Tim Jayson

Cs-305 Module 3 Journal

9/19/2025

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

A developer is an integral part of solving security concerns. The developer is literally the first line of defense in any software project as the quality of their code determines how vulnerable it is to attack. They are able to plan and implement secure code from the very beginning of the project all the way through patching a live system as vulnerabilities become known. This starts with designing and implementing secure code that is properly commented, tested, and maintained. A developer should consider input validation, encapsulation, error handling, secure login, and granting the least needed level of access.

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

Security is an integral part of each phase of the software stack and development life cycle. The importance of security at each phase can help to enhance the overall security of the software.

* Planning

In this phase of the software development life cycle, it is important to figure out the requirements of the software and possible security requirements. It is important to consider possible vulnerabilities, malicious attackers, and the best way to make the software secure from the start.

* Design

In this phase of the software development life cycle, the development team start to implement their security planning into the software. This is where they plan the security features into the code and they are able to determine how the software handles certain security concerns.

* Implementation

In this phase of the software development life cycle, the development team codes the security features into the software. Ideally they will code the software with secure practices and writing the code in a way that helps to mitigate the security concerns. It is important to have detailed comments and review the code to ensure that it is robust and easy for the development team to reference.

* Testing

In this phase of the software development life cycle, the testing team tries to find bugs or breaks in the software. They should also test it to determine how robust and secure the software is. This can be an very important step in not only ensuring the planned security features work but also find other vulnerabilities in the code.

* Deployment

In this phase of the software development life cycle, the software is released to the client/public. Security at this stage is making sure that the security features in the code are working properly. It is important to monitor any potential vulnerabilities during the release to ensure that the software is secure.

* Maintenance

In this phase of the software development life cycle, the software has been turned over to the client and the code is maintained by the development team. This includes adding patches to ensure that any potential security issues are patched to keep the software safe.

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

Adding security measures to a DevOps pipeline is critical for it to transform into a DevSecOps pipeline. This can be accomplished by emphasizing security and ensuring that it is continuously integrated into the development. Anther way to achieve a DevSecOps pipeline is to integrate the use of security controls and tools from the beginning of the project and work to reduce the vulnerabilities in software production. These changes can help to reduce the security issues that may arise later in the project or after it has been turned over to the client.

* The article suggests creating and following a plan to secure the entire DevOps life cycle. What is included in the suggested plan? Would you recommend following the plan? (I was unable to get the article to load so I used another one, please see the citation below.)

A good plan for securing the entire DevOps life cycle includes the following.

* Continuous Integration: During this phase the developers frequently commit their code to a repository and the code is automatically tested to catch issues with the integration early as well as any bugs in the code.
* Continuous Delivery: This phase builds upon the continuous integration to automate the process of moving code from the build environment to a staging environment. The software is automatically tested to ensure the software and user interface is working as designed.
* Continuous Security: During this phase the development team tests the software with automated security testing early and often throughout the software development process.
* Communication and collaboration: It is important for multiple teams to be able to work closely together to create secure and robust code. This can be enhanced by individuals and teams collaborating to find and address issues with the coding.

I would recommend the plan to implement a secure DevOps pipeline. This will help the development team to produce secure and robust code to the end user. This plan will help to ensure that the software is tested early and often to discover vulnerabilities that attackers could exploit.

Sources:

GeeksforGeeks. (2025b, August 9). *What is Secure Software Development Life Cycle*

*(SSDLC )?*. GeeksforGeeks. <https://www.geeksforgeeks.org/ethical-hacking/what-is->secure-software-development-life-cycle-ssdlc/

GeeksforGeeks. (2025, July 23). *Difference between DevOps and devsecops*.

GeeksforGeeks. <https://www.geeksforgeeks.org/devops/difference-between->devops-and-devsecops/

Microsoft. (n.d.). *What is DevSecOps? definition and best practices: Microsoft security*.

What Is DevSecOps? Definition and Best Practices | Microsoft Security. https://www.microsoft.com/en-us/security/business/security-101/what-is-devsecops?msockid=39cfb2b87ba669e4199dbdc97a5b6835