CS-305-J7997 Software Security

Module Three Journal – Reflection

Computer Science Department

Southern New Hampshire University

Eric Farkas

3/19/2023

Although I am not currently a developer (outside the classroom), this article highlighted security risks and everyday activities that I would have considered acceptable before reading it. Professional developers must isolate their development environment from other activities to maintain not only a secure environment but also a vulnerability-free environment. This seems very challenging based on the environment I currently work in. The article points out some best practices, some things to think about and the framework for a plan to move forward. The assignment asks some questions about the article.

**What is your role in solving security concerns as a developer? What might solving security concerns as a developer involve?**

It seems that a developer has complete ownership in solving security concerns. This seems overwhelming as being a security expert could easily be a full-time job in addition to being a developer. Often organizations place the responsibility on team members with subject-matter expertise. I believe this is what is happening with developers and security. Does this mean that organizing security training falls on the developers? Are developers responsible for maintaining clean laptops for development? What determines a clean environment? I think this assignment left me with more questions than answers. It seems counterproductive for developers to work on multiple machines but it also seems to be the responsible thing to do from a security perspective. Although I work in hardware development, my current environment requires email business systems, unified communications, instant messaging and many other tools. Almost all of these allow file transfer and download. This is why I’m perplexed by the idea of segregation. I believe some of these concerns are addressed by the continuous integration environment. I’ve never worked in one of these applications, but the developers seem to work in code containers. This concept may mitigate the risk that other business applications introduce in a mixed environment.

**Where does security fall within the software stack and development life cycle?**

Security is an aspect of all stages in the SDLC. It is most heavily discussed in the requirements phase to ensure all aspects are covered and documented. The discussion continues during development and testing to ensure all aspects are implemented. Last-mile testing can be automated from scripts, but some will test security aspects like input validation, API connectivity, client/server communications and cryptography must be verified active where required.

**How might you add security measures to transform a DevOps pipeline into a DevSecOps pipeline?**

The change from DevOps to DevSecOps would be a change to the entire culture of the development teams. This change would require the teams and systems to focus on security at every process step. Training is probably the first step in the process. Although everybody on the team probably has some level of expertise regarding security, threats are constantly evolving and establishing a current threat assessment seems like a good place to start. The development process must be reviewed and security must be inserted at each step. This ensures that all documentation and deliverables have security on the checklist. It is the best possibility for success. Adding a framework for automated testing is likely the next step. If the code is being developed with security in mind, it should also be tested to ensure it is done properly and without common vulnerabilities. There may also be some need to document threats that require no mitigation. Performing these checks and eliminations on each project would be a waste of effort.

**The article suggests creating and following a plan to secure the entire DevOps life cycle. What is included in the suggested plan, and would you recommend following it?**

The article puts together a plan where the various stages of development feedback through a loop to version control. This means the code feeds back when it completes the build stage. When the code completes the test stage, it feeds back. This strategy seems to be straying away from agile and back to waterfall a bit but not completely. With a total lack of experience, I can’t say this is a bad plan but I do feel like there is a more efficient way to do it. In my introduction, I talked about culture and I feel this plan forces security by reducing efficiency. It might be better to actually spend time changing the organization’s culture and getting the teams excited about producing secure products.