ESG Research Insights Paper

Application Security for Contemporary Software Development and Deployment

Cybersecurity in the Age of Agile, DevOps, and AppDev

By Doug Cahill, ESG Senior Analyst; and Jack Poller, ESG Analyst

May 2017

This ESG Study was commissioned by Veracode and is distributed under license from ESG.

Contents

Executive Summary

In the first half of 2017, Veracode commissioned Enterprise Strategy Group (ESG) to conduct a survey of 400 IT, cybersecurity, and application developer professionals with knowledge of, responsibility for, and involvement in the planning, implementation, and/or operations of their organization's application security requirements and testing procedures for internally developed software. Participants also were involved or familiar with their organizations' evaluation, selection, and purchasing of security testing tools for internally developed software.

Survey respondents were located in the US (51%), UK (24%), and Germany (25%), and were employed at enterprise-sized (1,000 or more employees, 98%) or large midmarket (500-999 employees, 2%) organizations. All organizations represented develop software internally, either for employee use (58%), resale (3%), or both (39%). The survey included representation from multiple industry verticals including IT (23%), financial services (banking, securities, and insurance, 21%), manufacturing (18%), retail/wholesale (8%), government (6%), business services (6%), communications and media (5%), and health care (4%), among others.

This research project was undertaken to evaluate the new application security (AppSec) ecosystem; the importance of application security and application security testing; how AppSec, Agile, and DevOps methodology and tools work together; the relationship between AppDev, AppSec, and DevOps teams; the use and adoption of static and dynamic application security testing, and composition analysis. Security decision makers and technical influencers within organizations operating internal application development teams were asked to detail their understanding of whether, how, and why application security is becoming a more important part of a security architecture. Based on the data collected from the research survey, this paper concludes:

- AppSec importance is trending. Efficiency and compliance are driving AppSec usage, but a dichotomy of competing priorities—code functionality versus security—indicates that AppDev and security teams may need to come to a better understanding that these are not mutually exclusive outcomes.
- Agile and DevOps adoption are in lockstep. Broad adoption of Agile methodologies is correlated with DevOps adoption. DevOps is viewed positively for both developer and security use cases. Organizations perceive increased value from the automation of AppSec via DevOps.
- Mobile- and cloud-first initiatives may be equalizing internal versus external AppSec priorities. While more organizations have conducted static testing for internal apps and dynamic testing for external apps, planned future deployments may equalize usage over time. Mobile- and cloud-first initiatives may be blurring the lines between internal and external applications.
- AppSec crosses organizational boundaries. Organizations realize efficiencies and enhanced security when developer and security teams collaborate throughout the software development lifecycle.
- AppSec needs to be operationalized. While developers grasp the benefits of AppSec, complexity, workflow integrations, and price create obstacles to greater adoption. There is a need to focus on operationalizing AppSec with DevOps, creating "DevSecOps" for greater efficiency.

Situational Analysis

Cybersecurity can be an intimidating discipline for most organizations. The threat landscape is becoming increasingly dangerous, as malicious actors focus their energy on developing sophisticated, targeted attacks. At the same time, mobile- and cloud-first initiatives, digital workplace transformation, and IoT applications are increasing the size and complexity of application infrastructures and their associated attack surfaces.

This ESG survey indicates that, as organizations have become more aware of cybersecurity threats, they are incorporating cybersecurity principles and tools into their contemporary software development and deployment methodologies.

AppSec Is Increasing in Importance

The widespread publicity devoted to major cyber-attacks in the last few years has demonstrated to application developers that no organization or application is immune. Targets have included government organizations, major websites, political campaigns, banks, retailers, and other organizations. A new insidious type of attack—ransomware—has recently gained prominence and publicity, and can affect anyone from the home user to major enterprises. The most recent ransomware attack, known as WannaCry, is reported to have affected more than 200,000 systems across more than 100 countries, including FedEx and the National Health Service of the UK. What these endpoint systems all had in common was a software vulnerability susceptible to exploitation, highlighting the importance to uncover and fix vulnerabilities as early in the lifecycle as possible.

Increasing publicity has driven increasing awareness to the point that cybersecurity is now a boardroom issue, with many organizations appointing a chief information security officer (CISO), and applying increasing pressure for more and better security. Thus, the importance of AppSec to research participants has increased over the last two years. Eighteen percent of participants in ESG's research indicated that code testing and application security was the software development teams' top priority, up seven percentage points from 2015. Likewise, an additional 62% said that AppSec was very important, compared with 55% in 2015.

Best Practices, Compliance, and Efficiency Drive Adoption

General security best practices have become well understood and well documented within the AppDev community, so it's no surprise that 65% of respondents indicated that best practices were a major driver behind the adoption of AppSec tools (namely static application security testing, or SAST, and dynamic application security testing, or DAST) and services (like third-party code audit or testing services).

Many contemporary applications must comply with security and security-related regulations, including the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes Oxley Act, the Federal Information Security Management Act of 2002 (FISMA), the Family Educational Rights and Privacy Act (FERPA), the Payment Card Industry Data Security Standard (PCI-DSS), the Gramm Leach Bliley Act (GLBA), and the General Data Protection Regulations (GDPR), among other acts and regulations. Developers also understand that finding and fixing vulnerabilities early in the development cycle provides more security and is more efficient than discovering vulnerabilities in production, or worse, learning the hard way that a vulnerability has been exploited by a successful cybersecurity attack. Thus, beyond best practices, developers were motivated to employ AppSec because of the need for regulatory compliance (53%), efficiency (43%), and corporate governance (40%) (see Figure 1).

Figure 1. Drivers for Adoption of AppSec Tools and Methodology

Source: Enterprise Strategy Group, 2017

Functionality Is the Most Important Developer Metric

Reliability and security go hand in hand; a vulnerability is a defect, and affects reliability. However, even with the increasing recognition of the importance of AppSec, developers are still measured first and foremost on functionality. One-third (33%) of those surveyed indicated that the single most important metric that they are evaluated on is the functionality of their code and the ability to meet user requirements. Twenty-four percent indicated that they are primarily measured on code reliability (e.g., low incidence of failures, or low mean time to resolution). Only 18% reported they are measured on the security of code first (e.g., delivering code with no/minimal vulnerabilities).

Security May Be Viewed as an Obstacle

Application developers know that effective AppSec is not easy; they must understand both the functionality they are trying to implement and how an attacker can exploit their code. The more complex the application, the larger the attack surface, and the greater the possibility for malicious actors to take advantage of unanticipated interaction between modules.

While implementing security requirements adds complexity and lengthens development time, less than a quarter (22%) of those surveyed agree or strongly agree with the statement, "Our software development team views security as a hindrance and avoids working with our security team." Over three-quarters were either neutral or disagreed, indicating that while AppSec is viewed as a hindrance by a minority, it may still be an impediment to AppDev.

Future AppSec Initiatives

Beyond training developers on security best practices and developing secure code, this survey shows that organizations perceive value from AppSec tools and services, and are investing to further enhance their security posture. Over the next 12 to 24 months, 33% of respondents indicated that all developers will be required to perform static application security testing (SAST) as part of unit testing before code is checked in, while 31% will incorporate dynamic application security testing (DAST) into their automated test and production environments through integration with DevOps tools. DAST will also be included as part of software quality assurance (SQA) by 29% of organizations.

Most applications include externally sourced libraries, modules, and source code components, and it is commonplace to include open source code in many applications. As part of improving AppSec, 28% of respondents indicated that they will conduct regular composition analysis to understand the provenance of their software componentry.

Agile and DevOps in Lockstep

Application development has evolved from the traditional, linearly planned and executed waterfall methodology to Agile development, an iterative, team-based approach encouraging rapid and flexible response to change. Agile focuses on adaptive planning, evolutionary development, early delivery, and continuous improvement.

Like AppDev, contemporary application deployment has evolved. The analogue to Agile is DevOps, which focuses on collaboration and communication between application developers and IT professionals while automating the process of software delivery and infrastructure changes. DevOps aims to establish a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably. Dev and Ops may collaborate, and in advanced organizations, may merge into a single team. DevOps sometimes integrates or subsumes software quality assurance (SQA) and security teams.

Broad Adoption of Agile and DevOps

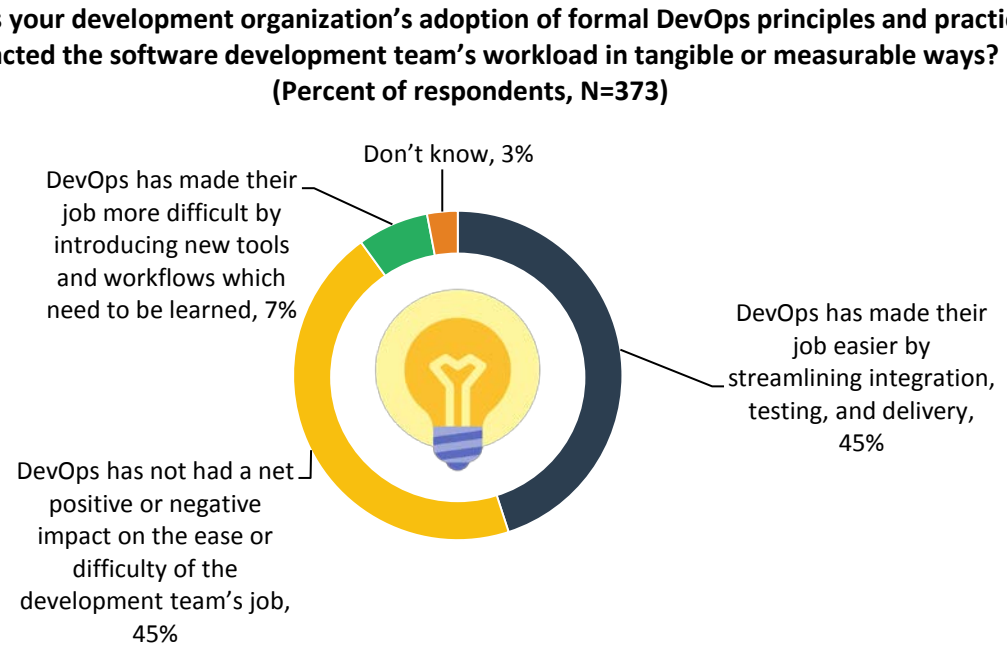
This ESG survey clearly shows that Agile is now the predominant AppDev methodology, with only 5% of respondent organizations indicating that they have not adopted Agile. Almost one-fifth (18%) of surveyed organizations use agile for most or all of their development efforts, and more than one-quarter (28%) use Agile for more than half of their development efforts. The remaining 32% say they have adopted Agile for less than half of their development teams.

Unsurprisingly, given their symbiotic nature, DevOps adoption appears to occur in lockstep with Agile adoption, with only 6% of respondent organizations indicating they have not formally adopted DevOps. Seventeen percent say they have extensively adopted DevOps.

Positive Impact of Agile and DevOps

DevOps is perceived as having a net-positive effect, streamlining the development process, and enabling the integration of security testing. Forty-five percent of IT decision makers say their move to DevOps has made development teams' jobs easer, a plurality of respondents (see Figure 2).

Figure 2. Impact of DevOps Methodology



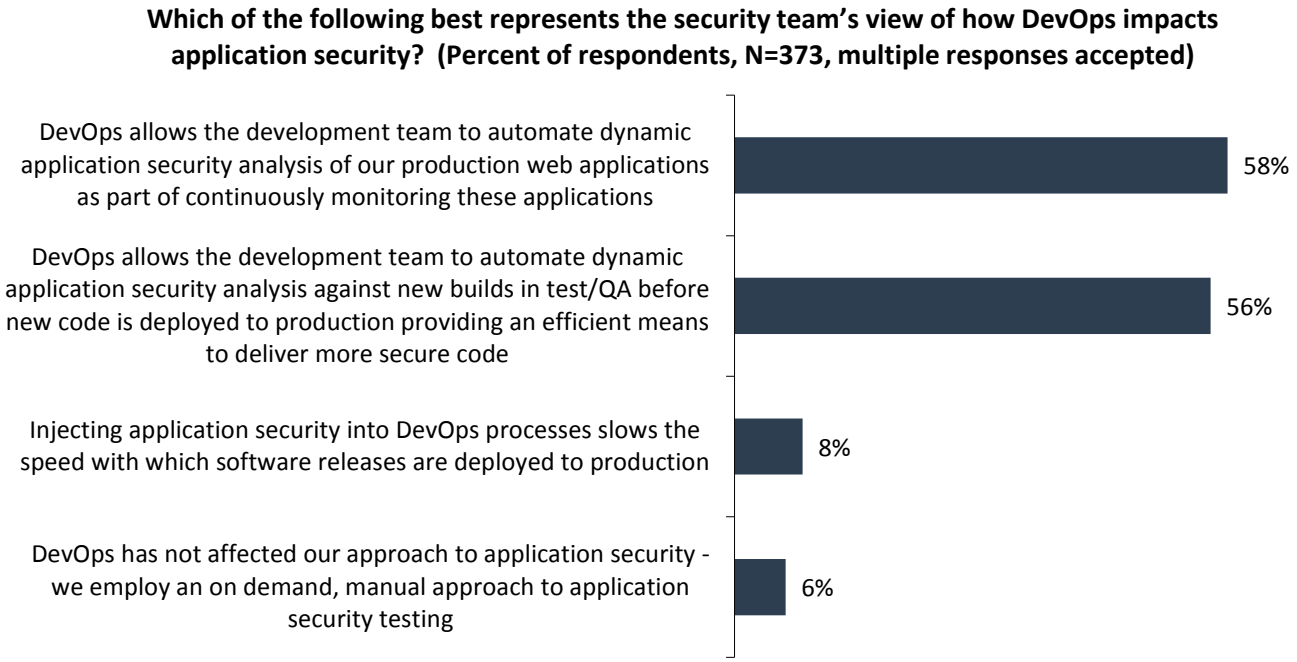
Source: Enterprise Strategy Group, 2017

Furthermore, when it comes to gaining efficiencies from incorporating application security testing into the development process, 43% of respondents noted that correcting security defects in the development stage is more efficient than patching production systems.

AppSec Automation via DevOps

Given the acute shortage of cybersecurity skills, every organization needs to be more operationally efficient. One method for improving efficiency is automation. Application developers and security teams have reaped security benefits from DevOps, easing automation of security testing. Over half (58%) of respondents indicated that DevOps enabled automation of DAST for continuous monitoring of web apps, while 56% said that DevOps enabled automation of DAST for test and SQA before deployment, improving development efficiency and code security (see Figure 3).

Figure 3 DevOps Security Benefits



Source: Enterprise Strategy Group, 2017

Composition Analysis Making Inroads

Agile and DevOps are dynamic methodologies. Externally developed software components can be injected into applications at any point in time, and tracking vulnerabilities can be challenging. Routine validation of software composition to understand the provenance of components can enhance application security. Fifteen percent of respondents indicate they conduct composition analysis on a scheduled cadence, and another 31% include composition analysis as part of the software build process. More than one-third (38%) of organizations are evaluating the benefits of composition analysis.

Mobile- and Cloud-first Equalize AppSec Priorities

Contemporary application development and deployment encompasses more than Agile and DevOps methodology. The digital landscape has changed, and companies have realized that users are now accessing more content on their mobile devices than anywhere else. Mobile-first development is the philosophy of designing applications for mobile devices before designing for the desktop, and requires a new approach to planning, design, development, and security.

Cloud-first is the contemporary application deployment methodology, analogous to mobile-first AppDev. When deploying new or existing applications, organizations should consider and fully evaluate potential cloud solutions first before considering on-premises or legacy infrastructures. In many cases, cloud-first offers a leaner, more efficient, and more manageable approach to enterprise functionality than legacy architectures.

New Apps Favor the Cloud

The US Chief Information Officer mandated cloud-first for all federal agencies in December 2010. Two years later, more than half of all federal agencies had adopted cloud computing for at least one application, and by 2014 more than 1,000 federal data centers had closed or were scheduled to close.

A separate ESG research survey demonstrates that the private sector quickly took notice of the cost reductions and increased efficiencies that come from adopting a cloud-first strategy. More than one-third (36%) of respondents indicated

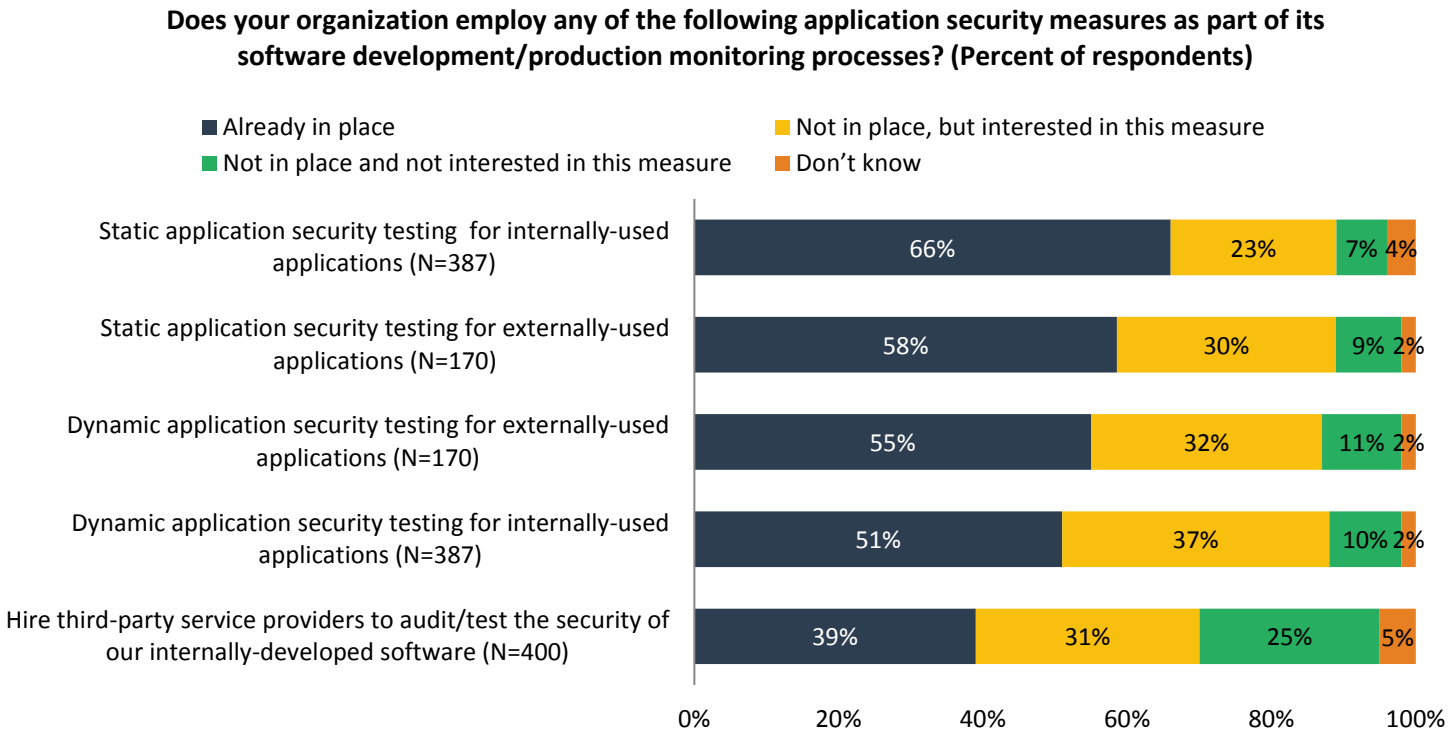
that they employ a cloud-first deployment methodology. An additional 44% said that they weigh on-premises technology and public cloud services equally when considering their deployment options. Only 19% of respondents employed an on-premises-first philosophy.[1]

SAST and DAST Obtain Equal Priority

Traditional application development distinguished between internal and external apps. There is a perception that internal apps have limited security risks and possibilities for compromise because they were hidden behind the corporate firewall, and were only used by internal employees. Externally facing apps, or apps with public users, were perceived to have much greater risk. This perception seems to explain the fact that 66% of respondents employ SAST for internal applications while only 51% employ DAST for internal applications.

However, mobile- and cloud-first initiatives may be blurring the lines between what is considered internal or external. Users accessing corporate data through cloud or mobile apps may bypass the corporate firewall to access apps using insecure public and cellular networks. Performing both static and dynamic testing of all apps, and incorporating and automating AppSec into AppDev should become a best practice, regardless of the app deployment or use model. This concept is reflected in the research results: Including respondents that are interested in SAST and DAST respectively, usage for each should normalize over time (see Figure 4).

Figure 4. Use of Application Security Measures as Part of Software Development/Production Monitoring



Source: Enterprise Strategy Group, 2017

AppSec Crossing Organizational Boundaries

In traditional development organizations, the security team was siloed from the development team. Often, security was an afterthought—something to be bolted on to the app at the end of the development processes. This led to vulnerabilities being discovered mostly after the fact, when the app was compromised and the organization was breached. However, ESG

[1] Source: ESG Research Report, *2017 IT Spending Intentions Survey*, March 2017.

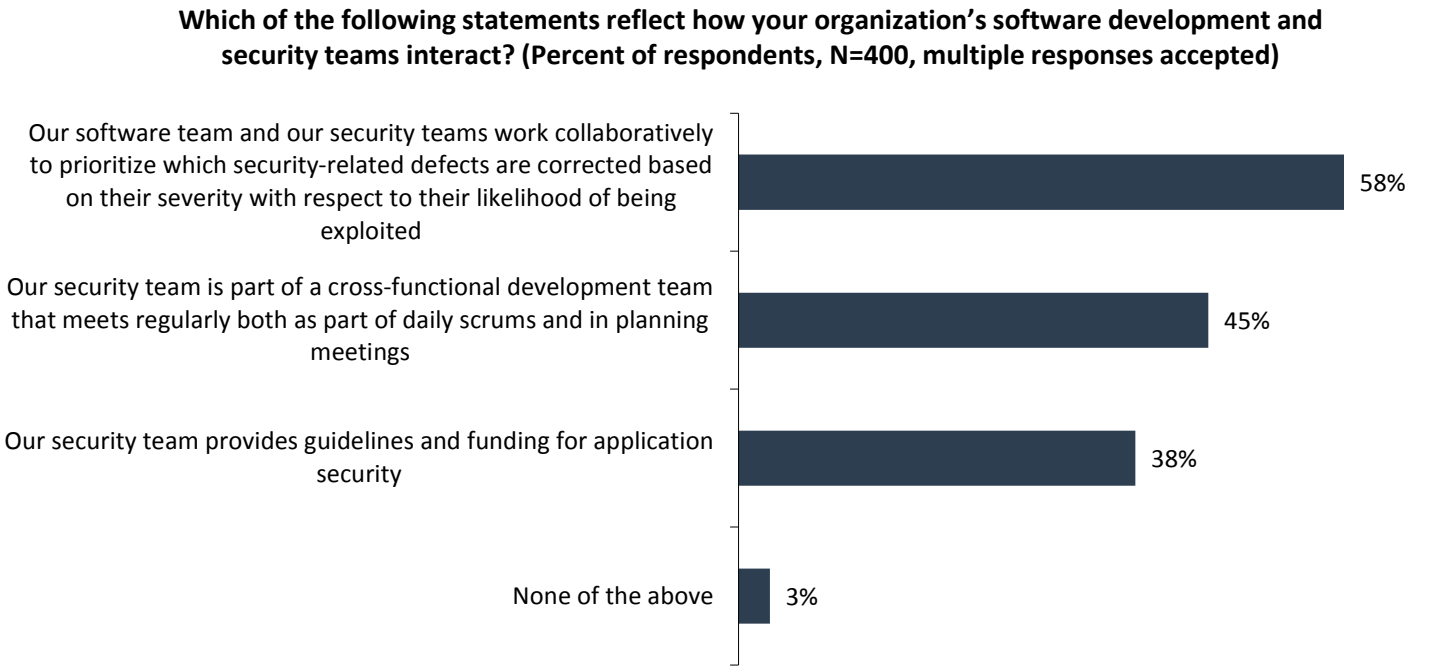
research has uncovered the fact that contemporary AppDev and DevOps methodologies foster communication and collaboration between the application development and security teams with the goal of identifying and fixing vulnerabilities as early as possible to increase efficiency and enhance security.

Better Together: AppDev and Security Collaboration

Contemporary methodologies like Agile and DevOps foster teamwork and communication. Developer and security teams working together at different stages throughout the software development lifecycle engenders the ability to find and fix vulnerabilities earlier in the process. Communication leads to developers having a better understanding of the security implications of their choices for architecture and implementation, and leads security professionals to have a better grasp of the limits of implementation.

Surprisingly, since it is contrary to the historical separation, 58% of respondents indicated that AppDev and security collaborate to prioritize security defects based on likelihood of exploitation, and 45% said that the security team regularly participates in daily scrums and planning meetings (see Figure 5).

Figure 5. Application Development and Security Team Interaction



Source: Enterprise Strategy Group, 2017

Communicating Vulnerabilities

Collaboration between AppDev and security teams includes communication of the identification and correction of vulnerabilities. This research reveals that while AppDev teams report security vulnerabilities, the cadence of reporting varies. Where the AppDev team provides the security team with a dedicated report of security defects that were identified and corrected, 34% of respondents indicated that reporting was provided for each software release, and 21% indicated that reporting was provided for every build. Thirty-two percent said AppDev teams provided only a summary report of security fixes as part of quality metrics. Only 12% fix but do not track or report security defects.

Operationalizing AppSec

AppDev and security teams recognize the benefits of AppSec tools, and appreciate the ability to integrate AppSec into the software development lifecycle. However, the difficulty of workflow integration, combined with tool complexity, and the lack of the requisite knowledge and skills are obstacles for AppSec adoption. This indicates the need to focus on operationalizing AppSec with DevOps, thus forming "DevSecOps."

Efficiency, Coverage, and Efficacy Top SAST and DAST Drivers

Standalone tools are insufficient in contemporary application environments, where development, test, and production are highly automated. The highest levels of effectiveness and efficiency can be realized when tools are tightly integrated into the workflow, and their use becomes not only mandatory but automatic. Thus, 42% of respondents indicated that the ability to integrate into the software development lifecycle was their most important criterion when evaluating SAST tools. Other considerations, almost equally important, are accuracy of test suites (40%), support for both client- and server-side code testing (36%), price (35%), and ease of use (34%).

Unsurprisingly, the requirements for DAST were similar to SAST, with workflow integration and ease of use both ranking as most important according to 34% of survey respondents. Accuracy and incorporation of malware detection were both cited by 33%, and support for both client- and server-side code testing was cited by 32% of respondents.

SAST and DAST Adoption Impediments

AppSec helps developers identify and fix vulnerabilities that may otherwise lead to major breaches. However, among developers who aren't incorporating security testing into the software development lifecycle, the leading reasons are that the process is too complex and they lack the skills or knowledge to use AppSec tools (see Figure 6).

Figure 6. SAST and DAST Adoption Impediments

Source: Enterprise Strategy Group, 2017

Next Steps: Embrace DevSecOps

The active use of SAST, DAST, and composition analysis can be an effective approach to enhancing application security and increasing efficiency, but realities of the ongoing global cybersecurity skills shortage, continuously morphing threats and attacks, and the automated nature of contemporary application development and deployment have created the following challenges for AppSec solutions:

- **Integration of AppSec** tools into AppDev and DevOps.
- **Security may be viewed as an obstacle** to efficient and timely application development.
- Not all developers are measured on the **security of applications.**
- **Mobile- and cloud-first initiatives** may blur the lines between internal and external applications and thus the security technologies and processes that should govern them.

Even with these challenges, and perhaps as a consequence of them, two-thirds (66%) of participants in this ESG research are planning to increase their application security investment over the next 12 to 24 months. The requirements for an AppSec solution, based on the results of ESG research, can best be characterized as integrated for automation, easy to use, and comprehensive in terms of coverage and efficacy.

Operationalizing AppSec to Achieve DevSecOps

ESG research indicates that organizations have embraced the contemporary application development and deployment of Agile and DevOps. Integrating and automating SAST, DAST, and composition analysis into AppDev and DevOps environments, representing, by extension, a DevSecOps methodology, enables the identification and correction of cybersecurity vulnerabilities earlier in the application development lifecycle. Automating AppSec as part of the process ensures that security testing is a standard part of Dev, Test, and Prod processes. This enhances security, increases efficiency, and helps alleviate challenges resulting from the global cybersecurity skills shortage.

Integrating and automating SAST, DAST, and composition analysis into AppDev and DevOps environments, representing, by extension, a DevSecOps methodology, enables the identification and correction of cybersecurity vulnerabilities earlier in the application development lifecycle.

These research results show that tool complexity and the lack of ability to integrate AppSec into the DevOps workflow are major obstacles to organizations employing these tools. At the same time, the ability to integrate SAST (42%) or DAST (34%) into the AppDev and DevOps processes is the most important tool selection criteria.

Make Application Security a Team Sport

Fifty-eight percent of participants in this ESG research indicate that their application development and security teams work collaboratively to prioritize which security-related defects are corrected based on their severity with respect to their likelihood of being exploited. AppSec automation as part of DevSecOps reinforces security as a primary focus for AppDev. What's needed next is to embrace AppSec in every facet of Agile and DevSecOps to ensure that security is never given short shrift. Steps to be taken can include:

- Organizations should make security a primary metric for all owners, from Dev to Test to Prod.
- Product owners should create and relate AppSec stories as part of the Agile sprint and scrum.

- AppSec workflows should be incorporated into the Kanban board.
- Security teams should participate in architecture definition and application development.
- AppDev teams should participate in attack remediation.
- Organizations should increase the pace of testing and reporting.

Embrace Mobile- and Cloud-first for both Internal and External Applications

In the past, the distinction between an internal or external application was considered important, and was the key determinant of the type of AppSec tool used for security testing. Internal apps were tested statically, while external apps were tested dynamically.

Mobile- and cloud-first initiatives may blur the line between what is considered an internal or an external application. Applications may be accessed from outside the corporate network, using insecure WiFi or cellular networks. Application users may extend beyond employees to include partners and vendors. Multiple apps may interact or exchange confidential data, even when designed for a limited audience.

To ensure comprehensive application security, regardless of the device, network, or user of the application, all types of security testing should be applied to every application. The DevSecOps process should automate static and dynamic testing, along with routine composition analysis, regardless of whether the app is considered internal or external.

The Bigger Truth

Application architecture is evolving as organizations adopt digital workplace transformations, IoT, and mobile- and cloud-first initiatives, resulting in an ever-increasing attack surface. Malicious actors are always on the lookout for new applications, and are continuously creating new types of attacks targeting previously unknown vulnerabilities.

Security, however, often takes a back seat to functionality, as shown by the research conducted by ESG. Some developers view security as an obstacle, and are measured first and foremost on application functionality. This highlights the need for the next iteration of contemporary application development and deployment to incorporate security as a fundamental objective. Integrating and automating application security testing and composition analysis into DevOps to create the hybrid DevSecOps should be a core element in shrinking an organization's attack surface and reducing potential avenues of compromise, enhancing the security posture to protect sensitive data and applications.

Enterprise Strategy Group is an IT analyst, research, validation, and strategy firm that provides actionable insight and intelligence to the global IT community.

 www.esg-global.com

 contact@esg-global.com

 P. 508.482.0188