# 

# CYBERSECURITY PROJECT

PROJECT: **INCIDENT RESPONSE AND AUTOMATION**

**Table of content:**

* Project Summary ………………………………………………….2
* Project scope ………………………………………………………3
* Project problem……………………………………………………..3
* Project Solution……………………………………………………..4
* Implementation
* Installation of virtualbox ……………………………………5
* Docker Installation ………………………………………….6
* Web Deployment on vm……………………………………8
* Securtiy Monitoring setup on vm …………………………9
* Monitoring Logs ……………………………………………12
* Installation of zaproxy……………………………………...19
* Conclusion………………………………………………….23

**Project Summary:**

* This project aims to provide a hands-on introduction to web application security and incident response using OWASP Juice Shop, an intentionally vulnerable web application. The project involves setting up a secure environment, deploying the application, monitoring for security events, simulating attacks, detecting incidents, and developing an incident response process. The tools used include VirtualBox, Ubuntu Server 20.04 LTS, Docker, ELK Stack (Elasticsearch, Logstash, Kibana), ModSecurity, Nmap, OWASP ZAP, and Git. By following the steps outlined, beginners will gain practical experience in web security, vulnerability assessment, log analysis, and incident response planning.
* The project is structured to guide beginners through each step, starting with setting up the environment and deploying the Juice Shop application. It emphasizes learning key concepts such as web application architecture, basic web vulnerabilities (focusing on the OWASP Top 10), and the use of security tools like OWASP ZAP for vulnerability scanning. Additionally, the project includes configuring the ELK Stack for comprehensive logging and visualization of security events, and using ModSecurity as a web application firewall.

**Project scope :**

# **1.Objective:**

* Gain practical Experience with Incident Response and Automation.
* Understanding Vulnerabilities ,Web security and Log analysis.
* Develop skills in detecting and responding to Security Incident.
* Implement and test security monitoring ,Automation for response.

# **2.Requriments:**

## Software:

* **Virtualization Software** : Vmware Workstation, VirtualBox,Docker
* **Operating System**: Windows,Linux
* **Server** : Ubuntu Server 20.04 LTS ISO

# **3.Environmental set up:**

## Virtual Machines

**• Ubuntu Server**: Ubuntu Server 20.04 LTS ISO

• **Windows Server:** Domain Controller, file server.

**• Linux Server**: Web server, database server.

**• Client Machines**: Windows 11.

**Project Problem:**

The rise of sophisticated cyber threats has made web application security a critical concern for organizations. Many web applications are susceptible to various vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), which can lead to data breaches, unauthorized access, and significant financial losses. Despite the availability of security tools and frameworks, there is often a gap in practical knowledge and experience among cybersecurity professionals, particularly beginners.

**Project Solution:**

This project aims to address the following key problems:

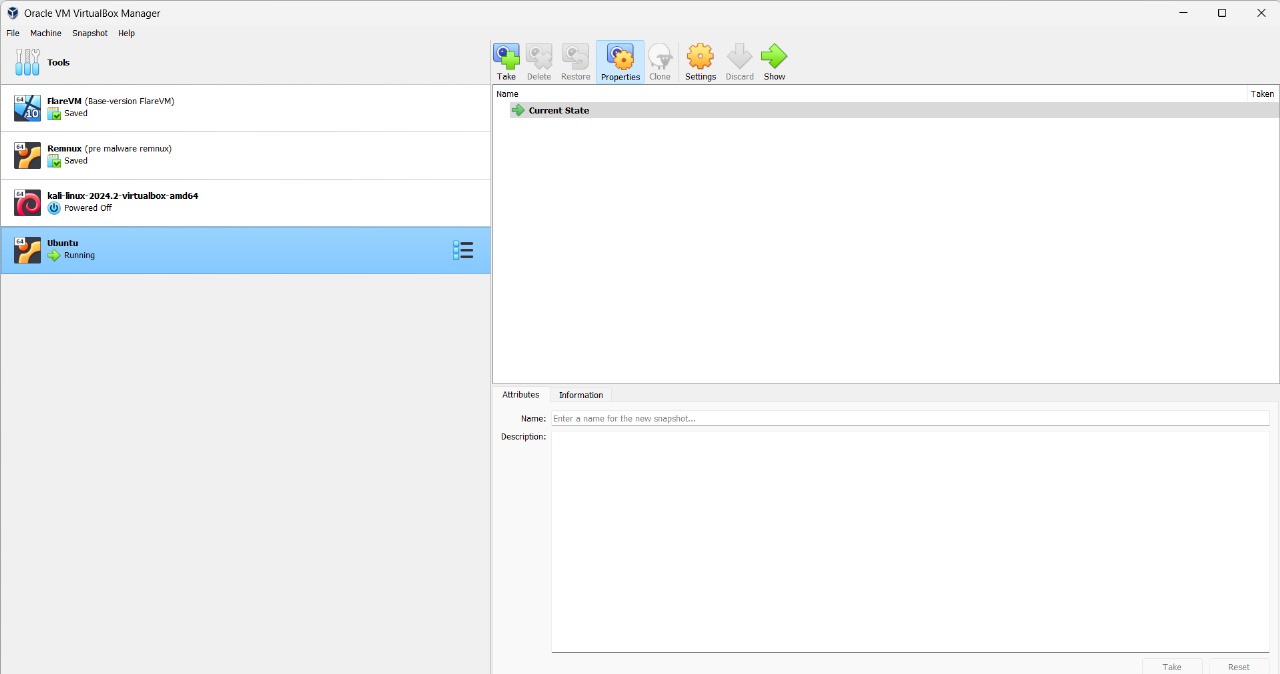
1. **Lack of Practical Experience:**
   * Many cybersecurity beginners lack hands-on experience with real-world web application security tools and techniques.
   * There is a need for an educational platform that provides a safe and controlled environment to practice these skills.
2. **Inadequate Incident Response:**
   * Organizations often struggle with developing effective incident response plans and playbooks.
   * There is a need to simulate and practice incident detection and response to improve readiness for actual security incidents.
3. **Difficulty in Monitoring and Analysis:**
   * Monitoring web applications for security events and analyzing logs can be challenging without the right tools and configurations.
   * Implementing comprehensive logging and monitoring systems is crucial for effective incident detection and response.

**4**.**Understanding of Web Vulnerabilities:**

* + Beginners need to understand common web vulnerabilities and how to exploit them to better defend against real-world attacks.
  + There is a need for structured learning that includes both vulnerability assessment and practical attack simulation.
  + **1.Installation of Ubuntu in Virtual Box:**

Step 1: Installation of ubuntu in virtual box <https://releases.ubuntu.com/20.04/>

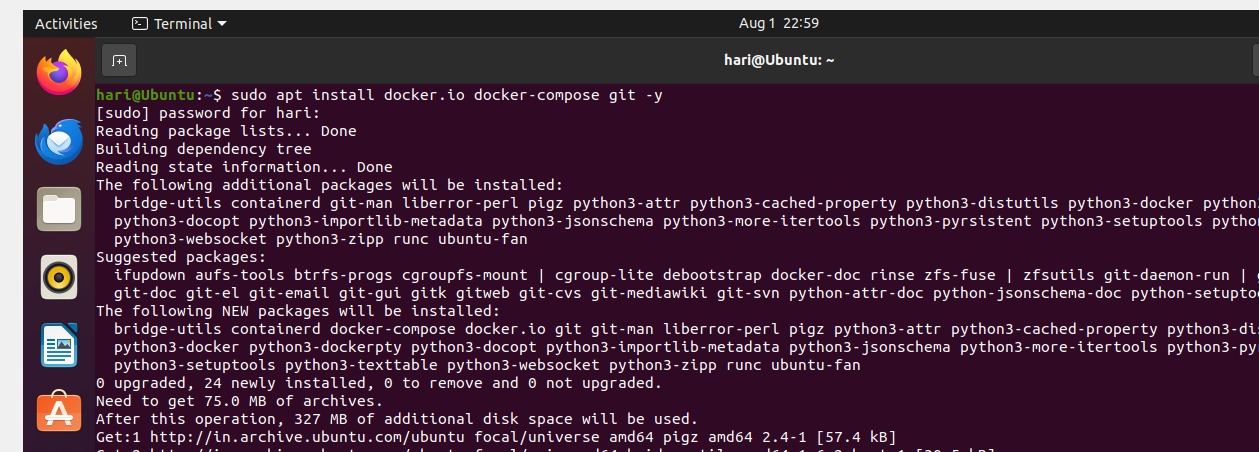
Step 2 : Configuration of Ubuntu in Virtual Box



**2.Docker Installation:**

Step 1:sudo apt install docker.io docker-compose git -y

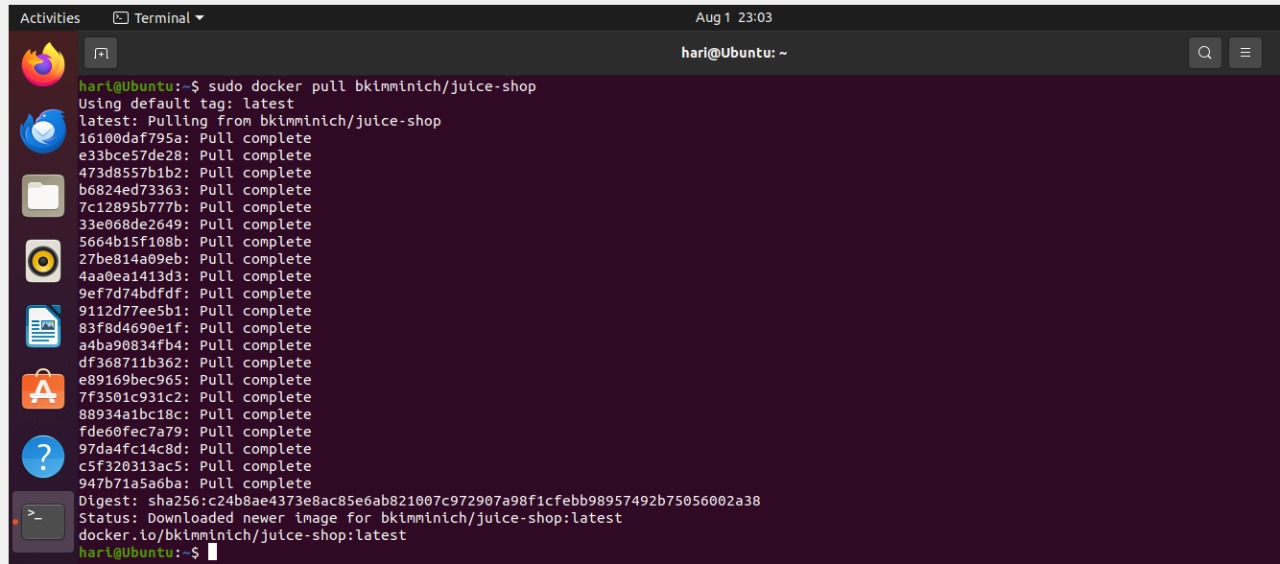
Cloning Docker from Github



* Docker is used in this project to provide a consistent, isolated environment for deploying the web application (**OWASP Juice Shop**) and monitoring tools (**Splunk**). It simplifies setup, improves reproducibility, and allows for easy management of different components. Docker's efficiency and modularity make it ideal for simulating and testing security scenarios, while also providing valuable experience with a tool commonly used in cybersecurity. This approach allows you to focus more on security aspects rather than complex installation procedures.
* Using Docker in this project enhances scalability and flexibility. By containerizing the OWASP Juice Shop application and the monitoring tools, you can easily scale individual components up or down based on the needs of your testing environment. This modular approach allows for quick adjustments and the ability to run multiple instances of the same application or different configurations without interference. Docker's integration with orchestration tools like Docker Compose also facilitates the management of multi-container setups, ensuring that all components work seamlessly together.

Step 2:

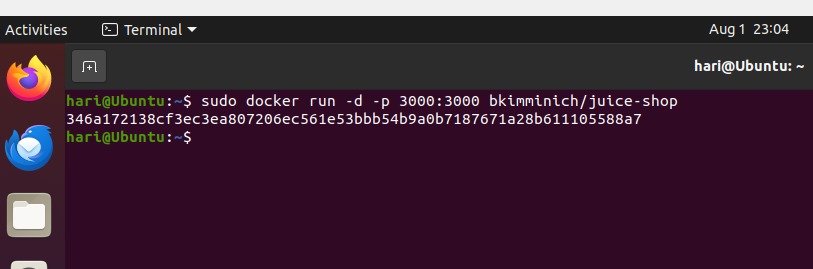
* Pull the OSWAP Juice shop image in the Docker
* Using command: sudo docker pull bkimminich/juice-shop



Step 3 :

* + Running the Juiceshop Application in the Docker
  + Using the command:

sudo docker run -d -p 3000:3000 bkimminich/juice-shop



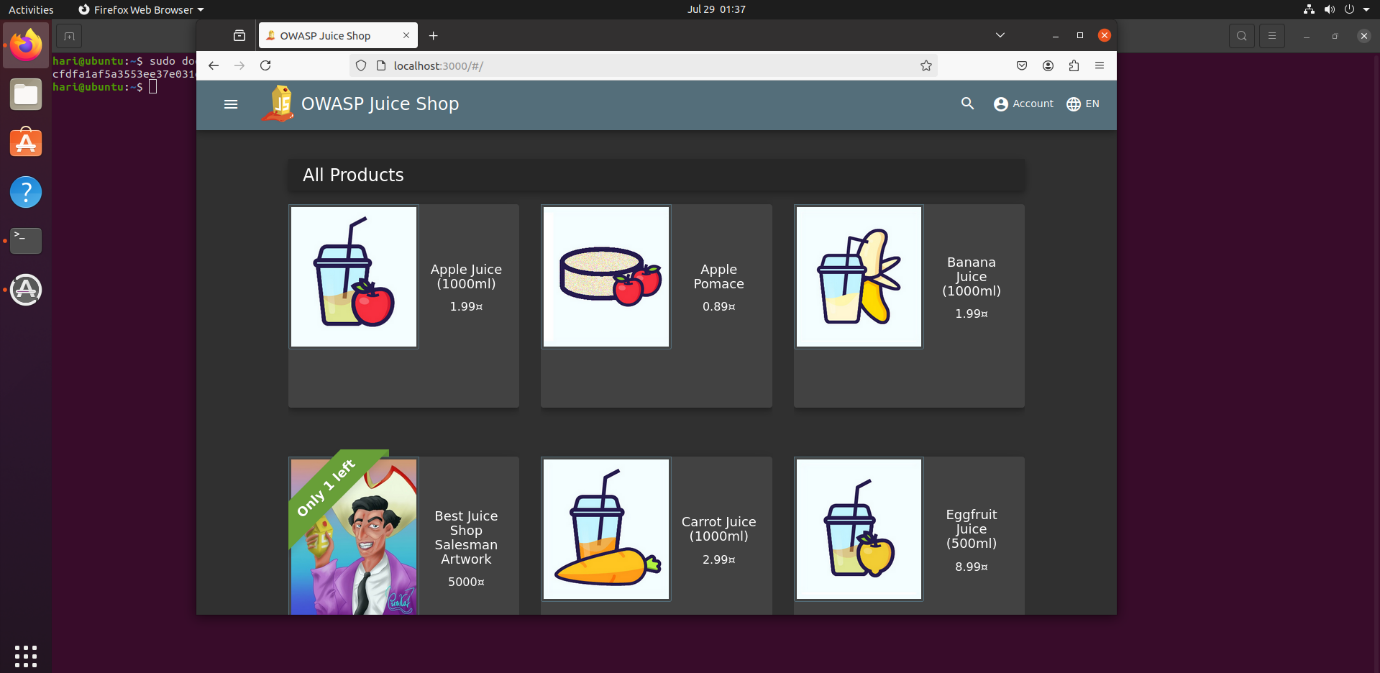
**3. Web Application Deployment on VM**

Step 1: Pull and Run OWASP Juice Shop:

* docker pull bkimminich/juice-shop
* docker run -d -p 3000:3000 bkimminich/juice-shop

Step 2: Verify the Application:

* Open a web browser on your host machine and navigate to <http://localhost:3000>



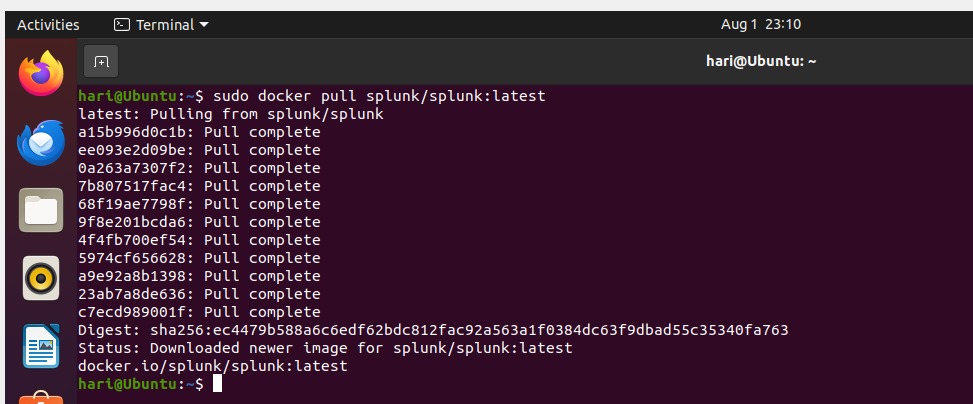
**4. Security Monitoring Setup on VM**

Step 1:

**Installation of Splunk:**

* + Installation of splunk in docker
  + Command for installing Splunk in docker

>>>> docker pull splunk/splunk:latest

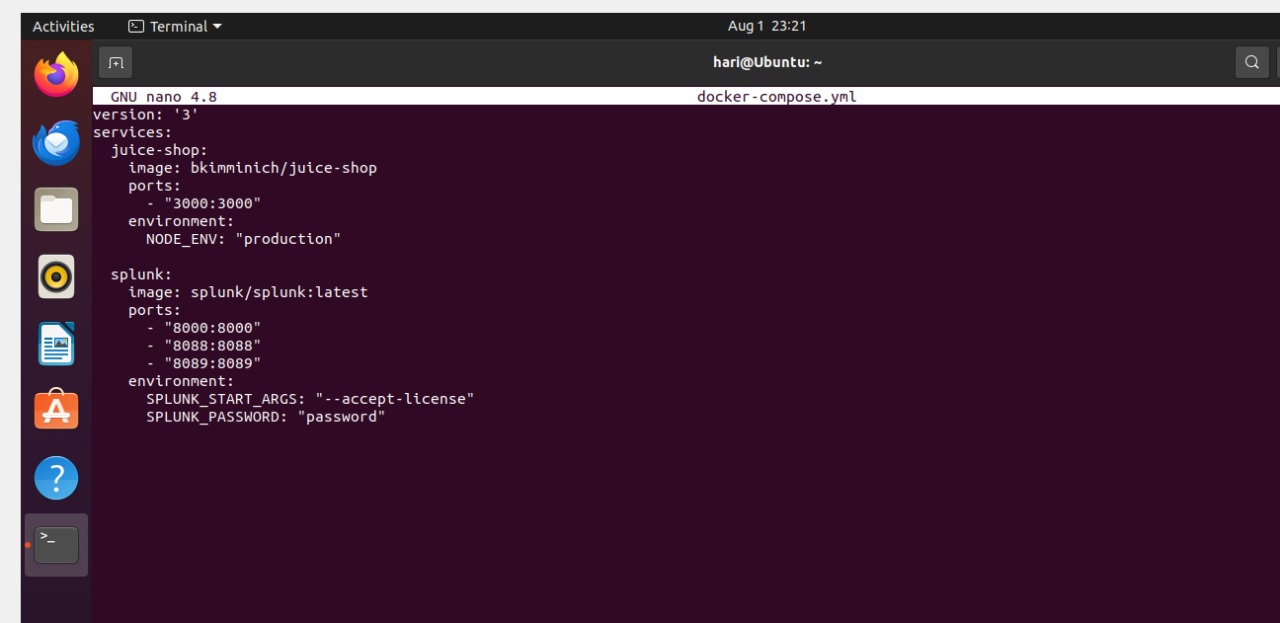


Step 1 of Installing Splunk in the Docker Platform

Step 2:

**Splunk Configuration:**

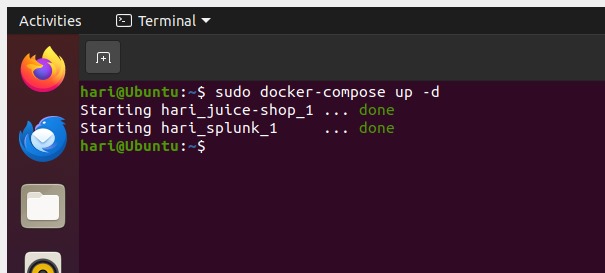
* Ensure data persistence by mounting a local directory to store Splunk data outside the container.
* Customize Splunk settings by mounting a local directory containing configuration files
* Configure environment variables to set the admin password and accept the license agreement.
* Expose necessary ports (e.g., 8000 for the web interface and 8088 for the HTTP Event Collector) to access Splunk services.
* Saving the file in the “docker-compose.yml”



Step 3:

**Running the configuration:**

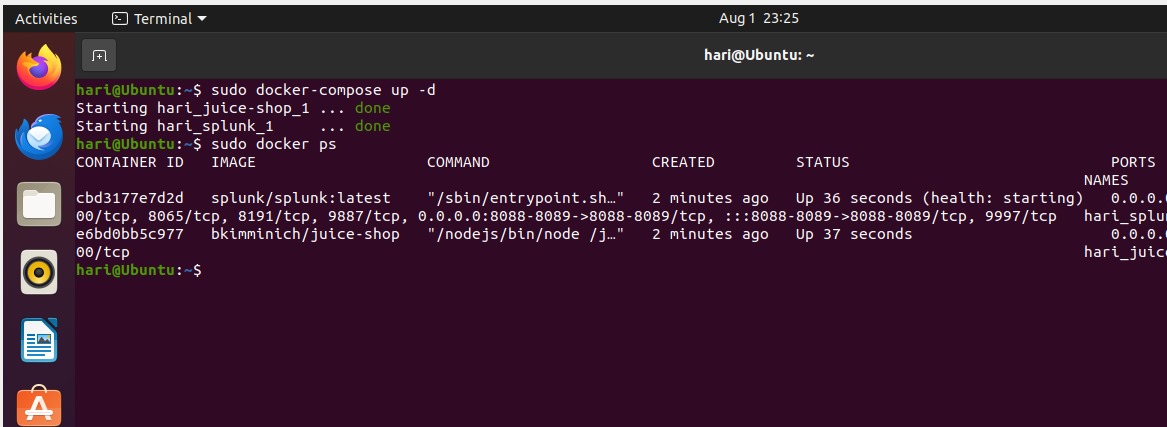
* Checking whether the splunk and juice shop application configuration is running
* Using the command: sudo docker-compose up -d



Step 4:

**Status of the Docker:**

* Checking the current status of the docker
* Using the command: sudo docker ps

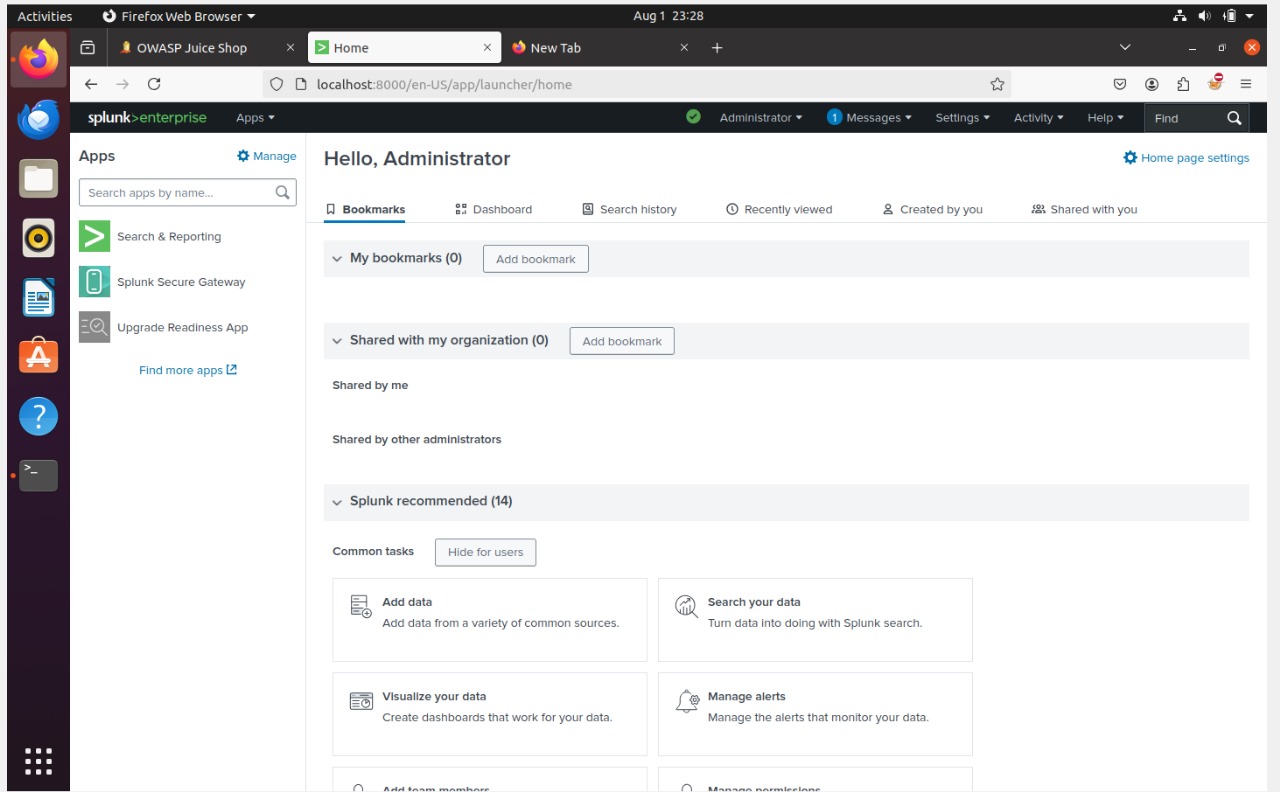


* The current status of the Docker consisting of :
* Container Id
* Image
* Command
* Created
* Status
* Ports
* Successfully Configured the Splunk and OSWAP Juice shop Application in the Docker

**5. Monitoring Logs**

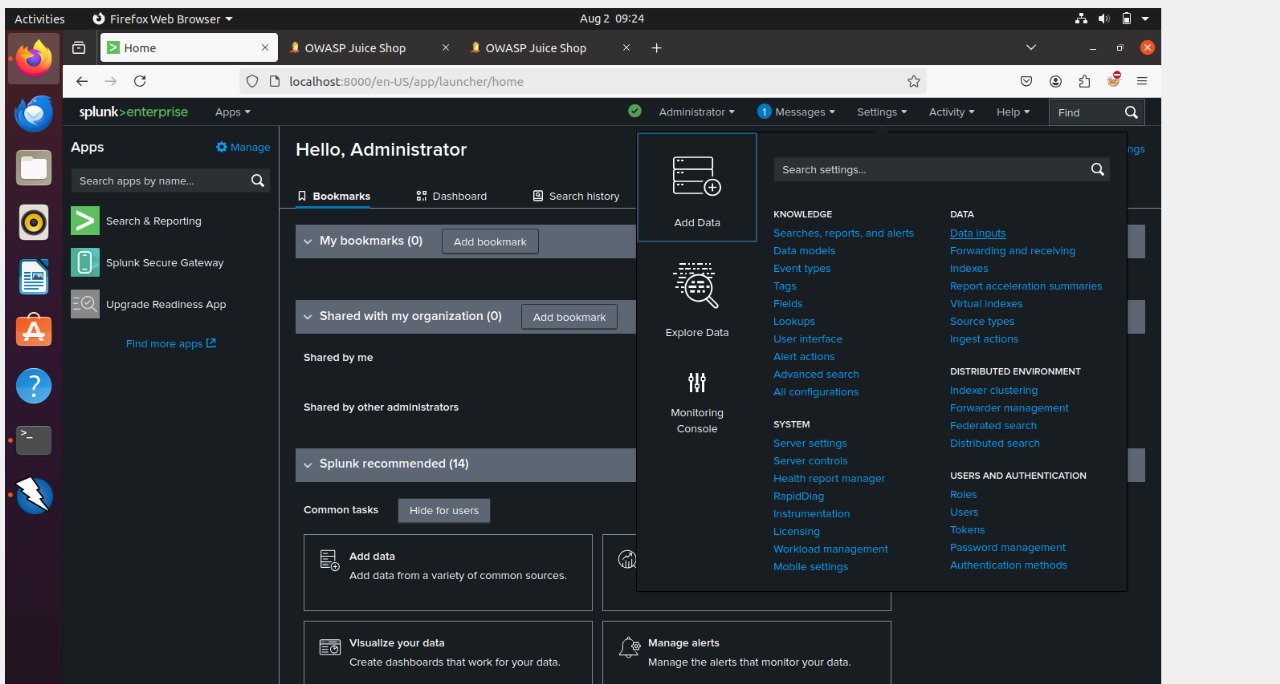
Step 1:

* Opening the Splunk in the Local host
* Local Host port number:8000
* Home Page of the Splunk has been Loaded



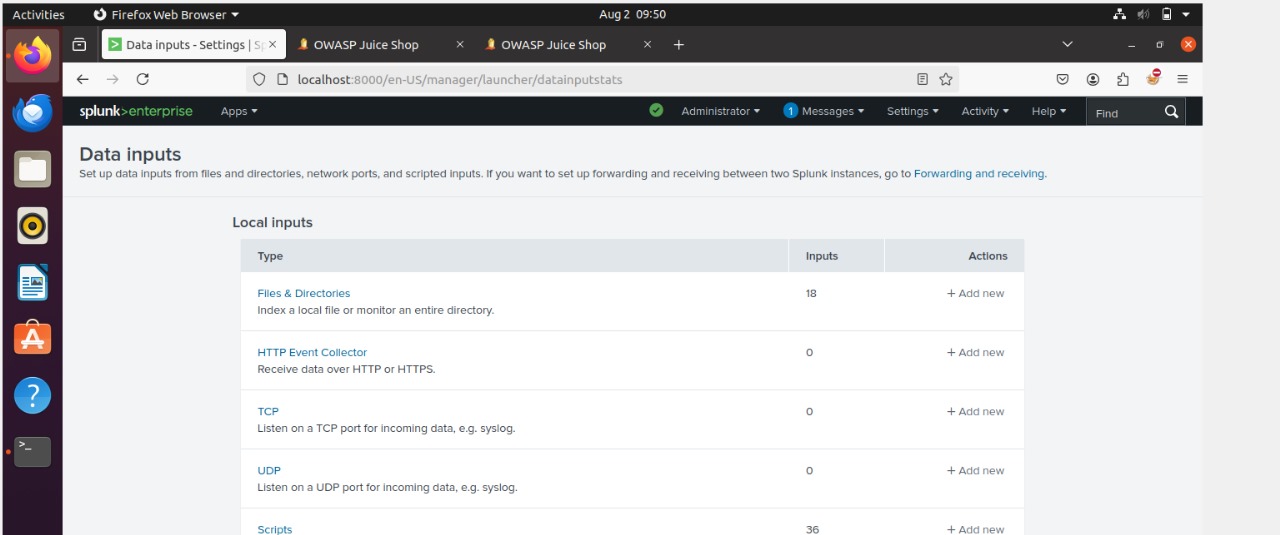
Step 2:

* Open the settings in the top Right corner
* Select the Data Input



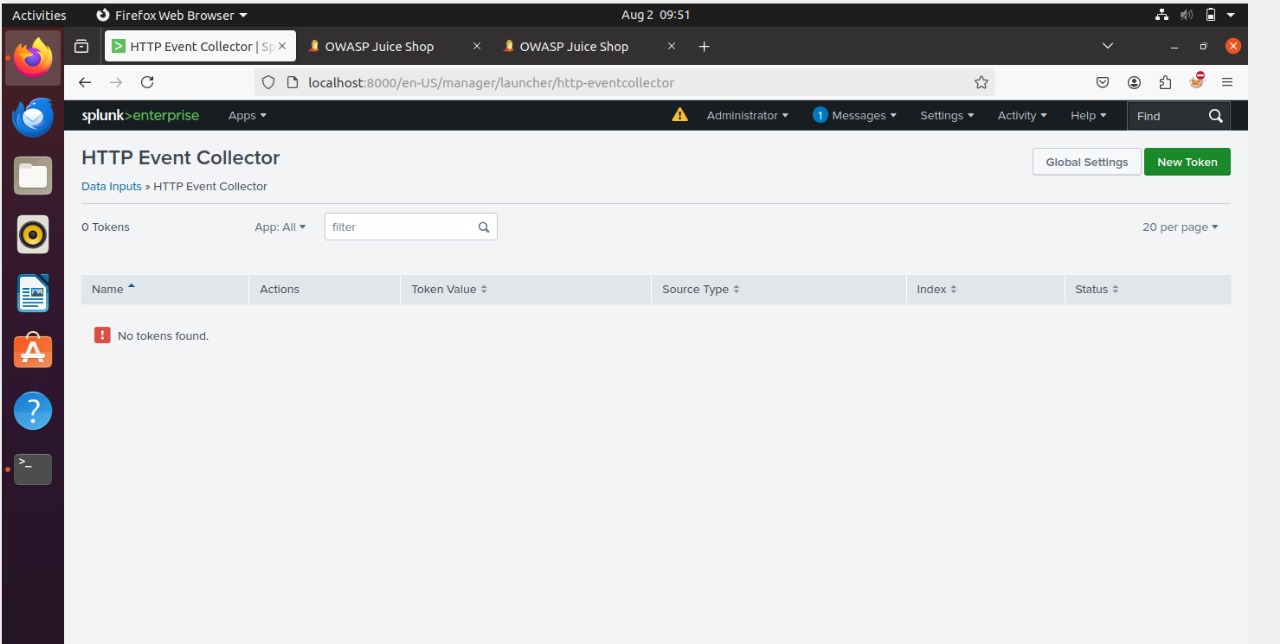
Step 3:

* In Data Input there will some Local Input
* In which select the http event collector
* Http event is used to collect the logs from the Owasp Juice Shop

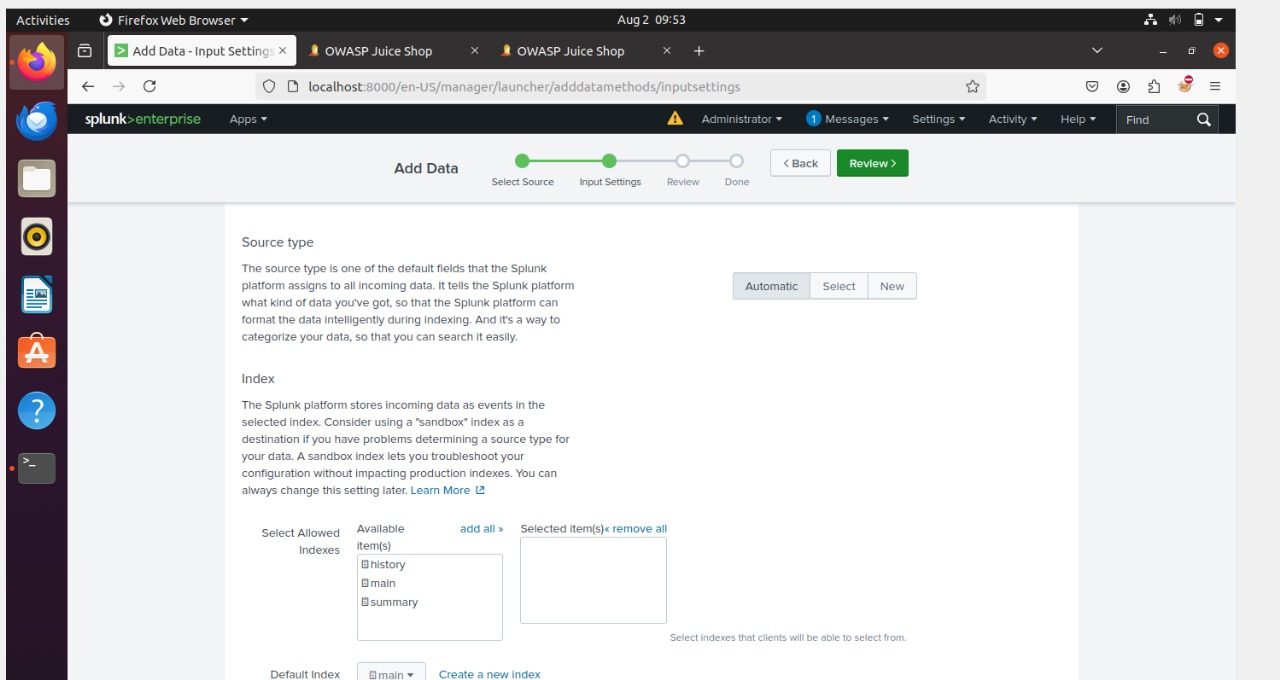


Step 4:

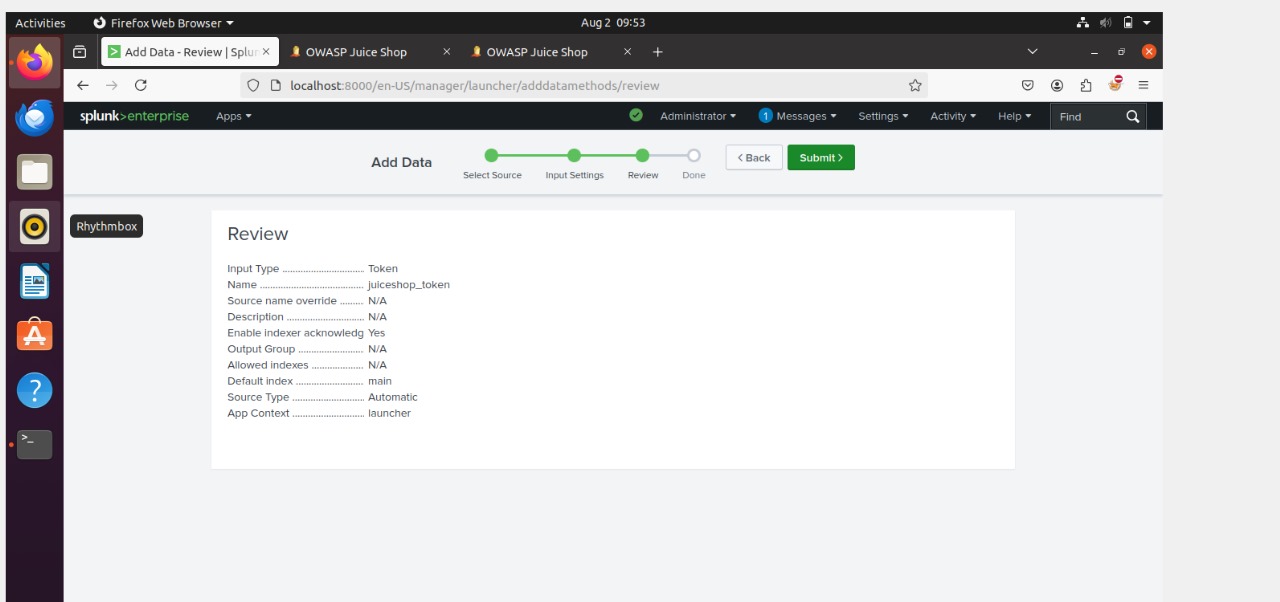
* After Opening the Http event collector there will option to add the “New token”
* New token is used for indicating the token id which the logs from website will allocated with the “New token”



Step 5:

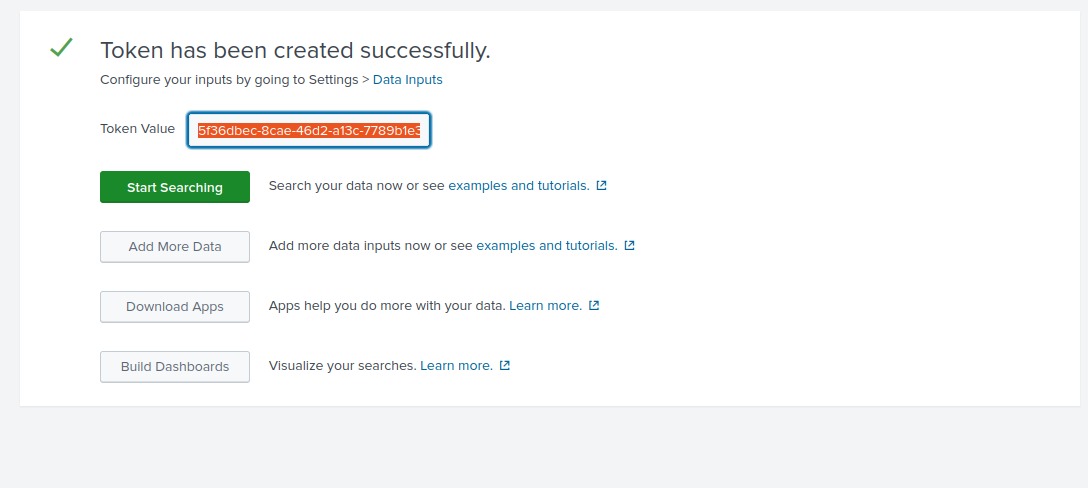


Step 6:Review



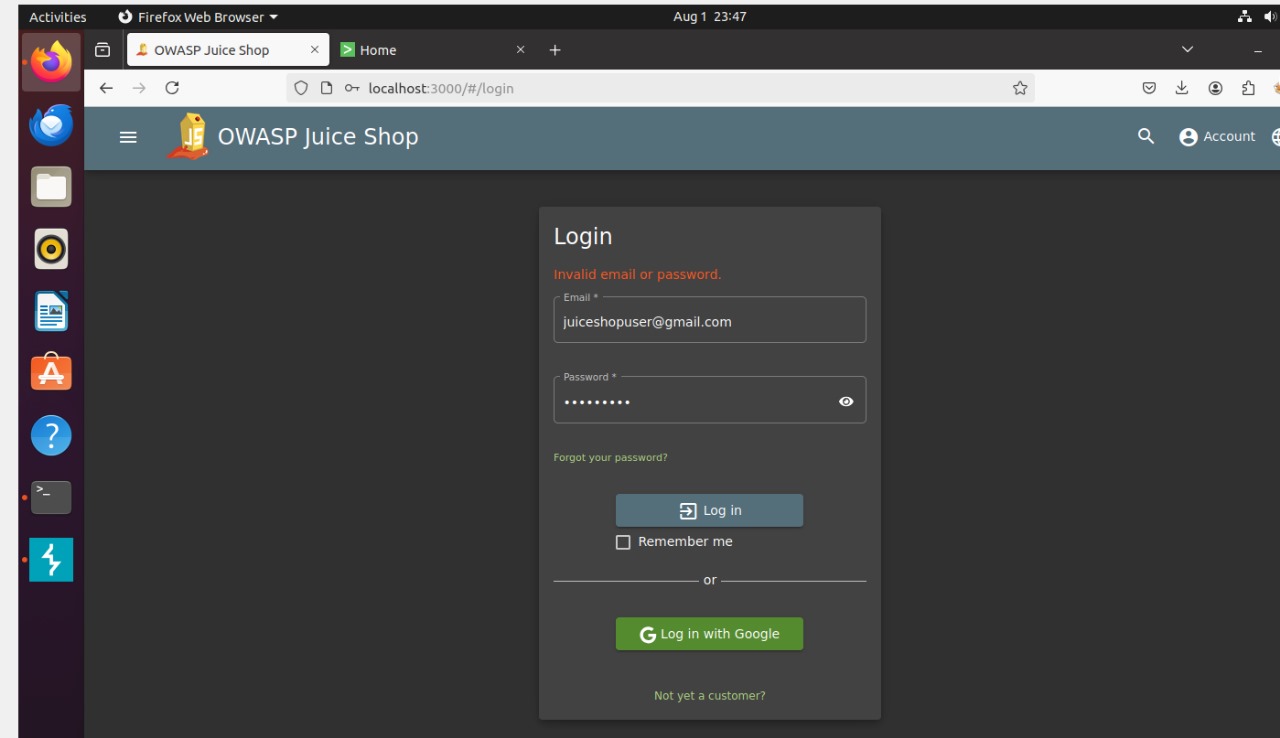
Step 7:

* Token has been created Successfully



**Step 8:**

Login credentials

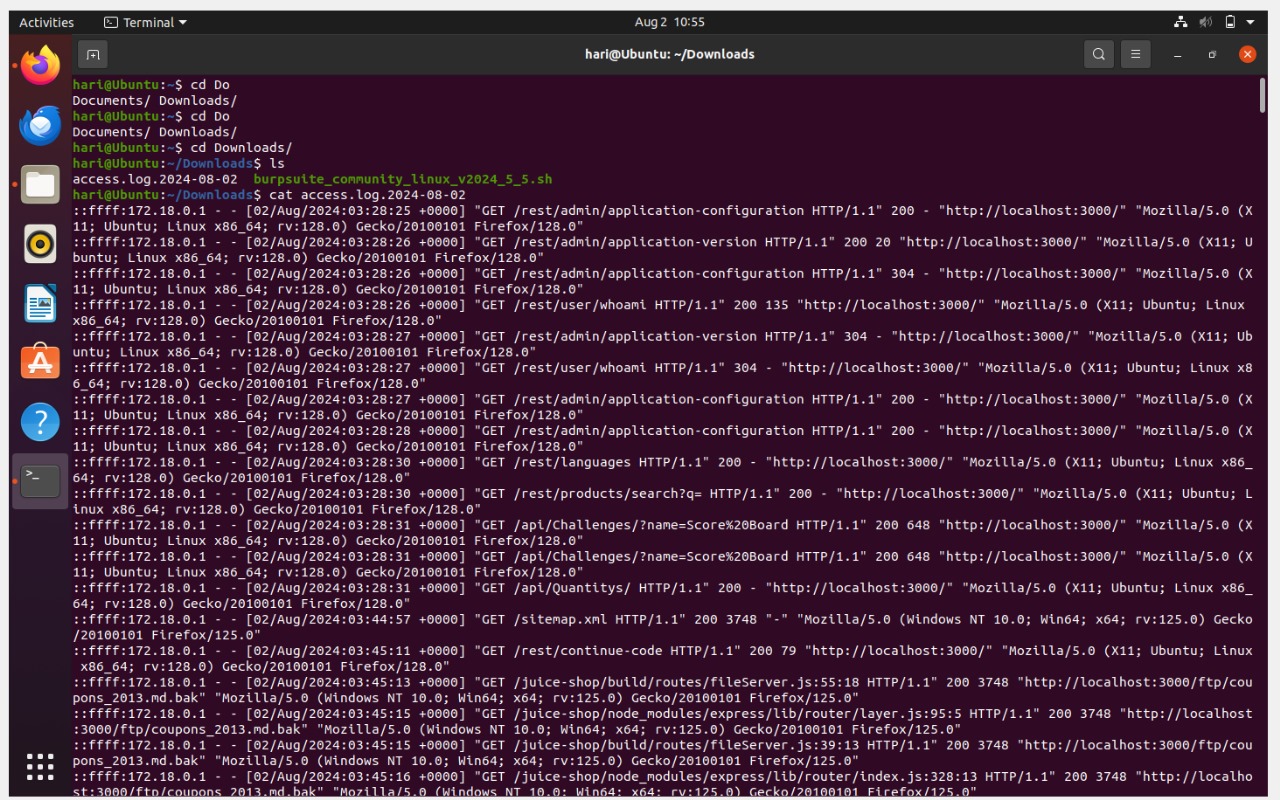


* Above image is the owasp juice shop login page where after entering the login credentials it will send the logs to splunk and we need to address the log in the splunk
* Need to check the log in the splunk and we need to check the alter using the automatic scanner tool which is “Zaproxy”.
* The log will be monitored through splunk

**Step 9:**

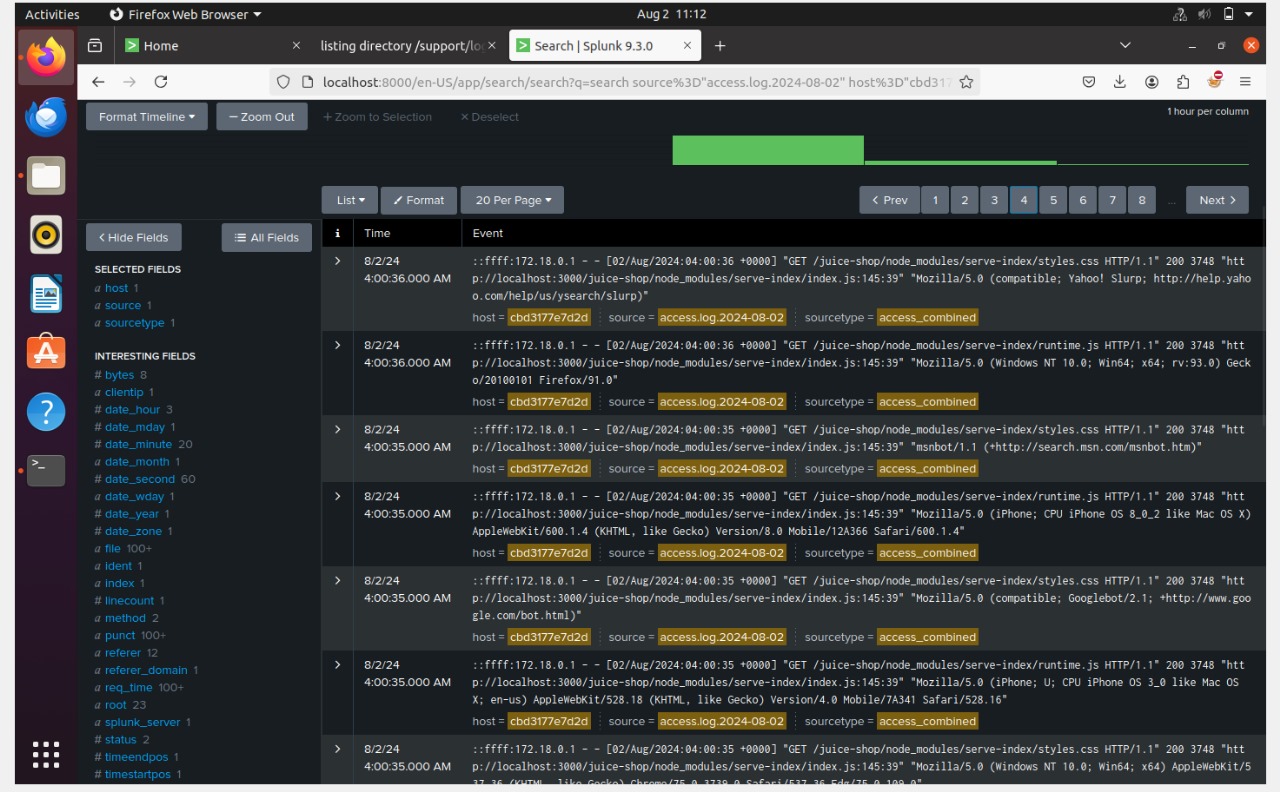
Accesing the log file

* Below image is the log source get collected in the splunk after entering the login credentials
* The logs are stored in the access.log file
* Cat command is used to see the access.log file



**Step 10:**

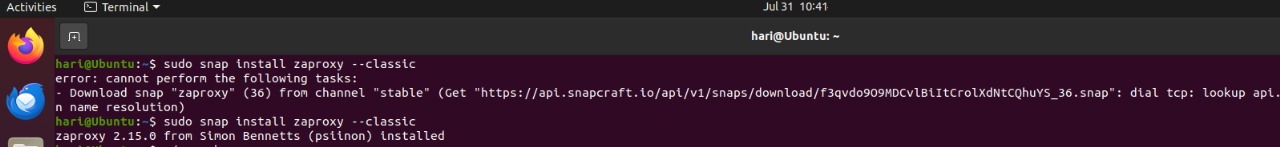
* Checking the log in the splunk
* Monitor and configure the log in the splunk



* The above is Http Event collector where we can able to see all the logs with including the time stramp when the log has been loaded etc..
* Thus the successful configuration and monitoring of the logs in the splunk

**6.Installation of Zaproxy**

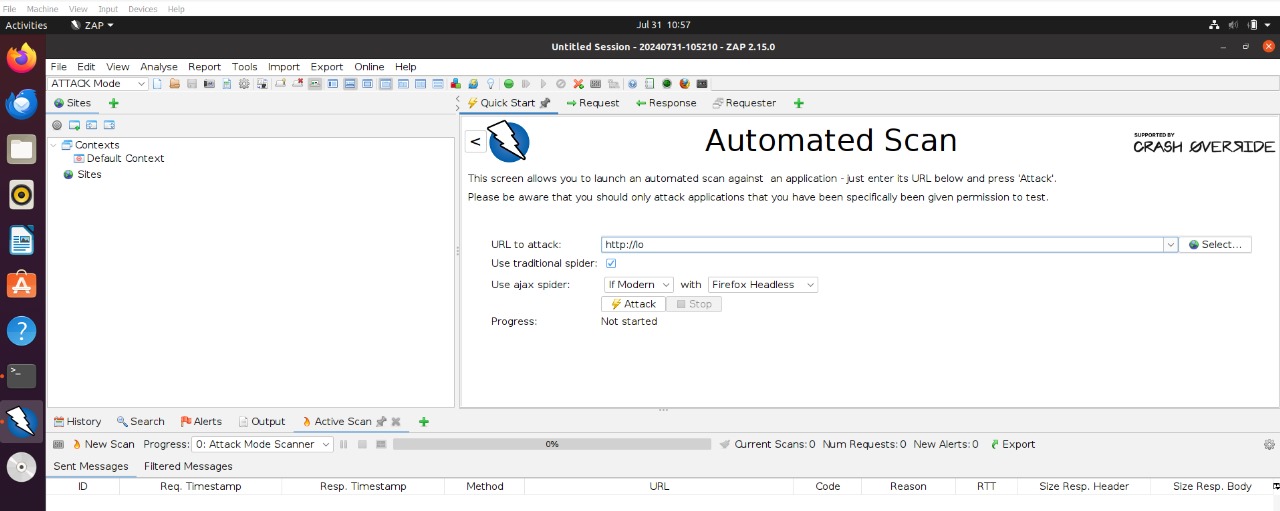
* OWASP Zed Attack Proxy (ZAP) is an open-source web application security scanner and penetration testing tool developed by the Open Web Application Security Project (OWASP). It is designed to find security vulnerabilities in web applications and services



**Interception Proxy**: ZAP can intercept and inspect HTTP and HTTPS traffic between the browser and the web application, allowing testers to see and modify requests and responses.

**Automated Scanner**: ZAP includes automated scanners that can crawl a web application and test for common vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF).

**Manual Testing Tools**: ZAP offers various tools for manual security testing, including a request editor, an active scanner, a spider for crawling, and a Fuzzer for exploring input fields.



Step by Step process for Vulnerability Scanning:

**Install and Launch ZAP**:

. Ensure ZAP is installed and then launch the application.

**Configure ZAP to Proxy Traffic**:

* Set up your browser to use ZAP as a proxy (default: localhost:8080).
* Configure SSL settings in your browser to accept ZAP’s root CA certificate if you need to intercept HTTPS traffic.

**Start ZAP**:

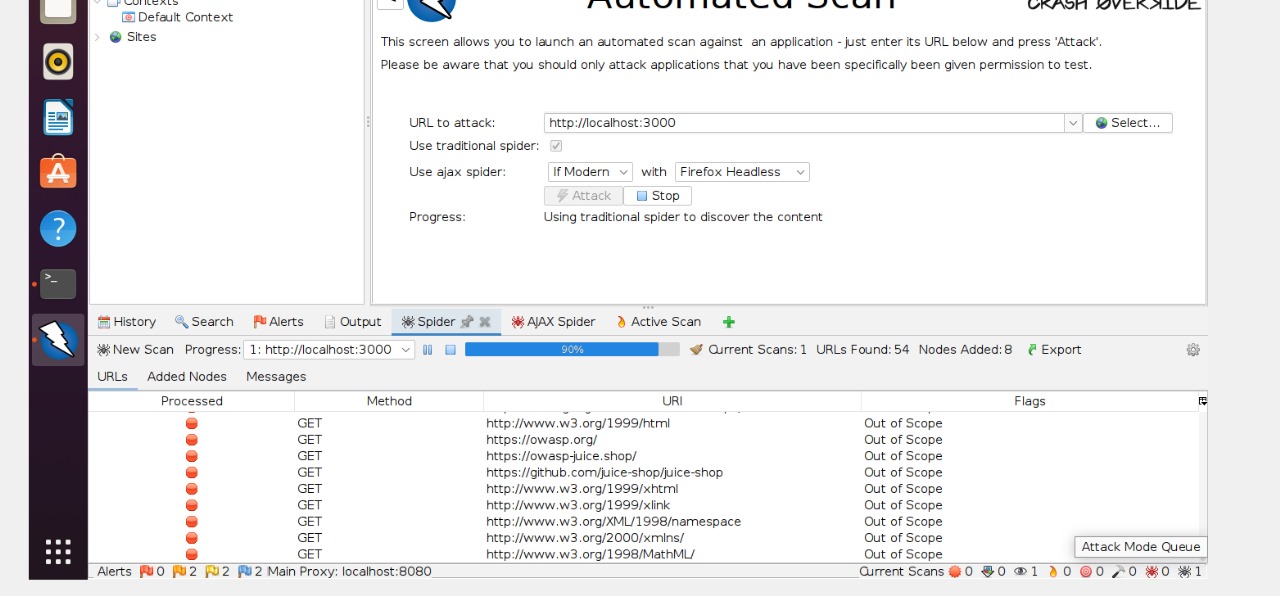
* Open ZAP and make sure it's running.

**Access Juice Shop through ZAP**:

* In your browser, navigate to http://localhost:3000 with ZAP running as the proxy. This will allow ZAP to intercept and log the traffic.

**Run Automated Scan**:

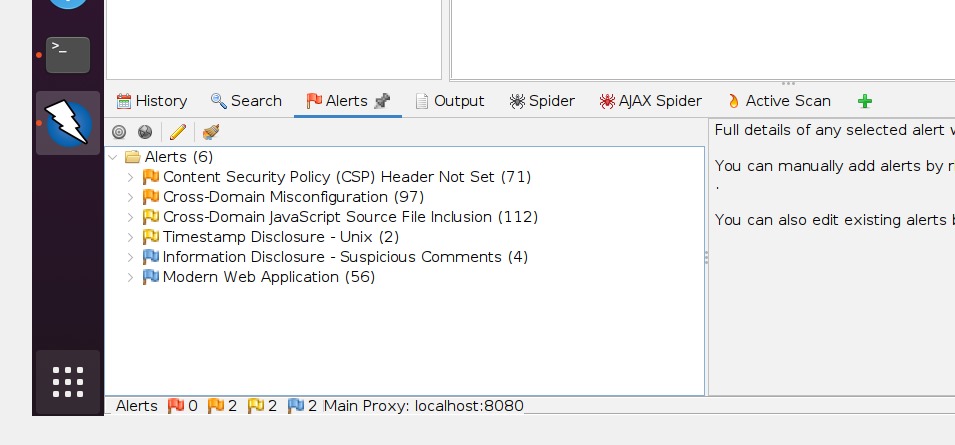
* In ZAP, right-click on the site node (e.g., http://localhost:3000) in the "Sites" tab.
* Select "Attack" > "Spider to discover content".
* After the spider completes, right-click on the site node again and select "Attack" > "Active Scan".



**Review Results**:

* Monitor the progress and results of the scan in the "Active Scan" tab.
* Analyze the identified vulnerabilities and issues reported by ZAP.

Reviewing the result by checking the high alerts



* The above image indicating the high vulnerabilities while Scanning using Zaproxy.

\*\*Conclusion\*\*

The integration of Docker, Splunk, and OWASP Juice Shop in this incident response and automation project has proven to be a highly effective approach for creating a robust and flexible security testing environment. Docker's containerization ensures consistent and isolated deployment of both the web application and monitoring tools, simplifying setup and improving reproducibility. By leveraging Splunk for comprehensive monitoring and analysis, we gain valuable insights into the security events and incidents, which enhances our incident response capabilities.

OWASP Juice Shop, with its intentionally vulnerable design, serves as an ideal platform for testing and simulating various security scenarios, allowing us to evaluate and refine our response strategies in a controlled setting. This approach not only demonstrates the practical benefits of containerization and advanced monitoring tools but also provides a valuable learning experience in managing and automating security operations.

Overall, this project successfully highlights the strengths of using Docker for efficient deployment and Splunk for powerful analytics, creating a dynamic environment that facilitates thorough security assessments and rapid incident response. The lessons learned and techniques developed here offer a solid foundation for future improvements in cybersecurity practices and automation.

Note:This is 100% of the project has been completed and reviewed by trainer.

Team Members:

1. Harishankar
2. Aswath
3. Sujarith