PART 1: VULNERABILITY ASSESSMENT IN OWASP JUICE SHOP (BURPSUITE)

Vulnerability: SQL injection

Type: login input validation as an admin and other users in owasp juice shop

Impact:

- Potential unauthorized DB access
- Potential configuration of the website and its users
- Unauthorized access of sensitive information such as users credientials
- Potential manipulation of data by inserting, deleting or updating DB record potentially corrupting essential business data.
- Potential injection of DoS attack by overloading a DB with malicious queries.

Affected area: login page

Cross Site Scripting (XSS)

Type: client-side attack (reflection of the XSS payload on the search box).

Impact:

- sensitive users data disclosure through cookie theft, session token
- compromised user devices, data breaches or ransomware injections (xss can inject malicious scripts into web pages that distribute malware).
- Full control of the web application

Affected area: search bar

Authentication weakness

Type:

- Weak input validation
- No multifactor authentication
- Poor session management
- Poor forget password procedure

Impact:

- Brute force login
- Theft of sensitive information e.g login as an admin, stealing of users credientials
- Fraudulent transactions
- Compromised accounts of users can be used for further attacks such as spamming.

Affected area: login page

BRIEF NOTES OF EACH VULNERABILITY

SQL INJECTION:

- A fake password and email was inputted in the login and I switched on the intercept. The post request login was captured when I logged in .
- The captured post request was sent to intruder and I highlighted the email and password to add payloads from the kali wordlists. I started the attack and the sql payload " 'or 1 - -' " displayed a 401 status which I copied and pasted in the username box with the fake password and successfully logged in as admin.
- I copied a user email and logged out as an admin and inputted the user email in box with any password and I captured the post request in intercept and sent it to repeater to add a sql injection (--) beside the user email and sent it and I got authentication token in the response box.
- I added the same payload injection in the intercept and forwarded it to the browser and I was able to log in as that user.

CROSS-SITE SCRIPTING:

- I gained admin privileges through sql injection and on the search bar in juice shop I inputted "1" and went back to the intercept to capture the get request and sent it to intruder.
- In the intruder I did automatic payload site highlighting which selected the cookie get request as a payload site.
- Xss payloads were loaded and the payload %3cscript%3alert("WXSS"),%3c/script%3e returned a 200 status code which I inputted on the search bar and it reflected on the search results showing xss reflection.

AUTHENTICATION WEAKNESS

- I registered a fake account as a user and was successful without email verification. I inputted the fake login credentials and captured the post request from intercept and sent to repeater.

- I added a character “1” to the email username and added the syntax “role”,: “admin”, after the email and password and sent it and was successful.
- I went back to the log in page and inputted the modified email and password and logged in and at the search box I added administration after the owasp juice link juice-shop.herokuapp.com/#/administration and was logged in as an admin.s

Part 2: Secure Network Design

Segment the network into zones:

- Public Zone: Hosts Juice Shop’s web interface.
- Internal Zone: Secure database servers isolated from public access.

Apply security controls:

- Firewalls: Filter traffic between zones.
- Intrusion Prevention Systems (IPS): Detect and block malicious activities.
- Web Application Firewall (WAF): Protect against SQLi, XSS, and more.
- Secure Configuration: Disable unnecessary services, enforce HTTPS, and limit ports.

Deliverable: Network Design Proposal

Firewall 1: Filters all external traffic into the DMZ.
Web Application Firewall (WAF): Deployed before the Juice Shop server.
Database Server: Placed in the internal zone, isolated from public exposure.
Monitoring Tools: For log analysis and alerting.

Descriptions of Components:

Firewall:

Purpose: Controls incoming and outgoing traffic
Justification: prevents unauthorized access to internal and DMZ zones

WAF:

Purpose: protects the juice shop web server from web based attacks

Justification: mitigates xss, sql injection and other injection attacks.

Database Isolation

Ensures the database is not directly accessible from the public internet

Ensures the database is not directly accessible from the public internet

IPS

Monitors and blocks suspicious activities

Stops brute-force or other exploitation attempts in real time.

TLS Encryption

Encrypts data in transit

Ensures secure communication between the client and server.

MFA:

Sets OTP for users both in registration and log in

Verifies users' credentials through email verification.

Recommendations:

Regularly patch and update software.

Conduct periodic vulnerability assessments and penetration tests.

Educate users on secure practices (e.g., password management).

Text-Based Network Design

Public Zone:

Internet Router: Connects to the external internet.

Firewall 1: Filters all incoming and outgoing traffic from the internet to the DMZ.

DMZ (Demilitarized Zone):

Web Application Firewall (WAF): Protects the web server from web-based attacks like SQLi and XSS.

Web Server: Hosts the Juice Shop application and handles user requests.

Internal Zone:

Firewall 2: Filters traffic between the DMZ and Internal Zone.

Database Server: Stores sensitive application data (e.g., user accounts and orders).

Monitoring Server: Analyzes logs, monitors traffic, and raises alerts for unusual activity.

Intrusion Prevention System (IPS): Monitors and blocks malicious activities in real-time.

Traffic Flow:

1. from Public to DMZ:

   Internet traffic flows through the router to Firewall 1.

   Only allowed HTTP/HTTPS requests pass to the WAF, which filters malicious payloads.

   Safe traffic is forwarded to the web server.

2. from DMZ to Internal Zone:

   The web server communicates with the database server through Firewall 2.

   Only whitelisted requests (e.g., SQL queries) are allowed.

3. Internal Monitoring:

   The monitoring server reviews logs from the web server, database, and firewalls.