WHITEPAPER

Security Program Stages of Maturity for Mid-Sized Companies

Security Compass
THE SECURITY BY DESIGN COMPANY

Mid-sized organizations face different challenges than their larger peers when building or maturing a software security program. While they have the same regulatory obligations, they have fewer resources. Most lack the ability to form teams with dozens of experts. This means that a security program in a growing company will look different from one in a large enterprise.

That's OK. To be successful, a security program must fit the needs, and capabilities, of the organization. Once you start a program, measure consistently and revisit priorities to make incremental improvements. The goal is to be better each year. Here are some ideas for your journey.

Beginner: Getting the Foundation

Organizations just starting their software security programs need to focus on the fundamentals: benchmarking where you stand today and building an understanding of your goals.

Third-Party Penetration Testing

You need to know where you are before you can map out an improvement plan, so you must understand what types of design flaws and coding errors exist in your software today. You are probably not ready to buy tools and you may not have internal resources that can leverage dynamic analysis tools fully. There are plenty of third parties offering pen testing as a service. You point them at your staging environment, and they will fuzz inputs, look for hidden fields, and find the type of vulnerabilities a skilled hacker would exploit. OWASP's Software Assurance Maturity Model (OpenSAMM) recommends application-specific tests (for business logic) in addition to common vulnerability tests.

Training

Undergraduate-level computer science programs do not teach secure coding in nearly enough depth, if at all. A recent study found out of Business Insider's top 50 ranking for computer science programs, only three require students to complete a cybersecurity class.

Computer-based training can be tailored to your development and deployment environment and cover application security fundamentals – including understanding security requirements in the design phase of development – through testing for security and validating controls.

Providing security awareness training to everyone in the organization, from the newest hire to the executive suite, helps build a culture of security.

Secure Coding Policies

Even new initiatives can include secure coding policies to complement their training. Not every developer will remember the rules in every sprint. But solutions like SD Elements can help by pushing reminders, code samples, and test plans into the same workflow used for functional requirements.

Start a DevOps Team

Adopting DevOps practices requires a team strategy. Siloed development, security, product, and operations teams are out. Integrated teams and decision making are in. In other words, adopting DevOps means a significant cultural change.

Start small – don't try to convert the entire organization at once. Set goals, including release velocity, security gates, and communication strategy. Bring in the executive team and ensure you have buy-in.

Moving to the Next Step

- **You've started the journey and now have a software security program!** By itself, that doesn't eliminate weaknesses and vulnerabilities. In fact, you will probably uncover more issues simply because you're testing more diligently, resulting in technical debt. Start to look at the types of issues you find to identify trends. Begin to measure your Mean Time to Fix (MTTF) issues. Quantify the cost of remediation by combining MTTF with the fully-loaded cost of the employees working on it.

- **Training is a process, not an event.** If you see trends in the issues you found, make sure your training addresses them. For example, it is not unusual to see SQL Injection and Cross-Site Scripting vulnerabilities dominate the results of your penetration testing. That is a sign to increase training on input validation.

- **Make plans to improve.** Much of the beginner phase is about measuring and benchmarking. In the next phase you want to think about process improvement.

Intermediate: Growing Up

The training wheels are coming off. Your team has worked on the foundational pieces for several months to a year, and it is time to add some activities to your process. In particular, the intermediate stage is the time to begin the shift left and address security earlier in the development process.

Adopt a Security Culture

Build a Security Champions program that recognizes employees' efforts to learn and grow. Security champions can be the eyes and ears of the security team and be part of every project across the organization.

Expand DevOps

You may have started with a single project or two. That's the right way to begin, but you need the perspective of more teams to mature your program. Try to expand to a satellite team in another business unit. Let them build off what you have done, but be open to letting them try new things as well, and gather feedback frequently. Everyone will learn from it.

Code Review

Code reviews can be performed internally, with (peer) reviews, or by outside service providers. If this is new to your organization, it is probably better to bring in some experts. They will review source code to find vulnerabilities or unsafe coding conventions. They can also help with remediation guidance and begin to map out secure coding policies.

Add Testing Tools

Penetration testing is an easy place to start because it does not require much internal security expertise. The downside is two-fold. It can only occur late in the software development life cycle (SDLC), making it costly to remediate findings, and it only tests for a subset of the possible weaknesses in an application. Software Composition Analysis tools are one possibility. These are user friendly and identify open source components in your software that have known vulnerabilities - one of the OWASP Top 10 issues. Static Application Security Testing (SAST) is another option, and it is frequently available as a service from established vendors.

Define Security Requirements

You can reduce the number of issues flagged by security testing tools by anticipating how those weaknesses could enter the application and avoid those coding errors in the first place. One way of doing this is by clearly defining security requirements in the design phase of the SDLC. Try it on a couple of projects. Security requirements focus on things a hacker might try to make an application do, that it should not do. This might include "do not store default passwords in the application" to prevent hackers from exploiting them or "do not allow more than 10 requests per second from the same IP address" to mitigate against denial of service attacks.

Moving to the Next Step

- **You find and fix vulnerabilities.** Too many! Adding tools seems to add more work. To mature the program, you need to be more proactive.
- **Security is still siloed.** DevOps is a good start. You can deliver new features more quickly. It is time to integrate security into the team.
- **Use the information you have collected.** You are tracking vulnerabilities by application and type. You know the MTTF. Pick a couple of large opportunities to focus on.

Advanced: DevSecOps at Scale

You have a few teams and business units practicing secure coding. Development and operations are now working together like a well-oiled machine. It is time to step up your game and roll this out across the organization.

Transition to DevSecOps

Bring security fully into the team, starting at the requirements phase. Help them understand objectives, frameworks, and deployment

environments so they can make recommendations before the application is built. This will save time and resources down the road.

Expand your footprint

Every project needs a DevSecOps perspective, even if it is using a traditional waterfall methodology. Adopt security requirements and secure coding policies across the enterprise.

Scale Threat Modeling to accelerate secure development.

Threat modeling identifies potential weaknesses based on the technology stack and deployment environment, then translates those into actionable controls for implementation. Like security requirements, this allows teams to avoid introducing vulnerabilities to an application and greatly reduces the findings from security testing tools later in the process. Automated threat modeling tools allow organizations to scale the exercise across their entire portfolio – without the need for additional security resources.

How SD Elements Helps

SD Elements is the ultimate shift left approach to threat modeling for DevSecOps. It automates the identification of potential weaknesses based on a brief survey of the application's technology stack, including programming languages, frameworks, components, and deployment environment. It translates those weaknesses into actionable controls – including code samples and test plans – and assigns them to development, security, and operations personnel through the tools they use every day. SD Elements integrates with popular DevOps tools like Jenkin and Microsoft Azure DevOps Pipelines and security testing tools like Veracode, Checkmarx, and Synopsys to validate that all controls are implemented.

As a centralized platform, SD Elements provides a single source of truth for all activity and full, evidentiary quality auditing for all actions. Teams have near real-time reporting on the status of each project with granularity to individual controls. Integrations with issue trackers allows organizations to assign and track each task for completion.

Next Steps

Contact us to discuss your security goals and challenges. We can help you take your first steps on your security journey or provide tips to incrementally mature your existing program.

Security Compass

THE SECURITY BY DESIGN COMPANY

Security Compass, the Security by Design Company, is a leading provider of cybersecurity solutions, enabling organizations to shift left and build secure applications by design, integrated directly with existing DevSecOps tools and workflows. Its developer-centric threat modeling offering, SD Elements, and Application Security Training solutions help organizations release secure and compliant software to market quickly and cost effectively. Security Compass is the trusted solution provider to leading financial and technology organizations, the U.S. Department of Defense, government agencies, and renowned global brands across multiple industries. The company is headquartered in Toronto, with offices in the U.S. and the UK. For more information, please visit www.securitycompass.com.

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS