BLACKDUCK®

How to Build a Rock-Solid Software Security Initiative

(Plus 10 Reasons Why You Need One)

Table of contents

Do you have a software security strategy?

This year you tested 46 web applications, 19 mobile apps, and 20 client-server apps. You purchased a new application security testing tool, and you found 112 vulnerabilities. You're feeling pretty good.

But before you get too excited, ask yourself this: Did you reduce your risk significantly? At all? Did you leave critical vulnerabilities unaddressed? Does your board understand the importance of what you're doing and the impact of what you did?

If you aren't sure of the answers to these questions, you may have a software security testing plan, but you don't have a software security strategy.

If you've invested in application security testing already, then you're on the right track to lowering risk. Now, however, it's time to take it to the next level: Turn your application security activities from a cost center to a competitive advantage for your organization by creating a software security initiative (SSI).

Who is this guide for?

This guide is for you if you've ever

- Relied solely on your instincts to decide where to invest your security budget
- Struggled with a development team over prioritizing and repairing security problems
- Had challenges communicating security requirements and results to executive leadership or other departments
- Found the same security defects again and again—from the same team
- Scrambled to find resources to address capacity issues, changing development schedules, or regulatory changes
- Had last-minute requests to test applications for vulnerabilities that delayed your product launch
- Heard about breaches in the news (e.g., Twitter, Uber, Twilio, DoorDash) and thought, "Could this happen at my company?"
- Been burned by poor security planning
- Had a deal delayed while a client demanded concessions because you didn't have a secure software development life cycle (SDLC) or you didn't know which of your vendors used good secure software practices
- Been in a position to come under scrutiny by the Federal Trade Commission or another regulatory body

Be honest. We aren't recording your answers. If any of these sounds familiar, read on to learn proven steps to building and evolving an SSI that will turn your current security efforts into a structured, strategic, rock-solid program.

But I already do application security testing. Isn't that enough?

In a word, no.

In case studies and white papers, we regularly see application security testing presented as the de facto software security technique—a kind of magic bullet organizations use to show they take security seriously.

Application security testing is a critical and necessary component of every security program. However, penetrate and patch application testing alone is not a security strategy at all. Application security testing is a starting block, not the finish line.

Proactive security saves time and money, but it is not going to be enough. A security program is what you need to put in place to lower your exposure across the board.

—Tyler Shields, Senior Analyst,
Forrester Research, Inc.

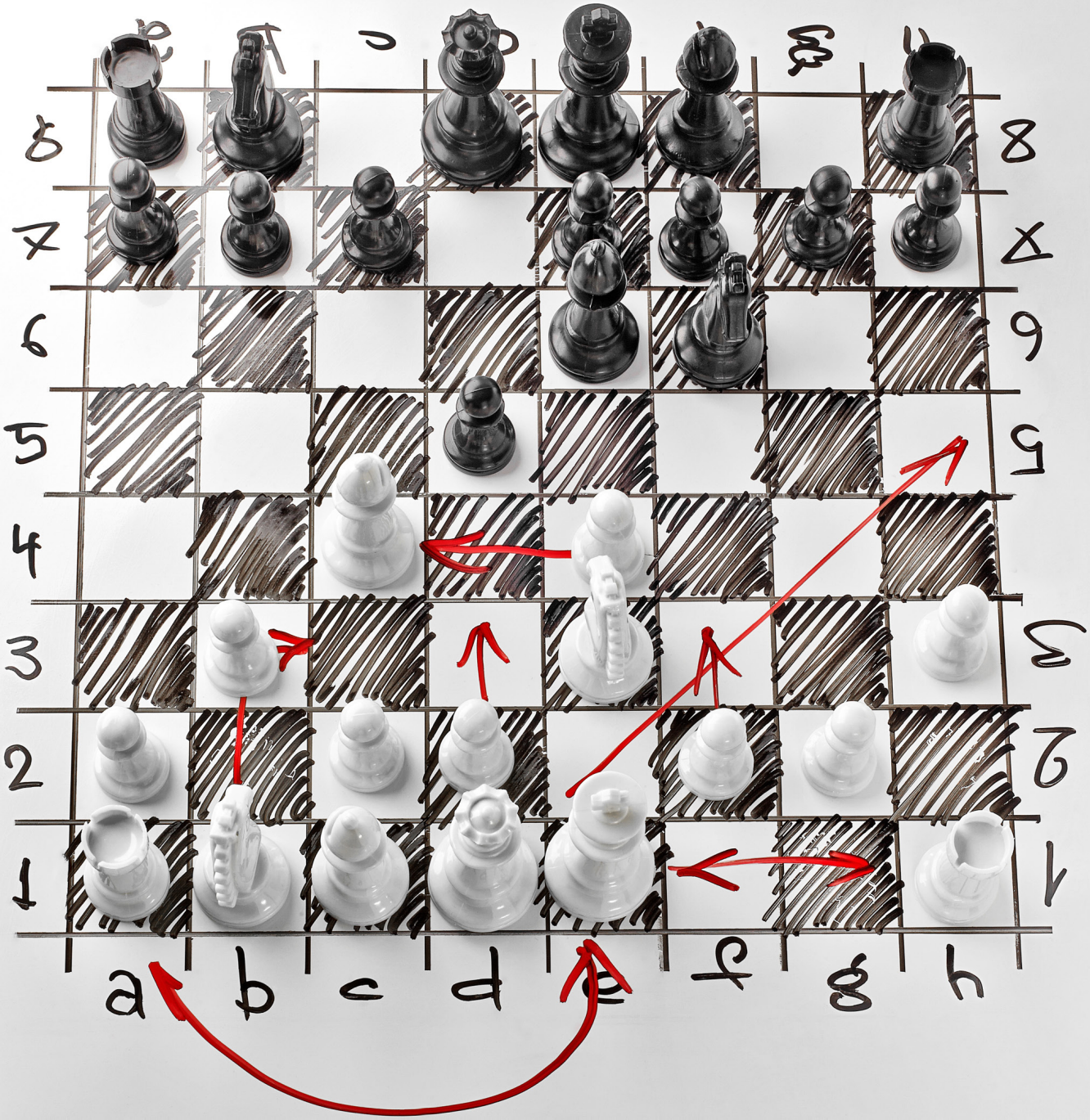
Top 10 reasons you need a software security initiative

Establishing an SSI has many benefits, including these:

1. Ensuring you address unacceptable risk as a priority
2. Providing developers a path to create secure software with minimal disruption, which will improve productivity
3. Giving a specific person or group the responsibility for reducing software security risk so that the hard work actually gets done
4. Building a formal bridge between security and development teams with shared priorities, responsibilities, and incentives to reduce confusion and help everyone work together more efficiently
5. Documenting and harmonizing software security requirements for product managers, architects, developers, testers, and all other stakeholders to create organizational alignment
6. Providing consistent expectations for everyone in your software supply chain, including internal teams and external vendors, so you can trust that software is built securely no matter where it originates
7. Providing a center of excellence for all software security needs—policy, standards, tools, experts, and so on—so that people have a place to get answers and improve their skills
8. Enabling you to measure and communicate success to customers, partners, and the board
9. Ensuring consistent outreach to and training for every stakeholder in the software development chain, strengthening a culture that prioritizes security
10. Meeting the changing needs of development teams while managing risk

Your blueprint for a rock-solid software security initiative

The most effective SSI is fine-tuned to fit your organization and built to scale around your staff, processes, and software portfolio. It helps you show your work by providing a clear and understandable methodology for reducing risk and explaining how you've made investment decisions.



We believe the best way to set a solid foundation for an SSI (or revive a moldy one) is a five-pronged approach.

Build **Measure** **Verify** **Improve** **Manage**

BUILD

To set the right foundation for your SSI, you're going to need a few things: some key pieces of information to set priorities, a governance structure, and training and tools that build security into development cycles.

Let's explore each of these in more detail.

5 things every security leader ought to know

Before you start setting priorities for application security activities, you need to understand the full scope of your challenge. Ask yourself these questions:

- What development projects are in progress, and what are their deadlines?
- Which teams are touching which applications?
- What code is developed in-house, and which software is commercial off-the-shelf or third party?
- Where is your greatest technical risk?
- What is in your application inventory, and which applications have the greatest impact on your business?

Go forth and find out. It's fine if you don't have all the answers to these questions right away. According to a recent SANS Institute study, more than a quarter of respondents didn't know how many applications their organization used or managed.

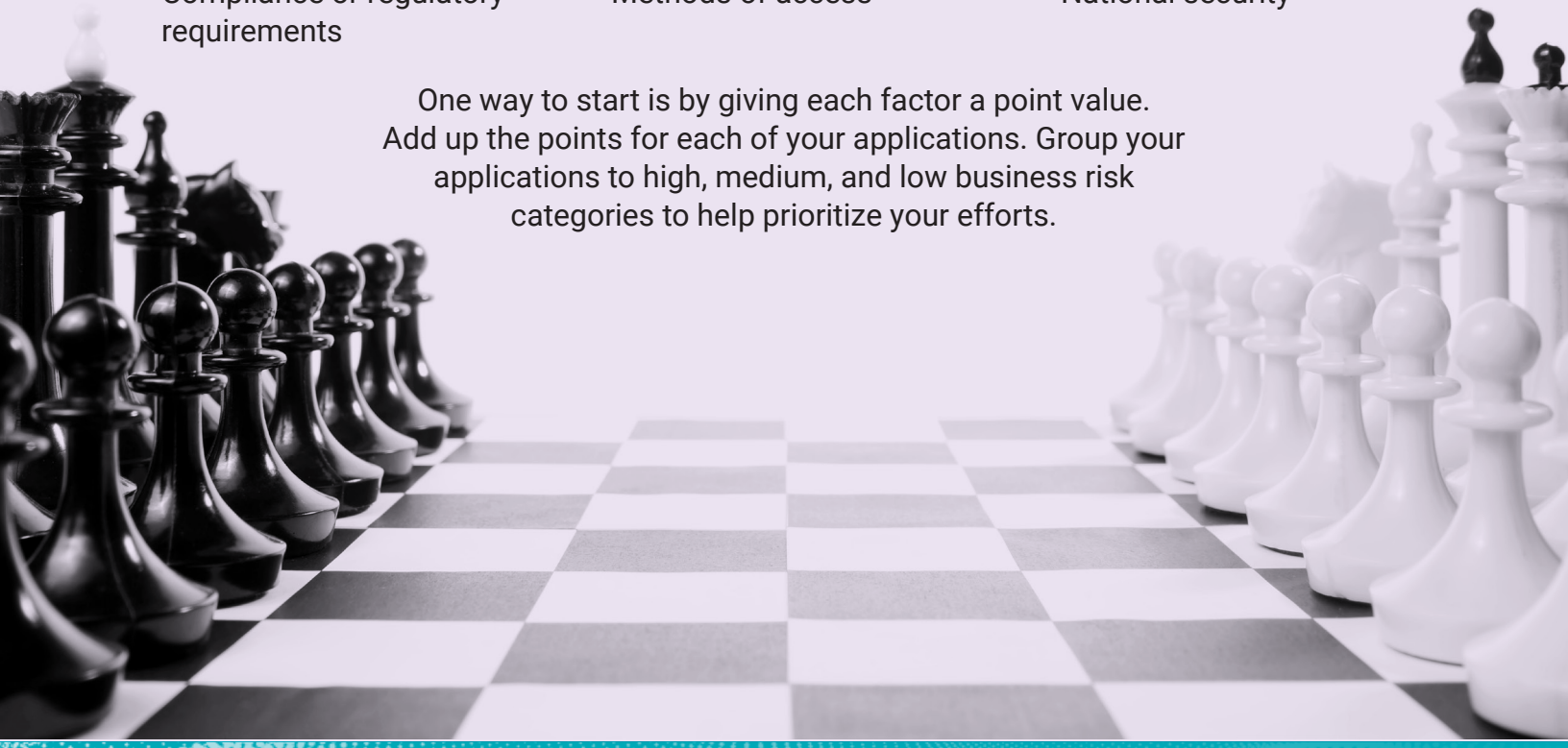
Start with what you know today, and continue to build on your inventory of knowledge.

Risk = Likelihood x Impact

Factors that contribute to an application's business impact include

- Relationship to revenue
- Effect on business continuity
- Compliance or regulatory requirements
- The audience it serves
- How much sensitive data it stores or accesses
- Methods of access
- Connection/integration with other systems
- Human safety
- National security

One way to start is by giving each factor a point value. Add up the points for each of your applications. Group your applications to high, medium, and low business risk categories to help prioritize your efforts.



The secret to a rock-solid SSI

The secret to a successful SSI is governance. Because governance establishes responsibility for security activities and expectations for behavior, it's a necessary step toward creating or right-sizing a foundation for a sustainable SSI.

It makes no difference whether governance is established by a centralized security group through formal policies, explicit standards, and some systematic processes, or by a scrum master through technology and coding standards for a set of application teams. The fact is, someone must be in charge, and there must be some mandatory expectations related to software security. If not, you don't have a secure SDLC.

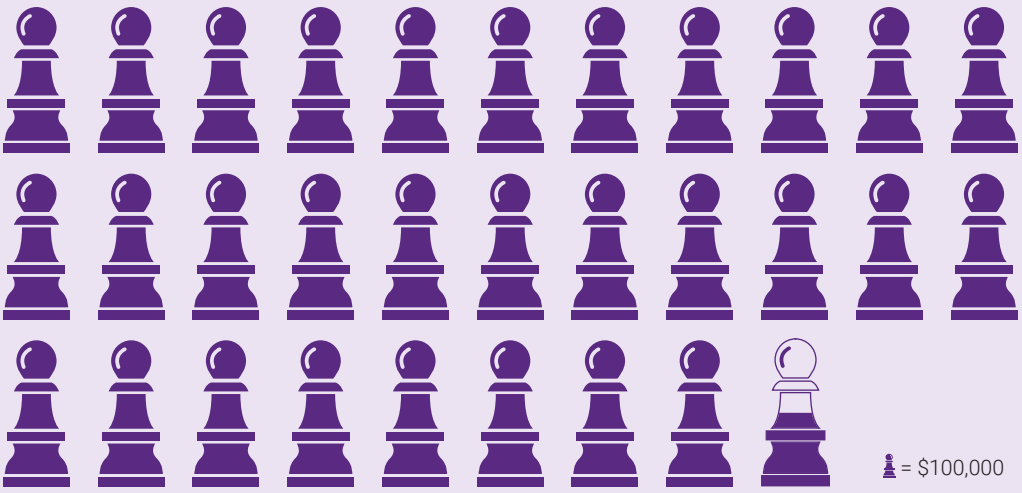
A word of caution: Don't create security policies in a vacuum. Ultimate responsibility may rest with the security head, but application security must be widely discussed with other company leaders and distributed throughout the organization. Most importantly, bring the development team in early and make sure they share ownership over the creation and execution of policies.

Most firms don't have a secure SDLC

That's because, despite many claims to the contrary, a majority of firms don't have software security governance or systematic control over the security posture of their application portfolio.

Don't be one of them. Customers don't like it, insurance companies don't like it, regulatory bodies don't like it, and—coming very soon to firms everywhere—executives and boards of directors don't like it. By way of analogy, look at how a lack of security policy affects things.

74 day shorter breach life cycle associated with deployed security automation



$3.05 million average cost savings associated with fully deployed security automation

5 key security policies

1. Software security. Communicate high-level expectations for getting software security done in your firm and build security into product requirements, implementation, procurement, deployment, and operations. Address the following topics at a minimum:

- **Secure SDLC.** Use is not optional.
- **Application risk ranking.** Give clear guidance on determining which applications are most important to the business.
- **Application design.** Require security controls to be built into your system design.
- **Application development.** Require specific technology stacks and mandatory coding standards. Provide developers clear guidance and pre-built secure-by-design modules.
- **Application testing.** Determine which applications must be tested and which gates they must pass. Set schedules for testing intervals.
- **Software project impact rankings.** Define impact rankings for software projects. Outline how the rankings drive associated assurance efforts.
- **Defect severity and remediation.** Establish rules for setting bug and flaw severities and timelines for fixing coding bugs and design flaws.

2. Network security. Determine protocols and authorization levels that help application security.

3. Data security. Identify and classify your valuable IP and sensitive customer data to help developers apply the correct security features.

4. Physical security. Govern access control and secure your physical infrastructure.

5. Disaster recovery. Determine steps to take in the event of an attack, including reporting, recording, and resolution for attacks against applications.

Ensure harmony between software security policies and other policies.

The trick to creating training that sticks

Everyone involved in the software development life cycle must know how to perform the software security duties associated with their roles: executive management, middle management, product owners, testers, system architects, developers, and everyone else.

Why developers?

Every three years, the Open Web Application Security Project (OWASP) publishes a list of the top 10 web application security vulnerabilities to raise security awareness. Well-known security vulnerabilities such as SQL injection and cross-site scripting have made the list year after year. Yet software developers still code those vulnerabilities into applications today.

An effective SSI must address application security at its core—the point when code is written. The earlier you can remove bugs and flaws from an application build, the less time-consuming and expensive remediation you need in the QA stage. As a result, secure applications can reach the market faster, perhaps even as a competitive advantage.

It's essential that you build incentives into your SSI plan to motivate developers to improve their ability to create secure code—not just deliver features. You can support developers by providing both in-person and online training opportunities. But you must build incentives into performance evaluations and compensation if you want developers to take them seriously and value them as part of their career path.

Tools that put security in the path of development

Security testing techniques such as dynamic analysis and pen testing help security teams consistently identify a broad range of vulnerabilities. But these techniques are used on applications that are already live or in pre-production status, when it's expensive and time-consuming to fix issues.

Look for tools that help you shift left in the secure SDLC. The earlier you can address security tests and fixes, the more cost-effective and productive you'll be.

> We get a 15% gain in productivity because defects are prevented early.
>
> —Jim Routh, Chief Information Security Officer, Aetna

You can help developers create secure code from the start by integrating security tools directly into the workflows and technologies they already use (e.g., integrated development environments). If a new tool requires developers to change their process or takes them away from their preferred systems, you'll have an uphill battle getting them to use it.

MEASURE

Many people believe you can't measure software security. The goal of software security is to prevent successful attacks, and how do you measure the absence of something happening?

Just because you don't yet know that a security breach has happened, that doesn't mean it hasn't. And if it hasn't, that doesn't mean it won't.

Even if you can't prove that you prevented a hacker from penetrating your organization, you can demonstrate the results of your SSI in other ways.

Internal metrics help you make continuous improvement toward business goals. When you set objectives for your SSI, tie them to underlying business goals. This way, when you share results, you'll be able to show how the SSI has fundamentally changed the way your organization operates.

By demonstrating to your board that you're not only improving operational processes but also getting software to market faster and saving money, you'll turn your security program from a series of check-the-box activities to an essential business function.

Absence of evidence is not evidence of absence.

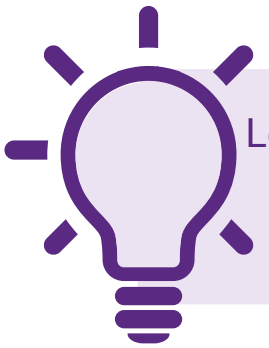
How to speak the language of the C-suite

When you focus on measurements that executive leadership understands and values, you'll be more likely to get continued support for your SSI—or make an argument for more resources.

External metrics allow you to make comparisons to a wider universe of SSIs. In addition to demonstrating internal improvements, you can give your C-suite a broader perspective on your progress by comparing your SSI to that of other organizations. Let's face it: Seeing what others do can be a powerful incentive for company leadership to take security seriously.

The leading industry-wide model for assessing and planning an SSI is Building Software Security In Maturity Model (BSIMM). The BSIMM project benchmarks software security practices used by all types of organizations and proven to enhance software security. It provides a comparison of your program against a data-backed security industry standard.

Consider conducting a BSIMM assessment and joining the BSIMM community. In addition to seeing how you measure up, you'll also have ongoing access to a group of folks who have built SSIs. You can learn from their experiences as your initiative evolves.



Learn more about BSIMM

10 important SSI measurements

Here are 10 measurements that can demonstrate continuous software security improvement (and a bonus measurement):

1. Currency of software inventory, including robust characteristics and risk data for each entry
2. Percentage of all applications that are tested, whether as needed or periodically
3. Number of applications that undergo each type and each level of risk-based testing, from none to lightweight to in-depth
4. Number of variances required due to not meeting software security policy or compliance requirements
5. Time to fix various types of security defects
6. Number of security bugs and design flaws that make it all the way to production
7. Percentage of software projects, whether development or procurement, that go through all the secure SDLC gates
8. Time developers could have spent on activities other than fixing vulnerabilities
9. Frequency of delays, from requirements through production stages, stemming from software security issues
10. Number of applications that meet or exceed compliance requirements
11. Number of software security stakeholders that have the appropriate skill levels for their job

VERIFY

Now that you have policies and a measurement plan in place, you can set up checkpoints to verify whether your teams are performing the activities in your SSI and meeting your SSI requirements, producing the impact you expect.

Think of the verification step this way: Your car needs to pass its safety inspection every year or two, but if you see a check engine light turn on, you'll bring it into the shop sooner and run some tests to find out what's going on.

Defect discovery can be just like the check engine light in your car. It warns you when you need to address a problem with the system ASAP.

Instead of waiting until the end of a development cycle to squeeze in a complex security testing regimen, you can execute small tests along the way. In other words, you can change the philosophy of your testing from "Let's see whether this software is too horrible to release" to "Let's verify that this software turned out as intended."

For organizations using waterfall development, that means adding tests in several phases: requirements, architecture, coding, and QA. For agile shops, that means building security into user stories and making sure developers can find and fix issues seamlessly.

You'll know your SSI is working because your pre-launch security stage will be much, much shorter. You'll no longer you have a mass discharge of security issues that strain your capacity, delay launch, and cause everyone heartache.

The value of an external perspective

If you're running your assessment and remediation work internally, it's good practice to get an external perspective from time to time. An external testing partner can give you an expert opinion to track whether your testing results are accurate and if your underlying system can defend against attacks.

External application testing vendors have the necessary tools and manual testing strategies to catch vulnerabilities your internal tools may miss. They can combine results to confirm suspicions and eliminate false positives. Most importantly, they can interpret results to help your team remediate any issues they find.

WARNING: Don't stop here! If defect discovery is your whole SSI, you'll never get better.

IMPROVE

Setting up an SSI is as easy as 1-2-3

Setting up an SSI is not a one-and-done activity. As you work with your initial SSI structure in the real-life environment of day-to-day pressures, you'll find areas for improvement. Remember that as the tools and techniques available to you improve every day, so do the attackers.

1. **Watch out for patterns.** If the same security issues keep showing up in your verification process, you may need to tune your standards, increase your training, or provide more effective tools to developers. Or if you find that vulnerabilities stem from fundamental design flaws, you'll have to go back to the design team for architecture changes.
2. **Prepare for elastic capacity.** Your SSI must be a living, breathing program that responds to changes in the number and type of applications in your portfolio, organizational shifts, updated compliance requirements, and new attack vectors. At times, demand for application testing inevitably outstrips your internal capacity. Finding the right testing partner can alleviate the burden on your internal team and help you maintain consistent testing and mitigation across all applications.
3. **Build a roadmap for the future.** Imagine a world in which your SSI is firing on all cylinders—with expertise in all software security capabilities, such as these:
    - Risk and compliance
    - Open source management
    - Vendor management
    - Secure architecture design review
    - Application security testing
    - Secure code review
    - Defect management

You don't need to build expertise in every capability at the same rate. But you'll want to make sure you are making steady progress, setting the expectation that your SSI will continue to evolve as a business priority.

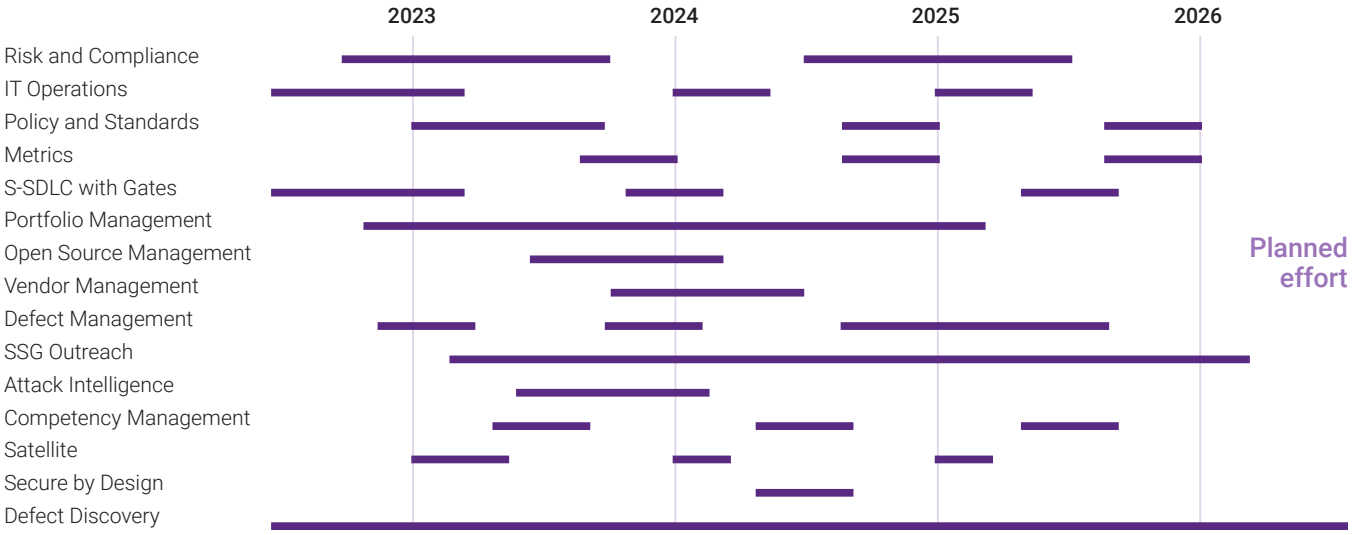
Setting up an SSI is not a one-and-done activity.

You must continuously look for patterns and tune your response.

- Do you need to increase your focus on training?
- Do you need a new coding standard?
- Could your enforcement plan be more effective?
- Should you adapt the tests you're running?

A sample roadmap

We've reached the next and final step of a successful SSI: management.



MANAGE

As you have read so far, an effective SSI has a lot of moving parts and involves many people and departments. As the person at the helm, you must steer the ship so you all stay the course.

To maintain structure and gain insight into each security activity, put in place a robust project management system that is responsive to your goals.

Look for a system that makes it easy to match your security activities to security gates in your development life cycle and your timeline for software launches and upgrades. Compared with a generic project management tool, a security-specific tool will save you time and ensure you don't miss any key elements managed in the SSI.

The right system will make it easy for you to run comparisons across time, application types, business units, and specific projects. At a glance, you'll be able to track your progress, see where you may be lagging behind goals, and identify areas needing additional attention.

Plus, you'll have timely, actionable data to report to company leadership.

A security-specific tool will save you time and ensure you don't miss any key elements managed in the SSI.

Summing it up

Every SSI will reflect its parent organization's structure and culture. Some firms will centralize management, and others will federate. Some will rely on outsourced resources, and others will hire new staff. Some will rely on managed services, and others will grow their own technical teams.

This five-step process will set you on the path to success: You'll have greater alignment across all stakeholders in your development cycle, you'll demonstrate impact against business goals, and you'll have a rock-solid program that you've built for the long term.

SSI cheat sheet

1. Build

- Gather information on your application portfolio, compliance requirements, and areas of technical and business risk.
- Set up a governance structure, including ownership responsibilities and policies backed by leadership.
- Communicate broadly across internal teams and third-party vendors.
- Bring in the right internal and external resources to perform assessments and remediation activities defined in your SSI.
- Structure opportunities for staff to improve security skills and incentivize them to do so.

2. Measure

- Determine measurements that link to underlying business goals and demonstrate continued progress.
- Compare your security practices with a Building Security In Maturity Model (BSIMM) and join the community.

3. Verify

- Build in defect discovery checkpoints throughout the development process, not just at the end.
- Compare your internal results with an external analysis to ensure accuracy and reduce false positives.

4. Improve

- Identify patterns and areas for additional resources, training, and ongoing investment.
- Set up a roadmap to build expertise in software security capabilities.

5. Manage

- Set up a security-specific project management tool to help you manage and steer your SSI.
- Analyze and compare the performance of teams and application types.
- Share progress with executive management and all stakeholders throughout the organization.

About Black Duck

Black Duck® offers the most comprehensive, powerful, and trusted portfolio of application security solutions in the industry. We have an unmatched track record of helping organizations around the world secure their software quickly, integrate security efficiently in their development environments, and safely innovate with new technologies. As the recognized leaders, experts, and innovators in software security, Black Duck has everything you need to build trust in your software. Learn more at www.blackduck.com.