**Portfolio Reflection**

Cristian Chalarca

Southern New Hampshire University

CS-405 Secure Coding

Professor: Dr. Mimi Tam

08/24/2025

**Adoption of a Secure Coding Standard**

Adopting a secure coding standard early in the software development life cycle ensures that security is not an afterthought but a built-in requirement. The SEI CERT C++ Coding Standard highlights the importance of consistent practices, such as proper input validation, memory management, and exception handling, to prevent vulnerabilities before they surface (Ballman, n.d.). When security is left to the end, vulnerabilities may be deeply embedded in design or logic, requiring expensive rework and creating risks if overlooked. By establishing a secure coding standard from the start, teams create a shared language of safety, making code reviews, audits, and long-term maintenance more effective while reducing overall risk.

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

A key principle from cybersecurity frameworks such as NIST is that not all risks can or should be mitigated equally; instead, organizations must evaluate both the likelihood and impact of threats. As emphasized in NIST Special Publication 800-30 on risk assessment, cost-benefit analysis is essential for understanding when the expense of mitigation outweighs the potential damage of exploitation (NIST, 2012). For example, encrypting sensitive customer data is a high-value mitigation with clear benefits, while implementing overly complex protections for low-risk processes may waste resources. By quantifying both risk and mitigation costs, developers and managers can make informed decisions that balance security, performance, and budget.

**Zero Trust**

The Zero Trust model shifts the mindset from perimeter-based defense to continuous verification. Instead of assuming users and systems inside a network are safe, Zero Trust requires constant authentication, authorization, and validation at every step. NIST SP 800-207 describes this approach as essential in today’s distributed and cloud-based environments, where implicit trust can easily be exploited (Rose et al., 2020). For developers, this translates into writing code that enforces least-privilege access, validates inputs, and strengthens identity checks. Zero Trust directly aligns with modern security realities, ensuring that every request is scrutinized and attackers face barriers at every layer.

**Implementation and Recommendations of Security Policies**

Effective security policies translate high-level principles into actionable practices and organizational procedures. These policies should cover areas such as access control, secure data handling, audit logging, and incident response, while also providing compliance guidelines aligned with secure coding standards (Cichonski et al., 2012). Implementation requires both technical measures, such as enforcing multi-factor authentication, and cultural ones, like developer training and ongoing awareness programs. Recommendations include periodic reviews of policies against evolving threats, automated compliance checks integrated into CI/CD pipelines, and alignment with frameworks such as ISO/IEC 27001 and NIST. As emphasized in course readings, security policies should be treated as living documents that evolve with technology and threat landscapes.

REFERENCES:

* Ballman, A. (n.d.). SEI CERT C++ Coding Standard Rules for Developing Safe, Reliable, and Secure Systems in C++ 2016 Edition. <https://resources.sei.cmu.edu/downloads/secure-coding/assets/sei-cert-cpp-coding-standard-2016-v01.pdf>
* Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer security incident handling guide. Computer Security Incident Handling Guide, 2(2). <https://doi.org/10.6028/nist.sp.800-61r2>
* NIST. (2012). Guide for Conducting Risk Assessments NIST Special Publication 800-30 Revision 1 JOINT TASK FORCE TRANSFORMATION INITIATIVE. In NIST. NIST. <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-30r1.pdf>
* Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture. NIST Special Publication 800-207, 1(800-207). <https://doi.org/10.6028/nist.sp.800-207>