



## **NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY**

### **DEPARTMENT OF COMPUTER SCIENCE**

#### **INFORMATION SECURITY LAB**

<b>NAME</b>	Ayesha Imran
<b>Class</b>	CS-A
<b>Lab</b>	08
<b>Course</b>	Information Security
<b>Date</b>	3-November-25
<b>Submitted To</b>	Lec. Attiya Ashraf

## IN LAB TASKS

### Pre-requisites

1. Basic Knowledge of SQL, HTTP protocols, and web application structure.
2. Node.js v18: Required for Juice Shop setup.
3. Juice Shop: A vulnerable web application for security training.
4. OWASP ZAP (Zaproxy): A tool for performing security scans on web applications.

### Part 1: Installation

Installing Node.js and NPM

Node.js and NPM are pre-requisites.

1. As always, start with updating package lists.
2. To ensure you install Node.js v18, add the NodeSource repository First install Curl

**Command:** `sudo apt install curl`

**Output:**

```
user@user-VMware-Virtual-Platform:~/Desktop$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 461 not upgraded.
```

3. To add a repository:-

**Command:** `curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -`

## Output:

```
user@user-VMware-Virtual-Platform:~/Desktop$ curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
[sudo] password for user:
2025-11-06 00:35:08 -
=====
DEPRECATION WARNING
=====
node.js 18.x is no longer actively supported!
You will not receive security or critical stability updates for this version.
You should migrate to a supported version of Node.js as soon as possible.

Please see https://nodesource.com/products/distributions for details about which
version may be appropriate for you.

The NodeSource Node.js distributions site contains
information both about supported versions of Node.js and N|Solid supported Linux
distributions. To learn more about usage, see:
https://nodesource.com/products/distributions

=====
Continuing in 10 seconds ...

2025-11-06 00:35:18 - Installing pre-requisites
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
75 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

```
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 https://deb.nodesource.com/node_18.x nodistro/main amd64 Packages [11.6 kB]
Fetched 23.7 kB in 1s (21.6 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
461 packages can be upgraded. Run 'apt list --upgradable' to see them.
2025-11-06 00:35:45 - Repository configured successfully.
2025-11-06 00:35:45 - To install Node.js, run: apt install nodejs -y
2025-11-06 00:35:45 - You can use N|solid Runtime as a node.js alternative
2025-11-06 00:35:45 - To install N|solid Runtime, run: apt install nsolid -y
```

- After adding the repository, install Node.js and npm:

**Command:** `sudo apt install -y nodejs`

## Output:

```
user@user-VMware-Virtual-Platform:~/Desktop$ sudo apt install -y nodejs
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 461 not upgraded.
Need to get 29.7 MB of archives.
After this operation, 187 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_18.x nodistro/main amd64 nodejs amd64 18.20.8-1nodesource1 [29.7 MB]
1% [1 nodejs 257 kB/29.7 MB 1%]
```

- Press Y when prompted to confirm installation. Check that the install was successful by querying node for its version number:

```
node -v && npm -v
```

#### Output

This installs node.js version 18.20 and NPM version 10.7.

```
user@user-VMware-Virtual-Platform:~/Desktop$ node -v && npm -v
v18.20.8
10.8.2
```

## Installing Juice Shop

OWASP Juice Shop is a deliberately vulnerable web application created by the Open Web Application Security Project (OWASP) for security education. It simulates an e-commerce platform, with real-world vulnerabilities that align with the OWASP Top Ten list, including SQL Injection, Cross-Site Scripting (XSS), and Broken Authentication.

By interacting with Juice Shop, students learn to:

1. Identify security weaknesses and understand the impact of vulnerabilities on application and database security.
2. Practice using security testing tools like OWASP ZAP to scan and exploit vulnerabilities.
3. Explore common security risks in a real-world context, with practical applications for securing modern web applications.

The steps to install Juice Shop are given below.

### 1. Install git to clone Github repositories.

Command: `sudo apt install git`

#### Output:

```
user@user-VMware-Virtual-Platform:~/Desktop$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 461 not upgraded.
Need to get 4,806 kB of archives.
After this operation, 24.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
Preparing to unpack .../git-man_1%3a2.43.0-1ubuntu7.3_all.deb ...
Unpacking git-man (1:2.43.0-1ubuntu7.3) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.43.0-1ubuntu7.3_amd64.deb ...
Unpacking git (1:2.43.0-1ubuntu7.3) ...
Setting up liberror-perl (0.17029-2) ...
Setting up git-man (1:2.43.0-1ubuntu7.3) ...
Setting up git (1:2.43.0-1ubuntu7.3) ...
Processing triggers for man-db (2.12.0-4build2) ...
```

- **Run the following command to clone into Juice Shop.**

Command : git clone <https://github.com/juice-shop/juice-shop.git> --depth 1

**Output:**

```
user@user-VMware-Virtual-Platform:~/Desktop$ git clone https://github.com/juice-shop/juice-shop.git --depth 1
Cloning into 'juice-shop'...
remote: Enumerating objects: 1259, done.
remote: Counting objects: 100% (1259/1259), done.
remote: Compressing objects: 100% (944/944), done.
remote: Total 1259 (delta 323), reused 893 (delta 296), pack-reused 0 (from 0)
Receiving objects: 100% (1259/1259), 45.94 MiB | 1.66 MiB/s, done.
Resolving deltas: 100% (323/323), done.
```

- **Go into the cloned folder and install the application.**

**Command :** cd juice-shop && npm install

**Output:**

Ignore the warnings you encounter during installation

- Give it some time to install. Once installed, the following output will show on the terminal.

## Output:

```
farish@farish-VMware-Virtual-Platform: ~/Desktop/juice-shop$ npm run build

✔ Browser application bundle generation complete.
✔ Copying assets complete.
✔ Index html generation complete.

Initial chunk files | Names | Raw size | Estimated transfer size
vendor.js          | vendor | 1.69 MB   | 366.75 kB
styles.css         | styles | 640.22 kB | 24.41 kB
main.js            | main  | 454.22 kB | 77.12 kB
polyfills.js       | polyfills | 34.84 kB | 11.33 kB
runtime.js         | runtime | 3.34 kB   | 1.53 kB

| Initial total | 2.82 MB | 481.14 kB

Lazy chunk files | Names | Raw size | Estimated transfer size
989.js           | faucet-faucet-module | 463.15 kB | 130.29 kB
380.js           | web3-sandbox-web3-sandbox-module | 426.55 kB | 98.03 kB
tutorial.js      | tutorial | 36.24 kB | 9.13 kB
300.js           | highlight-js-lib-core | 20.72 kB | 7.53 kB
388.js           | faucet-faucet-module | 11.90 kB | 3.64 kB
705.js           | confetti | 11.12 kB | 4.02 kB
common.js         | common  | 9.26 kB   | 856 bytes
188.js           | highlight.js-lib-languages-typescript | 7.77 kB | 2.77 kB
675.js           | wallet-web3-wallet-web3-module | 6.99 kB | 2.53 kB
236.js           | highlight.js-lib-languages-javascript | 6.52 kB | 2.36 kB
806.js           | highlightjs-line-numbers-js | 3.49 kB | 1.40 kB
928.js           | highlight.js-lib-languages-yaml | 1.93 kB | 772 bytes

Build at: 2023-11-07T10:00:00Z - Hash: d91e078a8cc0ed - Time: 870 ms
up to date, audited 2134 packages in 3m
236 packages are looking for funding
  run 'npm fund' for details
41 vulnerabilities (1 low, 15 moderate, 19 high, 6 critical)

To address all issues possible (including breaking changes), run:
  npm audit fix --force
```

- Once installed, start with the following command:

```
npm start
```

## Output:

```
farish@farish-VMware-Virtual-Platform: ~/Desktop/juice-shop$ npm start

> juice-shop@19.0.0 start
> node build/app

info: Detected Node.js version v24.11.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 20 of 20 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 runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Server listening on port 3000
```

- This application is running on port 3000. You can scan the ports using nmap.

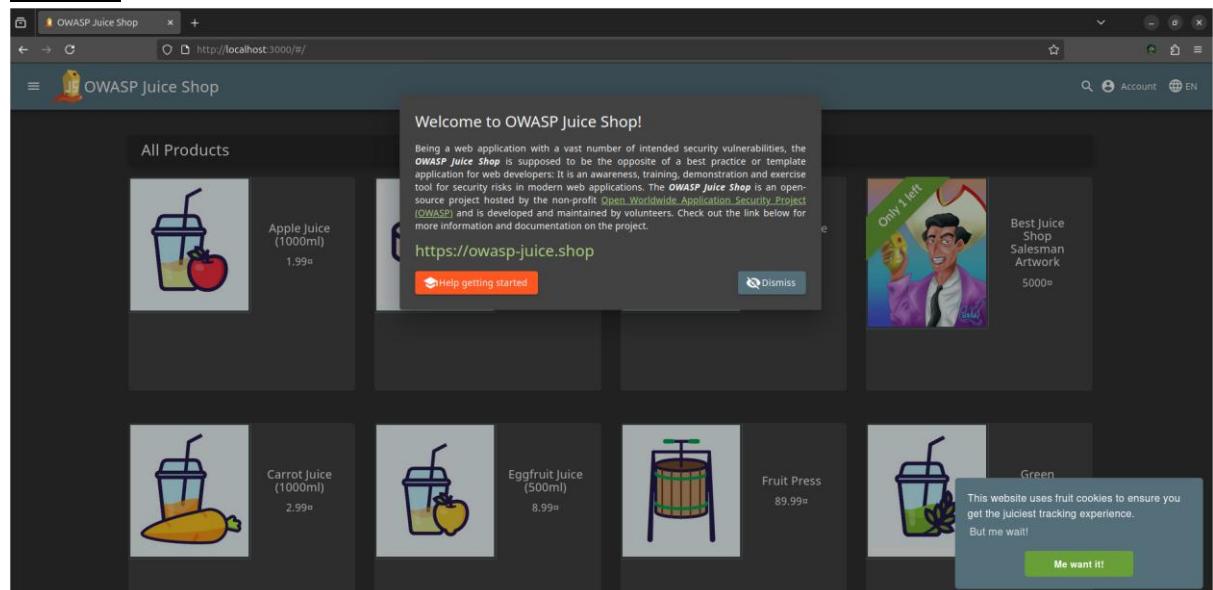
**Command:** nmap localhost

### **Output:**

```
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$ nmap localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-03 22:04 PKT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00026s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
631/tcp    open  ipp
3000/tcp   open  ppp
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$ nmap localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-03 22:04 PKT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00020s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
631/tcp    open  ipp
3000/tcp   open  ppp
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$
```

- This shows that port 3000 is open for `tcp` connections.
- Open the browser and browse <http://localhost:3000>

### **Output:**



## Installing and Setting up Zap

- Run the following command to install Zaproxy:

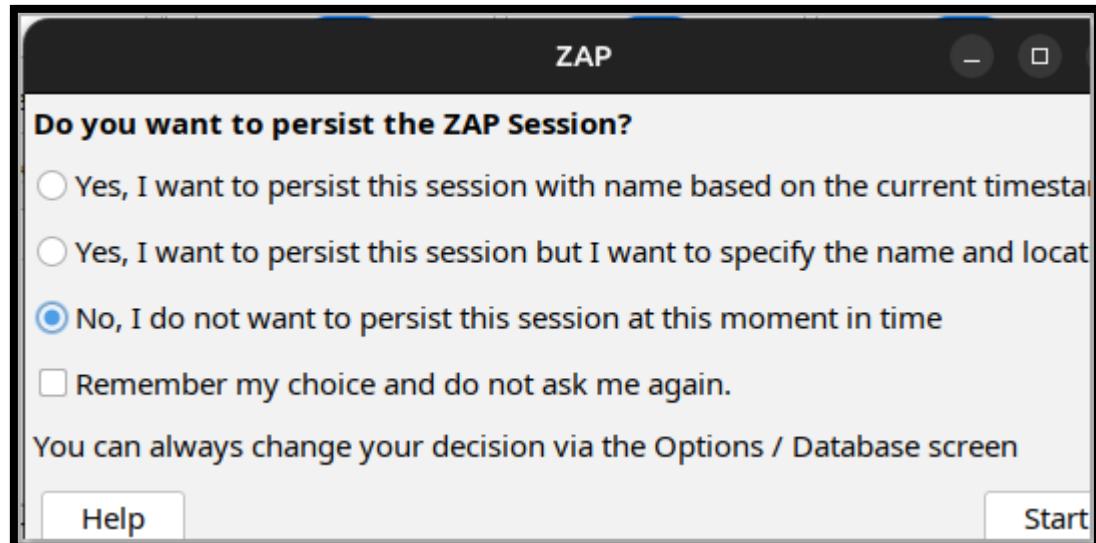
Command: `sudo snap install zaproxy --classic`

### Output:

```
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$ sudo snap install zaproxy --classic
zaproxy 2.16.1 from Simon Bennetts (psinon) installed
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$ sudo snap install zaproxy --classic
[sudo] password for fariah:
snap "zaproxy" is already installed, see 'snap help refresh'
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$
```

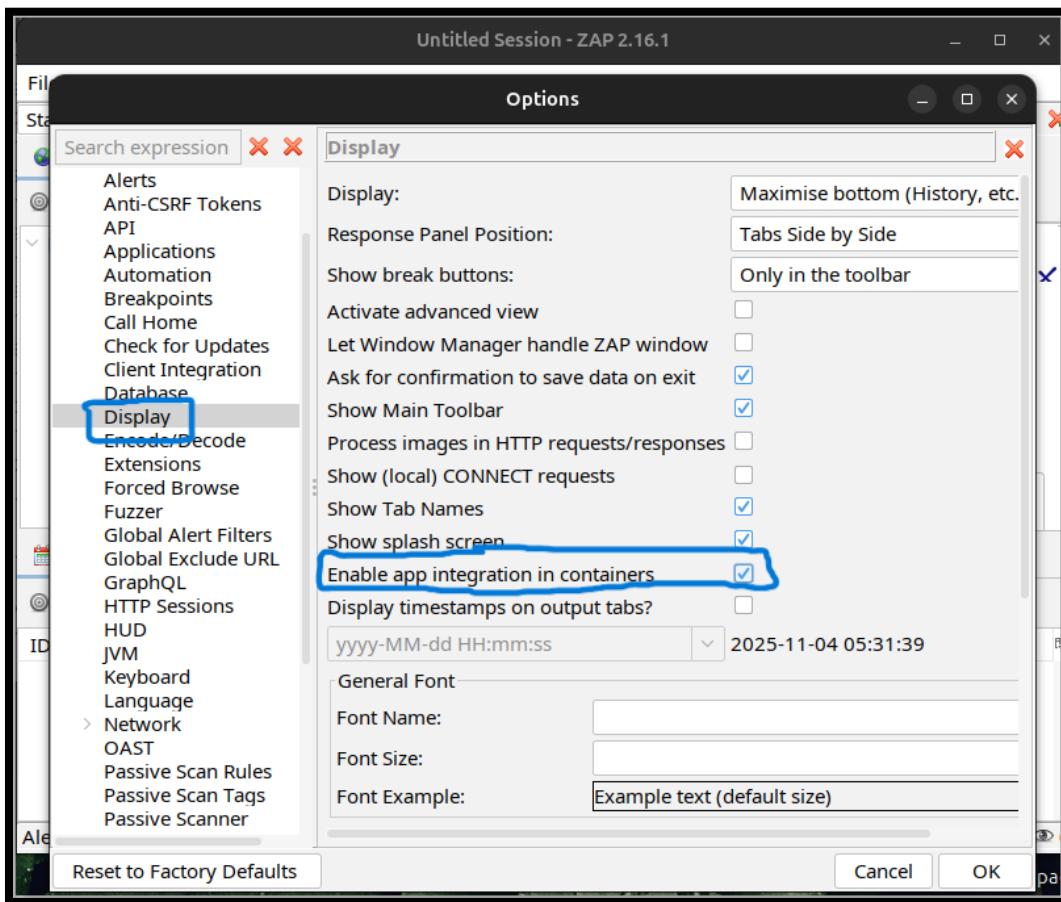
Run Zap : `zaproxy`

### Output:



- You can select the third option.
- And go to tools -> Options select Display and mark the option enable app containers and click ok integration in

## Output:

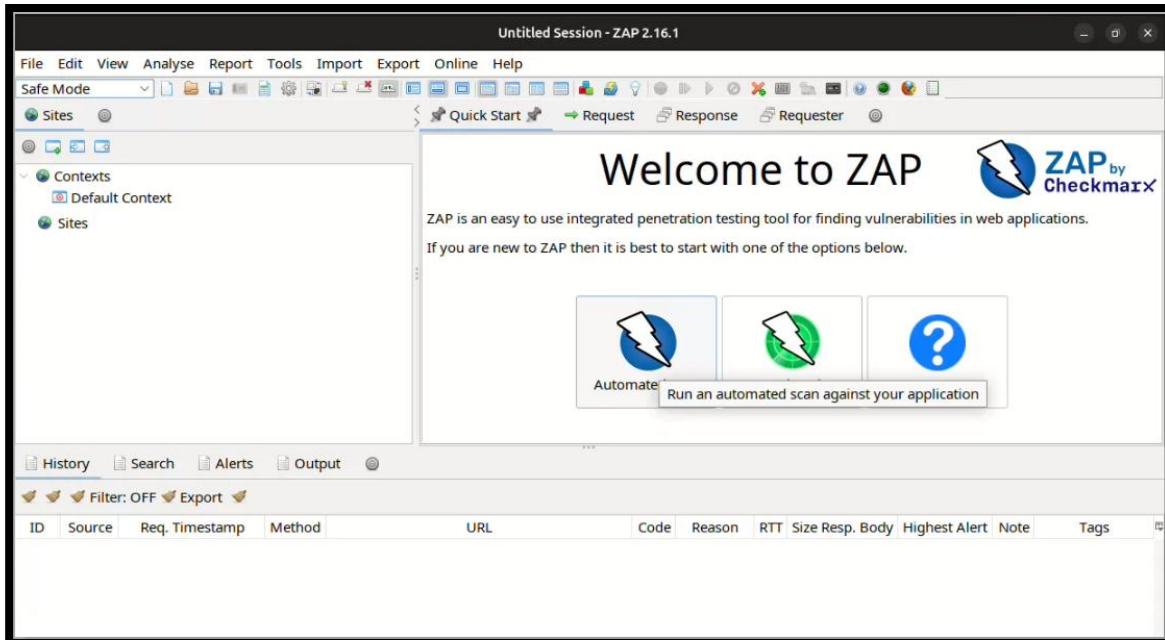


## Part 2: Running an Automated Scan

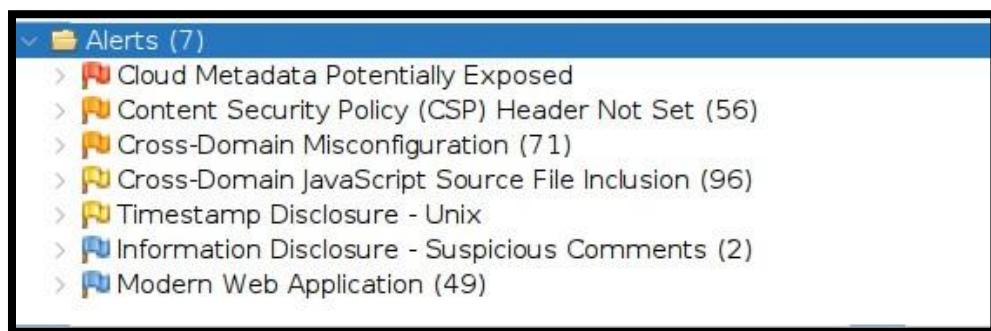
In an automated scan, ZAP passively scans traffic to identify potential vulnerabilities without actively attacking the application

From the home screen of Zap UI, click **Automated Scan**. In the URL, add <http://localhost:3000/#/> and click **Attack**.

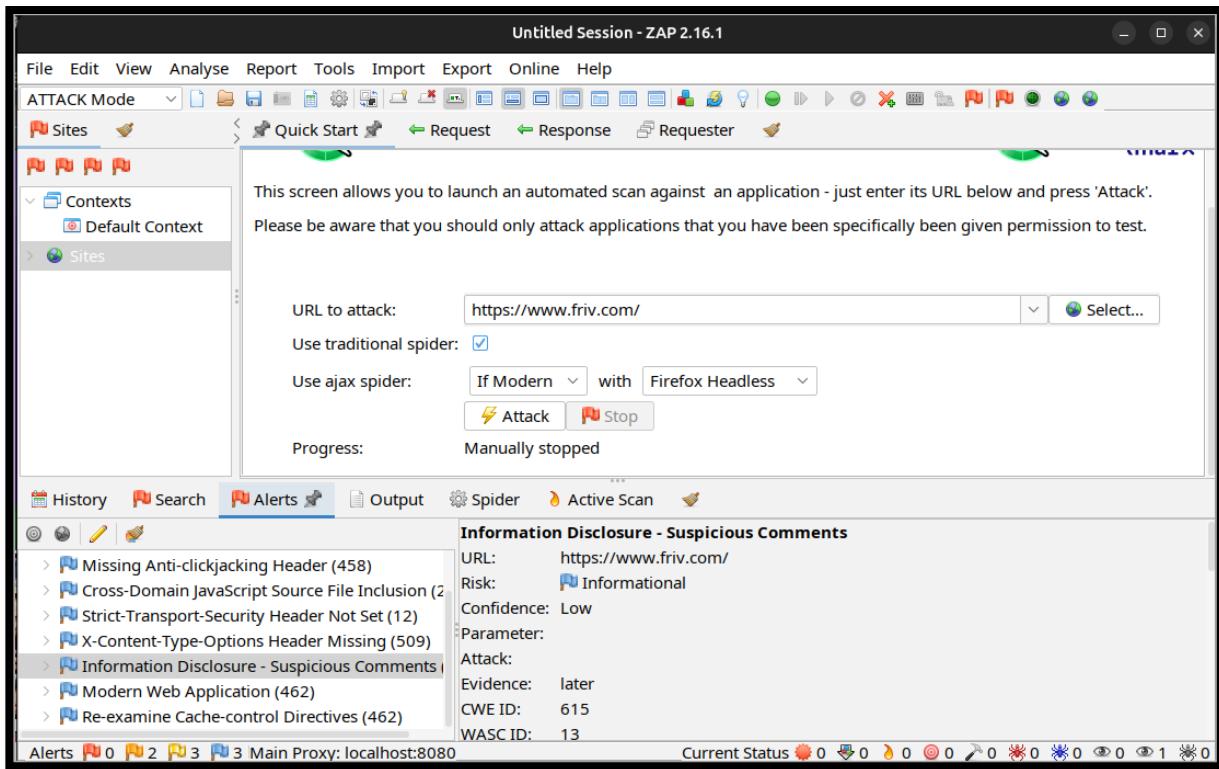
## Output:



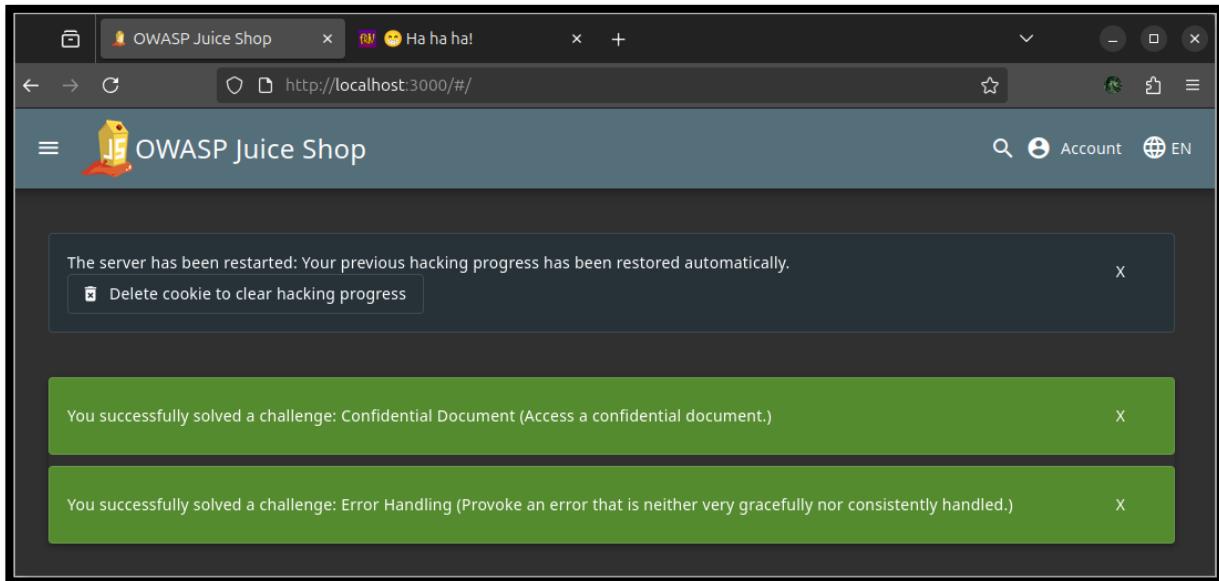
Once scanned, it shows security alerts. Check the `Alerts` tab to see the security issues identified.



You can click an alert to see details.



Once the automated attack is complete, you will see the following messages in the browser:



## Post-Lab Tasks

Perform a basic, manual SQL injection attack outside of ZAP by directly interacting with Juice Shop:

- Attempt an SQL injection on the login page of Juice Shop by entering typical SQL payloads (' OR '1='1 in the username or password fields).

The screenshot shows a browser window for the OWASP Juice Shop login page at <http://localhost:3000/#/login>. The login form has the following fields:

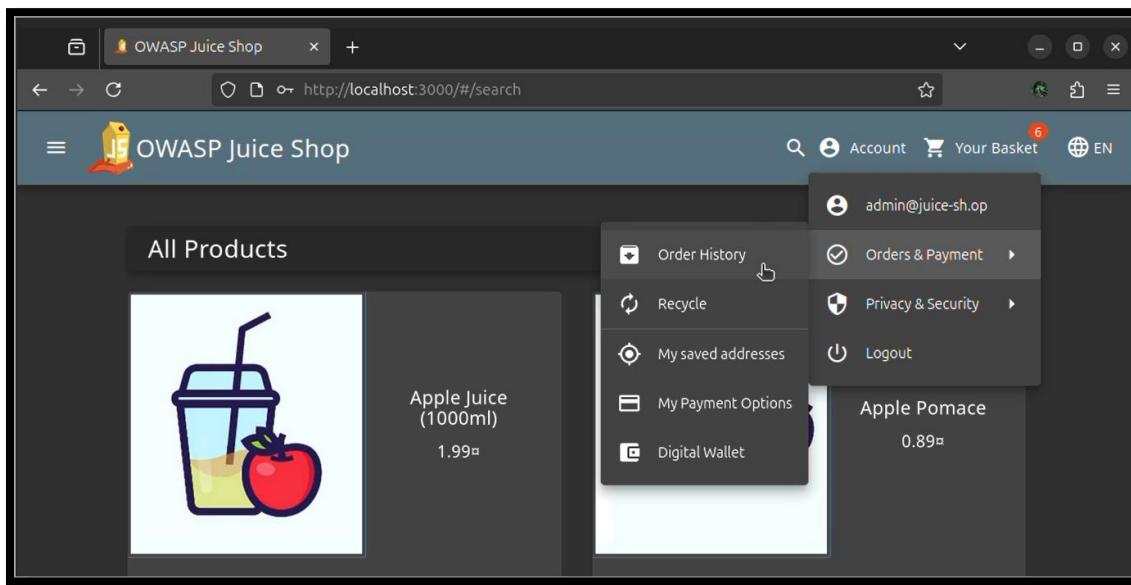
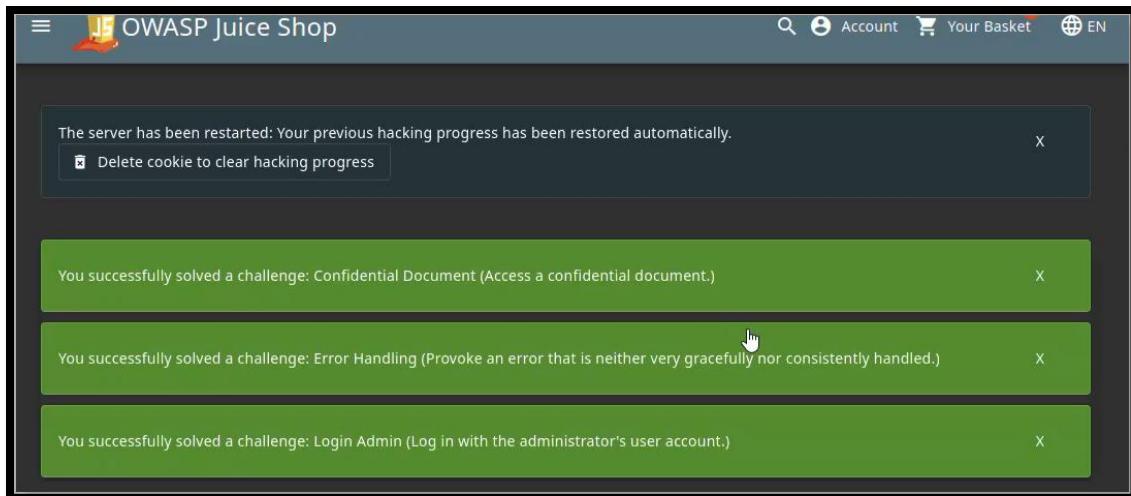
- Email\*: ' or 1 = 1;--
- Password\*: asdf1234

The 'Log in' button is highlighted in blue. Below the form, there is a 'Remember me' checkbox and a 'Log in with Google' button.

The screenshot shows a browser window for the OWASP Juice Shop login page at <http://localhost:3000/#/login>. The login form has the following fields:

- Email\*: ' OR '1='1
- Password\*: ' OR '1='1

An error message 'Invalid email or password.' is displayed above the form. The 'Log in' button is greyed out. Below the form, there is a 'Remember me' checkbox and a 'Log in with Google' button.



### Reference:

<https://csnainc.io/npm-install-stuck/> -  
A text message: "text=A%20stuck%20npm%20install%20happens%20for%20two%20main, If%20it%20fails%2C%20reinstall%20Node.js%2Fnpm%20and%20fix%20PATH."

**Q: Write a brief reflection on how a manual attack differs from automated scans and how it enhances understanding of vulnerabilities.**

Manual attacks differ from automated scans in both approach and depth of insight. While automated tools like vulnerability scanners quickly identify known issues across systems using

predefined signatures and rules, manual attacks require human intuition, creativity, and contextual understanding.

- **Automated scans** are efficient for broad coverage, flagging common misconfigurations, outdated software, or missing patches. However, they often miss logic flaws, chained vulnerabilities, or subtle misuses of functionality.
- **Manual attacks**, on the other hand, simulate real-world adversarial behavior. They involve probing inputs, manipulating workflows, and thinking like a hacker. This hands-on process reveals deeper insights into how vulnerabilities arise from design flaws, user behavior, or overlooked edge cases.

By performing manual attacks, you gain:

- A stronger grasp of **application logic and architecture**
- The ability to **interpret scanner results critically**
- Experience in **crafting exploits and understanding impact**
- Confidence in **thinking beyond the toolset**

Ultimately, manual testing enhances your understanding by connecting theory to practice transforming abstract vulnerabilities into tangible, exploitable weaknesses. It's where your analytical instincts and technical curiosity truly come alive.

---