Richard Schall
CS-405-H4327: Secure Coding 21EW4
Southern New Hampshire University
April 21, 2021

8-2 Journal: Portfolio Reflection

Adoption of a Secure Coding Standard, and Not Leaving Security to the End

Software vulnerabilities reveal weaknesses in code and are subject to exploitation by criminals. Preventing these incidents starts at the beginning of the process of writing code. When a secure coding standard is present and followed, the software developers of a company are working to minimize vulnerabilities in code. (Morrow, 2021)

The Open Web Application Security Project (OWASP) is a good place to look for guidelines on making code are secure as possible. Some of the topics covered are data validation, authentication and password management, cryptographic practices, error handling and logging, data protection and communication security. (Morrow, 2021)

Security does not apply to just code. There needs to be a focus in other areas as well like developing a computer system of least privilege, practicing defense in depth, and practicing good quality assurance. (Morrow, 2021)

Evaluation and Assessment of Risk and Cost Benefit of Mitigation

Threat modeling is an activity that will need to take place when evaluating risk and the cost benefit of mitigation. "Threat Modeling is the practice of examining a software design to find places where attackers could compromise it." (Johnson, 2020) The threat model helps developers implement defense into their code. (Johnson, 2020)

With the threat model risks are ranked. Ranking helps the development team decide which risks need to be addresses and when.

Zero Trust

Zero trust replaces the classis “castle and mote” model where a company’s network sits behind a secure perimeter. (Kueh, 2020) In the classic model, if a user needs remote access, they must use a VPN to gain access to resources inside the security perimeter. Zero trust moves security to each device, user, and application. (Kueh, 2020) Zero trust uses continuous verification of trust across each device, user, and application. “It does this by pivoting from a “trust but verify” to “never trust/,always verify” approach.” (Kueh, 2020) There are five pillars of zero trust, those are Device Trust, User Trust, Transport/Session Trust, Application Trust, and Data Trust.

Implementation and Recommendations for Security Policies

Developing a security policy is a worthwhile endeavor for companies to address security. Companies should use the CERT website (https://wiki.sei.cmu.edu/confluence/) to guide their security policies. A security policy standardizes a company’s handling of vulnerabilities in code. This ensures every developer is coding in the same way to prevent attacks. The best to make sure the security policy is known and adhered to is training and making the security a part of the software development life cycle through code reviews.

References

Johnson, P. (2020, February 16). Secure coding: A practical guide. Retrieved April 22, 2021, from https://resources.whitesourcesoftware.com/blog-whitesource/secure-coding

Kueh, T. (2020, January 25). A practical guide to zero-trust security. Retrieved April 21, 2021, from https://threatpost.com/practical-guide-zero-trust-security/151912/

Morrow, S. (2021, January 1). What is secure coding and why is it important? Retrieved April 21, 2021, from https://vpnoverview.com/internet-safety/business/what-is-secure-coding/#:~:text=Secure%20code%20will%20help%20to,Software%20vulnerabilities%20are%20rampant.