Table of Contents

Threat Model

Description

This ecommerce application is the implementation of a website which can support the following functions.

1. Cart Management
2. Customer Management
3. Notification Management
4. Payment Management
5. Order Management
6. Product Search

Currently, our application can support only a single type of user which is a customer. The customers can register/login, update their details and they will be able to access some parts of this application only after logging in. The functions like cart management, customer management, notification management and order management can only be performed when a user is logged in. Otherwise, a guest user can just come and search/browse for a product.

Assumptions that can be checked or challenged in the future as the threat landscape changes

1. Currently, we have a single user Customer but, in the future, we might need to add users like Admin/Seller.

Assets in Our Ecommerce Application

1. Application Server
2. Database Server
3. Data in the database
4. Source Code
5. JS files in CDN
6. User information

Attack Surface

1. The Server
2. The database
3. The Firewall
4. The APIs
5. The Login screens

Potential threats to the system

1. DoS and DDoS attacks
2. Direct access Attacks
3. Malware
4. Bots
5. Brute Force
6. Man in the Middle Attack (MITM)
7. SQL Injections
8. Express Language Injection
9. Command Injection
10. Data Protection
11. Cross Site Scripting
12. Content Delivery Networks
13. Privilege Escalation

Actions that can be taken to mitigate each threat

1. DoS and DDoS attacks
    a) We can have a global network of scrubbing centers that scale, on demand, to counter multi-gigabyte DDoS attacks.
    b) We can have a protocol check to remove “bad” traffic before it can even reach the servers.
    c) We can add captchas, cookie challenge etc. for challenging suspicious or unrecognized entities.
    d) We can use traffic monitoring for monitoring the traffic and act for unusual traffic levels or unrecognized IP addresses.

2. Direct access Attacks
    a) Keep your systems and softwares up to date. We should keep a track of any vulnerabilities in the softwares that we use and update or apply patches as soon as possible to our softwares and systems.
    b) Use VPNS. We should use VPNs for a secure connection.
    c) Install Firewalls. We should use firewalls for added network security.
    d) Control access to our systems. We should control the user access and should have a restricted machine/user access.
    e) Wifi Security. We should have a strong wifi security.
    f) Having a solid access management.

3. Malware
    a) Scan your sites regularly. We should scan our sites regularly for any malwares/trojans.
    b) Take regular backups. We should take regular backups of our data and we ready for any data loss.

c) Perform regular updates. Regular updates should be performed to be safe from any vulnerabilities
d) Upgrade your hosting plan.
e) Use SSL and HTTPS. SSL and HTTPS are used for secure web exploration, and we should use them.
f) Use and enforce Secure Passwords. We can enforce the various password policies.
g) Install WAF. Web Application Firewall should be installed to strict access to our system.

4. Bots
   a) Block or CAPTCHA outdated user agents/browsers.
   b) Block known hosting providers and proxy services.
   c) Protect every bad bot access point
   d) Carefully evaluate traffic sources.
   e) Investigate traffic spikes.
   f) Monitor for failed login attempts.
   g) Pay close attention to public data breached.

5. Brute Force
   a) Use Strong Passwords
   b) Limit login attempts.
   c) Monitor IP addresses.
   d) Use 2FA.
   e) Use CAPTCHAs.
   f) Use unique login URLs.
   g) Disable root SSH logins.
   h) Use WAF.

6. Man in the Middle Attack (MITM)
   a) Stop using public networks.
   b) Use VPN.
   c) Secure e-mails by employing SSl/TLS.
   d) Regular audit of networks and devices.
   e) Use latest version of secure browsers.
   f) Use browser plugins like ForceTLS.
   g) Separating Wi-Fi networks for guests.

7. SQL Injections
   a) Use Prepared statements
   b) Use properly constructed stored procedures
   c) Proper input validation
   d) Escape all user supplied input
   e) Enforce Least Privileges
   f) Using whitelist checks instead of blacklist checks

8. Express Language Injection

a) Avoid incorporating user-controllable data into dynamically evaluated code.
b) Data should be strictly validated

9. Command Injection
    a) Avoid system calls and user input
    b) Set up input validation
    c) Create a white list
    d) Use only secure APIs
    e) Use execFile() securely

10. Data Protection
    a) Enforce Password Complexity and length
    b) Look for vulnerabilities
    c) Have backups for the data
    d) Install a firewall
    e) Encrypt sensitive data
    f) Monitor database activity

11. Cross Site Scripting
    a) Don’t trust any user input
    b) Use escaping/encoding
    c) Sanitize HTML
    d) Set the HttpOnly Flag
    e) Use a Content Security policy

12. Content Delivery Networks
    a) Add integrity checks for all the data used from CDN.

13. Privilege Escalation
    a) Keep accounts up to date with comprehensive privilege account management
    b) Patch and update software
    c) Perform vulnerability scans
    d) Monitor network traffic and behaviour
    e) Institute a strong password policy

Security concerns into the STRIDE Model

Spoofing

1. Man in the Middle Attack (MITM)

Tampering

1. Malware
2. SQL Injections
3. Express Language Injection
4. Command Injection
5. Content Delivery Networks

Repudiation

1. Bots
2. Brute Force

Information Disclosure

1. Direct access Attacks

Denial of service (DoS)

1. DoS and DDoS attacks

Elevation of privilege

1. Data Protection
2. Cross Site Scripting
3. Privilege Escalation

Mitigation process is already specified with the attacks.

Security Points implemented in POC

1. XSS => We have implemented checks for Cross Site Scripting in all the jsp pages. All the contents of the page are encoded for scripting security.

2. Content Security Policy => **Content Security Policy** (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross-Site Scripting (XSS) and data injection attacks. We have added the meta tags for this

3. Roles => We have added Security roles for the users. In our application the user can have three roles; SELLER, USER and ADMIN.

4. Captcha => Login and Registration screens have captcha enabled. So, captcha will be verified before interacting with the database.

5. Referrer Policy => We have added strict-origin policy.

6. HSTS => We have added *Strict-Transport-Security: max-age=31536000; includeSubDomains; preload* in the server configuration.

7. X-Frame options => We can configure X-frame options in our security file.

8. Multiple Failure Attempts => We are locking the user account for 1 hour if incorrect password is entered consecutively for 3 times.

9. JWT => We are using JWT for authorisation with both access_tokens and refresh_tokens. The refresh_token can be used to generate a new token.

10. Encryption => We are encrypting the passwords in the database using Bcrypt.

11. Validations => We are performing validations at both client and server side for added security.

12. CSRF => We have disabled CSRF by default.