Leaving security to the end is a dangerous practice. This involves not thinking of security throughout the development process, and only at the “end” where a breach and impact have already occurred. This is a regressive process that only thinks about security retroactively. Instead, we need to take a proactive approach, where we start with security at the beginning and implement it throughout the design and development processes, up through the loading of production and afterwards when retrospective analyses occur after deployment. It is important to factory security into the “decision making in every department of [any] business” (FTC, 2015). It is also important to think through every single data decision. Data must consistently be protected.

Access to data must be controlled (FTC, 2015). This is bolstered by the Triple A framework: authentication, authorization, and auditing. Users must have the necessary roles to complete their daily functions. For example, if a vendor for a company does not need to access sensitive user data as part of their daily operations, then they do not need to be given access to it; effectively, they should not be authorized (FTC, 2015). In a similar vein, only users who are meant to be administrators should be given administrator roles, which have their own separate job duties and accesses to a system.

These fall inline to secure coding in implementing best practices across password usage and authentication. Input validations can be put in place to ensure that complex and unique passwords are created and stored by users (FTC, 2015). Strong password systems protect user data from being accessed and leaked. On top of that, there needs to be code which stores the passwords securely, with techniques like encryption.

Additionally, within the code, there needs to be protective measures against direct attacks such as brute force attacks. This can be done with input validation, input sanitization, and programmatic methods to defend against SQL injection and cross-site scripting. Security can also be implemented from the inside out, to protect against memory leaks and data leaks by ensuring proper memory management. Sometimes, it is our own rocky internal structures that cause vulnerabilities and crashes, rather than external actors enlisting attacks on the system.

*Start with Security: A Guide for Business*. (2015, June 29). Federal Trade Commission. https://www.ftc.gov/business-guidance/resources/start-security-guide-business

‌