NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

INFORMATION SECURITY LAB

NAME	Ayesha Imran
Class	CS-A
Lab	08
Course	Information Security
Date	3-November-25
Submitted To	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`

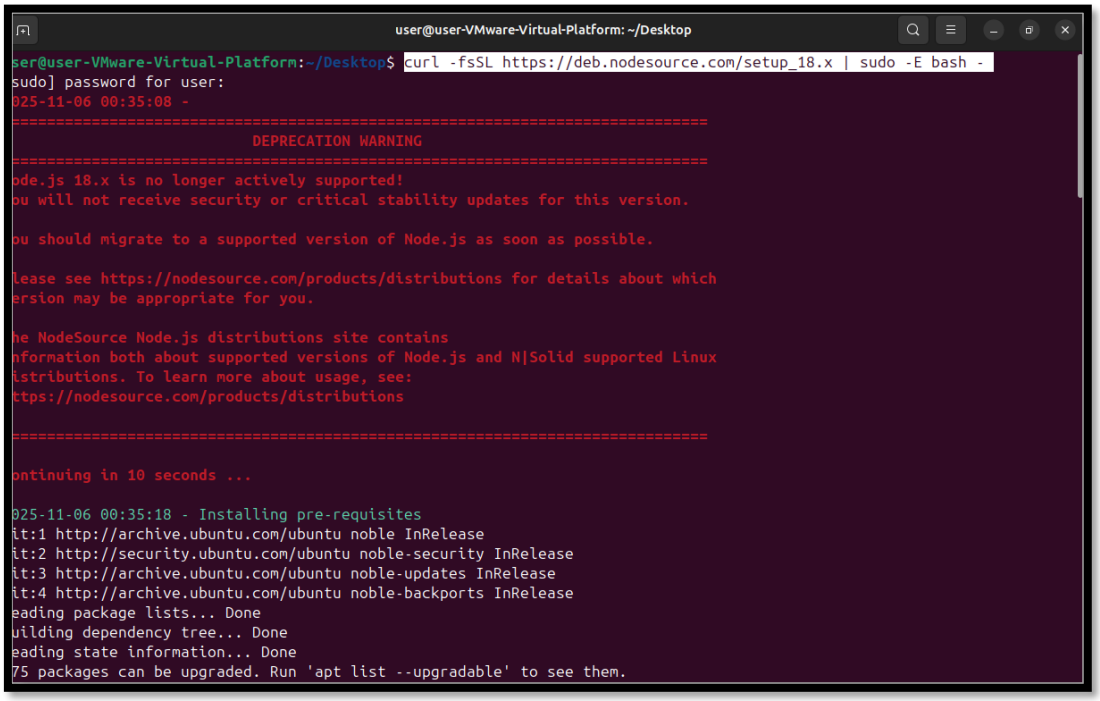
Output:

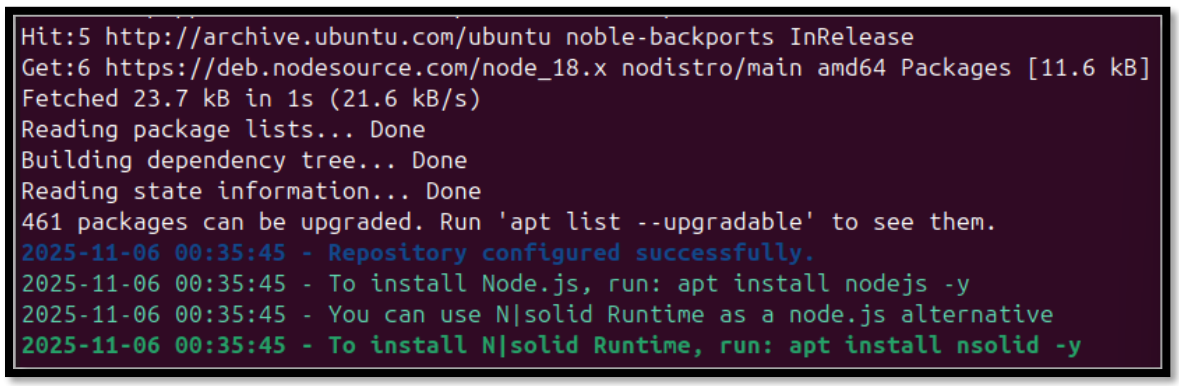
```
user@user-VMware-Virtual-Platform:~/Desktop$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
0 upgraded, 0 newly installed, 0 to remove and 461 not upgraded.
```

3. To add a repository:-

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

Output:

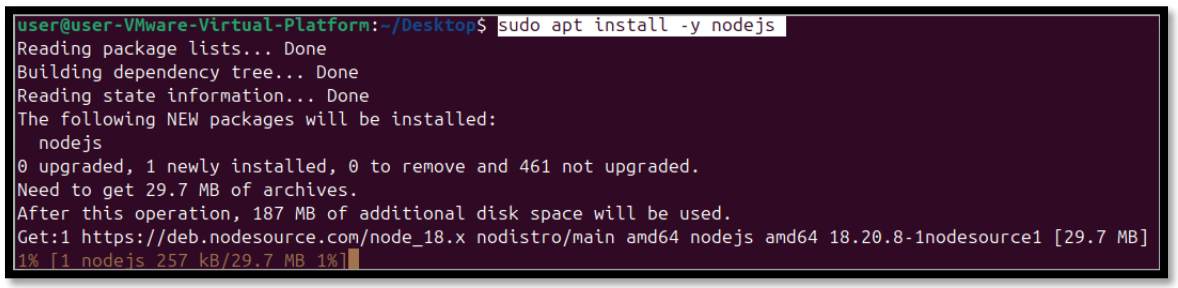




- After adding the repository, install Node.js and npm:

  Command: `sudo apt install -y nodejs`

  Output:



- 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**.

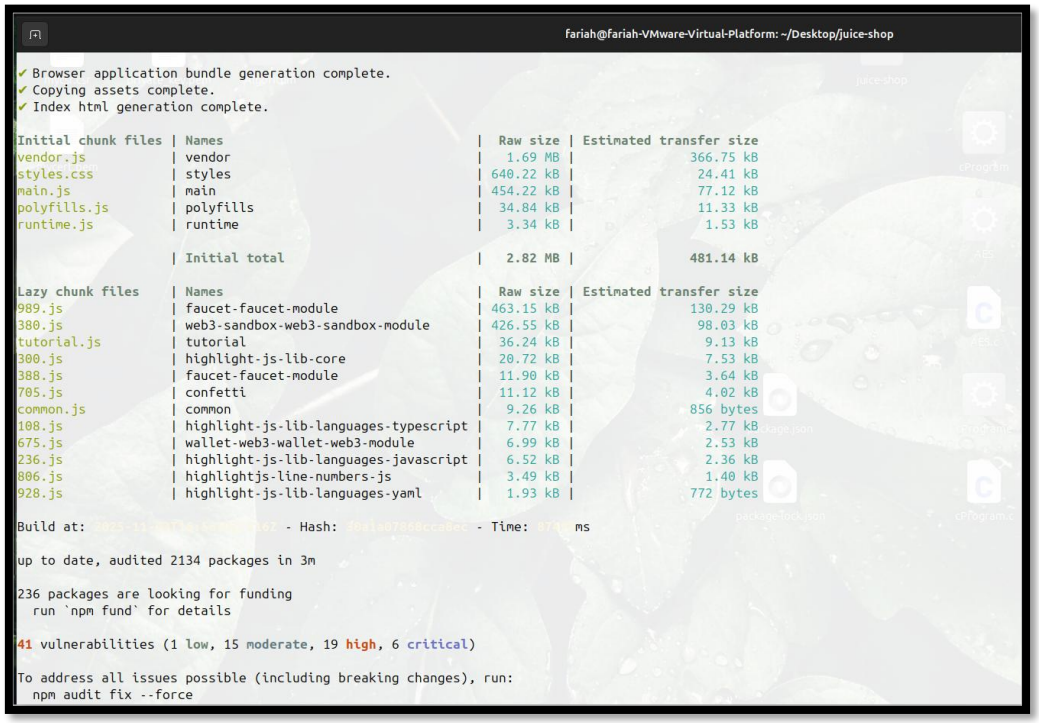
  Command : cd juice-shop && npm install

  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:



- Once installed, start with the following command:

```
npm start
```

Output:

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

Command: `nmap localhost`

Output:

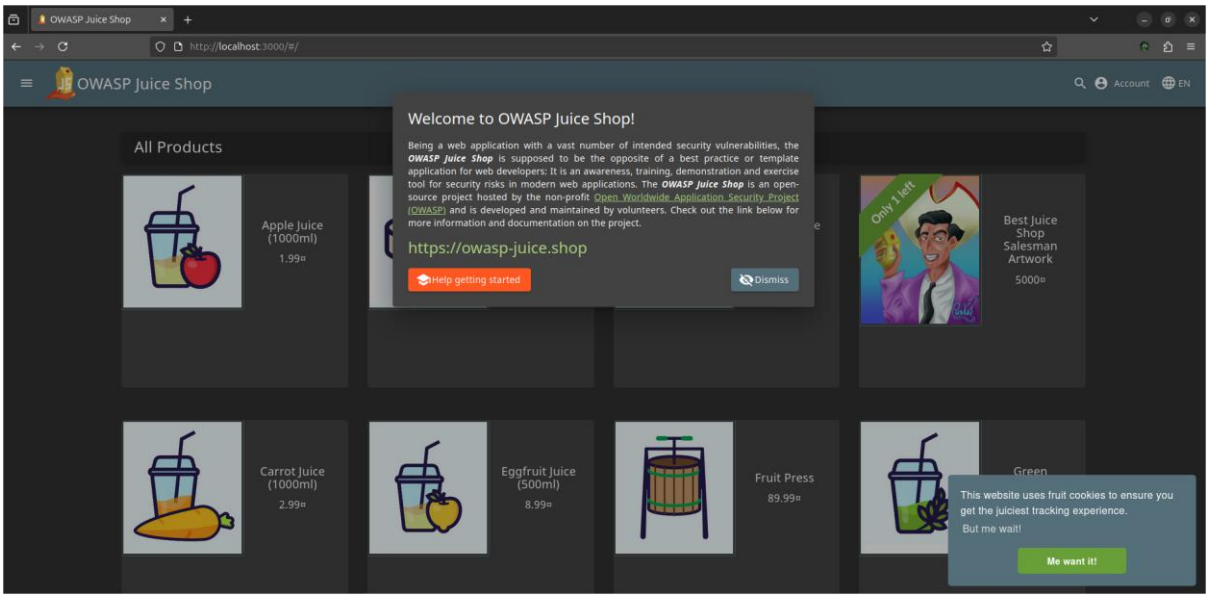
```
fariah@fariah-VMware-Virtual-Platform:~/Desktop/juice-shop$ nmap localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-03 22:04 PKT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00026s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
631/tcp  open  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:

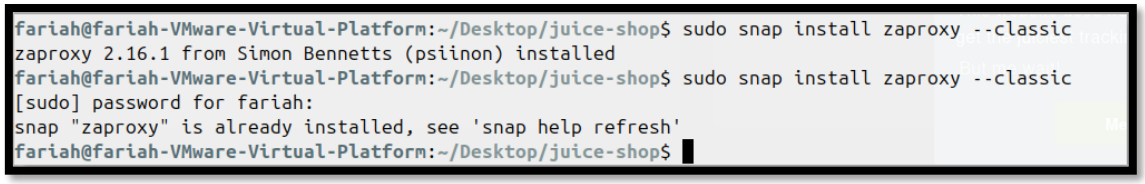


Installing and Setting up Zap

- Run the following command to install Zaproxy:

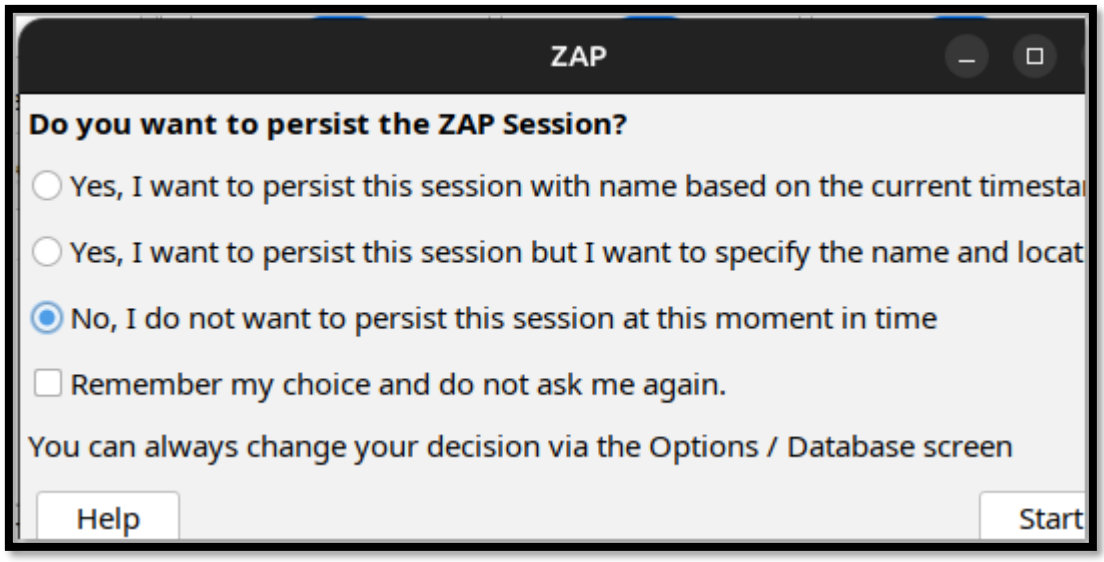
Command: sudo snap install zaproxy --classic

Output:



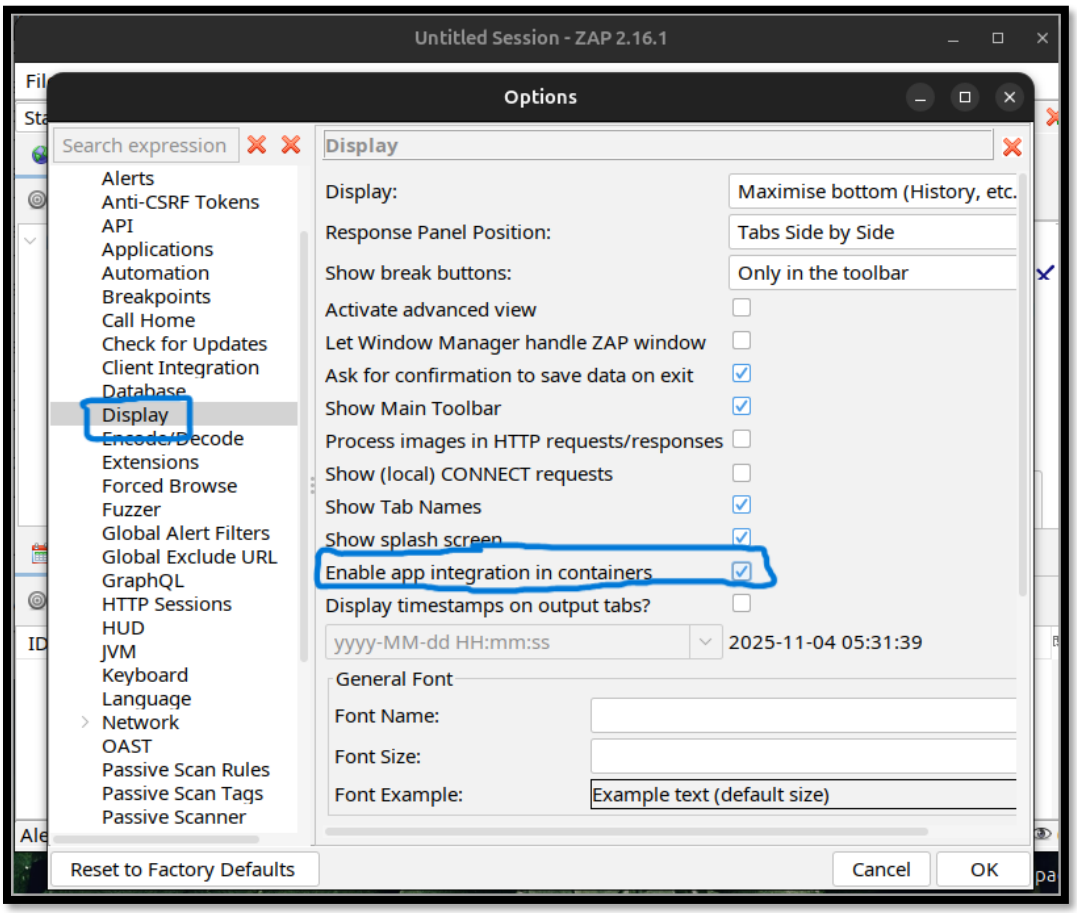
Run Zap : zaproxy

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:



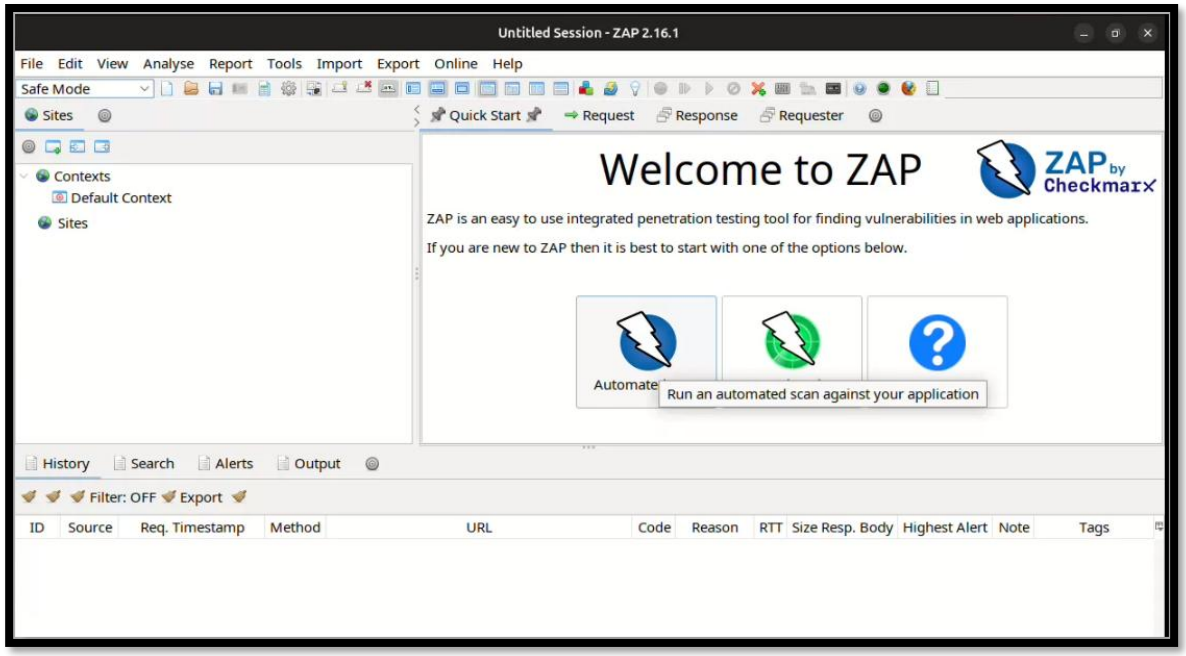
Part 2: Running an Automated Scan

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

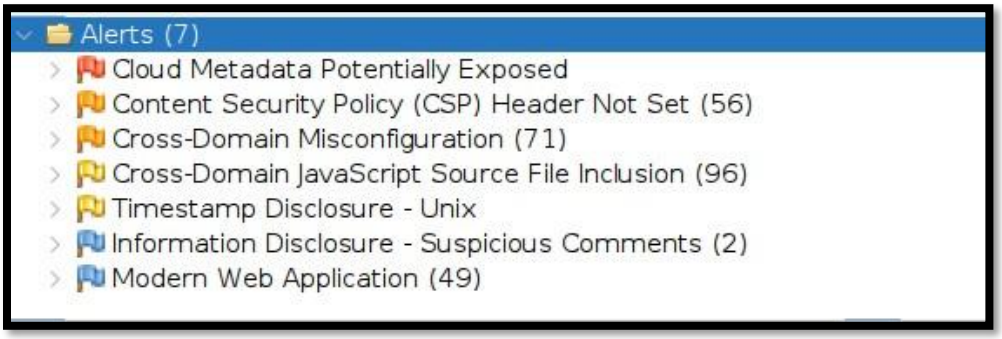
.

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

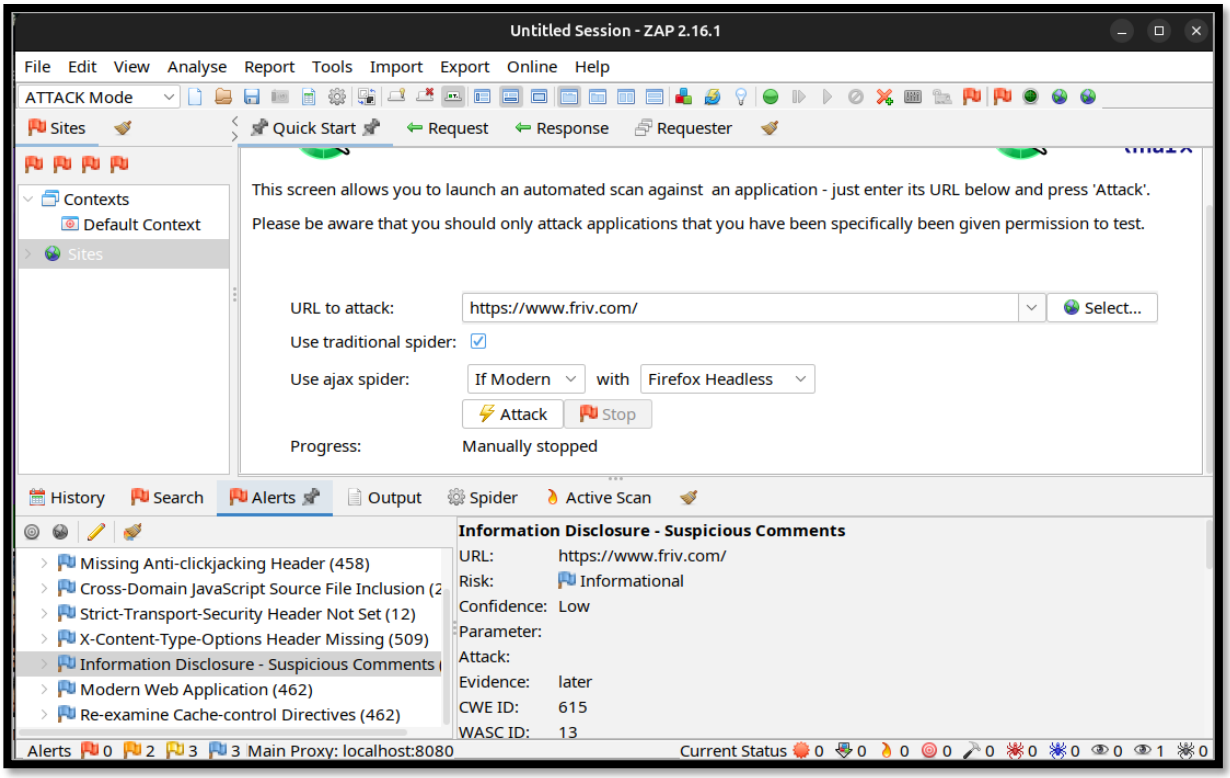
Output:



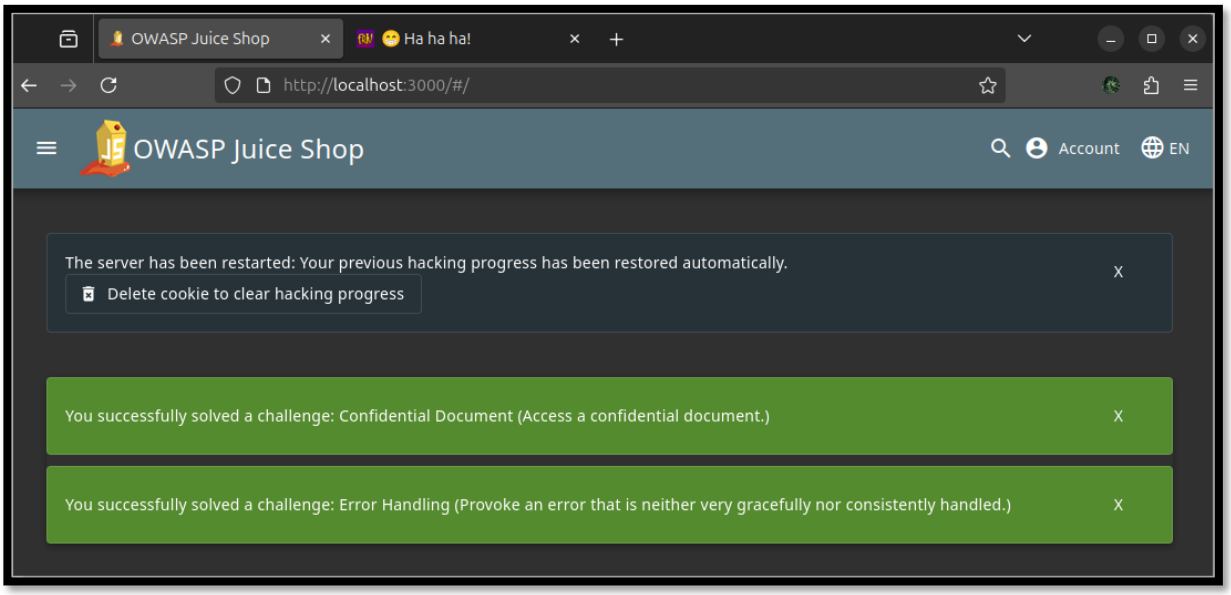
Once scanned, it shows security alerts. Check the Alerts tab to see the security issues identified.



You can click an alert to see details.



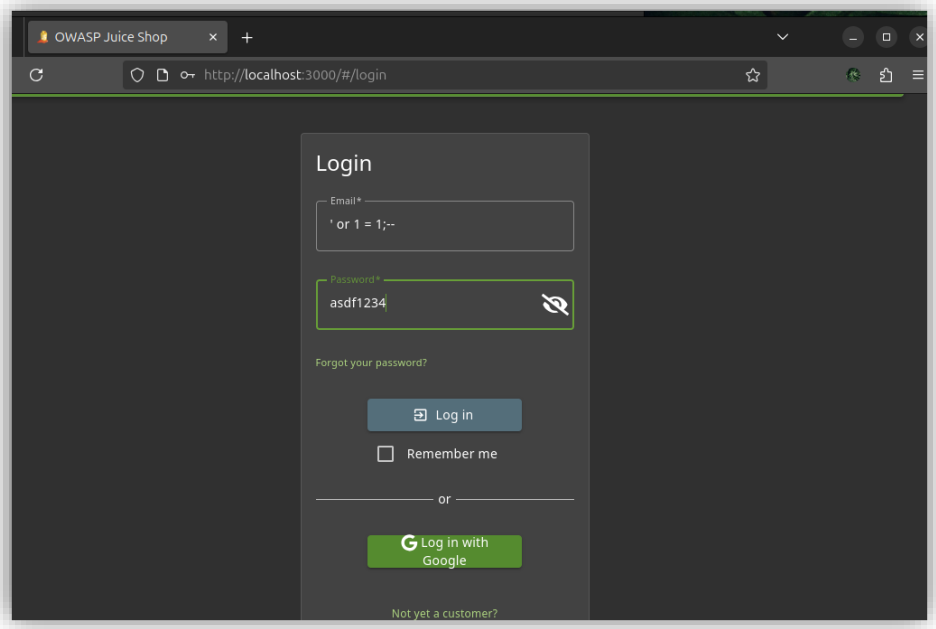
Once the automated attack is complete, you will see the following messages in the browser:

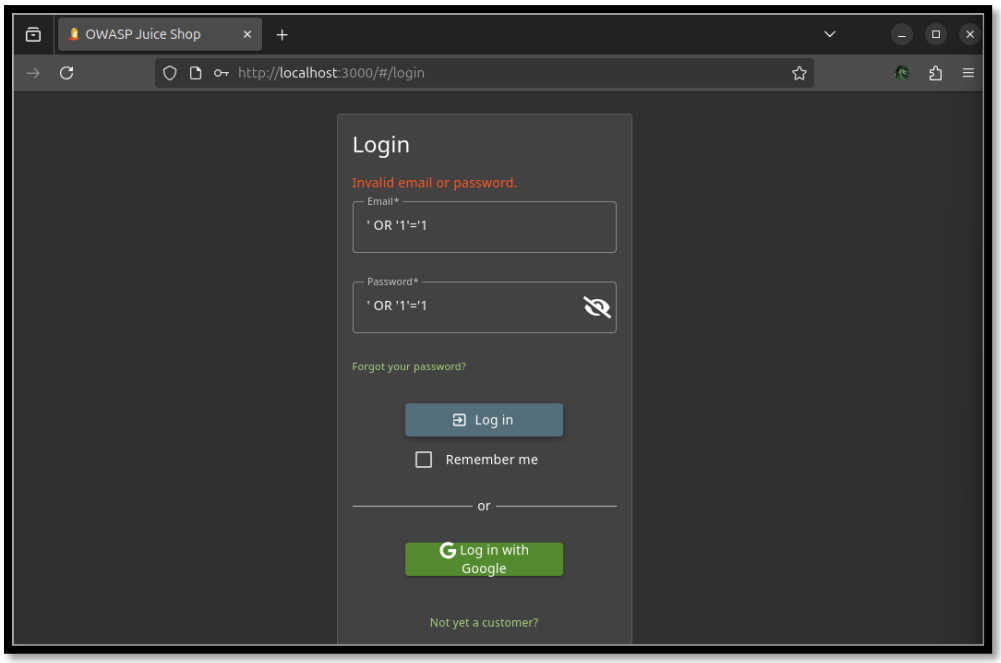


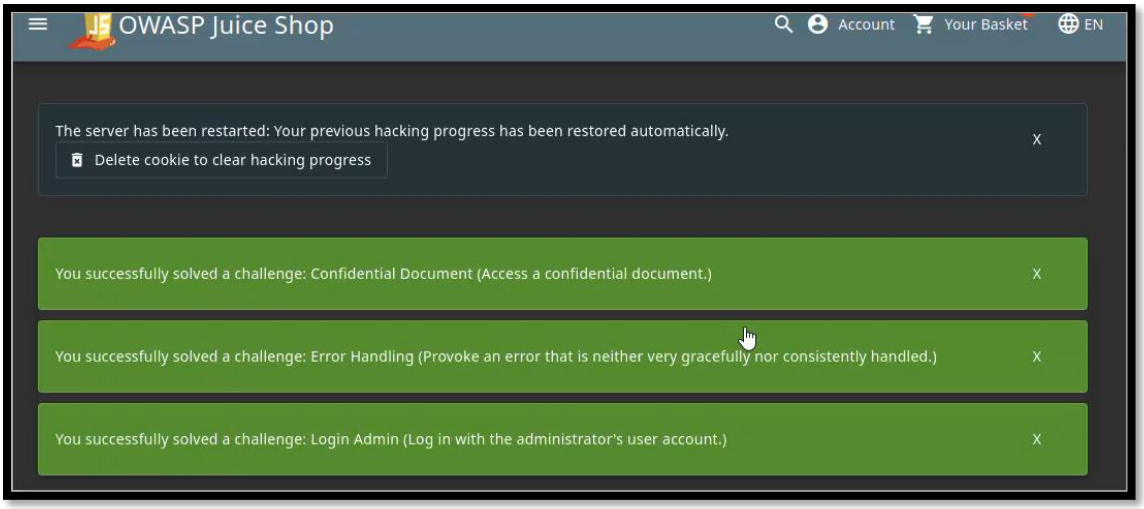
Post-Lab Tasks

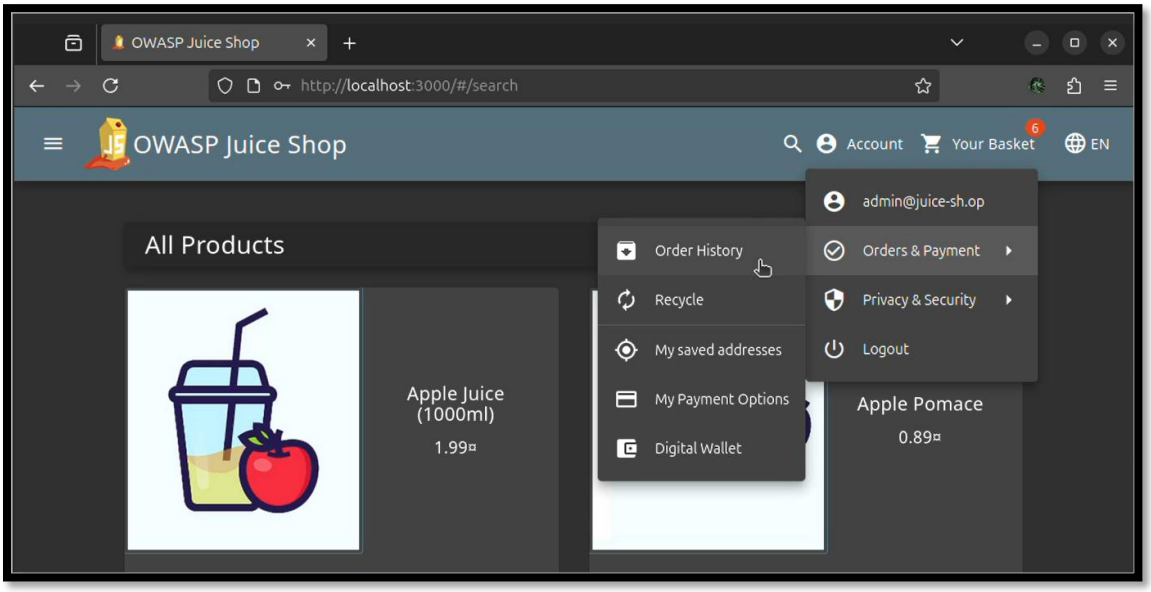
Perform a basic, manual SQL injection attack outside of ZAP by directly interacting with Juice Shop:

- 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).









Reference:

https://csnainc.io/npm-install-stuck/ -:~: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.
