Application Security

BEST PRACTICES HANDBOOK

FOR VERACODE CUSTOMERS

VERACODE

As a Veracode customer, you're already ahead of the application security curve. But no matter what stage of AppSec maturity your organization is at, your program may still have room for improvement.

We've been working with customers like you for more than 10 years to help build out AppSec programs big and small. In the process, we've learned a lot about what works — and what doesn't — when it comes to effective application security. To help you take your program to the next level, we've put together this guide of AppSec best practices.

Shift Left for Security Success

With the speed of software development today, your organization doesn't have the time to backload security or wait until the very end of a project to address security issues. Instead, security should be assessed continuously, which means it needs to be part of the development process from day one. By shifting security left, your teams can embed security into the software development process as they create code, checking for and removing vulnerabilities before they emerge instead of after the fact.

The essential steps for shifting security left include:

- Automating security as much as possible to reduce the amount of human intervention in the process.
- Giving your developers the tools and training they need to test for security on their own early and often.
- Avoiding false alarms by using security testing tools that tune for maximum coverage and low noise.
- Selecting and developing security champions so you have security advocates embedded into your development teams.
- Developing a culture of visibility so that your developers stay involved with your products' application security even after they go into production.

PRO TIP:

Take Advantage of Veracode Integrations

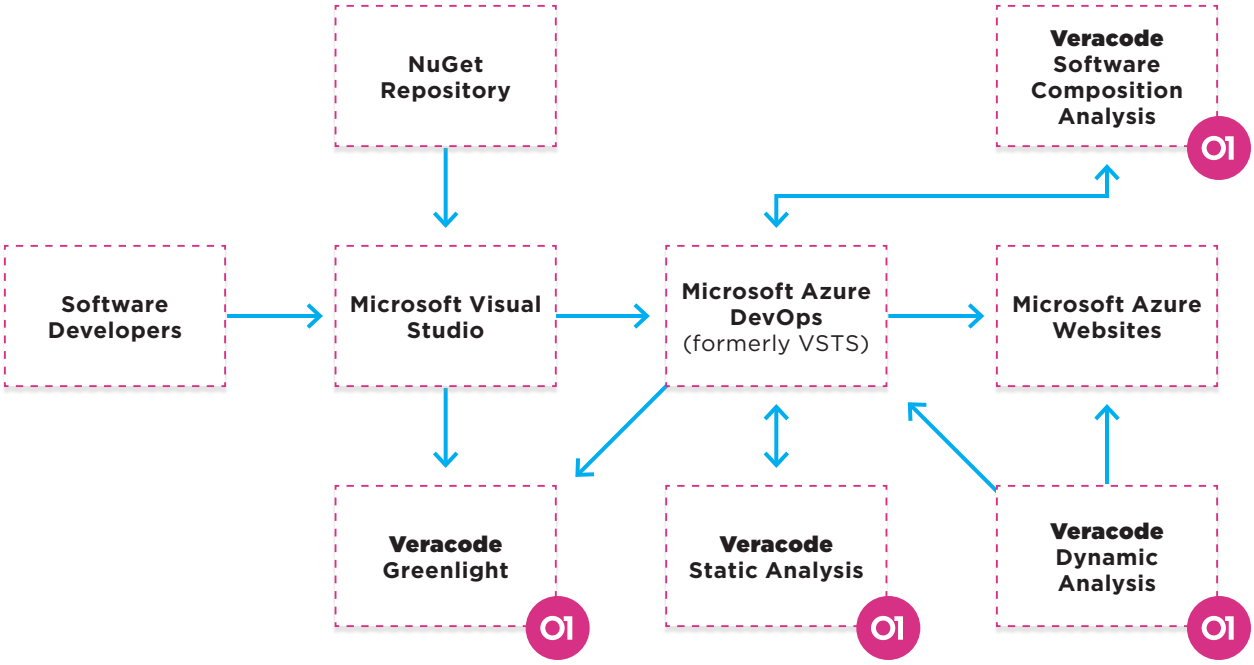
Fix vulnerabilities earlier in the SDLC by integrating with Veracode's plugins, wrappers, and APIs. Installing available plug-ins or leveraging standard Veracode APIs and wrappers can establish seamless, reciprocal data exchanges between our platform and your development teams' IDEs, build systems, bug tracking databases and other systems.

Encourage the development team or security lead to integrate where possible, accessing Veracode tools and instructions for their use via our Help Center. Schedule an integration meeting with support to help with setup and troubleshooting, and have an active development team be the champion for integrations to help drive other teams to integrate moving forward.

Learn more

EMBEDDING APPSEC TESTING INTO DEVELOPMENT

Here's an example of how Veracode tools can help you embed AppSec testing into development so you can shift security left:



→ Veracode Dynamic Analysis

Scan your web apps in production

→ Veracode Greenlight

Security "unit testing," fast and developer focused

→ Veracode Software Composition Analysis

Manage open source software security and license risk in a central location

→ Veracode Static Analysis

Comprehensive application analysis with low false positive rate

Learn more about the 5 Essential Steps to Shift Security Left.

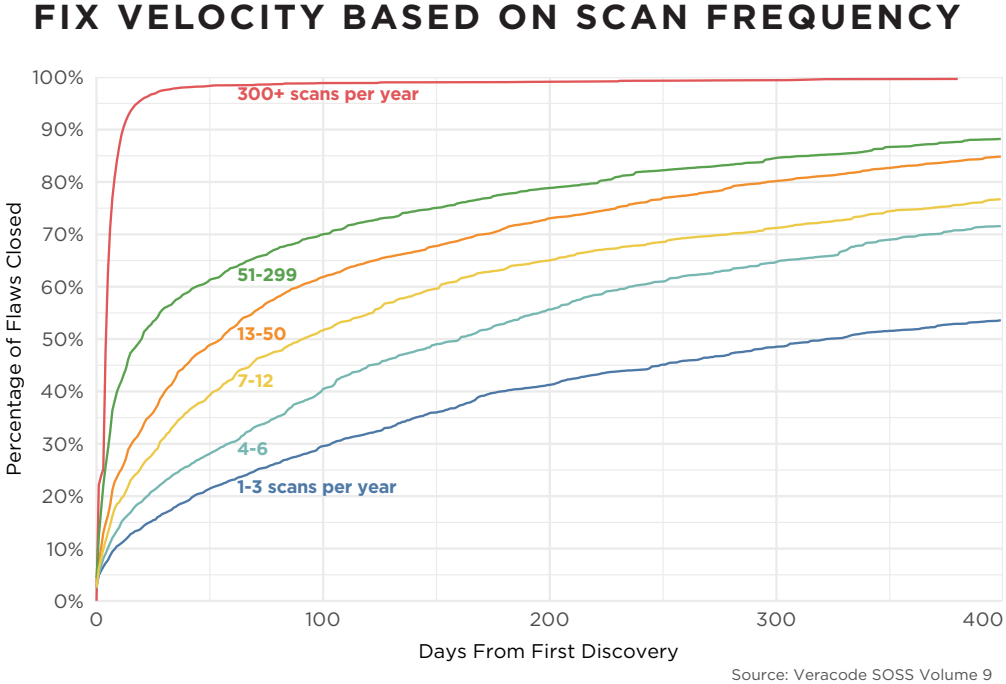
Always Be Scanning

The best time to scan is "right now." According to our most recent State of Software Security report, there's a strong correlation between how many times a year an organization scans and how quickly they address their vulnerabilities. That's because the same incremental processes and automation that security-minded teams put in place to make it easier to scan more frequently also lend themselves to faster remediation.

When creating a scan strategy, it's important to prioritize frequent scans of small builds over one big scan of a large build. This allows your developers to make gradual, continuous improvements to the security of your software when the code is still fresh in their minds and easier to fix.

To increase scanning frequency, work towards assessing security throughout the software lifecycle. For instance:

- **Early development (coding):** Aim for real-time scanning in the IDE.
- **Mid development (packaging):** Aim for scanning in line with the build schedule (using a sandbox).
- **Late development (release):** Aim for scanning integrated with QA and/or release testing.



Find out about the power of scan frequency in the Veracode State of Software Security report.

Which Test Is Best? All of Them

When it comes to AppSec testing types, you can't play favorites. Static, dynamic, manual testing, and software composition analysis all identify different types of vulnerabilities, with each playing a unique role as part of a holistic testing strategy.

According to our research, there are significant differences in the types of vulnerabilities you'll find by conducting different tests. For example, some vulnerabilities are easily found using automation, while complex logic flaws are more easily detected by human intuition. In addition, our research into vulnerabilities found by testing types revealed that two of the top five vulnerability categories we found during dynamic testing weren't among the top five found by static, with one not found by static at all.

A strategy that's overly reliant on just one testing type can leave software vulnerable while providing organizations with a false sense of security. Don't believe claims that any single type of test is better than another; each has its own strengths and weaknesses. It takes a balanced approach to properly evaluate and mitigate risks.

Read more about The Top 6 Application Security Mistakes.

VULNERABILITIES FOUND BY DIFFERENT TESTING METHODS

CAPABILITIES	STATIC ANALYSIS	SOFTWARE COMPOSITION ANALYSIS	DYNAMIC ANALYSIS	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)	○	●	○	● 1
Behavioral Issues (CWEs)	● 2	○	○	●
Configuration Errors (CWEs)	○	○	●	●
Business Logic Flaws (CWEs)	○	○	○	●
Repeatable Process for Automation	●	●	●	○
Scalable to All Corporate Applications	●	●	●	○
Scan Speed	Seconds to Hours	Seconds to Minutes	Hours	Days to Weeks
Cost Per Scan	$	$	$	$ $ $

1. Penetration testing can find known vulnerabilities in open source components, but this may not be as rigorous as Veracode Software Composition Analysis, which not only systematically flags CVEs but also crawls commit histories and bug tracking tickets in open source projects to identify silent fixes of security issues.

2. This is not true for all static analyzers. Veracode can exercise the code and manipulate the UI for behavioral analysis in mobile applications.

WHEN TO USE WHICH ASSESSMENT TYPE

ASSESSMENT TYPE	ADVANTAGES	LIMITATIONS
Static Analysis (with Entire Application in Scope)	• Very broad coverage of flaw types (CWEs) • Looks at the flaws in the context of the entire application, analyzing all the data paths • Can scan any type of application, including web, mobile, desktop, or microservices • Scanning frequency should be in line with how often developers can review scan results • Use static analysis as part of Continuous Delivery pipeline and file security issues in bug tracking system • Can track flaw history: new, open, fixed. Important for trending reports on mean time to remediation. • Suitable for compliance purposes	• Does not provide instant feedback to developers as they're coding • Cannot find CWEs related to server configurations • Limited to code that developers can remediate. Does not report vulnerabilities in third-party components (see: SCA).
Static Analysis (on File Level, e.g. Greenlight)	• Recommended for development teams who want to shift left in application security testing by scanning early and often. Scans usually complete in seconds. • Best suited when scanning multiple times per day • Recommended for use by developers working on the new code for continuous flaw feedback and remediation guidance • Developer friendliness: enhances learning, allows developers to find and address issues without exposing flaws in reports	• Scans individual files, so can only detect vulnerabilities where source and sink are in same file • Typically not suited for compliance scanning because scope limitations may cause false negatives • Does not report vulnerabilities in third-party components
Dynamic Analysis	• Scans web applications without having to integrate with the SDLC • Ability to scan in pre-production and production • Suitable for compliance purposes	• Scan times are often between 12 and 24 hours for complex applications, so recommended for overnight scans, or for asynchronous scanning
Software Composition Analysis	• Finds vulnerabilities in third-party components • Scans take seconds or minutes • Can scan any type of application, including web, mobile, desktop, or microservices • Suitable for compliance purposes	• Does not find flaws in first-party code
Manual Penetration Testing	• Works on all types of applications and languages • Only assessment type to find business logic flaws that require understanding of the application's business functionality and impact • Suitable for compliance purposes	• Slow, human-driven process, typically about five to 10 days per application

PRO TIP:

General Veracode Scanning Recommendations

Policy scans

These should represent the code that's deployed to production. With Veracode Static Analysis, you should scan any code not in production in a developer sandbox. For Veracode Dynamic Analysis, there's no concept of a sandbox, but you can conduct policy scans for the preproduction URL in a different application profile from the production URL to keep them separate. For Veracode Software Composition Analysis, you can scan every commit and simply compare the hashes values of the binaries to the software deployed in production.

Periodic policy scanning

Conduct a policy scan on code that has been deployed into production on a regular basis, regardless of whether there has been a code change. This ensures that you benefit from enhancements to the scanning technology. Assuming you have not touched the code, anything between monthly and annual scanning of the application is adequate, depending on the criticality of the application.

Sandbox scan promotion

This happens in Veracode Static Analysis when the developer sandbox scan is clean, i.e., no policy affecting flaws remaining or unmitigated. It must happen as close to release as possible to satisfy the point above. This could be as part of the release process. If the release gets cancelled, the promoted scan must be deleted. That might mean deleted in Veracode, and in any external reporting tool.

Scanning every commit

For scanning on every commit, we generally recommend using Veracode Greenlight, which provides a faster, light scan of your code. We generally don't recommend scanning every commit with Veracode Static Analysis, Veracode Dynamic Analysis, and Veracode Software Composition Analysis. If you are releasing more than daily, we recommend scanning nightly so scans don't overlap and cause errors in the pipeline due to existing scans running.

Teamwork Is What Makes Security Work

As your AppSec strategy changes the ways that security, development, and operations work together, don't forget that the role of the security professional should similarly evolve.

As developers take more responsibility for day-to-day testing, security's new role should include higher-level strategic work like setting policies, tracking KPIs, and providing security coaching to developers. In addition, security should be responsible for providing developers with the training and support they need to integrate Veracode's scalable tools into their SDLC.

Also, keep in mind that security and developers aren't just roles; they're people, too. As AppSec becomes everyone's responsibility, it's essential to help developers and their security team counterparts develop strong and supportive relationships from the very beginning. Each person on both teams should know who their peer is on the other side and meet with them regularly.

By understanding each other's goals and struggles, both sides will more readily engage in collaboration instead of simply mandating solutions. You can help create a shared sense of accountability between development and security by making security performance something that's measured and reported on regularly, as well as making it one of the annual goals for both teams.

Learn more about the security professional's role in a DevSecOps world.

Never Stop Learning

The only constant in AppSec is change. With new solutions and new vulnerabilities popping up regularly, and with employees of different levels of security expertise leaving or joining the development team, you must constantly analyze the results of your AppSec program to find opportunities for targeted training that address specific issues. Education can provide some of your greatest security ROI: According to our research, eLearning improved developer fix rates by 19 percent while remediation coaching improved fix rates by 88 percent.

Veracode Developer Training provides a combination of training options designed to get your developers up to par:

- **eLearning:** Veracode eLearning offers a library of on-demand, web-based training content so developers can learn on their schedule or whenever they're fixing a particular vulnerability. The material trains developers on secure coding for multiple languages, as well as on proactive techniques, such as threat modeling and secure architecture that can be used in the early stages of the SDLC.

- **Instructor-Led Training:** In addition to our eLearning tools, Veracode Instructor-Led Training gives your developers the chance to learn directly from the same application security consultants who provide remediation coaching for your team so they can learn best practices in a way that's specific to the application.

- **Remediation:** Veracode Remediation provides 1:1 results analysis and guidance so a developer can understand how to address a specific security finding while learning how to avoid similar findings in the future.

Find Your Security Champions

In almost every large organization, developers outnumber their security counterparts. By creating security champions on your development teams, you can ensure that security stays top of mind in your organization even if you're working with limited security resources.

A security champion doesn't have to be a security expert, but he or she should be someone familiar with the product and aware of what the security concerns might be; this could include developers, QA, architects, designers, DevOps, operations, or even product managers. By giving the security champion specialized training in basic security concepts, threat modeling, grooming guidelines, and secure code review, he or she can be armed with the knowledge needed to escalate any potential issues and maximize the effectiveness of the security team.

Learn more about how to turn developers into security champions.

Here are a few things to keep in mind when developing security champions in your organization:

- Being a security champion is a major responsibility, so have security champions self-identify and volunteer for the added work to ensure they'll stay engaged and effective.
- Since the security champion may need to shift some of his or her pre-existing responsibilities to other staff, get both leadership and team lead buy-in to ensure everyone is willing to invest the time, money, and resources to make your security champions effective.
- Don't just appoint security champions and send them on their way. Create expectations for what your security champions should do, and incorporate those tasks into their pre-existing peer review work, when possible, to minimize disruption.
- To evaluate the ROI of your security champion program, make security a KPI that you use to judge success.
- Most developers never receive security training, either in school or on the job. Be sure to provide your security champions with the training they need to review code and teach their fellow team members best practices.
- Keep security collaborative by giving security champions ample opportunity to meet with each other and the security team to discuss specific issues and overall trends.

Right-Size Your Security Requirements

AppSec isn't a one-size-fits-all program. When developing or updating your organization's AppSec policy, make sure that it's customized not only to your team's AppSec maturity level, but also to your specific environment. Keep in mind that those variables may change over time. As technology changes and your team becomes more adept, revise and develop new policies that continue to empower your developers to produce good code quickly.

WHEN RIGHT-SIZING YOUR SECURITY POLICIES, THERE ARE FOUR THINGS YOU SHOULD ALWAYS KEEP IN MIND:

1

Start by implementing achievable policies first, with whatever "achievable" means to your team. If you're just starting out, that could be a policy as simple but important as eliminating high or very high critical flaws. You can then get stricter as developers make security part of their daily routine.

2

Think beyond flaw types disallowed by also defining the types of assessments that should be included, how long developers have to fix flaws, and the frequency and stage requirements (for instance, run static scans daily, dynamic scans weekly, and pen tests quarterly).

3

As teams release code faster and faster, look for ways to take some of the security load off your teams by replacing manual testing with automated testing.

4

Not all apps are mission critical. Create different requirements for different apps depending on the app's importance to your business.

Find out why AppSec policies may need to be revisited as DevOps emerges.

Best Practices Checklist

Each of the following teams in your organization has a role to play in ensuring the security of your applications. Use these checklists to help define how each department can work to keep your code secure:

Development Team Best Practices

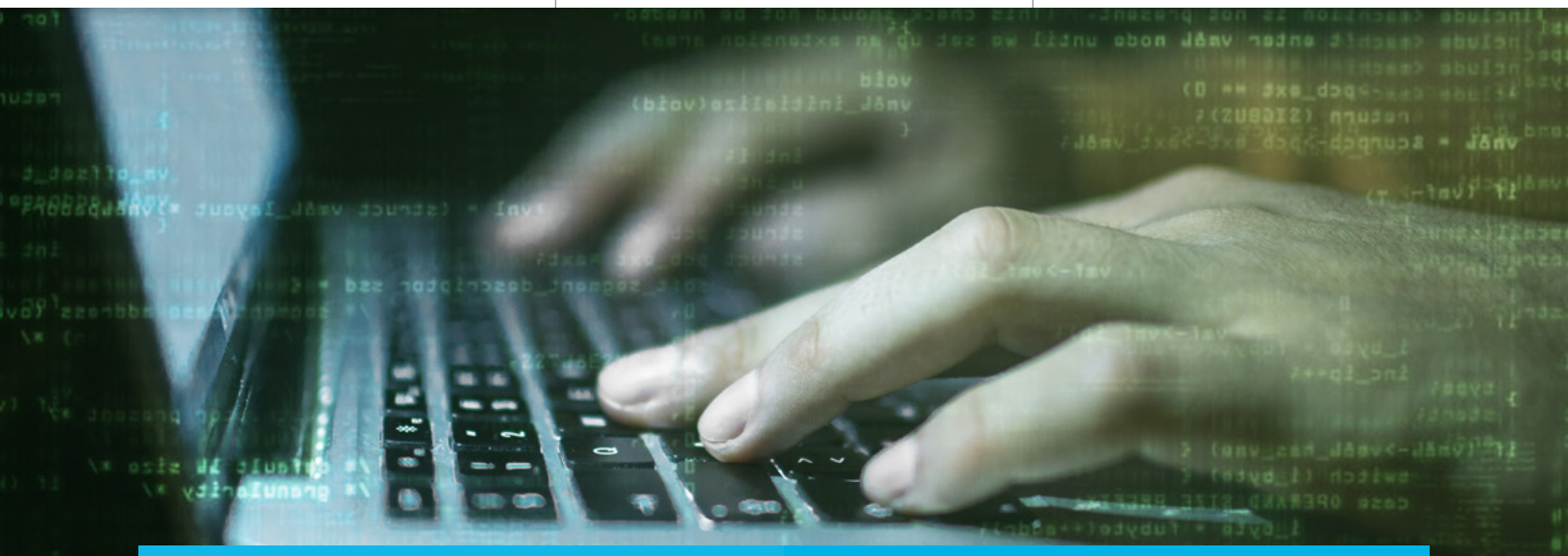
- Incorporate security testing into all stages of the SDLC.
- Be transparent about gaps in secure development.
- Make team members available for troubleshooting during deployment.
- Shift security testing further left.

Security Team Best Practices

- Gather feedback on security policies from all teams to ensure that controls make sense and are reasonable.
- Alleviate all unnecessary pain associated with security controls.
- Communicate early and provide ample time for teams to plan and prepare for new security deadlines.

Operations Team Best Practices

- Implement automation to reduce manual work required by other teams for security testing.
- Prioritize automation that mitigates human error and accelerates workflows.
- Keep systems performing efficiently, network environments secure, and computing resources available.



CHARACTERISTICS OF A WORLD-CLASS APPSEC PROGRAM

A great AppSec program requires more than just scanning. It takes seamless processes and services designed to help developers fix flaws and write more secure code.

Architecture Review in Design

Threat Modeling of Applications

Centralized AppSec Inventory

- **Applications**
  - Client Server
  - Web Apps
  - Mobile
- **Components**
  - Third Party
  - Vendor
  - Open Sourced

Much Broader Scale Than "Business Critical" Apps

- **Internally Developed**
- **Vendor Supplied**
- **Downloaded**

Defense in Depth

- **SAST**
- **SCA**
- **DAST**
- **MPT**

Risk-Based

- **Security sets the policies**

Developer Coaching / e-Learning

Developer Self-Service

Remediate/Mitigate

Integration into the SDLC

VERACODE

Veracode gives companies a comprehensive and accurate view of software security defects so they can create secure software, and ensure the software they are buying or downloading is free of vulnerabilities. As a result, companies using Veracode are free to boldly innovate, explore, discover, and change the world.

With its combination of automation, integrations, process, and speed, Veracode helps companies make security a seamless part of the development process. This allows them to both find and fix security defects so that they can use software to achieve their missions.

Veracode serves more than 2,000 customers worldwide across a wide range of industries. The Veracode Platform has assessed more than 8 trillion lines of code and helped companies fix more than 36 million security flaws.

Learn more at www.veracode.com, on the Veracode blog and on Twitter.