Wrocław 01.10.2022

C Y B E R S E C U R I T Y

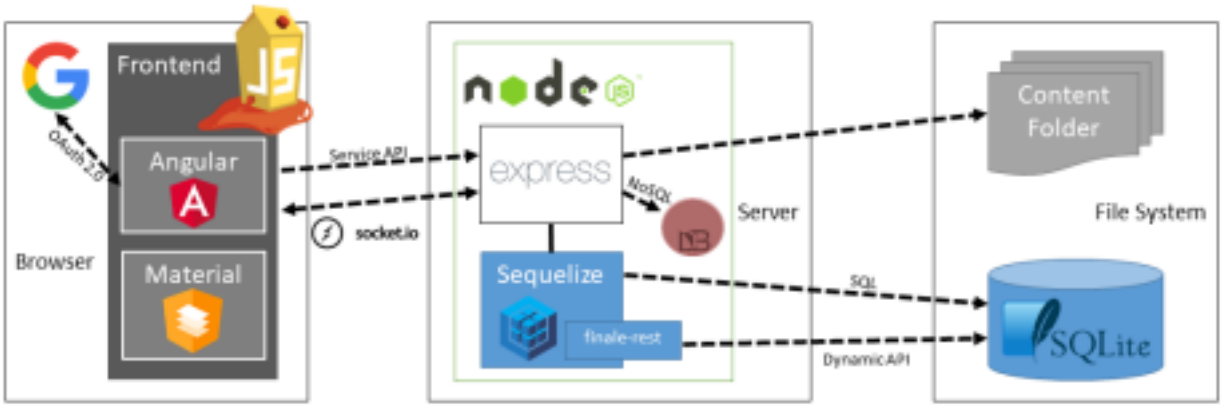
LAB 1 3

1. Introduction

The tools presented during this lab mainly focus on the front-end security of web infrastructure. They can be used to identify, analyze, and exploit a wide range of application security vulnerabilities. These include cross-site scripting (XSS), SQL injection, SSI injection, XML injection, application misconfiguration, abuse of functionality, session prediction, information disclosure, and many other attacks and weaknesses. There are various standards to classify these application vulnerabilities, which have been previously discussed in the Vulnerability taxonomy section. In order to understand the nuts and bolts of these vulnerabilities, we strongly recommend you to go through these standards.

OWASP (Open Web Application Security Project) is a non-profit foundation that works for improving the security of software. Thanks to the community's open source software projects, hundreds of local branches around the world, tens of thousands of members and leading education and training conferences is a great source of learning on software security.

OWASP develops many projects related to increasing knowledge about security and creates standards, e.g. for application security testing. One of such applications is JuiceShop, an application which is an example of a modern online store written using commercially used technologies and frameworks. JuiceShop is written in such a way that each of its elements is vulnerable to a number of attacks, which will allow users to develop skills of knowledge, use and prevention of attacks on web applications.



Burp Suite

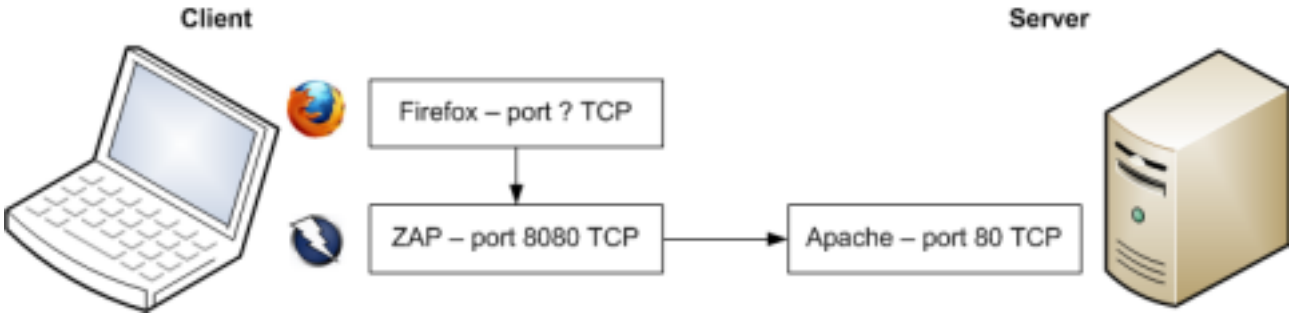
Burp Suite is a combination of powerful web application security tools. These tools demonstrate the real-world capabilities of an attacker penetrating web applications. They can scan, analyze, and exploit web applications using manual and automated techniques. The integration facility between the interfaces of these tools provides a complete attack platform to share information between one or more tools. This makes the Burp Suite a very effective and easy-to-use web application attack framework.

Zed Attack Proxy (OWASP ZAP)

The OWASP Zed Attack Proxy (ZAP) is one of the world's most popular free security tools and is actively maintained by hundreds of international volunteers. It can help you automatically find security vulnerabilities in your web applications while you are developing and testing your applications. It is also a great tool for experienced pentesters

to use for manual security testing.

The idea of functioning of applications such as Burp and Zap is presented in the picture below. To be sure to use the possibilities offered by these tools, you need to configure them as a proxy server for your browser. In this case all requests leaving the browser are captured by Burp/Zap. You could say that at this point it is a kind of MitM attack where the user of these tools is both attacker and attacker. The benefits of this are the possibility to change the requests sent to the servers after leaving the browser and thus bypass all validation mechanisms running on the client side.



The aim of this laboratory is to familiarize with the basic functionalities of OWASP ZAP, ways of testing web applications and vulnerabilities like:

- **Broken authentication**: Vulnerability group including security issues related to user authentication in the application. This vulnerability group includes many different vulnerabilities: logging data transfer via an unsecured transmission channel, lack of confirmation of registration/resetting password via e-mail, lack of policy on us ing strong passwords, inconsistencies in validating login data on server and client side, etc.
- **Broken Access Control:** Vulnerabilities related to errors in determining access to server resources, often associated with authorization. Often using methods such as value enumer ation it is possible to gain unauthorized access to data even without knowing the specific values of parameters (such as object id in the database). Similarly, it is possible to access elements of web services that have not been properly secured by checking user permis sions, but only deleted links to days at the application frontend level.
- **Improper Input Validation:** Incorrect input checking or unchecked input data is a type of vulnerability in the security of computer software that can be used for various types of attacks. When the software does not check the input data correctly, the attacker is able to produce the input data in a form that the rest of the application does not expect. This will cause parts of the system to receive unintentional input data, which can result in a changed control flow, any control the resource or execution of any code.
- **Security Misconfiguration:** Incorrect security configuration is the most common problem. This is usually the result of uncertain default configurations, incomplete or temporary con figurations, open cloud storage, poorly configured HTTP headers and full error messages containing confidential information. Not only must all operating systems, frameworks, li braries and applications be securely configured, but they must be patched/updated in a timely manner.

2. Required virtual machines

- Kali

L A B 1 3

3. Prerequisites

Get familiar with the following elements:

- OWASP-ZAP
- Firefox proxy settings

4. Problems and questions

I. Using an Internet search, what are X-Content-Type-Options and X-Frame-Options headers, and how might the protect your site? Using an Internet search, what are X-frame-option headers? What are typical applications of SQL Injection attacks (what actions can an attacker perform)?

> X-Content-Type-Options and X-Frame-Options headers are used to prevent certain types of attacks on web applications. The X-Content-Type-Options header is used to prevent content type sniffing, while the X-Frame-Options header is used to prevent clickjacking.
SQL Injection attacks are a type of attack in which an attacker can execute arbitrary SQL commands on a database, which can be used to steal data, modify data, or execute commands on the underlying system.

II. Why is it important to keep the data validation rule on both client and server sides?

> It is important to keep the data validation rule on both client and server sides because it ensures that the data is valid and secure, even if an attacker is able to bypass the client-side validation. This helps to protect the server and the data.

III. How to restrict access to resources to which the user has no permission?

> One way to restrict access to resources to which the user has no permission is to use access control lists or role-based access control to define which users are allowed to access which resources.

5. Tasks

I. Open terminal, go to the JuiceShop folder (on the Desktop) and execute:

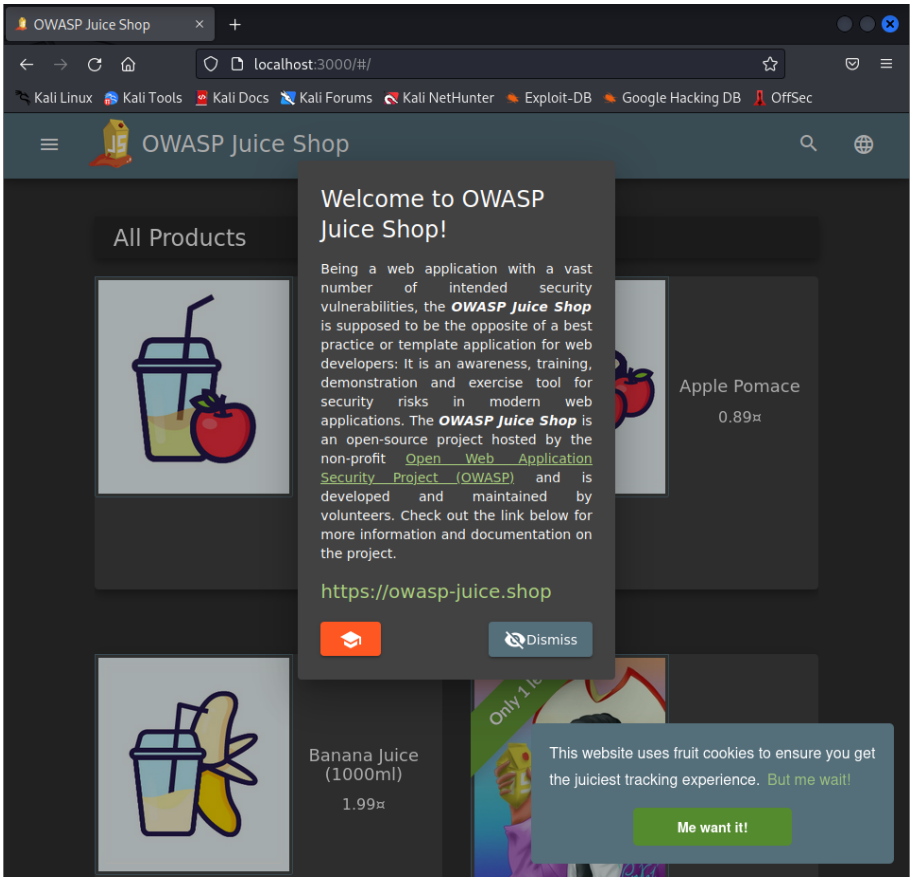
(before for first time*;

npm start

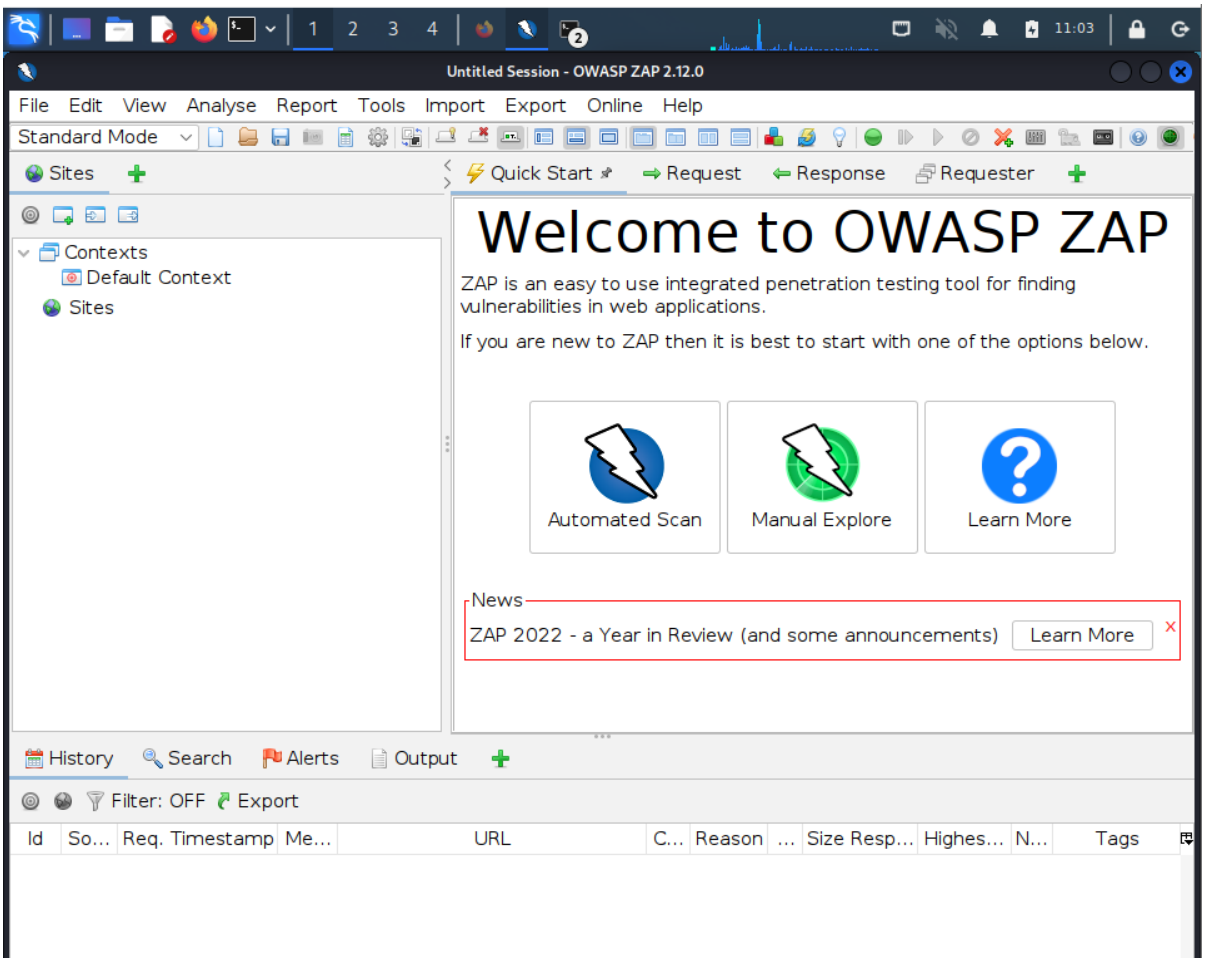
```
┌──(kali㉿kali)-[~/juice-shop]
└─$ npm start

> juice-shop@14.4.0 start
> node build/app

info: All dependencies in ./package.json are satisfied (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Detected Node.js version v18.13.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Server listening on port 3000
```

II. Open browser and use address: **localhost:3000**

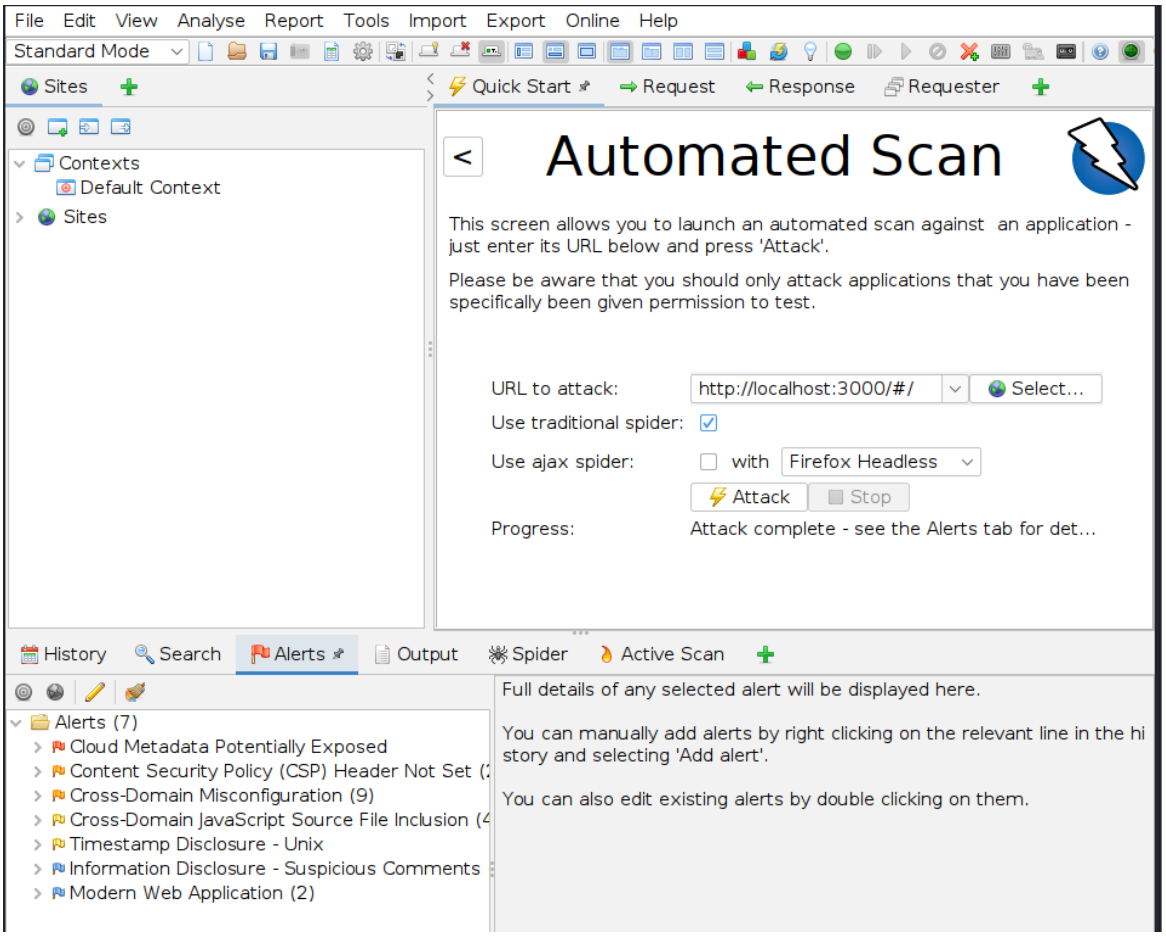
Please remember that JuiceShop with default configuration restores the database to its initial state every time it is started!
Security Misconfiguration



III. Open OWASP ZAP



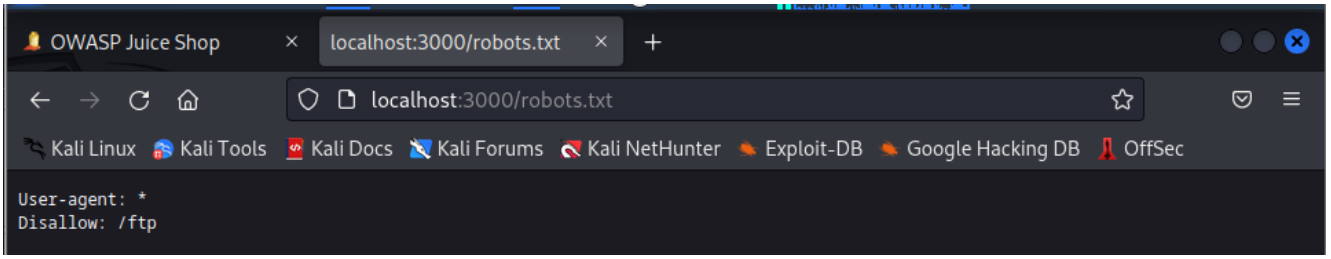
IV. Scan for the JuiceShop site.



After the scan answer the questions:

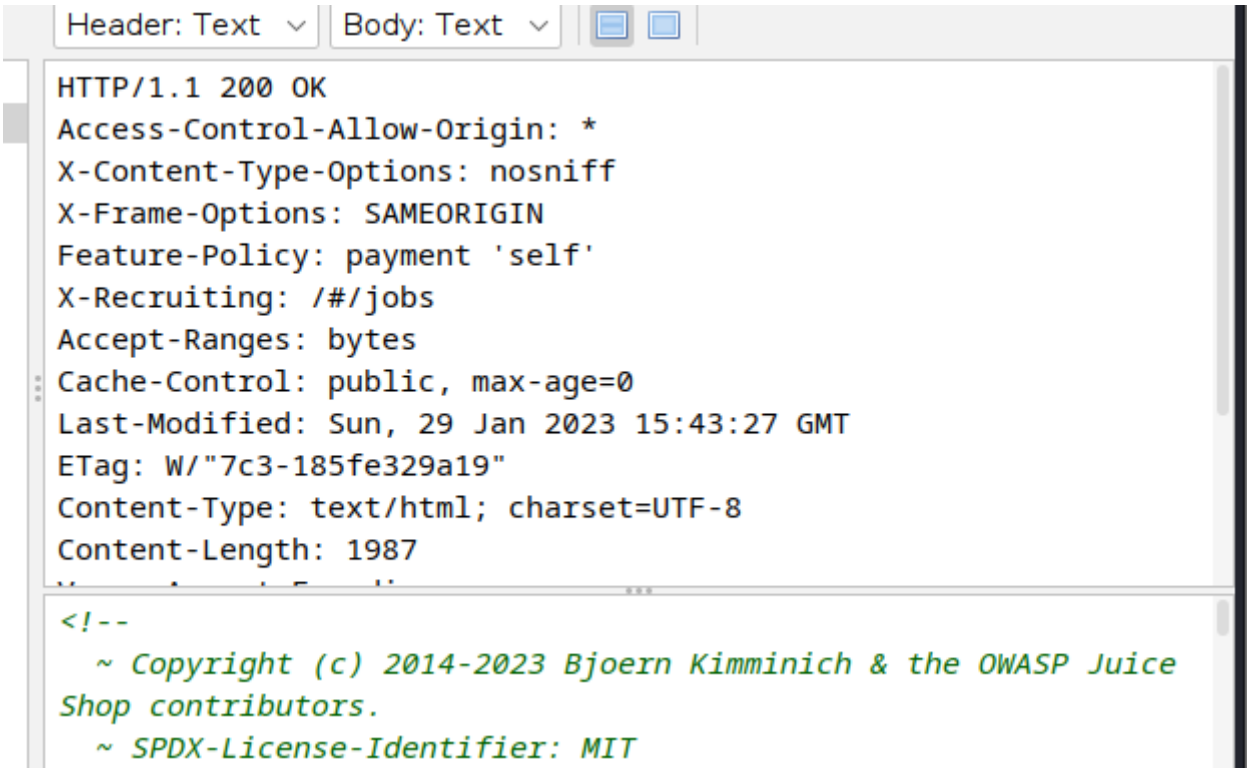
How many flags did it raise for:Red flag, Amber flag,Yellow flag,Blue flags. What directories has it exposed that can be browsed?

Can you browse them (try with a web browser)?



>They can be browsed !

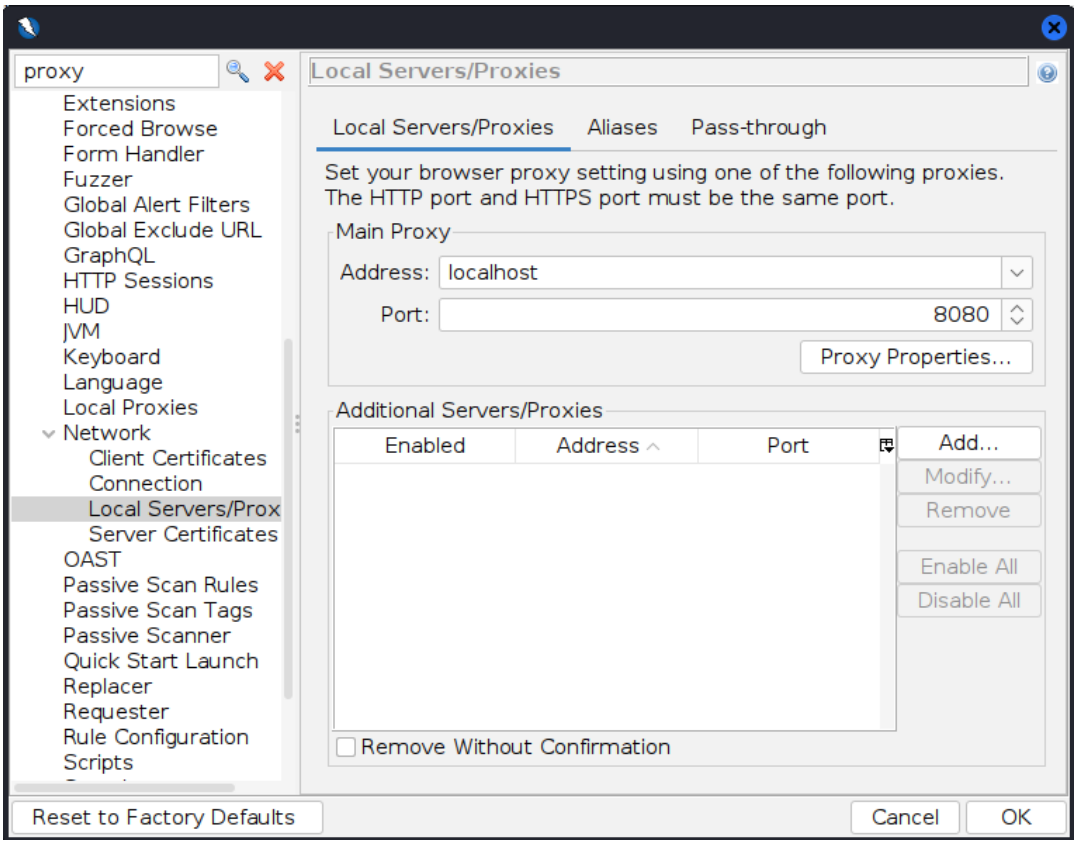
What cookie information risk has been discovered?



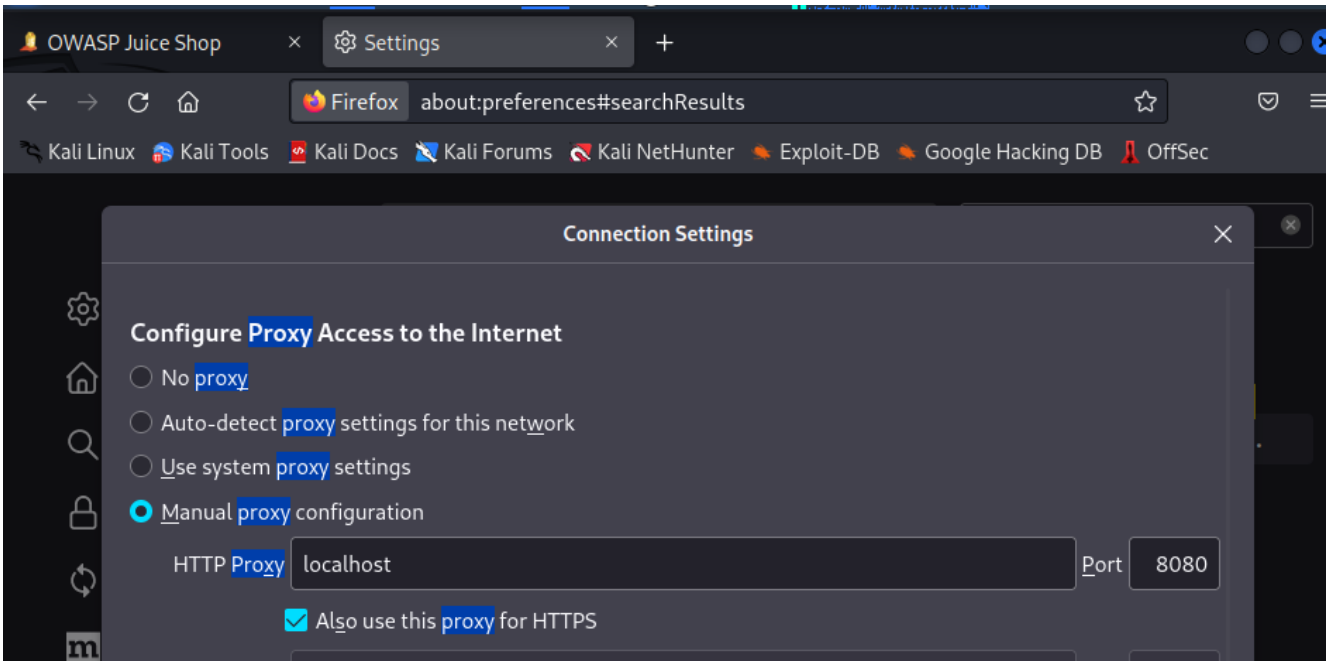
>

Improper Input Validation

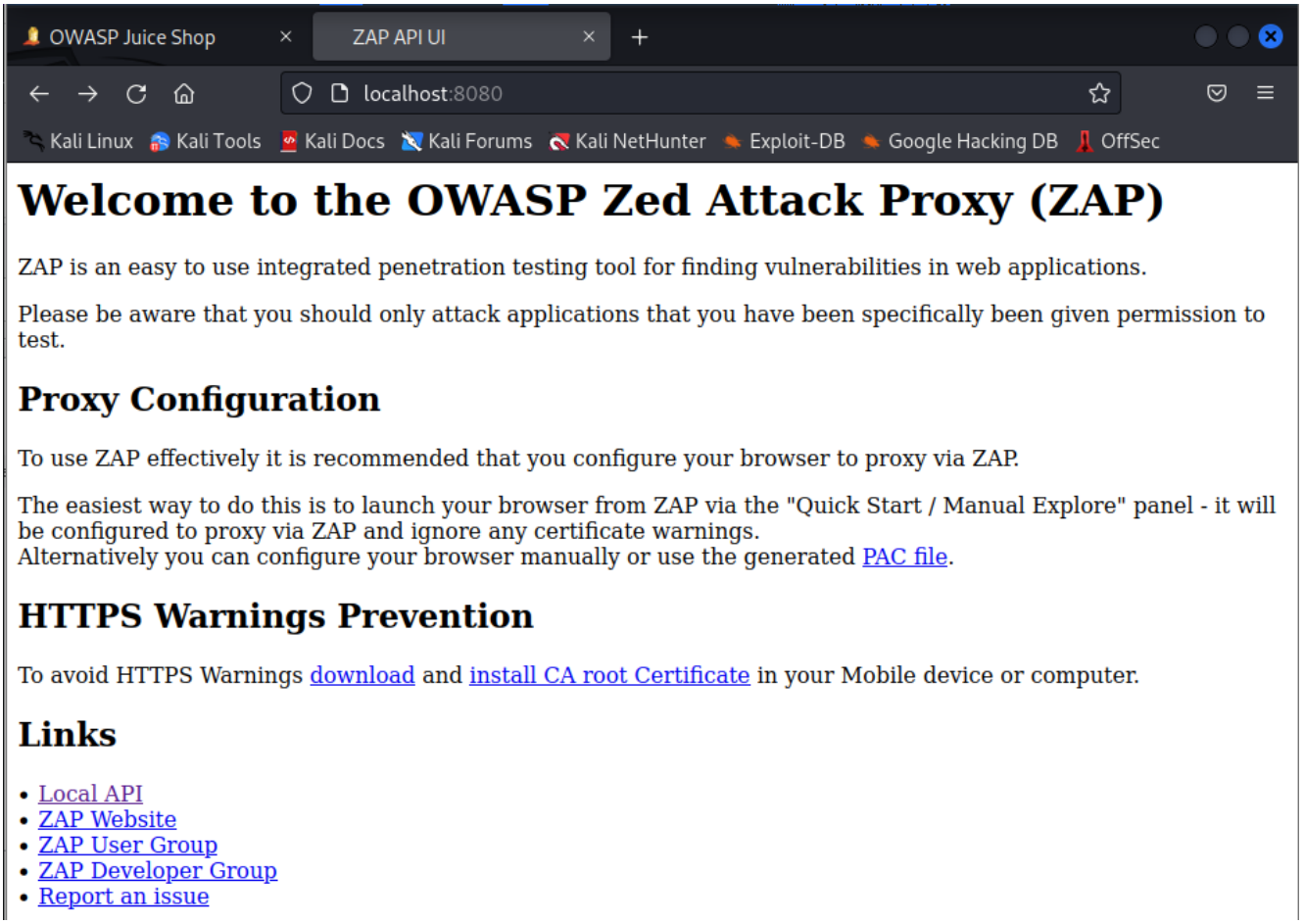
V. Check proxy settings in ZAP (Options -> Write “Proxy” in Search Field)



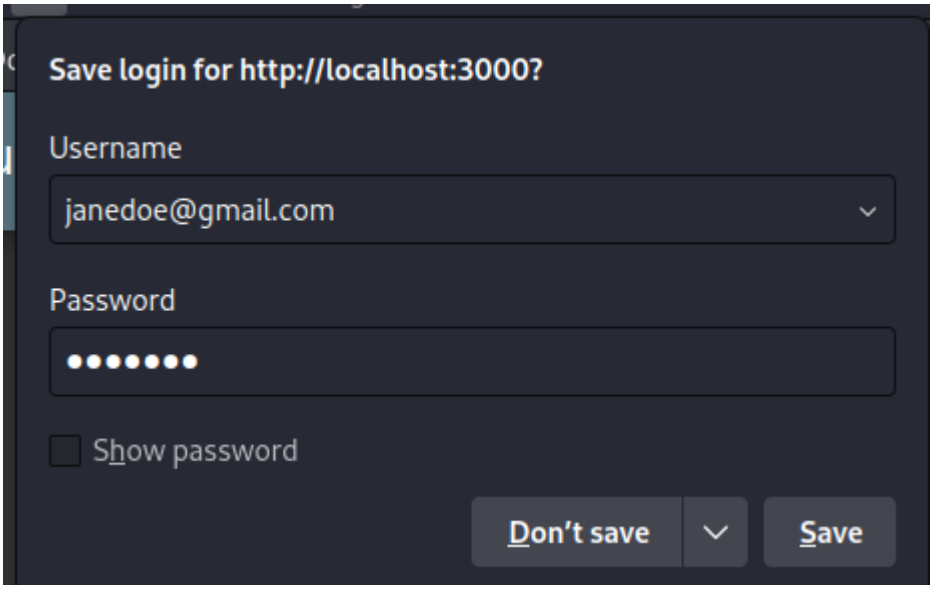
VI. Configure the Firefox in Kali VM (Preferences -> Write “Proxy” in Search Field) - Provide the same data as in ZAP



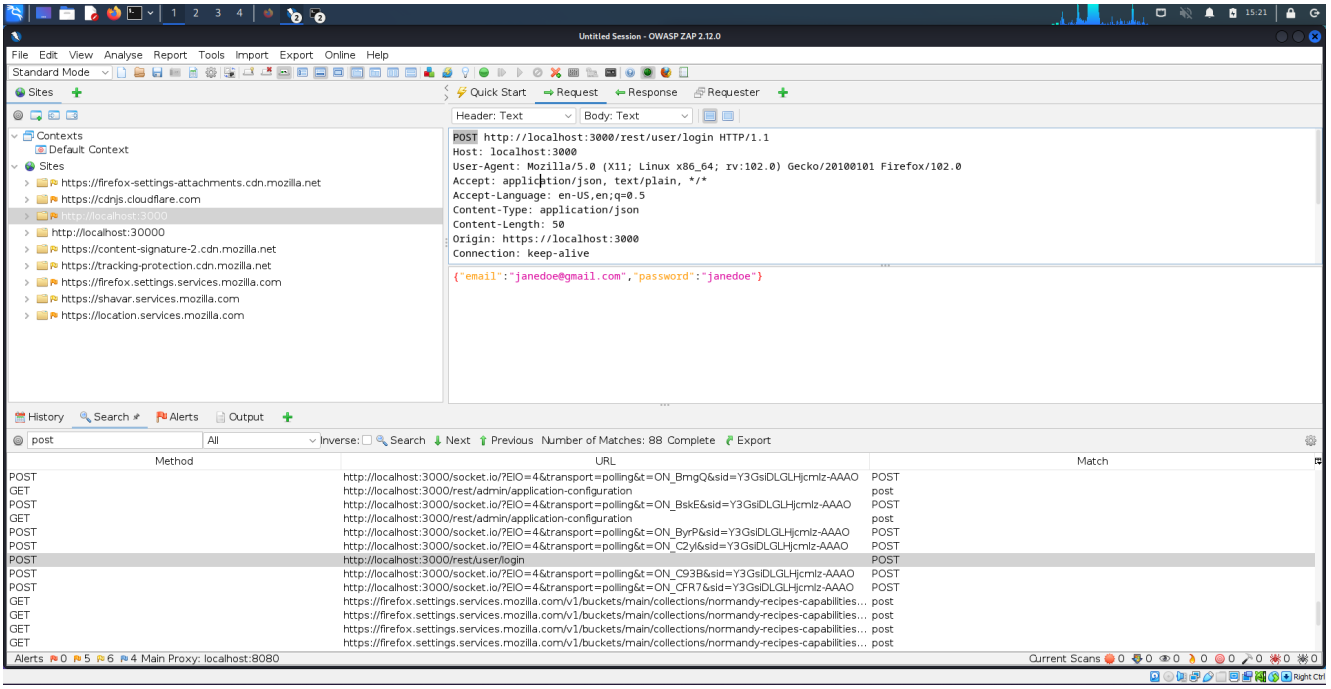
VII. Open JuiceShop main page and check if it presents in ZAP history **Alternatively, you can start a session for ZAP in Firefox by clicking on the Firefox icon in the ZAP interface**



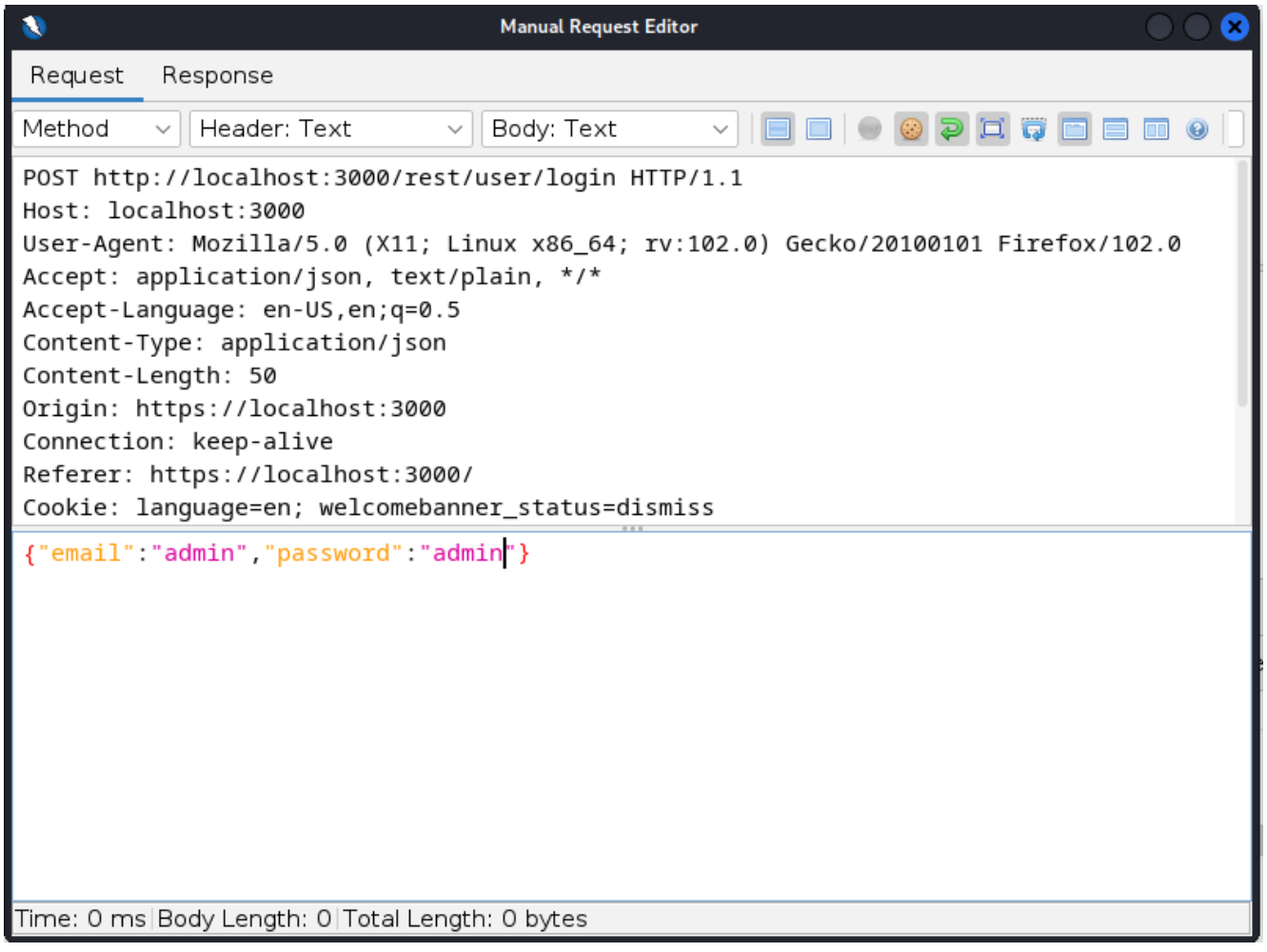
VIII. Go to juiceshop.domain/#/register , register new user and select “Mother’s birth date? (MM/DD/YY)’ as Security question (provide any valid date)



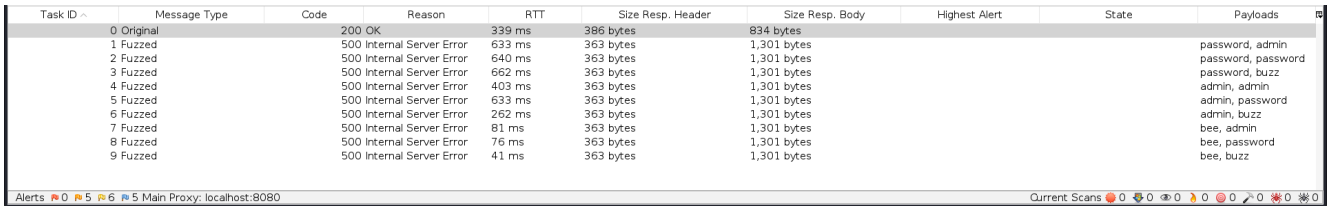
IX. Check the request and response for user registration (is there any parameter related to admin?)



X. Select registration request and send it to the Request Editor (Right mouse click -> Open/Resend with Request Editor)



XI. Modify the request (remember to change email) to register new user as admin (set role parameter)



Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200	OK	339 ms	386 bytes	834 bytes			
1	Fuzzed	500	Internal Server Error	633 ms	363 bytes	1,301 bytes			password, admin
2	Fuzzed	500	Internal Server Error	640 ms	363 bytes	1,301 bytes			password, password
3	Fuzzed	500	Internal Server Error	662 ms	363 bytes	1,301 bytes			password, buzz
4	Fuzzed	500	Internal Server Error	403 ms	363 bytes	1,301 bytes			admin, admin
5	Fuzzed	500	Internal Server Error	633 ms	363 bytes	1,301 bytes			admin, password
6	Fuzzed	500	Internal Server Error	262 ms	363 bytes	1,301 bytes			admin, buzz
7	Fuzzed	500	Internal Server Error	81 ms	363 bytes	1,301 bytes			bee, admin
8	Fuzzed	500	Internal Server Error	76 ms	363 bytes	1,301 bytes			bee, password
9	Fuzzed	500	Internal Server Error	41 ms	363 bytes	1,301 bytes			bee, buzz

Alerts 0 5 6 5 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0 0 0

> it did not fuzz :(

XII. Execute in terminal:

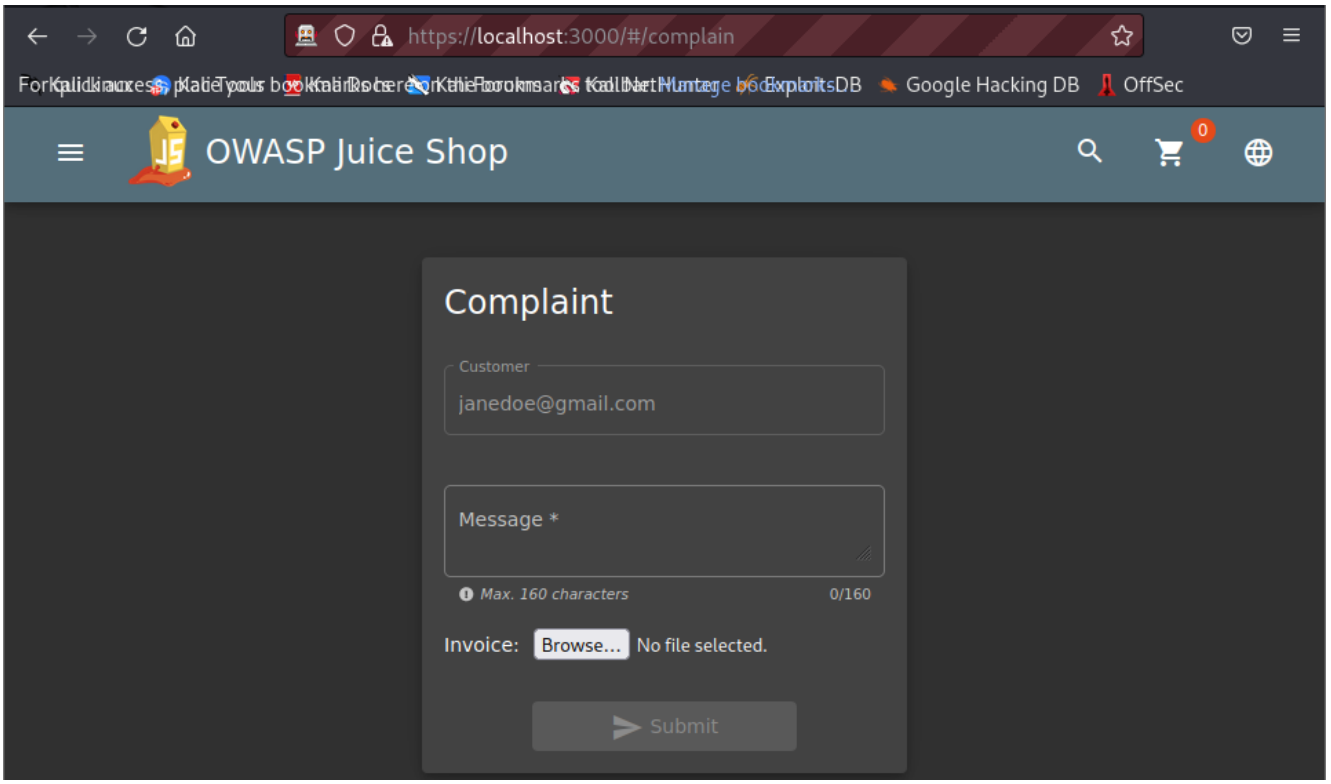
- **sudo apt install snapd (restart virtual machine after this step)**
- **systemctl enable --now snapd apparmor**
- **snap install postman**

```
┌──(kali㉿kali)-[~]
└─$ snap install postman
Download snap "snapd" (17950) from channel "stable"       60% 2.94MB/s 7.05s
```

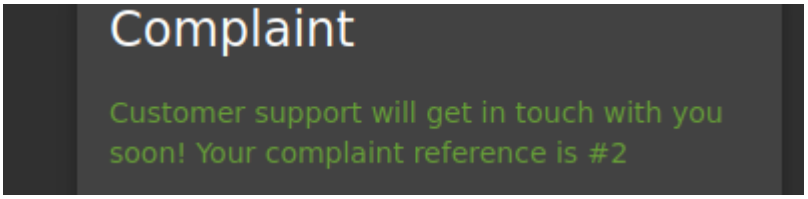
- **postman**

```
┌──(kali㉿kali)-[/snap/bin]
└─$ postman
Command 'postman' is available in '/snap/bin/postman'
The command could not be located because '/snap/bin' is not included in the P
ATH environment variable.
postman: command not found
```

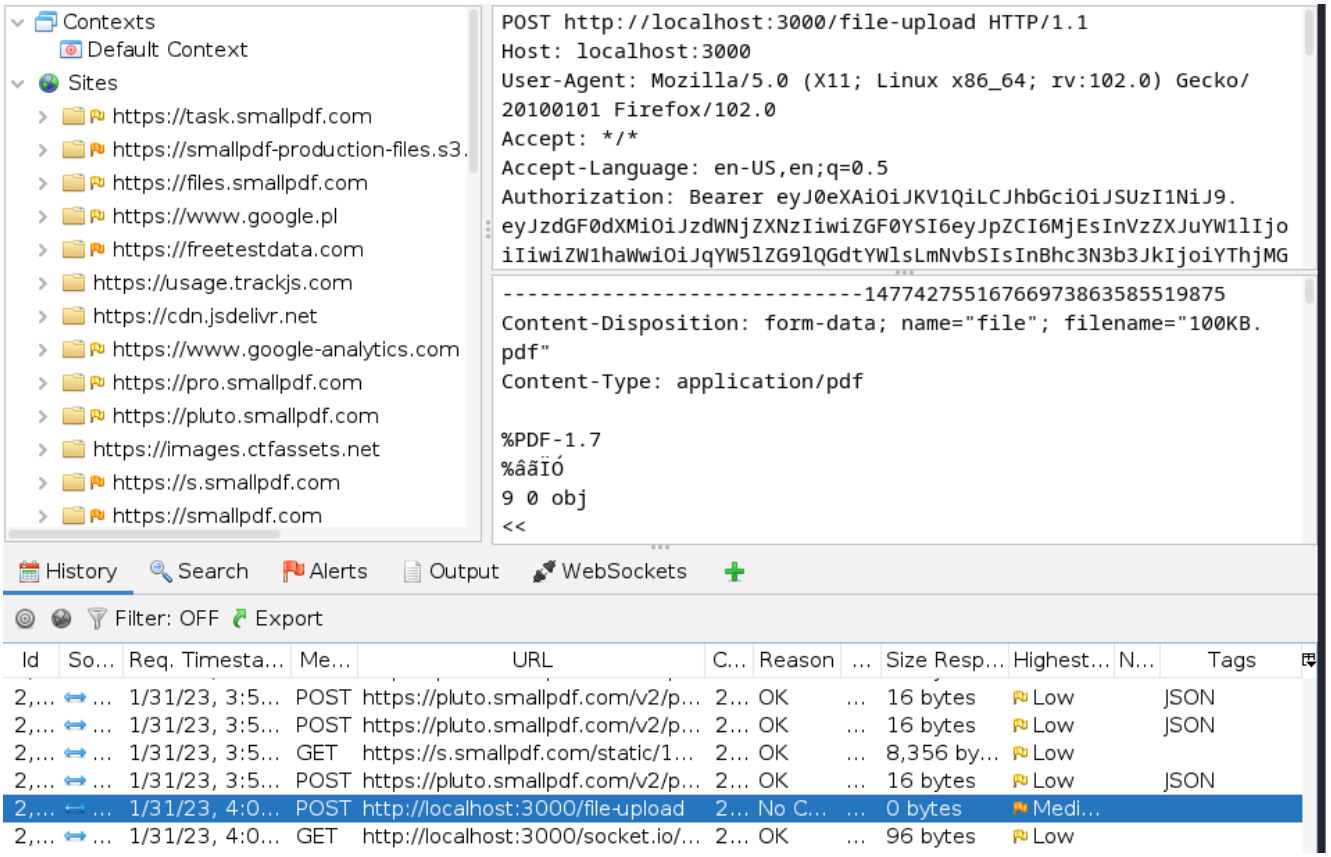
XIII. Open **Complaint** subpage in browser



XIV. Try to upload a few .pdf files (<100kB, 100-200kB, >200kB)



XV. Catch the request for uploading <100kB pdf file in ZAP



XVI. Change the content of the uploaded file in Request Editor to the content of the 100-200kB file. You have to open pdf in text editor, copy the content and exchange it with the data between **%PDF** and **%%EOF** flags, Alternatively, you can use Postman to create a similar request:

- Set method type to POST
- Body -> form-data -> Enter your parameter name **'file'**
- Change parameter type to file and select a file which you want to upload

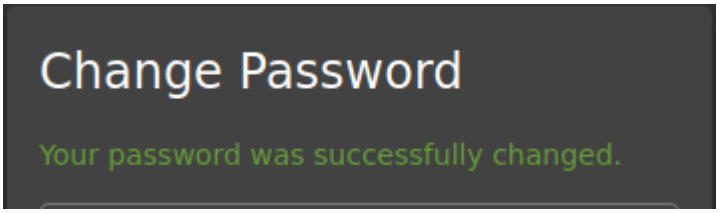
> did not understand the task

XVII. Repeat the procedure with >200kB content. What are the conclusions?

>> **FILE TOO LARGE**

Broken Authentication

XVIII. Try to change the password. JuiceShop requires a security password when changing your password, but no email confirmation is required. Provide **INVALID** answer for security questions.

> 

XIX. Find the request in ZAP and use option **Fuzz…**

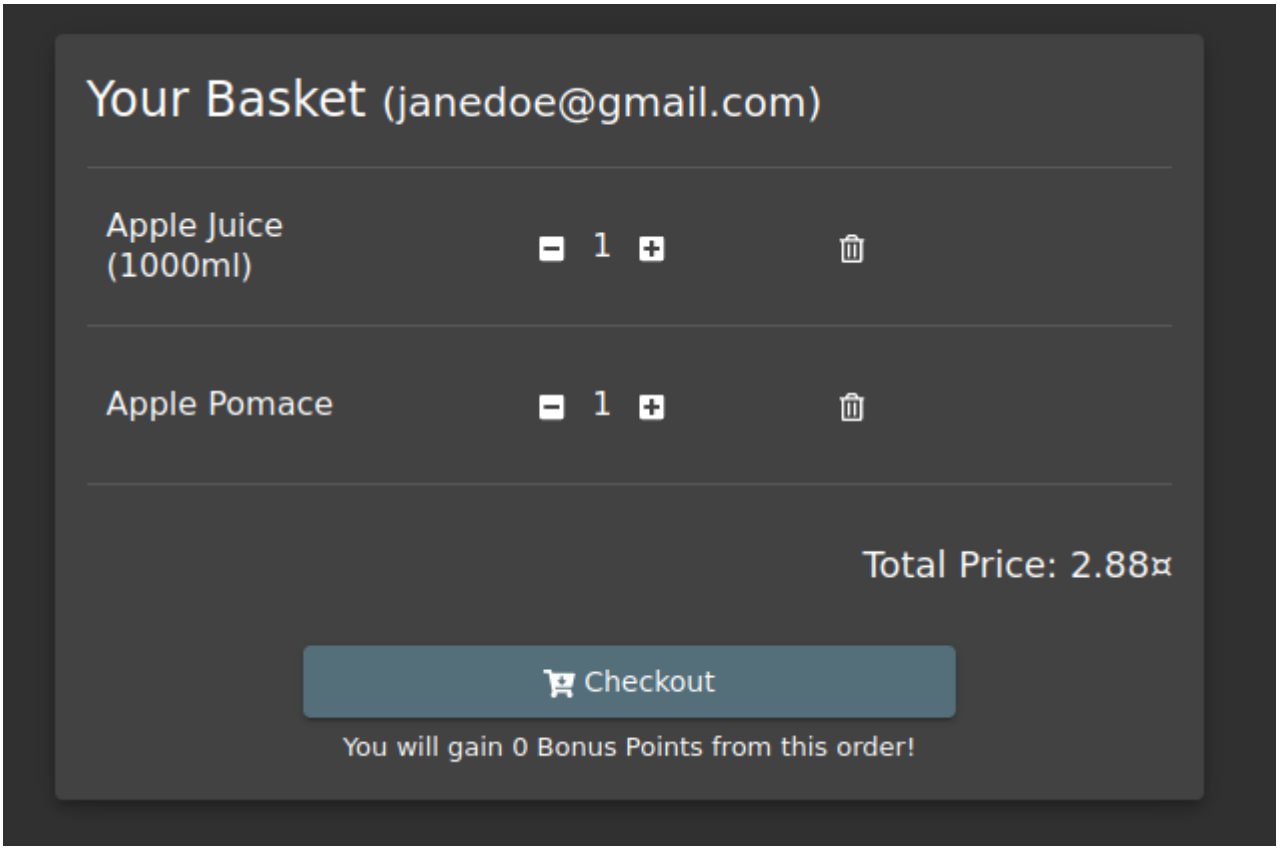
> fuzzing did not work as for task XI i followed the task directions from pdf and naturally i had to take help from internet, i used Fuzz to solve that task which did not work.

XX. Select parts of date from answer and prepare rules for fuzzing (Add… -> Add -> Numberzz -> provide valid values for days, months, and years, **limit values to a reasonable range, because JuiceShop is blocking too many requests**)

Broken Access Control

XXI. Add any products to your basket and open basket page

>



XXII. Check the REST request in ZAP, it should contain the id of your basket

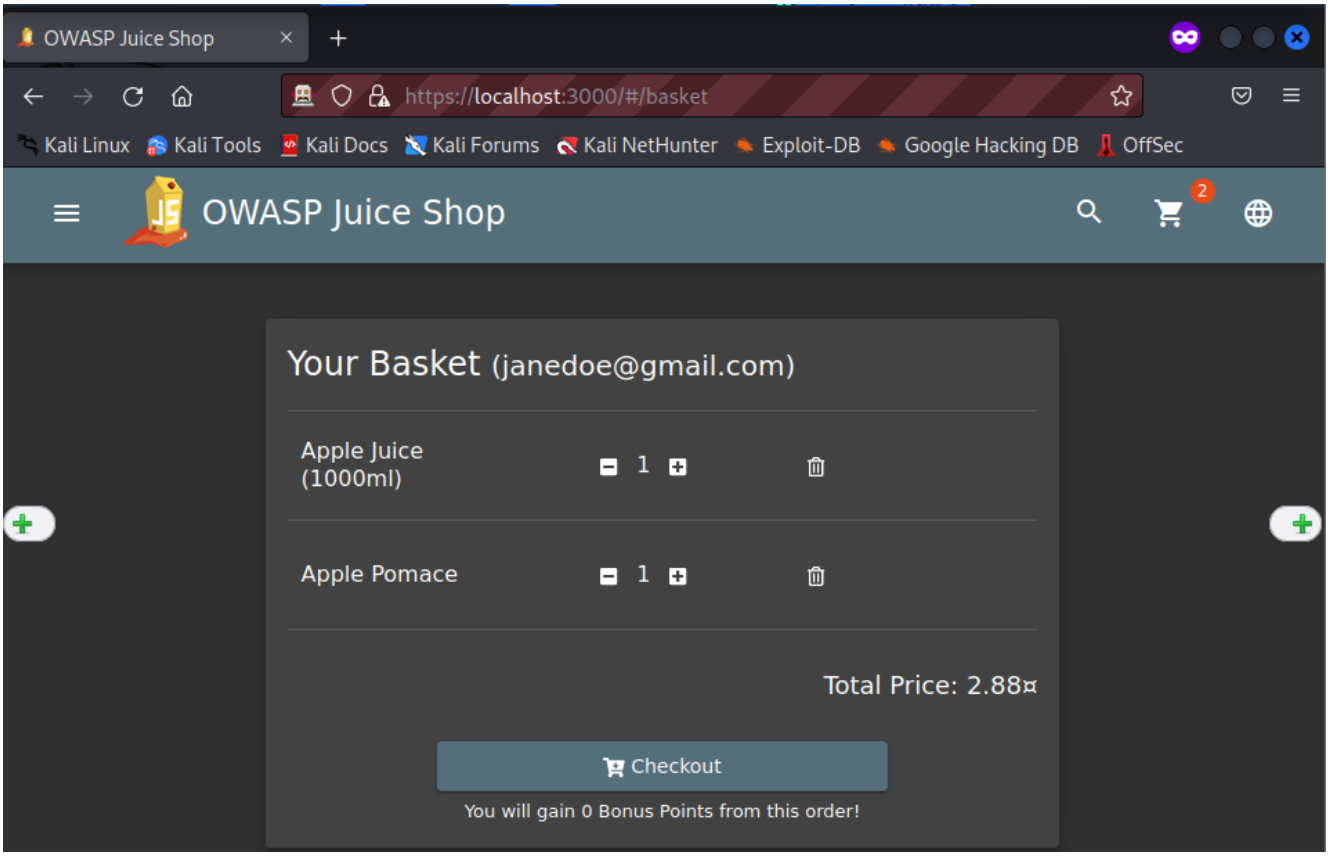
>

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 1013
ETag: W/"3f5-G3KQCC5x1JKhE214zSH4Z4rVtlY"
```


```
{"status":"success","data":{"id":6,"coupon":null,"UserId":21,
"createdAt":"2023-01-31T08:38:56.563Z","updatedAt":
"2023-01-31T08:38:56.563Z","Products":[{"id":1,"name":
"Apple Juice (1000ml)","description":"The all-time classic.",
"price":1.99,"deluxePrice":0.99,"image":"apple_juice.jpg",
"createdAt":"2023-01-29T22:45:22.180Z","updatedAt":
"2023-01-29T22:45:22.180Z","deletedAt":null,"BasketItem":{
"ProductId":1,"BasketId":6,"id":9,"quantity":1,"createdAt":
"2023-01-31T09:20:25.356Z","updatedAt":
```


XXIII. Open JuiceShop in private browser window (or in any other browser), log in as another user and add products to basket

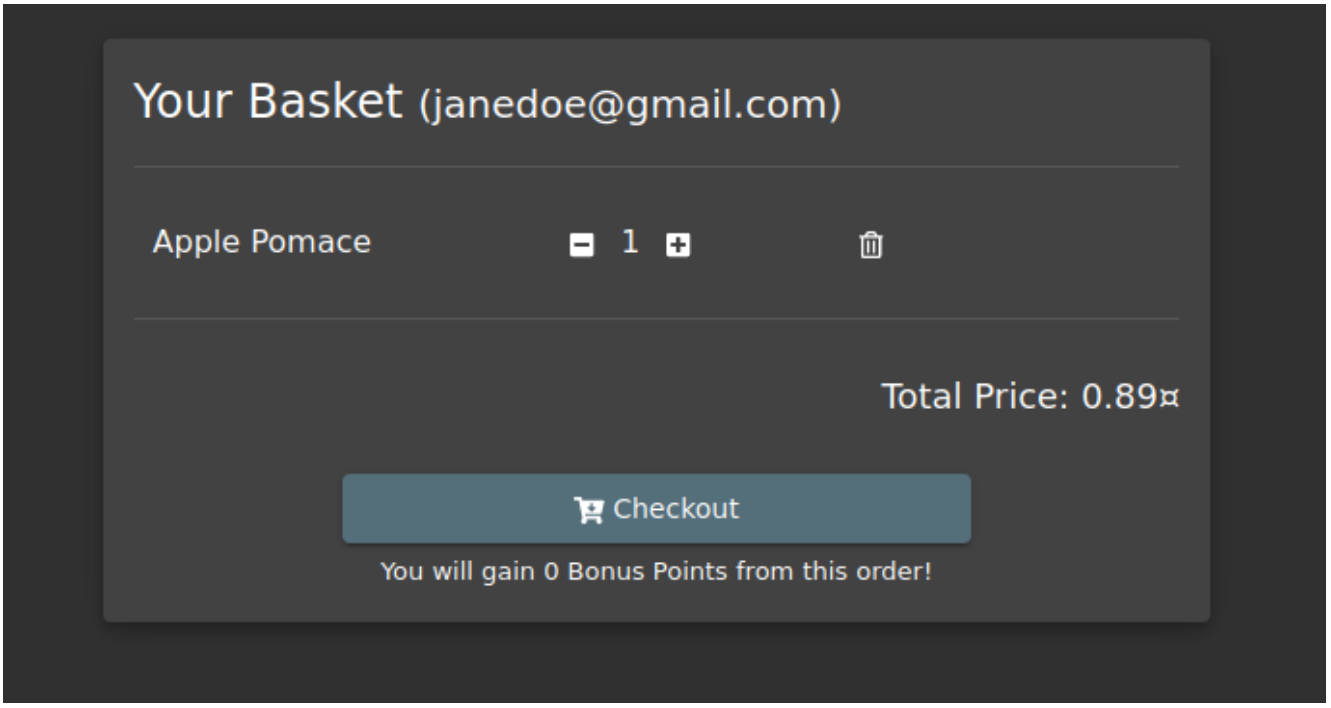
>



** another incognito tab

XXIV. Edit request in Request Editor and change id of basket

>



XXV. Check if you have an access to the basket as original use


```
{"status":"success","data":{"id":6,"coupon":null,"UserId":21,
"createdAt":"2023-01-31T08:38:56.563Z","updatedAt":
"2023-01-31T08:38:56.563Z","Products":[{"id":24,"name":
"Apple Pomace","description":
"Finest pressings of apples. Allergy disclaimer: Might contain
traces of worms. Can be <a href=\"/#recycle\">sent back to us</
a> for recycling.","price":0.89,"deluxePrice":0.89,"image":
"apple_pressings.jpg","createdAt":"2023-01-29T22:45:22.181Z",
"updatedAt":"2023-01-29T22:45:22.181Z","deletedAt":null,
```


>

It changed!