451 Research® | PATHFINDER REPORT

Designing a Modern Application Security Program

COMMISSIONED BY

SYNOPSYS®

OCTOBER 2019

About this paper

A Pathfinder paper navigates decision-makers through the issues surrounding a specific technology or business case, explores the business value of adoption, and recommends the range of considerations and concrete next steps in the decision-making process.

ABOUT THE AUTHOR

DAN KENNEDY

RESEARCH DIRECTOR, VOICE OF THE ENTERPRISE: INFORMATION SECURITY

Daniel Kennedy is the Research Director for Information Security for 451 Research's Voice of the Enterprise (VoTE) quantitative research product, where he is responsible for managing all phases of the research process. He is an experienced information security professional who has written for both Forbes online and Ziff Davis, has provided commentary to numerous news outlets including The New York Times and The Wall Street Journal, and his personal blog Praetorian Prefect was recognized as one of the top five technical blogs in information security by the RSA 2010 Conference.

Introduction

'Software is eating the world' has become nearly cliché in the years since Marc Andreessen first wrote it in a 2011 Wall Street Journal article. But the well-worn saying is still relevant today, and even served as the lead for the keynote address by Dino Dai Zovi, mobile security lead at Square, at the Black Hat 2019 conference. In his keynote, Dai Zovi emphasized the need for embedding information security into software development processes and automating those processes as much as possible. "As software is eating the world, every company is becoming a software company," he stated. "This doesn't mean that every company is shipping software products; it means that services and products in every field are becoming increasingly driven, powered and differentiated by software."

Background

Application development has become the key differentiator for many organizations' technology teams. The question is, how do information security teams support development teams with the tools needed to reduce vulnerabilities without interfering with developers' delivery-oriented priorities?

The penalties of not doing so are evident: some 62% of hacking incidents analyzed in Verizon's 2019 Data Breach Investigations Report were against web applications. Attackers used fairly-low-tech methods, such as stolen credentials, in the lion's share of these incidents. But attacks leading to prominent breaches in the last few years included an exploit targeting an open source library vulnerability that had disastrous results for Equifax and a script-injection attack against an online payment system that resulted in a record-breaking GDPR fine against British Airways.

Both breaches serve as clear case studies detailing the consequences of an inadequate approach to application security. Organizations cannot rely on traditional network- and infrastructure-based security protections as they once did; they need to build protections into applications as well as fortify them against attack.

Modern application development is characterized by iterative approaches that quickly layer in changes. The lengthy requirements-gathering and prototyping processes associated with a waterfall-style software development lifecycle (SDLC) have all but disappeared. The DevOps methodology introduced the concept of more closely aligning development and operations, breaking down silos that had formed around the teams responsible for moving code from creation through test and production. The concept is not particularly new; agile as a development methodology was introduced around 2001, DevOps around 2009. However, their increasing adoption has superheated the development process in many organizations.

Legacy process approaches to application security, such as scanning entire codebases or scanning websites ad hoc, do not offer full coverage against this pace of change. Application security practitioners – increasingly a combination of developers and security professionals – have to find ways to integrate incremental application security testing as new code is built, tested and released.

Components of a Holistic Application Security Program

Some manner of application security testing (AST) is present in 37% of organizations, with an additional 19% reporting their intention to implement AST over the next year, according to 451's Voice of the Enterprise (VotE) end-user research, which surveys senior security professionals in organizations throughout the world. That percentage of organizations using AST tools rises to 41% among large enterprises.

A prior study by Synopsys and 451 Research revealed that when enterprises have in-house application developers writing code for internal and external applications, the usage rates of both dynamic and static application security testing balloon to more than 80%. Overall, approximately 9% of security budgets are allocated to application security, according to VotE studies of security budgeting.

Figure 1: Application security testing usage within enterprises

Source: 451 Research's Voice of the Enterprise: Information Security, Workloads and Key Projects 2018

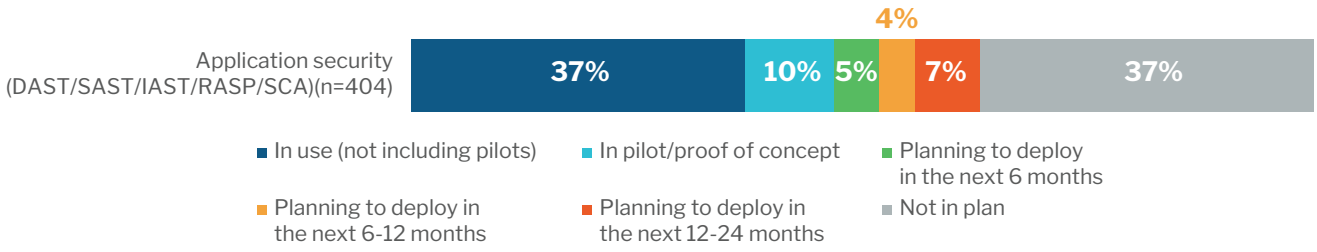
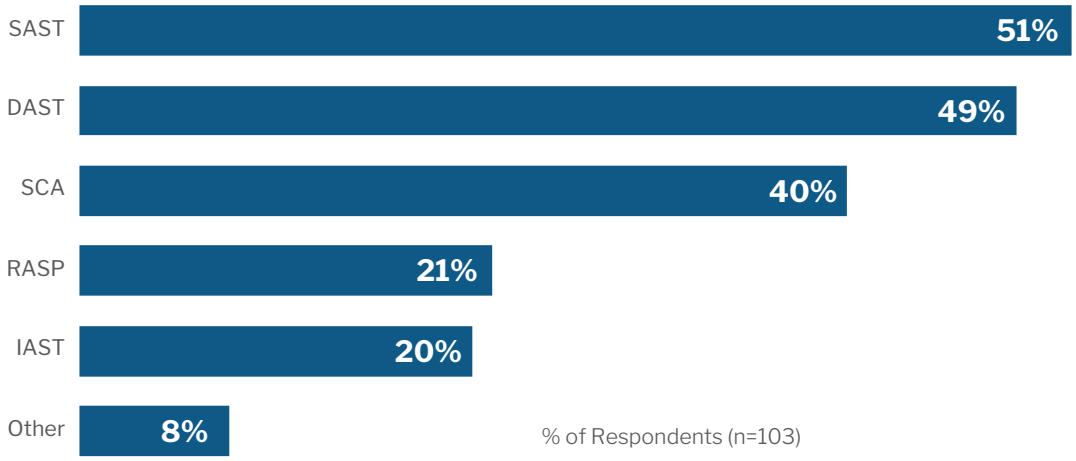


Figure 2: Application security feature sets purchased from vendor

(Respondents whose organization uses application security vendor in any capacity)
Source: 451 Research's Voice of the Enterprise: Information Security, Vendor Evaluations 2018



Static Application Security Testing: Shifting Left into the Developer Environment

Static application security testing (SAST) tools examine either source code or compiled binaries to identify security vulnerabilities, including problems such as unsafe function use, race conditions, buffer overflows and input validation errors that allow for attacks such as SQL injection. Of all testing methods deployed to support enterprise application security programs, SAST is among the most common and can be applied close to the process of code creation. Key to an effective SAST offering is the ability to produce real-time analysis results that identify issues and provide actionable remediation advice in the integrated development environment (IDE), which allows developers to fix issues before committing code for a build. A SAST tool's ability to do incremental scans only on changed code, coupled with accurate and comprehensive analysis results, greatly improves developer productivity.

Catching and fixing code issues early in the development stage enables significant time and resource savings by preventing the propagation of security vulnerabilities further downstream in the QA and dynamic testing stages of the SDLC. Further, baselining information from various scans, perhaps later in the process in a testbed, allows information security to determine the efficacy of vulnerability identification and remediation over time.

SAST vendors are increasingly building developer education components alongside their SAST offerings in the form of links to relevant e-learning courses triggered by issues identified in the code itself. Many college and university computer science programs and programming boot camps concentrate on teaching students how to turn requirements into code, without covering cybersecurity abuse cases, bad actors or even how inadvertent mistakes can create vulnerabilities that hackers can exploit. In response, application security vendors have started designing highly targeted training that keys off the mistakes identified in a SAST analysis. The efficiency of this type of training takes into account the constraints affecting developers, who aim to turn code around quickly and are likely to skip training whose scope is beyond what they immediately need.

Software Composition Analysis: When Most of your Code *Isn't* Your Code, How Do You Manage Risk Effectively?

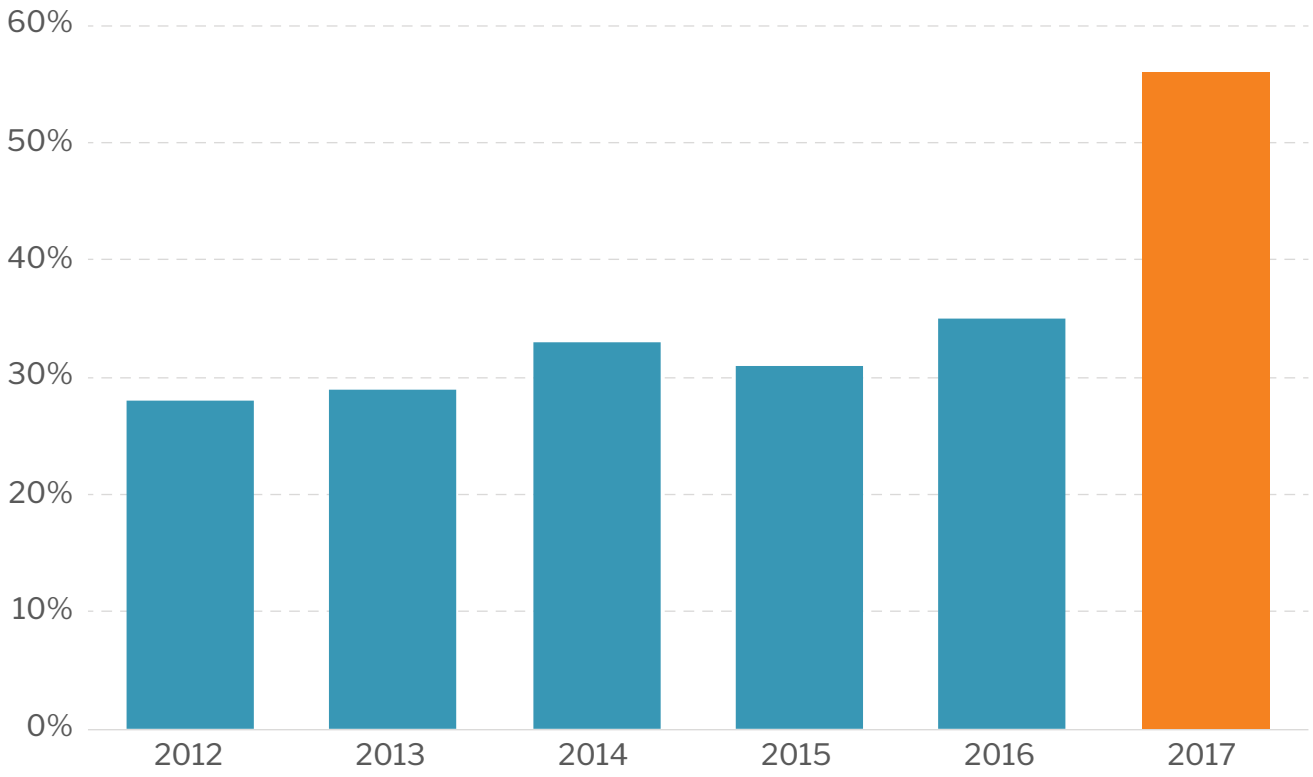
Software composition analysis (SCA), which can be implemented alongside SAST or as a stand-alone tool, has seen strong growth over the past year. At a basic level, SCA allows an organization to inventory its open source libraries, including what versions of an open source component are in use.

In 2018, 451's VotE Information Security study found SCA products in place in 11% of enterprises, with another 11% of respondents saying they were planning to implement SCA in the next 12 months. As predicted, 21% of respondents in 2019 stated they have SCA in place, with an additional 12% saying they're currently evaluating vendor offerings.

What's responsible for that growth? The first clear trend is the percentage of open source code composing modern applications. Developers, who are measured on speed and work in short iterative cycles, do not want to spend time building out functionality and components that are already available under open source licenses. This is not a new trend. According to the Open Source Security and Risk Analysis study of 2018, which looks at anonymized results from Synopsys' Black Duck Audits, the percentage of open source in modern applications reached nearly 60% in 2017.

Figure 3: Percentage of open source in applications

Source: Synopsys, Open Source Security and Risk Analysis, 2018

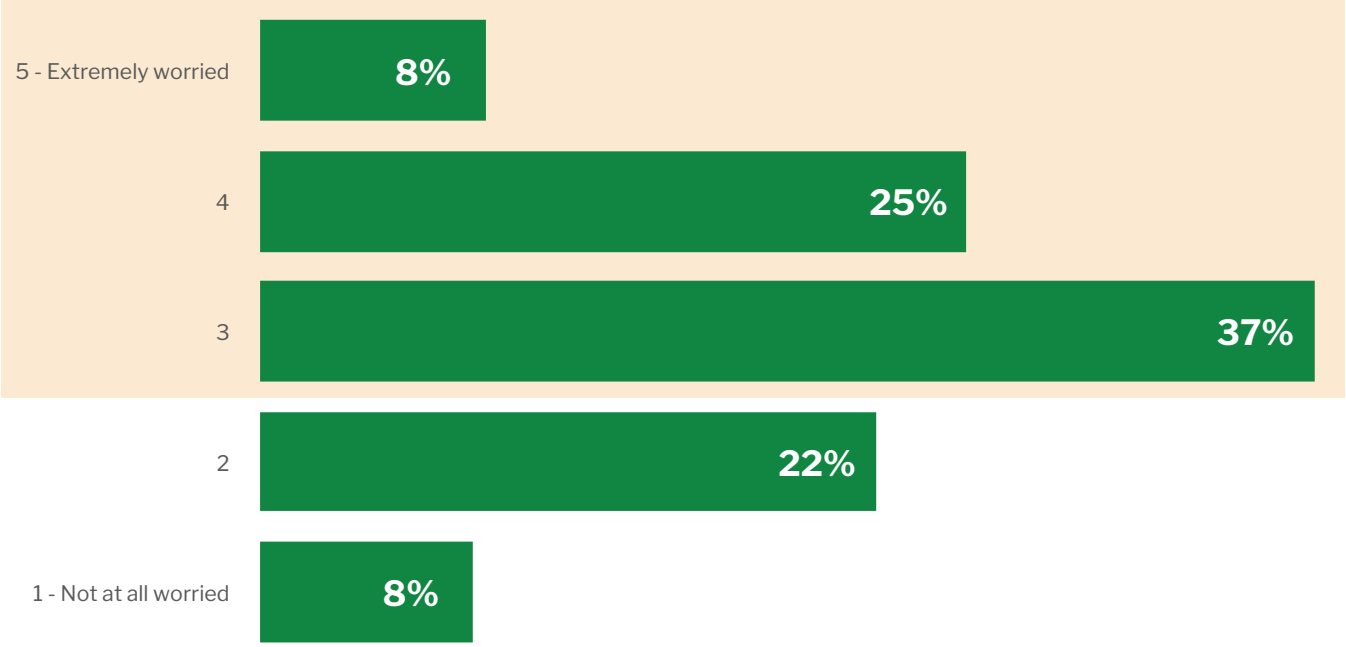


As mentioned earlier, a second major factor in the increased use of SCA is the breach that occurred at credit monitoring firm Equifax in 2017. The entry point for the breach was a well-publicized open source vulnerability for which a patch was available. Seventy percent of IT professionals responding to 451's 2018 VotE Digital Pulse study privately acknowledged that they might have been susceptible to the same type of vulnerability that provided the entry point into Equifax.

Figure 4: Could your organization be an Equifax?

Source: 451 Research's Voice of the Enterprise: Digital Pulse, Workloads and Key Projects 2018

Q: A number of factors contributed to the 2017 Equifax security breach, but the initial entry point was through an unpatched application vulnerability. How worried are you that your organization may be susceptible to a similar breach? Please answer using a 1 to 5 scale where 1 is 'not at all worried' and 5 is 'extremely worried.' (n=1,148)



The Equifax breach and the overall proliferation of open source use have given SCA adoption a tailwind. Organizations making heavy use of open source libraries typically have different versions of the same library used in different places, dated libraries and other inefficiencies. An SCA product can identify these problems, find and monitor inherent security vulnerabilities in open source libraries, and flag libraries with potential licensing issues.

More advanced SCA products warn of vulnerabilities beyond those found in public sources such as the National Vulnerability Database. The upper tier of SCA offerings can detect open source code fragments beyond declared libraries, automate policy enforcement for open source use, provide full remediation guidance, and even help automate a response when developers introduce unsafe libraries. Many SCA tools have begun shifting left, with integrations for development environments and workflows, similar to SAST tools. Some even provide in-browser guidance as developers explore potential open source components to meet a specific requirement.

Dynamic and Interactive Application Security Testing: Part of the CI/CD Workflow

Dynamic application security testing (DAST), one of the most common types of AST, essentially approaches an application the same way an attacker might, interacting with an application from the outside by sending requests and evaluating responses. Several stand-alone commercial and open source DAST tools are available, and some vulnerability management vendors have also introduced DAST-like offerings. However, one problem with DAST point-in-time testing is that mission-critical applications often update more often than tests can be conducted, leaving a potential gap in coverage.

Interactive application security testing (IAST) attempts to solve issues with legacy DAST approaches though the use of a passive agent that monitors application behavior and reports on the vulnerabilities it encounters as the application runs. IAST can either run as part of existing automated test cycles or simulate traffic to an application, as a DAST offering would, and monitor the application's response to a simulated attack. An IAST product might look at HTTP traffic, database queries, memory access, invocation of third-party libraries, external calls and file access, for example. Owing to agent instrumentation, IAST can clearly identify exploitable vulnerabilities and pinpoint their location in the application code. The automated nature of IAST scanning lends itself to passive scanning in build or test environments, allowing for tighter integration in the SDLC than legacy vulnerability-scanning-style approaches. A key consideration for organizations looking to adopt IAST is whether the IAST platform works with their existing technology stacks.

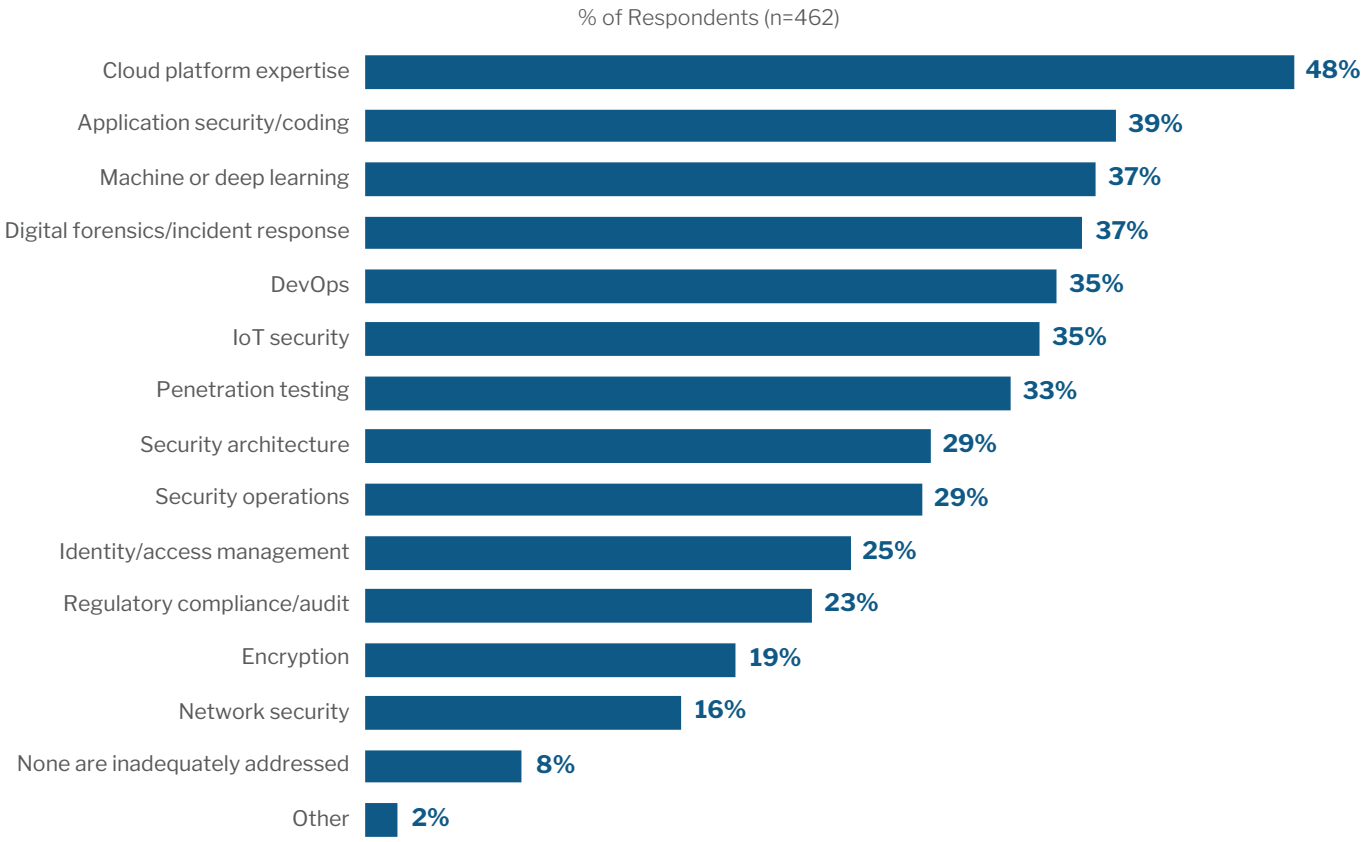
DevSecOps: Secure Your Applications at the Speed of DevOps with Automation and Integrations

Why Does Traditional Application Security Need to be Reimagined?

Having information security professionals running point-in-time scans and attempting to advise developers on their code is a difficult proposition even when information security teams have coding expertise – which, in most cases, they don't. When 451 asked respondents about the skill sets most inadequately addressed by their organizations' information security function, application security ranked behind only cloud platform expertise.

Figure 5: Security skill sets least addressed at organizations today

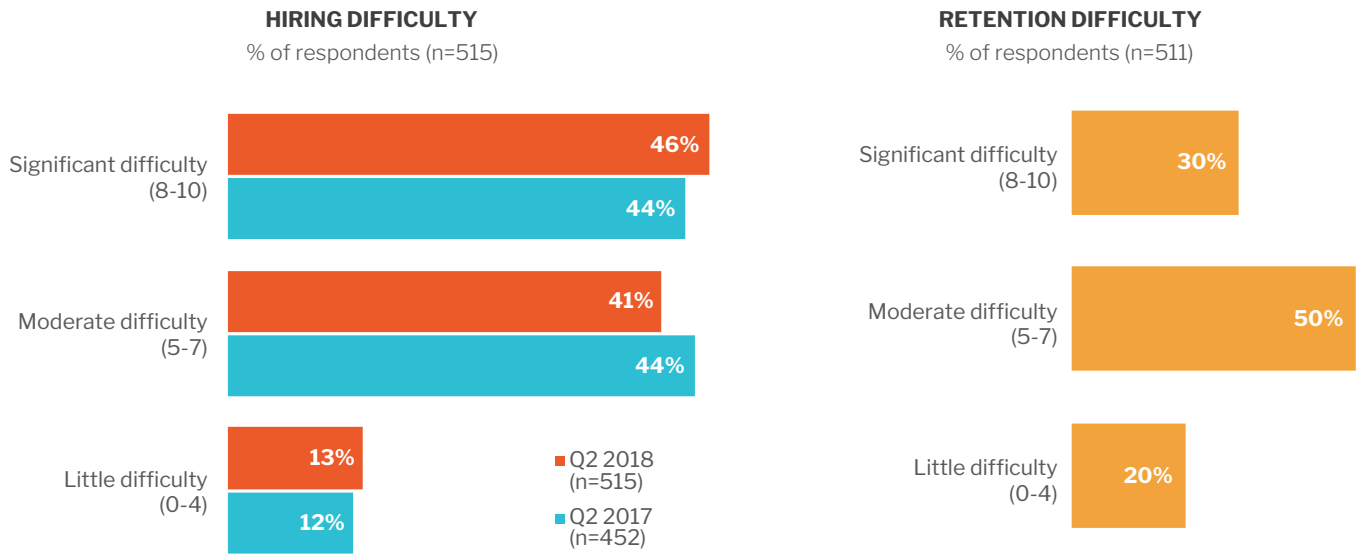
Source: 451 Research's Voice of the Enterprise: Information Security, Organizational Dynamics 2019



The simple fact is that at most organizations, there are more application developers than there are information security professionals. Sixty-six percent of organizations surveyed in 451's recent end-user survey on organizational dynamics in information security said their organizations do not have enough information security professionals. The highest percentage of respondents said that information security professionals are significantly difficult to hire and moderately difficult to retain.

Figure 6: Difficulty in hiring and retaining security professionals

Source: 451 Research's Voice of the Enterprise: Information Security, Organizational Dynamics 2018



This environment for security hires, alongside the iterative pace of change supported by modern application development teams, suggests that full coverage won't be achieved by on-demand scans run by security resources. Organizations can pull two levers in response. The first is shifting left by pushing some percentage of testing to developers. The second is automation in the SDLC, achieved by integration into DevOps and development toolchains.

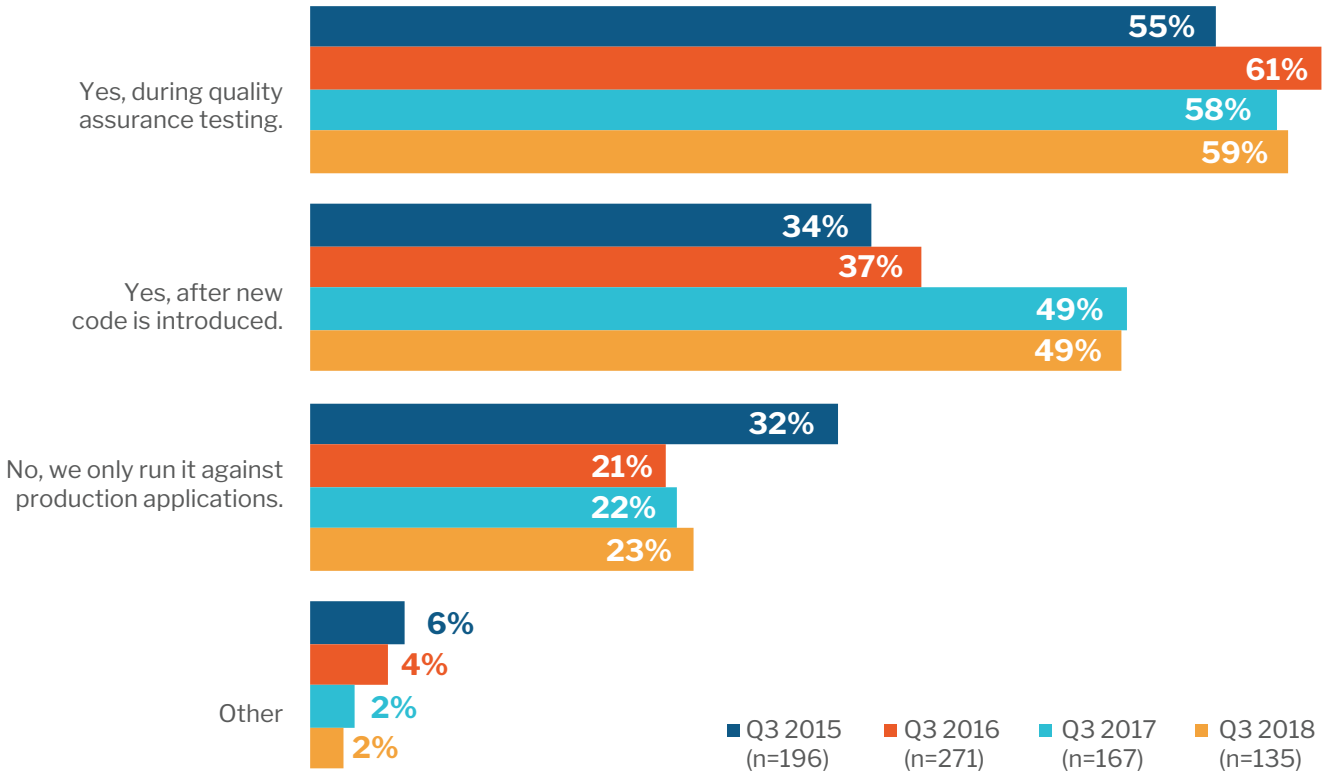
SHIFT LEFT

The prior section described the 'why' for DevSecOps. The 'how' is more complicated. The 'shift left' movement away from ad hoc testing at the end of a development cycle toward integrated testing throughout the SDLC has always made economic sense from an effort-versus-impact standpoint. Shifting left allows organizations to treat application security vulnerabilities as defects. A defect solved at the point of code creation is easiest to correct, whereas one that has escaped to the testing phase typically involves more people to remediate: developers to reengage the affected code, testers to retest. In the worst-case scenario, a defect that reaches production can involve technical support, the customer or user, and exposure of data or interruption of functionality by a bad actor.

Figure 7: Application security testing usage by SDLC phase

Source: 451 Research's Voice of the Enterprise: Information Security, Vendor Evaluations 2018

Q. Do you run your application security vendor's tools during different phases of the software development lifecycle (SDLC) as part of a secure SDLC?



Given the benefits of a shift-left testing approach, it is not surprising to see a four-year reduction from 32% to 23% in the percentage of teams running AST tools only in production. According to 451's end-user research, SAST caught up with DAST-style testing in 2017 as the most common method of AST implemented. In 2015, 34% of organizations ran SAST tools immediately after code was written. In the latest study, this percentage has risen to 49%.

AUTOMATE AND INTEGRATE

While information security has had a number of opportunities to shift left in the past, the standardization of DevOps toolchains offers more widespread automation of AST and a development culture more receptive to such automation – for example, using a DAST or IAST tool or performing a SAST scan via continuous integration tools such as Jenkins or Azure DevOps. The goal, which is to avoid switching out the tools that developers are already using, extends to issue tracking as well – for example, opening defect reports in Jira and sending alert messages through Slack.

Application developer enablement is necessary to make application security more project-oriented and integrated directly into the SDLC. Part of developer enablement involves integrating AST tools further into developer toolsets, as with the DevOps examples above. The most direct

approach is to integrate tools such as SAST directly into popular integrated development environments. For example, a SAST plug-in can check code on each save and limit the scope of scans to the code being worked on, representing a fairly-low-inhibitor approach to identifying security problems without derailing coding efforts. Some SCA tools can integrate similarly into the IDE to reveal open source dependencies that violate security and license compliance policies at the point developers introduce them. SCA browser-based plug-ins also allow developers to search for appropriate open source libraries to meet a requirement they're working on.

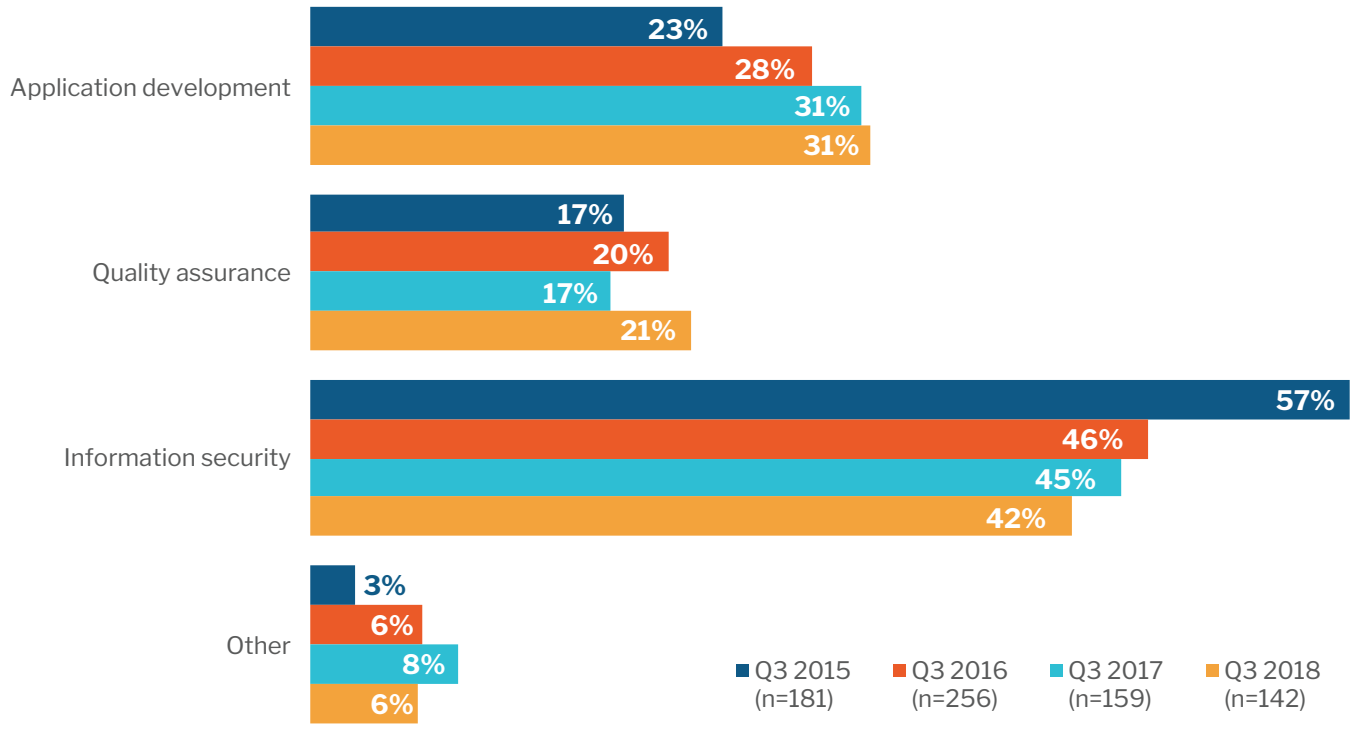
Roles and Responsibilities: Enable Developers to Address Application Security Risk Proactively

The presence of information security teams implies a certain scale of organization, trending toward medium-sized and larger organizations. These organizations typically have the greatest motivation for spending resources on application security; according to 451's end user research, they represent the largest percentage of users of AST tools.

451 asked survey respondents with AST tools in place to allocate 100 points of use across four teams: information security, application development, quality assurance and other. The researchers have asked the same question over a four-year period to discern differences over time in AST tool use. The figure below shows the results.

Figure 8: Application security testing usage by team

Source: 451 Research's Voice of the Enterprise: Information Security, Vendor Evaluations 2018



The trend line is self-evident. Though information security has remained the primary user over four years, that group's use of AST has shrunk from 57% to 42%. Application development's use of AST, by contrast, has risen over the same period from 23% to 31%.

The federation of certain day-to-day operational application security testing from information security to developers carries with it more complexity, better and more realistic coverage, and thus a greater need for coordination. Information security should embrace an ideal role as both application and process auditors, looking for security defects that have escaped the testing built into the SDLC and monitoring the efficacy of vulnerability identification and remediation overall, while passing project-level day-to-day issue identification and resolution to application development teams.

Security must include development leadership in the evaluation and rollout of AST tools while owning the primary motivation for implementation and, in many cases, the implementation budget. Security professionals are judged by their ability not only to resolve incidents but also to prevent them; developers, by their ability to release defect-free code that addresses business requirements quickly. It is only through the coordination of both teams' motivations and activities that an organization can build a sustainable long-term application security discipline into its culture.

A secondary aspect of the general shortage of security professionals, and of application security skills in particular, is the need to make up for this shortfall at the intersection of development and security expertise. Organizations can supplement missing skill sets in a number of ways, from making ad hoc requests for further information about identified vulnerabilities directly from an AST tool to relying on more traditional managed service or consultative support from AST vendors.

Conclusion

Different forms of application security testing have been available to enterprises for a while, but a handful of important trends are coalescing around the more direct integration of AST toolsets into the SDLC:

- A more iterative approach to application development that generally includes a larger number of builds and releases.
- An increased use of open source components in modern applications.
- DevOps toolchains standardizing and facilitating continuous integration techniques.
- Threats moving from network to application targets at a time when hybrid architectures are reducing the reliability of network-based defenses.

With these trends in mind, it seems that legacy approaches that rely heavily on DAST or penetration testing late in the SDLC or in production will not satisfy organizations' need for comprehensive analysis at the pace of modern development. To succeed, organizations need to shift application security left, with tools that allow them to integrate and automate security analysis in their DevOps toolchains, from the developer's desktop and throughout their CI/CD pipelines.

Synopsys helps development teams build secure, high-quality software, minimizing risks while maximizing speed and productivity. Synopsys, a recognized leader in application security, provides static analysis, software composition analysis, and dynamic analysis solutions that enable teams to quickly find and fix vulnerabilities and defects in proprietary code, open source components, and application behavior. With a combination of industry-leading tools, services, and expertise, only Synopsys helps organizations optimize security and quality in DevSecOps and throughout the software development life cycle. For more information, go to *www.synopsys.com/software*.

Coverity SAST. Coverity helps developers find and fix security defects early in the SDLC, with support for 20 languages and over 70 frameworks and template engines, as well as security checkers to help ensure compliance with *OWASP Top 10*, *CWE/SANS Top 25*, *PCI DSS*, and other standards. Coverity gives teams the flexibility to analyze code in the IDE and on the build server, on-premises, and in the cloud.

Black Duck SCA. Black Duck enables teams to secure and manage open source across their software supply chain. Black Duck's unique multifactor open source discovery technology accurately detects open source in source code, binaries, and container images, giving development, security, and legal teams complete visibility into their open source security and license compliance risks. In addition, integrated policy management allows teams to automate open source governance, so they can build fast while staying secure and compliant.

Seeker IAST. Seeker helps development, QA, and security teams automate application security testing with CI and test automation tools. Seeker is the only IAST solution that actively verifies that identified vulnerabilities are exploitable, using patented technology, reducing false positives to near zero. Its unique sensitive-data tracking feature automatically detects when user-designated sensitive data is exposed in logs, databases, or files.

Managed Security Testing. Synopsys Managed Security Testing Services deliver on-demand security testing performed by a team of security experts, helping organizations cost-effectively address complex test scenarios. Synopsys' Managed Penetration Testing combines testing tools and in-depth manual tests focusing on business logic to find vulnerabilities outside common standards, including authentication checks, access control testing, logging and monitoring, workflow bypass, and manual review to identify false positives.

Polaris Software Integrity Platform. Polaris brings Synopsys' tools together to provide a comprehensive, automated application security solution that enables teams to build secure software faster. The Code Sight IDE plugin integrates security analysis into the developer's desktop, while the Polaris central server gives security and development teams a single-pane-of-glass view of project vulnerability trends and helps them manage compliance with the security standards and regulations that are most important to their organization.

CONTENT PROVIDED BY:

SYNOPSYS®

About 451 Research

451 Research is a leading information technology research and advisory company focusing on technology innovation and market disruption. More than 100 analysts and consultants provide essential insight to more than 1,000 client organizations globally through a combination of syndicated research and data, advisory and go-to-market services, and live events. Founded in 2000 and headquartered in New York, 451 Research is a division of The 451 Group.

NEW YORK
Chrysler Building
405 Lexington Avenue,
9th Floor
New York, NY 10174
+1 212 505 3030

SAN FRANCISCO
505 Montgomery Street,
Suite 1052
San Francisco, CA 94111
+1 212 505 3030

LONDON
Paxton House
30, Artillery Lane
London, E1 7LS, UK
+44 (0) 203 929 5700

BOSTON
75-101 Federal Street
Boston, MA 02110
+1 617 598 7200

451 Research®