Activity 8: Part 4

Ryan Coon

CST-407

Professor Ty Howard

February 11, 2024

**Best Practices of Secure Programming**

How would you minimize systems risks, threats, and vulnerabilities using best practices of secure programming?

Minimizing system risks, threats, and vulnerabilities through best practices of secure programming involves implementing a comprehensive approach to software development.  Implement strong authentication mechanisms, such as multifactor authentication, and enforce the principle of least privilege to ensure that users and processes have access only to the resources they need. On top of that you should add encryption to sensitive information within the database.

Validate and sanitize all inputs to prevent injection attacks, buffer overflows, and other forms of malicious input manipulation. This includes validating user inputs, API requests, file uploads, and any data received from external sources. Adhere to secure coding standards and best practices, such as those outlined in industry-recognized resources. This includes using safe APIs, avoiding insecure functions, and following secure coding patterns.

 Use secure data storage and communication, such as HTTPS for web applications, and implement encryption for data in transit and at rest. Additionally, consider implementing secure communication channels for inter-process communication and API integrations. Integrate security testing throughout the software development lifecycle, including static analysis, dynamic analysis, fuzz testing, and penetration testing. This helps identify and remediate vulnerabilities early in the development process. Implement comprehensive logging and monitoring to detect and respond to security incidents and abnormal behavior. Log and monitor security-relevant events and activities within the application and infrastructure.

Implement robust error and exception handling to prevent information disclosure and to maintain the security and stability of the application in the event of unexpected conditions. Regularly update and patch third-party libraries and components to address known vulnerabilities and utilize dependency checkers to identify and mitigate security issues in external dependencies.

By following some of the best practices of secure programming, organizations and developers can minimize system risks, threats, and vulnerabilities. Overall this would enhance the overall security standing of their software and their systems.

References:

Pfleeger, C. P., Pfleeger, S. L., & Margulies, J. (2018). Security in Computing: 5th Edition. <https://dl.acm.org/citation.cfm?id=2756777>

Secure Coding Practice Guidelines | Information Security Office. (n.d.). Security.berkeley.edu. https://security.berkeley.edu/secure-coding-practice-guidelines

Secure Coding Practices | What is secure coding? | Snyk. (n.d.). Snyk.io. https://snyk.io/learn/secure-coding-practices/