**Brandon Hobbs**

**CS-305**

**May 20, 2022**

**Journal: DevSecOps**

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

Developers are the source of all security flaws. Does this mean that another team should “clean up” those mistakes? Probably not as with most things shifting left saves time and a potential disaster. “The shift-left movement, which pushes security as far to the left in the development lifecycle as possible, calls for every developer to focus on security” (Romeo).

Some of this shift left is the idea of secure coding which “…is a development practice in which security-related weaknesses, defects, and integration errors are addressed by following the established simple and complex threat models” (Jeganathan, 2019).

Automation can also help. In the Ci/CD pipelines for a project one pipeline might be static and dynamic testing. Many companies offer paid subscriptions so that teams can consume tools that do this automatically, e.g., Veracode.

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

Security starts before the code many times. Security is supposed to be the first thought, not the last. This means when the architecture is being planned security is being planned. Threat models should be generated early.

Unit Tests, which are written concurrently to code, should be written also with a view to security.

Lastly, even after the code is commercial if there are maintenance/sustaining contracts then there needs to be a view on the changing landscape of security. What are the new threats, are we susceptible, how sever does this affect my users, **do we need a patch**?

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

A simple idea is to integrate SAST and DAST into the CI/CD pipelines. Veracode is an often-used tool that integrates in large scale IDEs such as AzureDevOps.

***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 plan suggested by Jeganathn (2019) in the journal is:

* *Create a rapid, high-level threat model*
* *Quantify the risks*
* *Secure the DevOps lifecycle tool, e.g., Azure DevOps or GitLab*
  + *Secure access points*
  + *Protect user logons by using SSO/federation*
  + *Apply 2FA/MFA*
* *Secure keys, service accounts, API keys, etc.*
* *Define infrastructure protection controls, e.g., do developers actually need access to the production environment?*

Honestly, the plan is as good as any. As a list of things to pay attention it is fairly complete. I might add that testing also needs to be planned. For example, some data used in testing is confidential and also needs to be secured – or even controlled its access. This needs a plan and maybe even documentation on who and when they had access.

**Sources:**

Jeganathan, S. (2019). DevSecOps. A systemic approach for secure software development. *ISSA Journal*, *17*(11), 20-27.

Romeo, C. (n.d.). *How developers can take the lead on security*. TechBeacon. <https://techbeacon.com/security/how-developers-can-take-lead-security>