Fernando Lomeli

CS 405

Prof. Kevin Nguyen

Portfolio Reflection

Throughout this course we have reviewed several topics to reinforce our idea of security. We have reviewed topics of secure coding standards, not leaving security to the end, zero trust, and have created a security policy.

The goal of adopting security policies is to guide members of an organization on how to implement security principles, secure coding standards, and secure coding best practices to protect the organization and its data. The principles outline the organization’s values and objectives and are aligned with many of the secure coding standards. The secure coding standards adopted are followed to mitigate vulnerabilities and to develop secure and reliable code. The coding standards are evaluated by the severity level, likelihood, remediation cost, and priority level. Automations tools are also used to test for any coding standard errors that can lead to exploitable vulnerabilities and undefined program behaviors. Following coding standards is much more cost effective than having to deal with any issues/attacks that arise due to poor implementation.

Some recommendations of security policies include automation, triple A, and encryption. Automation is used to enforce coding standards and principles. Static application security testing tools like Cppcheck and dynamic application testing tools can be implemented into CI/CD pipelines to respectively scan for vulnerabilities or run simulated attacks to test for any exploitable vulnerabilities. Triple A is recommended to enhance security by first authenticating who someone is with tactics like multi-factor authentication. Authorization determines what services a user can access which adheres to the principle of least privilege. Accounting is essential, especially when issues arise, to track what resources were accessed and by who. Encryption is a policy needed to encrypt data in any state so that if attackers obtain it, they will still not be able to access it without a decryption key.

Zero trust security is also essential to software security as it takes a never trust anyone and always verify approach. Zero trust relates to triple A as authentication and authorization occurs before users access any resources. Zero trust security focuses on device, user, session, application, and data trust. For example, certain devices are listed and trusted by the company while users are authenticated in efficient ways like multi-factor authentication and the use of conditional-access policies. Overall, the zero-trust approach will reduce the risk of a data breach and reduce the attack surface.

Don’t leave security to the end is a best practice in secure coding in which security should be prioritized in the beginning stages of the software development lifecycle and should not be left until the end. Addressing security issues early on can be much more cost effective than having to deal with problems that arise in the end which are more costly or more complicated. There are many examples of how to implement this approach such as adopting security policies and threat modeling in the planning phase. Automation of continuous security testing would also be beneficial. Overall, not leaving security until the end is a good way to mitigate vulnerabilities and prevent threats.

References

Kueh, T. (2020, January 15). A Practical Guide to Zero-Trust Security. Threatpost. https://threatpost.com/practical-guide-zero-trust-security/151912/

SEI CERT C++ Coding Standard. SEI CERT C++ Coding Standard - Confluence. (n.d.). https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682