Module Three Journal

By

Dylan Cavazos

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

Being a developer and solving security concerns is a primary function of being an effective developer. As mentioned in (DevSecOps) security can often be an afterthought, not being implemented until later phases in the SDLC, specifically during testing. This often underestimates the importance of security concerns during development. As a developer it’s important to be aware of the conventional secure coding practices despite security often being overlooked until a later stage. This also helps with the overall product as it leads to greater customer satisfaction. This is one of the reasons why DevSecOps is an important field as it melds the two roles between security and development.

Solving security concerns as a developer might involve being knowledgeable in a multitude of areas, frameworks, tools, and other technologies required to fuse the two fields. For a developer who focuses on security, they may have to be well versed in various processes related to security but also be able to produce products at the pace of a regular DevOps team member. Knowing common security practices like utilizing the OWASP Top 10 vulnerabilities for web-based applications during development, as well adhering to secure coding standards for various languages like Java is crucial to addressing security concerns as a developer. During the coding stage, developing effective threat models is also a crucial skill for secure development.

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

The software development life cycle is dependent upon implementing the various aspects of each phase until the product is completed and reaches deployment. Security in a DevOps life cycle can be implemented using a new CI/CD development model. This model can implement the same cycles but include security as well. This is referred to as the Continuous Security pipeline (CS), which helps to build a process model that can be used by any organization to incorporate security into their development pipeline.

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

Security measures that can be implemented to transform a DevOps pipeline into a DevSecOps pipeline include secure design and engineering, secure testing, secure monitoring, and security risk management. Each of which help to transform the pipeline into a security focused pipeline. The security design and engineering help ensure that the products developed by the DevOps team adheres to “security best practices, regulations, standards, and laws and achieve data privacy and protection (Shapiro Library DevSecOps). Security testing is where various software modules undergo testing cycles and quality assurance. It incorporates each point of the pipeline and ensures that they are safe from security attacks. They must also adhere to the strict principles laid out from the Sans 25 and OWASP Top 10 so that the basic principles are followed. The security monitoring helps to monitor the online and offline analysis of logs created through the live systems and applications and scrutinizes them for known attacks and /or vulnerabilities. If an attack is found an alert will be sent to the security team for response. Lastly, security risk management aims at mitigating security risks by analyzing and applying cost-effective security controls. It’s important during the risk management process to continuously support the development cycle and maintain the speed expected during a DevOps model.

**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 illustrates a plan to incorporate best security practices into the DevOps life cycle so that the entire life cycle can be secured during software development. It begins with starting with a risk assessment for the new release and evaluates the threat model. Then implements multiple web-based tools like GitLab, Azure DevOps, etc. (Shapiro Library DevSecOps).”. The plan also illustrates the use of role-based access control which is a primary component of secure development. Other tactics include protecting user logon through an identity provider and web-access management tool to help mitigate the risk of violations on the principle of least privilege. Applying two-factor authentication and multi-factor authentication into the various environments and systems can also help in verifying employees’ identities and authenticated users securely to ensure they are granted the necessary access to information. Another suggestion is things like access keys, privileged service accounts, and API keys are all protected properly with privileged account security tools. Lastly, enforcing the segregation of duties for various roles to ensure various roles are granted access to the items they depend on.

To incorporate the security practices and principles into a DevOps lifecycle I think it’s important to follow a strict plan that lays out the best security principles. This will help in the event there is any attack or breach of sensitive customer information or worse. While this may impede the speed at which developers are able to complete their tasks or meet potential deadlines, the consequences are far worse of best security practices are not followed which can ultimately result in a more catastrophic scenario for the company and/or customer. Therefore, I believe it would be wise to incorporate this into the development process and lifecycle.

**https://eds-s-ebscohost-com.ezproxy.snhu.edu/eds/pdfviewer/pdfviewer?vid=1&sid=cf1060bc-33e9-4ceb-91cf-9e9c9fcbb5ac%40redis**