2/7/2025

OWSAP JUICE SHOP

SECURITY AUDIT REPORT

TABLES OF CONTENTS

1. EXECUTIVE SUMMARY

- This report document hereby describes the proceedings and results of a Black Box security assessment conducted against **OWSAP Juice shop Web Application**. The report hereby lists the findings and corresponding best practice mitigation actions and recommendations. The purpose of this assessment was to point out security loopholes, business logic, Errors, and missing best security practices. The tests were carried out assuming the Identity of an attacker or a malicious user but no harm was made to the functionality or working of the application.

1.1 OBJECTIVES

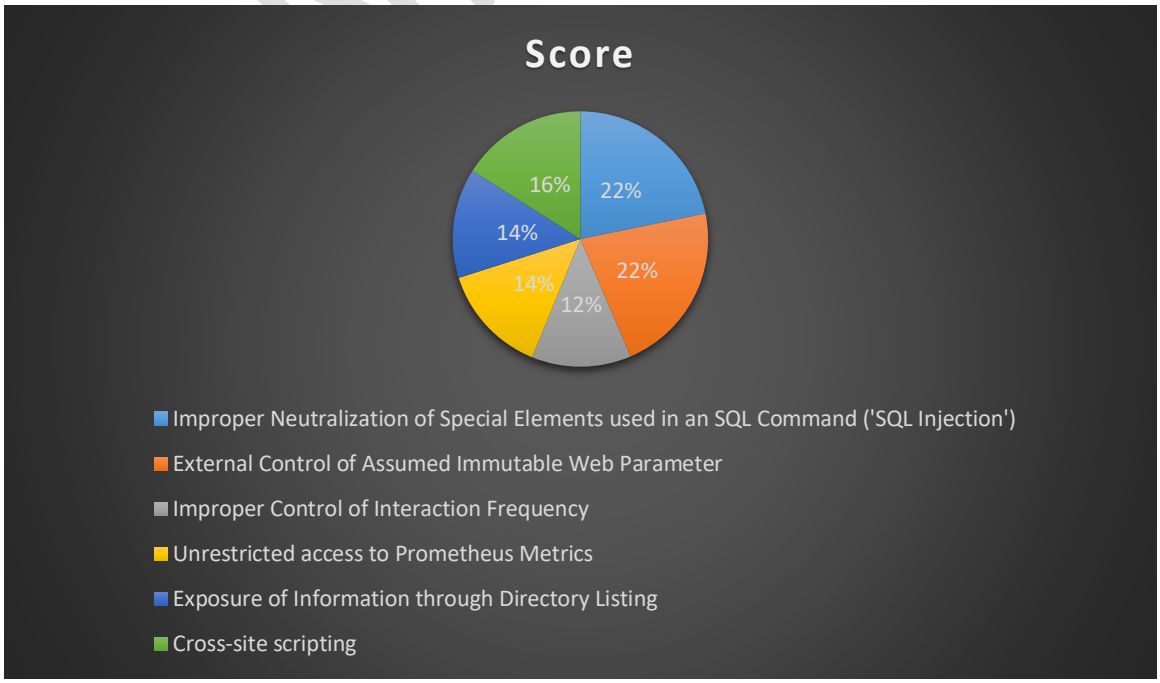
- The objective of the assessment was to assess the state of security and uncover vulnerabilities in **OWSAP Juice shop Web Application** and provide with a final security assessment report comprising vulnerabilities, remediation strategy and recommendation guidelines to help mitigate the identified vulnerabilities and risks during the activity.

1.2 SCOPE OF TESTING

- Security assessment includes testing for security loopholes in the scope defined below. Apart from the following, no other information was provided. Nothing was assumed at the start of the security assessment.

1.3 GRAPHICAL SUMMARY

- The below graphical representations from **OWSAP Juice shop VAPT** dashboard will provide you an overall summary of the security audit scan results, including, vulnerabilities discovered, severity, respective CVSS Score, and other vulnerability details such as its impact, detailed PoC, steps to reproduce, affected URLs/network parameters, and recommended fixes.



1.1 LIST OF VULNERABILITIES

Vulnerability	Score	Severity
Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	8.3	High
External Control of Assumed Immutable Web Parameter	8.3	High
Improper Control of Interaction Frequency	4.8	Medium
Unrestricted access to Prometheus Metrics	5.3	Medium
Exposure of Information through Directory Listing	5.3	Medium
Cross-site scripting	6.1	Medium

1. DISCOVERED VULNERABILITIES DETAILS

#1 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

CVSS score

8.3

Details of Vulnerability:

SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

Severity: High
CWE-89
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Impact:

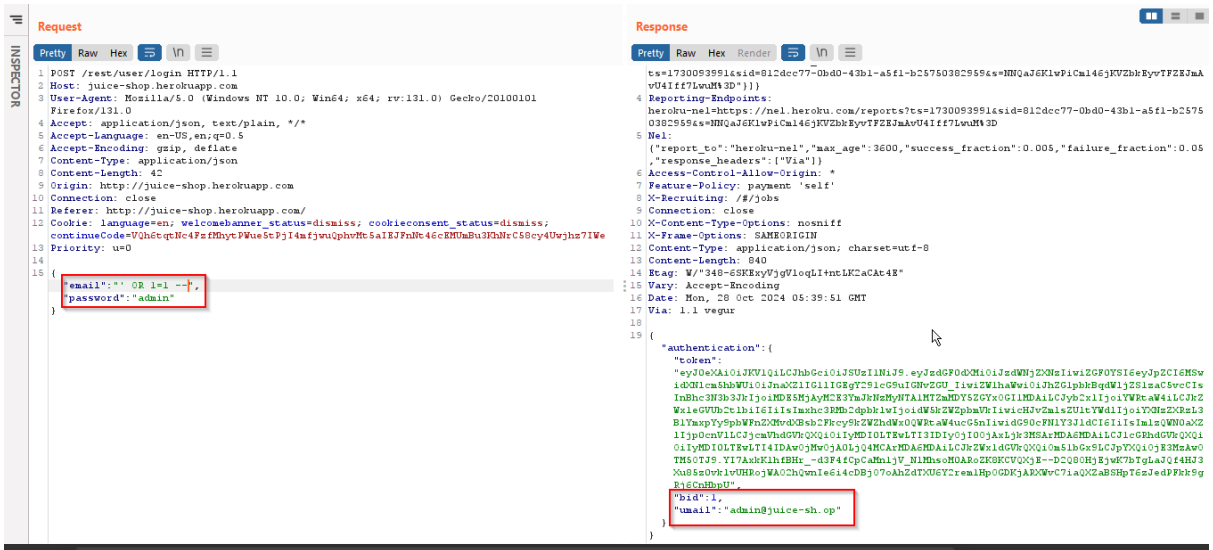
An attacker can use SQL injection to bypass a web application's authentication and authorization mechanisms and retrieve the contents of an entire database. SQLi can also be used to add, modify and delete records in database, affecting data integrity. Under the right circumstances, SQLi can also be used by an attacker to execute OS commands, which may then be used to escalate an attack even further.

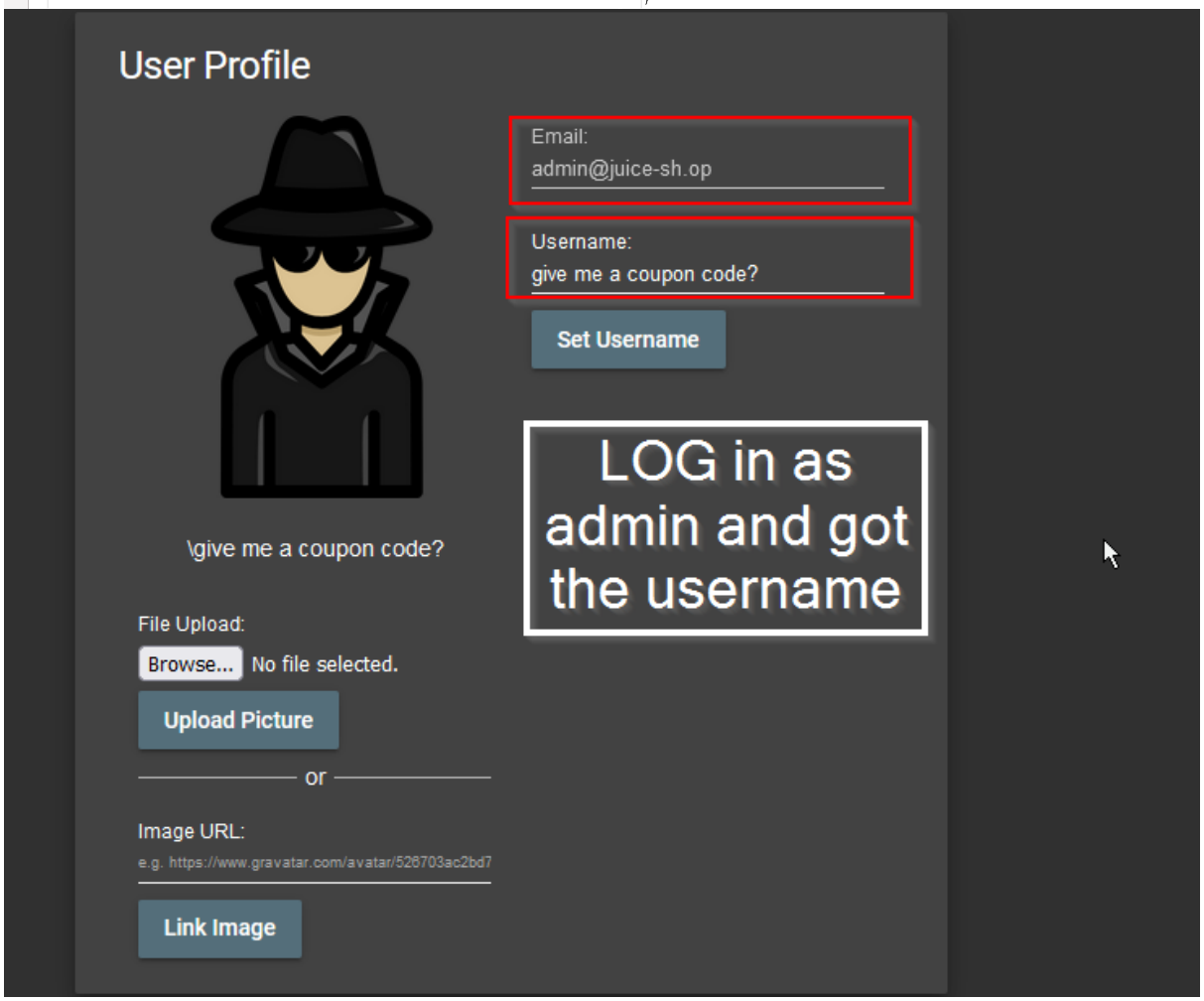
Mitigation:

The only sure way to prevent SQL Injection attacks is input validation and parameterized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.

Affected URL:http://juice-shop.herokuapp.com/#/login

POC:


Request
Pretty Raw Hex
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 42
Origin: http://juice-shop.herokuapp.com
Connection: close
Referer: http://juice-shop.herokuapp.com/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss;
Priority: u=0
{
"email":"' OR 1=1 --",
"password":"admin"
}
Response
Pretty Raw Hex Render
Reporting-Endpoints:
Nel:
Access-Control-Allow-Origin: *
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Connection: close
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Type: application/json; charset=utf-8
Content-Length: 840
Etag: W/"348-6SKExyVjgVloqLI+ntLKZaCAt4E"
Vary: Accept-Encoding
Date: Mon, 28 Oct 2024 05:39:51 GMT
Via: 1.1 vegur
"authentication":{
"token":
"bid":1,
"umail":"admin@juice-sh.op"


User Profile
Email:
admin@juice-sh.op
Username:
give me a coupon code?
Set Username
LOG in as admin and got the username
\give me a coupon code?
File Upload:
Browse... No file selected.
Upload Picture
or
Image URL:
e.g. https://www.gravatar.com/avatar/526703ac2bd7
Link Image

#2 External Control of Assumed Immutable Web Parameter

CVSS score

8.3

Details of Vulnerability:

If a web product does not properly protect assumed-immutable values from modification in hidden form fields, parameters, cookies, or URLs, this can lead to modification of critical data. Web applications often mistakenly make the assumption that data passed to the client in hidden fields or cookies is not susceptible to tampering. Improper validation of data that are user-controllable can lead to the application processing incorrect, and often malicious, input.

Severity: High
CWE-472
CVSS: 3.1/AV: N/AC: L/PR: N/UI: R/S: U/C: H/I: H/A: H

Impact:

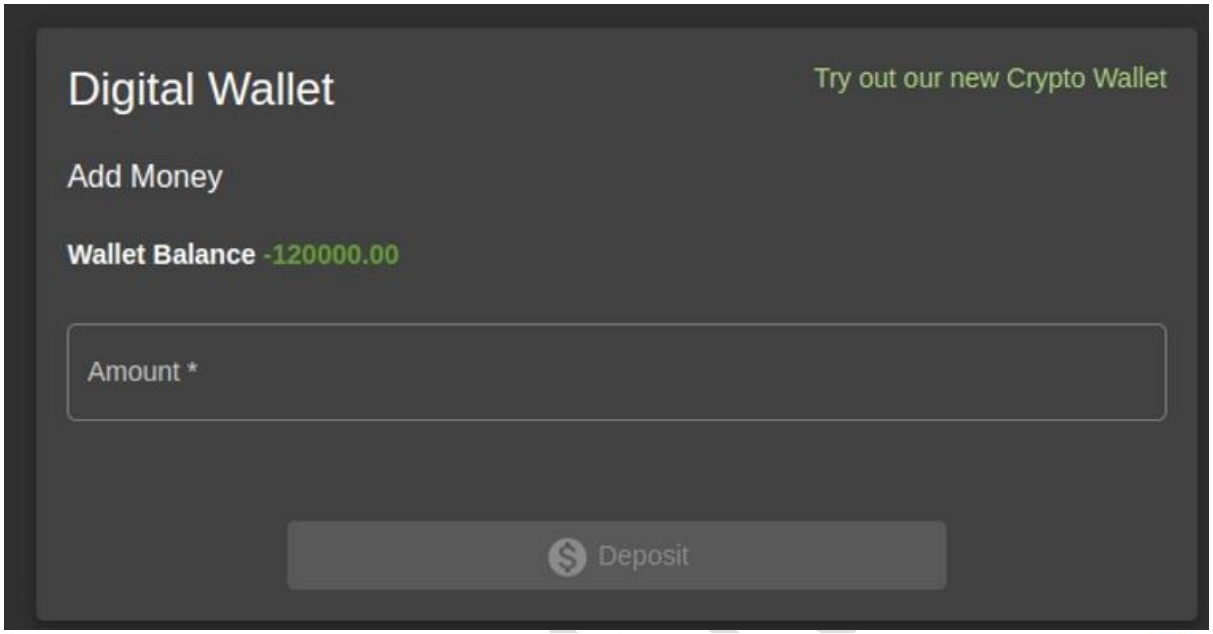
If a web product does not properly protect assumed-immutable values from modification in hidden form fields, parameters, cookies, or URLs, this can lead to modification of critical data.

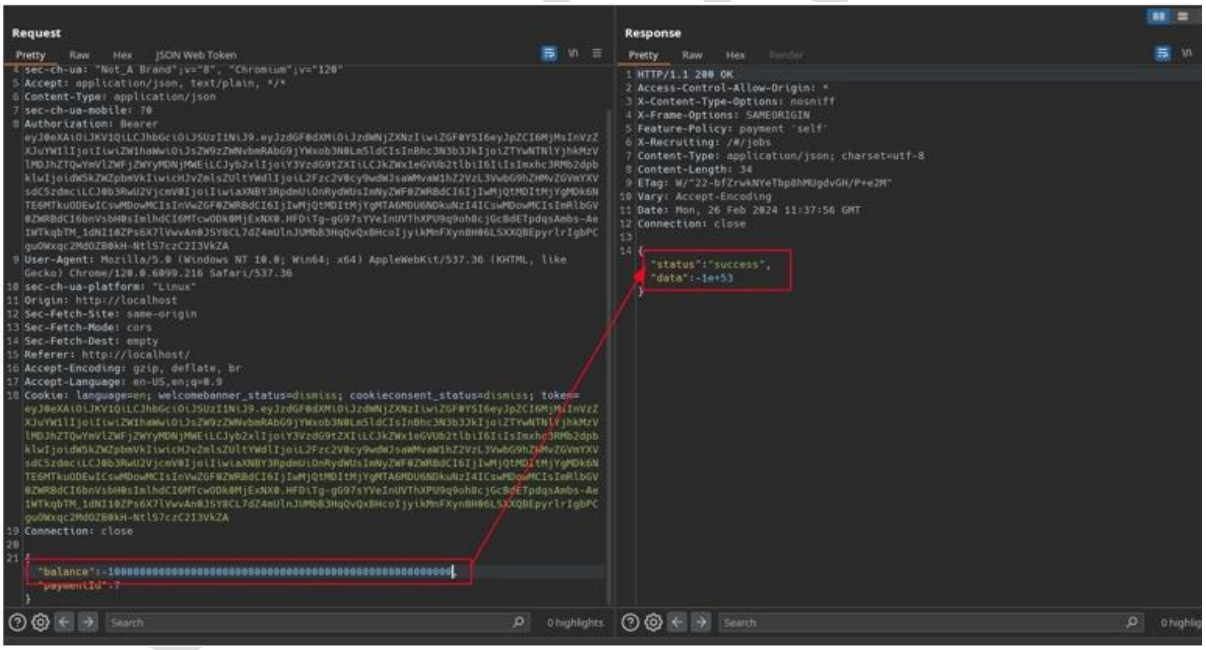
Mitigation:

Effective mitigation requires a layered approach. First, strong server-side validation is essential, ensuring parameters are checked and sanitized on the server rather than relying on client-side validation. Implementing role-based access control (RBAC) and authorization checks for every request further limits access, ensuring users can only modify resources they are permitted to.

Affected URL: https://juice-shop.herokuapp.com/#/order-summary

POC:


Digital Wallet
Try out our new Crypto Wallet
Add Money
Wallet Balance -120000.00
Amount *
Deposit


Request
Pretty
Raw
Hex
JSON Web Token
Response
Pretty
Raw
Hex
HTTP/1.1 200 OK
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: 34
ETag: W/"22-bfZrwANYeTbp8hMUgdvGH/P+e2M"
Vary: Accept-Encoding
Date: Mon, 26 Feb 2024 11:37:56 GMT
Connection: close
"status":"success",
"data":-1e+53

#3 Improper Control of Interaction Frequency

CVSS Score

Details of Vulnerability:

The weakness is caused due to lack of control for number of attempts or requests that are allowed to be sent to the application. A remote attacker can perform a brute-force attack and guess user's password, session token or cause a denial of service.

Severity: Medium
CWE-799
CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L

Impact:

The vulnerability allows an attacker to brute-force access credentials and gain unauthorized access to the application.

Mitigation:

There are several ways to implement protection against brute-force attacks. For example, you can use CAPTCHA to add additional level of protection against automated brute-force attacks. The best approach would be to count the number of unsuccessful attempts and block the user account when that number reaches a critical value. For example, we would recommend to block access to the account for 30 minutes after 5 unsuccessful attempts.

Affected URL: https://juice-shop.herokuapp.com/#/login

POC:

Attack Save Columns 3. Intruder attack of https://juice-shop.he

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Requ...	Position	Payload	Status	Error	Timeout	Length	Comment
185	2	chelsea	401			926	
186	2	biteme	401			926	
187	2	matthew	401			926	
188	2	access	401			926	
189	2	yankees	401			926	
190	2	987654321	401			926	
191	2	dallas	401			918	
192	2	austin	401			918	
193	2	thunder	401			918	
194	2	taylor	401			918	
195	2	matrix	401			918	
196	2	mobilemail	401			918	
197	2	mom	401			918	
198	2	monitor	401			918	
199	2	monitoring	401			918	
200	2	montana	401			918	
201	2	moon	401			918	
202	2	moscow	401			918	
203	2	test1234	401			918	
204	2		401			922	

Request Response

Pretty Raw Hex Render

Invalid email or password.

#4 unrestricted access to Prometheus Metrics

CVSS Score

Details of Vulnerability:

Prometheus is a monitoring system and time series database. It determined that it was possible to access Prometheus interface without authentication.

Severity: Medium
CWE-200
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Impact:

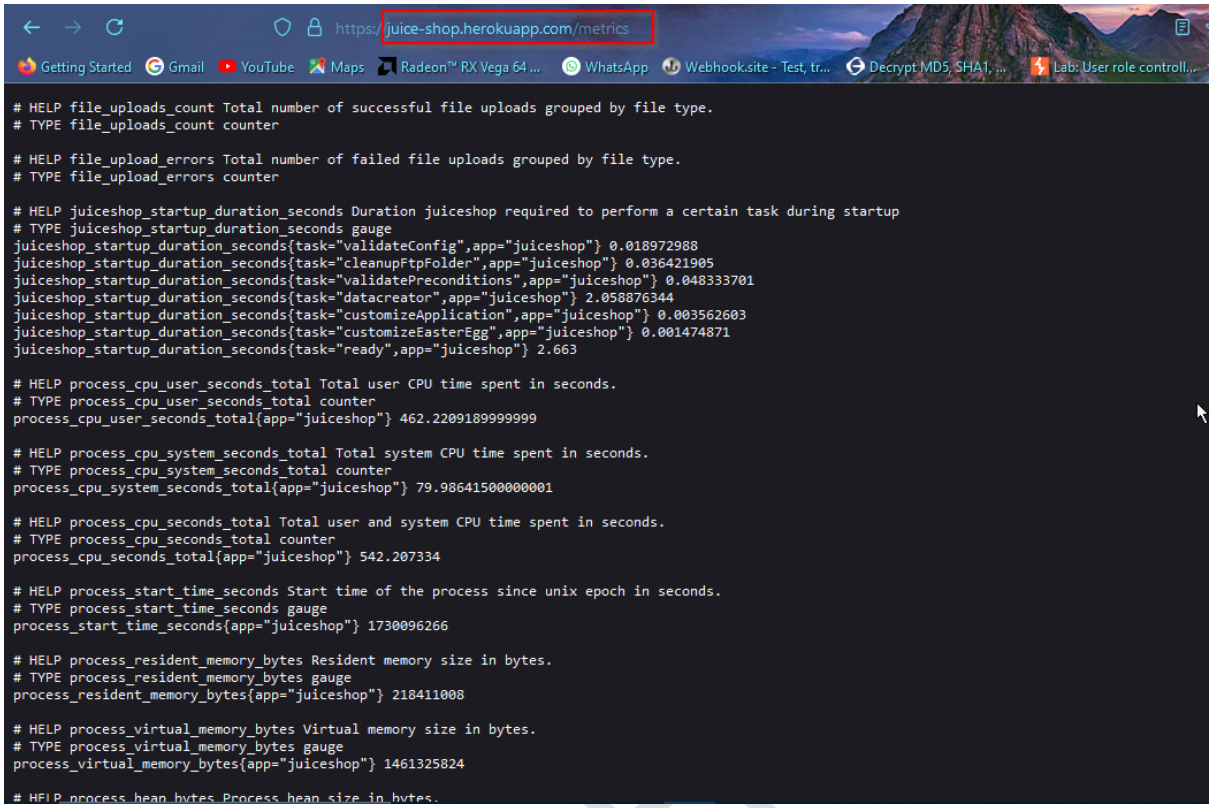
Unrestricted access to Prometheus metrics can pose significant security and operational risks to an organization. Prometheus collects detailed metrics about system and application performance, which, if exposed without restrictions, can provide attackers with a wealth of sensitive information. This includes insights into infrastructure configurations, application health, and service dependencies, which adversaries can leverage to exploit vulnerabilities or optimize attack strategies.

Mitigation:

To mitigate these risks, it's essential to enforce strict access controls on Prometheus endpoints. This can involve integrating authentication and authorization, using HTTPS to encrypt data in transit, and, if possible, restricting access to specific IP addresses or trusted networks. By securing Prometheus access, organizations can protect sensitive data, maintain operational efficiency, and reduce their risk profile against potential attacks.

Affected URL: https://juice-shop.herokuapp.com/metrics

POC:




```
# HELP file_uploads_count Total number of successful file uploads grouped by file type.
# TYPE file_uploads_count counter

# HELP file_upload_errors Total number of failed file uploads grouped by file type.
# TYPE file_upload_errors counter

# HELP juiceshop_startup_duration_seconds Duration juiceshop required to perform a certain task during startup
# TYPE juiceshop_startup_duration_seconds gauge
juiceshop_startup_duration_seconds{task="validateConfig",app="juiceshop"} 0.018972988
juiceshop_startup_duration_seconds{task="cleanupFtpFolder",app="juiceshop"} 0.036421905
juiceshop_startup_duration_seconds{task="validatePreconditions",app="juiceshop"} 0.048333701
juiceshop_startup_duration_seconds{task="datacreator",app="juiceshop"} 2.058876344
juiceshop_startup_duration_seconds{task="customizeApplication",app="juiceshop"} 0.003562603
juiceshop_startup_duration_seconds{task="customizeEasterEgg",app="juiceshop"} 0.001474871
juiceshop_startup_duration_seconds{task="ready",app="juiceshop"} 2.663

# HELP process_cpu_user_seconds_total Total user CPU time spent in seconds.
# TYPE process_cpu_user_seconds_total counter
process_cpu_user_seconds_total{app="juiceshop"} 462.2209189999999

# HELP process_cpu_system_seconds_total Total system CPU time spent in seconds.
# TYPE process_cpu_system_seconds_total counter
process_cpu_system_seconds_total{app="juiceshop"} 79.98641500000001

# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total{app="juiceshop"} 542.207334

# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds{app="juiceshop"} 1730096266

# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes{app="juiceshop"} 218411008

# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes{app="juiceshop"} 1461325824

# HELP process_heap_bytes Process heap size in bytes.
```

#5 Exposure of Information through Directory Listing

CVSS Score

5.3

Details of Vulnerability:

A directory listing vulnerability means that the webserver lists the contents of its directories, allowing the adversary to easily browse all the files within the affected directories. Often, this causes sensitive files to be exposed to the world, such as internal reports, logs, backups and even the source code of the application.

Severity: Medium
CWE-548
CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

Impact:

When directory listing is enabled, it can inadvertently reveal files and directories that were not meant to be public. This exposure can lead to unauthorized users accessing sensitive information such as configuration files, source code, and personal data.

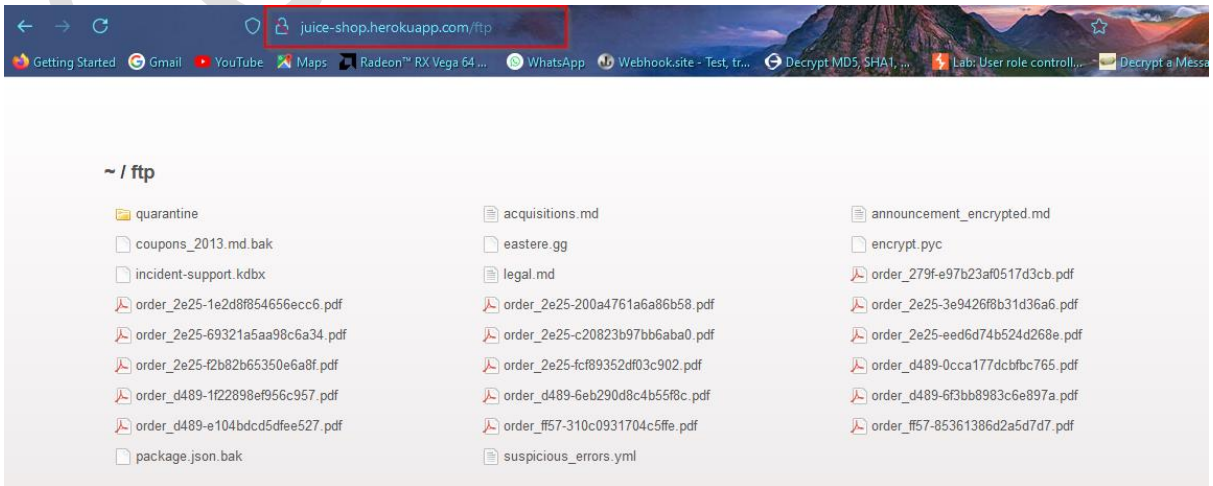
Mitigation:

There is not usually any good reason to provide directory listings, and disabling them may place additional hurdles in the path of an attacker. This can normally be achieved:

- Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.

Affected URL: http://juice-shop.herokuapp.com/ftp

POC:



#6 Cross-site scripting (XSS)

CVSS score

Details of Vulnerability:

Cross-site scripting (XSS) is a web security vulnerability that allows attackers to inject malicious code into a website, which can then be executed by the user's browser.

Severity: Medium
CWE-79
CVSS:3.0/AV:N/.AC:L/.PR:N/.UI:R/.S:C/.C:L/.I:L/.A:N

Impact:

Cross site scripting attacks can have devastating consequences. Code injected into a vulnerable application can exhilarate data or install malware on the user's machine. Attackers can masquerade as authorized users via session cookies, allowing them to perform any action allowed by the user account.

Mitigation:

To keep web-application safe from XSS, you must sanitize your input. Your application code should never output data received as input directly to the browser without checking it for malicious code.

Affected URL:

https://juice-shop.herokuapp.com/#/search?q=%3Ciframe%20src%3D%22javascript:alert(1)%22%3E%60

POC: