Jason Yaeger

Journal

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

Security must first be something developers and organizations must be aware of. With awareness, design and architecture can be planned and built with security in mind. Secure coding practices and proper testing throughout development will be much more effective than leaving it all until the end. It is near impossible to implement good coverage and proper defense in depth unless it is thoroughly planned.

* Evaluation and assessment of risk and cost benefit of mitigation

Having a good understanding of not only the systems itself but also the business and organization as a whole can help in the evaluation of risks. Understanding what an attacker might want to compromise, what data is valuable, what dangers a breach could pose, etc. There can be tremendous loss of money, trust, safety, and more if attacks occur. Implementing good security to mitigate might cost more upfront, but over the long run will save time, money, and headaches from dealing with the consequences of security issues.

* Zero trust

This week's discussion went over zero trust, and from this it was very clear that implementing this improves security greatly. The principle of least privilege and micro-segmentation are key concepts that enable zero trust to function.

* Implementation and recommendations of security policies

My recommendations for security policies would start out as having an organizational awareness and understanding of security. This means having chief security officers, and dedicated personnel that are experts in security. Designing and architecting systems with defense in depth and coverage as a priority. Implementing good secure coding procedures and testing practices. A rigorous approach is the best approach.