Joel Hays

CS-405

Module 8 Journal: Portfolio reflection

Adopting a secure coding standard is critical for maintaining consistent practices and reducing vulnerabilities across development teams. Standards like CERT C and OWASP provide clear guidelines for preventing common security issues, such as buffer overflows or injection attacks, which can lead to significant breaches if left unaddressed. Additionally, the principle of "not leaving security to the end" ensures security is incorporated into every phase of the software development lifecycle (SDLC). By addressing vulnerabilities early, teams can avoid costly, time-consuming fixes later. For example, using input validation during development helps to prevent SQL injection attacks before they reach production, saving both time and resources.

Evaluating and assessing risk is a major part of effective security practices. This involves identifying potential threats, determining their likelihood, and evaluating the potential damage they could cause. Cost-benefit analysis helps balance the expense of security measures with the value of what they protect. For instance, implementing multi-factor authentication (MFA) might require upfront costs, but it drastically reduces the risk of compromised accounts, protecting sensitive data and reducing the likelihood of a costly breach. Taking a structured approach to risk evaluation ensures resources are allocated where they will have the most impact.

Zero trust is another important concept that strengthens security by requiring constant verification of users and devices before granting access. Unlike traditional approaches that assume users within a network are trustworthy, zero trust ensures access is restricted based on identity and authorization. This approach greatly reduces the risk of internal and external threats. For example, in a healthcare setting, employees accessing patient records might need to authenticate through MFA and work within segmented networks, ensuring data is accessed only by authorized users.

Implementing strong security policies provides the foundation for protecting systems and data. Policies ensure employees follow best practices and help organizations comply with industry standards. An effective example is an access control policy that enforces the principle of least privilege, limiting access to only what is necessary for a specific role. This minimizes insider threats and accidental exposure of sensitive data. To remain effective, these policies should be regularly updated to address new threats, and security awareness training should be conducted to ensure employees are aware of their responsibilities.

In summary, adopting secure coding standards, evaluating risks, applying zero trust principles, and maintaining strong security policies are essential practices for creating resilient systems. These strategies help organizations address potential vulnerabilities proactively, reducing risks and protecting valuable assets in an ever-changing threat landscape.

References:  
Kueh, T. (2020, January 15). *A Practical Guide to Zero-Trust Security*. Threatpost.com. <https://threatpost.com/practical-guide-zero-trust-security/151912/>

Seacord, R. C. (2013). *Secure Coding in C and C++* (2nd ed.). Pearson Technology Group. <https://mbsdirect.vitalsource.com/books/9780132981972>

NIST. (2002). The Economic Impacts of Inadequate Infrastructure for Software Testing [Review of *The Economic Impacts of Inadequate Infrastructure for Software Testing*]. In RTI (Ed.), *NIST.gov*. RTI. <https://www.nist.gov/system/files/documents/director/planning/report02-3.pdf>