DevSecOps Plays a Key Role in Modernizing Software Innovation and Delivery

In recent years, the defense agencies have made major strides in modernizing its software development process to enable increasingly rapid innovation, as detailed in Airforce Magazine. To optimize its software innovation and delivery capabilities in support of mission execution, one agency's software development site evaluated various tools and found an answer in DevSecOps functionality to boost its coding efforts.

Software Factory Tooling – Less is More

Technology-driven whirlwind change across all industries has led organizations to cast a wide net in search of tools to perform the many functional tasks needed in successful software deployment. The key desired features of these tools are **enablement of innovation**, **speed of mission delivery**, and **threat reduction**.

Traditionally, software development environments have included multiple siloed tools, sometimes up to 30, which may meet the need for a range in performance but presents a number of challenges such as:

- Fitting each software component into the overall security posture
- Struggling with obtaining end-to-end visibility, lack of a 'cookie trail' to track issues
- Continuous updating and maintenance needs for each tool
- Complicated communication between different products caused by absence of a common interface

Being able to replace many solutions with a single one, such as GitLab, offers obvious benefits.

Mobile Applications Security Requirements and Scanners

There are many common mobile application vulnerabilities calling for comprehensive security measures to defend against them. Common practices include inspecting the structure of the source code; static application security testing (SAST) with a focus on the compiling process; dependency scanning to identify snippets of code introduced from open source; and fuzz testing – an AI driven security methodology that simulates attacks at both interface level and API program interface level.

Additionally, it is important to accurately monitor active logins and proper application use, which involves appointing an application that's either vulnerability free or accepting of its risk profile, authentication and access of the application with regard to what users can do with it, as well as the way the app connects to the network.

Again, multiple tools can be used to tackle the different aspects of security – examining the source code, the application, and the dependencies for vulnerabilities and container exposure. This siloed approach, however, brings about inefficiencies in scanning performance, end-to-end visibility, reportability, and accountability. A software factory approach reduces the number of security tools involved and raises the prospect of overall robust safeguarding.

Defense Agency Challenges and Resolutions

On a journey to leverage improved software development to enhance mission delivery, an agency tackled several cultural and organizational challenges.

Shedding legacy approach to coding, which often involves spending an exorbitant amount of time gathering requirements, writing code, securing it, and finally deploying – by which time both requirements and security posture will have changed. Working in very short cycles, the software factory method can produce a minimum viable product in just a few days, while accommodating nimble reactions to altered requirements. The military 's readiness to adapt helped overcome many challenges associated with the project.

Bringing in specialized expertise on temporary duty. Programmers writing code for missile systems, for example, can receive valuable feedback from the front-line pilot perspective of firing those missiles. To facilitate such multi-departmental collaboration, a shared coding environment enables a range of contributors from airmen to civilians to interns to write code in a shared office environment off base.

Leveraging pre-approved software components to speed up deployment. As with any federal government entity, the defense departments have requirements for authority to operate (ATO) – a lengthy and rigorous security scanning process. To curtail potential ATO related project delays, the Air Force utilized DOD's Iron Bank, a centralized repository

of digitally signed, already hardened container images, allowing for more agile development.

Building developer teams' skill sets through a two-fold approach centered on the software factory's ability to carefully assign and monitor varying levels of access, permission sets, and organizational authorizations. From one angle, interns and less experienced developers can gain experience working in the open and internet-connected general development environment before graduating to the separate restricted access side, insulated from the internet. From another aspect, teaming and mentorship between more senior staff and those with less experience is enabled by the software's participatory dynamic and viewing only guest access options.

Targeting appropriate end user audiences through user authentication and other relevant protocols. While the broadest types of applications need to be accessible by anyone in the service, there are narrower settings for more niche apps, such as for flight deck mechanics to log issues with an airframe, or an app geared towards Human Resources (HR) personnel.

Optimizing overall development team performance through centralizing the project tracking and management processes in a DevSecOps environment. Not only does the military need to comply with DOD regulations governing roles and permissions. It is just as crucial to zero in on project leaders' oversight on assignment distribution, deliverables at risk, outstanding issue priorities, and overall optimization of project portfolio management.

Ensuring application security compliance through an integrated third-party end-to-end security solution. For cases where highly sensitive corporate or mission data sets must adhere to a particularly stringent set of NIAP security controls, Monkton has created an application development framework that enables mobile developers to inherit all the security controls and produce fully NIAP-compliant apps from the very beginning. In a software factory environment, such niche requirements can be addressed by integrating with appropriate partner solutions.

Expediting innovation through DevSecOps methodology, which enables full collaboration across all of an organization's workforce. Developers, pilots, security experts, and other stakeholders have visibility and access to participate, comment, share and generally monitor the project direction and progress.

GitLab's Software Factory Solution in Support of Mission

To optimize mission support, GitLab prioritizes understanding the customer's challenges and goals, day-to-day process steps, and what is ultimately the purpose of the application they are trying to deliver. Only then can GitLab start applying ways to assist with the process and the end goal.

First off, GitLab's mission is to minimize risk, while having complete understanding of the risk profile. Being a one-stop shop, its built-in security scanners are comparable in scope and efficacy to solo solutions, but more importantly they become an integral part of the process. Handling a full range of security functions with a single tool results in a simpler coding process and reduced external vendor involvement.

Potential daily changes in vulnerabilities warrant frequent scans on an organization's runtime environment. In case of a sudden break, a bread crumb trail is needed to understand what, when, why, and how happened, at the touch of who's hands. Enabling seamless backtracking is an advantage of GitLab's software factory approach.

Obviously, many organizations now on their DevSecOps journey have previously procured siloed solutions. To help protect that investment, Gitlab openly integrates with other solutions, enabling customers to continue using a legacy product while evaluating migration options. This integration capability also allows folding in other complementary products for added functionality, or pre-certified software components from the DOD Iron Bank repository.

The centralized tooling in a software factory promotes collaboration between departments, teaming among contributors of varying skill levels, and an agile framework to work in short iterative cycles to write code, find problems, fix them, and get to app deployment.

A portfolio of in-person and virtual instructional programs include online workshops, recorded sessions for self-paced learning, and less formal lunch-and-learns. The training is focused on demonstrating real life software applications, best practices in stakeholder collaboration, and team management within a unified framework. It is GitLab's goal to help the customer from the beginning to the end of whatever process they are working on as expeditiously and efficiently as possible.

For more information, please contact us.