VERACODE

Good, Better, and Best Practices to Follow When Starting an AppSec Program

When building out an AppSec program, it's ideal to follow best practices. But that's not always possible. Time, budget, culture, expertise, and executive buy-in often restrict organizations from following best practices. However, that doesn't mean that you can't create an impactful AppSec program.

Check out some key AppSec best practices, below, as well as first steps you can take now to position your program for future improvements.

To select an AppSec testing type:

Good:

Begin with the AppSec testing type that makes the most sense

Start by implementing the AppSec testing type that will have the most impact, in the shortest amount of time, for the least amount of money. This will depend on factors like your release cadence and risk tolerance.

Better:

Use more than one AppSec testing type

Now that you have the most impactful AppSec test implemented, begin to incorporate additional testing types as necessary and as budget allows.

Best:

Use multiple AppSec testing types

Each testing type has its own strengths and weaknesses, with no one tool able to do it all. You need to use the different advantages of multiple analysis techniques to achieve comprehensive application security. Ideally, you should be using every testing type.

Capabilities	Static Analysis	Dynamic Analysis	Software Composition Analysis	Interactive Application Security Testing	Manual Penetration Testing
Flaws in Custom Web Apps (CWEs)	✓	✓		✓	✓
Flaws in Custom Non-Web Apps (CWEs)	✓			✓	✓
Flaws in Custom Mobile Apps (CWEs)	✓			✓	✓
Known Vulnerabilities in Open Source Components (CVEs)			✓		✓
Behavioral Issues (CWEs)	✓			✓	✓
Configuration Errors (CWEs)		✓			✓
DOM-Based Cross-Site Scripting	✓	✓			✓
Business Logic Flaws (CWEs)					✓
Coverage of Full Application	✓	✓	✓		✓
Repeatable Process for Automation	✓	✓	✓	✓	
Scalable to All Corporate Applications	✓	✓	✓	✓	
Scan Speed	Seconds to Hours	Hours	Seconds to Minutes	Seconds to Minutes	Days to Weeks
Cost	$$	$	$	$$$	$$$$

To remediate security flaws quickly:

Good:

Prioritize security fixes

Prioritize which flaws you should fix first by considering defect severity, the criticality of the application, and how easy it would be to exploit the flaw.

Better:

Prioritize security fixes while creating fewer vulnerabilities

Aside from prioritizing which flaws you should fix first, you should begin to work on introducing fewer flaws into your code by scanning frequently.

According to our State of Software Security v11 report, scanning frequently can reduce the time it takes to remediate 50 percent of security flaws by 22.5 days.

Best:

Fix everything fast

You need to have a solid plan in place to fix vulnerabilities once they are discovered. Automating security testing in CI/CD pipelines allows organizations to not only find flaws faster, but it also speeds up the remediation process.

To create security champions:

Good:

Introduce the concept of a security champions program

Start by making sure your organization's security, development, and leadership teams are all on board with the security champions concept and are willing to invest the time, money, and resources needed to make security champions successful.

Better:

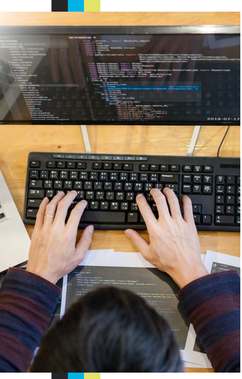
Get developers interested in the security champions program

Security needs to build relationships with development teams so they buy into the idea of security champions. By making it easy for developers to learn about security, it will help them get invested and want to be a part of the program.

Best:

Embed security champions on each development team

Select interested volunteers from each development team to become security champions. Give them the tools and knowledge they need to be the security experts on their teams. Nurture your champions with special programs, goals, and incentives.



To shift security left:

Good:

Help developers and security professionals build a relationship

Moving security testing into the development cycle means that developers will play a bigger security role. Since most development and security teams have never worked together, "shifting security left" can be a significant cultural change. Before making this change, help security and development understand each other's roles and build a relationship.

Better:

Shift security knowledge left

Security needs to be everyone's responsibility – not just the security professionals'. The message needs to come from the top down. For developers to adopt a security mindset, they need security training (a skill that is often not taught in IT courses).

In a Veracode sponsored survey, 76 percent of developers responded that reported that they weren't required to complete any security courses while in school.

Best:

Shift security left

To keep up with rapid software releases, security testing needs to be *woven into* the development cycle instead of *after* the development cycle. That way, when it is time to release the product, security testing will not stand in the way.

To measure your AppSec program:

Good:

Focus on your policy metric

Start measuring the success of your AppSec program by focusing on one metric: how your AppSec program is complying with your internal AppSec policy. Make sure that the AppSec policy you start out with is attainable and easy for developers to understand.

Better:

Introduce additional metrics

Add other metrics over time, such as time to resolution and flaw prevalence. You can also make your AppSec policy a bit more stringent.

Best:

Use multiple metrics to measure your AppSec results

Identify which metrics are most important to your organization's key decision-makers, then display the metrics in an easy-to-understand, actionable manner. This helps ensure continued progress and buy-in.

For additional tips on starting or maturing your AppSec program, check out our guide, *Application Security Best Practices vs. Practicalities: What to Strive for and Where to Start.*

Download Guide

VERACODE

northdoor
Store IT · Protect IT · Use IT