Portfolio Reflection

Kenneth J Gollaher

CS-405 Secure Coding, Southern New Hampshire

Module 8-2 Journal

Professor Ivan Gappy

Oct 23rd, 2022

**Adoption of a secure coding standard, and not leaving security to the end**

There are many reasons why the implementation of secure coding is an important element and why it shouldn’t be a last task in development. During development, there is always a margin for human error. Although we do our best to reduce errors or vulnerabilities, mistakes happen, which gives developers the opportunity to fix and learn from their mistakes. Sometimes these mistakes are minor, but some are catastrophic. Cyber Criminals are always looking for those mistakes and plan to exploit for immoral purposes. The loss of data can sustain financial and cause damage to an organization’s reputation (Delgado, 2019).

Errors and bugs can be missed if not tested during development. Companies can avoid a variety of security pitfalls if they adopt secure coding best practices early. One of the many vulnerabilities that we learned throughout this course is buffer overflow, where data can be overwritten. Adopting secure coding can prevent this with the use of dynamic memory allocation.

**Evaluation and assessment of risk and cost benefit of mitigation**

As time passes and technology advances, security is the upmost importance for any software application. Security is becoming more and more of a concern for unauthorized access to secure data. If teams take a shortcut and fail to properly handle any vulnerabilities, there can be significant damage to the organization and the cost may be detrimental to recover, both financially and by reputation. Developers that evaluate and assess risk throughout the entire build, they will spend far less time mitigating vulnerabilities in the future. Teams that incorporate testing from start to end, may potentially reduce any threats and vulnerabilities that threat actors can exploit. When teams maintain the system, patching any warnings is vital to the environment’s success at keeping walls up and strong. Security needs to happen in all stages of the design and development.

**Zero trust**

Trust no one, verify everyone. It’s a common saying for those that implement the concepts and principles of “zero-trust.” According to Kueh (2020), “There are five different pillars to implement when moving to a modern, zero-trust security model.” Those pillars are device trust, user trust, transport/Session trust, application trust, and data trust. He states that, “Only trust is established across all five pillars, informed decisions can be made to grant or deny access.” Verification and reverification are crucial elements to any environment. Once the level of access changes, organizations must be able to immediately act and rectify the changes.

Timeline

Description automatically generated

Image retrieved from: <https://threatpost.com/practical-guide-zero-trust-security/151912/>

**Implementation and recommendations of security policies**

Policies promote accountability for each member of the team, and each member needs to adopt those security policies to remain vigilant. One of the first steps a team and/or organization can do is to plan what is needed to implement the appropriate security policies. Once the team determines those needs, they can apply those policies to their environment during design and development. The team must use secure perspectives to continuously manage the security policy that was determined earlier and implemented within the organization’s environment. It is crucial that the team maintains security compliance. They can do this be routinely conducting compliance checks and collect reports for any future audits.

References

Delgado, C. (2019, September 18). *Best practices for secure coding*. Our Code World. Retrieved from [https://ourcodeworld.com/articles/read/1038/best-practices-for-secure-coding](https://ourcodeworld.com/articles/read/1038/best-practices-for-secure-coding#:~:text=8%20Best%20Practices%20for%20Secure%20Coding%201%201.,privilege%20...%208%208.%20Automatic%20code%20generation%20)

Kueh, T. (2020, January 15). *A practical guide to zero-trust security*. Threatpost English Global threatpostcom. Retrieved from <https://threatpost.com/practical-guide-zero-trust-security/151912/>

IBM. (2021, September 8). *Implementing the policy*. How to implement effective security policies. Retrieved from <https://www.ibm.com/docs/en/i/7.3?topic=policies-implementing-policy>