***AATT Security Requirements***

October 10, 2021

Developed By: Arielle Collins, Alexander Marquis, Tony DePlato, and Tim Oehrlein

**Table Of Contents**

1. INTRODUCTION
   1. Background
   2. Purpose
   3. Assessment Overview
   4. Identifiers
   5. Risk Rating Scale
2. REQUIREMENTS
   1. Security Requirements
   2. Security Requirements Matrix
3. ASSEMENTS
   1. Costs
   2. Threats
   3. Vulnerabilities
   4. Exploits
   5. Mitigation
   6. Design
   7. Implementation
   8. Verification
4. CONCLUSIONS

APPEDIX A

1. INTRODUCTION

Security Matrix Identifiers:

* ID: Requirement Identifier
* Title: Requirement Name
* Security Concern: Description of the security concern
* Threat: Description of threat mentioned
* Vulnerability: Description of the vulnerability
* Risk Rating: Rating system on a 1-5 scale
* Impact: Impact of the requirement
* Notes: Any extra comments regarding the requirement

1.4 Identifiers

The following assessments are to review security requirements for the web service. Each of the security requirements in question will be assessed and implemented on a case-by-case basis. Each security recommendation is based on a 5-point Risk Rating scale.

1.3 Assessment Overview

The purpose of the AATT security requirements is to check and identify any security issues within the software of the web service.

1.2 Purpose

AATT is an E-commerce company with user ease of access in mind and to provide users with

* User-friendly shopping experience
* Order updates
* Simple ordering process
* Tracking updates
* Many other shopping services

AATT will provide a user interface to customers in order to better relations with customers and increase effectiveness of shopping services.

1.1 Background

1.5 Risk Rating Scale

|  |  |  |
| --- | --- | --- |
| Critical Risk  5 | High Risk  4 | Medium Risk  3 |
| Minimal Risk  2 | Informational  1 |  |

Risk Scale Definitions:

1. Informational: Not a major concern for risk, used as information
2. Minimal Risk: Very small security risk, may require attention
3. Medium Risk: Poses some threat, will require some attention to be fixed
4. High Risk: Poses major threat, requires immediate attention to be fixed
5. Critical Risk: Poses the most threat to software, requires immediate attention from developers to be fixed

2. REQUIREMENTS

2.1 SECURITY REQUIREMENTS

* CWPEC 1 - Ecommerce Customer Portal shall comply with OWASP requirements to keep data separate from commands and queries, to prevent injection attacks. A03:2021
* CWPEC 2 - Ecommerce Customer Portal shall comply with OWASP requirements to manage or minimize cross-site scripting and forgery. A8:2017

* CWPEC 3 - Ecommerce Customer Portal shall comply with minimum password requirements

* CWPEC 4 - Ecommerce Customer Portal shall comply with OWASP requirements to manage or minimize the use of Vulnerable and Outdated Components. A06:2021
* CWPEC 5 - Ecommerce Customer Portal shall comply with OWASP requirements to manage or minimize exposure of sensitive data. A02:2021

* CWPEC 6 - Ecommerce Customer Portal Application shall be patched and updated regularly and checked for security vulnerabilities and flaws. A06:2021

* CWPEC 7 - Ecommerce Customer Portal Application shall deserialize information securely

* CWPEC 8 - Ecommerce Customer Portal Application shall require logging, detection, monitoring techniques. A09:2021
* CWPEC 9 - Ecommerce Customer Portal shall comply with OWASP Broken Access Control requirements. A5:2017
* CWPEC 10 - Ecommerce Customer Portal shall comply with OWASP requirements to manage or minimize server-side request forgery. A10:2021

2.2 SECURITY REQUIREMENTS MATRIX

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Title | Security Concern | Threat | Vulnerability | Risk  Rating | Mitigation | Impact | Notes |
| CWPEC 1 | E-Commerce Customer Portal Prevention of SQL Injection Attacks | Unauthorized access of data | Loss and or modification of data | Unchecked user access to SQL commands | 5 | Implement input sanitization and validation | Loss and or modification of data | SQL injections can lead to the loss of sensitive and critical data. |
| CWPEC 2 | E-Commerce Customer Portal Prevention of XSS and CSRF Attacks | Execution of unauthorized code | Malicious code injection | Allowing code input | 4 | Implement input sanitization and validation | Loss and manipulation of client data, Outcomes unknown | Only authorized, sanitized, and validated code should be executed |
| CWPEC 3 | E-Commerce Customer Portal Prevention of Password Vulnerabilities | Ease of unauthorized access to user accounts | Unauthorized usage of user accounts | Failure to uphold minimum password requirements | 4 | Achieve compliance with OWASP’s Password Security Requirements | Unauthorized usage of user accounts | Strong passwords help defend against unauthorized usage of user accounts. |
| CWPEC 4 | E-Commerce Customer Portal Prevention of Vulnerable and Outdated Components | Exploitation of components with known vulnerabilities | Malicious exploitation of vulnerabilities within components | Usage of components with vulnerabilities | 4 | Patching, swapping, updating, or removal of vulnerable or out of date components | Malicious attacks dependent upon the exploited component | Components with known vulnerabilities should be avoided or patched |
| CWPEC 5 | E-commerce Customer Portal Prevention of Sensitive Data Exposure | Transmission or storage of unencrypted or improperly encrypted sensitive data | Unauthorized access of the contents of sensitive data | Failure to meet cryptographic standards,  Lack of or broken access controls | 5 | Achieve compliance with OWASP’s stored cryptography requirements,  Rectify access controls per the access control policy | Data leaks | Proper utilization of encryption keeps sensitive and critical data secure |
| CWPEC 6 | E-commerce Customer Portal  Proper Maintenance and Upkeep | Exploitation of newly discovered vulnerabilities or outdated components | Malicious exploitation of vulnerabilities within components | Once secure components either becoming insecure or being revealed to have never been secure | 3 | Periodic usage of code reviews and code audits,  Patching and updating of outdated or insecure components | Malicious attacks dependent upon the exploited vulnerability | Regular maintenance and review of released code is essential to making sure that it stays secure. |
| CWPEC 7 | E-commerce Customer Portal Prevention of Vulnerable Deserialization | Malicious code repurposing from deserialized objects | Malicious code execution | Deserializing objects supplied from unauthorized sources,  Lack of integrity checks. | 2 | Achieve compliance with OWASP’s requirements for deserialization prevention,  Use of language specific guidelines | Lack of availability, Remote code execution,  data loss,  data leaks | Proper deserialization procedures help ensure that malicious code is not executed. Especially when received from untrusted sources |
| CWPEC 8 | E-commerce Customer  Portal  Prevention of Improper Logging, Detection, and Monitoring | Inability to detect, monitor, or identify critical events | Inability to detect, monitor or identify exploitations of vulnerable software or malicious user behavior | Lack of or improper usage of error handling and logging verification | 3 | Achieve compliance with OWASP’s requirements for error handling and logging verification | Inability to detect, identify, and mitigate vulnerabilities and exploits | Logging helps to efficiently detect and identify problems so that the proper mitigations may be implemented |
| CWPEC  9 | E-commerce Customer  Portal Prevention of  Broken Access Control | Users having unauthorized permissions | Unauthorized access and usage of data and server resources | Lack of or improper implementation of access controls | 5 | Implement proper access controls per the access control policy | Loss and or manipulation of data,  Unauthorized usage of system resources | Access control policies ensure that users only access intended data and resources |
| CWPEC 10 | E-commerce Customer  Portal Prevention of  SSRF | Execution of unauthorized requests | Fetching remote resources | Allowing user supplied URLs | 5 | Implement input sanitization and validation | Loss and manipulation of server-side data,  Outcomes unknown | SSRF can quickly spiral into RCE and DOS attacks so it’s essential to stop early |

3. ASSESSMENTS

Costs

A proactive stance in terms of cost stands to be the most beneficial in the long run. With the average cost of a cybersecurity breach reaching a new high of $8.19 million dollars per breach according to CSOonline.com, a budget for developer’s salaries of $100,000-$125,000 per year will provide adequate room to hire experienced and high value developers and reduce the risk of insecure code.

By hiring from a pool of experienced developers, cost can be mitigated in training, and further mitigated by providing open-source tools where appropriate.

Threats

E-commerce is a rapidly growing and massively lucrative enterprise, and as such has attracted bad actors in many forms. To defend against these threats, it is best to understand their effects and urgency from historical data as well as current data.

Threatening entities include:

* Hackers
* Bots
* Scammers
* Governments
* Phishers

Vulnerabilities

Defense against bad actors who would exploit vulnerabilities starts with strong code. By ensuring that the development process is focused on security, these vulnerabilities can be avoided and thus avoid exploits.

Exploits

The most common targets of exploits in ecommerce are financial gain and personal information. Bad actors can exploit insecure code and gain these by a growing variety of means:

* Brute Force Attacks
* Account Take Over (ATO)
* Cross-site Scripting
* SQL injection
* Cookie Poisoning
* File-path traversal
* Denial of Service (DoS)

Mitigation

Proactive mitigation will be the best stance to maintain. Strategies to ensure that we stay ahead of potential insecurities include a bug bounty program and strong QA during development, as well as:

* Examining for known defects
* Implementing scanning tools early & often
* Follow strict development and maintenance guidelines
* Utilization of OWASP security knowledge framework
* Secure and timely patching

Design

Keeping the web service design simple and easy to understand in order to avoid any attacks. By adding open-source components will help lessen the cost of production and provide additional security resources. The most concerning design factor will be the end-user interface. Creating a web service that users will be able to navigate easily. Online shoppers will be the primary target of this web service so the customer experience will be a high priority. Including an authentication system for the user to comply will help secure the accounts of the customers using the web service.

DESIGN SUMMARY:

1. Simple overlay
2. Open-Source Resources
3. Friendly user-interface
4. User Authentication

Implementation

By using a secure Web Application Development Process this will help strengthen the design of the web service. Each part of the development process shall be checked during development. After all of the steps of the WADP have been complete the web service can be safely deployed and be used by end-users.

See Appendix A.

Verification

To further ensure security and program quality are met and free of known security issues, a thorough verification process will also be implemented to ensure no unforeseen insecurities arise and that the program functions adequately and will include:

* Code audits
* Code review
* External vulnerability audits
* Testing and QA

4. Conclusions

Following the security requirements mentioned in this document will help create an excellent web service. By implementing the discussed design and development process into the web service will ensure a safe and secure system for the end users. Including the WADP as an outline, AATT can guarantee a simple strategy to launch our service.

APPENDIX A

**Web Application Development Process**

