

Capstone Project: Full VAPT Engagement

Target: OWASP Juice Shop (10.49.177.96)

Author: Gyanesh Chand

Date: 26-02-2026

Tools Used: Kali Linux, Burp Suite

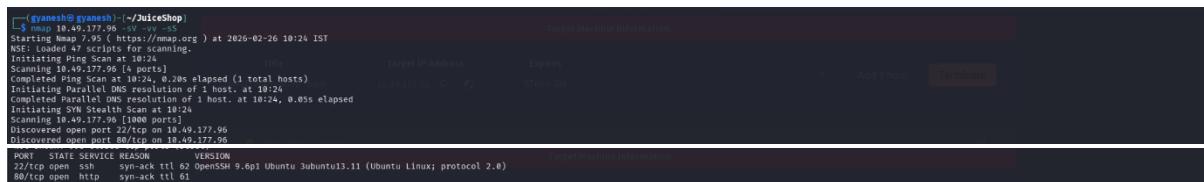
1. Executive Summary

A full Vulnerability Assessment and Penetration Testing (VAPT) engagement was conducted against the OWASP Juice Shop application hosted on the TryHackMe platform. The assessment followed the PTES (Penetration Testing Execution Standard) methodology including reconnaissance, enumeration, exploitation, and validation.

2. Reconnaissance & Enumeration

Command Used:

```
nmap -sV -sC 10.49.177.96
```



The screenshot shows a terminal window with the command `nmap -sV -sC 10.49.177.96` running. The output shows the host is up and has port 80 open. A browser window titled "Target Machine Information" is visible in the background, showing details like the target IP address (10.49.177.96), port 80, and version 8.0.0. The browser also shows a "Add 1 hour" button and a "Terminate" button.

```
[gyanesh@gyanesh:~/JuiceShop]$ nmap -sV -sC 10.49.177.96
Starting Nmap 7.7.0 ( https://nmap.org ) at 2026-02-26 10:24 IST
NSE: Loaded 47 scripts for scanning.
Initiating Ping Scan at 10:24
Scanning 10.49.177.96 [1 hosts]
Completed Ping Scan at 10:24 [0.08s elapsed (1 total hosts)]
Initiating Parallel DNS resolution of 1 host at 10:24
Completed Parallel DNS resolution of 1 host at 10:24 [0.05s elapsed]
Initiating NSE at 10:24
NSE: Starting parallel runlevel [parallel=4]
Scanning 10.49.177.96 [1000 ports]
Discovered open port 22/tcp on 10.49.177.96
Discovered open port 80/tcp on 10.49.177.96
Discovered open port 80/tcp on 10.49.177.96

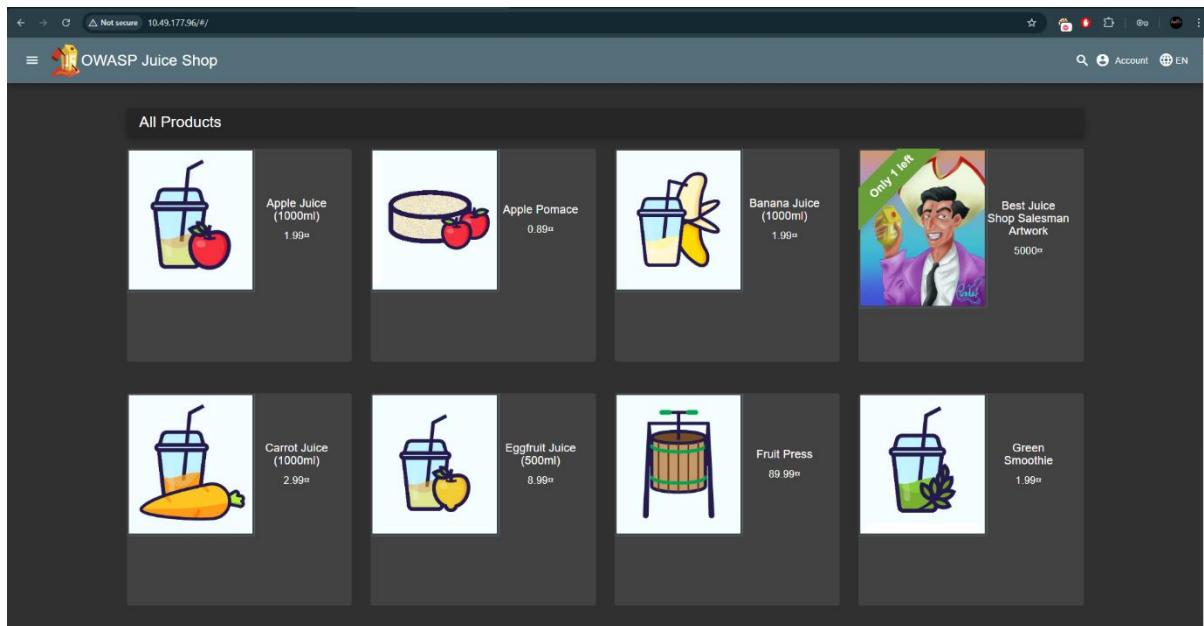
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh     syn-ack ttl 62  OpenSSH 9.6p1 Ubuntu 3ubuntu1.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http   syn-ack ttl 61
```



Findings:

- Port 22 → OpenSSH 9.6p1 Ubuntu
- Port 80 → HTTP service hosting Juice Shop

Attack surface confirmed as web-based application.



3. Web Application Analysis

The application was accessed via HTTP.

Burp Suite was configured as an intercepting proxy to analyze requests.

Identified endpoint:

/rest/products/search?q=



Intercept • HTTP history WebSockets history Match and replace ⚙ Proxy settings

⌚ Intercept on ➔ Forward | ▾ Drop Request to ht... ⚡ Open browser ⚡ :

| Time | Type | Direction | Method | URL | Status code |
|------------|-------|-----------|--------|---|-------------|
| 10:48:4... | HT... | → Request | GET | http://10.49.177.96/rest/products/search?q= | |
| 10:48:4... | HT... | → Request | GET | http://10.49.177.96/api/Quantitys/ | |

Request

Pretty Raw Hex

```
1 GET /rest/products/search?q= HTTP/1.1
2 Host: 10.49.177.96
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/134.0.6842.92 Safari/537.36
4 Accept: application/json, text/plain, /*
5 Authorization: Bearer
eyJxaiO1JKV1QilCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsInVzZXJuYWlIjoiiIwiZWih
WviOiJO2XNQHr1c3QuY29tIiwiCgfz3dvcmgiOiJmOTI1OTE2ZT13NTR1NWUwM2Y3NWRkNTThhNTczMzI1MSIsInJvbGUoiJjdXNb021ciI
sImRlbHV4ZVRva2VuIjoiiIwiBGFzdzExvZ1luSXAiOiIwljAuMC4wiIwiChJvZmlsZUltyWd1Ijoil2Fzc2V0cy9wdWjsaWMvaWhz2VzL3Vwb
G9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmVOiJoiIiwiiaXNBV3RpdmUiOnPydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDItMjYgMDU6MDY6NTQ
uOTc5ICswMDowMCIsInVwZGFOZWRBdCI6IjIwMjYtMDItMjYgMDU6MDY6NTQuOTc5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsBHOsImIhdCi6M
Tc3MjA4MjYqOX0.1gTpOpuxcI1mG8723JYuXL5xLd0z8_tt2EL2CnIJIA-UuCRJfD0vcPVYErshSSqFO-4whzKq0-O5jpvKvPUKqRx7IEf1
QUMuic_vK9IucBYwO9Ys-Lq5PcgMAIGzoFA3jn9E9imPiM2508zrUgjyN1N9K7V2Omh80013q28w
6 Referer: http://10.49.177.96/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
9 Cookie: cookieconsent_status=dismiss; language=en; token=
eyJxaiO1JKV1QilCJhbGciOiJSUzIiNiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsInVzZXJuYWlIjoiiIwiZWih
WviOiJO2XNQHr1c3QuY29tIiwiCgfz3dvcmgiOiJmOTI1OTE2ZT13NTR1NWUwM2Y3NWRkNTThhNTczMzI1MSIsInJvbGUoiJjdXNb021ciI
sImRlbHV4ZVRva2VuIjoiiIwiBGFzdzExvZ1luSXAiOiIwljAuMC4wiIwiChJvZmlsZUltyWd1Ijoil2Fzc2V0cy9wdWjsaWMvaWhz2VzL3Vwb
G9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmVOiJoiIiwiiaXNBV3RpdmUiOnPydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDItMjYgMDU6MDY6NTQ
uOTc5ICswMDowMCIsInVwZGFOZWRBdCI6IjIwMjYtMDItMjYgMDU6MDY6NTQuOTc5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsBHOsImIhdCi6M
Tc3MjA4MjYqOX0.1gTpOpuxcI1mG8723JYuXL5xLd0z8_tt2EL2CnIJIA-UuCRJfD0vcPVYErshSSqFO-4whzKq0-O5jpvKvPUKqRx7IEf1
QUMuic_vK9IucBYwO9Ys-Lq5PcgMAIGzoFA3jn9E9imPiM2508zrUgjyN1N9K7V2Omh80013q28w
0 highlights
```

⌚ Event log All issues ⚡ Memory: 123.3MB of 3.89GB ⚡ Disabled

This parameter appeared to reflect user input in the response.

Observation:

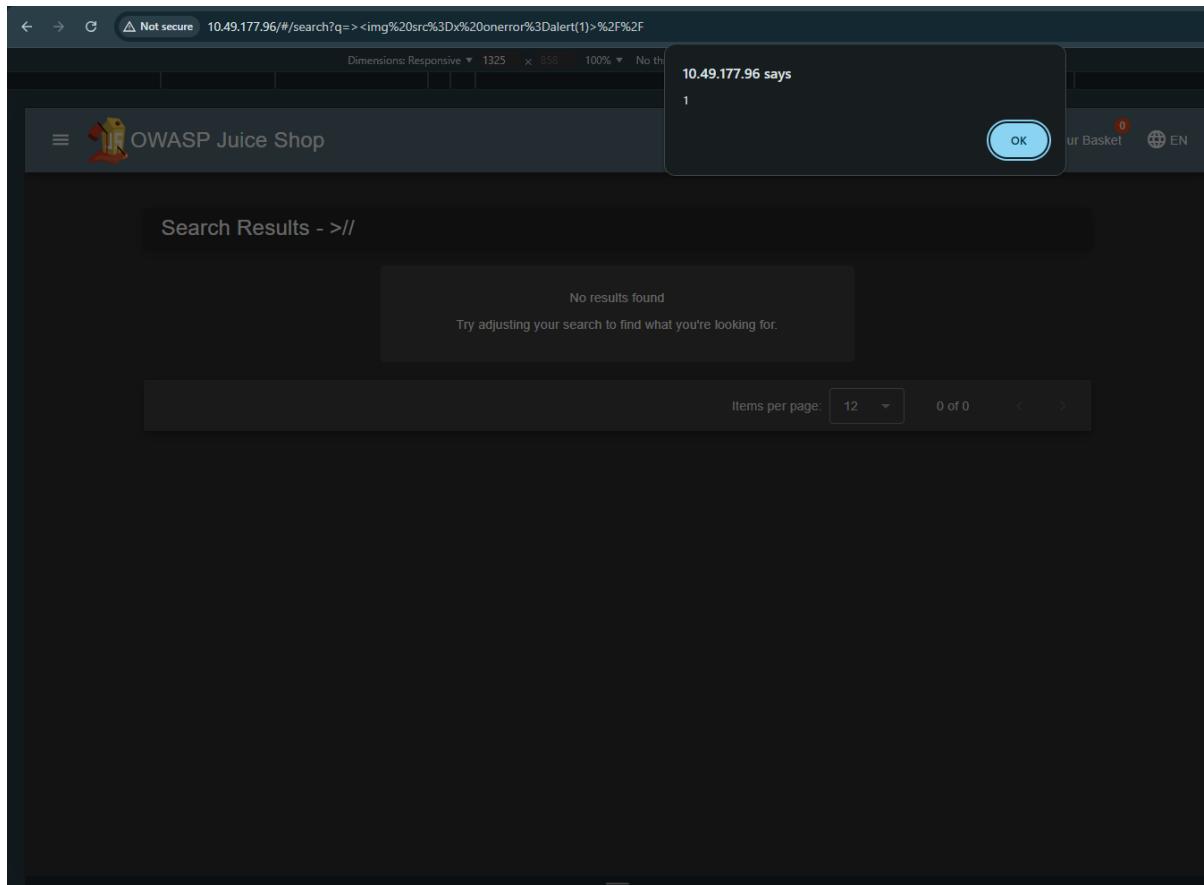
- Search input reflected in response
 - No proper output encoding

```
<span _ngcontent-ng-c3242600146 id="searchValue">hi</span> == $0
```

4. Exploitation – Reflected XSS

Payload Injected:

```
><img src=x onerror=alert(1)>//
```



The payload was URL encoded and injected into the search parameter.

The application executed the script successfully, triggering a JavaScript alert.

Technical Impact:

- Arbitrary JavaScript execution
- Session token theft possible
- DOM manipulation
- Account takeover potential

5. Remediation Plan

1. Implement strict server-side input validation
2. Apply context-aware output encoding
3. Deploy Content Security Policy (CSP)
4. Sanitize all user inputs before rendering
5. Use secure frameworks with built-in protection

Non-Technical Briefing

During the security assessment of the OWASP Juice Shop application, a critical security weakness was identified in the product search functionality. The issue allows attackers to inject malicious scripts into the application, which are executed in the user's browser. This type of vulnerability, known as Cross-Site Scripting (XSS), can enable attackers to steal login sessions, impersonate users, or manipulate displayed content.

Although no sensitive data was accessed during this controlled test, the vulnerability presents a significant business risk if exploited in a real-world environment. Immediate remediation is recommended by implementing proper input validation, secure coding practices, and browser security policies.

Addressing this issue will significantly improve the application's resilience against client-side attacks and enhance overall security posture.