BSIMM13

TRENDS & INSIGHTS REPORT 2022

TABLE OF CONTENTS

WHEN IT COMES TO SOFTWARE SECURITY, NO ONE IS AN ISLAND 3

IF YOU'RE IN CHARGE OF OR BUILDING A SOFTWARE SECURITY PROGRAM 3

WHAT IS THE BUILDING SECURITY IN MATURITY MODEL (BSIMM)? 3

SIGNIFICANT APPSEC TRENDS 5

GENERAL INSIGHTS DERIVED FROM BSIMM13 TREND OBSERVATIONS 6

"SHIFT EVERYWHERE": CONTINUOUS TESTING THROUGHOUT THE SDLC 6

- The Importance of Continuous Defect Discovery 6
- Data Now Driving More Security Decisions 6

SOFTWARE SUPPLY CHAIN RISK MANAGEMENT AND THE RISE OF SBOMS 7

- Managing Open Source Risk through SCA 7
- Struggles with API Security & Visibility 7
- The Rise of Software Bills of Materials 7

INTEGRATING SECURITY INTO DEVELOPER TOOLCHAINS 8

- Moving to Smaller, Automated Checks within the SDLC 8
- Automating and Enforcing Secure Coding Standards 8

EXPANDING SOFTWARE SECURITY BEYOND APPLICATIONS AND PRODUCTS 8

- Capturing Security Knowledge for Knowledge-as-Code 8
- Intelligent Orchestration on the Rise for Containers 8

NO SECURITY WITHOUT A PROGRAM 9

ROLES IN A SOFTWARE SECURITY INITIATIVE 9

- Executive Leadership 9
- Software Security Group Leaders 11
- Key Stakeholders 11

SECURITY CHAMPIONS: THE JIMINY CRICKETS OF SOFTWARE SECURITY 12

- Security Champions Programs Work! 12
- How to Build a Security Champions Program 12

RECOMMENDATIONS 13

ACKNOWLEDGEMENTS 14

WHEN IT COMES TO SOFTWARE SECURITY, NO ONE IS AN ISLAND

Individual or organization, no one is self-sufficient; everyone relies on a community of others. In the world of software security, it can be critical to know what your peers are doing in terms of their own software security programs—what's worked, what's failed—perhaps most importantly, what's changing, and how they're responding to change.

For example, just a few years ago, open source and supply chain security attacks were on the radar of only a handful of security professionals. Today, these subjects are top-of-mind from the board room to development team scrums.

Understanding significant AppSec trends can help you plan strategic improvements to your own security efforts.

As will be seen in this report, many organizations have answered the challenge of software supply chain risk management with software composition analysis (SCA) tools to manage open source risk and mandating software bills of materials (SBOMs) for the code they consume and build. Tomorrow will bring new challenges—perhaps involving API or cryptocurrency attacks, perhaps something so completely new that no jargon has yet been invented to describe the threat. The only certainty is that there will be more challenges, and your software security program needs to be prepared to address them.

IF YOU'RE IN CHARGE OF OR BUILDING A SOFTWARE SECURITY PROGRAM

Comparing other software security groups (SSG) with your own can guide the strategy for your efforts, whether you're in the early stages of implementing a security program or want to ensure your existing program can address changing business and security needs.

If you're in charge of or building a software security program, understanding significant AppSec trends can help you plan strategic improvements to your own security efforts. If you're running a security program from the technical side, you can use the information presented in this report to define tactical improvements for people and processes—by building a security champions program, for example, which is described later in this report.

WHAT IS THE BUILDING SECURITY IN MATURITY MODEL (BSIMM)?

A unique program running for well over a decade, the BSIMM examines the strategies organizations employ to build security into software development. Participants in the BSIMM include members from the cloud, financial services, financial technology (FinTech), independent software vendor (ISV), insurance, Internet of Things (IoT), healthcare, and technology verticals.

The BSIMM13 Trends and Insights report distills the lessons learned from more than 130 BSIMM organizations that collectively have nearly 11,900 security professionals helping over 410,000 developers secure software on 145,000 applications. For those wishing to learn more about the BSIMM project, the "BSIMM13 Foundations" report provides in-depth detail on BSIMM background and data, and can be found at www.bsimm.com/resources.html.

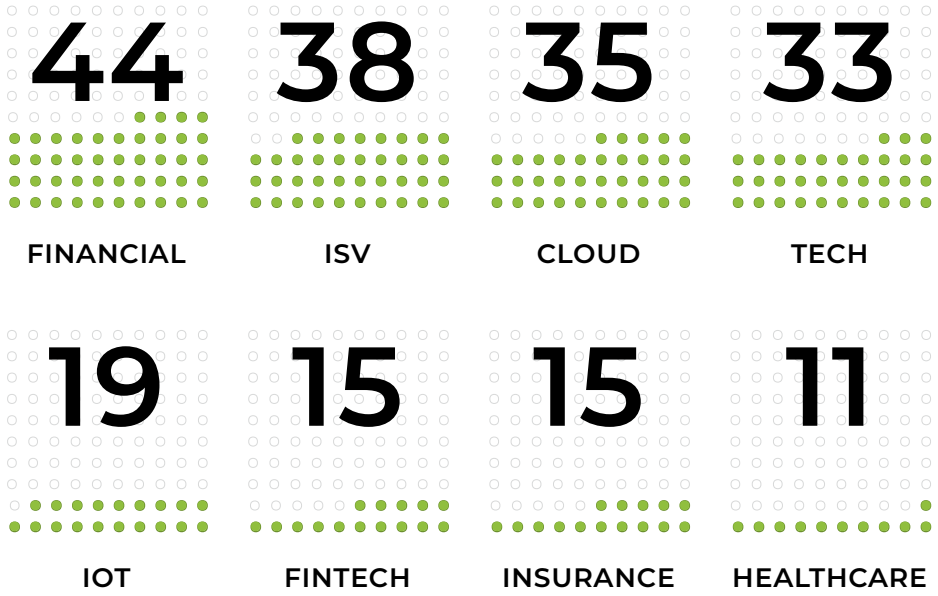
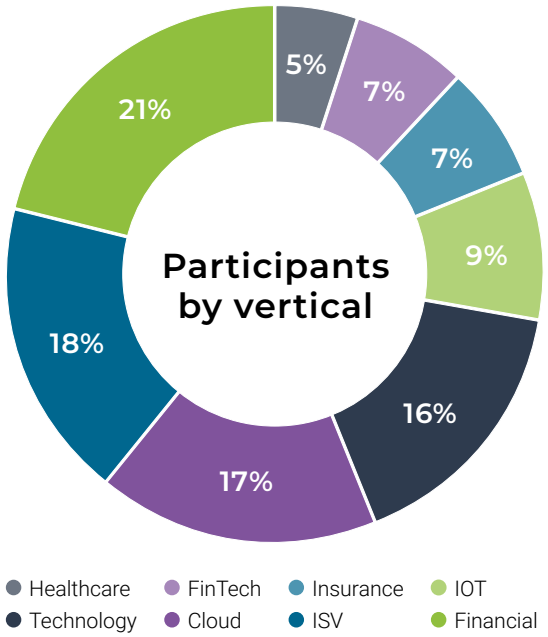
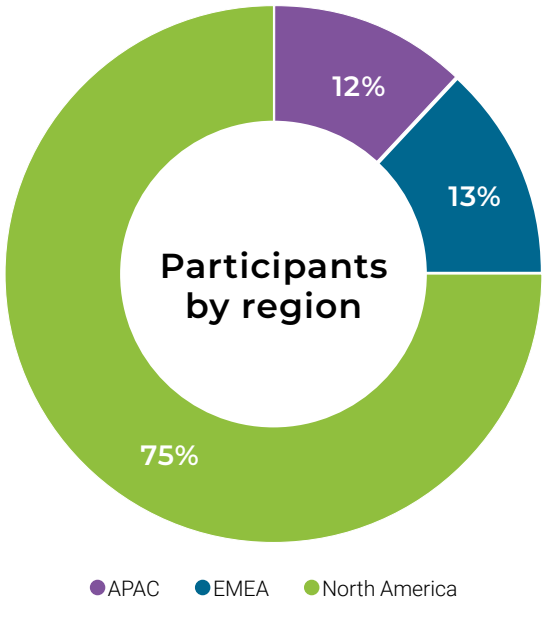


Figure 1: BSIMM VERTICAL PARTICIPANTS*

*Note: a company may be in more than one vertical





130
Firms in BSIMM13

5 years is the average age of SSGs in BSIMM13

3,342 SSG Members • **8,508** Security Champions

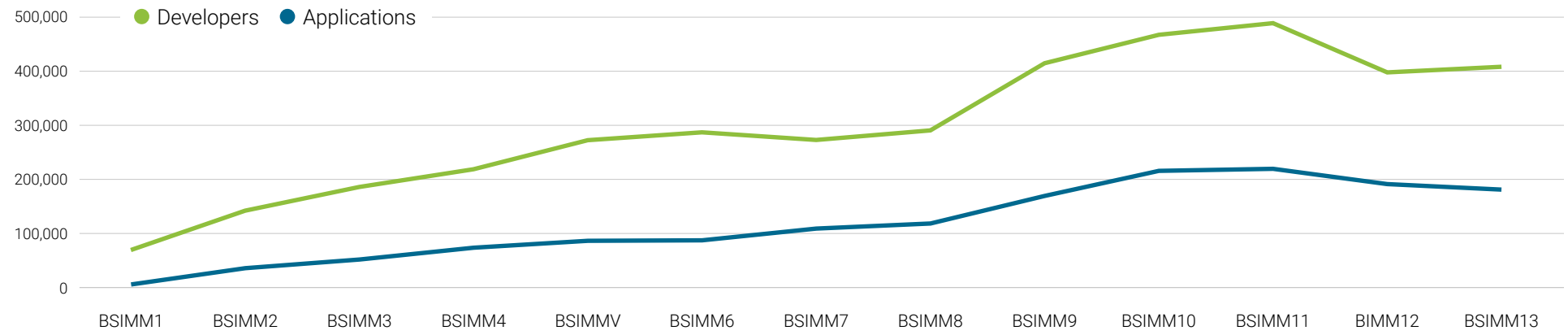


Figure 2: BSIMM BY THE NUMBERS

SIGNIFICANT APPSEC TRENDS

This section of the report details major trends observed during analysis of the BSIMM13 data and how those trends might influence software security strategies. Comparing your efforts against these trends can directly inform your strategies, whether you're in the early stages of implementing a software security initiative or want to fine-tune an extant security program for changing business and security needs.

For example, Figure 3 lists the five most observed activities in the BSIMM13 data pool. The numbers suggest that if your organization is starting its own software security program, you would do well to consider implementing these activities.

Where the data comes from

BSIMM data originates in interviews conducted with member firms during a BSIMM assessment. After each assessment, the observation data is anonymized and added to the BSIMM data pool, where statistical analysis is performed to highlight trends in how BSIMM firms are securing their software.

To look at the first activity, 90% of the organizations in the BSIMM13 data pool have established software security checkpoints in their software development lifecycles (SDLCs), indicating that the majority feel this is an important step to success in their software security initiatives. Checkpoints might include such things as in-IDE static analysis, code commit analysis, build-time static analysis, manual code review, dynamic scanning in a QA/integration test, and pre- and post-production penetration testing.

Proper metrics can help identify toolchain issues, or conversely, justify their expense. For example, if your static analysis tools fail to capture the security defects that surface during penetration testing, then there may be a problem in your code coverage. By capturing which checkpoint or tool was used to discover specific security defects, you can track such trends across your application portfolio.

One way to examine differences between last year's BSIMM12 and BSIMM13 is to look for trends, such as a high growth in observation rates among common activities. As an example, the observation rates for several activities grew at 20% or higher in BSIMM13 observations (see Figure 4).

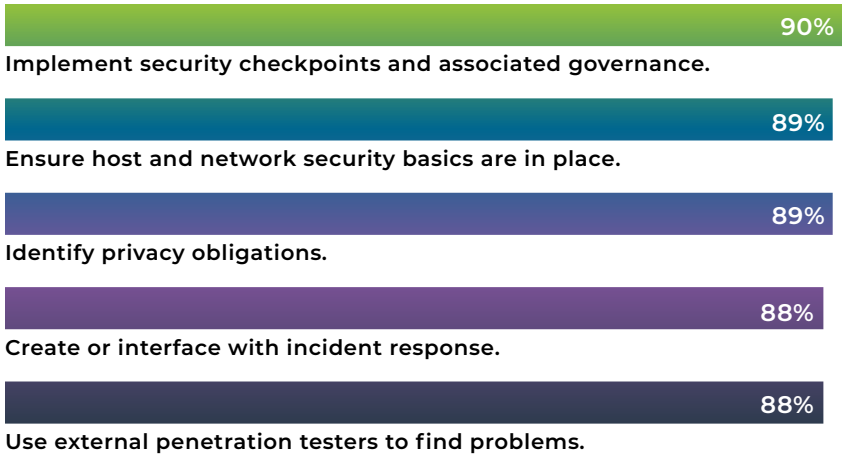


Figure 3: TOP 5 ACTIVITIES AS MEASURED BY BSIMM13

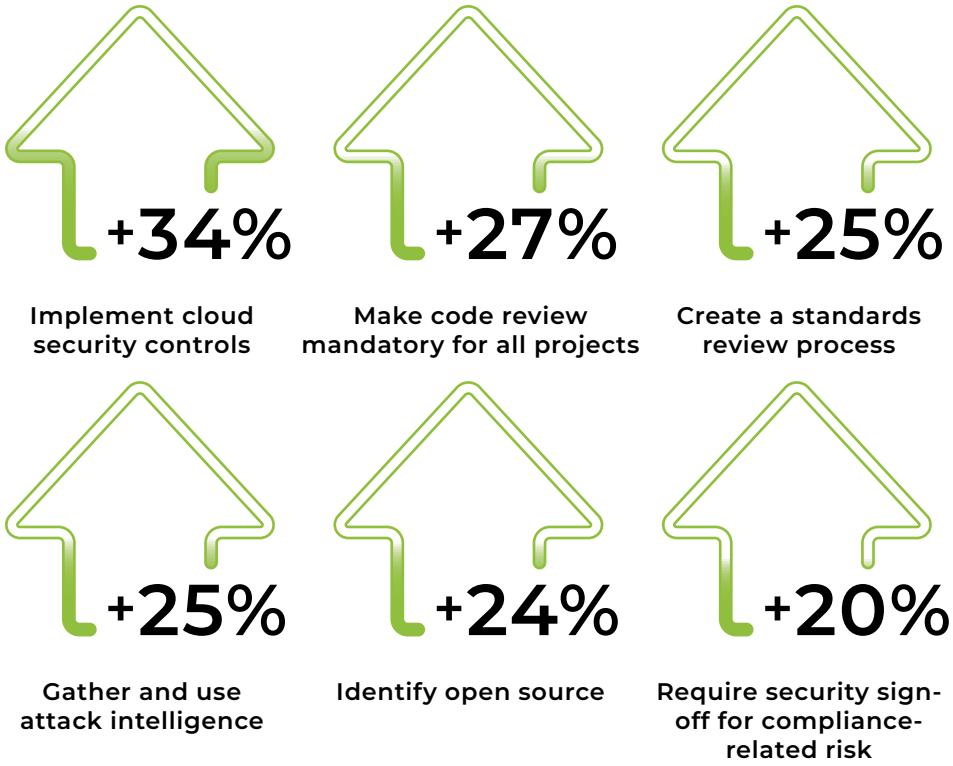


Figure 4: ACTIVITIES WITH HIGH GROWTH IN OBSERVATION RATES

The BSIMM takes a data-driven approach, using the industry's largest data set of worldwide cybersecurity practices to examine what organizations employ to build security into software development.

GENERAL INSIGHTS DERIVED FROM BSIMM13 TREND OBSERVATIONS

- With the use of automated code review tools activity being observed in more than 82% of all firms, software security groups are starting to make code review mandatory for all projects.
- Organizations are starting to scale their security testing across their complete application portfolio and including security testing in QA automation.
- More software security groups are moving to the maturing phase of their software security initiatives and are now working on the scalability, efficiency, and effectiveness aspects of their programs.
- Many BSIMM organizations are increasing their efforts to manage compliance risk, creating a repeatable way to document their compliance story. These activities are examples of what organizations do once they enter the maturing phase of their software security programs.
- In response to multiple high-profile breaches in the last few years, BSIMM data showed significant growth in activities used to address security, compliance, and risk mitigation. Organizations are responding to these breaches by investing in attack intelligence activities that they can use to improve their programs..

"SHIFT EVERYWHERE": CONTINUOUS TESTING THROUGHOUT THE SDLC

Starting more than 15 years ago, the "shift left" movement encouraged organizations to put security testing as early as possible in the development process. "Shift everywhere" has extended the trend into automated continuous testing throughout the software lifecycle.

A shift everywhere approach is useful for more than just testing for vulnerabilities in a timely fashion, it also facilitates automating governance checks and measuring risk in various parts of the software lifecycle. For example, "shifting" might entail using automated tests to continuously verify that only APIs with proper documentation are allowed to receive certain traffic.

The Importance of Continuous Defect Discovery

There is a trend toward continuous defect discovery, especially testing that can be automated into lifecycle tooling. For example, effort in the BSIMM Code Review and Security Testing practices each grew at almost twice the rate of effort in the Penetration Testing and Architecture Analysis practices.

Data Now Driving More Security Decisions

There was growth in security efforts among members of the BSIMM community in "*build a capability to combine AST results*" (56%), "*identify metrics and use them to drive resourcing*" (24%), and "*publish data about software security internally and drive change*" (16%).

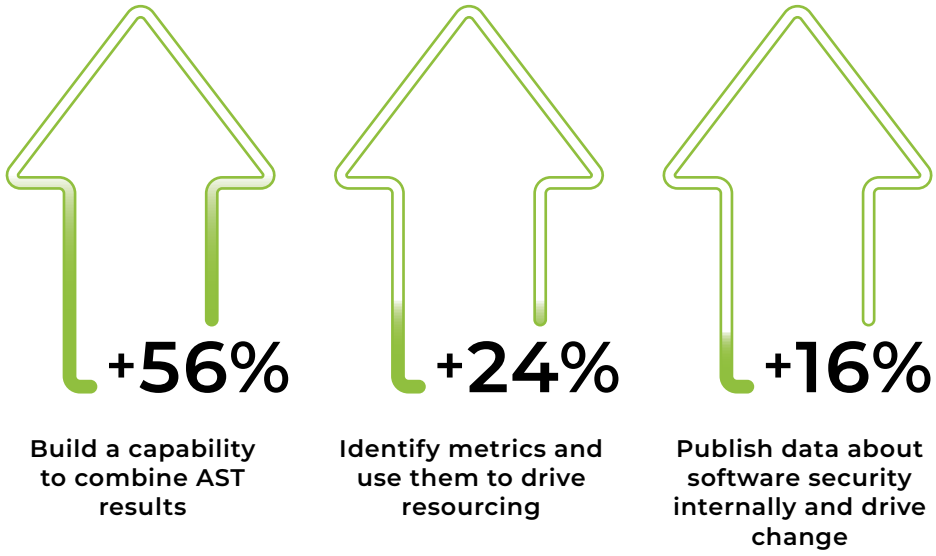


FIGURE 5: DATA NOW DRIVING MORE SECURITY DECISIONS

SOFTWARE SUPPLY CHAIN RISK MANAGEMENT AND THE RISE OF SBOMS

Software supply chain security has become a major concern for all organizations dependent on third-party software in the light of high-profile supply chain attacks. Most programs managing software supply chain risk focus on identifying and securing software—often open source software—that is destined for integration into in-house-developed software.

Many organizations are also enforcing policies to help ensure third-party suppliers are following best practices in securing their software. A recent report from Synopsys and the Enterprise Strategy Group (ESG) found that 73% of respondents have increased their efforts to secure their organizations' software supply chain through a variety of such initiatives.

Managing Open Source Risk through SCA

Open source software is now such a common part of software development that recent BSIMMs have reported significant increases in efforts to identify and manage open source. BSIMM data also confirms that more firms are getting better at managing open source risk. SCA tools continue to fuel year-over-year growth of the BSIMM's *"identify open source"* and *"control open source risk"* activities, both of which grew by nearly 35%.

Struggles with API Security & Visibility

While unmanaged open source software is known as a major supply chain concern, organizations are becoming increasingly aware of risks posed by the shift toward cloud-native application development—how these apps are stored, packaged, and deployed, as well as how they interface with one another through application programming interfaces (APIs). Indeed, the BSIMM13 "Foundations" report notes that almost all organizations are struggling with API security and visibility, given that many APIs aren't easily discoverable and can be labor-intensive to document once discovered. Some organizations are answering the API challenge by using automated tests to continuously verify that only APIs with proper vetting are allowed to receive sensitive traffic.

The Rise of Software Bills of Materials

To better manage supply chain risk, more BSIMM organizations are adding automated SBOM generation to fully identify the third-party software they use and to improve their ability to respond to disclosed vulnerabilities. Evidence of the movement toward SBOMs can be seen in the 30% growth of the *"create bills of materials for software"* BSIMM activity.

BETTER VENDOR MANAGEMENT BUT LESS EMPHASIS ON VENDOR TRAINING

As an outcome of the need for better supply chain management, many in the BSIMM community are demanding software security standards be enforced on vendor-supplied software. BSIMM13 observations of the *"communicate standards to vendors"* and *"ensure compatible vendor policies"* activities grew by 46% and 56% respectively.

BSIMM data shows that organizations are also increasing (by an average 15% year-over-year) their use of SLA terms in contracts with vendors to ensure that third-party software won't jeopardize compliance with their own software security standards.

However, not all trends happen in a positive direction. An activity with the largest drop in observations in BSIMM13 was *"provide training for vendors and outsourced workers."* Classically, observations of this activity have grown steadily over the lifetime of the BSIMM. In BSIMM13, however, the observation rate fell by 30%.

The BSIMM13 "Foundations" report speculates the decline might be linked to the observed growth in *"create SLA boilerplate"* and *"include software security SLAs in all vendor contracts"* activities. In other words, more organizations are specifying training requirements to their vendors rather than providing that training themselves.

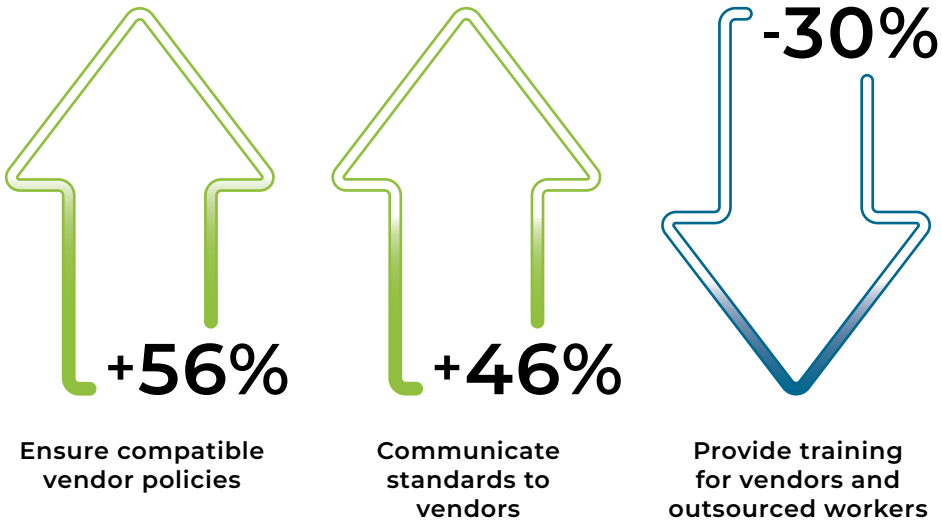


FIGURE 6: EMPHASIS ON VENDOR TRAINING

INTEGRATING SECURITY INTO DEVELOPER TOOLCHAINS

BSIMM13 data indicates that software development teams continue to make progress in integrating security options into CI/CD pipelines and toolchains. These integrations provide faster and tighter processes that reduce friction, improve coverage, and make the shift everywhere concept a reality.

In the early days of application security, firms found vulnerabilities everywhere they looked—in production, in released products, and, when things went wrong, in negative news about their software.

Moving to Smaller, Automated Checks within the SDLC

"Shift left" was a call to move testing efforts earlier into the development lifecycle to find and fix software vulnerabilities before they could be exploited. "Shift everywhere" uses smaller, faster, sometimes pipeline-driven, testing whenever there is an opportunity to check software. This is reflected by firms shifting to smaller automated checks embedded within the SDLC. For example, the *"include security tests in QA automation"* activity grew by nearly 50%. BSIMM13 also saw growth in the use of automated code review tools, as opposed to little or no growth in activities associated with pen testing or manual code review.

Automating and Enforcing Secure Coding Standards

It appears from BSIMM13 data that many organizations are finding success in enforcing coding standards by taking advantage of improved automation. Traditionally, the activities around creating and enforcing coding standards have been among the rarer activities observed in BSIMM assessments. In BSIMM13, however, observations of "*use secure coding standards"* grew by almost 90%.

EXPANDING SOFTWARE SECURITY BEYOND APPLICATIONS AND PRODUCTS

BSIMM13 data indicates security and operations teams are getting better at working together. For example, observations of the "*fix all occurrences of software bugs found in operations"* activity grew by 175%.

Of course, even better than fixing bugs is understanding how the bugs came to be, then building safeguards to prevent recurrences. Security and operations teams have readily taken on the task, as shown by the BSIMM *"enhance the SSDL to prevent software bugs found in operations"* activity growing by over 70%. Observations of the *"drive feedback from software lifecycle data back to policy"* activity grew by over 80%, showing that many BSIMM firms are updating policy based on their bug eradication efforts.

Capturing Security Knowledge for Knowledge-as-Code

Software security groups are increasingly working with infrastructure teams to capture security knowledge and encode it in human-readable, machine-deployable configurations. BSIMM activities related to building knowledge-as-code libraries grew by an average of 20%. Observations of efforts to build libraries of reusable and vetted security knowledge also grew by nearly 20%.

Intelligent Orchestration on the Rise for Containers

BSIMM13 firms are taking advantage of improved infrastructure automation and orchestration to deploy applications in containers that are monitored for configuration drift and non-compliance. This was reflected in a nearly 30% growth in observations of the *"use orchestration for containers and virtualized environments"* activity.

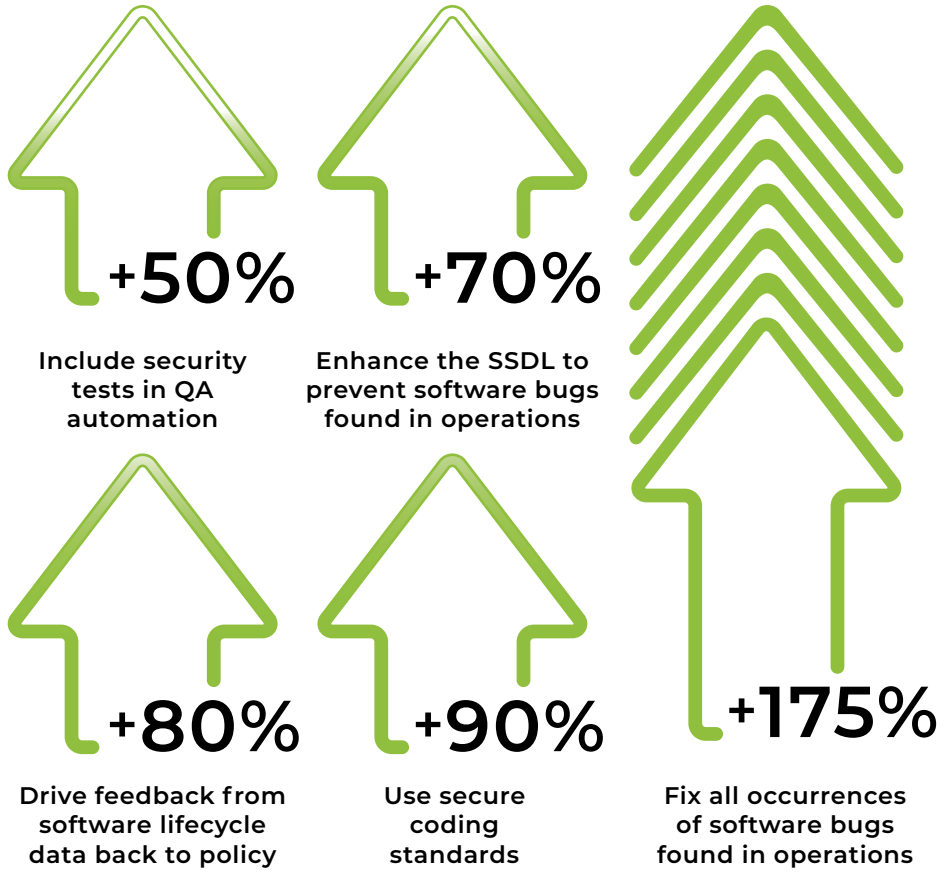


FIGURE 7: EXPANDING SOFTWARE SECURITY BEYOND APPLICATIONS AND PRODUCTS

NO SECURITY WITHOUT A PROGRAM

One thing that all members of the BSIMM community share in common is that each has a group dedicated to software security. In fact, without such a group, successfully carrying out a software security program is highly unlikely. The group might start as a team of one—just the software security leader—and expand over time. The group might start life as a corporate or engineering team, or some hybrid of two more teams, but BSIMM data strongly indicates that creating a software security group is a crucial first step in creating a viable software security program.

No two of the 130 firms in the BSIMM community have the same structure for their respective software security groups and programs, but at the highest level, they all share common features, such as:

- Organized to provide software security services
- Organized around setting and verifying adherence to policy
- Designed to mirror business unit organizations
- Organized with a hybrid policy and services approach
- Structured around managing a team of experts doing software security work across the development or engineering organizations

One of the first initiatives that many software security groups undertake is to identify people such as developers, testers, architects, and DevOps engineers who are a driving force in improving software security but may not be directly connected to the software security group. Collectively referred to as "software security champions," these people can enable a software security group to scale its efforts while not having to expand the group itself. Champions in engineering teams, for instance, encourage engineers to own the security of their software deliverables.

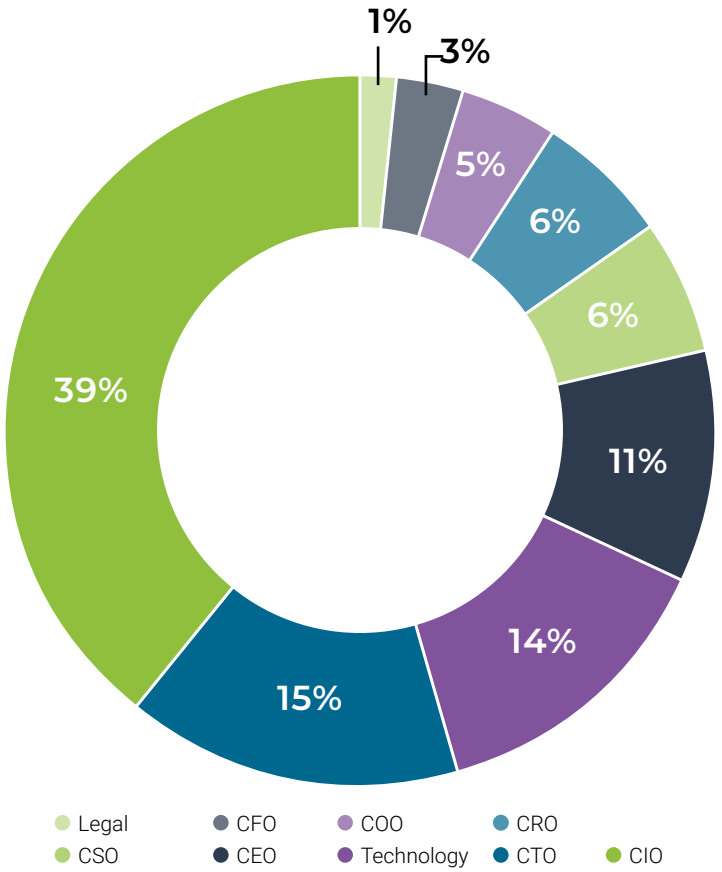


Figure 8: WHO THE CISO REPORTS TO

ROLES IN A SOFTWARE SECURITY INITIATIVE

Executive Leadership

Historically, security initiatives that have impact are sponsored by a senior executive who creates a software security group where software security testing and operations are distinctly separate from software delivery. Security initiatives without that executive sponsorship have had little lasting impact across any given organization. By identifying a senior executive and putting them in charge of software security, the organization can address two "Management 101" concerns: accountability and empowerment.

Although many software security groups have a CISO as their nearest executive, there are a variety of executives overseeing software security efforts in the 130 BSIMM13 firms.

BSIMM-V found CISOs as the nearest executive in 21 of 67 firms, which grew in BSIMM6 to 31 of 78, and again for BSIMM7 with 52 of 95. Since BSIMM7, the percentage has remained relatively flat even as the BSIMM community has grown.

Looking across all the executives nearest to software security group owners, there is a large spread in the reporting path to executive leadership for BSIMM10 through BSIMM13, as shown below.

The larger green circles show by percentage the leader's nearest executive in the BSIMM13 data pool, while smaller circles show the percentages for previous BSIMMs.

For example, a CISO is the closest executive in 51% of organizations in the BSIMM13 community, and that percentage ranged from 50% to 55% in BSIMM10 through BSIMM12.

CISOs in turn report to different executives among the 130 BSIMM13 firms. Figure 9 shows that CISOs report most commonly to CIOs (26 of 66, or almost 40% of the time) and report directly to the CEO only 10% of the time (7 of 66).

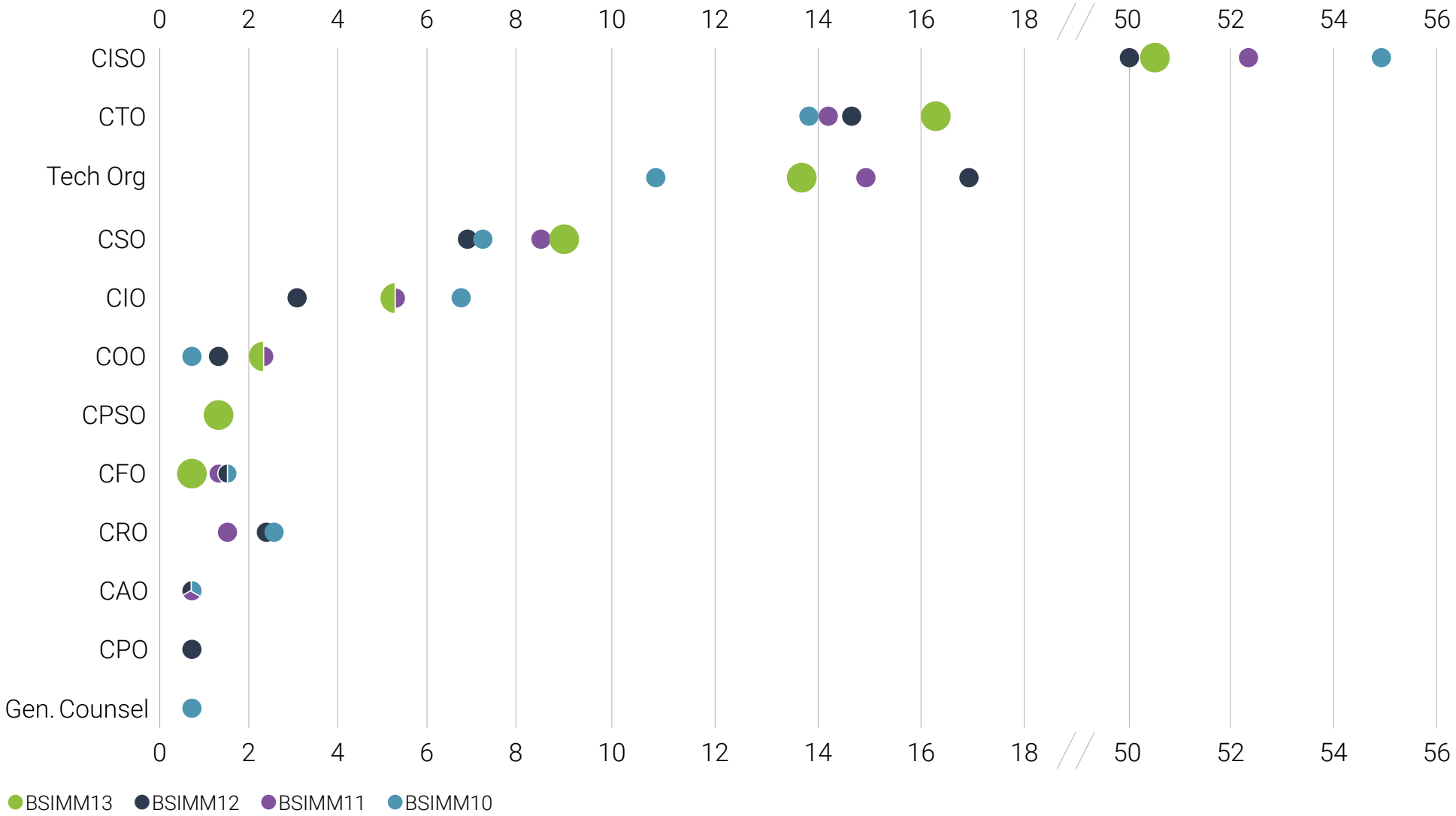


Figure 9: NEAREST EXECUTIVE TO SOFTWARE SECURITY GROUP

Software Security Group Leaders

The BSIMM defines "leaders" as the people in charge of day-to-day software security efforts in the 130 BSIMM13 software security initiatives. These leaders have a variety of titles, such as:

- Application Security Architect
- Application Security Manager
- Director Application Security
- Director Cybersecurity
- Director IT Risk Management
- Director IT Shared Services
- Director Product Security
- Director Security Assurance
- Executive Director Product Security
- Information Assurance Director
- Lead Security Architect
- Manager Software Security Engineering
- Product Security AppSec
- Security Director
- Security Engineering Manager
- Security Architect
- Senior Director Product Security
- SVP Product Security & Technology
- VP Product and Application Security
- VP Security Architecture
- VP Security Compliance

As shown in Figure 10, the leaders are typically one to two hops from the nearest executive (e.g., a CxO or related technology organization title), who is a further one to two hops away from the CEO. When the software security group leader is an executive, they are CISOs almost 70% of the time, with the other most common titles being CTO and CPSO (Chief Product Security Officer).

Key Stakeholders

Most software security groups are true cross-departmental efforts that involve a variety of stakeholders:

- Builders, including developers, architects, and their managers.
- Testers, who typically conduct functional and feature testing but may also include security testing. Some testers are beginning to anticipate how software architectures and infrastructures can be attacked and are using both automated and manual testing to ensure adequate security testing coverage.
- Operations teams. Development and operations are collapsing into one or more DevOps teams, resulting in an increasing amount of security effort through that combination.
- Administrators tasked to create and maintain secure builds, especially when it comes to the applications they host or attach to as services in the cloud.
- Executives and middle management, including business owners and product managers. Any sizable business today depends on software to work; thus, software security is a business necessity. Executives are the group that must provide resources for efforts that directly improve software security and efforts related to infrastructure and governance-as-code.
- Data privacy specialists, who form an integral part of the software security effort in some firms and combine forces with security specialists when engaging with engineering.
- Vendors, including those who supply on-premises products, custom software, and software-as-a-service, who are increasingly tasked to assure that their products are part of a secure software supply chain.

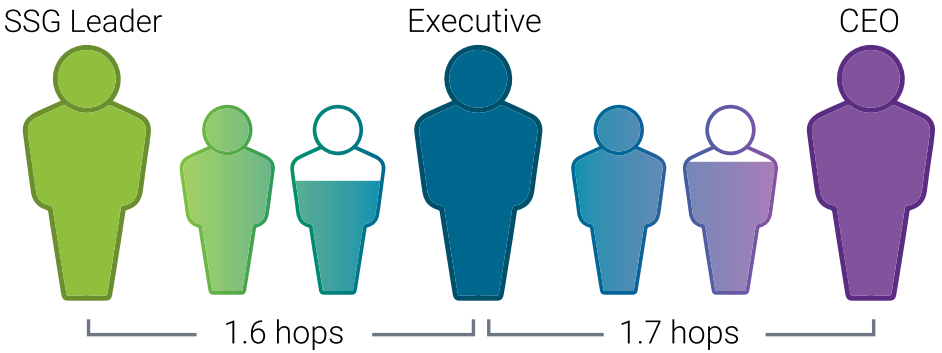


Figure 10: SOFTWARE SECURITY GROUP LEADERSHIP REPORTING CHAINS

SECURITY CHAMPIONS: THE JIMINY CRICKETS OF SOFTWARE SECURITY

> **"I dub you Pinocchio's conscience, lord high keeper of the knowledge of right and wrong, counselor in moments of high temptation, and guide along the straight and narrow path. Arise, Sir Jiminy Cricket."**
>
> —THE BLUE FAIRY, "PINOCCHIO"

No matter its size, no software security group can be everywhere or provide all the security coverage an organization needs. A security champions program puts a formal mantle on deputizing people throughout the SDLC to execute tasks such as integrating security tools, remediating security defects, responding to security incidents, offering just-in-time training, and promoting good security practices.

The term, "security champion" was originally coined to describe a developer with an interest in software security willing to champion security awareness at the team level. Over the years, the term has broadened to embrace everyone from those building the code to software architects to testers to operations teams, managers and the executive team, to even external vendors who are key links in securing the software supply chain. Think of them as people who may not be formally connected to a software security group but who act as drivers in efforts to improve an organization's software security posture.

Security champions don't need to be security pros, they just need to act as the security conscience of the team, keeping their eyes and ears open for potential issues and surfacing them when discovered. They're the "Jiminy Crickets" of software security, high keepers of knowledge of right and wrong, the guides along the straight and narrow.

Security champions can act as a sounding board for the feasibility of proposed software security changes and improvements. Understanding how changes might affect project timelines and budgets helps software security groups identify potential issues and address them before they become roadblocks.

Successful security champions get together regularly to compare notes, learn new technologies, and expand stakeholder understanding of the organization's overall software security challenges. Champions regularly meet to share code, scripts, tools, and new security features while promoting security awareness.

Security Champions Programs Work!

A perennial BSIMM trend is that, on average, firms having a security champions program score higher in BSIMM assessments than firms without one. In BSIMM13, that difference was a dramatic 35%.

Sixty-nine percent of BSIMM13 participants that have been assessed more than once have a security champions program, while 62% of the firms on their first assessment did not. In fact, BSIMM assessments indicate that many organizations new to software security implement a champions program as one of the first steps of their software security initiative.

How to Build a Security Champions Program

- **Get leadership buy-in.** Make sure stakeholders, including management and especially the leaders of your software security group, are willing to invest the time and money to make the champions program effective.
- **Identify potential champions.** Champions are often members of the development team, but a comprehensive security champions program should try to include members from QA, architects, designers, DevOps, operations, product managers, and even contractors or those working for external vendors. Few of us look forward to additional work, so recruit those people who already have a proactive attitude toward software security. Again, assure you have buy-in from the prospective champions' managers and that management understands and values the role of security champion.
- **Set expectations.** Define what each security champion is expected to do and incorporate those goals into their preexisting workflow to minimize confusion and conflicts.
- **Build community.** Make sure your security champions have ample opportunity to meet with each other, the security team, and outside experts to discuss specific issues and overall trends.
- **Provide training.** Provide the training, tools, and knowledge your security champions will need to succeed—be it eLearning training in best practices for code development to automated static analysis tools for reviewing code for flaws.
- **Track and measure.** Set up goals, metrics, and KPIs for your security champions program to demonstrate ROI to your organization.

> **On average, firms having a security champions program score higher in BSIMM assessments than firms without one. In BSIMM13, that difference in scores was 35%!**

RECOMMENDATIONS

Whether you're in the process of creating a software security initiative or maintaining a mature program, BSIMM13 data indicates you should be considering the following actions:

- **Put automated software security tools into place.** Whether used for static or dynamic testing or SCA, these tools can help remedy defects and identify known vulnerabilities in your software, whether that software was developed in-house, is commercial third-party software, or is open source.
- **Use data to drive security decisions**. Collect and combine data from your security testing tools and use that data to create and enforce software security policies. Gather data on what testing was performed and what issues were discovered to drive security improvements in both the SDLC and your governance processes.
- **Move toward automating security testing and decisions**. Move away from human-intensive manual approaches to more effective, consistent, and repeatable automated approaches.
- **Move to smaller, automated checks within the SDLC.** Whenever possible, replace manual activities such as pen testing or manual code review with smaller, faster, pipeline-driven testing when there is an opportunity to check software.
- **Create a comprehensive SBOM as soon as possible.** An SBOM should inventory your assets, along with open source and third-party code. In its 2020 Magic Quadrant for Application Security Testing, Gartner predicted, "By 2024, the provision of a detailed, regularly updated software Bill of Materials by software vendors will be a non-negotiable requirement for at least half of enterprise software buyers, up from less than 5% in 2019."[1]

While it may be theoretically possible to create a BOM manually, maintaining one requires a significant investment of human time. A BOM generated by an automated tool can provide comprehensive information (such as specific versions, vulnerability information, and licenses of the code in use) and, in the case of open source, a better understanding of dependencies that the open source components may be using.

For those readers who don't have a formal software security initiative, you need to begin working toward one without delay. Start by creating an actionable roadmap for your security and development teams—engage a professional software security assessment team to help you create that roadmap if necessary.

Assess the current state of your security program. Define the target future state you want to achieve, then identify the gaps between where you are today and where you need to go. After that, you can build out your action plan, using Chapter IV ("A Quick Guide to SSI Maturity") of the BSIMM13 "Foundations" report as your baseline to plan out improvements.

For more information and access to BSIMM resources, please visit www.bsimm.com/

1. Mark Horvath, Dionisio Zumerle, and Dale Gardner, Magic Quadrant for Application Security Testing, Gartner, 4/29/2020.

ACKNOWLEDGEMENTS

Our thanks to the 130 executives, including those who wish to remain anonymous, from the SSIs we studied to create BSIMM13.

Our thanks also to the nearly 140 individuals who helped gather the data for the BSIMM data pool over time.

In particular, we thank Tony Blakemore, Adam Brown, Matthew Chartrand, Eli Erlikhman, Jacob Ewers, Stephen Gardner, iMan Louis, Sammy Migues, Alistair Nash, Kevin Nassery, Donald Pollicino, Brendan Sheairs, Denis Sheridan, and Li Zhao.

BSIMM13 was authored by Jamie Boote, Eli Erlikhman, Stephen Gardner, and Sammy Migues. In addition, we give a special thank you to Kathy Clark-Fisher and Ryan Francis, whose behind-the-scenes work keeps the BSIMM science project, conferences, and community on track.

AARP
Adobe
Aetna
Ally Bank
Axway
Bank of America
Bell Network
CIBC
Cisco
Citi
Diebold Nixdorf
Depository Trust & Clearing Corporation
Egis
Eli Lilly and Company
eMoney Advisor
EQBank
Equifax
Fidelity
Finastra
Freddie Mac
F-Secure
Genetec
HCA Healthcare
Honeywell CE
HSBC
Imperva
Inspur Software
Intralinks
iPipeline
Johnson & Johnson
Landis+Gyr
Lenovo
MassMutual
MediaTek
Medtronic
Navient
Navy Federal Credit Union
NEC
NetApp
Oppo
PayPal
Pegasystems
Principal Financial
Realtek
Reckitt
SambaSafety
ServiceNow
Signify
SonicWall
Synchrony Financial
TD Ameritrade
Teradata
Trainline
Trane
U.S. Bank
Veritas
Verizon Media
Vivo
World Wide Technology
ZoomInfo