BSIMM12
2021 INSIGHTS & TRENDS REPORT

TABLE OF CONTENTS

PART ONE: EXECUTIVE SUMMARY

PART TWO: ACTIVITIES

PART THREE: APPENDIX

PART ONE

EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

UNDERSTANDING THE BSIMM

In 2008, consultant, research, and data experts from what is now the Synopsys Software Integrity Group set out to gather data on the different paths that organizations take to address the challenges of securing software. Their goal was to examine organizations that were highly effective in software security initiatives, to conduct in-person interviews on those organizations' activities, and to publish their findings.

The result was the Building Security In Maturity Model (better known as the BSIMM)—a descriptive model that provides a baseline of observed activities for software security initiatives. Because these initiatives often use different methodologies and different terminology, the BSIMM also creates a common vocabulary for software security initiatives.

BSIMM12 PARTICIPANTS

This 2021 edition of the BSIMM report—BSIMM12—examines anonymized data from the software security activities of 128 organizations across various verticals, including financial services, FinTech, independent software vendors (ISVs), Internet of Things (IoT), healthcare, cloud, and technology organizations (see Figure 1).

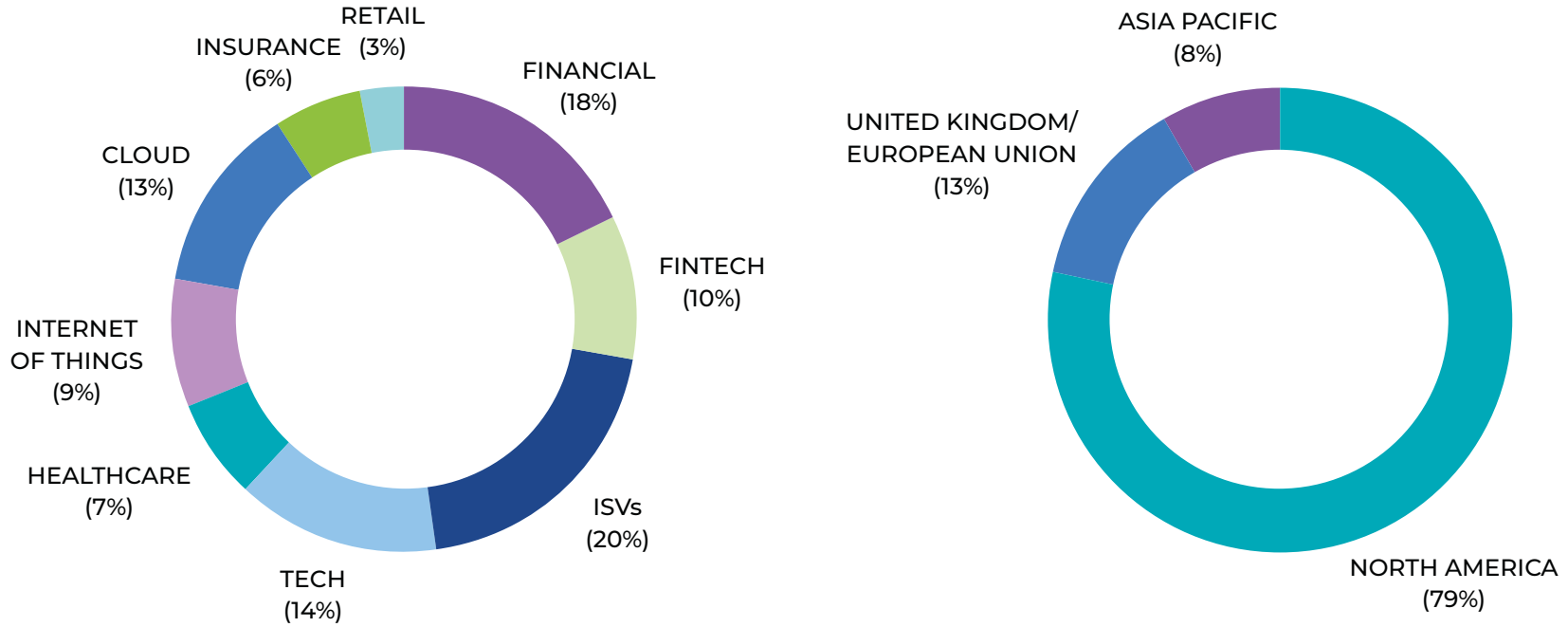


FIGURE 1. BSIMM12 PARTICIPATING FIRMS. *Participant percentages per tracked vertical and geography.*

THE BSIMM FRAMEWORK

BSIMM observations use a framework of 12 software security practices organized under four domains, with those domains—Governance, Intelligence, SSDL Touchpoints, and Deployment—currently embracing 122 activities. The Governance domain, for example, includes activities that fall under the organization, management, and measurement practices of a software security initiative.

Descriptions of the BSIMM domains, practices, and activities can be found at **https://www.bsimm.com**. A companion *Foundations of BSIMM12* document providing more in-depth detail on BSIMM12 background, data, and observations can be found at **https://www.bsimm.com/resources.html**.

From an executive perspective, you can view BSIMM activities as controls implemented in a software security risk management framework. The implemented activities might function as preventive, detective, corrective, or compensating controls in your software security initiative. Positioning the activities as controls allows for easier understanding of the BSIMM's value by Governance, Risk & Compliance, Legal, Audit, and other risk management groups. BSIMM activity levels distinguish the frequency with which activities are observed in the participating organizations. Frequently observed activities are designated level 1, with less frequent and infrequently observed activities designated as levels 2 and 3, respectively.

BSIMM NUMBERS OVER TIME

The BSIMM project has grown from nine participating companies in 2008 to 128 in 2021, with now nearly 3,000 software security group members and over 6,000 satellite (aka security champion) members. The average age of the participants' software security initiatives is 4.4 years. As Table 1 shows, the BSIMM numbers may slightly fluctuate year-over-year as participants enter and leave the BSIMM community.

BSIMM NUMBERS OVER TIME												
	BSIMM12	BSIMM11	BSIMM10	BSIMM9	BSIMM8	BSIMM7	BSIMM6	BSIMM-V	BSIMM4	BSIMM3	BSIMM2	BSIMM1
FIRMS	128	130	122	120	109	95	78	67	51	42	30	9
MEASUREMENTS	341	357	339	320	256	237	202	161	95	81	49	9
2ND MEASURES	31	32	50	42	36	30	26	21	13	11	0	0
3RD MEASURES	14	12	32	20	16	15	10	4	1	0	0	0
4TH MEASURES	4	7	8	7	5	2	2					
SSG* MEMBERS	2,837	1,801	1,596	1,600	1,268	1,111	1,084	976	978	786	635	370
SATELLITE MEMBERS	6,448	6,656	6,298	6,291	3,501	3,595	2,111	1,954	2,039	1,750	1,150	710
DEVELOPERS	398,544	490,167	468,500	415,598	290,582	272,782	287,006	272,358	218,286	185,316	141,175	67,950
APPLICATIONS	153,519	176,269	173,233	135,881	94,802	87,244	69,750	69,039	58,739	41,157	28,243	3,970
AVG. SSG AGE (YEARS)	4.41	4.32	4.53	4.13	3.88	3.94	3.98	4.28	4.13	4.32	4.49	5.32

TABLE 1. BSIMM NUMBERS OVER TIME. *The chart shows how the BSIMM study has grown over the years (*SSG=software security group).*

THE BSIMM ROADMAP

In the rapidly changing software security field, understanding what other organizations are doing in their software security initiatives can directly inform your strategy through a comparison of your own software security activities against BSIMM data. This document provides a high-level summary of observed trends and insights gained from the data gathered in BSIMM12.

BSIMM TERMINOLOGY

- **ACTIVITY.** Actions carried out or facilitated by a software security group (SSG) as part of a practice. Activities are divided into three levels in the BSIMM based on observation rates. Frequently observed activities are designated level 1, with less frequent and infrequently observed activities designated as levels 2 and 3, respectively.
- **CAPABILITY.** A set of BSIMM activities spanning one or more practices working together to serve a cohesive security function.
- **CHAMPIONS.** Interested and engaged developers, cloud security engineers, deployment engineers, architects, software managers, testers, and others who contribute to the security posture of the organization and its software.
- **DOMAIN.** One of the four categories the BSIMM framework is divided into. The domains include Governance, Intelligence, Secure Software Development Lifecycle (SSDL) Touchpoints, and Deployment.
- **PRACTICE.** BSIMM activities are organized into 12 practices. Each of the four BSIMM domains has three practices.
- **SATELLITE.** A group, sometimes termed "champions," organized and leveraged by a software security group (SSG).
- **SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SSDL).** A software lifecycle with integrated software security checkpoints and activities.
- **SOFTWARE SECURITY FRAMEWORK (SSF).** The underlying structure of the BSIMM, comprising 12 practices divided into four domains.
- **SOFTWARE SECURITY GROUP (SSG).** The internal group charged with carrying out and facilitating software security.
- **SOFTWARE SECURITY INITIATIVE (SSI).** An organization-wide program to instill, measure, manage, and evolve software security activities in a coordinated fashion.

PART TWO

ACTIVITIES

SOFTWARE SECURITY ACTIVITIES AS MEASURED BY BSIMM12

The popular business book, *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 software security initiatives. Table 2 lists the 10 most observed activities in the BSIMM12 data pool, all commonly found in highly successful software security initiatives. The data suggest that if your organization is working on its own software security initiative, you should consider implementing these activities.

BSIMM12 TOP 10 ACTIVITIES BY OBSERVATION COUNT		
	OBSERVED COUNT FROM 128 PARTICIPANTS	ACTIVITY DESCRIPTION
1	92% (118 PARTICIPANTS)	Implement lifecycle instrumentation and use to define governance
2	91% (117 PARTICIPANTS)	Ensure host and network security basics are in place
3	89% (114 PARTICIPANTS)	Identify PII obligations
4	88% (113 PARTICIPANTS)	Perform security feature review
5	87% (111 PARTICIPANTS)	Use external penetration testers to find problems
6	84% (108 PARTICIPANTS)	Create or interface with incident response
7	80% (102 PARTICIPANTS)	Integrate and deliver security features
8	80% (102 PARTICIPANTS)	Use automated tools
9	78% (100 PARTICIPANTS)	Ensure QA performs edge/boundary value condition testing
10	77% (99 PARTICIPANTS)	Translate compliance constraints to requirements

TABLE 2. TOP 10 BSIMM12 ACTIVITIES.

IS YOUR SOFTWARE SECURITY INITIATIVE KEEPING PACE WITH CHANGE?

1. **Do you maintain a current view of all your software assets, including internal code, third-party code, open source, automation scripts, infrastructure-as-code, and other software assets?**
2. **Are you making risk management decisions using a bill of materials detailing all software in the software security initiative's purview?**

BREAKING DOWN THE TOP 10 ACTIVITIES

1 IMPLEMENT LIFECYCLE INSTRUMENTATION AND USE TO DEFINE GOVERNANCE

BSIMM12 found that software security leaders are dramatically shifting to implementation of risk-based controls across the entire software portfolio, enabling development teams to find and fix issues earlier in the software development lifecycle. The vast majority—92%—of BSIMM12 participants have implemented some form of this activity.

Secure software lifecycle processes are proactive approaches to building security into an application throughout development. In essence, "lifecycle instrumentation" advocates are working software security tightly into the application development process by collecting data at various stages of the software development lifecycle and using that data to create and enforce software security policies.

2 ENSURE HOST AND NETWORK SECURITY BASICS ARE IN PLACE

Trying to implement software security before putting host and network security in place is like putting on shoes before socks. Almost all BSIMM12 participants—91%—have started a good foundation for software security by ensuring that host and network security basics are in place across their data centers and networks.

3 IDENTIFY PII OBLIGATIONS

As the BSIMM12 observations indicate, securing Personally Identifiable Information (PII) is a top priority for many organizations, with 89% of participants having identified their PII requirements and 43% having also built a PII inventory. Outsourcing to hosted environments doesn't relax PII obligations and can even increase the difficulty of recognizing all associated obligations. Understanding where PII resides and preventing unauthorized disclosure of PII data are steps every security-minded business needs to take.

4 PERFORM SECURITY FEATURE REVIEW

When beginning an architecture analysis, security-aware organizations center the process on a review of security features. For example, a security feature review would identify a system that was subject to escalation of privilege attacks or a mobile application that incorrectly puts PII in local storage. Eighty-eight percent of BSIMM12 participants have implemented this activity.

5 USE EXTERNAL PENETRATION TESTERS TO FIND PROBLEMS

No one is a prophet in their own land, an adage that 87% of BSIMM12 participants have recognized. While internal software security champions may go unheard, using external penetration testers can clearly demonstrate to the organization that it isn't immune to security issues.

BREAKING DOWN THE TOP 10 ACTIVITIES continued

6 CREATE OR INTERFACE WITH INCIDENT RESPONSE

Eighty-four percent of BSIMM12 participants have initiated a process to have their software security group link up with the organization's incident response team to keep critical security information flowing in both directions. Opening communication channels with infrastructure and software vendors is also a very important task for software security groups.

7 INTEGRATE AND DELIVER SECURITY FEATURES

Rather than having each project team implement its own security features, 80% of BSIMM12 software security groups drive or participate in clearinghouse efforts for approved security features. Project teams benefit from implementations that come preapproved by the software security group, and the group benefits by not having to repeatedly track down errors that often creep into security features.

8 USE AUTOMATED TOOLS

As applications and networks grow in complexity, it becomes increasingly difficult to manually manage security and compliance. Manual operations can result in slower detection and remediation of issues, errors in resource configuration, and inconsistent policy application, leaving an organization vulnerable to compliance issues and attack. Eighty percent of BSIMM12 participants noted that they were using automated tools to help secure their code and applications. Conversely, in 2021, it's notable that as many as 26 BSIMM12 participants are apparently still reliant on manual software security and compliance processes, although each may have valid reasons to do so.

9 ENSURE QA PERFORMS EDGE/BOUNDARY VALUE CONDITION TESTING

The majority—78%—of BSIMM12 participants acknowledge the value of pushing past standard functional testing that only uses expected input. More and more QA teams are moving toward thinking like an adversary and taking a proactive software security mindset.

10 TRANSLATE COMPLIANCE CONSTRAINTS TO REQUIREMENTS

Translating compliance constraints into software requirements that are then communicated to development teams is in 77% of BSIMM12 participants' strategies. Representing compliance constraints as software requirements helps with traceability and visibility in the event of an audit.

GROWTH IN ACTIVITIES

Activities—that is, actions taken or facilitated by software security groups—tend to change over time as the software security environment and the group's priorities change. For example, the data indicate a 61% increase in the *identify open source* activity over the past two years, probably due to the prevalence of open source components in modern software and the rise of attacks using popular open projects as vectors.

Table 3 shows the highest growth activities observed in the BSIMM data over the past 24 months. Three activities introduced as early as BSIMM7—*ensure cloud security basics, use orchestration for containers and virtualized environments,* and *use application containers to support security goals*—show high rates of percentage increases due to their relative newness to the BSIMM project. For example, only five observations of *use orchestration for containers and virtualized environments* were made in the 12 months after the activity was first introduced in BSIMM9, while 33 observations were made two years later for BSIMM12. Nonetheless, the growth in these activities demonstrates the impact that cloud platforms and container technologies have had on how organizations use and secure software.

ACTIVITY	OBSERVATION COUNT BSIMM10	OBSERVATION COUNT BSIMM12	PERCENTAGE INCREASE
Use orchestration for containers and virtualized environments	5	33	560%
Ensure cloud security basics	9	59	555%
Use application containers to support security goals	14	44	214%
Perform design review for high-risk applications	29	49	69%
Include security resources in onboarding	28	46	64%
Identify open source	46	74	61%
Create SLA boilerplate	35	55	57%
Integrate opaque-box security tools into the QA process	32	50	56%
Ensure executive awareness of compliance and privacy obligations	56	74	32%
Unify regulatory pressures	81	98	21%
Use automated tools	85	102	20%

TABLE 3. HIGHEST GROWTH ACTIVITIES. *Activity increases from BSIMM10 to BSIMM12 as measured by overall changes in observation percentages.*

COMPARING BSIMM12 VERTICALS

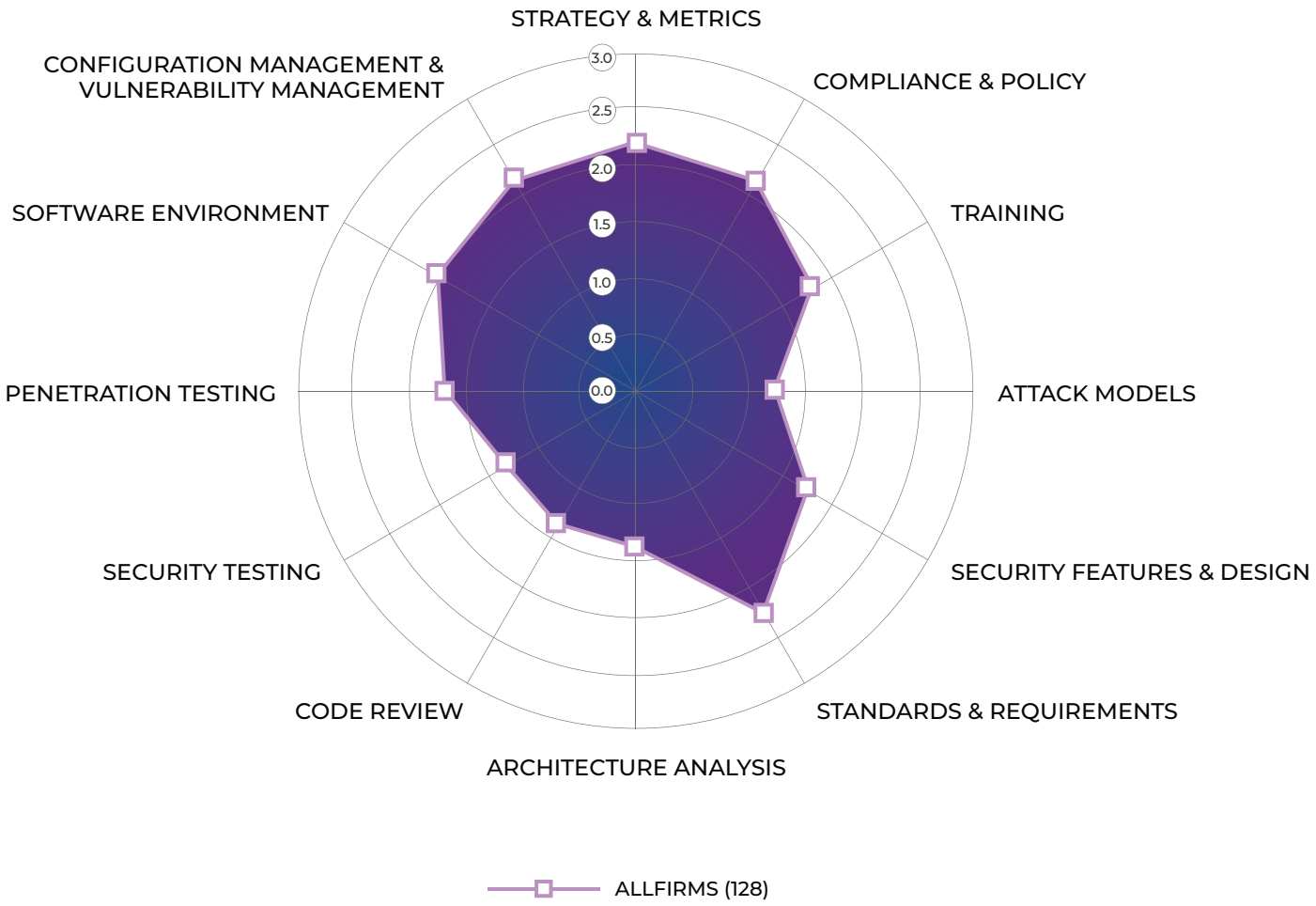


FIGURE 2. OBSERVATION OF LEVEL 2 AND LEVEL 3 ACTIVITIES. *The average of the high-water mark collectively reached in each practice by the 128 BSIMM12 firms.*

BSIMM high-water mark charts provide a baseline for comparing the frequency of high-level activities of the 128 firms for each practice. Activity levels distinguish the frequency with which activities are observed in the participating organizations. Frequently observed activities are designated level 1, with less frequent and infrequently observed activities designated as levels 2 and 3, respectively.

A high-water mark of level 3, and to a lesser extent level 2, typically indicates maturity. As shown in Figure 2 above, collectively across the firms observed for BSIMM12, more level 2 and 3 activities were observed in practices such as Strategy & Metrics, Compliance & Policy, and Standards & Requirements as opposed to practices such as Attack Models, Architecture Analysis, Code Review, and Security Testing.

IOT, CLOUD, AND ISV VERTICALS

IoT, cloud, and ISV firms create software solutions that are commonly deployed differently than each other. For example, IoT firms show a higher level of maturity in practices related to front-loading design (that is, an emphasis on decisions at earlier stages of the design process), such as the Training, Security Features & Design, and Architecture Analysis practices. In Architecture Analysis, IoT marks are significantly higher than other verticals, perhaps because many IoT devices are expected to live in production environments for long periods of time.

Cloud and ISV firms share a similar pattern, except for the Code Review practice where members of the Cloud vertical are ahead of the other two verticals (see Figure 3). This trend might be explained by the explosion of code created by cloud firms over the past few years, which has consequently resulted in an expanded need for code review.

Partially in response to the growing awareness of IoT device abuse and privacy issues, IoT has outpaced the ISV vertical in several practices, including the training and testing practices, which IoT firms have adopted at a greater pace than ISVs.

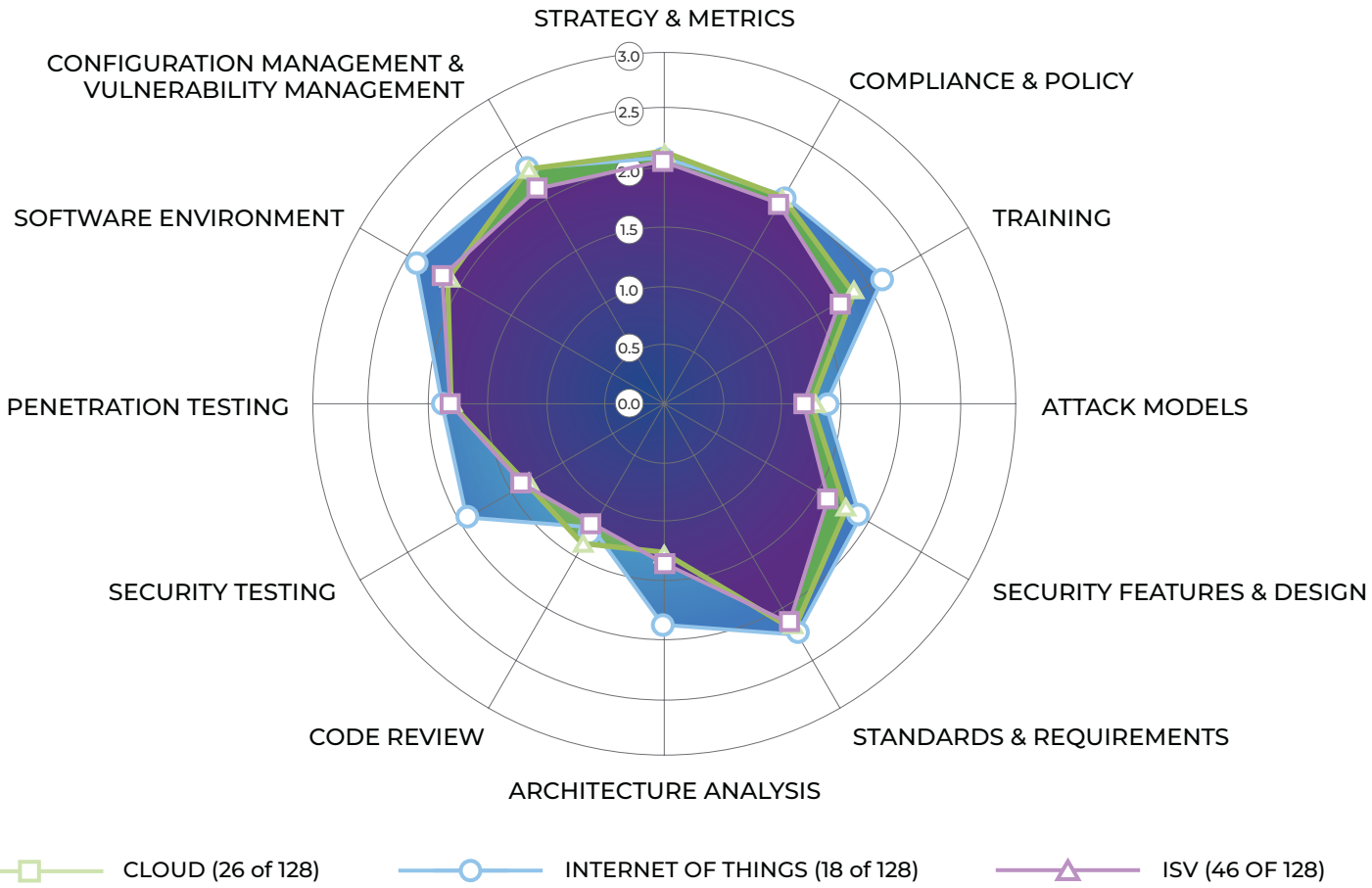


FIGURE 3. COMPARING CLOUD VS. IOT VS. ISV.

IOT AND FINTECH

IoT security programs include a focus on protecting software on devices, and we see this emphasis in the adoption of binary secret protection and integrity validation activities. Interestingly, IoT and FinTech firms are identifying open source at similar observation rates, but FinTech firms show more than double the observation rate in controlling open source risk. This may be due to FinTech firms perceiving more consequences associated with open source vulnerability risks, as compared to IoT firms perceiving greater risks from license violations in their devices' software.

As Figure 4 demonstrates, it's also interesting to review what each vertical considers valuable. Both the IoT and FinTech verticals perceive the *perform security feature review* activity to be important, but the *perform design review for high-risk applications* was observed much more frequently in IoT firms. On the other hand, the *use a risk methodology to rank applications* activity was seen much more frequently in FinTech firms.

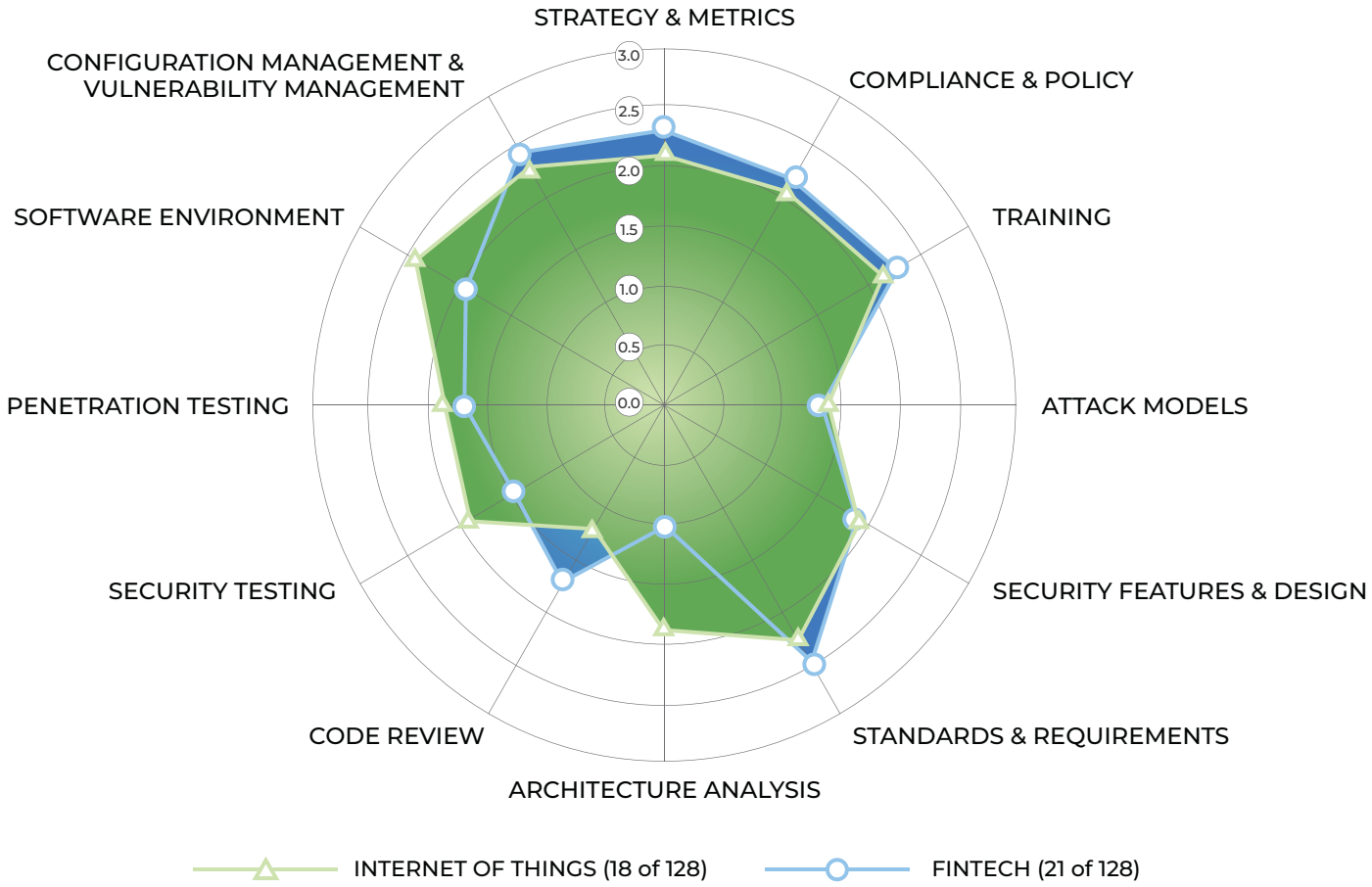


FIGURE 4. IOT VS. FINTECH.

FINANCIAL SERVICES, HEALTHCARE, AND INSURANCE VERTICALS

Three verticals in the BSIMM operate in highly regulated industries: financial services, healthcare, and insurance (see Figure 5). In our decade-long experience with the BSIMM, we've seen large financial services firms reacting to regulatory pressures by starting software security initiatives much earlier than insurance and healthcare firms.

Even as the number of financial services firms in the BSIMM data pool has grown significantly over the past five years, including a large influx into the BSIMM data pool of newly started initiatives, the financial services software security group average age at last assessment was 5.4 years versus 4.4 years for insurance and 4.2 years for healthcare. Time spent by financial services firms maturing their collective software security initiatives shows up clearly in the side-by-side comparison.

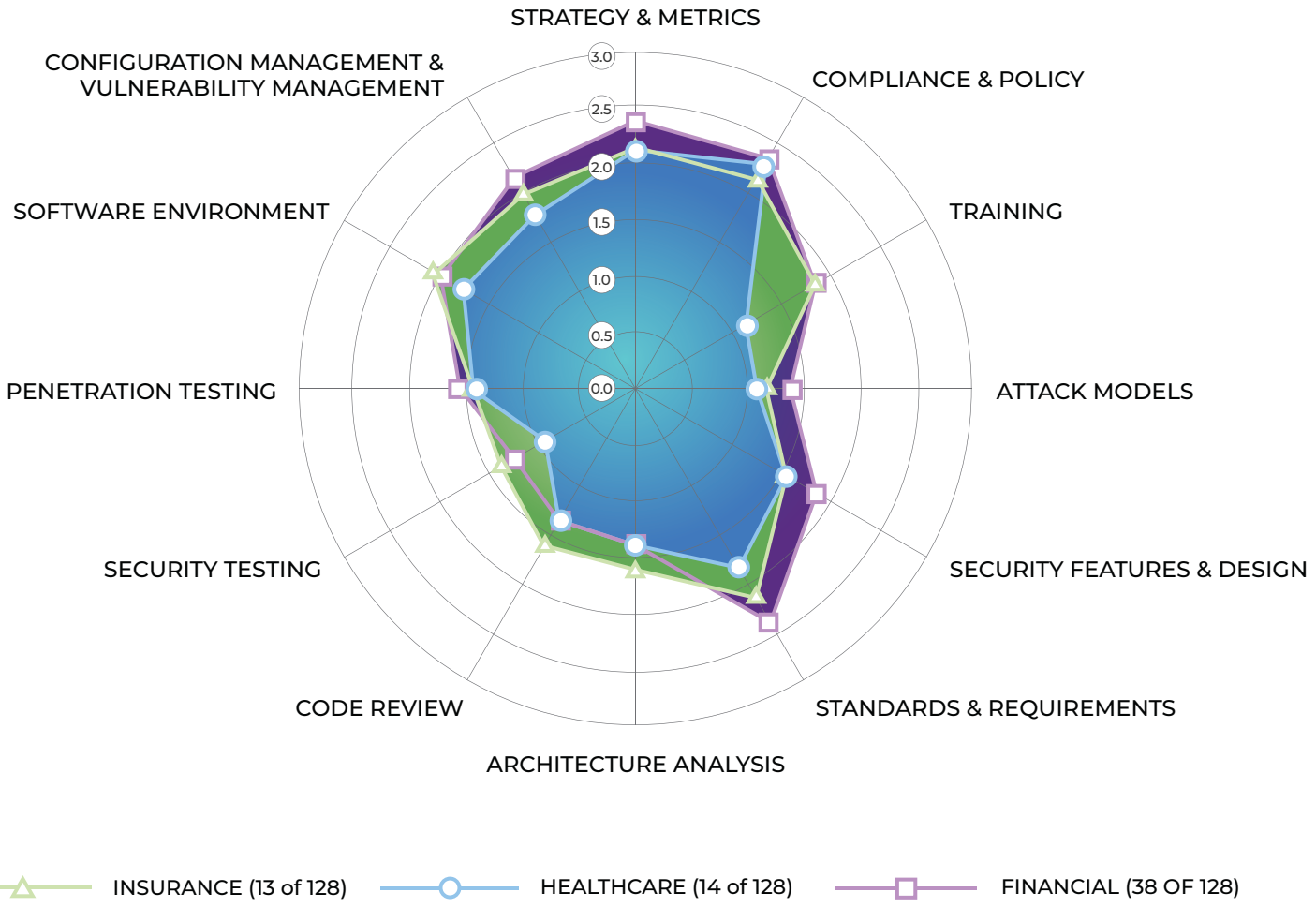


FIGURE 5. FINANCIAL VS. HEALTHCARE VS. INSURANCE.

Although organizations in the insurance vertical include some mature outliers, the data for these three regulated verticals show insurance still lags the others in the Strategy & Metrics, Attack Models, and Security Features & Design practices, while moving ahead in the Code Review and Security Testing practices. Compared against financial services firms, we see a similar contrast in healthcare, which nearly achieves par in Compliance & Policy, Architecture Analysis, Code Review, and Penetration Testing, but lags in other practices.

Despite the similarity of compliance and regulatory drivers across these three verticals, healthcare vertical high-water marks are generally trailing insurance and financial, possibly due to the need to prioritize patient care ahead of compliance activities.

EMERGING TRENDS IN BSIMM12 SOFTWARE SECURITY ACTIVITIES

LENDING RESOURCES, STAFF, AND KNOWLEDGE TO DevSecOps PRACTICES

Data collected over the past 24 months show a shift by software security groups away from mandating software security behaviors to more of a partnership role—providing resources, staff, and knowledge to DevSecOps practices with an objective to directly contribute security efforts to the critical path for software delivery.

For example, the observation rate for *use application containers to support security goals* increased from 14 observations in BSIMM10 to 44 observations in BSIMM12. The *use orchestration for containers and virtualized environments* also grew tremendously from BSIMM10 to BSIMM12—from only five observations to 33.

GOVERNANCE-AS-CODE

BSIMM10 and BSIMM11 data indicated that organizations were beginning the process of replacing manual, human-driven governance activities with automation. BSIMM12 observations increasingly indicate the sole source of software security standards and policy is becoming human-readable configuration code or simplified code that conducts vulnerability discovery—the essence of software-defined lifecycle governance.

CONTINUOUS DEFECT DISCOVERY AND CONTINUOUS IMPROVEMENT

BSIMM12 data indicate that more firms are implementing modern defect discovery approaches and favoring continuous monitoring and reporting rather than using a point-in-time defect discovery approach. Doing continuous testing initially requires additional human effort, which may explain the growth in satellite groups as measured by the Strategy and Metrics activity, *create or grow a satellite*. We have seen an average eight new observations across both the *monitor automated asset creation* and *automate verification of operational infrastructure security* activities over the past two years, and as this trend continues, we expect to see satellite groups implement even more automation.

The trends in continuous testing in both secure software development lifecycles and production environments increase the security issues to be triaged and processed. This new cadence is driving the need to provide data to leadership to support governance decisions as shown in the 30% growth of the BSIMM activity, *publish data about software security internally*, over the past two years. While governance processes remain mostly manual today, more organizations are trending toward governance-as-code, *integrate software-defined lifecycle governance*, as observed in 15% of the firms measured for BSIMM12.

CONTINUOUS SECURE SOFTWARE DEVELOPMENT LIFECYCLE IMPROVEMENT

When conducting smaller testing activities earlier and continuously, organizations are learning that security telemetry must be passed from one lifecycle phase to the next, just as the software artifacts themselves pass from one lifecycle phase to the next. This ongoing trend is reflected in two new activities, *automate verification of operational infrastructure security*, introduced in BSIMM10, and *implement event-driven security testing in automation*, introduced in BSIMM11.

The data indicate that continuous improvement is reflected in important secure software development lifecycle efforts including satellite outreach observations, automated asset discovery processes, incremental secure design efforts, conversion of manual efforts to “as-code,” and an increase in finding security and quality issues as soon as possible to facilitate development velocity. The imperative to identify issues as early as possible remains, driving the need to decompose big testing events into smaller, timely checks. But there is also a growing realization among software security groups that sometimes deployment orchestration or the post-deployment environment reflects the earliest, best opportunity for some tests.

This shift to continuous effort is reflected in the level of granularity at which governance is applied (for example, assigning application owners), at which testing is applied (SAST at a gate), and because it’s just easier to build a software inventory that way. There’s a great deal more code today that is not being used in traditional applications and that can easily escape current asset management efforts.

SECURITY AS RESILIENCE AND QUALITY

The observation rate of the activity *integrate opaque-box security tools into the QA process* increased by more than 50% over the past two years. Similarly, the observation rate of the activity *include security tests in QA automation* also more than doubled over the past two years.

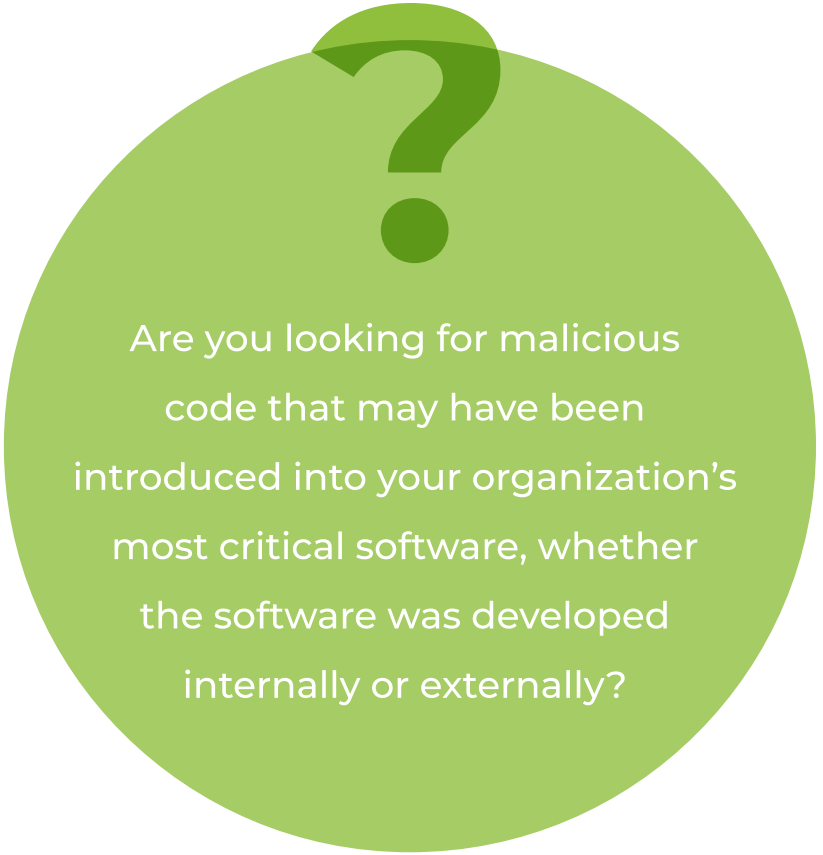
BSIMM participants have dramatically improved functional quality assurance practices, with resilience practices achieving impressive maturity within some engineering-led organizations. Organizations have also added engineering processes like A/B testing, chaos engineering, rollback mechanisms, and others in pursuit of resilience.

Many security efforts can fit naturally within quality assurance practices for engineering-led initiatives. While security testing tools such as SAST, SCA, and DAST have traditionally been implemented as out-of-band activities executed by software security groups, the opportunity exists for those activities to be integrated into quality assurance automation.

Software security fits naturally within resilience practices as well. We observed engineering-led security initiatives building the capability to provide resilience not only in the face of reliability and scalability challenges but also in the face of attack.

This effort requires capabilities focused on inventorying software, creating a software bill of materials and understanding how the software was built, configured, and deployed, and on the organization's ability to re-deploy based on security telemetry. An inventory detailing the components, dependencies, configurations, external services, and so on for all production software helps organizations tighten their security posture.

Demonstrating that many organizations have taken to heart the need for a comprehensive up-to-date software bill of materials, the BSIMM activity related to those capabilities—*enhance application inventory with operations bill of materials*—increased dramatically over the past two years.



OTHER TAKEAWAYS

HIGH-PROFILE RANSOMWARE AND SOFTWARE SUPPLY CHAIN DISRUPTIONS ARE DRIVING INCREASED ATTENTION ON SOFTWARE SECURITY

Over the past two years, the data show a 61% increase in the *identify open source* activity and a 57% increase in the *create SLA boilerplates* activity among participant organizations. Given media reports that attacks are succeeding more often, perhaps we will also begin to see an increase in malicious code detection efforts, an activity currently observed infrequently.

Increased executive attention, likely combined with engineering-driven efforts, has also resulted in organizations developing their own capabilities for managing cloud security and evaluating their shared responsibility models. There was an average of 36 new observations over the past two years across activities typically related to cloud security. However, despite increased executive attention, we have yet to see a corresponding increase in direct executive-level education and preparation.

"SHIFT LEFT" PROGRESSES TO "SHIFT EVERYWHERE" TO BETTER MANAGE RISK

If "shift left" focuses on moving security testing earlier in the development process, "shift everywhere" extends the idea to making security testing continuous throughout the software lifecycle. This means that smaller, faster, pipeline-driven security tests are conducted at the earliest opportunity, which might be in design but might be all the way over in production.

The move from maintaining traditional operational inventories toward automated asset discovery and creating a bill of materials includes adding "shift everywhere" activities such as using containers to enforce security controls, orchestration, and scanning infrastructure as code. Increased BSIMM observation rates of activities such as *enhance application inventory with operations bill of materials, use orchestration for containers and virtualized environments*, and *monitor automated asset creation* all demonstrate this trend.

ARE YOU SHIFTING EVERYWHERE?

1. Does your software security initiative favor smaller, phased-based secure software development lifecycle security activities over larger, pass-fail release gates?
2. Are you automating security decisions to remove time-consuming manual review and move toward an auditable governance-as-code secure software development lifecycle?
3. Are you following a "shift everywhere" strategy to move from large, time-consuming security tests to smaller, faster, pipeline-driven security tests conducted to improve engineering team performance?

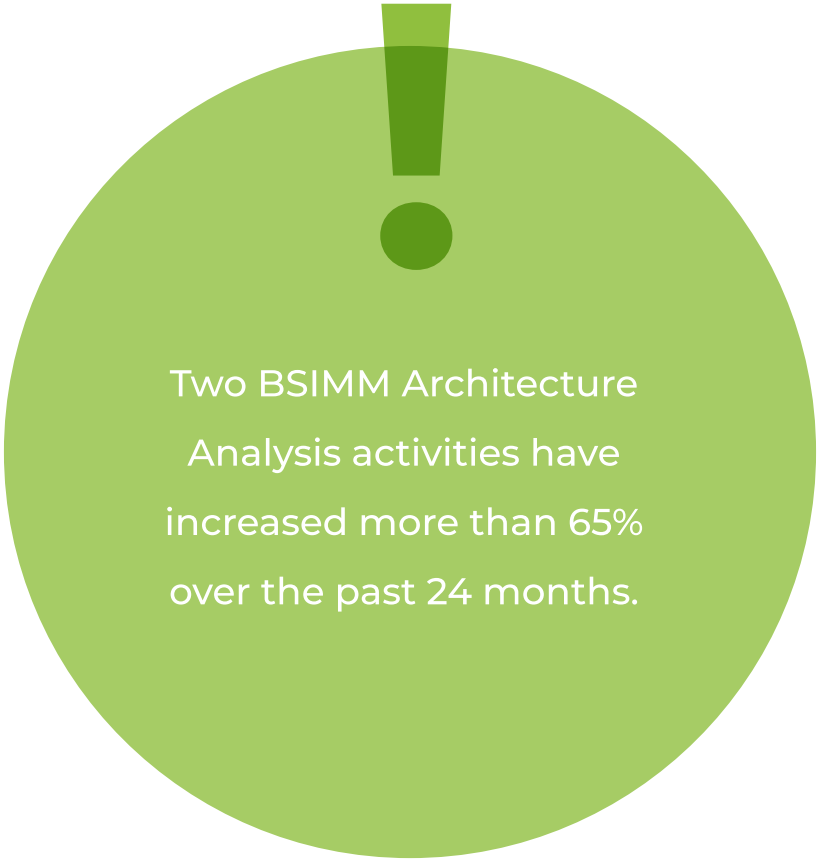
ORGANIZATIONS ARE LEARNING HOW TO TRANSLATE RISK INTO NUMBERS

We saw organizations exerting more effort to collect and publish their software security initiative data, demonstrated by a 30% increase of the *publish data about software security internally* activity over the past 24 months.

Firms that have begun to incorporate governance-as-code are finding that building software security metrics first is critical to transforming governance ideals into success.

ARCHITECTURE ANALYSIS AND DESIGN REVIEWS OF HIGH-RISK APPLICATIONS ARE BECOMING MORE COMMON

The observation rates of such activities as *performing design reviews of high-risk applications* and *defining/using an Architecture Analysis methodology* increased by more than 65% in the past 24 months.



CONCLUSION AND RECOMMENDATIONS

Whether you're in the process of creating a software security initiative or already maintaining a mature program, you should either have implemented or should be considering the following key actions:

- USE SECURITY TESTING TELEMETRY WHENEVER POSSIBLE. Gather data such as what testing was performed and what issues were discovered to drive improvements in your software development lifecycle and governance processes. Collect data at various stages of the lifecycle and use that data to create and enforce software security policies.
- MOVE TOWARD AUTOMATING SECURITY DECISIONS. The goal should be auditable governance-as-code that moves security practices and adherence to compliance away from a manual approach to a more consistent, efficient, and repeatable automated approach.
- CREATE A COMPREHENSIVE SOFTWARE INVENTORY. This inventory should include a software bill of materials of 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 bill of materials manually, maintaining one requires a significant investment of engineering and automation time. A bill of materials 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.
- IMPLEMENT SMALL, PHASED, SECURITY ACTIVITIES THROUGHOUT THE SOFTWARE DEVELOPMENT LIFECYCLE. Rather than using large, slow pass/fail gates that delay pipeline progress, think small.
- PUT AUTOMATED SECURITY TOOLS INTO PLACE. These tools can identify and help you remediate defects, vulnerabilities, and malicious code in your organization's critical software, whether that software was developed in-house, is commercial third-party software, or is open source.

For those 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 and identify the gaps between where you are today and where you need to go. Then build out your action plan, using the BSIMM12 results as a comparative yardstick that measures the key software security activities that your peer organizations have undertaken.

IS YOUR SECURITY INITIATIVE KEEPING PACE WITH CHANGE?

- **Do you use data-driven evidence—expressed as telemetry, measurements, metrics, KPIs, KRIs, OKRs, etc.—to formulate and track the success of software security investments?**
- **Does your software security initiative account for the impact on software security caused by disciplines such as cloud security, container security, orchestration security, source content management security, development pipeline security, and shared responsibility models?**

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

ADVANCING SECURITY AWARENESS AND ADOPTION WITH THE BSIMM

Since 1997, Genetec Inc. has delivered innovative technologies, across a broad solutions portfolio encompassing security, intelligence, and operations. Tasked with building out a software security program, Mathieu Chevalier, Lead Security Architect at Genetec, set out to promote a culture where security came first. A critical problem Mathieu faced was to validate his strategy and foster trust in the software security initiative Genetec needed to build with proven methods and approaches.

Genetec used Synopsys' Building Security In Maturity Model (BSIMM) assessment to help Mathieu gain a clear picture of the company's current software security stance and to identify areas of growth for the company's software security initiative. BSIMM assessments offer software security groups a model and framework to test, measure, and benchmark their current AppSec activities. Based on the security programs of more than 120 organizations, BSIMM data provide insight into the key activities, practices, and tools software security teams should consider implementing in their own organization.

GENETEC BEGAN WORK WITH THE BSIMM IN DECEMBER 2016, AND THE COMPANY HAS HAD TWO ASSESSMENTS IN THE PAST FIVE YEARS

Mathieu says his original decision to perform a BSIMM assessment on his organization's fledgling security program provided critical third-party insights. "The BSIMM helped us assess our product security initiative and guide us to where we should go," he notes. "It is a valuable tool for anyone building a product security program."

More recently, Genetec participated in a second assessment to gain insight into its improvements and areas that continue to present pain points. The second assessment has helped support new initiatives Genetec is now putting in place by demonstrating the benefits to the security team as well as the organization as a whole.

> "The BSIMM helped us assess our product security initiative and guide us to where we should go. It is a valuable tool for anyone building a product security program."

Genetec™

THE BSIMM ASSESSMENT AS THE FOUNDATION OF A SECURITY PROGRAM

One of the world's leading technology companies has been a BSIMM member for over a decade. "The BSIMM is really the foundation of our security champion and training program," notes its senior manager with the product security team. "The BSIMM assessment has been instrumental to our success, showing us areas of strength—and areas where our software security initiative needs to mature. The BSIMM is a perfect means to help evaluate and build out the maturity of our security program."

SECURITY CHAMPIONS AND TRAINING

"When I joined the company a little over three years ago, I formalized the champion program, creating the right framework for enrollment, getting champions trained as needed, and continually supporting them through engagements. We currently have more than 800 security champions," says the product security manager.

"I'm also responsible for security training. We've built a progressive 'Security Belt' training program that goes from 'yellow belt' (basic security awareness through learning) through 'green belt' (secure coding training and best practices for securing the software development lifecycle) and then into higher-level 'blue belt' activities (such as technologies that we leverage to build our products and applications, e.g., securing Kubernetes REST APIs).

"To strengthen the security champion program, we integrate knowledge and learnings from product security customer inquiries back into the program. By gathering this information, our security champions not only help support our customers in their inquiries but also understand our customer security needs. We use this information to help keep the security champions informed of what is happening in the industry and the role they play in enabling sales through security."

> "The BSIMM is a perfect means to help evaluate and build out the maturity of our security program."

WHAT DOES A BSIMM ASSESSMENT DO?

- **Enables you to communicate your software security posture to customers, partners, executives and regulators with independent assessment data. BSIMM helps your executives understand where you are in your journey and where you want to go when you're creating your strategy plan and budgeting.**
- **Assesses your level of maturity so you can evolve your software security journey in stages, first building a strong foundation, then undertaking more complex activities over time.**
- **Provides actual measurement data from the field. The BSIMM makes it possible to build a long-term plan for a software security initiative and track progress against that plan.**
- **Offers access to the BSIMM community. You can attend annual conferences and participate in a private online group to ask questions about your software security challenges and get direct, confidential feedback from your peers.**

ACKNOWLEDGEMENTS

Our thanks to the 128 executives, including those who wish to remain anonymous, from the software security initiatives we studied to create BSIMM12.

AARP
Adobe
Aetna
Alibaba
Ally Bank
Autodesk
Axway
Bank of America
Bell
Black Duck Software
Black Knight Financial Services
Canadian Imperial Bank of Commerce
Cisco
Citigroup
Depository Trust & Clearing Corporation
Eli Lilly
eMoney Advisor
EQ Bank
Equifax
F-Secure
Fannie Mae
Finastra
Freddie Mac
Genetec
Global Payments
HCA Healthcare
Highmark Health Solutions
Honeywell
HSBC
iPipeline
Johnson & Johnson
Landis+Gyr
Lenovo
MassMutual
McKesson
Medtronic
MediaTek
Morningstar
Navient
Navy Federal Credit Union
NCR
NEC Platforms
NetApp
NewsCorp
NVIDIA
Oppo
PayPal
Pegasystems
Principal Financial Group
RB
SambaSafety
ServiceNow
Synopsys
TD Ameritrade
Teradata
The Home Depot
The Vanguard Group
Trainline
Trane
U.S. Bank
Veritas
Verizon Media

Our thanks also to the nearly 130 individuals who helped gather the data for the BSIMM.

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

It takes a lot of effort to schedule dozens of BSIMMs on a yearly basis, and we thank Maatia Rickard for helping to make it happen. In addition, we give a special thank you to Kathy Clark-Fisher, whose behind-the-scenes work keeps the BSIMM science project, conferences, and community on track.

BSIMM12 was authored by Eli Erlikhman, Jacob Ewers, Sammy Migues, and Kevin Nassery.

PART THREE

APPENDIX

APPENDIX

THE BSIMM FRAMEWORK

BSIMM12 is organized as a set of 122 activities in a software security framework. The framework includes 12 practices that are organized into four domains, as shown in Table A.

DOMAINS			
 GOVERNANCE	 INTELLIGENCE	 SSDL TOUCHPOINTS	 DEPLOYMENT
Practices that help organize, manage, and measure a software security initiative. Staff development is also a central governance practice.	Practices that result in collections of corporate knowledge used in carrying out software security activities throughout the organization. Collections include both proactive security guidance and organizational threat modeling.	Practices associated with analysis and assurance of particular software development artifacts and processes. All software security methodologies include these practices.	Practices that interface with traditional network security and software maintenance organizations. Software configuration, maintenance, and other environment issues have direct impact on software security.
PRACTICES			
GOVERNANCE	**INTELLIGENCE**	**SSDL TOUCHPOINTS**	**DEPLOYMENT**
1. Strategy & Metrics (SM) 2. Compliance & Policy (CP) 3. Training (T)	4. Attack Models (AM) 5. Security Features & Design (SFD) 6. Standards & Requirements (SR)	7. Architecture Analysis (AA) 8. Code Review (CR) 9. Security Testing (ST)	10. Penetration Testing (PT) 11. Software Environment (SE) 12. ConfigurationManagement & Vulnerability Management (CMVM)

TABLE A. THE SOFTWARE SECURITY FRAMEWORK. *Twelve practices align with the four high-level domains.*

THE BSIMM SKELETON

GOVERNANCE		
STRATEGY & METRICS (SM)	**COMPLIANCE & POLICY (CP)**	**TRAINING (T)**
LEVEL 1	**LEVEL 1**	**LEVEL 1**
• [SM1.1] Publish process and evolve as necessary. • [SM1.3] Educate executives on software security. • [SM1.4] Implement lifecycle instrumentation and use to define governance.	• [CP1.1] Unify regulatory pressures. • [CP1.2] Identify PII obligations. • [CP1.3] Create policy.	• [T1.1] Conduct software security awareness training. • [T1.7] Deliver on-demand individual training. • [T1.8] Include security resources in onboarding.
LEVEL 2	**LEVEL 2**	**LEVEL 2**
• [SM2.1] Publish data about software security internally and drive change. • [SM2.2] Verify release conditions with measurements and track exceptions. • [SM2.3] Create or grow a satellite. • [SM2.6] Require security sign-off prior to software release. • [SM2.7] Create evangelism role and perform internal marketing.	• [CP2.1] Build PII inventory. • [CP2.2] Require security sign-off for compliance-related risk. • [CP2.3] Implement and track controls for compliance. • [CP2.4] Include software security SLAs in all vendor contracts. • [CP2.5] Ensure executive awareness of compliance and privacy obligations.	• [T2.5] Enhance satellite through training and events. • [T2.8] Create and use material specific to company history. • [T2.9] Deliver role-specific advanced curriculum.
LEVEL 3	**LEVEL 3**	**LEVEL 3**
• [SM3.1] Use an internal tracking application with portfolio view. • [SM3.2] SSI efforts are part ofexternal marketing. • [SM3.3] Identify metrics and use them to drive resourcing. • [SM3.4] Integrate software-defined lifecycle governance.	• [CP3.1] Create a regulator compliance story. • [CP3.2] Impose policy on vendors. • [CP3.3] Drive feedback from software lifecycle data back to policy.	• [T3.1] Reward progression through curriculum. • [T3.2] Provide training for vendors and outsourced workers. • [T3.3] Host software security events. • [T3.4] Require an annual refresher. • [T3.5] Establish SSG office hours. • [T3.6] Identify new satellite members through observation.

TABLE B. THE BSIMM SKELETON. *Within the SSF, the 122 activities are organized across different levels.*

INTELLIGENCE		
ATTACK MODELS (AM)	**SECURITY FEATURES & DESIGN (SFD)**	**STANDARDS & REQUIREMENTS (SR)**
LEVEL 1	**LEVEL 1**	**LEVEL 1**
• [AM1.2] Create a data classification scheme and inventory. • [AM1.3] Identify potential attackers. • [AM1.5] Gather and use attack intelligence.	• [SFD1.1] Integrate and deliver security features. • [SFD1.2] Engage the SSG with architecture teams.	• [SR1.1] Create security standards. • [SR1.2] Create a security portal. • [SR1.3] Translate compliance constraints to requirements.
LEVEL 2	**LEVEL 2**	**LEVEL 2**
• [AM2.1] Build attack patterns and abuse cases tied to potential attackers. • [AM2.2] Create technology-specific attack patterns. • [AM2.5] Maintain and use a top *N* possible attacks list. • [AM2.6] Collect and publish attack stories. • [AM2.7] Build an internal forum to discuss attacks.	• [SFD2.1] Leverage secure-by-design components and services. • [SFD2.2] Create capability to solve difficult design problems.	• [SR2.2] Create a standards review board. • [SR2.4] Identify open source. • [SR2.5] Create SLA boilerplate.
LEVEL 3	**LEVEL 3**	**LEVEL 3**
• [AM3.1] Have a research group that develops new attack methods. • [AM3.2] Create and use automation to mimic attackers. • [AM3.3] Monitor automated asset creation.	• [SFD3.1] Form a review board or central committee to approve and maintain secure design patterns. • [SFD3.2] Require use of approved security features and frameworks. • [SFD3.3] Find and publish secure design patterns from the organization.	• [SR3.1] Control open source risk. • [SR3.2] Communicate standards to vendors. • [SR3.3] Use secure coding standards. • [SR3.4] Create standards for technology stacks.

TABLE B. THE BSIMM SKELETON. *Within the SSF, the 122 activities are organized across different levels.*

SSDL TOUCHPOINTS		
ARCHITECTURE ANALYSIS (AA)	**CODE REVIEW (CR)**	**SECURITY TESTING (ST)**
LEVEL 1	**LEVEL 1**	**LEVEL 1**
• [AA1.1] Perform security feature review. • [AA1.2] Perform design review for high-risk applications. • [AA1.3] Have SSG lead design review efforts. • [AA1.4] Use a risk methodology to rank applications.	• [CR1.2] Perform opportunistic code review. • [CR1.4] Use automated tools. • [CR1.5] Make code review mandatory for all projects. • [CR1.6] Use centralized reporting to close the knowledge loop. • [CR1.7] Assign tool mentors.	• [ST1.1] Ensure QA performs edge/boundary value condition testing. • [ST1.3] Drive tests with security requirements and security features. • [ST1.4] Integrate opaque-box security tools into the QA process.
LEVEL 2	**LEVEL 2**	**LEVEL 2**
• [AA2.1] Define and use AA process. • [AA2.2] Standardize architectural descriptions.	• [CR2.6] Use automated tools with tailored rules. • [CR2.7] Use a top *N* bugs list (real data preferred).	• [ST2.4] Share security results with QA. • [ST2.5] Include security tests in QA automation. • [ST2.6] Perform fuzz testing customized to application APIs.
LEVEL 3	**LEVEL 3**	**LEVEL 3**
• [AA3.1] Have engineering teams lead AA process. • [AA3.2] Drive analysis results into standard design patterns. • [AA3.3] Make the SSG available as an AA resource or mentor.	• [CR3.2] Build a capability to combine assessment results. • [CR3.3] Create capability to eradicate bugs. • [CR3.4] Automate malicious code detection. • [CR3.5] Enforce coding standards.	• [ST3.3] Drive tests with risk analysis results. • [ST3.4] Leverage coverage analysis. • [ST3.5] Begin to build and apply adversarial security tests (abuse cases). • [ST3.6] Implement event-driven security testing in automation.

TABLE B. THE BSIMM SKELETON. *Within the SSF, the 122 activities are organized across different levels.*

DEPLOYMENT		
PENETRATION TESTING (PT)	SOFTWARE ENVIRONMENT (SE)	CONFIGURATION MANAGEMENT & VULNERABILITY MANAGEMENT (CMVM)
LEVEL 1	LEVEL 1	LEVEL 1
• [PT1.1] Use external penetration testers to find problems. • [PT1.2] Feed results to the defect management and mitigation system. • [PT1.3] Use penetration testing tools internally.	• [SE1.1] Use application input monitoring. • [SE1.2] Ensure host and network security basics are in place.	• [CMVM1.1] Create or interface with incident response. • [CMVM1.2] Identify software defects found in operations monitoring and feed them back to development.
LEVEL 2	LEVEL 2	LEVEL 2
• [PT2.2] Penetration testers use all available information. • [PT2.3] Schedule periodic penetration tests for application coverage.	• [SE2.2] Define secure deployment parameters and configurations. • [SE2.4] Protect code integrity. • [SE2.5] Use application containers to support security goals. • [SE2.6] Ensure cloud security basics. • [SE2.7] Use orchestration for containers and virtualized environments.	• [CMVM2.1] Have emergency response. • [CMVM2.2] Track software bugs found in operations through the fix process. • [CMVM2.3] Develop an operations inventory of software delivery value streams.
LEVEL 3	LEVEL 3	LEVEL 3
• [PT3.1] Use external penetration testers to perform deep-dive analysis. • [PT3.2] Customize penetration testing tools.	• [SE3.2] Use code protection. • [SE3.3] Use application behavior monitoring and diagnostics. • [SE3.6] Enhance application inventory with operations bill of materials.	• [CMVM3.1] Fix all occurrences of software bugs found in operations. • [CMVM3.2] Enhance the SSDL to prevent software bugs found in operations. • [CMVM3.3] Simulate software crises. • [CMVM3.4] Operate a bug bounty program. • [CMVM3.5] Automate verification of operational infrastructure security. • [CMVM3.6] Publish risk data for deployable artifacts. • [CMVM3.7] Streamline incoming responsible vulnerability disclosure.

TABLE B. THE BSIMM SKELETON. *Within the SSF, the 122 activities are organized across different levels.*

THE BSIMM ONLINE COMMUNITY

The BSIMM Online Community is a unique, members-only forum that helps you address software security challenges in today's complex business environments.

Firms that have completed a BSIMM assessment have access to the members-only BSIMM community web site.

As a member, you:

- Receive regular blog and discussion posts that show best practices, tips, and case studies.
- Bounce ideas and questions off of the 700-member community.
- Attend exclusive conferences.

From content authored by industry leaders to hands-on interactions with fellow BSIMM members, it is a powerful resource for collaborative problem solving, thought leadership, and access to valuable resources not available anywhere else.

Find out how to unlock access to an engaged BSIMM member community, including conferences, newsletters, and original content.

www.bsimm.com