Interested in learning more about security?

SANS Institute InfoSec Reading Room

2016 State of Application Security: Skills, Configurations and Components

Survey results reveal that it is critical for an overall enterprise security program to coordinate efforts among developers, architects and system administrators particularly since many software vulnerabilities are rooted in configuration issues or third-party components, not just in code written by the development team. Read on to learn more.

2016 State of Application Security: Skills, Configurations and Components

A SANS Survey

Written by Johannes Ullrich, PhD

Advisor: Eric Johnson

April 2016

Sponsored by

Checkmarx, Veracode, and WhiteHat Security

Executive Summary

Application security (AppSec) is maturing for most organizations, according to the 475 respondents who took the SANS 2016 State of Application Security survey. In it, respondents recognize the need for AppSec programs and are working to improve them, despite a lack of the necessary skills, lack of funding and management buy-in, and silos between departments hampering their AppSec programs.

Despite these mostly organizational inhibitors, the majority say their programs are maturing or mature: 38% say their AppSec programs are "Maturing," while 22% say their programs are "Mature" and 4% report programs that are "Very Mature." The majority (67%) have also partially integrated AppSec into their overall security, risk management and incident response (IR) programs, while another 17% have achieved full integration.

Key Findings

have "Maturing" AppSec programs

have partially integrated AppSec into overall security, risk management and IR programs

have documented approaches and policies to which third-party vendors must adhere

report applications are the source of breaches, attacks on others, or sensitive data leaks

name public-facing web apps as the leading cause of breaches

They are also making stronger demands on third-party vendors: 40% of the 2016 survey respondents have documented approaches and policies to which third-party software vendors must adhere, while in 2015, only 28% had any comprehensive vendor risk management program and the majority relied on the word of the vendors.[1]

Respondents identified training as the most useful AppSec process, even ahead of vulnerability scanning. Much of that training may be going to developers. Unlike last year, when 22% of respondents indicated that the development team was responsible for security testing, now 30% of respondents assign responsibility for security testing to the development team.

Results also show that organizations are defining AppSec testing roles and responsibilities across their security, development, business, architecture and QA teams. This may explain why only 23% said their applications were the source of actual breaches that resulted in attacks on others or loss of sensitive data.

Of those, public-facing web applications were the largest items involved in breaches and experienced the most widespread breaches, which aligns with respondents' ranking of different applications by risk. Accordingly, most AppSec resources are allocated to public-facing web applications.

Overall, the survey results reveal that it is critical for an overall enterprise security program to coordinate efforts among developers, architects and system administrators—particularly since many software vulnerabilities are rooted in configuration issues or third-party components, not just in code written by the development team.

[1] "2015 State of Application Security: Closing the Gap," www.sans.org/reading-room/whitepapers/analyst/2015-state-application-security-closing-gap-35942, Figure 10, p. 19.

Participation

AppSec is not a problem of a particular industry. Today's companies all rely on data and software to process data. As a result, AppSec affects all sectors and sizes of organizations, and our respondents represent a wide array of businesses of different sizes.

The respondents for our survey were split about evenly between small and medium size companies (<1,000 employees), large companies (1001–10,000 employees) and very large enterprises and governments (> 10,001 employees).

Even smaller companies often invest heavily in custom applications to achieve a competitive advantage. AppSec protects these systems and ensures not only that proprietary data is secure from theft, but that decisions are made based on correct and reliable data.

Industry Type

The financial services, government and application development verticals were the most common industries chosen by participants. As noted in the 2015 survey, application development companies feel pressure from customers to provide security assurance for their products. See Figure 1.

What is your organization's primary industry?

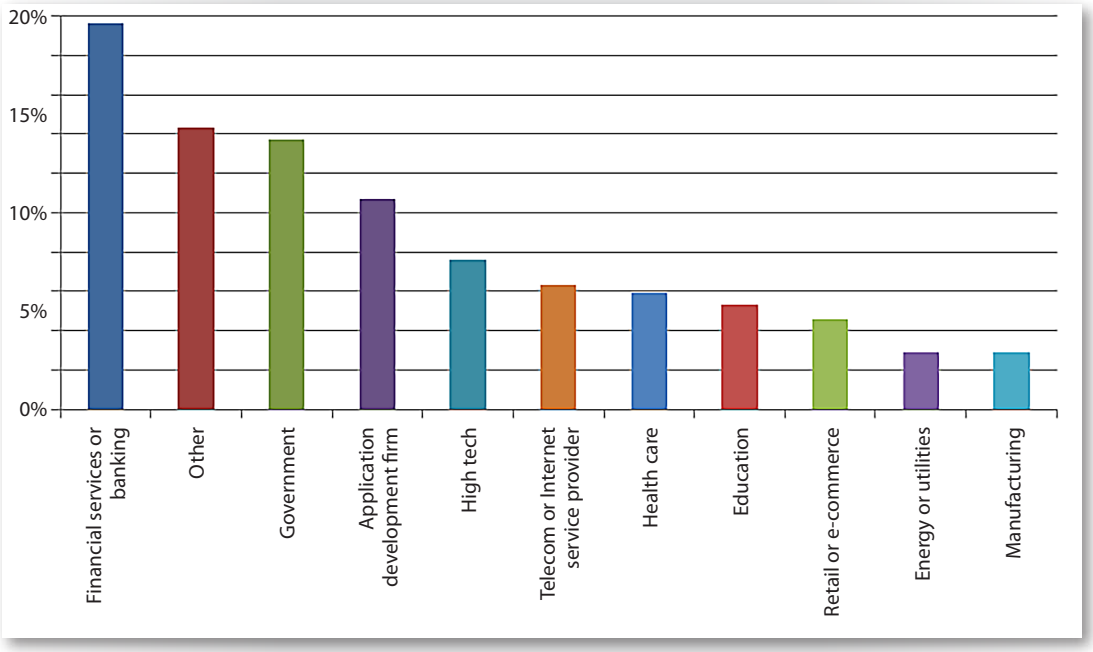


Figure 1. Top Industries Represented

The "Other" category ranks second highest among the industries represented. It includes a variety of respondents, such as consulting and professional services firms, as well as media-related industries, engineering and construction, transportation and pharmaceuticals, that reflect the ubiquitous nature of software development and the need for AppSec.

Roles

Security administrators and analysts made up 30% of respondents, while 21% represented senior-level security managers and 12% were security architects, as illustrated in Figure 2.

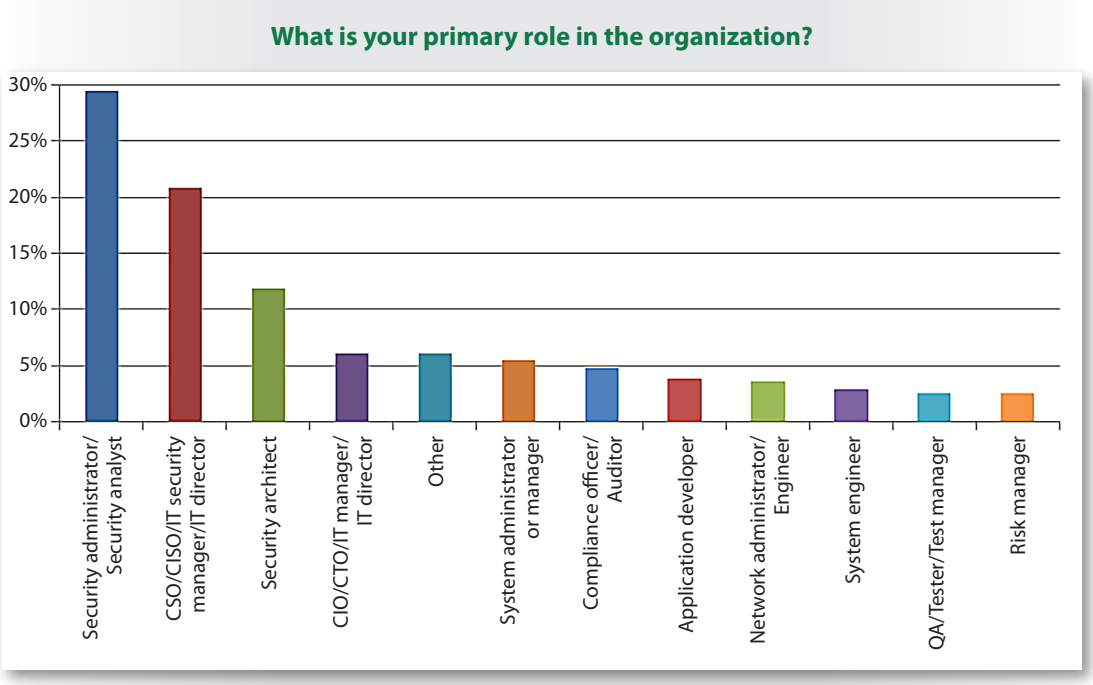


Figure 2. Respondent Roles

This survey base is consistent with the SANS membership, which is made up of administrators, engineers and managers focused on security and risk management.

Responsibility for AppSec

Although security professionals represented the largest group in this survey, they are not necessarily the ones who are managing risk associated with their applications. For example, responses reveal a large and distributed group of roles that are responsible for testing AppSec, developing and executing the corrective action plan, performing final acceptance and signing off on test results. See Figure 3.

Who is responsible for running the application security testing for your organization or work group? Who is responsible for final acceptance of the testing results and any corrective actions resulting from that testing? ***Select all that apply to your organization.***

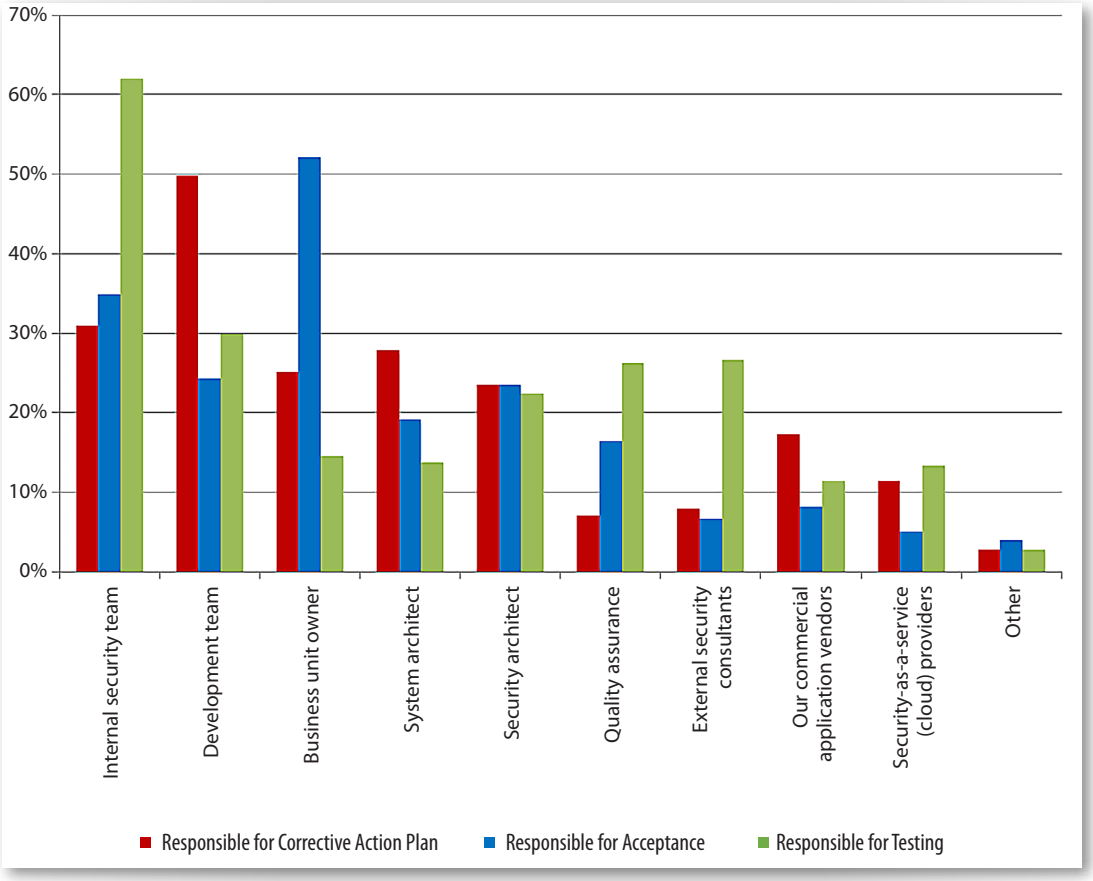


Figure 3. Responsibility for AppSec Testing, Acceptance and Correction

As expected, for most respondents, internal teams take the lead for testing, with the development team taking the lead for the corrective action plan. Business owners take the lead for final acceptance.

Use Independent Testers

Treat quality assurance and security bugs as having equal importance. Use an independent team of testers who are, necessarily, separate from the developers who write the original code. A different set of eyes is more likely to find bugs because they don't already know how the application is supposed to work.

Unlike last year, when 22% of respondents indicated that the development team is responsible for security testing, now 30% of respondents assign responsibility for security testing to the development team. This may reflect a difference in responding organizations, who is considered a member of the development team, or a trend toward developing more security competencies on the development team. Such a trend follows what we saw in last year's survey, where developers indicated they were improving their secure DevOps practices and finding secure development training to be highly effective in reducing their risk.[2]

[2] "2015 State of Application Security: Closing the Gap," www.sans.org/reading-room/whitepapers/analyst/2015-state-application-security-closing-gap-35942, p. 23.

Maturity of Programs

AppSec is still a developing area and is not as mature as many infrastructure and system security programs. The largest response group (38%) considers its AppSec program to be "maturing," while only 26% of respondents consider their programs to be "mature" or "very mature," as shown in Figure 4.

How mature do you consider your AppSec program to be?

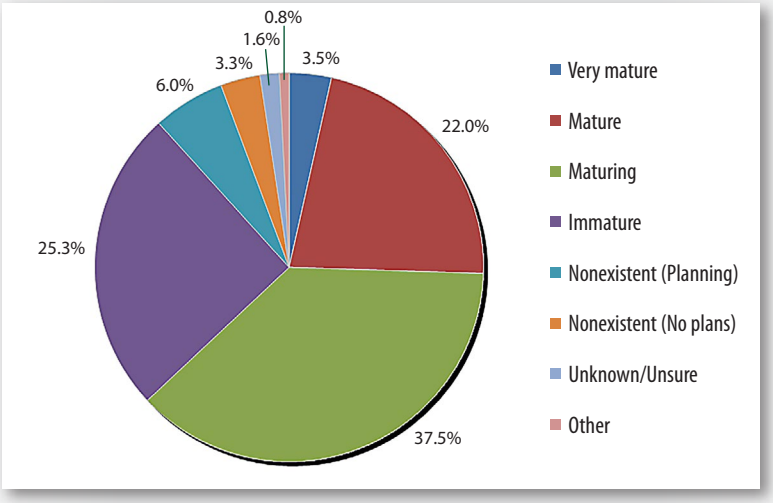


Figure 4. Maturity of AppSec Programs

Any corporate risk assessment should include an AppSec security component to be meaningful. For instance, more mature organizations use models, such as the Capability Maturity Model Integration for Development (CMMI-DEV), as the guide for their application development programs.[3] However, many organizations have a limited focus on security-related best practices. To that end, the CMMI Institute released a guide for improving processes relating to the development and delivery of secure applications. Organizations invested in CMM-DEV should review the application guide, "Security by Design with CMMI for Development," Version 1.3, which provides guidance on improving the existing processes with security components.[4]

[3] www.cio.com/article/2437864/process-improvement/capability-maturity-model-integration--cmmi--definition-and-solutions.html and http://cmmiinstitute.com

[4] www.cmmiconsultantblog.com/information-security/what-is-security-by-design-with-cmmi-for-development

Most Mature Sectors

Only 3% of respondents have no AppSec program at all and no plans to enact one, which indicates the importance of AppSec. In particular, in the financial industry, and for larger companies that are subject to industry and government regulations, AppSec is becoming a compliance issue and receiving C-level attention as a result. Table 1 provides an informal look at how mature respondents believe their AppSec programs are by the most represented industries.

Table 1. AppSec Maturity by Industry

Industry (Percent of Sample)	Very Mature	Mature	Maturing	Immature	Nonexistent (w/AppSec Plans)	Nonexistent (No AppSec Plans)
Financial Services/Banking (21.6%)	**1%**	**28%**	**47%**	19%	1%	3%
Government (13.7%)	4%	14%	38%	24%	12%	4%
Application Development Firm (11.5%)	10%	29%	24%	29%	10%	0%
High Tech (7.1%)	**8%**	**50%**	**19%**	15%	4%	4%
Health Care (6.3%)	4%	9%	17%	70%	0%	0%
Telecom or ISP (6.3%)	**13%**	**22%**	**39%**	13%	4%	4%
Education (4.9%)	0%	17%	11%	**50%**	**6%**	**17%**
Retail or E-commerce (4.9%)	0%	11%	50%	28%	6%	0%

In viewing these results, it is important to note that sample sizes for each industry varied, potentially affecting results. These results illustrate a trend that is not necessarily statistically significant. However, it is clear that the relative maturity of implementation of AppSec programs is higher in some industries. The high-tech industry, financial and banking organizations, and telecom, for example, appear to have higher levels for program maturity, as evidenced by the higher totals of the top maturity levels (77%, 76% and 74%, respectively. Maturity for these industries is essential, given the number of applications they likely develop. A second tier, including retail and application development firms, are maturing. Again, this is not surprising, given today's digital world.

Perhaps surprisingly, though, education leads the list of verticals with immature or nonexistent AppSec programs, with 73% across those options. Most enlightening is that 17% of education respondents neither have an AppSec program nor plans to institute one. This lack of concern for application security is alarming when we consider the number of public-facing web applications used by educational institutions for everything from registration to purchasing textbooks.

Integration

One of the best measures of AppSec maturity is how integrated these processes are with security and IR operations. Despite their concerns about silo mentalities, 67% of respondents have partially integrated AppSec into these operations, and 65% are partially satisfied with this stage of their integration. Another 17% have integrated fully, and 13% are satisfied with this full level of integration. See Figure 5.

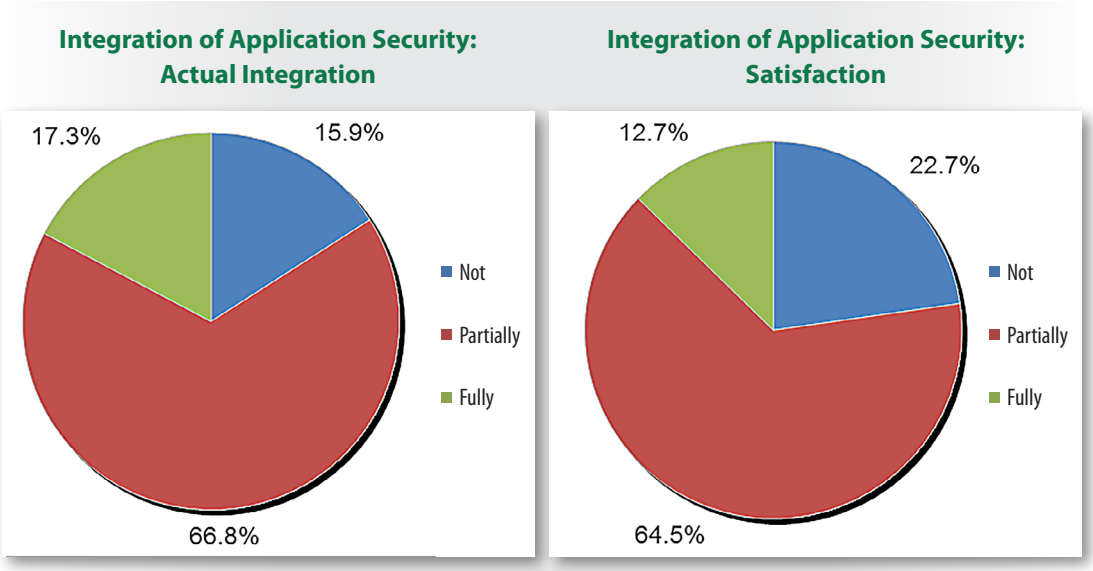


Figure 5. Integration of AppSec and Satisfaction Levels

A fully integrated AppSec program can reap benefits in overall security posture and IR capabilities. An AppSec program spans internally developed applications and applications procured from outside vendors. Integrating such a program provides valuable input for the overall enterprise security program, including IR. For example, for a purchased application, a predeployment AppSec review will identify configuration requirements to ensure that the application is used securely. The review will also identify log management/review requirements and establish a baseline for expected application behavior. In case of an incident, this information can be valuable in helping responders identify the incident and analyze a possible compromise of the application.

Respondents report worrying most about public-facing web applications, as well as their legacy applications. These applications are also those most frequently breached, according to the 23% of respondents who say that applications were the source of actual breach, data loss and attacks on others. See Figure 6.

What applications or components were involved or were the cause of these breaches, and how widespread was their impact? ***Leave blank those that don't apply.***

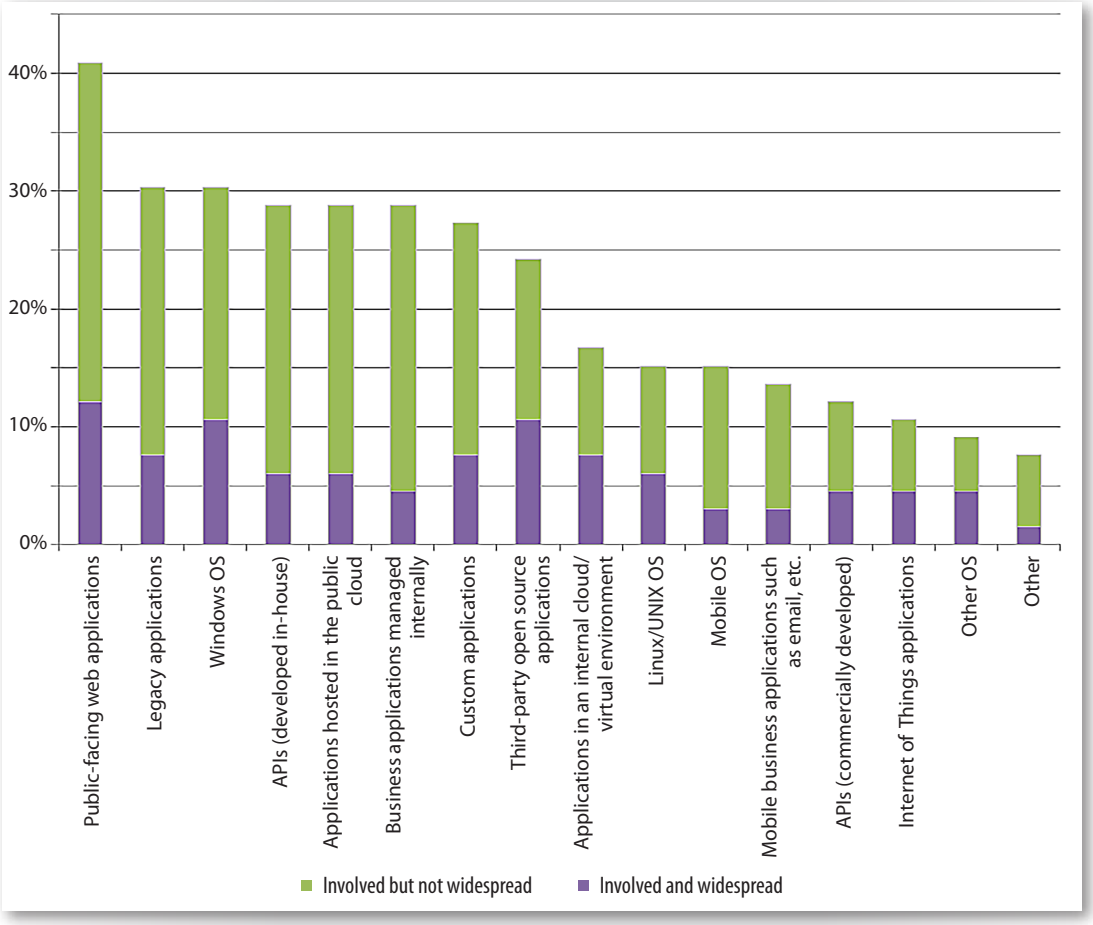


Figure 6. Applications Leading to Breaches

Many web applications are directly exposed to external attacks and, while infrastructure systems such as web application firewalls exist, they are often considered inadequate for deterring a sophisticated attacker. Interestingly, we are also seeing breaches into applications hosted in the cloud, which is an area we should be watching more. Cloud-based web applications are often more exposed than web applications hosted in traditional enterprise networks. In cloud environments, implementing network controls such as firewalls, web application firewalls, intrusion detection systems and similar controls can be difficult. In many cases, implementing these controls requires buying additional expensive services from the cloud provider.

Risky Languages

As they were in last year's survey, respondents are most concerned about applications developed in Java and .NET, the predominant languages used in modern enterprise web applications. The focus on these languages is likely due to their popularity in these environments, not a particular weakness in these languages.

.NET Improving

.NET has added incrementally improved security controls in each version. Regularly review any legacy applications written in .NET to take advantage of these additional controls. For example, ASP.NET 5 added a completely new authorization API. The old API used specific, hard-coded role or even usernames to provide access control, which has been difficult to maintain for larger applications. The new authorization API allows for more flexible policies that can be defined with specific requirements and privileges.

JavaScript has been an up and coming language in many large web applications on the client side. With technologies such as Ajax and browsers using newer JavaScript APIs as part of HTML5, web applications are taking advantage of JavaScript by pushing more business logic and data to the client. In particular, on websites designed for mobile devices, JavaScript is used heavily to provide users with an "app-like" user experience. However, this trend does make applications more vulnerable by exposing internal data and APIs to external users. Testing tools need to mature enough to adequately support this new breed of applications.

More recently, JavaScript has also become popular as an option for server-side tools, with frameworks such as AngularJS and Node.js being used to deliver complex applications. The security implications of these frameworks have not yet been fully explored. As with client-side JavaScript, testing of these applications is difficult to automate in the same way testing for traditional web applications is automated.

Resources Aligned to Risk

When it comes to risk and investment to protect against that risk, web applications are directly followed by legacy applications, in particular legacy applications for which the source code is available. Because they are difficult to patch and upgrade, legacy applications are often considered to be at high risk, even if they are not exposed to the public. Figure 7 illustrates which types of applications are consuming the most security resources.

On what types of applications are your AppSec resources being spent?
Select those that most apply.

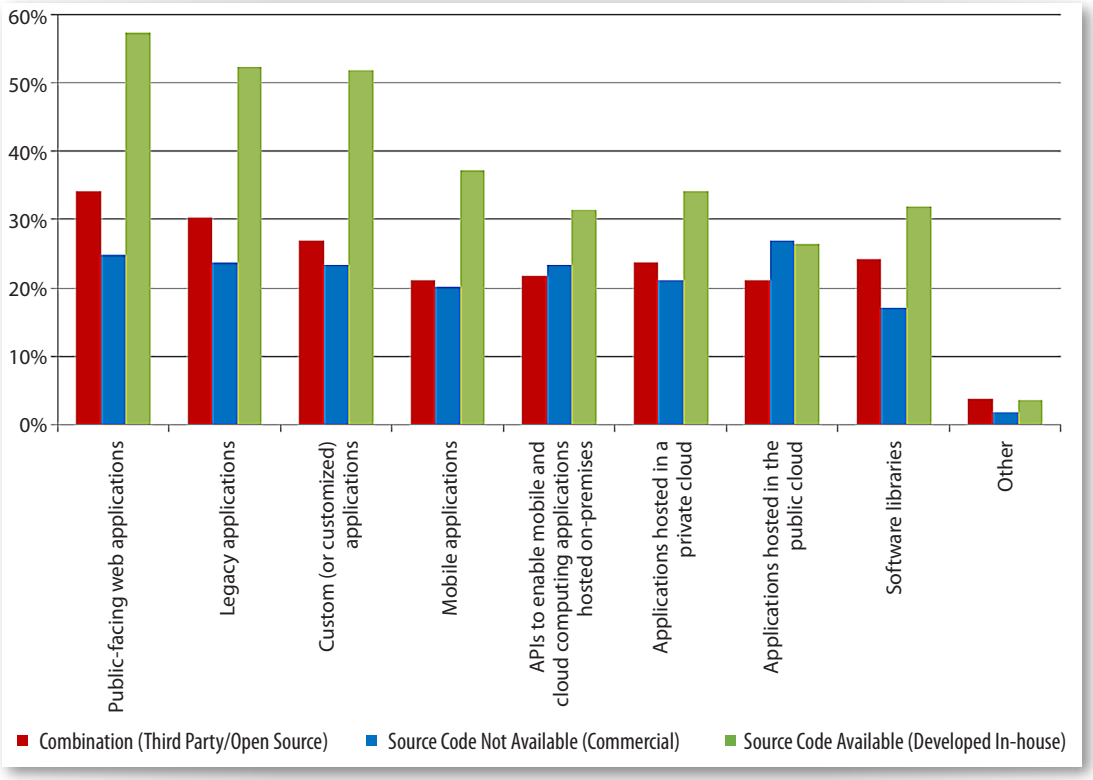


Figure 7. AppSec Resource Allocation by Type of Application

In the fast-moving world of security, organizations often review and amend secure coding guidelines as new attack vectors are uncovered. The result is that older applications need to be reviewed from time to time to apply new protective measures to the code. This can be a rather time-consuming and expensive undertaking that usually does not add any new features or improve performance. Quite the opposite, the revisions may reduce performance if, for example, newer and stronger cryptographic algorithms are added. Survey results, however, show that organizations recognize the problem and are dedicating a high level of resources to securing legacy applications.

Top Challenges and Most Useful Controls

The lack of AppSec skills, tools and methods was ranked as being in the top three challenges to implementing AppSec by 38% of respondents, followed by lack of funding or management buy-in (37%), silos between security, development and business units (33%), and identifying all applications in the portfolio (32%), as shown in Figure 8.

What are your top three challenges in implementing application security for systems in production at your organization? ***Indicate the top three, in no particular order.***

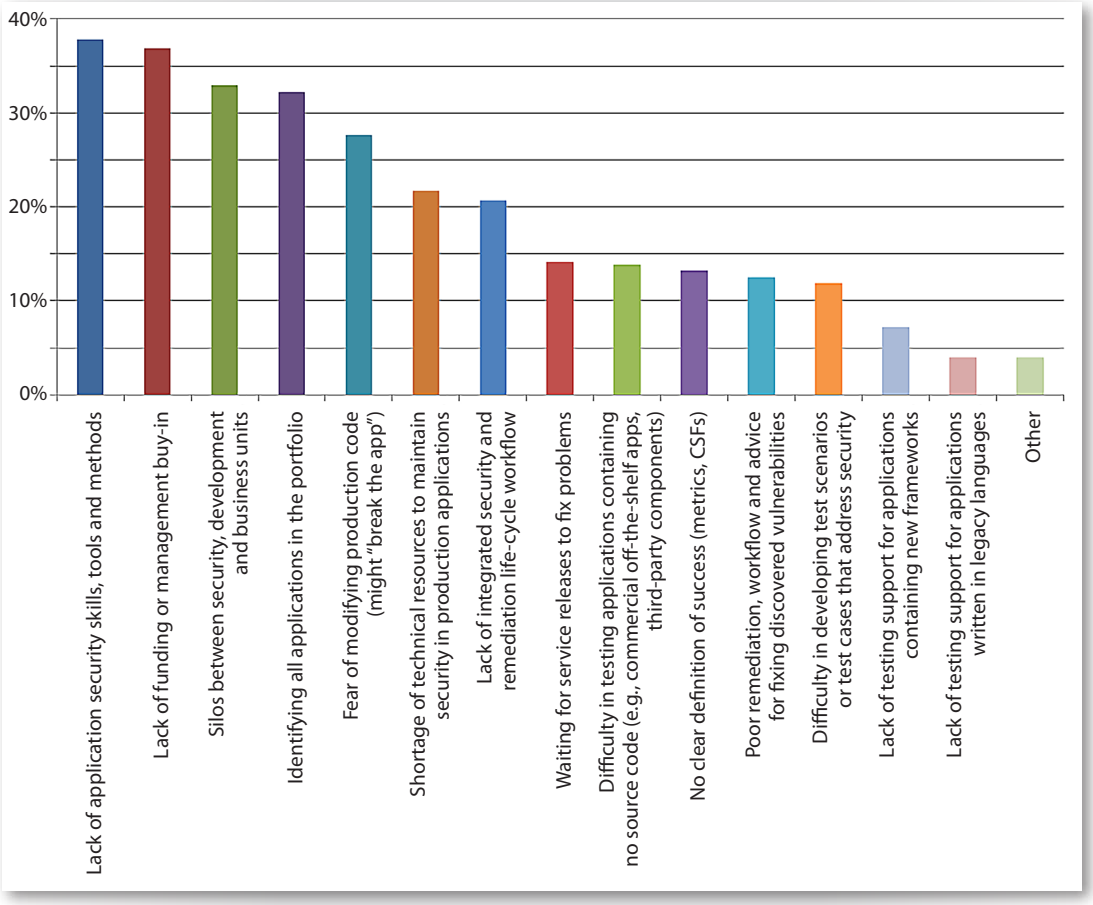


Figure 8. Top Challenges

To solve this fundamental gap—the lack of AppSec skills, tools and methods—training emerges as an important enabler and foundational process needed to conduct testing and implement better development practices. Organizations appear to have the right idea about how to address their concerns on this issue. Respondents overwhelmingly pointed to training developers on AppSec as among the top three AppSec processes and tools, with 48% choosing that option. See Figure 9.

Select in no particular order the three most useful application security processes and tools your organization uses.

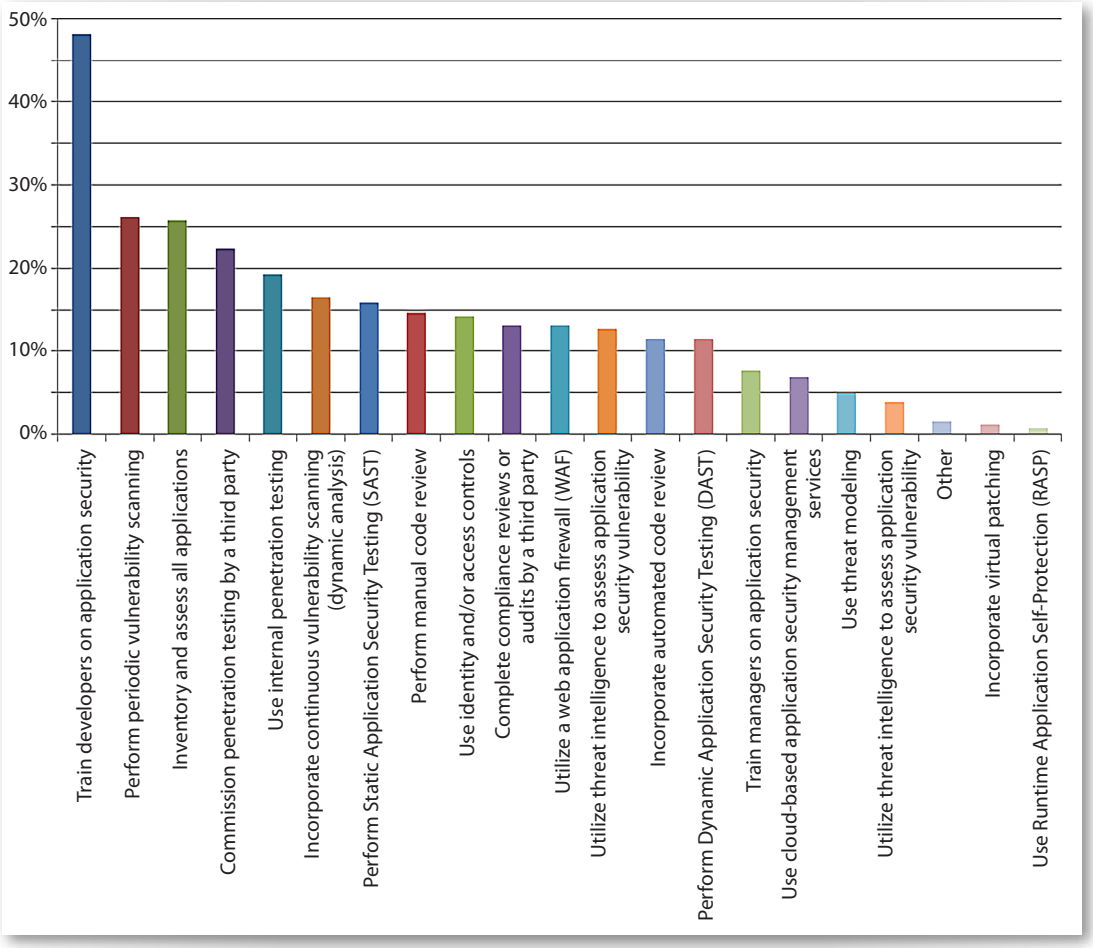


Figure 9. Top AppSec Processes and Controls in Use

Funding and Budget

Lack of funding or management buy-in is the second biggest challenge to AppSec programs, as illustrated in Figure 8 (on page 12). In the survey, 18% spend less than 1% of their IT budgets on security, 11% spend 1%, and 23% spend between 2% and 5%. See Figure 10.

What percent of your overall IT budget is spent on AppSec?

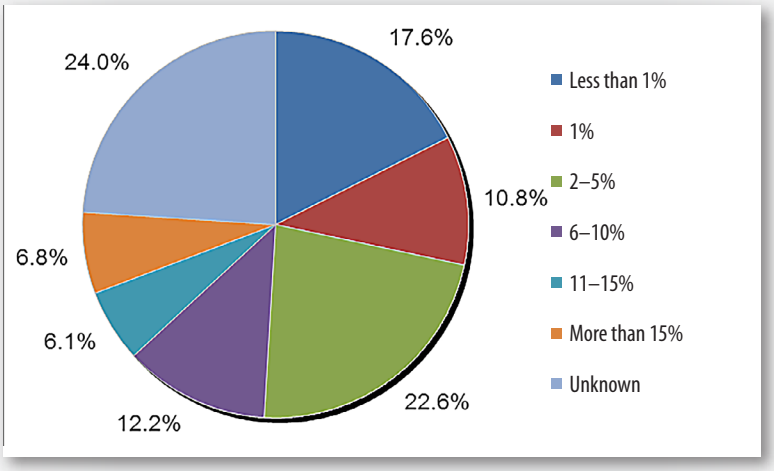
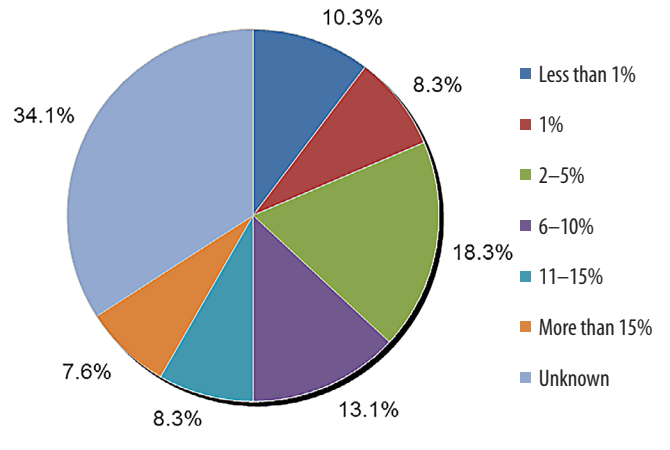


Figure 10. Portion of Budget Devoted to AppSec

Comparing the percentage of the overall IT budget to results from last year's survey does not show a clear trend. In both surveys, the largest portion of respondents didn't know what portion of their budget was devoted to AppSec. However, that percentage did decrease in the 2016 survey. In 2015, 37% of respondents reported up to 5% of their budget went to AppSec, compared with 51% making the same report this year. However, 2015 saw a larger percentage (29%) of organizations devoting more than 6% of their budgets to AppSec efforts, compared to 25% in 2016. See Figure 11.

2015 Survey:
What percent of your overall IT budget is spent on AppSec?



2016 Survey:
What percent of your overall IT budget is spent on AppSec?

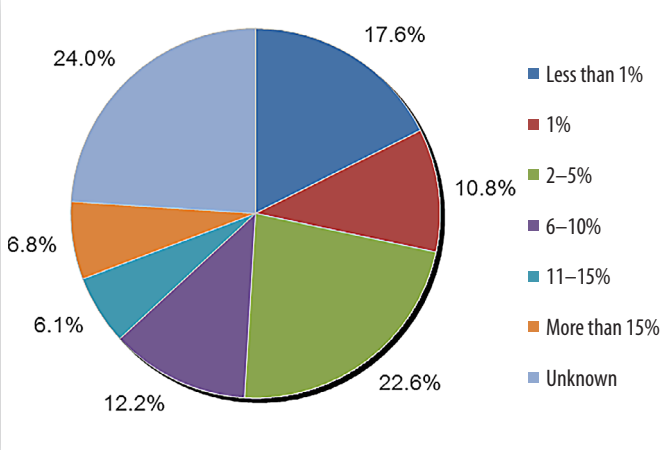


Figure 11. Comparison of AppSec Budgets Between 2015 and 2016

It is not surprising that AppSec spending varies widely for the diverse set of organizations represented in this survey. The size of an AppSec program and the budget dedicated to it depends widely on the amount of internal and external development being undertaken. Another important factor is whether the cost for AppSec is rolled into purchase agreements or broken out as a line item in purchase contracts. The overwhelming majority of organizations (61%) expect AppSec spending to increase in the future. However, about a fifth of respondents didn't provide an answer, which may be due, in part, to those respondents not having budget authority.

Testing

The software development life cycle (SDLC) begins with the planning and development of applications and upgrades and doesn't end until the application has expired and been removed from the environment. Fortunately, respondents get this. All but 14% of respondents test their AppSec.

Test Prior to Launch

Modern Agile development practices can make it difficult to perform comprehensive code scans and practical vulnerability scans (including penetration tests) at the end of each sprint. This makes it important to perform continuous vulnerability scans, as well as periodic (e.g., every six months) penetration tests that cover the entire application in the live environment.

Test schedules are diverse, but 60% indicate that they test applications continuously, with 27% using continuous assessment in their Agile development processes and 53% of respondents testing applications when they are initially launched into production. This means some of those doing continuous monitoring may not be testing at initial launch. However, it is possible that many of these organizations use a faster update cycle and usually test applications when they are updated, patched or otherwise changed, an option 41% of respondents selected. Figure 12 illustrates the testing cycles followed by respondents' organizations.

When do you assess or test the security of your business-critical applications?
Select those that most apply.

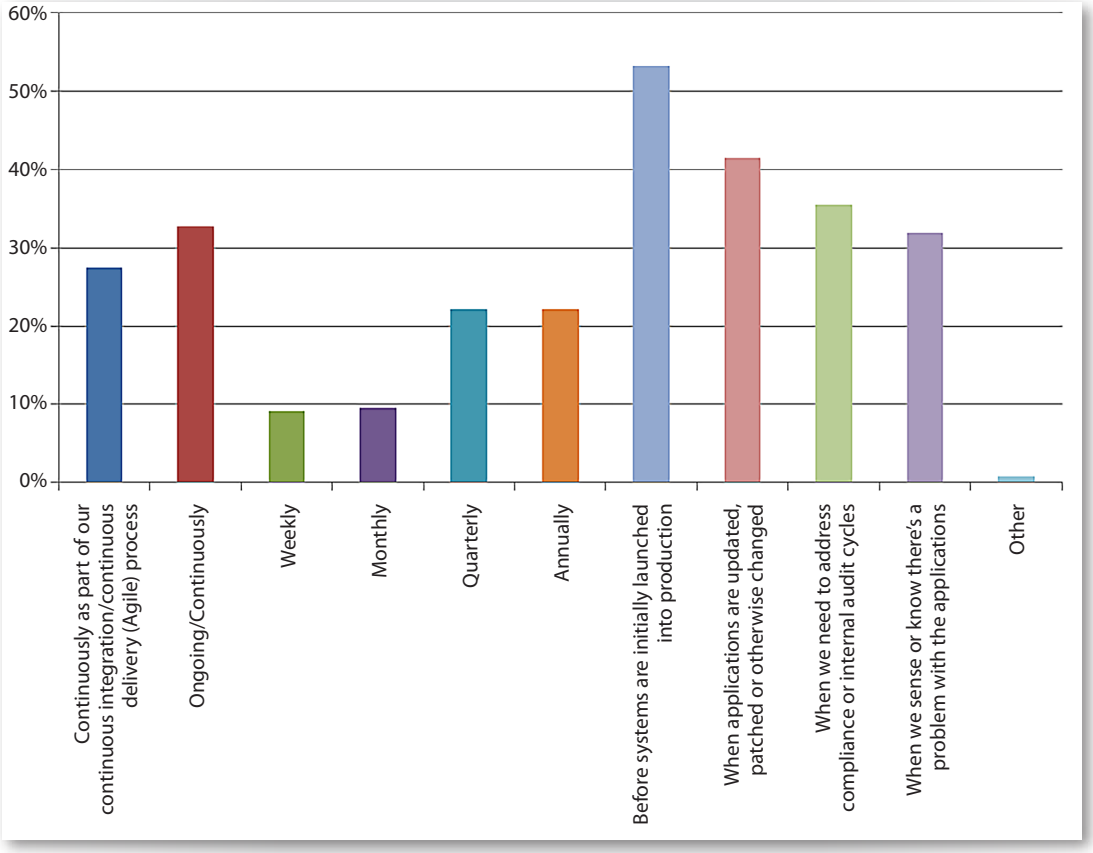


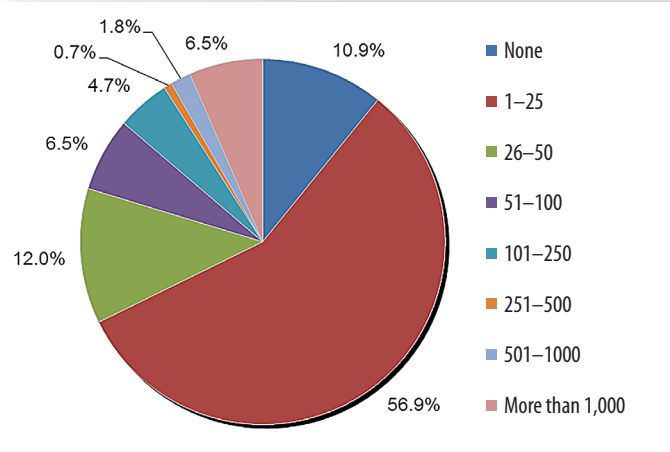
Figure 12. Testing Cycles and Practices

Organizations still rely heavily on various forms of runtime testing that are typically performed in the final stages of development or after the application has been deployed. This is also reflected in most of the testing being performed by the security department. Internal teams are responsible for testing applications, according to 62% of respondents. Note that this survey was primarily focused on nondeveloper organizations, which would explain why, for these organizations, the IT team typically performs vulnerability scans and penetration tests.

What They're Finding

With all the ways they're testing their apps, organizations are finding fewer flaws than we had expected, which can potentially be attributed to the fact that this survey was more focused on apps in production than apps in development. The largest group (57%) said they find one to 25 vulnerabilities per month, while 12% find 26 to 50 vulnerabilities per month through their testing efforts. Of those vulnerabilities discovered, 54% said that only 1% to 10% were critical and in need of immediate patching or countermeasures (such as virtual patching or RASP). See Figure 13.

How many vulnerabilities are you discovering per month in your applications?



Of these, how many do you rank as critical and in need of immediate remediation?

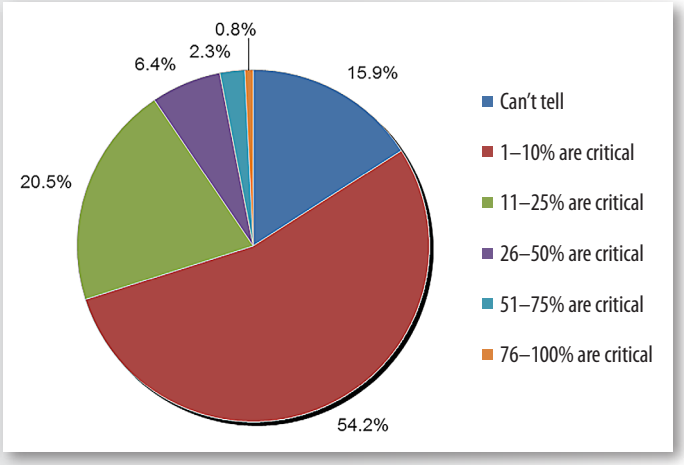


Figure 13. Vulnerabilities Found and Their Criticality

In the survey, the largest number (24%) said 50–74% of critical vulnerabilities they found were related to code bugs rather than to misconfigurations, while 21% indicated that only 10–24% of the critical vulnerabilities they found were the result of code-based bugs, as shown in Figure 14.

What percent of those critical application vulnerabilities resulted from bugs in code versus misconfigurations in the environment or other vulnerabilities?

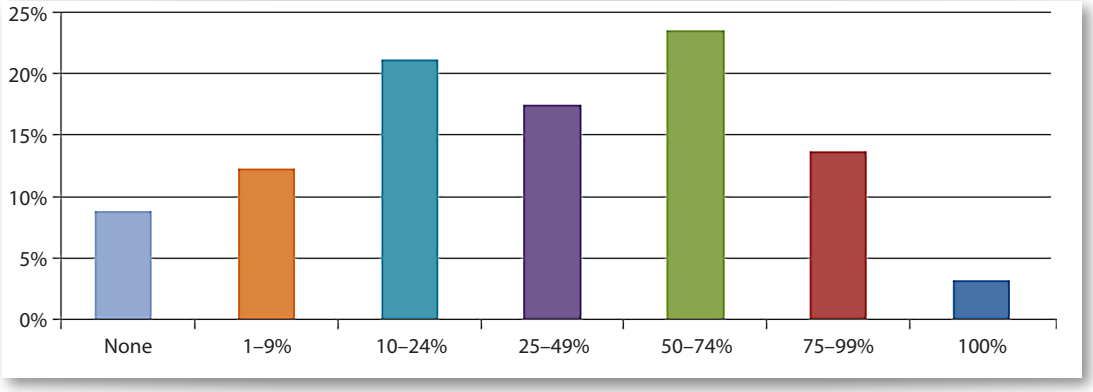


Figure 14. Vulnerabilities as a Result of Code Errors Versus Misconfiguration or Other Vulnerabilities

Recently, SSL configuration issues have gotten a lot of attention, and many web server installations have had to be reviewed to harden the SSL configuration. An application often cannot easily verify that SSL is configured correctly. In fact, developers are usually not deploying SSL configurations. Although an application may test that it is accessed over SSL, and could block access without SSL, the cipher used, or the SSL version used, is usually not something the application can control.

On the other hand, encryption of data at rest is often handled by the application. For example, password hashing requirements have typically been increased over the last few years. While in the past, a simple salted MD5 or SHA1 hash may have been considered sufficient, advances in brute-forcing techniques and computing power require modern web applications to use stronger hashing algorithms or to apply the same algorithm multiple times.

Other common vulnerabilities, for example SQL injection, are not mitigated by configuration choices. The impact of the vulnerability can be reduced by restricting access to a database to a user account with limited privileges to connect to the database. Using an administrator account to connect a web application to a database may be considered a vulnerability, even though exploitation of that vulnerability would require a SQL injection or business logic problem.

Remediation

Respondents unfortunately register a low level of satisfaction with their patching and repair process. Less than 30% are achieving a 75%–99% level of satisfaction with the speed it takes to repair their vulnerabilities, while only 11% felt 100% satisfied. The speed at which patches are applied is comparable to last year's survey, with 26% of vulnerabilities being patched within two to seven days, and another 26% within eight to 30 days, as illustrated in Figure 15.

On average, how long does it take your organization to fix and deploy a patch to a critical application security vulnerability for systems already in use?

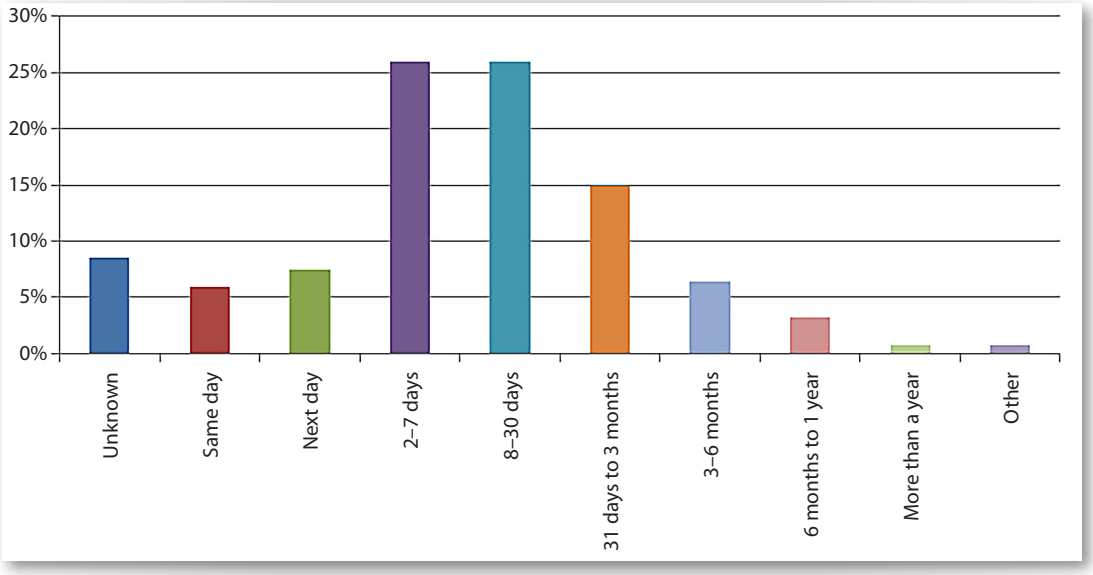


Figure 15. Time to Patch a Vulnerability

Vulnerabilities are repaired a variety of ways, with 58% saying they do thorough updates to the entire environment, while 51% work to resolve the root cause through secure SDLC practices. "Quick and dirty" software patching was cited by the third largest respondent group (50%), and third-party libraries and configuration issues took fourth place (48%). See Figure 16.

How are discovered vulnerabilities repaired?

Select up to three. Order is not important.

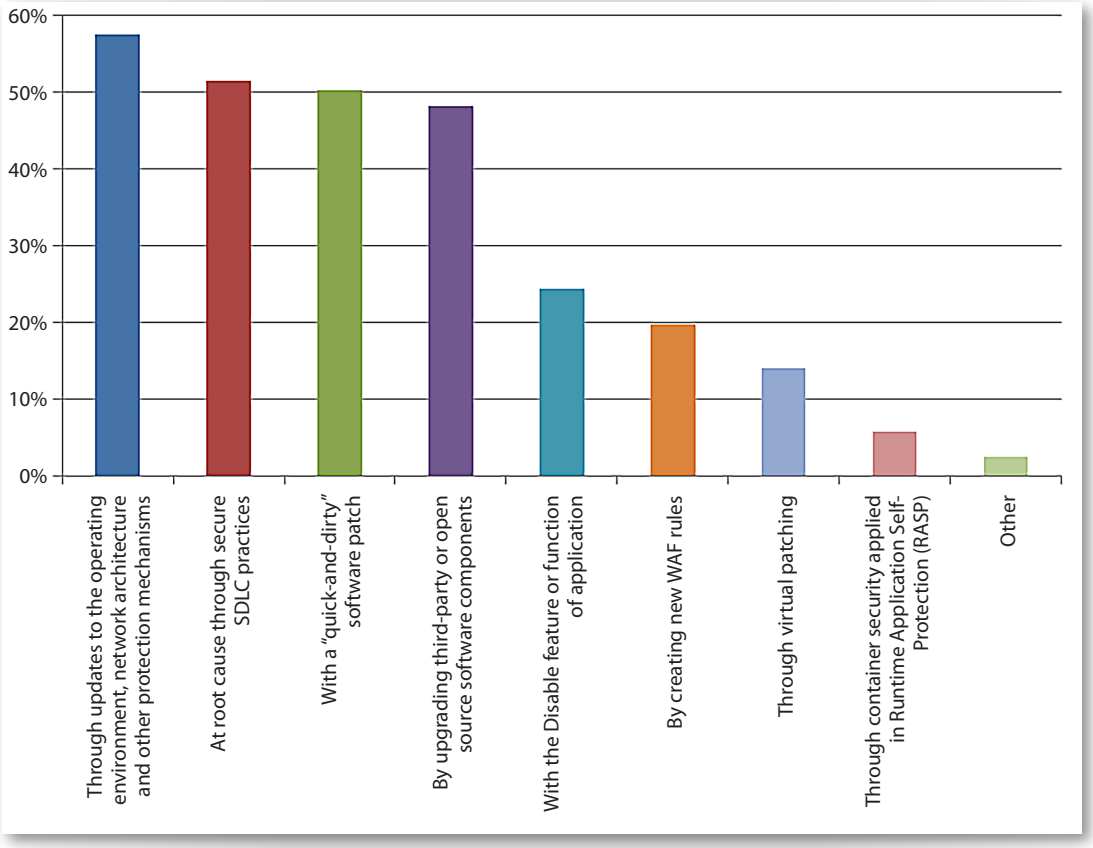


Figure 16. Repair Methodologies

Patching the operating system and fixing configurations are very common methods used to fix flaws caused by configuration issues and third-party libraries, so it is no surprise that these are two of the top four methods to resolve vulnerabilities.

Implementing secure SDLC practices often takes time. Developers have to be educated, and existing code has to be reviewed for similar flaws, which tends to be time consuming. A quick fix can make sense if it prevents exploitation of the flaw until the more permanent fix can be applied after the issue has been sufficiently researched.

Vendor Accountability

In the 2016 survey, 40% of respondents have documented approaches and policies that third-party software vendors must adhere to, while in 2015, only 28% had any comprehensive vendor risk-management program.[5] It has become common practice, particularly among larger customers, to add security performance benchmarks to contract language. Application development companies are asked to provide long-term support to provide security updates. The total cost to create software may depend on the cost of these long-term support agreements. To correctly estimate and reduce these costs, application development companies need to invest more up front to limit their exposure to security vulnerabilities.

It's in the Contract

If you are with an application development company, review your contract to determine what your obligations are with respect to long-term AppSec and be sure to add the related expenses to your price quotes. Security should be part of your software development process. You may find it beneficial to engage the support of vendors that are experts in AppSec.

If you are with a company seeking to purchase an application or retain the services of an application development company, be sure to include contract language that places responsibility for securing the application on the vendor.

[5] "2015 State of Application Security: Closing the Gap," www.sans.org/reading-room/whitepapers/analyst/2015-state-application-security-closing-gap-35942, Figure 10, p. 19.

Conclusion

Results show that it takes a village to protect applications. Security teams, developers, business units, architects and quality assurance personnel are all part of the ecosystem that protects applications. Together, all parties are maturing their AppSec security programs and are aware that they need to mature more.

Skills shortages will continue to be a problem as new technologies emerge. Skills shortages have, historically, been a problem for almost all InfoSec disciplines. Organizations will need to continue to leverage training and education to develop their skill sets.

Successful AppSec programs are tightly integrated with development life-cycle and procurement processes. Currently, most AppSec programs are still new, and growing them will require sufficient resources. To leverage limited budgets for AppSec, it is critical for these programs to overcome silos so that communication among all stakeholders will be promoted.

Important ideas to strengthen AppSec programs include:

- Use independent testers to check applications in production.
- Consider legacy applications, public-facing web applications and cloud-based applications as key applications that need frequent testing.
- Upgrade to continuously test AppSec.
- Do penetration testing before releasing an application.
- Be aware of how user SSL implementations might affect your AppSec.
- Hold vendors accountable for AppSec through inclusion of specific AppSec contract language.

About the Authoring Team

Johannes Ullrich, dean of research at the SANS Technology Institute, is currently responsible for the SANS Internet Storm Center (ISC) and the GIAC Gold program. His research interests include IPv6, network traffic analysis and secure software development. In 2004, Network World named Johannes one of the 50 most powerful people in the networking industry, and SC Magazine named him one of the top five influential IT security thinkers for 2005. Prior to working for SANS, Johannes served as a lead support engineer for a web development company and as a research physicist.

Eric Johnson, the Application Security Curriculum product manager at SANS, is the lead author and instructor for DEV544 Secure Coding in .NET, as well as an instructor for DEV541 Secure Coding in Java/JEE. A senior security consultant at Cypress Data Defense, Eric's experience includes web and mobile application penetration testing, secure code review, risk assessment, static source code analysis, security research and developing security tools. He currently holds the CISSP, GWAPT, GSSP-.NET and GSSP-Java certifications.

Sponsors

SANS would like to thank this survey's sponsors:

CHECKMARX

VERACODE

WhiteHat SECURITY

Last Updated: July 22nd, 2016

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Industrial Control Systems Security Training	**Houston, TXUS**	**Jul 25, 2016 - Jul 30, 2016**	**Live Event**
SANS Vienna	**Vienna, AT**	**Aug 01, 2016 - Aug 06, 2016**	**Live Event**
SANS Boston 2016	**Boston, MAUS**	**Aug 01, 2016 - Aug 06, 2016**	**Live Event**
Security Awareness Summit & Training	**San Francisco, CAUS**	**Aug 01, 2016 - Aug 10, 2016**	**Live Event**
DEV531: Defending Mobile Apps	**San Francisco, CAUS**	**Aug 08, 2016 - Aug 09, 2016**	**Live Event**
SANS Portland 2016	**Portland, ORUS**	**Aug 08, 2016 - Aug 13, 2016**	**Live Event**
SANS Dallas 2016	**Dallas, TXUS**	**Aug 08, 2016 - Aug 13, 2016**	**Live Event**
DEV534: Secure DevOps	**San Francisco, CAUS**	**Aug 10, 2016 - Aug 11, 2016**	**Live Event**
Data Breach Summit	**Chicago, ILUS**	**Aug 18, 2016 - Aug 18, 2016**	**Live Event**
SANS Alaska 2016	**Anchorage, AKUS**	**Aug 22, 2016 - Aug 27, 2016**	**Live Event**
SANS Virginia Beach 2016	**Virginia Beach, VAUS**	**Aug 22, 2016 - Sep 02, 2016**	**Live Event**
SANS Bangalore 2016	**Bangalore, IN**	**Aug 22, 2016 - Sep 03, 2016**	**Live Event**
SANS Chicago 2016	**Chicago, ILUS**	**Aug 22, 2016 - Aug 27, 2016**	**Live Event**
SANS Adelaide 2016	**Adelaide, AU**	**Sep 05, 2016 - Sep 10, 2016**	**Live Event**
SANS Brussels Autumn 2016	**Brussels, BE**	**Sep 05, 2016 - Sep 10, 2016**	**Live Event**
SANS Northern Virginia - Crystal City 2016	**Crystal City, VAUS**	**Sep 06, 2016 - Sep 11, 2016**	**Live Event**
SANS Network Security 2016	**Las Vegas, NVUS**	**Sep 10, 2016 - Sep 19, 2016**	**Live Event**
SANS ICS London 2016	**London, GB**	**Sep 19, 2016 - Sep 25, 2016**	**Live Event**
SANS London Autumn	**London, GB**	**Sep 19, 2016 - Sep 24, 2016**	**Live Event**
Security Leadership Summit	**Dallas, TXUS**	**Sep 27, 2016 - Oct 04, 2016**	**Live Event**
SANS DFIR Prague 2016	**Prague, CZ**	**Oct 03, 2016 - Oct 15, 2016**	**Live Event**
SANS Oslo 2016	**Oslo, NO**	**Oct 03, 2016 - Oct 08, 2016**	**Live Event**
SANS Seattle 2016	**Seattle, WAUS**	**Oct 03, 2016 - Oct 08, 2016**	**Live Event**
SANS Baltimore 2016	**Baltimore, MDUS**	**Oct 10, 2016 - Oct 15, 2016**	**Live Event**
SANS Tokyo Autumn 2016	**Tokyo, JP**	**Oct 17, 2016 - Oct 29, 2016**	**Live Event**
SANS San Jose 2016	**OnlineCAUS**	**Jul 25, 2016 - Jul 30, 2016**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**