# CS 405 Secure Coding

Shayne Rushton

Portfolio Reflection

08/15//2023

**Reflection**

Security is a huge topic now in the world of cloud computing, data storage, transfer and usage. Given the breaches researched in this course, and the amount of damage caused, it is clear that no company can afford to “wing it” and hope for the best. Without understanding the common vulnerabilities used by hackers today, and understanding the tenacity with which they keep trying to break into systems, it’s not enough to write secure code. While following a policy for secure coding is necessary, companies can’t stop there.

Not only do developers need to move security into the development phase of the software development life cycle, they need to also constantly monitor software after deployment. Waiting until the code is written, a traditional step, can lead to costly fixes or even data breaches from overlooking security to save money upfront in fixing the issues in production. Adopting the mentality of Zero Trust is the way into the future.

With so many access points to systems in this current economy, the need to secure data and systems at every point along the way, while also continuously checking for authentication at each point, Zero Trust will stop bad actors at various points in the process, making it much more difficult to break into systems and steal data. The risks are too great to ignore this methodology and the benefits far outweigh it

Risks associated with bad security practices include customer data being sold on the internet, companies settling lawsuits, poor reputation and profit loss. Any of these can cause a company to close. If your company stores information that hackers want, failing to secure the data is asking for trouble. Customers do not want to do business with a company that does not take their security seriously, which leads to lawsuits and loss of profits from a sullied reputation. The only answer is implementing a security policy that is followed from step on in the process, and never stops throughout the lifecycle of a system.

In conclusion, security isn’t an optional step any more, and it can’t be held off until the end of the development phase. Zero Trust will keep all users and devices from accessing data they should not have access to, thereby preventing hackers from obtaining that information, and a company that implements a strong security policy and enforces it will maintain a solid reputation for customer protections and will therefore thrive.

**References**

Seacord, R. C. (2013). Secure Coding in C and C++ (2nd ed.). Pearson Technology Group.

<https://mbsdirect.vitalsource.com/books/9780132981972>

Haldar, D. (2016). Top 15 C++ Exception handling mistakes and how to avoid them. A Coder’s Journey.

<https://www.acodersjourney.com/top-15-c-exception-handling-mistakes-avoid/>

Vineet, C. (2023). Simple and Static Assertion (assert) in C Programming Language. Developer Insider.

<https://developerinsider.co/simple-and-static-assertion-assert-in-c-programming-language/>

Google. (n.d.). DevOps tech: Shifting left on security.

<https://cloud.google.com/architecture/devops/devops-tech-shifting-left-on-security>

Programming Algorithms. (2023). XOR Encryption.

<https://www.programmingalgorithms.com/algorithm/xor-encryption/cpp/>

Beddoe, W. (2015). Understanding the Hierarchy of Principles, Policies, Standards, Procedures, and Guidelines.

<https://www.linkedin.com/pulse/understanding-hierarchy-principles-policies-standards-wally-beddoe/>

Carnegie Mellon University. (2020). SEI CERT C++ Coding Standard. Confluence.

https://wiki.sei.cmu.edu/confluence/