Alex Surprenant

CS-305

11/17/2024

Journal: Reflection

As a software developer, solving security concerns is a critical role of the job. As the builder of the program, a good developer will have the greatest understanding of how the program functions, and will be able to identify at which areas security should be implemented, or improved upon. Developers should be conducting testing using solid, up to date security tools throughout the development lifecycle to make sure the code they are producing is secure. Developers should be using up to date security libraries and be making sure to implement validation on all inputs into the system.

A good quality, and secure software stack will involve security at all levels of the stack. Having secure code at every level will ensure a deeply secure code base, as well as create multiple layers in which the application is capable of preventing any malicious activity from occurring. Just as well, it is important to have security considerations present at all stages of the software development lifecycle. At each stage, from initial planning, through coding and testing and continuing into maintenance, up to date security measures should be considered and implemented.

DevOps can be defined as essentially a new approach to software development, in which the bridge is gapped between developers and IT professionals, to create a more collaborative approach to the software development lifecycle. With DevOps, security is usually left outside of the lifecycle and is viewed as an afterthought, and testing is only done during certain phases of the cycle (Jeganathan, 2019). To change this approach and transform DevOps into DevSecOps, security measures and testing should be integrated into all levels of the development lifecycle. Security should be integrated earlier in the planning process and security skills should be considered when hiring developers and other members of the team. I would even include Scrum Masters among those who should have at least a basic understanding of the necessity of solid security integration.

Shifting from a DevOps mindset into a DevSecOps mindset is a complete culture shift that the whole team will need to have buy-in from the top level down in order to be successful. The article suggests a plan with some decent suggestions as to how to add the security level to the DevOps lifecycle. It first suggests assessing the current release of the platform to search for any risks to the system, and evaluating their threat level. The second step suggests securing the tools used for development, which outlines securing access points based on role, protecting user login and applying two/multi factor authentication based on criticality. Third it suggests protection of critical access and API keys. Finally, the plan suggests separating duties and defining specific controls based on roles, so that members of the team don’t have access to infrastructure that isn’t necessary to their position (Jeganathan, 2019). I of course can not imagine a better plan for implementation of DevSevOps and would recommend this plan.

References:

Jeganathan, S. (2019). DevSecOps: A Systemic Approach for Secure Software Development. *ISSA Journal*, *17*(11), 20–27.