THE DZONE GUIDE TO
APPLICATION AND DATA SECURITY
VOLUME II

RESEARCH PARTNER SPOTLIGHT
Key Research Findings

517 software professionals completed DZone’s 2016 Security survey. Respondent demographics include:

- 74% of respondents identify as developers/engineers (42%) or development team leads (32%).
- 72% of respondents have 10 years of experience or more as IT professionals. 46% of respondents have 15 years or more.
- 41% of respondents work at companies headquartered in Europe; 30% work in companies headquartered in the US.
- 22% of respondents work at companies with more than 10,000 employees; 27% work at companies between 500 and 10,000 employees.
- 60% identify as developers or developer team leads.

WITH GREAT POWER...

50% of survey respondents said that primary responsibility for application security should be in the hands of developers, over security teams (29%) and frameworks. This is in keeping with the results of our 2015 security survey, in which 53% of respondents put security on developers first. Furthermore, respondents in development roles (74%) skewed slightly above the overall average of respondents who believe that developers are primarily responsible for AppSec, showing a broad understanding of the importance of security throughout the development process and a resistance to finger-pointing and scapegoating in the SDLC.

APPSEC BAG OF TRICKS

While there is no single right answer for how you should develop a secure application, there are some definite trends among our respondents regarding the practices, techniques, and tools they use. For application architectural patterns, the use of roles and sessions were both very popular among our respondents, with 77% and 75% claiming to use these patterns, respectively. For verifying message integrity, 68% of respondents said they use authentication tokens (including digital signatures). 70% of respondents said they use the OpenSSL toolkit for implementing encryption in their applications. And in terms of secure coding 64% of respondents said they consider security while architecting and designing an application, while 90% said they use input validation to help secure their application.

SECURITY TESTING HAS BEEN WEIGHED, HAS BEEN MEASURED, AND HAS BEEN FOUND WANTING

What is the main type of application security testing you do? Penetration testing and security code review were popular responses among our survey respondents this year, with 24% and 19% of responses, respectively. However, about one in six respondents said they have no formal security testing at all. Even when applications are tested for security, that testing may be inadequate. When asked how sufficient testing is determined for their applications, 15% of respondents answered “the attack surface is addressed to some acceptable level.” 15% said “the threat model is handled in an acceptable way,” and 22% responded that both of these occur. But 48% of respondents answered that “security testing is never sufficiently covered.”

SECURITY TRAINING COULD USE SOME WORK, TOO

When asked how frequently developers in their organization are trained in security, 27% of survey respondents said that no such training occurs. 33% of respondents said

---

**WHAT IS THE MAIN TYPE OF APPLICATION SECURITY TESTING YOU DO?**

- Penetration Testing: 24%
- Security Code Review: 19%
- Vulnerability Assessment: 13%
- No Formal Security Testing: 16%
- I Don’t Know: 4%
- White Box Testing: 2%
- Black Box Testing: 6%
- Source Code Analysis: 15%
- Dynamic Analysis: 2%

**OF THE APPLICATIONS YOU TEST FOR SECURITY, HOW IS SUFFICIENT TESTING DETERMINED?**

- The attack surface is addressed to some acceptable level: 15%
- The threat model is handled in an acceptable way: 14%
- Security testing is never sufficiently covered: 48%
- Both 1 and 2: 22%
that security training occurred in an ad-hoc manner in their organization, leaving 39% of respondents whose organizations have organized security training on a yearly, or more frequent, basis. Of the training discussed, organizations lean towards completely custom training: 66% of respondents said that their organization uses custom training materials, while 46% say that the OWASP Top 10 is used, and only 16% say that the SANS Top 25 is used for training purposes within their organization.

**SECURITY STILL TAKES A BACK SEAT TO PERFORMANCE**
Last year, between performance, maintainability, scalability, and security, respondents on average ranked security as third in terms of importance, behind performance and maintainability as numbers one and two. This year, security passed maintainability, but still lags behind performance in respondents’ priorities. The gap, however, is closing as the importance of application security is increasingly acknowledged, and the gap in rank distribution between performance and security responses has shrunk considerably from last year.

**TO RELEASE OR NOT TO RELEASE...**
On average, respondents to our 2016 Application Security survey said that 20% of application releases contained known security vulnerabilities, with a median response of 10%. This aligns closely with last year’s responses, which estimated an average of 22% of releases containing known security vulnerabilities with a median response of 13%. 78% of respondents said that application security could override security concerns with half of respondents saying that this occurred either sometimes (30%), often (14%), or all the time (5%), rather than rarely or never. 10% of respondents did not know how often application security was overridden by release schedules. Also, 50% of respondents said that security patches had to be released once every two months or more frequently, while 30% of respondents said that security patch release was not applicable to their application/situation.

**WEB APPS ARE CONCERNING**
We asked respondents to rate how worried they were about security among several popular programming languages, on a scale from 1 to 10 (10 being maximally worrying). Results showed that the most concerning languages were JavaScript, with an average concern rating of 6.9 out of 478 responses, and PHP, with an average concern of 5.8 out of 341 responses (median responses were 8 and 7, respectively). Third most concerning was Java, with an average concern rating of 4.6 out of 504 responses. The inherent interconnectivity of web apps makes some sense of these concerns, but interestingly, languages used primarily for mobile applications, such as Objective-C and Swift, had very low ratings of concern (each had an average rating of 3.5 out of 298 and 293 responses, respectively, with median responses of 3).

**HOW OFTEN DO RELEASE SCHEDULES OVERRIDE SECURITY CONCERNS?**

**COMMON SECURITY PATTERNS, TECHNIQUES, AND TOOLS**

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Technique/Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSSl: Encryption API</td>
<td>Authentication tokens: message integrity verification</td>
</tr>
<tr>
<td>Secure architecting and designing: coding techniques</td>
<td>Sessions: architectural pattern</td>
</tr>
<tr>
<td>Input validation: coding techniques</td>
<td>Roles: architectural pattern</td>
</tr>
</tbody>
</table>
Developers have become increasingly aware and concerned about security threats. In fact, according to this year’s survey, over half of the respondents noted that developers are primarily responsible for security over frameworks and security teams. One of the first steps in building secure applications and preventing any attacks is to know what types of attacks there are.

We asked our audience what their top security threat concerns were. Drawing from over 1,000 responses, we broke down the top 7 security threats, defined them, and suggested actions to take to defend your kingdom against these attacks.

**PHISHING** 43%
Phishing is an attempt through email, chat, or social networks to trick users into divulging information like passwords or credit card numbers by posing as a legitimate source, such as a bank.

**DEFENSE:** Education around spotting phishing attempts and browser certificates are the best ways to counteract phishing.

**SQL INJECTION** 49%
An attack where harmful SQL statements will be entered into a data entry field, then executed.

**DEFENSE:** One of the most basic defenses is to use parameterized statements that create placeholders out of an entry that cannot be executed as SQL statements.

**TROJANS** 26%
A program that disguises itself as a trusted application, when in fact it harbors malicious code.

**DEFENSE:** Education—because Trojans need your permission to run on your computer, always be wary of opening files when you don’t know their source.

**DDoS** 46%
A Distributed Denial of Service attack occurs when multiple systems flood the bandwidth of a target system, such as a web site, often coordinated by using computers infected with a virus to automatically ping the site.

**DEFENSE:** Using automatic traffic pattern analysis technologies to identify threats, traffic scrubbing filters, and using cloud provider solutions can help mitigate attacks.

**XSS** 37%
Cross-site scripting is when an attacker injects client-side scripts into web applications viewed by other users, so the web site attacks its own users without the knowledge of the site owners.

**DEFENSE:** Using output escaping techniques to cover XML significant characters, disabling scripts, and using HTML sanitation engines can all prevent XSS attacks.

**CSRF** 30%
An attack that tricks an end user into executing unwanted actions on a web application in which they're currently authenticated.

**DEFENSE:** For an automated defense, check the standard headers to verify the request is same origin and check the CSRF token. For a manual defense, require user interaction for authorizing transactions (re-authentication, etc) or, less intrusively, use transaction IDs.

**MAN-IN-THE-MIDDLE** 31%
When an attacker is secretly intercepting and potentially subverting communication between two parties who think that they are in direct communication with one another.

**DEFENSE:** There are cryptographic protocols with various forms of endpoint authentication specifically to prevent MITM attacks. Examples include public key infrastructure (like TLS), secure DNS extensions, etc.
Agile and Secure?

Development and Security CAN Work in Peace and Harmony

There has been a lot written about DevOps and Continuous Delivery practices and how to apply these practices to software development to improve efficiencies. With the growing momentum around adding secure coding practices into the mix, White Hat Security has some recommendations on how static (SAST) and dynamic (DAST) application security testing solutions can help teams achieve their objectives, while also saving time and resources and reducing the security risk of deployed apps.

1. **Reduce the time required from definition complete to code release.** Model application development methodologies such as Agile, Scrum, or Kanban focus on releasing software early and often. The old days of PRDs and Waterfall approaches have been replaced with user stories as inputs to development. The best user stories clearly address any assumptions and constraints, as well as acceptance criteria. With this in mind:

   A. Define application security requirements (e.g., compliance to OWASP Top Ten, SANS 25, PCI-DSS, etc., or “no vulnerabilities rated medium or high risk”) up front.

   B. Implement static application security testing (SAST) early in the development process and continuously fix security vulnerabilities as they are identified to help ensure that security requirements are met prior to code release.

2. **Improve the quality and speed of feedback for developers.** Developers need to get feedback on which software flaws and security vulnerabilities are present in their code, ideally as they are developing it, not months later when they’ve forgotten exactly what they did.

   A. Make sure that your AppSec testing solutions have plugins for bug trackers (e.g., Jira), so that vulnerabilities in scanned code are automatically synchronized and appear as tickets in Jira.

   B. Check whether plugins for continuous integration build tools, such as Jenkins, are available, so that automated security scanning of websites (DAST) and code (SAST) can be scheduled as often as needed as part of an Agile workflow.

   C. Use both SAST and DAST AppSec solutions to ensure maximum test coverage. Use DAST for continuous scanning at pre-production and production phases. Use SAST for development and QA phases. All identified security vulnerabilities should be input into Jira and tracked.

3. **Reduce rework.** “Fixing security bugs at design time costs 1/60th of what it costs to fix the same bugs with a patch after the release.”

   A. Use a SAST solution to not only save time, money, and resources, but also to reduce the security risk of your deployed apps. WhiteHat Security has seen instances where customers have experienced a 60% decrease in the number of Cross-Site Scripting (XSS) vulnerabilities detected in their deployed production apps, if they use a SAST solution during app development.

---

Synopsys is Setting the Standard for How Companies Create and Secure Software

Synopsys provides tools that are the standard for software development, testing, and procurement processes in 19 of the top 25 software companies, 11 out of the top 15 Automotive OEMs, and in organizations across many different industries. Our end-to-end offering gives companies a turnkey, scalable approach for minimizing software-related business risks, maximizing release predictability and speed, and ensuring internal and external standards compliance.

We’re also involved with software standards bodies (FDA, SAE, and UL, just to name a few), helping to define next generation software quality and security testing. Today, “good enough” just isn’t good enough anymore, especially when it comes with the daunting prospect of recalls, updates, and emergency patches. Our fast and accurate platform of testing tools fits within your existing Software Development Life Cycle, works well with Agile methodologies, and is embraced by developers worldwide.

Our end-to-end offering gives companies a turnkey, scalable approach to ensure internal and external standards compliance.

And given that up to 90% of software consists of code obtained through third parties (the cyber supply chain), our Synopsys Software Integrity Platform identifies third-party components and their vulnerabilities, manages open source licenses and copyright issues, and secures the intellectual property inside your applications and firmware.

Written by Robert Vamosi
CISSP, Security Strategist, Synopsys

---

**Synopsys**

Synopsys, Inc. provides the solutions needed to deliver smart, secure products for the era of connected everything.

**Category**
Application Security

---

**New Releases**
- Synopsys Releases Coverity 8.5 Static Analysis Tool
- Synopsys Releases Seeker 3.8 Runtime Security Analysis Tool for Web Applications

---

**Strengths**
- Automated Security Testing for Agility
- Advanced Protection of Sensitive Data
- Unmatched Accuracy
- Clear Paths to Remediation

---

**Case Study**
Parkeon is a key player in the urban mobility sector and a global provider of parking and transport management solutions. Parkeon offers a unique range of parking control and payment services in 55 countries and more than 3,000 cities around the world. While using Synopsys Seeker, Parkeon has identified three key benefits that demonstrate that it is the tool for them. First, Seeker ensures that the entire system, end to end, complies with security standards such as PCI-DSS by understanding how data flows throughout the entire application. It identifies vulnerabilities in relation to their impact on sensitive data. Second, Seeker facilitates communication between the test and development teams by linking vulnerabilities back to the offending source code. Unlike other dynamic testing tools which report vulnerabilities by the offending URL, Seeker automatically ties those vulnerabilities back to the source code where the fix needs to be applied. And third, Seeker improves security awareness and helps train developers for more secure coding practices. Parkeon’s developers and testers are trained on the basis of OWASP TOP10, but they are not information security experts. By providing a replay of every attack, explaining the business risks and providing relevant remediation suggestions, Seeker helps their test and development teams to acquire awareness and training in an ongoing manner, thus improving the security of their code.

---

**BLOG** blogs.synopsys.com/codereview
**TWITTER** @SW_Integrity
**WEBSITE** synopsys.com

---

Synopsys is Setting the Standard for How Companies Create and Secure Software
Setting the Standard for Application Security

Speed
Fast results combined with unmatched breadth, depth, and accuracy of analysis.

Trust
Our tools are widely adopted by developers, the security community, and standards groups.

Insight
A unified view of overall software health that supports total risk visibility.

Flexibility
Our solutions are designed to scale as your business needs change.

Visit www.synopsys.com/software or call 1(800) 541-7737 to learn more.