Secure Coding.
Practical steps to defend your web apps.

Copyright SANS Institute
Author Retains Full Rights

Interested in learning more?
Check out the list of upcoming events offering
"Defending Web Applications Security Essentials (DEV522)"
at http://software-security.sans.orghttp://software-security.sans.org/events/
A SANS Analyst Survey

Written by Jim Bird and Frank Kim
Advisor: Barbara Filkins

February 2014

Sponsored by
Hewlett-Packard, Qualys and Veracode
This is the SANS Institute’s second survey on application security programs and practices. In this year’s survey, we wanted to uncover answers to the following questions:

- How widespread are application security programs, and how mature are the programs that are in place today?
- How effective are these programs?
- What practices and tools are organizations relying on the most today, and what are they finding the most useful?
- How is secure coding training for developers being done, and how effective is this training?
- How are people justifying spending on Appsec, and where are they spending most of their efforts? Does this spending align with organizational risk?
- What will the future of Appsec look like? Are organizations planning to invest more in Appsec? And what programs or technologies are on their future roadmaps?

We asked some of the same questions in our first survey on application security practices, just in a different way. Some of the trends we identified include the following:

- There was a significant improvement in the number of organizations implementing application security programs and practices. The percentage of organizations that have an active Appsec program increased from 66% last year to 83% this year—and many of the organizations that do not have a program in place yet are at least following some kind of ad hoc security practices.
- Organizations continue to rely heavily on dynamic testing, vulnerability scanning and penetration testing to find security vulnerabilities.
- Organizations are testing more frequently. In this year’s survey, more than one-third are doing continuous, ongoing security testing of their applications, whereas only 23% indicated doing so in our 2012 survey.
- The primary focus of most Appsec programs continues to be web applications, because this is where organizations see the highest security risks.
- Organizations continue to face the same kinds of challenges in getting management buy-in for application security programs. But the leading inhibitor for putting effective Appsec programs in place is now a shortage of application security skills, whereas in last year’s survey, the leading inhibitor was management buy-in and funding. In this year’s survey, organizations also ranked technical resources to maintain security in production as their fourth most difficult problem.

1 www.sans.org/reading-room/analysts-program/sans-survey-appsec
The 488 respondents to this survey represented a broad range of industries. In this year’s survey, financial services (17%), government (15%), “other” (13%) and high-tech firms (9%) led the way; similarly, in last year’s survey, financial services and government were tied at 17% each and high-tech followed the “other” category. Although not the next in terms of representation, it is noteworthy that 6% of respondents came from application development houses. Figure 1 illustrates the diversity of the industries represented in this survey.

Application security is a consideration for every organization, regardless of size. Small and mid-size organizations and large enterprises were all included in the survey, as illustrated in Figure 2.
One-quarter of the respondents worked in very large enterprises of more than 15,000 people, and almost 39% were from organizations with 1,000 or fewer people, lending a representative sampling of organizational size to the survey results.

We also asked participants to identify the principal role they play in their organization (whether as a consultant or an employee). Most respondents were from the security community, as shown in Figure 3.

Security analysts or security managers made up 44% of the sample. Software developers (developers, engineers, architects, and testers) accounted for 12%, and IT managers and executives also accounted for 12%. IT operations was also well represented, with 14% of the respondents in system admin or network engineering. Approximately 28% of the respondents are in a management or executive role.
To further refine our understanding of survey responses, we wanted to know how big the application development teams were in our responding organizations. Figure 4 shows the number of developers employed by responding organizations.

Although 10% of respondents had software development organizations with more than 2,000 developers, 30% of development teams were small, with fewer than 25 developers—and 6% of respondents had no developers at all, relying completely on third parties for software development.

There seems to be a distinction between the practices for designing and developing applications. Although most organizations design their systems internally, either using their own employees (75%) or consultants (38%), fewer use internal employees (52%) or consultants (33%) to develop the applications after design, as shown in Figure 5.
Only 18% of respondents hire third-party firms to complete their application design work, and 22% hire third parties to do their development work. A full 41% of respondents also rely on commercial off the shelf (COTS) applications, and just under 24% of firms rely on open source software.
Application Development Priorities

Where are organizations spending most of their development dollars? Web applications and business-critical apps, which are often the same, (both at 67%) stand well above the others as recipients of development dollars, as shown in Figure 6.

Mobile applications (28%) are becoming a major focus for organizations, ahead of spending on legacy apps (25%).

Application Security Risks

In last year’s survey, we asked what kinds of applications posed the biggest security risks to an organization. In order, the results were:

- Customer-facing web apps (by far the highest risk)
- Internal web apps
- Mobile apps
- Legacy apps and CRM/databases (usually accessed through Web and mobile channels)
This year’s survey didn’t distinguish between types of web apps, but it’s clear that the highest security risk continues to come from web applications, with 38% selecting this as their biggest application risk area, and business-critical applications (19%), as shown in Figure 7.

Mobile risk has slipped in the ranking, with only 6% feeling that to be their biggest risk; only 7% see cloud-based services as a major security risk. Organizations also continue to downplay the risks of working with third parties, whether COTS providers (8%) or outsourced development organizations (3%).

Figure 7. Application Security Risks
Application Security Programs

We wanted to know how many Appsec programs are in place, how long they have been in place, how administrators justify their programs, and what practices and tools people rely on the most.

**Maturity of Appsec Programs**

Almost 74% have programs that have been in place for at least one year, and more than one-third (37%) have programs that have been running for more than five years (see Figure 8).

Even in organizations that don’t have a formal program today, most (79% of those without a formal program) are following ad hoc Appsec practices.

The number of organizations with an active Appsec program has increased significantly over the past year. Table 1 shows how the maturity of programs has changed since our 2012 survey.

**Table 1. Growth in Appsec Programs**

<table>
<thead>
<tr>
<th>How Long Has Your Appsec Program Been in Place?</th>
<th>2012</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>No formal program</td>
<td>34.3%</td>
<td>16.9%</td>
</tr>
<tr>
<td>Less than 1 year</td>
<td>9.8%</td>
<td>9.0%</td>
</tr>
<tr>
<td>1 to 5 years</td>
<td>32.9%</td>
<td>36.7%</td>
</tr>
<tr>
<td>More than 5 years</td>
<td>22.9%</td>
<td>37.3%</td>
</tr>
</tbody>
</table>

*Figure 8. Maturity of Appsec Programs*
Justification of Appsec Program Support

Earlier this year, John Pescatore at the SANS Institute analyzed the different approaches and tools that organizations can use to secure management support and funding for an application security program. He reported the following options:

- Using a publicized incident to illustrate risk/benefit
- Managing regulatory pull—meeting regulatory requirements such as PCI, NIST, HIPAA, FDA and NERC
- Taking advantage of industry governance standards (ITIL, COBIT and ISO 27034)
- Capability Maturity Models (Cigital’s BSIMM or OWASP’s OpenSAMM)
- Industry benchmarking

Our survey results quantify the use of these options. Organizations are taking proactive and reactive approaches to justifying application security spending, as illustrated in Figure 9.

How do you justify funding for your Appsec program?

![Figure 9. Justifications for Appsec Spending](image-url)
Risk analysis based on industry benchmarks is used to justify spending by 43% of organizations, and 21% benchmark spending to justify their programs. Reactive approaches include justifying spending in response to audit findings (39%), a security incident (26%) and customer demands (25%). Responding to customer demands is a driver that we identified in last year’s survey: Organizations, especially large enterprises, are being pushed more by their customers, and are, in turn, pushing their software and software-as-a-service (SaaS) suppliers to implement responsible Appsec programs.

Costs for Appsec programs are being included in general IT security programs 33% of the time, in regulatory compliance programs 31% of the time, and in specific IT programs or project budgets 27% of the time. Only 17% of Appsec costs are included in software quality spending.

Most of the justifications for Appsec spending are focused on security, compliance and risk management—not on enabling the business or supporting the business strategy. Spending on application security programs will continue to lag until the information security team can make an explicit connection not just to incidents and hacking or staying up-to-date on compliance requirements, but also to enabling business strategy and meeting customer demands.

**Support of Appsec Programs**

On the whole, Appsec initiatives seem to be aligned with where organizations are spending development and IT dollars—and to where organizations see the greatest risk. The priorities for Appsec security spending are highlighted in Figure 10 and align closely with development spending (Figure 6) and perceived risks (Figure 7).
Most organizations are focusing their Appsec programs where it makes the most sense today, on where they are spending most of their development dollars: web apps (80%) and business-critical apps (72%), which are often the same. But they are also trying to keep pace with emerging threats. While only 27% of development/IT resources are being spent on developing mobile apps, 35% of organizations are focusing their Appsec attention on mobile security issues; and application security focus on cloud implementations (23%) matches the amount of development and other IT resources spent in this area (19%).

However, even though 23% of respondents rely heavily on third-party software products and services (COTS, cloud-based services and open source software), they are not taking enough responsibility for ensuring the security of third-party solutions. Only 23% of security programs include COTS. The same is true for cloud services, and only 14% focus on open source software.
This situation should improve as the security industry continues to highlight the risks of relying on outsource and third-party providers and the open source community to police themselves. For example, a recent study conducted by Sonatype and Aspect Security[^3] on the use of open source software found that more than 50% of Global 500 organizations are using open source code with known security vulnerabilities. In 2013, OWASP added the use of insecure third-party software components to the OWASP Top 10 risk list[^4], a widely used application security risk management tool. The Financial Services Information Sharing and Analysis Center (FS-ISAC)[^5] published a set of guidelines that banks and other organizations can use to assess the application security programs of their software and software service providers, and SAFECODE and the Cloud Security Alliance[^6] released a new set of guidelines for securing cloud applications.

Organizations are using multiple technologies and services in the attempt to protect their applications. In last year’s survey, we found that the technologies or practices most used by organizations in their security programs were (in order of use): static analysis testing, dynamic analysis testing, pen testing, third-party assessments, application firewalls and virtual patching.

This year we asked organizations to rate which Appsec tools and practices they found the most useful. The tools and practices that ranked the highest include application penetration testing, testing with dynamic analysis (DAST) or vulnerability scanning tools, and using application firewalls to detect or block attacks, as shown in Figure 11.

But organizations are not getting as much value as they should out of other practices, especially virtual patching, secure DevOps, static analysis (SAST) and threat modeling.
Virtual Patching

Virtual patching builds on the effective use of application firewalls, as well as application security testing, and requires the close coordination of Infosec and Operations. It involves setting up an application firewall in blocking mode, testing and finding vulnerabilities in an online application, taking the testing results and creating signatures or rules for the firewall to block attacks against these vulnerabilities, and implementing these rules in production. Virtual patching is intended to be a temporary solution until the development team can fix the code—or for use when the organization doesn’t have access to the code (for example, patching a security vulnerability in commercial third-party software). But it’s time-consuming and difficult to scale, even when using dynamic testing tools and firewalls that are designed to work together.

Secure System Operations/Devops

With continued adoption of Agile development and the demand for faster time-to-delivery, we expect more organizations to take up Devops practices such as “infrastructure as code” and Continuous Delivery or Continuous Deployment, which build on standardized configuration management for infrastructure and applications, automated deployment and fast feedback loops between operations and development. Security checks and balances can—and should—be built into all of the steps involved, from automated security testing in Continuous Integration through to deployment checks and run-time security self-tests (following the example of Netflix’s Simian Army). 7

Static Analysis

While Infosec can run a dynamic scan or pen test on the system and pass the results back to development to be fixed, SAST (scanning source code or binaries for common security vulnerabilities and bug patterns) requires more hands-on involvement from developers because it works directly on the code. Developers have to assist with setup, take the time to review and understand what the tools find and then weed through all of the false positives before they can begin triage, fix bugs and roll out patches.

Although suppliers continue to improve the speed and accuracy of SAST tools and make them easier to use, developers need security training or expert help to understand what the tools are telling them, which vulnerabilities are important, why they need to be fixed and how to fix them. Developers—and managers—need to be convinced that all of this is worth their time. Although bridging the gap between Infosec and development teams and getting developers to use static analysis testing effectively can take time and effort, it can also pay dividends by providing a much faster feedback loop. By running static analysis checks frequently, developers can find out quickly when they have made a mistake—and they can fix the problem while they are still working on the code, rather than waiting days or weeks or months for the results of a penetration test.

Finally, the cost of static analysis tools is an issue for many organizations. Good commercial tools are expensive and are generally out of the reach of all but large enterprises, which account for only 25% of the respondents to this survey.

**Threat Modeling**

Static analysis testing is one way that organizations can solve security problems early in development. Threat modeling is another.

More than 75% of the organizations surveyed design applications in-house. However, only a small percentage of them do threat modeling or find it useful. Threat modeling—understanding and managing security threats in application architecture and design through a structured process that involves developers and security experts working together—demands a significant commitment from the development organization. The shortage of application security skills noted earlier is also a major limiting factor here. It is difficult to find security engineers who understand application design and architecture and application architects who understand security risks in application design.

Organizations need less-expensive alternatives to threat modeling in order to identify and manage application security risks up front. Most enterprises whose main business is not selling software or SaaS cloud services should at least focus on higher-level strategic threat modeling to understand what threat actors will likely target the organization and which applications are likely to be the targets of attack. They can then use this information to prioritize Appsec initiatives across the application portfolio and to build a business case for funding them.

Smaller software development organizations, especially Agile development teams, should adopt lighter weight, incremental approaches to add security risk and threat analysis into architecture and design. Threat modeling, as it is commonly described, is a formal, document-heavy security walkthrough of system design artifacts and does not work well for teams following Agile development practices, where design details are worked out iteratively and incrementally and the design is always in flux. Dr. Gary McGraw, for one, has recently outlined a simpler, more scalable method for application risk assessment called a “Security Architecture Survey.” As he points out, although this kind of analysis is less comprehensive and less robust than more formal techniques, organizations are more likely to do it because this analysis is much less expensive and more scalable.

---

We asked our respondents how frequently they assess the security of their business-critical applications that were in production. Figure 12 shows the frequency of testing reported in this survey.

The frequency at which organizations are doing security testing has increased significantly over the past year, as illustrated in Table 2, which shows our 2012 survey results compared to the 2014 results.

Table 2. Comparison of Testing Results

<table>
<thead>
<tr>
<th>Frequency of Security Testing for Applications in Production</th>
<th>2012</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>No security testing done</td>
<td>13.5%</td>
<td>2.7%</td>
</tr>
<tr>
<td>Only when applications are updated, patched or changed</td>
<td>21.3%</td>
<td>10.1%</td>
</tr>
<tr>
<td>Every year</td>
<td>14.3%</td>
<td>19.5%</td>
</tr>
<tr>
<td>Every three months</td>
<td>18.0%</td>
<td>12.1%</td>
</tr>
<tr>
<td>Once a month</td>
<td>9.5%</td>
<td>8.1%</td>
</tr>
<tr>
<td>Ongoing, continuous testing</td>
<td>23.3%</td>
<td>35.6%</td>
</tr>
</tbody>
</table>

Only a small percentage of the organizations surveyed are not doing application security testing today (2.7%). More organizations are taking advantage of automated testing tools and practices and SaaS testing services to do ongoing, continuous testing. This is especially important where development teams are adopting Agile development methods to make continuous incremental changes to software.
Training in secure software development ranked low in the list of practices that organizations find useful. Figure 13 shows the distribution of secure code training programs.

Slightly fewer than 26% of organizations had ongoing secure coding training programs that were working well or were mandated for all development. But almost half of organizations (41%) have programs that are not consistently implemented or are not consistently being followed, and another 27% did not train developers in secure coding at all.
In rating the effectiveness of their organization’s Appsec programs, approximately 28% felt that their programs were exceptional (3%) or above average (25%). The majority of respondents felt that their programs needed improvement (54%) or even complete rework (10%), as shown in Figure 14.

Breaches Caused by Application Vulnerabilities

The lack of effective Appsec programs is highlighted by the number of organizations that experienced security breaches as a result of application vulnerabilities in the last 18 months. As shown in Figure 15, 29% of responding organizations experienced at least one security breach as a result of application vulnerabilities in the last 18 months, with 14% experiencing 3–5 breaches and 3% experiencing at least 10 breaches.
Most of these breaches were reported by larger organizations. Because of their size, they offer a much larger attack surface, they are generally more interesting targets to nonopportunistic hackers, and they have the resources to detect breaches and to determine the root cause. More small organizations may have been breached because of a software vulnerability without being aware of it, as shown in Figure 16.
Challenges to Implementing an Effective Appsec Program

Many large enterprises (38%) do not have sufficient control over their application portfolios and cannot identify all of the applications that they need to secure. And organizations continue to struggle with creating an effective bridge between security and in-house, outsourced and third-party development (34%). Figure 17 illustrates the results.

Testing makes up the backbone of many application security programs today. The good news is that testing—getting access to the tools and resources to do security testing for new applications and for legacy applications—is not holding organizations back.

But the number one challenge facing most organizations this year, edging out lack of funding and management buy-in, is a lack of Appsec security skills to develop organizational programs and secure production systems (46%).

Figure 17. Implementation Challenges
Plans for Spending on Appsec

Although lack of funding or management buy-in is the second largest challenge facing organizations, the picture may be improving. Respondents indicated that their organizations, in general, expect to spend more on Appsec in the coming year (see Figure 18).

More than half (58%) of responding organizations expect to spend more money on their Appsec programs over the next year: almost 38% expect to spend a bit more; almost 21% expect to spend a lot more. Only a very small percentage (3%) will spend less, and 29% expect no change in funding.

Future Ideas and Roadmap

Finally we asked respondents to list future plans, ideas and technologies that they are looking at to improve their Appsec programs. Most organizations have no clear next steps on the future roadmap. Some are looking at application security practice maturity models like Cigital’s Build Security in Maturity Model (BSIMM),\(^{10}\) OWASP’s OpenSAMM\(^{11}\) or Application Security Verification Standard (ASVS)\(^{12}\) as guidelines. A few are evaluating advanced intrusion prevention systems and cloud-based security offerings. Others are investigating how to use Big Data analytics to support their application security initiatives.

But most organizations are not looking beyond their current set of ideas and tools. They still have a lot of work ahead of them.

---

\(^{10}\) [http://bsimm.com](http://bsimm.com)

\(^{11}\) [www.opensamm.org](http://www.opensamm.org)

Organizations are continuing to invest more in application security. Last year more than one-third of those surveyed did not have an Appsec program in place. More than 80% have formal programs in place, and most of these organizations are doing something about Appsec now or are planning to implement a program in the coming year. More organizations will spend more on application security next year (more than 58% plan to increase spending in the next 12 months.)

So far, however, most of these programs are not proving to be effective. Almost two-thirds of respondents said that their programs needed to be improved, including 10% who said their programs needed a complete overhaul. Almost 29% of the organizations surveyed had experienced one or more security breaches due to an application security vulnerability in the last 18 months, and some (4%) experienced 10 or more breaches.

Organizations continue to rely heavily on looking for security vulnerabilities after the fact (using black box dynamic testing and vulnerability scanning tools and services, as well as pen testing) and blocking these vulnerabilities with application firewalls and intrusion prevention systems. The good news is that organizations are taking advantage of better tools and online services to test their applications for security vulnerabilities much more frequently, even testing continuously, which could dramatically shorten vulnerability windows—if developers can fix the bugs when they are found.

The bad news is that organizations are not attacking the root cause of application security problems—stopping developers from writing insecure software in the first place. Developers continue to create security holes because they don’t understand enough about secure design, threat modeling and secure coding practices. Developers aren’t taking enough advantage of static analysis tools to catch security bugs early (when they are less costly to repair), while they are still working on the code, because they don’t understand what the tools are telling them. They aren’t leveraging security libraries and the security features of their frameworks to reduce risks and costs because they—and their managers—don’t know that it is important.

A lack of knowledge and skills is holding back Appsec programs today, and it is preventing organizations from making real progress in Appsec in the future. The number one obstacle to success reported in this year’s survey is a shortage of skilled people, part of a bigger problem facing the IT security industry in general, as recent studies by Forrester Research13 and (ISC)214 show.

Training and education are needed to address this skills shortage—not just training more Infosec and Appsec specialists, but training developers and managers, too. Fewer than one-quarter of respondents have training programs that are ongoing and working well, and secure coding training ranks low in the list of practices that organizations depend on in their Appsec programs today. This needs to change.

There aren’t any next generation tools or other silver bullets on the horizon that will solve the problem of secure software. Writing secure software is about fundamentals: thoughtful design, careful coding, disciplined testing and informed and responsible management. The sooner that organizations understand this—and start doing it—the sooner they will solve their security problems.

13 www.informationweek.com/traffic-management/security-skills-shortage-or-training-failure/d/d-id/1105895
**About the Authors**

**Jim Bird** is an application development manager and CTO with more than 25 years of experience in software engineering, with a special focus on high-integrity and high-reliability systems. Jim is currently the co-founder and CTO of a major US-based institutional trading service, where he is responsible for managing the company’s technology organization and information security programs. Jim has worked as a consultant to IBM and to major stock exchanges and banks globally. He was also the CTO of a technology firm (now part of NASDAQ OMX) that built custom IT solutions for stock exchanges and national clearinghouses in more than 30 countries. Jim is an active contributor to OWASP, helps out with the SANS Appsec blog and blogs on Agile software development, project management and application security topics at “Building Real Software.”

**Frank Kim** is a security leader with more than 16 years of experience in information security, risk management and enterprise IT. He has a passion for developing security strategies and building teams focused on practical solutions to business risks. He currently serves as the curriculum lead for application security at the SANS Institute and is the author of the “Secure Coding in Java” course. Frank is a popular public speaker and has presented at security, software development and leadership events around the world.

**Sponsors**

*SANS would like to thank this paper’s sponsors:*

![HP Logo](logo.png)

![Qualys Logo](logo.png)

![Veracode Logo](logo.png)
## Upcoming SANS App Sec Training

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS San Francisco Spring 2020</td>
<td>San Francisco, CA</td>
<td>Mar 16, 2020 - Mar 27, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS 2020</td>
<td>Orlando, FL</td>
<td>Apr 03, 2020 - Apr 10, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>Mentor Session - DEV522</td>
<td>Novi, MI</td>
<td>Apr 21, 2020 - May 21, 2020</td>
<td>Mentor</td>
</tr>
<tr>
<td>SANS Amsterdam May 2020</td>
<td>Amsterdam, Netherlands</td>
<td>May 11, 2020 - May 18, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Silicon Valley - Cupertino 2020</td>
<td>Cupertino, CA</td>
<td>Jun 22, 2020 - Jun 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Copenhagen August 2020</td>
<td>Copenhagen, Denmark</td>
<td>Aug 24, 2020 - Aug 29, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS Network Security 2020</td>
<td>Las Vegas, NV</td>
<td>Sep 20, 2020 - Sep 27, 2020</td>
<td>Live Event</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>