EXECUTIVE SUMMARY

In 2008, application security, research, and analysis experts set out to gather data on the different paths that organizations take to address the challenges of securing software. Their goal was to conduct in-person interviews with organizations that were known to be highly effective in software security initiatives (SSIs), gather details about their efforts, analyze the data, and publish their findings to help others.

The result was the Building Security In Maturity Model (BSIMM), a descriptive model—published as BSIMM1—that provides a baseline of observed activities (i.e., controls) for SSIs to build security into software and software development. Because these initiatives often use different methodologies and different terminology, the BSIMM also creates a common vocabulary everyone can use. In addition, the BSIMM provides a common methodology for starting and improving SSIs of any size and in any vertical market.

Since BSIMM1 in 2009, we've been early reporters on security program changes across people, process, technology, culture, compliance, digital transformation, and much more. Welcome to the BSIMM14 report, and thank you for reading.

WELCOME TO BSIMM14

> If you're in charge of an SSI, understanding the BSIMM and its use by participants will help you plan strategic improvements. If you're running the technical aspects of an initiative, you can use the how-to guide (in Part 4) and activity descriptions (in Part 6) to help define tactical improvements to people, process, technology, and culture.

Each BSIMM annual report is the result of studying real-world SSIs, which many organizations refer to as their application or product security program or as their DevSecOps effort. Each year, a variety of firms in different industry verticals use the BSIMM to create a software security scorecard for their programs that they then use to inform their SSI improvements. Here, we present BSIMM14 as built directly out of the data we observed in 130 firms.

In the rapidly changing software security field, it's important to understand what other organizations are doing in their SSIs. Comparing the efforts of more than 100 companies to your own will directly inform your strategy for improvement and growth.

BSIMM core knowledge is the activities we have directly observed in our participants—the group of firms that use the BSIMM as part of their SSI management. Each participant has their own unique SSI with an emphasis on the building security in activities important to their business objectives, but they collectively use the activities captured here. We organize that core knowledge into a software security framework (SSF), represented in Part 5. The SSF comprises four domains—Governance, Intelligence, SSDL Touchpoints, and Deployment—with those domains currently composed of 126 activities. The Governance domain, for example, includes activities that fall under the organization, management, and measurement efforts of an SSI.

From an executive perspective, you can view BSIMM activities as preventive, detective, corrective, or compensating controls implemented in a software security risk management framework. Positioning the activities as controls allows for easier understanding of the BSIMM's value by governance, risk, compliance, legal, audit, and other executive management groups.

As with any research work, there are some terms that have specific meanings in the BSIMM. The box below shows the most common BSIMM terminology.

BSIMM Terminology

Nomenclature has always been a problem in computer security, and software security is no exception. Several terms used in the BSIMM have particular meaning for us. The following list highlights some of the most important terms used throughout this document:

- **Activity.** Actions or efforts carried out or facilitated by the SSG as part of a practice. Activities are divided into three levels in the BSIMM based on observation rates.
- **Capability.** A set of BSIMM activities spanning one or more practices working together to serve a cohesive security function.
- **Champions.** A group of interested and engaged developers, cloud security engineers, deployment engineers, architects, software managers, testers, or people in similar roles who have an active interest in software security and contribute to the security posture of the organization and its software.
- **Data pool.** The collection of assessment data from the current participants.
- **Domain.** One of the four categories the framework is divided into, i.e., Governance, Intelligence, SSDL Touchpoints, and Deployment.
- **Participants.** The group of firms in the current data pool.
- **Practice.** A grouping of BSIMM activities. The SSF is organized into 12 practices, three in each of four domains.
- **Satellite.** A group of individuals, often called security champions, that is organized and leveraged by an SSG.
- **Secure SDL (SSDL).** Any software lifecycle with integrated software security checkpoints and activities.
- **Software security framework (SSF).** The basic structure underlying the BSIMM, comprising 12 practices divided into four domains.
- **Software security group (SSG).** The internal group charged with carrying out and facilitating software security. The group's name might also have an appropriate organizational focus, such as application security group or product security group.
- **Software security initiative (SSI).** An organization-wide program to instill, measure, manage, and evolve software security activities in a coordinated fashion. Also referred to in some organizations as an application security program, product security program, or perhaps as a DevSecOps program.

BSIMM14 DATA HIGHLIGHTS

Use the information in this section to answer common questions about BSIMM data, such as, "What are some data pool statistics?," "Which activities are most firms doing?," and "How are software security efforts changing over time?"

Note: Items in italic green refer to specific BSIMM activities in Part 6.

Activities are the building blocks of the BSIMM, the smallest units of granularity implemented across organizations to build SSIs. Rather than dictating a set of prescriptive activities, the purpose of the BSIMM is to descriptively observe and quantify the actual activities carried out by various kinds of SSIs across many organizations.

The BSIMM is an observational model that reflects current software security efforts, so we adjust it annually to keep it current. For BSIMM14, we've made the following changes to the model based on what we see in the BSIMM data pool:

- We moved the activities *provide expertise via open collaboration channels, have a research group that develops new attack methods, monitor automated asset creation, identify open source,* and *track software defects found in operations through the fix process* because we now see them more frequently.
- We moved the activities *create technology-specific attack patterns* and *maintain and use a top N possible attacks list* because they're not growing as fast as other common activities in their practice area.
- We added the activity *protect integrity of development toolchains* because we are beginning to see this more.

Unique in the software security industry, the BSIMM project has grown from nine participating companies in 2008 to 130 in 2023, now with approximately 3,600 software security group (SSG) members and 7,500 security champions. The average age of the participants' SSIs is 5.2 years. The BSIMM project shows consistent growth even as participants enter and leave over time—we added 23 firms for BSIMM14 and dropped 23 others whose data hadn't been refreshed.

This 2023 edition of the BSIMM report—BSIMM14—examines anonymized data from the software security activities of 130 organizations across various verticals, including cloud, financial services, financial technology (FinTech), healthcare, independent software vendors (ISVs), insurance, Internet of Things (IoT), and technology organizations.

The 7 Habits of Highly Effective People explores the theory that successful individuals share common qualities in achieving their goals and that these qualities can be identified and applied by others. The same premise can be applied to SSIs. Listed in Table 1 are the 10 most observed activities in the BSIMM14 data pool. The data suggests that if your organization is working on its own SSI, you should consider implementing these activities.

Table 2 shows some activities that have experienced exceptionally high growth over the past 12 months. Not surprisingly, some of these activities, such as *make code review mandatory for all projects* and *identify open source*, are mentioned in the Trends and Insights section. In addition, the *streamline incoming responsible vulnerability disclosure* activity introduced in BSIMM12 has the largest increase in observation count. Note that for some of the activities in Table 2, the growth in observation is a relatively new change. For example, the activity *have a research group that develops new attack methods* saw virtually no growth from BSIMM9-BSIMM12 but had a significant jump in observation rates in BSIMM13, and BSIMM14 has continued that climb. Two years of growth suggests the change is meaningful and the activities are worth considering for your program.

In BSIMM13, we reported new growth after little change over time in the *enforce security checkpoints and track exceptions* activity. This activity has continued to grow in BSIMM14 as firms are able to take advantage of modern automation options in the development pipeline.

In the other direction, in BSIMM13, we reported that the *have SSG lead design review efforts* activity saw continued growth for years but then decreased significantly for BSIMM13. In BSIMM14, this decrease has corrected, with a small growth in observations this year.

BSIMM14 TOP 10 ACTIVITIES	
PERCENT	DESCRIPTION
90.8%	Implement security checkpoints and associated governance.
90.0%	Create or interface with incident response.
87.7%	Identify privacy obligations.
87.7%	Use external penetration testers to find problems.
86.9%	Ensure host and network security basics are in place.
86.2%	Use automated code review tools.
84.6%	Perform edge/boundary value condition testing during QA.
83.1%	Perform security feature review.
79.2%	Unify regulatory pressures.
79.2%	Create a security portal.

TABLE 1. TOP ACTIVITIES BY OBSERVATION PERCENTAGE. The most frequently observed activities in BSIMM14 are likely important to all SSIs.

BSIMM14 TOP 10 ACTIVITIES GROWTH BY COUNT	
INCREASE	DESCRIPTION
15	Streamline incoming responsible vulnerability disclosure.
13	Implement cloud security controls.
12	Make code review mandatory for all projects.
11	Have a research group that develops new attack methods.
11	Define secure deployment parameters and configurations.
11	Use application containers to support security goals.
10	Schedule periodic penetration tests for application coverage.
9	Identify open source.
8	Document a software compliance story.
8	Enforce security checkpoints and track exceptions.

TABLE 2. TOP ACTIVITIES BY RECENT GROWTH IN OBSERVATION COUNT. These activities had the largest growth in BSIMM14, out of 32 firms measured during the last 12 months, which means they are likely important to your program now or will be soon.

TRENDS AND INSIGHTS SUMMARY

These BSIMM trends and insights are a distillation of software security lessons learned across 130 organizations that collectively have 11,100 security professionals helping about 270,000 developers do good security work on about 97,000 applications. Use this information to inform your own strategy for improvement.

Trends describe shifts in SSI behavior that affect activity implementation across multiple areas. Larger in scope than an activity, or even a capability that combines multiple activities within a workflow, we believe these trends show the way organizations are executing groups of activities within their evolving culture. For example, there's a clear trend of firms taking advantage of security automation over manual subject-matter expert (SME)-driven security activities. Over time, we've seen a trend in testing being applied throughout the software lifecycle ("shift everywhere"), followed by trends in additional testing (e.g., composition analysis) and in testing automation (e.g., as checkpoints in the software development lifecycle [SDLC]).

Refer to Part 2: Trends and Insights later in this document for more.

How Software Security Is Changing

Organizations are modernizing development toolchains to give their developers the best tools for building software. Security leaders are taking advantage of the easy-to-use yet powerful automation available in these toolchains to update security testing and touchpoints. This is allowing shift everywhere as a philosophy to move beyond testing to decisions and governance.

When automation makes security tasks easier, trends emerge around automated activities. Modern toolchains, for example, allow for security testing in the QA stage to be automated, much like SAST scans that happen earlier in the development process. This has led to a 10% growth in the *integrate opaque-box security tools into the QA process* and *include security tests in QA automation* activities.

Security teams that embraced the shift everywhere testing philosophy found that their pipelines were able to take scripted actions based on the results of those automated security tests. The automated decisions enabled by these pipelines led to a 60% growth in the *integrate software-defined lifecycle governance* activity in the past year.

Firms are also using automation to better gather and make use of intelligence provided by sensors in the pipeline. Observations of firms that *build a capability to combine AST results* have nearly doubled. Additionally, the use of captured knowledge by the *enforce secure coding standards* activity is again seeing growth after a period of decline.

Finally, some firms are using the insights gleaned from sensors throughout the development lifecycle to proactively prevent vulnerabilities before they become an issue for developers. *Drive feedback from software lifecycle data back to policy* was observed at an increased rate of 36% in the past year, further assisting the engineers who drive the development lifecycle.

What Is Shift Everywhere, Really?

To define shift everywhere, let's start by stating what it's not: it's not trying to do all the security things everywhere in all parts of the software lifecycle (SLC) all the time. Instead, shift everywhere is a philosophy; it's an approach to SLC governance that acknowledges the reality that consistently achieving acceptably secure software is a shared responsibility, and that this responsibility traverses legal, audit, risk, governance, IT, cloud, technology, vendor management, and resilience, among others. Each stakeholder has their own business processes to execute, but each also needs to do their version of security sign-off, which requires understandable and usable telemetry from the SLC toolchain.

Not so very long ago, the only view into the SLC afforded to stakeholders was, "We shipped it yesterday!" That was horrible then and is much worse now, mainly because automation generates telemetry that is easy to route to stakeholders. A shift everywhere approach starts by asking how these roles get the information they need, when they need it, in the processes they normally use, with little or no additional friction, then it bridges that gap, giving each role access to appropriate sensors, whenever they need it, from anywhere in the SLC. Shift everywhere is a philosophy about the security testing and sensors that generate information for all stakeholders in the company, it's not rooted in increasing the security spend or effort. Accordingly, shift everywhere is not adding more security for security's sake, instead, it's ensuring that every security stakeholder can knowledgably make risk management decisions.

Expanding Security's Scope

External pressures like government regulations and increased awareness of supply chain threats are leading organizations to extend risk management to the software that they integrate from outside sources, the toolchains used by their developers, and the software present in their operating environments. We have added the new activity *protect integrity of development toolchains* to begin tracking how firms protect software and artifacts as they pass through their development pipeline.

The first step many firms take in understanding the risk they're bringing into their software by integrating third-party and open source components is scanning with a software composition analysis tool. These moment-in-time checks allow security teams to uncover newly published vulnerabilities in software. After scanning all of the integrated components, teams *create bills of materials for deployed software*, observations of which grew by 22% from BSIMM13 to BSIMM14.

After scanning individual projects and compiling software bills of materials (SBOMs), firms seek to take a more holistic approach to managing open source risk across the portfolio. Two activities associated with this portfolio-wide risk management, *identify open source* and *control open source risk*, both saw just under 10% growth from BSIMM13 to BSIMM14.

Firms are also getting tough on vendors and expecting the software they buy to be secure at the time of acceptance. Observations of the *ensure compatible vendor policies* activity, which reflects how firms enforce security standards on organizations that provide bought and bespoke software, grew by 21% as firms held vendors to similar standards as they use internally.

Who Owns Security

In a trend a decade in the making, we see a growing number of organizations referring to their centralized effort as a product security program (vs. application or software security). We measure this by noting where SSI reporting chains pass through a Chief Product Officer, VP of Products, or Product Security Manager, which now accounts for almost a quarter of the data pool (31 of 130 firms). This naming trend seems to correlate with product vendors creating security programs to manage the risk associated with software that leaves the organization to exist in hostile environments for years to decades (as compared to applications in private data centers).

Initially, product security teams were formed to deal with the unique attack surfaces of their products compared to the web applications in heavy use in financial verticals. Firms continue to deal with unique threats with *create technology-specific attack patterns*, an activity that has grown by 15% since BSIMM13.

Understanding and building technology-specific guidance in the absence of industry best practices for products with unique operating requirements is the first step in securing software that exists in uncontrolled or potentially dangerous locations. To deal with vulnerabilities discovered after software is deployed to external environments, security teams will stand up a Product Security Incident Response Team (PSIRT) function to handle communication about and reactions to reported vulnerabilities. Observations show that the associated *streamline incoming responsible vulnerability disclosure* activity is now present in more than a quarter of the BSIMM14 data pool.

Take stock of your SSI. It's important to periodically look at your program through a different lens.

Important Decisions in Software Security

For such a complicated endeavor, software development and its associated security governance is simple on paper: write some code, then build it, applying all the security testing there was time for. Development fixed the worst security defects discovered, with some of the remainder becoming requirements for the next release. However, actually performing all those steps in the real world can be expensive in terms of hours spent on manual processes. BSIMM data shows some of the decisions made by firms that can help scale security in spite of those expenses.

The oldest insight provided by BSIMM data is that the decision to build and operate a security champions program has a measurable impact on total BSIMM scores. In BSIMM14, firms with security champions scored on average 25% higher than firms without one. Observations of training activities such as *conduct software security awareness training*, *deliver on-demand individual training*, and *include security resources in onboarding* were also positively correlated with the presence of a security champions program.

Joining security champions as an enabler of security capabilities is the organizational decision to target cloud architectures. When we assess firms that *implement cloud security controls*, we also see scoring gains in the Compliance & Policy and Software Environment practices of 21% and 44%, respectively.

While cloud architectures have made certain security activities easier and more affordable for firms, recent economic conditions have caused a reduction in expensive, SME-driven activities that are not easy to automate. Observations of *build attack patterns and abuse cases tied to potential attackers* declined by 25%, *use centralized defect reporting to close the knowledge loop* shrank 18%, and *maintain and use a top N possible attacks list* decreased by 31%.

CALL TO ACTION

Use the information in this section to prioritize improvements in your SSI and perhaps also in the SSIs of your most important software suppliers and partners.

Every SSI has room for improvement, whether it's improving scale, effectiveness, depth, risk management, the framework of deployed activities, resourcing, or anything similar. The following suggestions represent the broad efforts we see happening in the BSIMM participants, with various parts likely right for your program as well.

Plan Your Journey

- Take stock of your SSI. It's important to periodically look at your program through a different lens, and the BSIMM enables that. Use the guidance in Part 4 to create your own SSI scorecard and compare it to your expectations.
- Create a vision and a strategic plan. Use the activity descriptions in Part 6 when creating a prioritized action plan for business areas where your current SSI efforts fall short. Typical investment areas include risk management, digital transformation, technical debt removal, technology insertion, and process improvement.

Get a Handle on What You Have

- Inventory all your code. It's likely that you'll need specialized automation to keep track of all the code you write and all the code you bring in from outside the organization. A simple application inventory will be useful for some things, such as naming risk managers, but you'll quickly need specialized inventories, such as SBOMs, API and microservices lists, various as-code artifacts, code that is subject to specific compliance needs, and much more.
- Automate, automate, automate. Search for ways to eliminate error-prone manual processes and reduce friction between governance and engineering groups, including automating security decisions. This will require some policy-as-code effort and tools integration, and might require bringing development skills into the SSG.
- Gather all the data. As more processes become code and more policy and standards become machine-readable, day-to-day development and operations will generate significantly more telemetry about what's happening and why. Use this data to ensure that everything's working as expected.

Make the Right Investments

- Innovate in digital transformation. Encourage your SSG and other security stakeholders to experiment with ways to deliver security value directly into engineering processes, especially where current security testing tools don't always keep up with engineering changes, such as with serverless architectures, single-page applications, AI, and zero trust.
- Secure the software supply chain. Nearly every organization today uses third-party code and provides code as a third party to other organizations. While producing SBOMs is easy, the management of software, SBOMs, vendors, and vulnerability information is much more complicated.
- Expand software security into adjacencies. Even perfect software can have its security undermined by mistakes elsewhere in the organization. Make explicit ties between the SSI and other security stakeholders working in areas such as container security, orchestration security, cloud security, infrastructure security, and site reliability.

In summary, the data shows that new SSIs—from just started to 18 months old—are typically doing about 33 BSIMM activities. These organizations are also beginning to scale these activities across their software portfolio, deal with all the change going on around them, and evolve their risk management strategy.

Here are some suggestions on reading through this BSIMM report:

- If you're experienced with the BSIMM, or if you need some content to help make your case with executive management, then Part 2: Trends and Insights is probably what you're looking for.
- If this is your first time with the BSIMM, we recommend first reading Part 5 for context and then returning here to decide what to read next.
- If you're starting an SSI or an SSG, or looking to mature an existing program, start with Part 4: Quick Guide to SSI Maturity, then move to Appendix B: How to Build or Upgrade an SSI, and then read through the activities in Part 6.
- If you want to get right into the types of software security controls organizations are using in their SSIs, or if you are working on building out capabilities, then read Part 6: The BSIMM Activities.
- If you want to see a summary of the BSIMM14 data, review Appendix D.
- If you want to look at our analysis of the BSIMM data, review Appendices E though H.