Secure Coding Journal

Gonzalo Patino

Southern New Hampshire University

CS-405 Secure Coding

October 23th, 2025

# Adoption of a Secure Coding Standard

Seacord explains that security should start at the beginning of a software project, not at the end. A secure coding standard, like the CERT C Secure Coding Standard, gives clear rules that help developers avoid known bugs and vulnerabilities. When developers follow these rules early, they prevent mistakes and save money later. Fixing security problems after release is much more expensive and risky than writing secure code from the start (Seacord, 2013).

# Evaluation and Assessment of Risk and Cost-Benefit of Mitigation

Seacord also says that every security decision should balance risk and cost. The best fix is to remove a defect completely, but sometimes it is cheaper and faster to reduce risk in another way such as filtering bad input or limiting access. Each fix has a cost, so developers must decide which actions give the best protection for the effort required. This type of risk evaluation helps focus on the most serious problems first (Seacord, 2013).

# Zero Trust

Even though Seacord wrote before the term “Zero Trust” became popular, his ideas fit the same mindset. He teaches that no system or input should be trusted by default. Each part of a program must check and validate what it receives. This matches the Zero Trust idea: always verify, limit permissions, and assume that any part of the system could be attacked.

# Implementation and Recommendations of Security Policies

Security policies turn general rules into daily practice. Seacord says developers should validate all input, limit privileges, and test software often. These policies should be part of code reviews and training so everyone on the team follows them. He also recommends using static and dynamic analysis tools to find errors early and to make sure policies are being followed.

# Reflection

This course helped me understand that secure coding is not something added at the end—it must be part of every step. Using a standard like CERT C helps create safer code from the start. Evaluating risks helps focus time and money where it matters most. The Zero Trust mindset and strong security policies build layers of protection that reduce both human error and outside attacks. Secure coding is really about forming disciplined habits that last through every stage of development.

# References

Seacord, R. C. (2013). *Secure coding in C and C++* (2nd ed.). Addison-Wesley.