eBook

Your handbook to developer-driven security

SECURE CODE WARRIOR

securecodewarrior.com

Make Shift Left More than Just a Buzzword

DevOps and security professionals have been talking about shifting left for years, but what does that really mean and why does it matter? The goal is to move security and testing from the last step in the process to the very beginning, helping to reduce the number of bugs introduced into the code base and mitigating rework. Today, most developers know they are shipping code with bugs but they rarely can find them, leaving an unsustainable amount of work for AppSec.

Developers often find 24%[1] or less of the available bugs in their existing code - leaving a huge portion of the work of scanning the code for security issues for AppSec to find. Shifting left has been consistently cited as the key to shipping code successfully and without significant delays. But why does it seem to be so difficult to do in practice?

Secure Code Warrior helps you ship code successfully and securely, without significant delays, starting at the beginning of the Software Development Lifecycle.

Major breaches and vulnerabilities in the news show us that data breaches are costly to the business, time-consuming for developers to fix, and distract from more productive development. This reactive approach to security is ignoring a missing opportunity to shift security from a last check to a first priority, by empowering your developer teams to upskill and increase their security knowledge.

Developers often find

24% or less

of the available bugs in their existing code - leaving a huge portion of the work of scanning the code for security issues for AppSec to find.

The Challenge Facing Developer Teams Today

60%[2] of developers are reporting that they are shipping code faster than ever before. This may seem like a good thing, but it introduces a new risk because security is often deprioritized in order to meet tight deadlines and market demand. This pressure doesn't give developers enough time to code securely, and teams are commonly emphasizing functionality over security. However, this pressure to constantly deliver actually sets teams back because of the lost productivity associated with the rework and fixes to the code due to mistakes and vulnerabilities that are not caught by developers.

Companies are sacrificing security for speed, writing secure code is difficult to learn and skills take time to build, and developers neither have the time nor the knowledge to address common vulnerabilities. Other challenges included not having a detailed plan about how to write secure code at their organization, a lack of interest from management and not having the skills needed to properly implement secure code.

Vulnerability Reduction Requires a Holistic Approach

Today developers and AppSec are often in silos, rife with tension. There's reason for optimism though as we're seeing that DevOps is gaining popularity. In 2022, a majority of developer teams said DevOps or DevSecOps was their methodology of choice, and it's no surprise why. DevSecOps integrates security at every stage of the software development lifecycle to deliver better and more secure applications. Security and Development teams continue to work in silos and have tension, but it's clear that this needs to change to help businesses succeed.

DevOps is part of how organizations are trying to break down barriers and reshape culture. The fundamental goal of DevSecOps is to increase collaboration between AppSec/ Security with developers from the very beginning of the software development lifecycle.

DevSecOps integrates security at every stage of the software development lifecycle to deliver better and more secure applications.

But why is DevOps gaining steam in popularity? First of all, it facilitates better code quality by increasing the output of developer teams through consistent collaboration with security teams. This results in better overall quality and security in the code itself, a faster time to market, and better communication and collaboration between teams. Happier developers and happier security and ops teams lead to ever more productive outputs - so it's a win-win from both perspectives.

However, running a successful DevSecOps program must start with a culture shift. In the past, security teams and developer teams worked separately. Organizations that successfully move to a DevSecOps methodology have made application security an integrated strategy by engaging both developer teams and security. Success is not simply achieved by introducing a new scanning tool or doing an annual training, but by breaking down silos and increasing knowledge about security across developer teams and encouraging their active participation in DevSecOps.

The Phases of Security Maturity

Everyone has to start somewhere, and we recognize that there are many components to developer-driven security. It is important to think about where you are on the spectrum of security maturity as you seek to improve your overall posture and build a robust DevSecOps approach.

Building security maturity in developer teams can be approached in stages. Based on Secure Code Warrior's experience with 500+ organizations, we've identified the common practices and traits that align to specific phases of security maturity.

Through our experience and research, we have identified three different phases organizations evolve to when implementing a secure coding culture:

Defining

> ***An organization has identified the need to define and build security maturity.***

When it comes to security, you have to start somewhere. That usually begins with defining your goals and identifying your most common pain points and vulnerabilities. This is all about starting to audit and identify where your organization stands today - what are the strengths and weaknesses, plus what is the plan you are actually going to develop. Common characteristics of this stage are if your organization is lacking an intentional strategy, or perhaps considering security to be a check box at the end of the cycle, or that overall developers lack a general understanding of security knowledge at times when they need it the most.

It's common for organizations in this phase to reuse existing code without properly scanning it, or needing to consistently rework their code as vulnerabilities are exploited or discovered.

An opportunity at this stage is to leverage a company tournament with SCW or a hackathon that will help audit the organization and get the developers curious about secure coding. This will set a baseline of where you are today so that you can plan for where you want to be in the future with the knowledge of where your developers need the most coaching, and who out of your teams can help evangelize security as your security champion.

Adopting

> ***They are beginning to adopt secure coding practices into all stages of the SDLC.***

During the adoption phase, the work has to be done to gain your developers' trust and build their confidence in their secure coding skills. This can be done through an easily accessible and relevant training program. Almost half of all developers have said they want more hands-on training, so it's crucial that you provide the time and investment into an engaging program meant to cover the vulnerabilities and challenges your developers commonly face.

Though meeting compliance and mitigating risk is also a major goal during this phase, it's important to keep in mind that 75%[3] of bugs are missed by developers and are left for AppSec to find. Unfortunately, it's an "if" but a "when" situation when it comes to a vulnerability being exploited, forcing people to improve their security posture. The time to start adopting a broader approach to security is now, when you can be proactive rather than reactive to address a potential security vulnerability or threat.

Building a program that is both fun, engaging, and highly relevant to your developer's challenges, and tailored to your goals around compliance and security will help improve adoption over time.

Scaling

> ***They now have implemented a cohesive approach and created a security forward culture.***

The last and most important phase is how to scale your program from a select group to teams to the majority of your developers in your organization. Broad adoption of a security-first culture can take time, but it doesn't have to be impossible. By shifting your organization's focus to developer enablement and strategic approaches to ongoing education and culture changes - you will see broad adoption and scale over time.

More and more security professionals are joining cross-functional teams and taking part in the daily tasks of software development. Yet despite increased collaboration and knowledge sharing, security teams are still finding bugs that are easily preventable. With broader adoption and scale, you will see increased collaboration between engineering and AppSec, and maybe even the implementation of DevSecOps where security is implemented in all stages.

Rolling out a scalable secure code training program requires support from all levels - from CISO, to AppSec, to your Dev. Managers. Frequently holding tournaments or hackathons is a good way to keep security top of mind, and required secure code training or certifications is a way to ensure widespread adoption from all developer teams.

The solutions are well within reach to get started or to expand your strategy for improving your security posture and mitigate the risks of vulnerable code. This comes in the form of shifting left, which means empowering and enabling your developer teams to own their education and build the skills they need to ship code efficiently and securely.

Developers express that they are willing and able to learn about secure coding, and with roles changing to be more cross-functional, now is the time to invest in a training program that scales to your organization's needs, and helps developers to upskill their secure coding abilities.

This ultimately helps them to save time on exhaustive code reviews, reworks, and firedrills that can be mitigated by empowering and enabling developer teams to be hands-on with their secure code training.

Creating a Culture of Security with Secure Code Warrior

At Secure Code Warrior, we're committed to building the best platform to empower your organization to code securely and prevent threats before they are introduced into your code base. Our learning platform empowers your teams to shift their code left with developer-focused tools and learning modules - improving the learning experiences and outcomes for developers and security teams.

Increase Engagement to Facilitate Active Learning

Increased developer engagement starts with highly relevant content based on real-world security vulnerabilities.

Secure Code Warrior's unique learning approach allows your developers to build a strong foundation of skills with video resources and guided courses that then keep leveling up in difficulty depending on their skillset. Continuously challenge your developers' skills with harder courses, measure and benchmark with assessments, and even have a friendly tournament to determine who is your secure code champion.

Coding Labs enables developers to learn in an environment that simulates the way they work, helping them hone their skills by better engaging with the subject matter without distractions.

In addition, Secure Code Warrior's SCORM integration allows you to manage all developer learning, including secure coding training, in a single platform - your enterprise Learning Management Systems. Secure Code Warrior also offers contextual tools and integrations with Jira, GitLab, and GitHub enabling just-in-time remediation and deeper learning.

By training in a familiar environment, it's easier than ever for developers to go from learning new skills to applying them to actual code and preventing vulnerabilities before they're introduced.

Coding Labs enables developers to learn in an environment that simulates the way they work, helping them hone their skills by better engaging with the subject matter without distractions.

Key Actions

Keep your program job-relevant and timely - developers will appreciate training that feels tailored to their daily challenges and easily accessible through your LMS.

Enable just-in-time remediation and learning by integrating SCW with GitLab, Jira, and GitHub.

Choose your content- with over 63 different languages and frameworks, you can create a program with the content covering your organization's unique needs and common vulnerabilities.

Simulate a real and familiar environment - contextual learning in an environment that mimics an in-browser IDE will increase the stickiness of what developers' are learning and practicing.

Measure, Set Goals, and Benchmark for the Future

To design the right education program and goals, start with a baseline of your existing team. You can do this with assessments that help you to understand areas of strength and areas for improvement by testing your developers' knowledge around specific vulnerabilities, common threats, and in the language or framework they most commonly use.

Key metrics like number of courses completed and time spent on courses, either at the team or individual level, help you to make strategic decisions as to how to build richer training programs. However, it's important to not just measure the training but to measure the impact. Measure the number of weaknesses that get picked up in the development life-cycle through code analysis, bug-bounties or classic vulnerability testing before you start the program in each team. One of the simplest ways to know your training program is having the desired outcome is by measuring the decrease in vulnerabilities being introduced into your code base overall.

Recommended Metrics to Track

Engagement

How much time are your developers spending on training? Are they completing courses, assessments, and participating in tournaments?

- Completion rates
- Time spent
- Participation
- User activity, signups, logins
- User adoption
- Developer feedback

Vulnerability reduction

Have you noticed a measurable decrease in vulnerabilities during code review? Are you seeing less rework come back from AppSec?

- Correlation between training and decrease in vulnerability during code review
- Amount of rework passed back from AppSec

Skills

Where are the areas of strength? What areas have you identified as needing improvement?

- Compliance rates
- Accuracy
- Language proficiency
- Belting program progress
- Areas of strength & weaknesses

Productivity

How long does it take to remediate an issue? Have you noticed an increase in productivity or velocity with vulnerability reduction?

- Impact on Mean Time Remediation
- Code productivity, where secure code = quality code

The second thing to measure is the time it takes to fix a vulnerability. If it takes a developer a month to fix it, this clearly shows they need some additional training, but if they can fix it in an hour, you know they have mastered those skills.

Key Actions

> Leverage insights and reporting to view individual, team, and departmental performance.

> Benchmarking and assessing your teams will help you verify skills and understand your areas of expertise and areas for improvement.

> Give incentives to become a security-skilled developer - the more you reward and recognize the security champions, the more others will want to follow suit.

> Measure the impact of your training program but how many vulnerabilities are reduced, and how long it takes to remediate an issue.

Nurture your Security Champions to Build A Community of Experts

If your organization employs dozens, hundreds, or maybe even thousands of developers - it can feel like finding a needle in a haystack to identify those who prioritize security and write the most secure code. This is a missed opportunity to nurture your security ambassadors to mentor your other developers to code more securely.

Many organizations are familiar with Hackathons - which help teams collaborate and compete to improve upon or build a new application. Why not do something similar for security? Holding regular tournaments creates a fun, competitive environment that emphasizes continuous learning about real-world vulnerabilities and aid in building a strong, security-minded culture.

Tournaments are gamified, so developers can bring their competitive spirit and vie for the prize of becoming the top security champion. This helps you to recognize those who are your best advocates for security, and also institute bite-sized and interactive learning to help your teams take the time to train throughout the year, not just during compliance cycles.

Key Actions

Test your developers' knowledge against their peers - either within your organization or across the globe with tournaments or hackathons.

Find and recognize your security champions to continue evangelizing and instituting secure code across your organization.

Give time to train with bite-sized, interactive learning.

Make it fun and consistent! Secure Code Warrior makes it easy to stay on top of training throughout the year, not just during annual compliance certification.

Success Story: Colgate-Palmolive

Colgate-Palmolive, just like nearly every other organization, is going through a digital transformation to better serve its customers. The ever-increasing online and direct-to-consumer presence has led to a shift in what it means for the organization to embrace digital security.

"Like many in the industry, Colgate-Palmolive has been switching to a more digital presence and is becoming a more digital-first company," says Alex Schuchman, CISO at Colgate-Palmolive. "Working on the build side of applications has really been helpful when I made the switch over to my role as CISO. I understand the pain of getting tickets back from AppSec or the frustration of missing deadlines because of re-work. As a result, my goal as CISO hasn't just been increasing security in the software development lifecycle, but also streamlining how it is implemented."

It is very important to us that we are protecting our customer's data and therefore are able to build trust - not just in our products but in the digital interactions our customers have with us.

– Alex Schuchman, *CISO at Colgate-Palmolive*

Colgate-Palmolive approached this challenge by breaking up their security training into smaller, bite-sized chunks. This allowed their developers to remain motivated and increase their maturity incrementally.

"I wanted to roll out these best practices while keeping the developers engaged," says Alex. "We still have mandated critical parts of the program but keeping the training manageable and listening to the developer's feedback has helped the program be successful."

In addition to making the program manageable, by leveraging Secure Code Warrior's gamified and contextual approach to training, the developers were able to have fun while learning. This led to higher engagement, retention, and application of what was learned when the time came to write code.

According to Alex, "We understood that to optimize for success we needed to have our developers on board from the start. So we made sure the developers knew they would be a critical part of the success of the program. As a result, we found that there was a much better relationship between our security team and our developers, and it really felt like we were working together as a team on the program. We are continuing to expand and scale the security maturity program, building on the success we have already enjoyed."

As we can see from this example, a successful security program can be achieved by having a clearly defined program and emphasizing developer input and engagement. This partnership with Secure Code Warrior enabled CISOs like Alex Schuchman to create a successful security maturity program at Colgate-Palmolive.

Experience greater ROI with Secure Code Warrior's Strategic Approach to Developer-Driven Security

Creating a culture change isn't easy, but Secure Code warrior helps you to identify your security champions and help equip developers and organizations with the right skills to tackle today's ever-changing security challenges. Implementing an engaging and scalable secure code training program is a worthy investment because of the long-term preventative approach to security, instead of the reactive way of the past. This ultimately helps to mitigate costly risks of a breach, educate developers how to find and fix vulnerabilities quickly, and facilitate better collaboration with AppSec in order to focus on product development and accelerated time to market.

More and more breaches and vulnerabilities are being exploited each year, therefore it's important to start proactively building a holistic approach to security - starting first with your code. This will help to:

> Reduce complexity by investing in your most valuable resources - your people, instead of throwing money at tools that only solve part of the problem.

> Improve efficiency and overall effectiveness by limiting rework or fixes that would normally be identified by AppSec after the code is deployed.

> Mitigate risk and achieve compliance, avoiding costly fines, the loss of customer trust, or worse- the cost of a data breach.

Empowering and enabling your developers helps them to catch security weaknesses earlier in the development process before they become expensive- or worse leaving vulnerabilities that can be exploited later. Secure Code Warrior helps shift your security culture left by starting at the source - your developers.

Curious to learn more?

Watch our ProductTalk webinar to learn about all the new exciting features in the SCW platform

Learn about Colgate-Palmolive's approach to building developer security maturity, the challenges faced and key learnings along the way in this on-demand webinar

Want to learn about your company's security posture? Take this quiz to assess your team's security maturity.

Advocate for a shift left with the Importance of Developer-Driven Security in Developer Teams whitepaper

Start planning your program with the Secure Code Training Blueprint

Try Secure Code Warrior for Free

About Secure Code Warrior

Smarter, faster secure coding

Secure Code Warrior builds a culture of security-driven developers by giving them the skills to code securely. Our flagship Learning Platform delivers relevant skills-based pathways, hands-on missions, and contextual tools for developers to rapidly learn, build, and apply their skills to write secure code at speed.

Established in 2015, Secure Code Warrior has become a critical component for over 450 enterprises including leading financial services, retail and global technology companies across the world.

Sources:

[1] *GitLab 2022 Global DevSecOps Survey: Thriving in an insecure world*

[2] *GitLab 2022 Global DevSecOps Survey: Thriving in an insecure world*

[3] *GitLab 2022 Global DevSecOps Survey: Thriving in an insecure world*

SECURE
CODE
WARRIOR

securecodewarrior.com