Module 8 Journal

Sean J Born

CS 405

June 29, 2025

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

One of the most important takeaways from this course is that security must be integrated into the development lifecycle from the start, not treated as an afterthought. Adopting a secure coding standard provides developers with clear, enforceable rules to prevent common vulnerabilities like buffer overflows, integer overflows, and input validation flaws. Early security integration reduces the likelihood of costly vulnerabilities being discovered during deployment or after release. Applying consistent standards also fosters a shared security culture among development teams and ensures maintainability.

**Evaluation and Assessment of Risk and Cost Benefit of Mitigation**

Security decisions should be guided by risk assessment and cost-benefit analysis. Throughout the course, we examined frameworks like NIST SP 800-30, which help quantify the likelihood and impact of various threats. This structured approach allows organizations to prioritize their mitigation strategies effectively. For instance, it may not be cost-effective to defend against every minor threat, but mitigating a high-impact vulnerability in a business-critical system might be non-negotiable. Balancing the cost of a mitigation technique (like encrypted databases) against the potential loss of sensitive data provides a clearer picture for decision-making. A rational, metrics-based approach helps justify security spending to stakeholders and ensures resources are used where they can provide the most benefit.

**Zero Trust**

The principle of zero trust, which operates under the assumption that no user or system should be automatically trusted, represents a shift in modern cybersecurity. Zero trust dismantles the traditional perimeter-based model and instead enforces continuous authentication, authorization, and validation of all access attempts. Implementing this model means restricting access using least privilege, micro-segmentation, and identity-aware proxies. This approach not only limits the damages caused by a potential breach but also provides resilience against insider threats and credential-based attacks. Although it may introduce more complexity into the user experience, zero trust provides a much-needed balance between usability and proactive security.

**Implementation and Recommendations of Security Policies**

Effective security policies serve as the foundation for consistent and enforceable security practices across an organization. The course emphasized that policies must be clear, practical, and aligned with organizational goals. For example, a password policy that enforces complexity but doesn’t account for usability may result in workarounds that compromise security. Implementation of policies should be supported by ongoing training, regular audits, and automation where appropriate. My recommendation would be to incorporate policy enforcement into the pipeline, ensuring security rules are automatically checked during development. A living policy that is revisited often and updated based on emerging threats keeps the organization agile and protected.