Portfolio Reflection

Joe Andersen

Southern New Hampshire University

Adopting a secure coding standard, and not leaving security to the end are two things every Software Engineer should practice. Starting with security is significantly easier than saving security for the end. There are a few best practices that we should follow in regards to data security. Accordign to the Federal Trade Commission “You should protect the information that you keep, and properly dispose of what you no longer need. And, of course, you should create a plan to respond to security incidents.” (Federal Trade Commission., 2020, February 5) It’s important to only store information that is needed for your application to function. Holding onto a user’s personal information that is not needed for your application to function is simply asking for trouble.

Evaluation and assessment of risk can be done internally if your company has a cybersecurity business unit. You also have the option of going with a third-party company that provides security risk assessments. Paying to perform such evaluations can be invaluable in keeping your networks and applications safe against a cyberattack. The cost benefit of mitigation is incredibly difficult to measure. If no malicious actors are penetrating your systems, then technically the mitigation efforts are doing their jobs. This is where companies can easily become complacent and feel that their current efforts are enough for now and the future. This way of thinking is typically what gets companies in trouble and will ultimately leave them exposed to an attack.

During a risk assessment it might come to light that security practices should be updated. What worked five years ago may not work as well today. This is a great segue into a more modern approach to security know as zero trust. In a traditional security model, you would typically classify users, devices, and applications all within the corporate network as trustworthy. (Kueh, I. S. I. T., & Kueh, T., n.d.) This approach typically relies on building security around the perimeter of the companies’ network. The major problem with this approach is once an attacker gets through the perimeter defenses there is little to nothing stopping them from accessing much of the data that perimeter was setup to protect.

With the zero-trust security model you continuously verify “every device, user and application”. (Kueh, I. S. I. T., & Kueh, T., n.d.) It’s built on the idea of five different trust pillars. Device trust, user trust, transport/session trust, application trust, and data trust. This model is able to do continuous verification “by pivoting from a “trust but verify” to “never trust/,always verify” approach. In practice, this model considers all resources to be external and continuously verifies trust before granting only the required access.” (Kueh, I. S. I. T., & Kueh, T., n.d.) This makes for a more modern and secure approach to security.

The implementation and recommendations of security polices will vary from project to project. I would recommend picking a few that makes sense in the larger picture of the security model and make those mandatory. Things like no storing credentials in plain text or the use of a common verification process that all must abide by to name a few. Such higher level polices should just be the start though. I would recommend having each team define their own standards that they must follow in order to make sure they are practicing secure coding practices. I would have the cybersecurity team review the standards the teams come up with to help guide any teams that may be as experienced in defining standards.

References

*Start with Security: A Guide for Business*. Federal Trade Commission. (2020, February 5). https://www.ftc.gov/tips-advice/business-center/guidance/start-security-guide-business.

Kueh, I. S. I. T., & Kueh, T. (n.d.). *A Practical Guide to Zero-Trust Security*. Threatpost English Global threatpostcom. https://threatpost.com/practical-guide-zero-trust-security/151912/.