* **Adoption of a secure coding standard, and not leaving security to the end**

While I am still often unsure if this course is referencing a “standard” or a “rule” when it states to adopt a “standard” as the two were often referred to interchangeably, I am going to assume the broader context of “standard” applies here, such as adopting STD-001-CPP. Using a coding standard such as this broadly, yet specifically, defines the principles that dictate one’s security policy for developers. As noted in the FTC’s article *Start with Security: A Guide for Business* security should be factored into decision making for every department of a business (Federal Trade Commission, 2023).

* **Evaluation and assessment of risk and cost benefit of mitigation**

Anytime one is going to apply security measures, it is important to ask oneself if the measures for security, and the cost to implement them make sense given the entity/entities one wishes to protect. For instance, a wedding ring with a large diamond costing $10,000 may be expensive, but it would be absolutely ludicrous to spend $20,000 on a vault to store the ring in. Not only does the defense cost more than the object being defended, but you have added a complex, tedious task in between the wearer of the jewelry and the jewelry. The same principle should be applied in software development. A user, for example, should not have to expend insane amounts of effort memorizing and changing a 16-digit password with extreme requirements 12 times a year in order to access a database where only low-risk information is stored/processed. Alternatively, nor should the social security numbers of 4 billion people be stored on a Microsoft Excel spreadsheet in plain text. These are both examples of poor risk assessment, with unequal benefit to mitigation ratios. A useful stance is to put oneself in the “shoes” of a would-be malicious actor, so-to-speak, and assess to what lengths, and through what avenues one might go to attempt intrusion into a system. Anytime it makes sense to add security, given those conditions, add it. If it can be added at a low cost with no user interference, then add it as well.

* **Zero trust**

Zero trust is the principle and practice of exactly that- trust nothing and no one until verified. Anytime that anything needs access to any entity within a system, a verification occurs, and access is denied, by default, unless that entity is authorized access to that compartment of the system. This highly compartmentalized, rejection-by-default type of system is robust in comparison to perimeter-based security protocols, which require only one entry point to access anything inside.

**Implementation and recommendations of security policies**

As stated in the second paragraph of this work, implementation and recommendations of security policies should be dictated by necessity. A security policy that costs more than the damage that would come from intrusion into a system does not make sense. A security policy that mitigates damages and costs by being smartly applied such that maximum protection is afforded at the lowest cost and complexity level is ideal.

**References**

Federal Trade Commission. (2023, August 16). *Start with security: A guide for business*. <https://www.ftc.gov/business-guidance/resources/start-security-guide-business>