Alex Surprenant

CS-405

7/17/25

Journal: Defense in Depth

Defense in Depth is an excellent solution to preventing hackers from gaining access to systems by implementing multiple levels of security at each layer of the system. There are definitely some things to consider, however, about when the strategy is being implemented too deeply, as this can also come with some drawbacks. If too many layers of security are used at each layer, the complexity of the system may become overwhelming for developers to implement and maintain. Just as well, the more complexity that is added to a system, the greater time it will take to develop, and the more costly it will become.

There are some tradeoffs to consider as well when implementing the Defense in Depth strategy. With more complexity comes longer development lifecycles, which means more cost upfront when developing the system and more tooling and potential staff needs to consider as well. This will also mean more monitoring and maintenance of the system at the other end as well, which will again drive up cost and time. On the flip side though, a security breach to a system, especially if it negatively affects the customer, will take a huge toll on the companies credibility and reputation. Depending on the use case of the system itself, the cost of going deep into security defense might be greatly overshadowed by the negative potential outcomes of not doing so.

Every system is different, and they will all need to consider their own needs when it comes to utilizing a defense in depth strategy. A large bank might need to consider multi-factor authentication at multiple levels to ensure customer financial security, while a medical office will need to transfer files back and fourth using encryption, while also needing MFA as well. A small app developer of a recipe app may just need to use password authentication in their app, as they may not have the need, or even the skill to implement anything more robust.