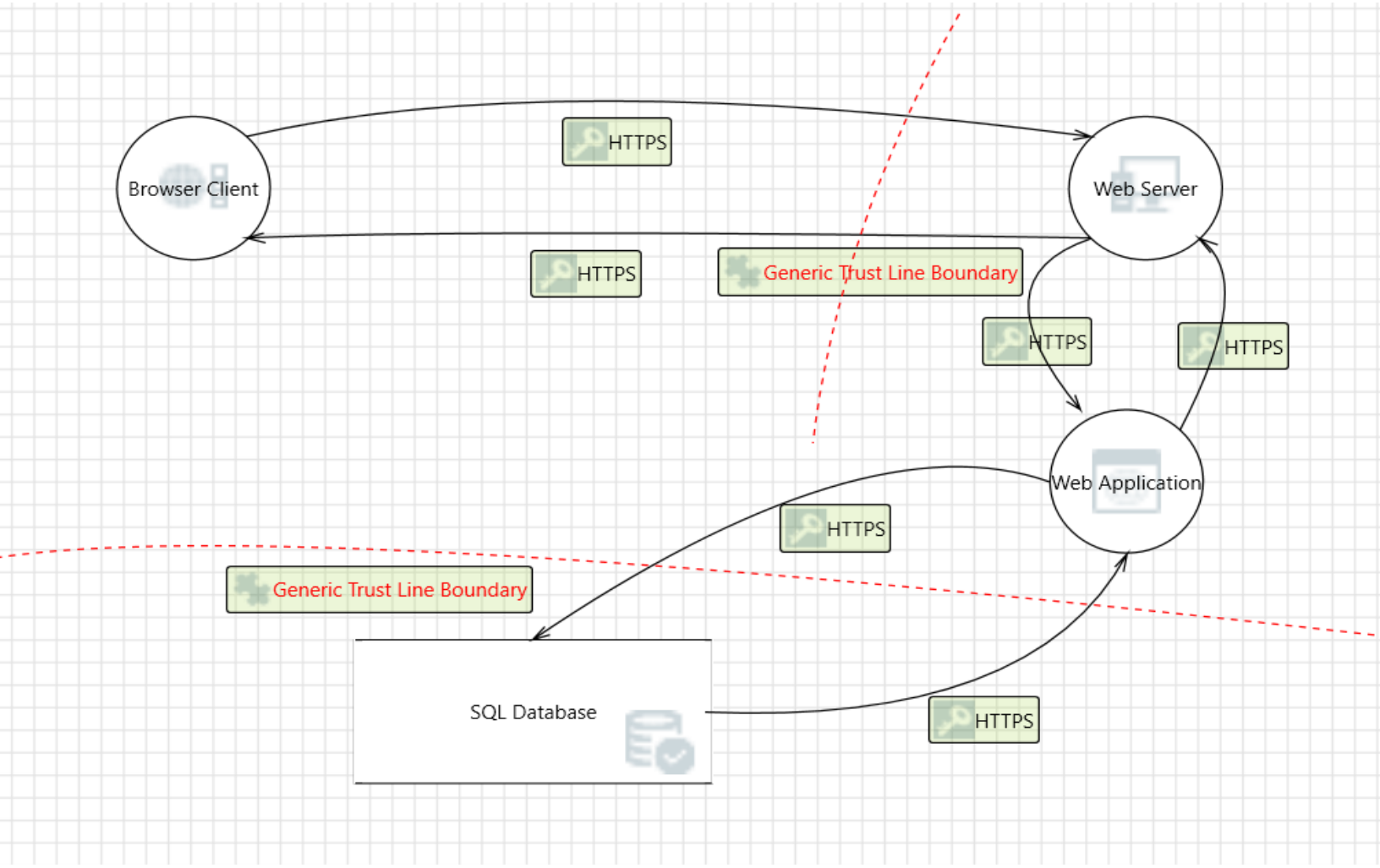
- ✓ *No trust boundary between client-web server.*
- ✓ *No trust boundaries between application sever-database.*
- ✓ *No secure request (HTTPS, SQL..).*
- ✓ No CDN to resolve the problem of SPOF (single point of failure) using distributed server clusters.
- ✓ No load balancers to resolve the issue of balancing high traffic across clustered servers.

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



Threat Model Diagram

1.3 Threat Model Diagram (cont)

Three possible threats that would cause the server to go constantly down are:

- ✓ DoS
- ✓ XML external entities (XXE)
- ✓ Using components with know vulnerabilities

1.4 Threat Analysis

What Type of Attack Caused the Crash?

DDoS (Distribution Denial of Service) attack

- ✓ The system receives too much traffic to the server to buffer.
- ✓ Attacker may attack the web application er server by creating as many processes and transactions possible.

What in the Logs Proves Your Theory?

- ✓ No trust boundaries between **client-server-database** which cause Man in the-Middle attack.
- ✓ No CDN and load balancers

1.5 Threat Actor Analysis

Who is the Most Likely Threat Actor?

Script kiddies

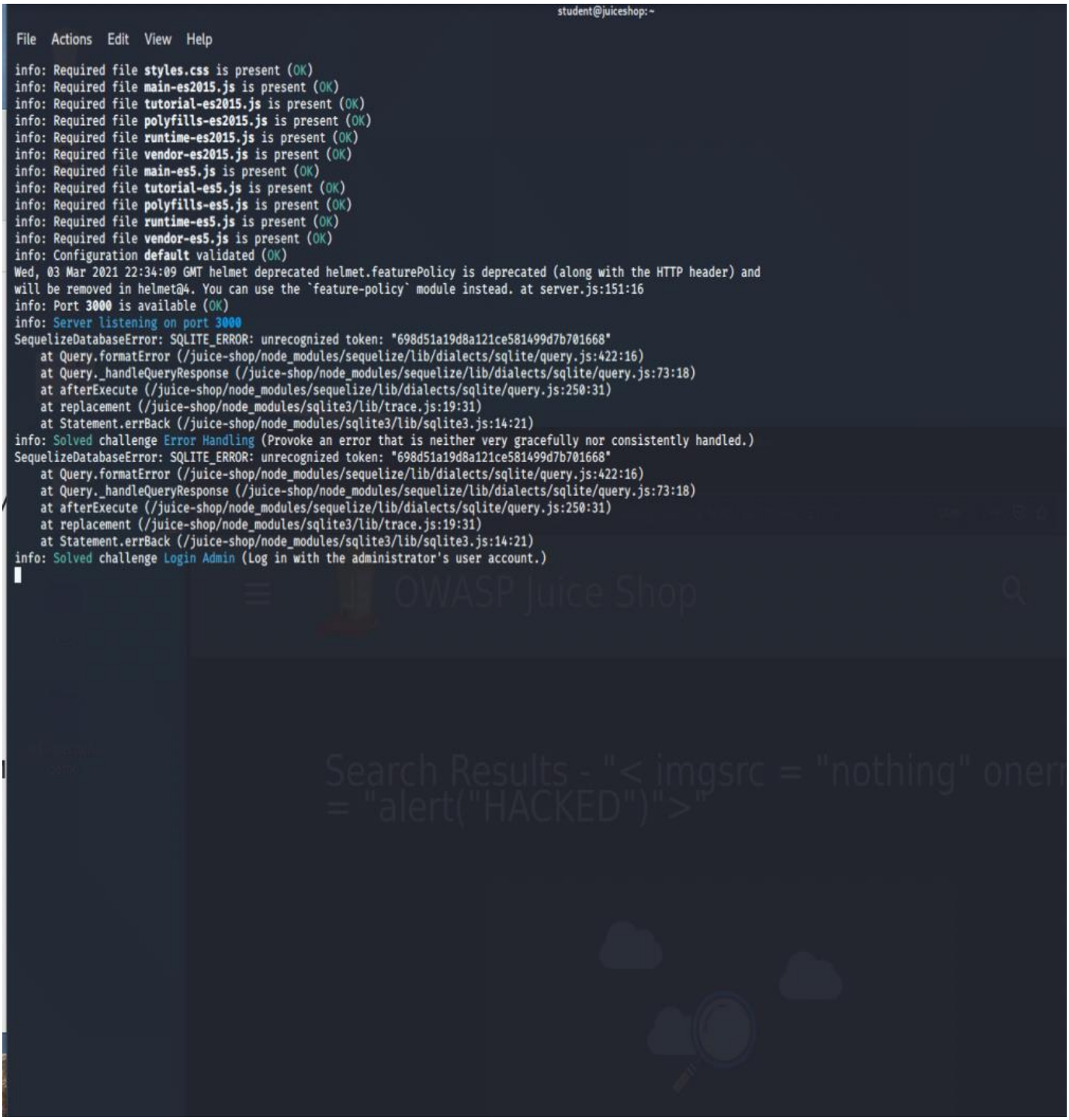
What Proves Your Theory?

- ✓ The server is open to anyone.
- ✓ The attacker may exploite vulnerabilities of websites when placing an order.
- ✓ The attacker may have limited knowledge since the system did not crash immediately.

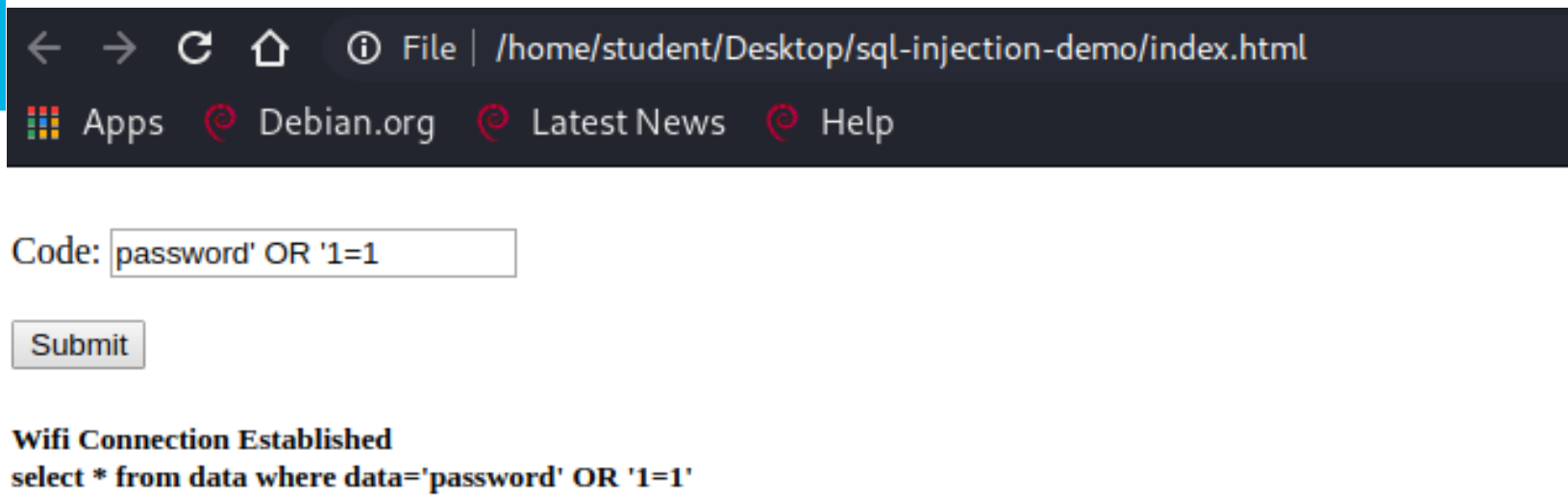
Section 2

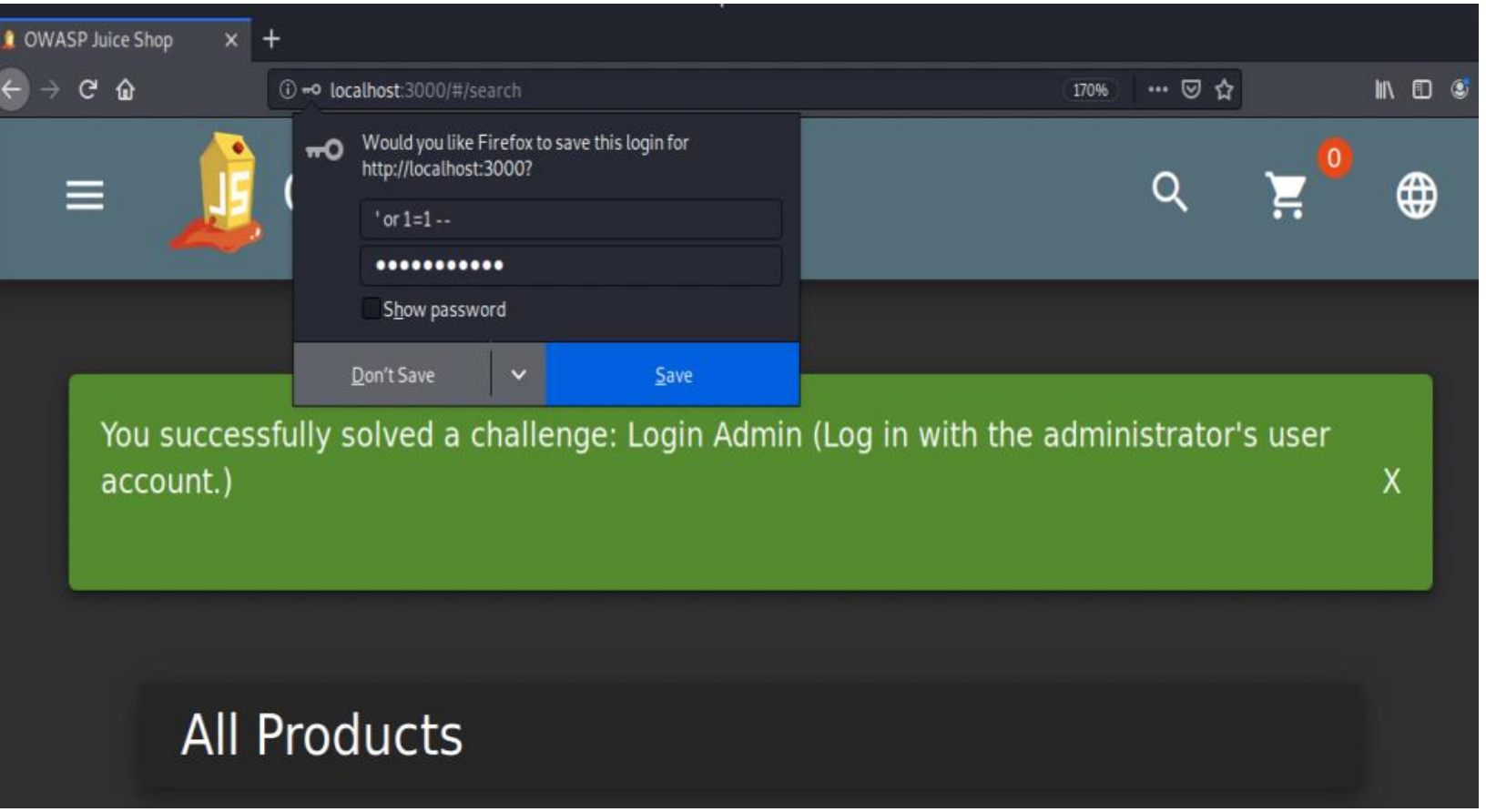
Vulnerability Analysis

2.1 SQL Injection

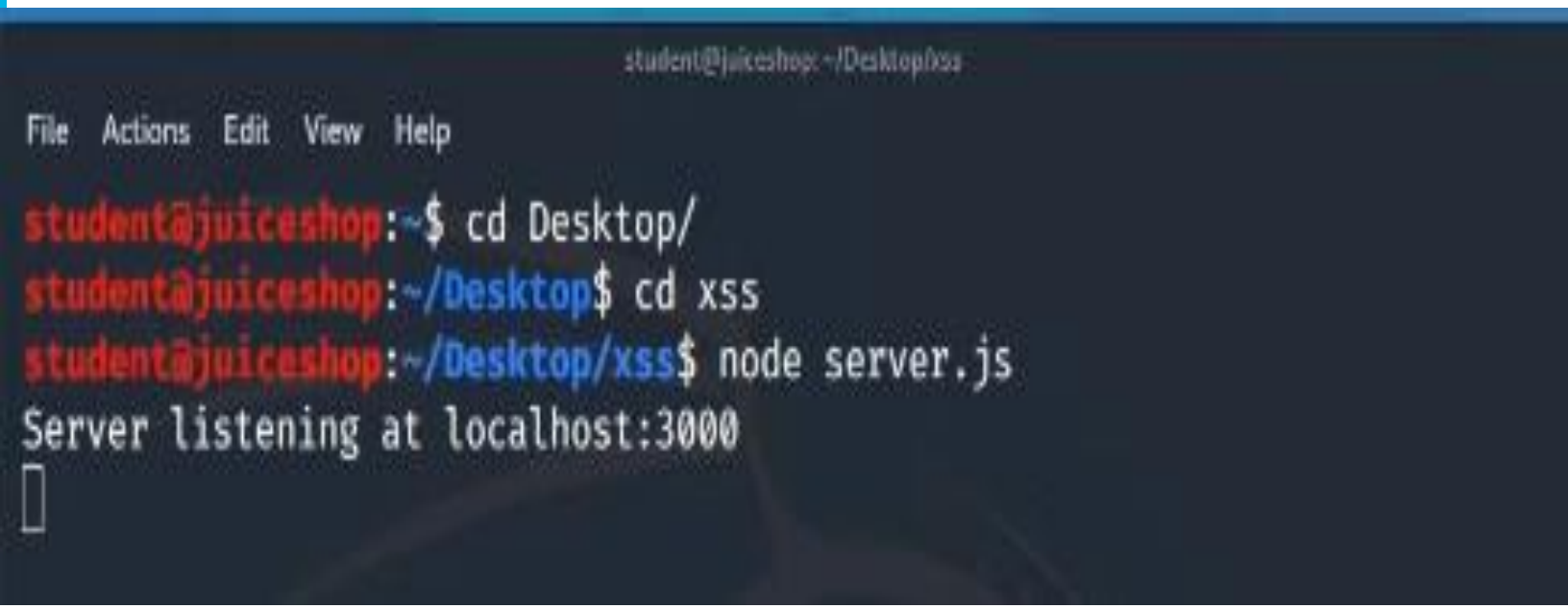


2.1 SQL Injection (cont)

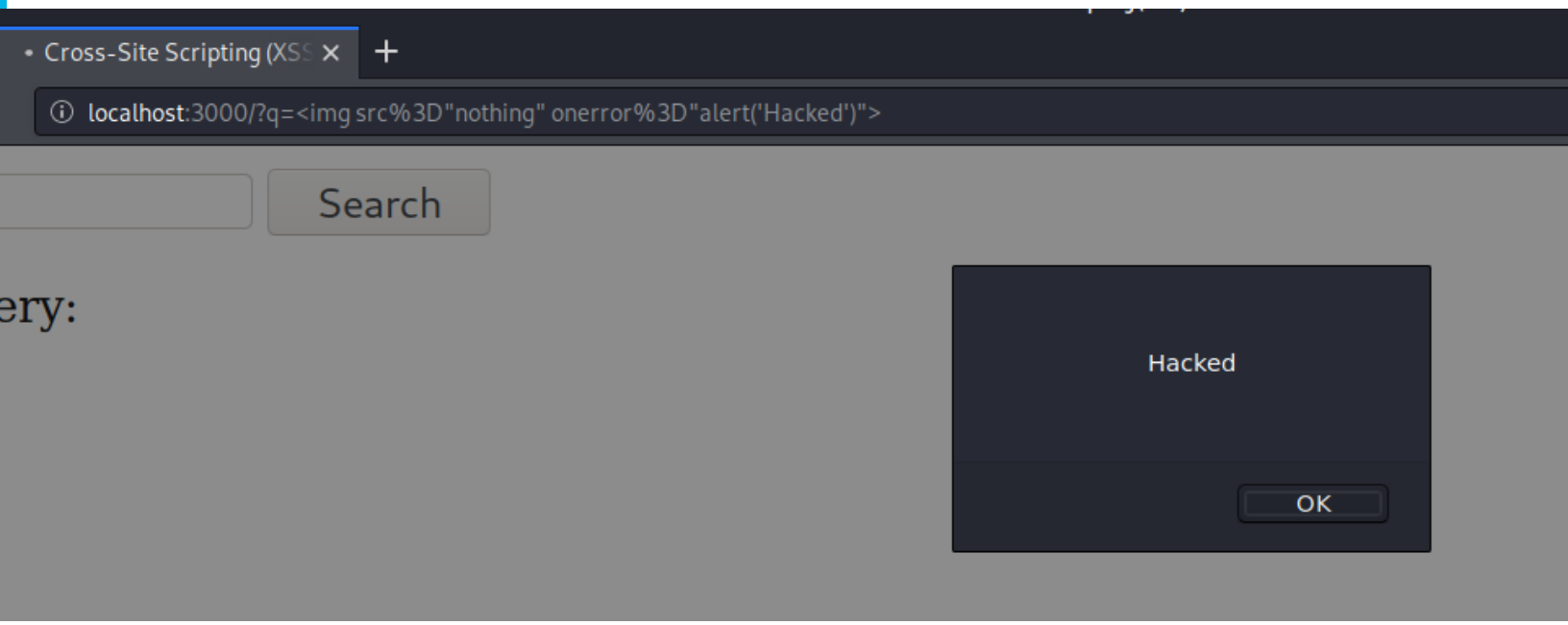




2.2 *XSS*


student@juiceshop: ~/Desktop/xss
File Actions Edit View Help
student@juiceshop:~$ cd Desktop/
student@juiceshop:~/Desktop$ cd xss
student@juiceshop:~/Desktop/xss$ node server.js
Server listening at localhost:3000

2.2 XSS (cont)


Cross-Site Scripting (XSS
localhost:3000/?q=<img src%3D"nothing" onerror%3D"alert('Hacked')">
Search
ery:
Hacked
OK

Section 3

Risk Analysis

3.1 Scoring Risks

Risk	**Score** *(1 is most dangerous, 4 is least dangerous)*
Denial of service	1
Insecure Architecture	2
SQL Injection	3
XSS Vulnerability	4

3.2 Risk Rationale

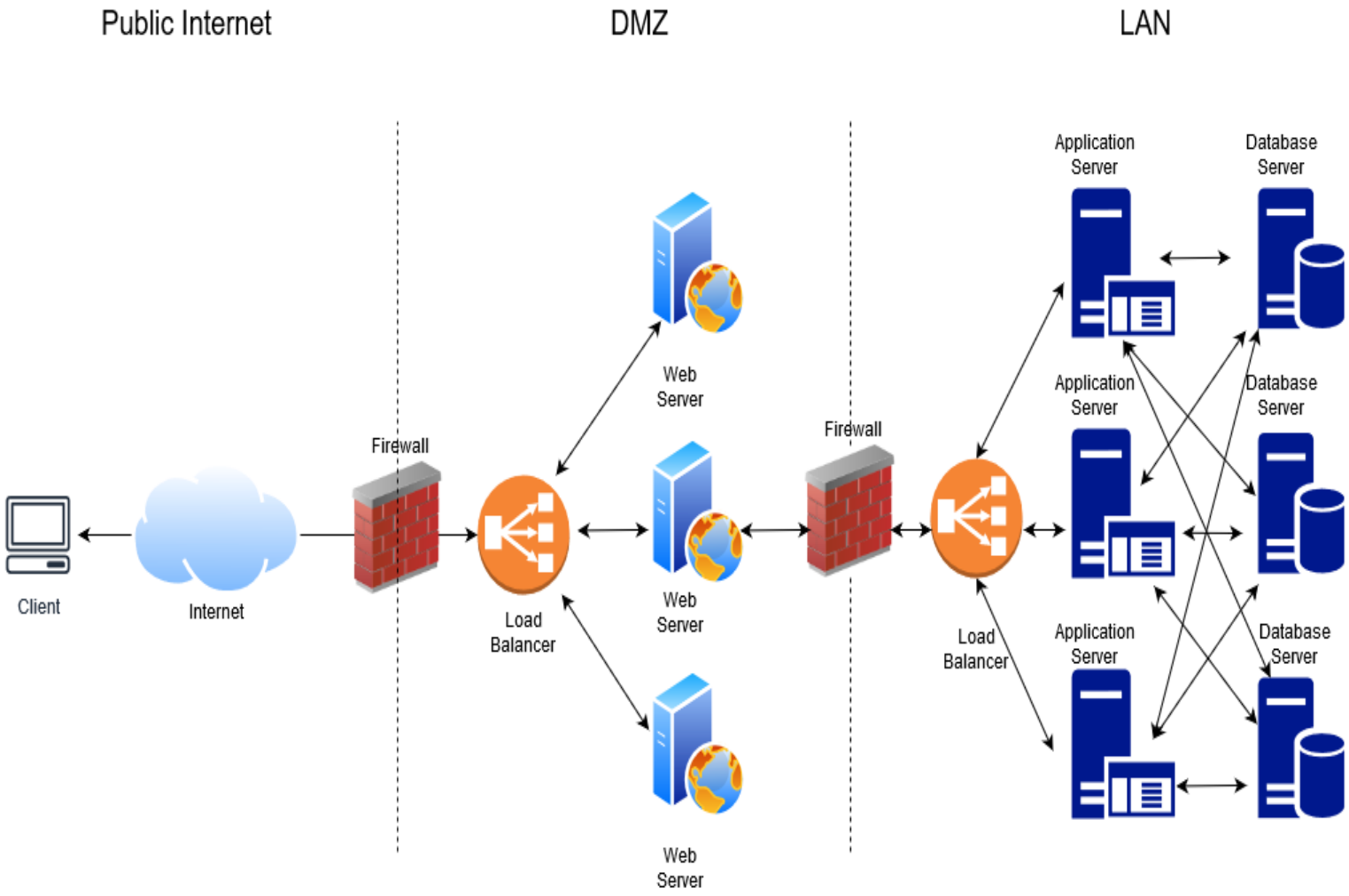
Why Did You Choose That Ranking?

- ✓ First, **DoS** which is the primary concern of the business right now. The server is receiving too much traffic and the system is constantly going down.
- ✓ Second, insure architecture because the company should first take care of the present DoS vulnerability situation.
- ✓ Third, SQL due also to the online demand. This is another type of vulnerability that can slow down or crash the server.
- ✓ Last, XSS, due to the COVID-19, the shop function with online orders where customers can place orders for delivery/pickup. This is another type of vulnerability that can slow down or crash the server as well.

Section 4

Mitigation Plan

4.1 Secure Architecture



Example of secure architecture

4.2 Mystery Attack Mitigation

Mitigation Plan:

- ✓ A secure architecture
- ✓ Trust boundaries (perimeter firewall, WAF…)
- ✓ Encrypt the data
- ✓ Input filtering and sanitization
- ✓ Load balancing
- ✓ CDN

4.3 SQL Injection Mitigation

Mitigation Plan:

- ✓ Input sanitization
- ✓ Input validation
- ✓ Prepared statements with parameterized queries
- ✓ Escaping

4.4 XSS Mitigation

Mitigation Plan:

- ✓ Sanitizing user input
- ✓ Validating user input
- ✓ Escaping