|  |  |
| --- | --- |
| Name: | Deekshith KR |
| Lab User ID: | 23SEK3324\_U08 |
| Date: | 10/01/2024 |
| Application Name: | Juice-shop |

**Follow the below guidelines:**





System Architecture:

(Understand the system and document the physical and logical architecture of the system, use the shapes and icons to capture the system architecture)

AWS

Ubuntu instance

Docker

Container

Web Server/juice-shop

Docker image

Define system’s normal behavior:

(Define the steady state of the system is defined, thereby defining some measurable outputs which can indicate the system’s normal behavior)

The application is accessible and available to users without downtime or disruptions. Users can navigate the site, view products and place orders without encountering errors or unresponsive pages.

Pages load reasonably fast, transactions complete without delays, and user interactions remain responsive .The application is secure, free from known vulnerabilities, and follows best practice to protect user data.

Users can browse products , add items to carts, complete purchases, and interact with various elements without encountering bugs.Continuous monitoring to identify potential issues or threats.

Hypothesis:

(During an experiment, we need a hypothesis for comparing to a stable control group, and the same applies here too. If there is a reasonable expectation for a particular action according to which we will change the steady state of a system, then the first thing to do is to fix the system so that we accommodate for the action that will potentially have that effect on the system. For eg: "If one of our database servers fails, our service will automatically switch to a backup server, and users will not experience any downtime or data loss.")



**Known**

Things we are aware of but don’t understand.

Things we are aware of and understand.

942236

**Unknown**

**Unknown**

**Known**

Things we are neither aware of nor understand.

Things we understand but are not aware of.

Juice Shop is written in Node.js, Express and Angular. It was the first application written entirely in JavaScript listed in the OWASP VWA Directory.

The OWASP Juice Shop has been created by Björn Kimminich and is developed, maintained and translated by a team of volunteers.

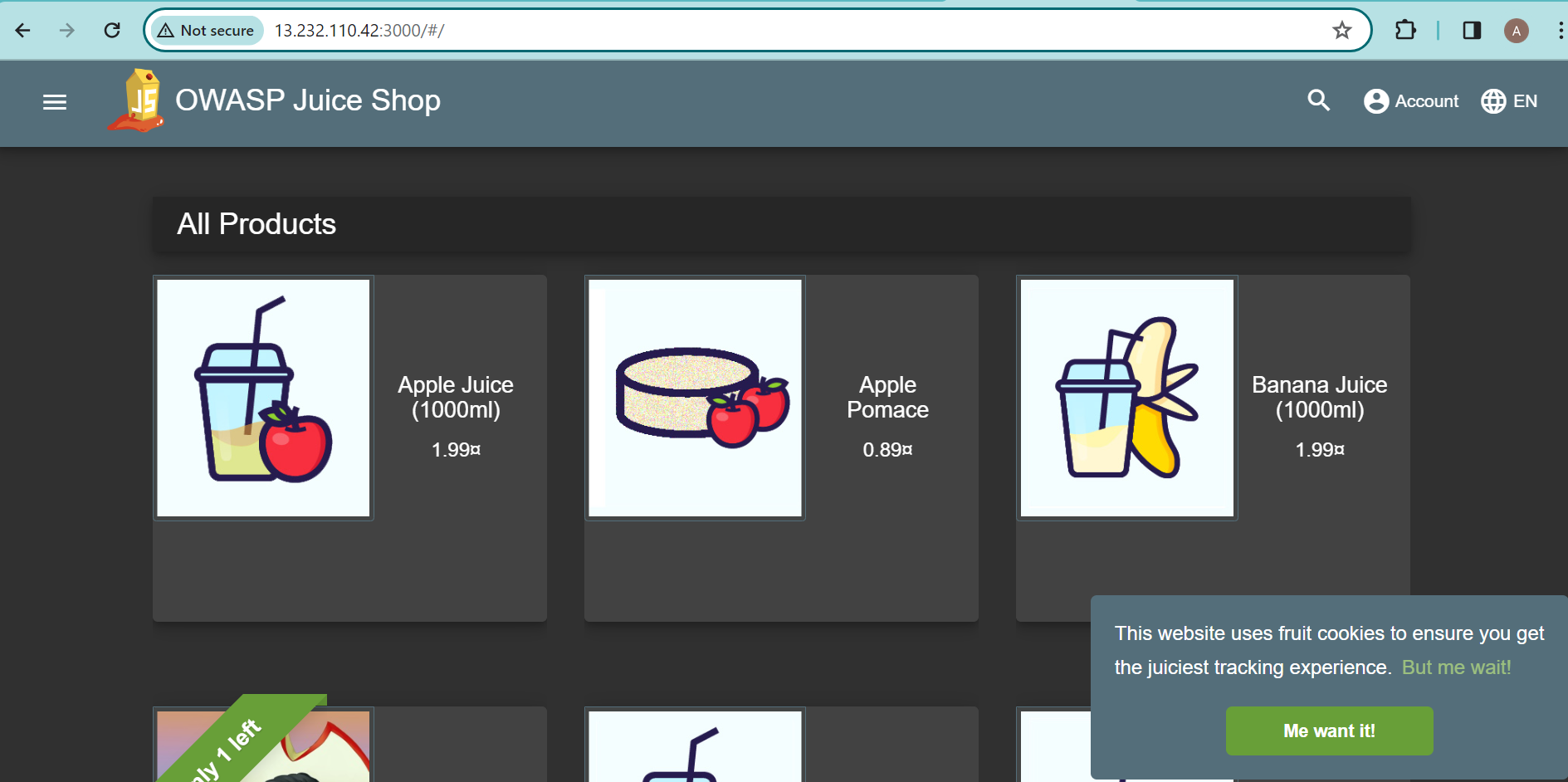
This application consists of a various levels of hacking challenges where the user is supposed to exploit the underlying vulnerabilities.

To make the uplication live

We need to the docker image , once the image is available we can the run the image with a specific name

One the container is running paste the ip address in the web browser with the port number 3000

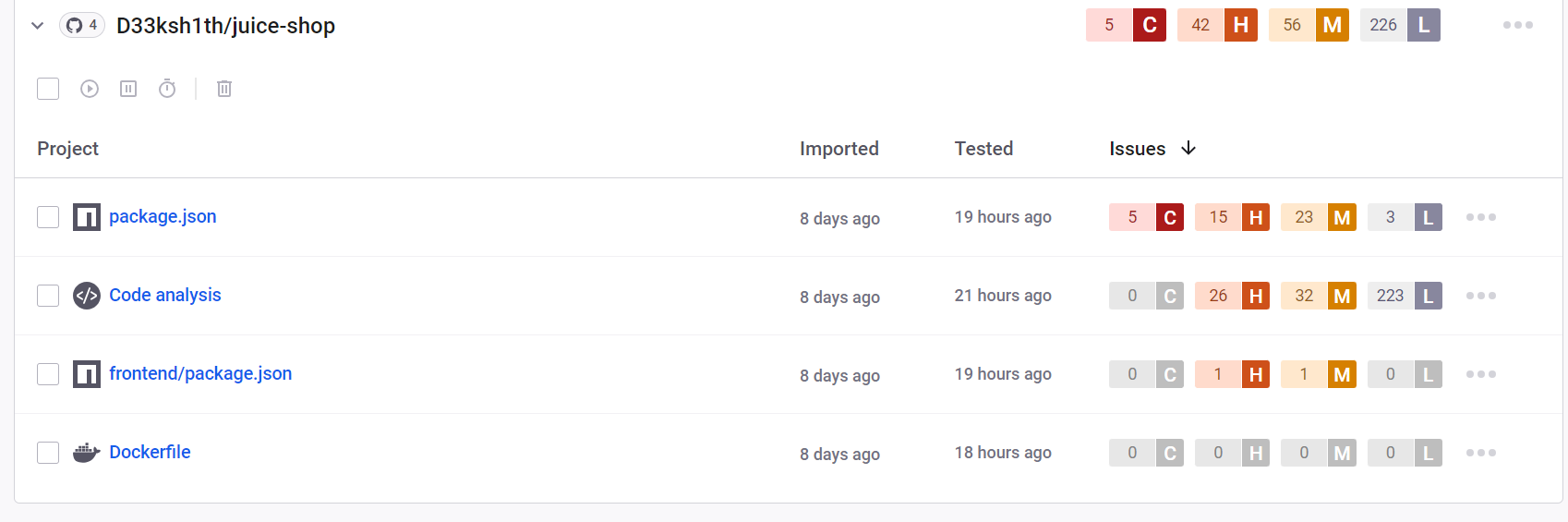
Then we shall get this page:

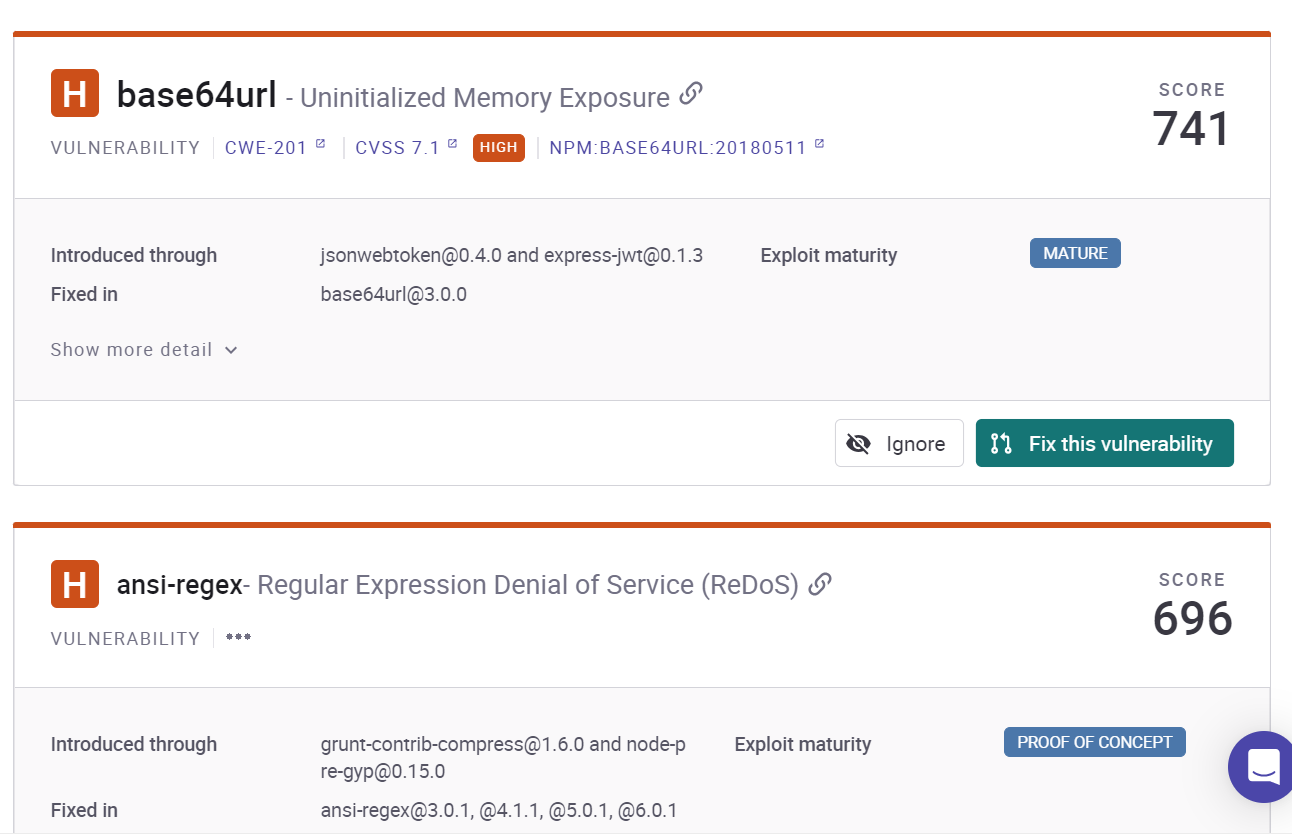


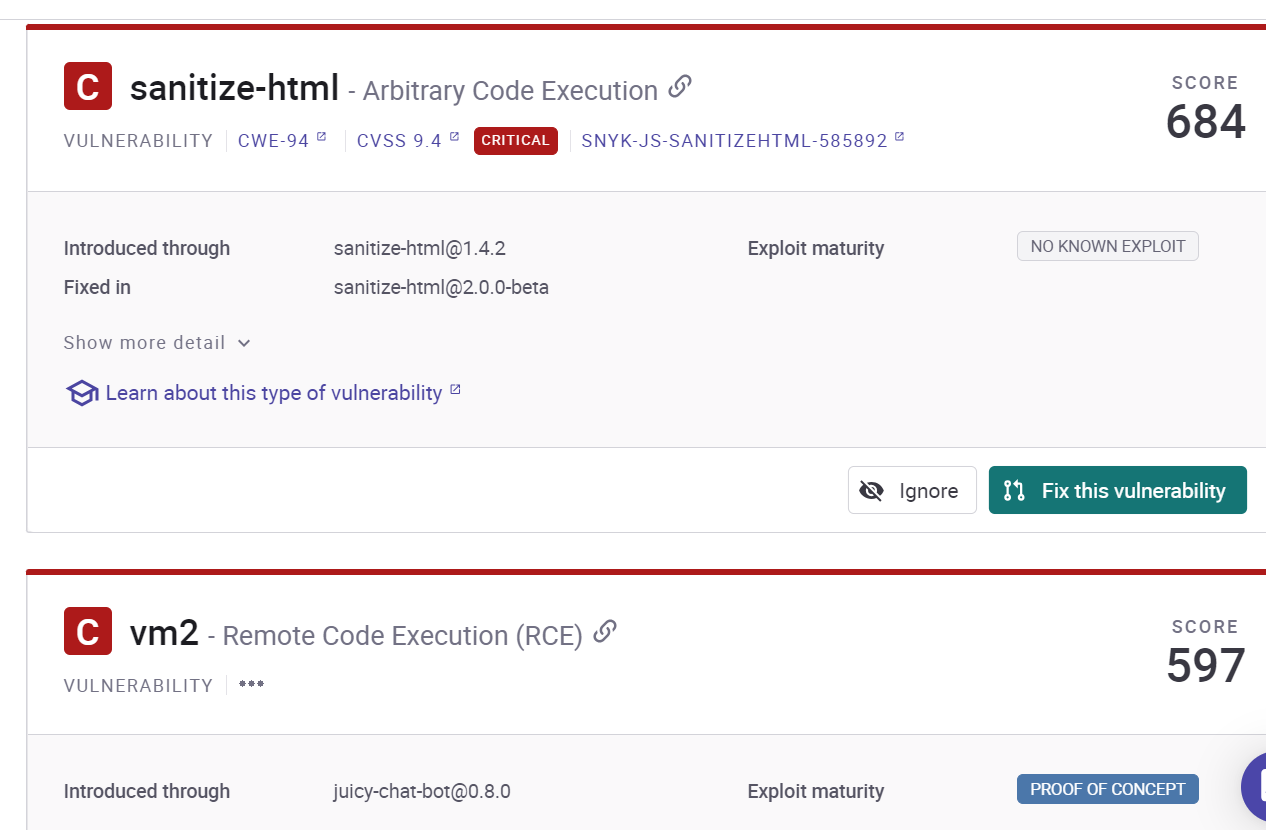
The tests we are using here is :

Snyk:

Using this testing application we can test the vulnerabilities in the juice shop web page:







sanitize-html is a library that allows you to clean up user-submitted HTML, preserving whitelisted elements and whitelisted attributes on a per-element basis. Affected versions of this package are vulnerable to Arbitrary Code Execution. Tag transformations which turn an attribute value into a text node using transformTags could be vulnerable to code execution.

vm2 is a sandbox that can run untrusted code with whitelisted Node's built-in modules. Affected versions of this package are vulnerable to Sandbox Bypass by abusing an unexpected creation of a host object based on the maliciously crafted specification of Proxy. Exploiting this vulnerability allows an attacker to gain remote code execution rights on the host running the sandbox via the Function constructor.

Next using trivy tool:

By using this took we get the tool vulnerability of

Total: 18 (UNKNOWN: 0, LOW: 11, MEDIUM: 7, HIGH: 0, CRITICAL: 0)

CVE-203-46233:

Asymmetric Private Key Exposure

Error Description:

The private key used for asymmetric encryption is exposed in the file /juice-shop/lib/insecurity.ts at line 23.

Mitigation:

Avoid storing private keys in source code repositories. ,Store sensitive keys securely outside the source code.

Use environment variables or secure key management solutions to handle private keys

.Recommendation:

Remove the private key from the source code and store it securely., Rotate the exposed key and update any dependencies that might be using it.

Error Description:

The private key exposure is exacerbated by copying the entire /juice-shop directory during the build process ('COPY /juice-shop . # buildkit').

Mitigation:

Avoid copying unnecessary sensitive files or directories during the build process.

Minimize the inclusion of unnecessary files in the final image.

CVE-2015-9235:

Use of Placeholder Public Key

Error Description:

A placeholder public key is used in case the file encryptionkeys/jwt.pub is not found.

Mitigation:

Use a proper error handling mechanism to alert developers/administrators if the public key file is missing.

Avoid using placeholder keys in production.

Experiment:

(Document your Preparation, Implementation, Observation and Analysis )