Blaze Halderman

SNHU-CS405

Module Eight Journal

12/15/2022

Adoption of a coding standard is an important aspect to securing a development team and the software associated with it. Having a secure coding standard allows for a development team to comply with security protocols as well as follow best practices for security. An example of a coding standard would be a list of principles or coding standards, such as the DCL60-CPP which is for obeying the one-definition rule. These standard states that two different translations of the same variable cannot and should not be used with differing definitions. It’s also important to note that waiting to incorporate security into a software application can cost more as well as leave the application vulnerable for future users. This means implementing security as early as possible is the best approach.

Evaluation and assessment of risk involves understanding principles and standards for coding and basing the evaluation and risk off severity level for vulnerability, likelihood of vulnerability or occurrence, priority as well as level (both for assessing importance and impact). Cost benefit for mitigation includes assessing the overall risk and assessing that risk into value when mitigated. An example of these metrics can be found on the same coding standard DCL60-CPP, with severity as high, likelihood as Unlikely, Remediation cost high, priority P3, and Level L3. These values assess the overall standard and it’s value for security and compliance.

Zero trust involves assessing the user base on an application as well as the developers, and ensuring that privacy, security and compliance are at the core of development. This means there should be no trust for users, regardless of if they’ve been authenticated or not, and always assuming the potential for a breach or risk from any user regardless of security systems or practices in place. This ensures that no user gains “special” or overlooked treatment and ensures the application security is at the focal point. Putting authentication and authorization practices in place helps ensure this standard along with other best practices for ensuring safety for users. Timed access is also a great practice to incorporate since no single user can gain access for an unlimited amount of time to the system.

Finally, implementation and recommendation for security policies must keep the users, developers, as well as management and the company in mind, and keeping the focus on security. Presenting security policies includes providing all aspects to the policy, which include risk assessment, potential benefits, long term review and assessment, implementing now versus later, and why having a security policy is important. Security policies can keep software consistent and compliant with coding standards as well as security across the board. It’s important to assess risk and assess the benefits of incorporating a security policy in order to protect developers, users and data.

Citations

*DCL60-CPP. obey the one-definition rule*. DCL60-CPP. Obey the one-definition rule - SEI CERT C++ Coding Standard - Confluence. (n.d.). Retrieved December 17, 2022, from https://wiki.sei.cmu.edu/confluence/display/cplusplus/DCL60-CPP.+Obey+the+one-definition+rule

*What is Zero trust security? principles of the zero trust model*. crowdstrike.com. (2022, November 10). Retrieved December 17, 2022, from https://www.crowdstrike.com/cybersecurity-101/zero-trust-security/