SECURITY ASSESSMENT

Udajuicer Vulnerability Assessment Report

Submitted to: Udeajuicer developers
Security Analyst: Mohammad Alsowaini

Date of Testing: 10/12/2022
Date of Report Delivery: 11/12/2022

Table of Contents

Contents

Security Engagement Summary

Engagement Overview

This report is made for Udajuicer development team for assessing any kind of vulnerability in the web page. The purpose of this report is to cover any possible vulnerability. The analyst will try his best to assess the vulnerabilities. Vulnerability assessment report may never be completed, but it reduce the risk to the organization. Vulnerability assessment frequently depend on your risk level and the type of data you have on your systems and networks, but we can say at least quarterly.

Scope

This assessment will be scanning Udajuicer web server port 3000. And must be delivered by 11 Dec. This report the developers team to reduce the risk to the organization

Executive Risk Analysis

Overall risk level is High.

The vulnerabilities I found in this is such a high risk vulnerabilities, not all of it.

Vulnerabilities such as XSS which will lead to upload malicious files, Broken Access Control which will allow to access the administration section of the store and Change the name of a user by performing Cross-Site Request Forgery from another origin.

Executive Recommendation

Remediation is the most important part of this report. As a big company any kind of vulnerability must be denied. But we can Prioritize vulnerabilities so we should start with high risk vulnerabilities. Such as Broken Access Control we can remediate it different ways, we can Except for public resources, deny by default or Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage and so on for other vulnerabilities.

Significant Vulnerability Summary

High Risk Vulnerabilities

- Broken Access Control
- Cross Site Scripting (XSS)
- Sensitive Data Exposure

Medium Risk Vulnerabilities

- Security Misconfiguration

Low Risk Vulnerabilities

- Insecure Deserialization
- Broken Authentication

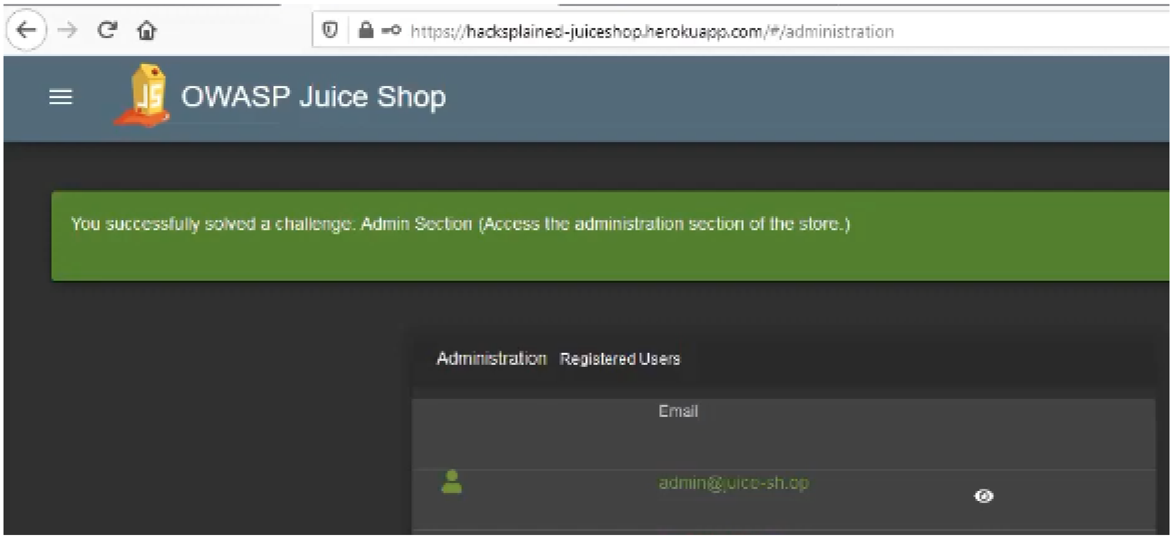
Broken Access Control Detail

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.

Broken Access Control

RISK LEVEL HIGH

- Admin section can be visited simply by typing it's path in the URL.
- This vulnerability is so easy it have a low attack complexity.
- If this vulnerability exploited, admin section maybe used by the attacker, so the administration.
- It can be prevented many ways:
    - Except for public resources, deny by default.
    - Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
    - Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
- Since this vulnerability is against Authentication and it gain a high privileges, it have low complexity and doesn't need any tools, it's probability is very high.
- Screenshot:



https://hacksplained-juiceshop.herokuapp.com/#/administration

OWASP Juice Shop

You successfully solved a challenge: Admin Section (Access the administration section of the store.)

Administration Registered Users

Email
admin@juice-sh.op

Cross Site Scripting (XSS) Detail

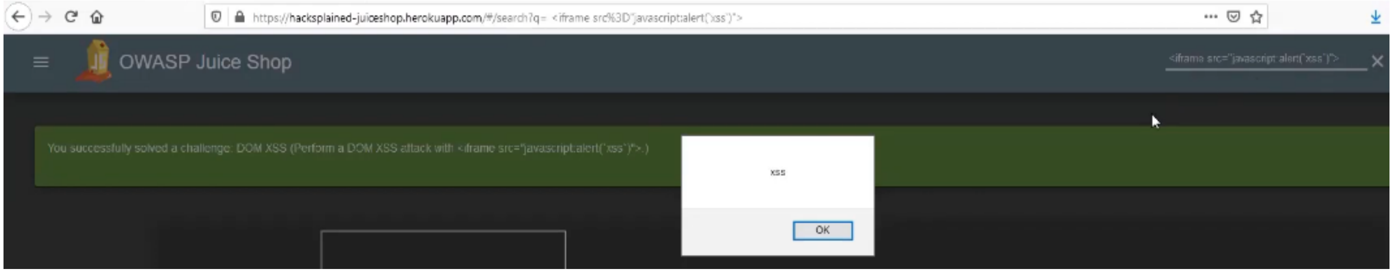
The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information and business secrets require extra protection, particularly if that data falls under privacy laws

Cross Site Scripting (XSS)

RISK LEVEL HIGH

Vulnerability detail

- The search bar can be injected with html commands.
- This vulnerability's attack complexity is low. So it has high probability of exploit.
- The organization, Because of any change in the page components or any malicious uploaded.
- It can be prevented too many ways:
  - The preferred option is to use a safe API, which avoids using the interpreter entirely, provides a parameterized interface, or migrates to Object Relational Mapping Tools (ORMs).
  - Use positive server-side input validation. This is not a complete defense as many applications require special characters, such as text areas or APIs for mobile applications.
- Since this Vulnerability is very popular, low complexity and our website is very famous so script-kiddies may try to have fun or test their knowledge in our website, also real attackers have a high chance to hide malicious scripts in our website, so it's high probability.
- Screenshot:



https://hacksplained-juiceshop.herokuapp.com/#/search?q= <iframe src%3D"javascript:alert(`xss`)">

OWASP Juice Shop

<iframe src="javascript:alert(`xss`)">

You successfully solved a challenge: DOM XSS (Perform a DOM XSS attack with <iframe src="javascript:alert(`xss`)">.)

xss

Sensitive Data Exposure Detail

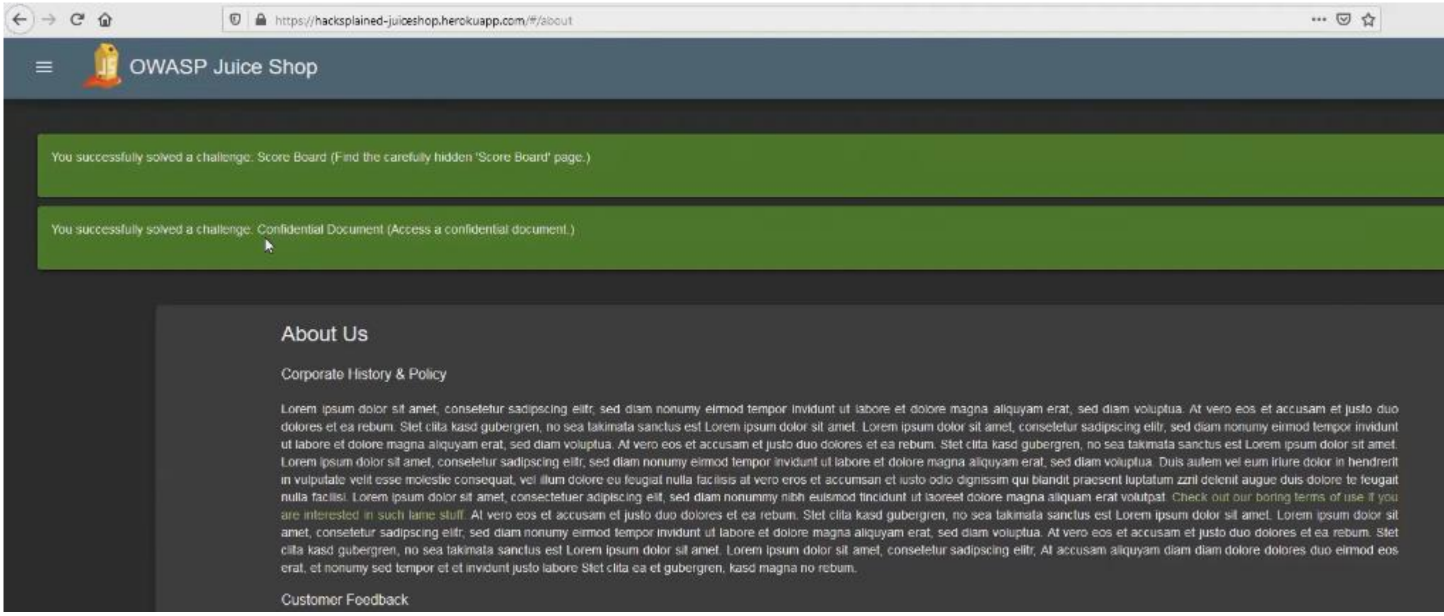
The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information and business secrets require extra protection, particularly if that data falls under privacy laws

Sensitive Data Exposure

RISK LEVEL HIGH

Vulnerability detail

- A document has been found in the http response, I put it in the request and then it seems that it's very confidential.
- This vulnerability's attack complexity is low. So it has high probability of exploit.
- The organization, Because any data breach will be in their responsibilities.
- It can be prevented too many ways:
  - Apply controls as per the classification.
  - Don't store sensitive data unnecessarily. Discard it as soon as possible or use PCI DSS compliant tokenization or even truncation. Data that is not retained cannot be stolen.
  - Make sure to encrypt all sensitive data at rest.
- Data is the most expensive component between Information system components, so even if the complexity isn't low the attacker will look for any kind of data so this Vulnerability have high probability.
- Screenshot:



You successfully solved a challenge: Score Board (Find the carefully hidden 'Score Board' page.)

You successfully solved a challenge: Confidential Document (Access a confidential document.)

About Us

Corporate History & Policy

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. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Check out our boring terms of use if you are interested in such lame stuff. 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. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, At accusam aliquyam diam diam dolore dolores duo eirmod eos erat, et nonumy sed tempor et et invidunt justo labore Stet clita ea et gubergren, kasd magna no rebum.

Customer Feedback

Security Misconfiguration Detail

The application might be vulnerable if the application is:

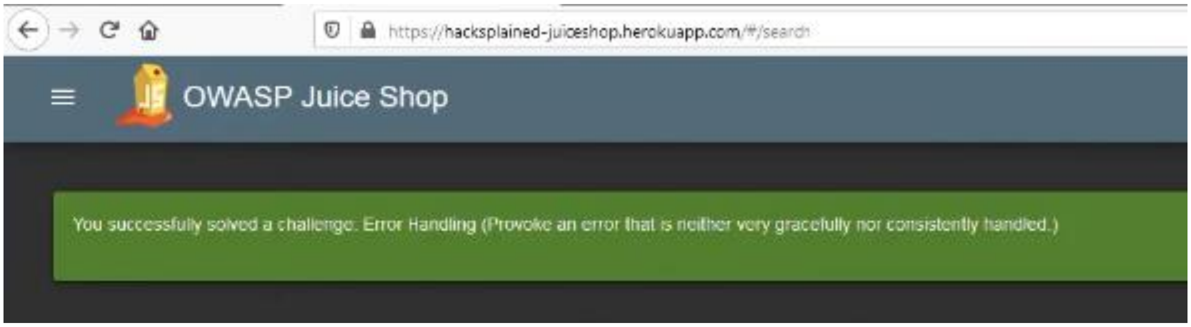
- Missing appropriate security hardening across any part of the application stack or improperly configured permissions on cloud services.
- Unnecessary features are enabled or installed (e.g., unnecessary ports, services, pages, accounts, or privileges).
- Default accounts and their passwords are still enabled and unchanged.

Security Misconfiguration

RISK LEVEL MEDIUM

Vulnerability detail

- Error handling is so weak so I could fake a http request and still having response.
- This vulnerability's attack complexity is low. So it has high probability of exploit.
- The organization, Because it's relative to the developers team.
- Secure installation processes should be implemented, including:
  - A repeatable hardening process makes it fast and easy to deploy another environment that is appropriately locked down.
  - A minimal platform without any unnecessary features, components, documentation, and samples. Remove or do not install unused features and frameworks.
- This vulnerability include multiple levels sometimes it lead to high losses and sometimes it's just show how bad the web-page is developed, in our case it's just an error handling misconfigured so it has low risk and no asset loss, so it has low probability.
- Screenshot:



https://hacksplained-juiceshop.herokuapp.com/#/search

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

Methodology

In this Vulnerability assessment report I started working and scanning for vulnerabilities manually. I did found some of them manually.

Then after days of scanning manually I switched to use automated scans using some tools. Such as Nmap which scans the network that a computer is connected to and outputs a list of ports, device names, operating systems, and several other identifiers that help the user understand the details behind their connection status. It also give you a huge list of scripts to use. You can specify a service used in the web page and look for vulnerabilities in that service.

Burpsuite which allow you to have a clear vision on the http request and response. It allow you to modify the request or reading unreadable additional information in the response.

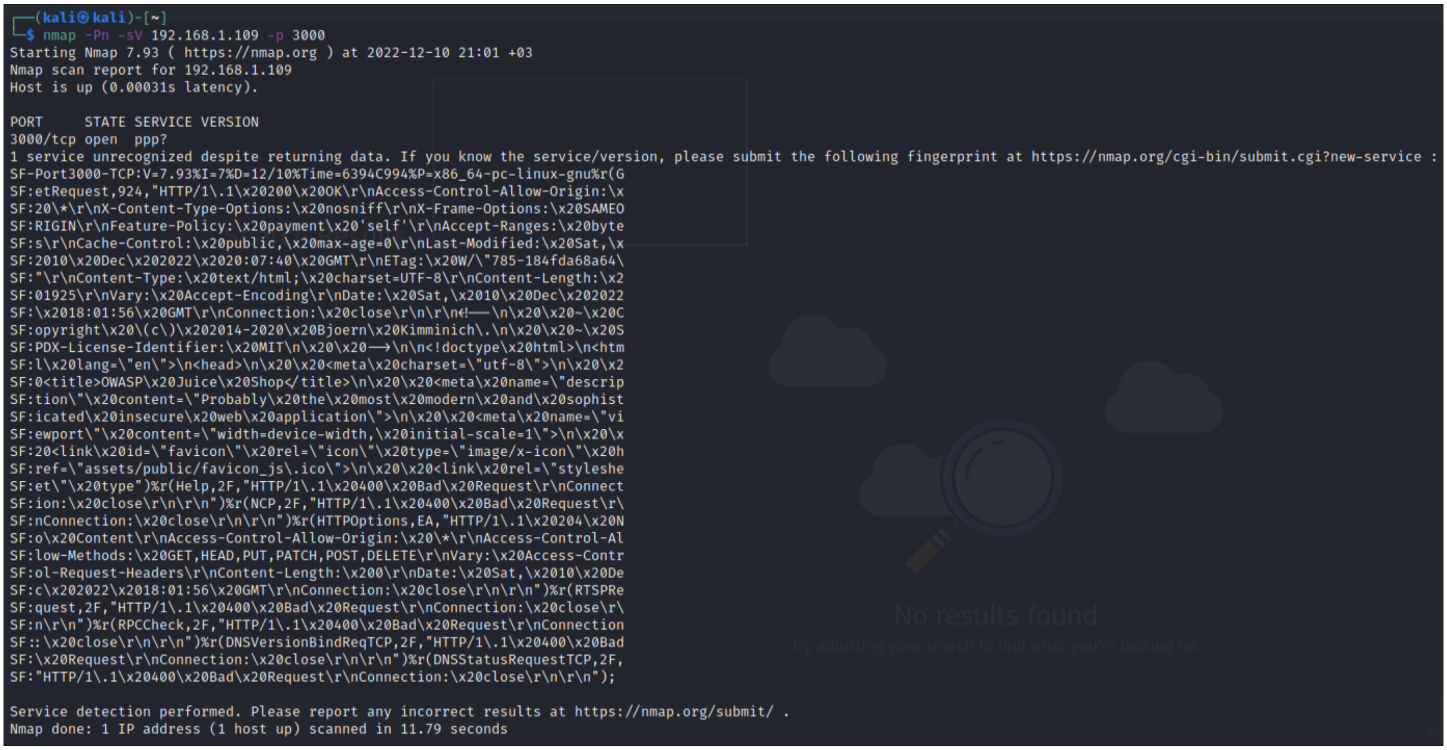
Assessment Toolset Selection

Tools used during the execution phase:

- Nmap.
- Burpsuite.

Assessment Methodology Detail

Using Nmap I couldn't have enough information or instructions to follow. Using the command below that was the output I got.



```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn -sV 192.168.1.109 -p 3000
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-10 21:01 +03
Nmap scan report for 192.168.1.109
Host is up (0.00031s latency).

PORT     STATE SERVICE VERSION
3000/tcp open  ppp?
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3000-TCP:V=7.93%I=7%D=12/10%Time=6394C994%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,924,"HTTP/1\.1\x20200\x20OK\r\nAccess-Control-Allow-Origin:\x
SF:20\*\r\nX-Content-Type-Options:\x20nosniff\r\nX-Frame-Options:\x20SAMEO
SF:RIGIN\r\nFeature-Policy:\x20payment\x20'self'\r\nAccept-Ranges:\x20byte
SF:s\r\nCache-Control:\x20public,\x20max-age=0\r\nLast-Modified:\x20Sat,\x
SF:2010\x20Dec\x202022\x2020:07:40\x20GMT\r\nETag:\x20W/\"785-184fda68a64\
SF:"\r\nContent-Type:\x20text/html;\x20charset=UTF-8\r\nContent-Length:\x2
SF:01925\r\nVary:\x20Accept-Encoding\r\nDate:\x20Sat,\x2010\x20Dec\x202022
SF:\x2018:01:56\x20GMT\r\nConnection:\x20close\r\n\r\n<!——\n\x20\x20~\x20C
SF:opyright\x20\(c\)\x202014-2020\x20Bjoern\x20Kimminich\.\n\x20\x20~\x20S
SF:PDX-License-Identifier:\x20MIT\n\x20\x20——>\n\n<!doctype\x20html>\n<htm
SF:l\x20lang=\"en\">\n<head>\n\x20\x20<meta\x20charset=\"utf-8\">\n\x20\x2
SF:0<title>OWASP\x20Juice\x20Shop</title>\n\x20\x20<meta\x20name=\"descrip
SF:tion\"\x20content=\"Probably\x20the\x20most\x20modern\x20and\x20sophist
SF:icated\x20insecure\x20web\x20application\">\n\x20\x20<meta\x20name=\"vi
SF:ewport\"\x20content=\"width=device-width,\x20initial-scale=1\">\n\x20\x
SF:20<link\x20id=\"favicon\"\x20rel=\"icon\"\x20type=\"image/x-icon\"\x20h
SF:ref=\"assets/public/favicon_js\.ico\">\n\x20\x20<link\x20rel=\"styleshe
SF:et\"\x20type")%r(Help,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnect
SF:ion:\x20close\r\n\r\n")%r(NCP,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\
SF:nConnection:\x20close\r\n\r\n")%r(HTTPOptions,EA,"HTTP/1\.1\x20204\x20N
SF:o\x20Content\r\nAccess-Control-Allow-Origin:\x20\*\r\nAccess-Control-Al
SF:low-Methods:\x20GET,HEAD,PUT,PATCH,POST,DELETE\r\nVary:\x20Access-Contr
SF:ol-Request-Headers\r\nContent-Length:\x200\r\nDate:\x20Sat,\x2010\x20De
SF:c\x202022\x2018:01:56\x20GMT\r\nConnection:\x20close\r\n\r\n")%r(RTSPRe
SF:quest,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection:\x20close\r\
SF:n\r\n")%r(RPCCheck,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection
SF::\x20close\r\n\r\n")%r(DNSVersionBindReqTCP,2F,"HTTP/1\.1\x20400\x20Bad
SF:\x20Request\r\nConnection:\x20close\r\n\r\n")%r(DNSStatusRequestTCP,2F,
SF:"HTTP/1\.1\x20400\x20Bad\x20Request\r\nConnection:\x20close\r\n\r\n");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.79 seconds
```

This concluded the vulnerability assessment methodology portion of this report.