E- commerce application decomposed

Owner: Kiiru
Reviewer:
Contributors:
Date Generated: Wed Oct 23 2024

OWASP Threat Dragon

Executive Summary

High level system description

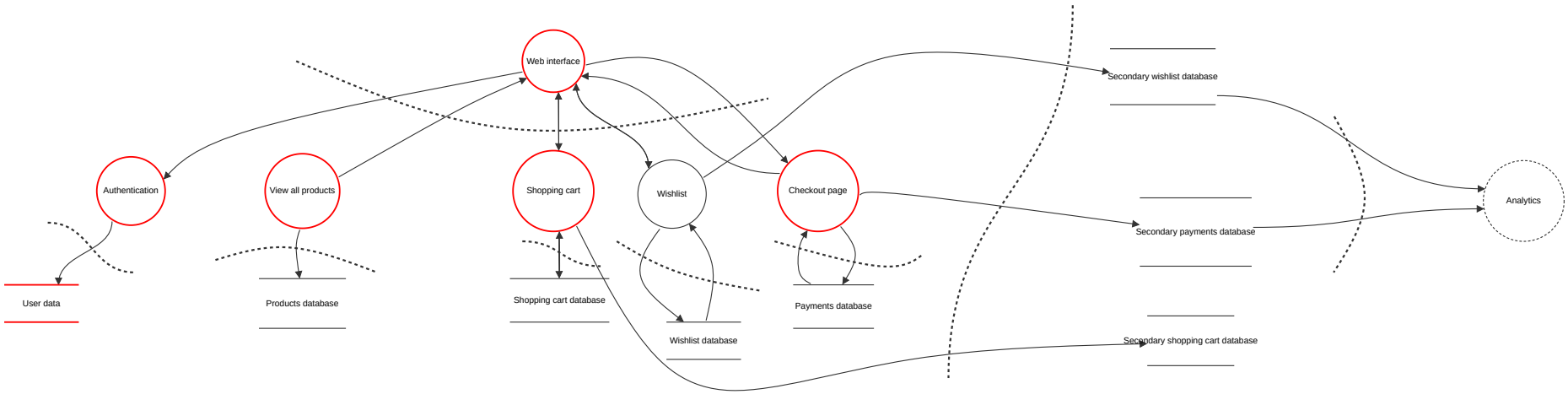
PASTA Threat modelling: Cyber shujaa

Summary

Total Threats	16
Total Mitigated	0
Not Mitigated	16
Open / High Priority	0
Open / Medium Priority	15
Open / Low Priority	1
Open / Unknown Priority	0

E-commerce application

Test



E-commerce application

View all products (Process)

Ability to view all products from a generic page and select specific products.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
4	SQL injection	Information disclosure	Medium	Open		Ability to view "hidden" products by making arbitrary SQL commands.	Utilizing the Django ORM to prevent use of raw SQL queries to the database that may disclose "confidential" information.

Shopping cart (Process)

Users can add or remove products from the shopping cart

Number	Title	Type	Priority	Status	Score	Description	Mitigations
10	Race conditions	Non-repudiation	Medium	Open		Threat actors may use coupons twice within a short period of time (ms) to get double offers so if I use two 5% coupons that is meant to be used once.	Avoid having a shared state, we can use message brokers as messages can be queued.
11	Price manipulation	Tampering	Medium	Open		Intercepting and manipulation of HTTP requests to the shopping cart API, adjusting product prices during the checkout process leading to financial losses.	Validation of prices on the server side and comparing the UI price to the prices stored in the database.
12	Insecure Direct Object Reference (IDOR)	Spoofing	Medium	Open		A threat actor can be able to view other user's carts by manipulating identifiers in the API requests.	1. Implementation of authorization logic to check if the user has permissions to access or modify the requested resource.
13	Negative purchases	Spoofing	Medium	Open		Threat actors may try to manipulate quantities and price by tampering with request data leading to fraudlent transactions.	1. Implement server-side validation for cart-related data.

Checkout page (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
14	Insecure transmission of payment data.	Information disclosure	Medium	Open		Inadequate security measures can lead to unauthorized access to sensitive user data such as credit card numbers.	Encryption of data in transit.
15	Insecure payment processing.	Non-compliance	Medium	Open		Using unsecured payment gateways or ourdated libraries can lead to not complying with PCI-DSS can expose users to fraud and data theft and reputation damage in the case of a breach and massive fines.	1. Use of updated payment libraries. 2. Compliance to PCI-DSS. 3. Use of a reputable Payment Service Provider (PSP).

Web interface (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
3	Injection attacks	Tampering	Low	Open		Ability to conduct attacks such as cross site scripting and SQL injection, XSS,Cross site Request Forgery and path traversal.	Django provides protection against attacks such as: 1. SQL injection through the Django ORM. 2. Cross site scripting by automatically escaping variables in templates by default. 3. Default CSRF protection attacks. 4. Reduces the risk of path traversal attacks by sanitizing user input when interacting with the file system.

Authentication (Process)

Number	Title	Type	Priority	Status	Score	Description	Mitigations
5	Ability to perform bruteforce attacks	Spoofing	Medium	Open		A threat actor can perform a brute force attack to and gain access to user accounts that have weak passwords.	1. Implementation of MFA as an extra step of authentication. 2. Adoption of a password policy. 3. Using a reputable and actively maintained third party authentication service.
6	Credential stuffing	Spoofing	Medium	Open		Use of credentials retrieved from previous data breaches to attempt to perform logging in. This may lead to user account compromise.	Implementation of MFA and use CAPTCHA for failed login attempts.
7	Phishing	Elevation of privilege	Medium	Open		Use of deceptive methods to get valid credentials from users.	Educate users about phishing.
8	Session hijacking	Spoofing	Medium	Open		Attackers can steal valid session cookies which may be used to impersonate users without needing a password.	Have cookies with the Secure flages, session expiration and require reauthentication for sensitive action.
9	Insecure credential recovery procedures	Elevation of privilege	Medium	Open		Use of flawed steps for users to recover their passwords may lead to account compromise using the forgot password functionality.	Implementation of secure and sound password recovery methods.

User data (Store)

All data about the personal users is stored here.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
16	Improper access controls	Spoofing	Medium	Open		Improper access control practices can allow unauthorized users to gain access to sensitive data.	Implementation of proper access control practices.
17	Improper configuration	Spoofing	Medium	Open		Misconfigured database settings can expose sensitive information or leave the database vulnerable to attacks.	Implementation of database hardening practices.
18	Data loss	Availability	Medium	Open		Without proper backup, there is risk of data loss as a result of accidental deletion and hardware failures.	Have a backup or set up the database in a way that there is High Availability.