3-2 Journal: Reflection

SNHU

CS-305 Software Security

Nicholle Caudy

In my developer role, solving security concerns is vital to my job. I need to always develop all codes with security at the forefront of my mind and practices. I need to use the available tools such as Fortify, Checkmarx, OWASP, etc. Adopting security principles early on as a developer is a great way to start these principles include integrity, accountability, availability, and confidentiality. I also need to avoid using any vulnerable components. I will work in smaller chunks and continuously do security checks on my code so that security is always present and not left to the end to check. This will prevent having to return and rework code as well as ensure that the application we will be delivering is secure and has the least number of vulnerabilities possible for our customer and avoid any lost time for making fixes later down the road.

Software security is present at all phases of the software stack and development lifecycle. It needs to be present and planned for in the planning phase, developers need to use secure coding practices throughout the development phase. The build phase also needs to follow security processes to avoid unauthorized access and use the industry-followed automated tools available for building software modules. During testing extensive system and integration testing is completed to help prevent any flaws in the modules. The red team launches offensive attacks while the blue team works on preventing these attacks by the red team. Security scanning for web applications is completed to look for vulnerabilities. In the deploy and release phase, automated validations are used to scan for vulnerabilities. The operation phase is still focused on security as the security team is actively involved in monitoring, protecting, and analyzing the live environment.

I would add security measures to transform a DevOps pipeline into a DevSecOps pipeline by starting small and creating a focus group that implements the new changes and gets comfortable with the DevSecOps pipeline and then branches out as it becomes more successful. Failures have been considered and adjusted for. We can then integrate those people from the focus group into other applications just starting to be produced to help lead new teams in the way of DevSecOps. This is mentioned in the article, and I think it is a great idea. The whole company does not need to adopt this all at once, it can happen slowly and methodically. The article also suggests the DevOps team implements controls during the coding process such as controls for input validation, user credential validation, session management, user control access, data protection, logging, API security, and detecting security misconfigurations. The article also points out that security testing is a vital part of the new pipeline and should not simply focus on the module but the end-to-end pipeline, infrastructure, live production systems, and other aspects to protect the application from attacks. I would recommend following this suggested plan because it involves every phase throughout the entire process and starting small instead of trying to force this on every team all at once. After all, trying to make these changes on a mass scale could cause mass failures and likely little to no buy-in from the teams which is vital if DevSecOps changes are to be implemented and successful.

References

* Jeganathan, S. (2019). DevSecOps: A Systemic Approach for Software Development. *ISSA Journal*, *17*(Developing and Connecting Cybersecurity Leaders Globally), 20-27. <https://doi.org/https://eds-p-ebscohost-com.ezproxy.snhu.edu/eds/detail/detail?vid=2&sid=7dbd33f0-0db0-4a95-8639-7483b1dac66d%40redis&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=139557317&db=tsh>