Brennon Fultz

CS 405

Module Eight Journal

The adoption of a secure coding standard is one of the most effective ways an organization can improve software security and produce more secure code. By implementing a secure coding standard, developers can gain a clear understanding of the expectations and best practices they must employ. This can help to prevent vulnerabilities like buffer overflows and undefined behavior that unsecure code may produce, vulnerabilities which could lead to a security breach (Seacord, 2005). For example, passing an untrusted string from input to a function can allow arbitrary code execution, as outlined in Coding Standard 4 from the Project One security policy. Adopting a secure coding policy is also part of the best practice of not leaving security to the end. This best practice means that security should be considered at every step of the development process. By designing with security in mind from the beginning, it is much easier to implement security along the way that attempting to simply tack on security features last. One possible example of this would be in deciding which data should be collected and only collecting what is necessary, therefore making the software more secure by having less data to protect.

Evaluating and assessing the risks and cost of mitigation is an important part of defining a security policy. While all security risks are worth addressing, it is also critical to prioritize the potential vulnerabilities so that the most threatening are fixed first. Some vulnerabilities may only pose a minimal risk and these should be treated with less concern than more potentially-costly issues. Analysis of the cost benefit of mitigation can allow an organization to better understand how best to allocate resources when addressing these threats. This approach combined with effective evaluation of risks can allow a company to prevent wasting resources while fixing the most important security risks.

Zero trust security involves treating all entities in the software system as untrustworthy until they can be verified, for each interaction or point of access. By removing trust and requiring verification of identity and context such as device or session trust, this approach prevents a single point of failure from causing a total breach of the system (Kueh, 2020). This is different from traditional approaches to security which rely on initial authentication but then allow entities to operate freely within the system. For example, a user may be required to authenticate to access a private network and then is treated as a trusted entity afterward. This tactic has become less reliable as systems have become more complex and distributed such as in cloud-based architectures, which is why zero trust is the preferred solution (Kueh, 2020). Zero trust and other security policies are extremely important to an organization but may be difficult to implement. Security can often make work more difficult and slow, such as when multi-factor authentication is needed or when data must be encrypted and decrypted. For this reason, it is important to remember the potential cost of security failures, which can be devastating to a company. One example is National Public Data, a company which is now bankrupt due to a security breach in 2023 that leaked information on millions of Americans, including social security numbers (Gregory, 2024). Due to the potential costs of failure, security policies and best practices should always be carefully constructed, implemented, and followed as recommended.

References

Seacord, R. 2005. *Top 10 Secure Coding Principles.* CMU Software Engineering Institute. Web. https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices?f ocusedCommentId=88044413

Kueh, T. (2020). *A practical guide to zero-trust security*. Threatpost. Web. https://threatpost.com/practical-guide-zero-trust-security/151912/

Gregory, J. (2024). *National public data breach publishes private data of 2.9b US citizens.* SecurityIntelligence. https://securityintelligence.com/news/national-public-data-breach- publishes-private-data-billions-us-citizens/