Heading Off Risk:

A Unified Approach to Application Security and Delivery

State and local governments are developing applications and digital services as fast as they can to meet urgent needs. In addition to contending with accelerated deployment schedules and demands for complex functionality, their development teams must still perform security testing to ensure applications are free from vulnerabilities.

Application security has taken center stage as investigators get a fuller picture of the recent nation-sponsored attacks that exploited vulnerabilities in SolarWinds, Microsoft and other vendors' software to enter systems, steal data, demand ransom payments and wreak havoc on federal, state and local government organizations. With billions of dollars in American Rescue Plan funds starting to flow to state and local governments, the motivation to target government applications will only increase.

To strengthen application security and accelerate delivery, many organizations are turning to a unified DevSecOps approach that incorporates security design, implementation and testing into the software development life cycle sooner. The intent is to catch and remediate security issues early in development so coding errors do not proliferate into finished code and IT teams do not have to backtrack when they discover an error.

A mature, unified DevSecOps solution enables organizations to realize the full value of "shifting left" by integrating and unifying toolsets to get full visibility into security issues across the development life cycle, automating key steps in the application security testing process, and incorporating just-in-time training so developers can get the help they need when they need it. Using this approach, organizations can strengthen

application security, expedite application development, and keep developers and security teams more engaged and focused on meeting an organization's core mission.

A recipe for trouble: new complexities and scattered approaches

The following trends and challenges drive the need for a more unified approach to application security:

- **Demand for rapid application delivery.** An urgent need for digital services and remote work has forced development teams to work faster than ever. Nearly 80 percent of organizations forward vulnerable code to production, often to meet critical deadlines or because issues were discovered too late in the release cycle.[1]

- **Ransomware and other attacks that target application vulnerabilities**. Thirty percent of successful attacks happen at the application layer.[2] The attack that targeted vulnerabilities in SolarWinds and other vendors' applications offered a lesson no organization wants to repeat.

- **Proliferation of IoT devices.** As internet of things (IoT) devices are increasingly integrated into workflow processes, they become a doorway to exploitation. In Florida, for example, a hacker unsuccessfully attempted to poison a municipal water supply by using an IoT/operational technology device to remotely access water management software on an outdated, unsupported computer.[3]

- **Cloud-native development.** Analysts project that by 2022, 90 percent of new applications will be developed using agile methodologies and API-driven architectures that leverage microservices, containers and serverless functions.[4] Undiscovered vulnerabilities in these chunks of code can quickly proliferate into other infrastructure and applications.

- **Reliance on open-source software.** According to ESG, 80 percent of development teams draw at least 26 to 49 percent of their code from open-source libraries; however, fewer than half are using tools such as software composition analysis to test the security of open-source code.[5]

- **Insufficiently trained developers.** Most developers are not adequately trained in security-related practices for writing and remediating code. While proper training helps eliminate design flaws at the root, few top universities embed cybersecurity training into their coding programs, and traditional teaching approaches don't always provide the immediacy and relevance that developers seek.

- **Ad hoc and incomplete toolsets for app security.** Agencies usually have and array of app security testing tools, but many of them are siloed or ad hoc, niche tools that once addressed an immediate security challenge but now are rarely used. In many cases, organizations have difficulty aggregating data from all these tools and integrating security tool data with their development tools.

A unified approach to application security

In a recent survey, software developers identified the most impactful things that their organization could do to simplify management of application security. Their top choices were more opportunities for application security training (36 percent); integrating security testing directly into workflows (27 percent); and investing more in automated security testing tools (23 percent).[6]

A modern, unified approach to application security addresses these needs and strengthens application security — without interrupting development workflows. This approach focuses on aligning resources to create a consolidated strategy and execution plan for application security.

Again, the strategy is based on a DevSecOps approach that embeds security into the entire software development life cycle.

The execution is done via a centralized, comprehensive and highly automated application security testing platform that allows organizations to identify, triage, intelligently prioritize and remediate security risks from the coding stage through runtime application testing. The platform tightly integrates security testing tools as well as software release orchestration and agile planning to provide a consolidated view of risks, unify and centralize user and policy management, automatically aggregate scanning data from across the application security testing toolset and prioritize remediation.

With a DevSecOps approach and a unified, enterprise-grade testing platform, organizations can automatically scan source code for vulnerabilities before it's even compiled — and before developers have invested hundreds of hours in writing entire packages of code. When scans identify an issue, intelligent remediation tools automatically suggest a path to resolution.

This modern, unified approach to development addresses a fundamental drawback of traditional app development.

"A lot of development teams say they discovered issues too late in the release cycle, and they didn't have time to

With a unified approach, says former New York state Chief Information Security Officer Deborah Snyder, “right from the start, you get out ahead of specifying and designing the security requirements.”

go back and fix them prior to the migration. That has to stop,” says Deborah Snyder, Center for Digital Government Senior Fellow and former chief information security officer for New York state. “There has to be sufficient runway and preliminary testing, and testing throughout the whole development cycle.”

By nipping issues in the bud, organizations strengthen security, accelerate time to application delivery and give development teams more time to focus on innovation. Shifting left — along with centralization — also allows organizations to apply best practices consistently and mature their “security by design” approach incrementally.

Instead of working with siloed versions of security roles and policies and then passing around PDF files to update co-workers, developers can collaborate in real time to work security issues as tickets within their workflow. They can start from a small focus set of recurring issues and move to broader sets over time. Besides creating a sense of ownership among developers and improving code quality overall, a unified approach delivers significant returns in other areas.

“Right from the start, you get out ahead of specifying and designing the security requirements,” says Snyder. “You integrate that discussion along the whole route. But this concept of shifting left also has a secondary benefit. It reduces the cost of remediation. One recent study projected potential savings of up to 30 times per bug when the problem is fixed earlier in the development process versus later,” says Snyder.

Critical tools

No single app security testing tool can do it all. To find and correct security vulnerabilities early in the development cycle, organizations can integrate a range of tools into their centralized app security testing platform.

🟩 **Static application security testing (SAST).** With traditional approaches, developers wait for the QA phase to perform dynamic application security testing (DAST), which uses penetration tests and other tools to identify configuration errors and other exploitable vulnerabilities. SAST tools check for flaws and vulnerabilities in source code before compiling code (literally, down to the line number of where a vulnerability exists) and provide intelligent remediation advice so developers know exactly how to fix the code. Depending on the vendor, developers can scan all their different development languages with a single tool. They can also customize code scans to address the organization’s unique coding requirements and provide the most valuable impact in the least amount of time — for example by scanning a targeted set of criteria or prioritizing next steps based on results.

🟩 **Interactive application security testing (IAST).** “Very integrated leading organizations employ interactive application security testing, where they combine the elements of both SAST and DAST,” says Snyder. IAST allows developers to find security issues in real time, during functional testing of code. An agent runs in the background to continuously monitor the running application — without actually exploiting the application through invasive penetration testing or interrupting the development process. Real-time results greatly reduce the time and operational overhead associated with the traditional DAST process.

🟩 **Software composition analysis (SCA).** SCA tools check dependencies and publicly documented common vulnerabilities and exposures (CVEs) associated with open-source or third-party code libraries. In many cases, these tools also allow developers to find newer software versions that have already resolved a particular vulnerability.

🟩 **Secure coding education for developers.** In mature DevSecOps environments, application security training is an ongoing process that begins as soon as a new developer comes on board and continues throughout their career. “Software development has undergone significant changes, and security has to keep up with the rapid speed at which code is produced, released and re-used,” Snyder says. “Everything’s moving very rapidly, and that means investing in your development teams. Every coder has some understanding of secure applications, but most coders don’t have sufficient knowledge to code securely. They’ve got to be trained how to think like a hacker and how to code for those things.”

Interactive, just-in-time training tools are often the best approach. As soon as a SAST or other tool discovers a vulnerability, a developer can link directly to content that helps them understand the vulnerability and how best to remediate it. By keeping training focused and in small chunks, these tools make training more relevant, memorable and engaging.

Getting started

The following best practices help organizations get started on a more unified DevSecOps approach to secure development:

- **Gain executive sponsorship.** Active executive sponsorship helps ensure initiatives to shift left and centralize application security are sufficiently funded, remain a priority and are seen through to full execution.

"Security and risk management leaders have a huge role here. They need to lean into this area of risk and evaluate whether their organization has the maturity and capability in terms of seamlessly integrating and automating testing and building in security by design. The tighter the deadlines and development cycles, the more critical the need to reduce potential vulnerabilities, especially in complex, distributed application environments," says Snyder.

- **Establish goals and success criteria.** Identify metrics or key performance indicators for the overall initiative (e.g., reduction in coding errors, faster time to application delivery or lower remediation costs) as well as technology under consideration (e.g., fewer false positives).

- **Develop a framework that addresses people and processes.** Success criteria, proofs of concept and tool demonstrations won't matter if the right people and processes are not in place to make sure tools are properly implemented and used.

- **Educate both developers and security teams.** Make developers aware of what vulnerabilities exist in their code and teach them in a way that they don't repeat the same mistake in the future. On the flip side, educate security teams about how software is developed so they can provide meaningful input on how to avoid or remediate security issues.

- **Work with an established industry partner.** Look for a partner with proven technology, core expertise in application security and hands-on experience working with public sector organizations. In many cases, industry leaders have team members who have committed their entire career to advancing the public sector. By relying on their partner's subject matter expertise, organizations can focus on their core initiatives.

As the pace of application development escalates, organizations need to shift even further left on security testing. Shifting further left includes embedding security testing into the very first lines of code, training developers to code securely, and centralizing security tools onto an enterprise-grade platform that improves visibility and enables intelligent automation of scanning and remediation tasks. It also includes the recognition that security is a constantly moving target. To keep applications secure, organizations must instill continuous vigilance, learning and improvement into their application development practices and culture.

This paper was written and produced by the Center for Digital Government Content Studio, with information and input from Checkmarx.

Endnotes:

1. Enterprise Strategy Group. Modern Application Development Security. August 2020. https://www.prnewswire.com/news-releases/devsecops-study-finds-that-nearly-half-of-organizations-consciously-deploy-vulnerable-applications-due-to-time-pressures-301107632.html
2. Checkmarx. Deliver Secure Software at the Speed of DevOps.
3. K. Walker, Government Technology & Services Coalition. Managing Necessary Risks in Critical Infrastructure Tech After Florida Water Hack. March 2021. https://www.hstoday.us/subject-matter-areas/infrastructure-security/managing-necessary-risks-in-critical-infrastructure-technology-after-the-florida-water-hack/
4. IDC. DC FutureScape: Worldwide Cloud 2020 Predictions. October 2019. https://www.idc.com/getdoc.jsp?containerId=US44640719
5. Enterprise Strategy Group. Modern Application Development Security. August 2020. https://www.prnewswire.com/news-releases/devsecops-study-finds-that-nearly-half-of-organizations-consciously-deploy-vulnerable-applications-due-to-time-pressures-301107632.html
6. J. Brotsos, Checkmarx. Developer Pulse Check: Coding Amidst COVID-19, One Year Later. March 2021. https://www.checkmarx.com/blog/developer-pulse-check-coding-amidst-covid19-one-year-later/

Produced by: CENTER FOR DIGITAL GOVERNMENT

The Center for Digital Government, a division of e.Republic, is a national research and advisory institute on information technology policies and best practices in state and local government. Through its diverse and dynamic programs and services, the Center provides public and private sector leaders with decision support, knowledge and opportunities to help them effectively incorporate new technologies in the 21st century. **www.centerdigitalgov.com.**

For: Checkmarx

Visit **www.checkmarx.com/public-sector** for more information on how to unify your Application Security.