Sean Churchill

4/21/21

8-2 Journal

When adopting secure coding standards and not leaving security till the end general makes a more secure software program. Having a secure system will lower the risk of having a data breach. When evaluating risks, the user has to prioritize them the lower the risk normally means lower the cost and sometimes do not have to take care of right away. With a high-level risk, it has the potential of costing the company a lot of money. Also, having a higher-level risk the company will want to fix the issue right away instead of waiting to fix it. Zero trust security is a good way of doing secure coding. The things that the user should have zero trust is what they call the five pillars (device, user, transport/session, application, and data). But once trust is established between them all the user will still want to re-verify the sources they are using constantly. When making these implementations want to keep in mind encryption strategies and triple-a framework. A few recommendations that I have is to use defense in depth, so the system has protection across the board and not just in certain areas. My second recommendation would be to use Static Application Security Testing and Dynamic Application Security Testing. By using both of these systems it will find errors that the human eye might miss.