LTIMindtree

POV

The DevSecOps Journey

Improve delivery excellence with DevSecOps practices and Maturity model

Table of Contents

Introduction 3

The Need For Devsecops 4

What Is Devsecops? 6

How Devsecops Fits In 7

Tools Categories 8

Standards & Maturity Model 9

Key Benefits & Success Areas 11

Conclusion 13

References 14

Authors Profile 15

01 Introduction

"Fast and secure code delivery" may seem like an oxymoron to most companies, but DevSecOps is determined to change that. DevSecOps is a way of approaching IT security with an "everyone is responsible for security" mindset.This integrates security practices into an application's DevOps pipeline. The aim is to integrate security into all stages of the software development workflow. The discovery of a vulnerability in an extensively used open-source library has always been a common cause of panic across the IT industry. Similar chaos emanated from the Log4j vulnerability-related outbreak early last year.

Log4j refers to an open-source logging library widely used by apps and services on the web. If the problem is not resolved quickly, attackers could break into systems, steal passwords and credentials, extract data, and infect networks with malware.

The outbreak showed how well teams were equipped to manage such a catastrophic event. While some teams could identify, evaluate, and mitigate it easily, others had to go through a lot of code to determine its impact.

This point of view introduces DevSecOps as an evolving field in the software development landscape. It explains why we need it, how we can learn about its practices, the different categories of DevSecOps tools, and the benefits of successful implementations. While this is an evolving practice, we also look at the widely accepted maturity model, helping teams to evaluate where they are and what they can achieve on their DevSecOps journey. We have also framed our maturity model based on extensive analysis of the maturity models available and well accepted across industry. It is intended to help companies categorize their DevSecOps processes into different maturity levels.

02 The Need For DevSecOps

There has been a rapid increase in cloud adoption, large-scale digitization, and digital transformation of businesses. This calls for business processes and applications to be accessible anywhere, anytime, from any device. The pandemic, too, has significantly influenced accelerating digital adoption among businesses. Amidst this, the vulnerabilities and cyber threats that businesses face have risen exponentially. Such proliferation of digital usage has multiplied the attack vectors and is evident in the stats below.

Some alarming cyber statistics: Vulnerabilities Identified By Year

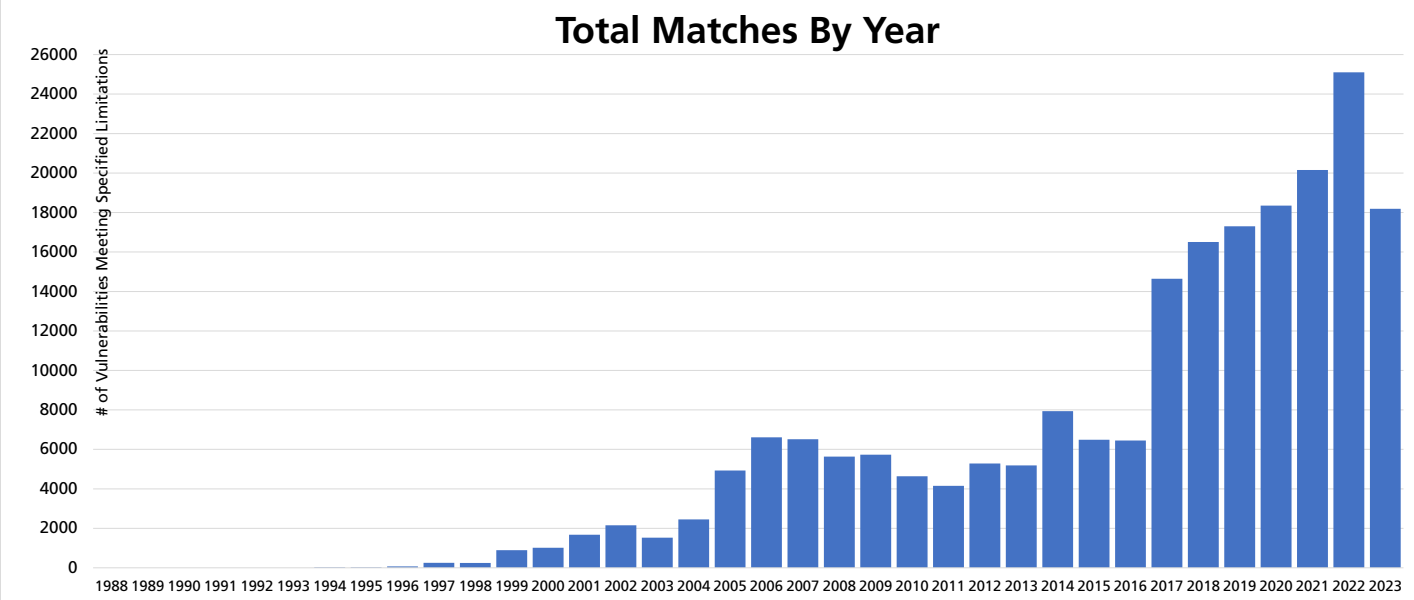

Fig 1: Number of vulnerabilities meeting specified limitations by year : https://nvd.nist.gov/

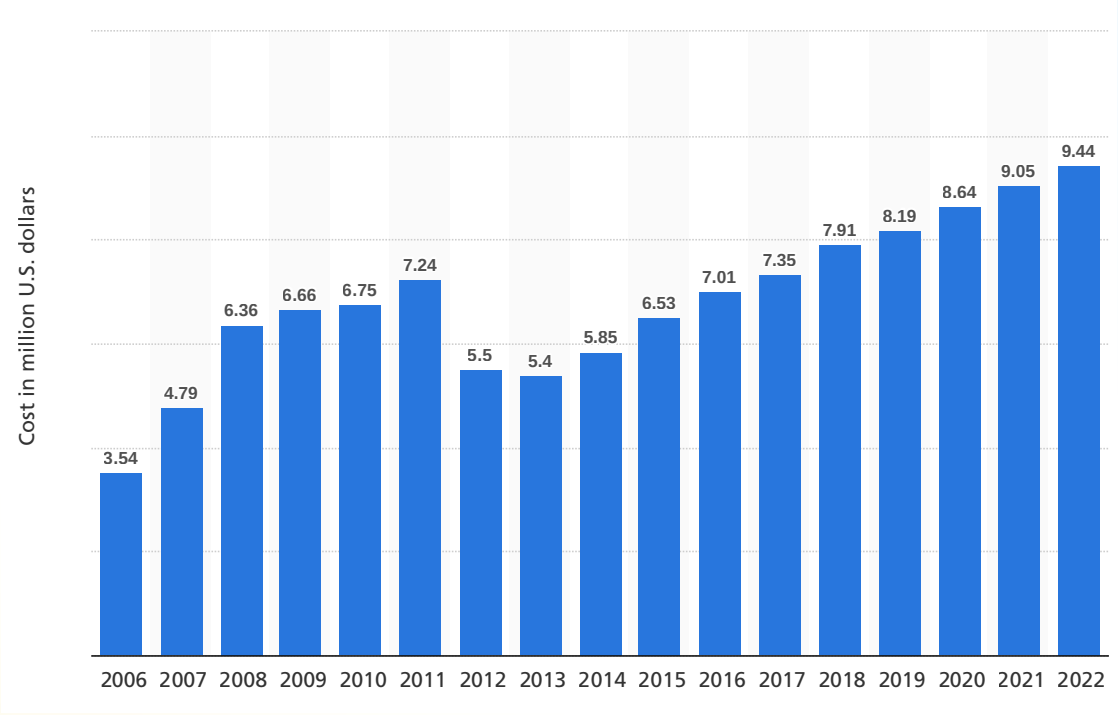

Fig 2: Average cost of a data breach in the US from 2006 to 2022 (in million U.S. dollars) : https://www.statista.com

With such significant impacts, companies view application security as an integral part of the software development life cycle (SDLC) starting from Day Zero. This is where DevSecOps comes into the picture.

03 What Is DevSecOps?

DevOps have been at the forefront of all modern applications in recent years. It has been championing rapid development, quicker go-to-market, and agility with continuous integration and delivery. Considered one of the best practices, it has given rise to a plethora of specialized tools, techniques, technologies, and associated skills. An essential facilitator of quicker go-to-market, engineering teams have willingly adopted DevOps processes as part of their product lifecycle.

DevSecOps is an attempt to get security included in the regular product development lifecycle at an early stage as a critical component of development. It aims to change software engineers' mindset to a shift-left approach. The core development team takes responsibility for application security instead of security testers or engineers brought in at a later stage.

While there are limitations to manual security checks in terms of efforts, human errors, etc., DevSecOps further enhances these processes. It helps automate, report, and alert security issues along with the regular CI/CD pipelines.

04 How DevSecOps Fits In

Although DevSecOps is a continuous process, several steps can be performed in different stages of the SDLC. Here are the steps and the activities that can be performed at every stage of the DevSecOps journey –

Plan 01

- Evaluating and identifying security tools
- Identifying security experts
- Security training
- Eliciting security requirements and goals
- Defining SLAs for security bug resolution based on severity.
- Approval flow for non-compliance

Design 02

- Designing the DevSecOps pipelines
- Threat modeling
- Setting up SAST and DAST scans
- Defining schedules of pipeline execution
- Configuring recipients of the security reports
- Formalizing steps for post identification
- Configuring alerts
- Establishing triage mechanism

Implement 03

- Executing the pipeline as per schedule
- Triaging issues observed
- Creating security backlog
- Resolving identified security bugs
- Rescanning
- Hardening security

Review 04

- Performing network scans
- Testing vulnerability and penetration
- Verifying scan reports
- Fixing any defects

Monitor 05

- Monitor alerts & events
- Perform security scans as per scheduled frequency with tools such as Splunk Enterprise Security, IBM Security QRadar, and other SIEM tools that are available to use at this stage.
- Manage incidents

05 Tools Categories

One can choose specific tools for each stage of the Software Development Lifecycle. Each tool performs specific functions from inception to post-production monitoring, enabling security as a continuous process.

Here are some key categories of tools that one can use in different phases.

Plan

- Threat modeling
- Security training
- Planning and issue tracking

Design & Implement

- Static Application Security testing (SAST)
- Dynamic Application Security testing (DAST)
- Mobile Application Security testing
- API Security testing
- Secure key and certificate management

Review

- Penetration Testing
- Network scans
- Infrastructure as Code scanning
- Cloud security Posture management

Monitor

- Web application firewall
- API protection suite
- Security Information and Event Management
- Automated key/certificate rotation
- Intrusion detection systems

06 Standards & Maturity Model

Standards

The values of DevSecOps, as outlined by the National Institute of Standards and Technology (NIST) , are -

- Reduction of vulnerabilities, the malicious code without slowing down teams
- Mitigation of the potential impact of vulnerability exploitation throughout the application lifecycle
- Addressal of root causes of vulnerabilities
- Reduction of friction between different teams, such as development, security, and operations

National Cybersecurity Center of Excellence (NCCoE), a branch of NIST, has launched a project for DevSecOps called Software Supply Chain and DevOps Security Practices. The project focuses on developing and documenting an applied risk-based approach and recommendations for DevSecOps practices. The outcome is to provide guidelines as part of a NIST cybersecurity practice guide to integrate security practices into the development lifecycle seamlessly.

Maturity Model

DevSecOps is a relatively new approach to software development, and the standards are still being researched and formed. In the interim, different nonprofit organizations and open platforms have developed their maturity models for DevSecOps implementation.

Most maturity models are primarily based on delivery speed, automation, process efficiency, collaboration, security awareness, and standardized approaches.

We have created a maturity model based on our study of the DevSecOps maturity models proposed by GitLabs and other such organizations. It is designed to help organizations categorize their DevSecOps processes in different maturity levels.

Beginner

- Manual activities
- Remediation of defects is slow and performed at later stages
- Processes such deployment, testing and application security testing are performed ad hoc
- Only dedicated team members are aware of the overall processes.
- Documentation of the steps in the process are needed
- Change of personnel rely on accurate documentation for knowledge transfer
- Minimal time spent to improve the existing processes

Intermediate

- Partial automation of processes such as deployment and unit testing
- Manual intervention required to execute the deployment pipelines
- Static code analysis tools are used by developers to identify and fix vulnerabilities
- Tracking of security defects is done independently
- Teams are dependent on other teams to provision key Infrastructure components
- Remediation of defects is done in a more proactive manner with consideration to severity, priority
- Standard operating procedures, recommendations, common toolsets are identified, however, large scale adoption across teams is still to be achieved

Advanced

- Complete automation of multiple processes including automated testing, code quality checks, deployment
- Security processes such as execution of SAST and DAST tools are integrated in the development phase itself
- Security training is conducted, and all team members have a basic understanding of secure coding practices and different types of vulnerabilities
- Defects are logged in work administration tools and remediated as per their severity and priority
- Time for resolving security defects is accounted for in each development cycle/sprint
- Teams can push features at a much quicker rate

Highly Advanced

- In addition to the Advanced level objectives, the teams operate with higher levels of automation
- Quality measures configured in code quality and other automated tools
- Automated creation of defects in centralized work administration tools
- Mandatory security training
- Automated policies and regular compliance checks are applied to infrastructure resources
- Inclusion of security processes from inception phase itself for every application
- Continuous improvement of security processes, practices, and tools with complete adoption by teams

07 Key Benefits & Success Areas

Key Benefits

- Vulnerabilities are **identified early** in the SDLC.
- **Industry-proven** tools are used to identify issues related to vulnerabilities in code, libraries referenced, and base libraries used in underlying operating systems.
- **Automation is leveraged** extensively to identify flagged vulnerabilities, along with additional information, such as the level of severity, which helps the developer address the issue appropriately.
- Evaluating open-source libraries involves **automated license checks** that help flag any license-related issues. It helps teams to ensure libraries with the right license type are used.
- **Security as a practice** is embedded into the product development life cycle and is duly adopted by teams. It leads to a more security-aware work culture in the organization.
- **Regular security training** and involving team members in the security aspects right from initial development leads to more security experts in the teams and the organization.
- Software development teams become **more aware and gain knowledge of the security aspects.**
- Early identification, diagnosis, and remediation of security defects lead to **savings in costs and efforts**

Success Areas

Here are some areas where qualitative benefits were observed from programs where DevSecOps implementations were successful -

Use of automated tools for security checks and reports generated by them to identify vulnerable third-party libraries.

This led to quicker identification of code vulnerabilities and container images and, therefore, quicker remediation. E.g., The use of a vulnerable Log4j library was identified from automated tool reports.

Outdated libraries in base container images having vulnerabilities mitigated.

Base container images with outdated libraries with any vulnerabilities are remediated by replacing them with newer secure versions.

Open-source libraries with license and security issues evaluated using automated tools.

Applications using open-source libraries having license and security issues were identified early in the development cycle and, thus, mitigated early.

Included DevSecOps practices and security training from the initial stage to improve quality and overall productivity.

It improved team members' awareness of secure coding practices as a process, and security issues were identified & mitigated from the start.

Quality gates configured in builds in collaboration with code analysis tools.

There was an improvement in the secure code quality, with no major issues related to security reported. A high percentage of code coverage was achieved consistently.

Automation of security checks for quicker movement of features to production

Security checks were included early in the cycle and as a part of a regular development cycle. Thus, new features pushed to production every quarter were available per sprint as security checks.

08 Conclusion

There is no doubt that DevSecOps is revolutionizing how companies approach security. While the adoption of DevSecOps has increased, the growth of the DevSecOps market size is expected to reach USD 41.66 billion by 2030.

The impact of DevSecOps has changed the mindset of teams across organizations. This is evident from the key security-related insights from a global survey conducted by GitLab in May 2022. As per the survey, 57% of security professionals felt their organizations either shifted left or are planning for the same, while 47% of respondents pointed to DevSecOps or DevOps as their methodology of choice.

The survey also mentions that there has been a significant increase in secure coding practices as compared to the previous year. About 53% of developers ran static application security testing (SAST) scans, 55% executed Dynamic Security testing scans (DAST), and 60% ran scans on containers.

Among all the changes in mindset, the most important has been the change of ownership of security among teams. With DevSecOps practices, application security is no longer considered the domain of security professionals. It is now a shared responsibility of all team members. This single change of mindset has been the biggest success story of DevSecOps.

Our maturity model is a tool that would further help accelerate the shift for the better. It would lead to countless benefits, such as early identification, diagnosis, and remediation of security defects, reduced costs and efforts, increased awareness amongst software development teams, and security-aware work culture in the organization.

10 References

- *DevSecOps: 8 Essential Elements for Your DevSecOps Program, Cloud native wiki by Aqua:* https://www.aquasec.com/cloud-native-academy/devsecops/devsecops/#8-Key-Elements-for-Implementing-DevSecOps
- *DevSecOps Tools, Ken Zettler, Atlassian:* https://www.atlassian.com/devops/devops-tools/devsecops-tools
- *2023 Global DevSecOps Report Series:* What's next in DevSecOps, Gitlab, 2023: https://about.gitlab.com/developer-survey/
- *Software Supply Chain and DevOps Security Practices, National Cybersecurity Center of Excellence:* https://www.nccoe.nist.gov/projects/software-supply-chain-and-devops-security-practices
- *NCCoE DevSecOps Project, National Cybersecurity Center of Excellence, Created October 21, 2020, Updated May 09, 2023:* https://csrc.nist.gov/Projects/devsecops
- *30+ DevSecOps Statistics You Should Know in 2023, Jeff Smith, Strongdm, February 13, 2023:* https://www.strongdm.com/blog/devsecops-statistics
- *20 Capabilities for DevSecOps success, GitLab's DevSecOps Maturity Assessment:* https://about.gitlab.com/resources/devsecops-methodology-assessment/
- *DevSecOps Maturity Model, CheckPoint:* https://www.checkpoint.com/cyber-hub/cloud-security/devsecops/devsecops-maturity-model/
- *Gartner® Report: "How to Select DevSecOps Tools for Secure Software Delivery," Manjunath Bhat, Gartner, February 24, 2023:* https://start.paloaltonetworks.com/gartner-devsecops-tools-for-secure-software-delivery

11 About the Author

Vamsikrishna Nyayapathi,

Senior Principal Architect,
Global Technology Office

Vamsi is a seasoned architect with over 19 years of experience in software development and is based out of Hyderabad, India. At LTIMindtree, Vamsi is a Senior Principal Architect and is part of the Global Technology Office. Vamsi has rich and diverse industry domain experience, including Travel, Tourism, Hospitality, Healthcare, Hitech, Supply Chain, Manufacturing, Banking & Finance. He has led large and challenging digital transformation programs for major global organizations across these different industry verticals as the lead architect, bringing in his expertise in Architecture, Cloud, Security, Infrastructure, Automation & Blockchain and thereby helping customers to build innovative products that are resilient, secure, sustainable, and cost-effective in solving complex business problems.

LTIMindtree is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 700 clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by 82,000+ talented and entrepreneurial professionals across more than 30 countries, LTIMindtree — a Larsen & Toubro Group company — combines the industry-acclaimed strengths of erstwhile Larsen and Toubro Infotech and Mindtree in solving the most complex business challenges and delivering transformation at scale. **For more information, please visit https://www.ltimindtree.com/**