FORRESTER®

FOR APPLICATION DEVELOPMENT & DELIVERY PROFESSIONALS

Brief: App Security Can't Happen Without Developers

Use A Combination Of People And Tools To Deliver Secure Apps

by John M. Wargo

May 26, 2016

Why Read This Brief

Application development and delivery (AD&D) professionals must keep security front and center when they're crafting apps that run outside of the firewall. Otherwise, they risk finding their brands plastered across the headlines and their customers fleeing in droves. A multitude of tools exists to help validate app security post development, but that's not enough. In this report, we'll show that a dev team's awareness and skills in delivering on its app's security requirements are critical from the very beginning regardless of the security tools it uses.

Key Takeaways

You'll Need More Than Tools To Deliver Secure Apps
Security professionals use code-scanning tools to find security vulnerabilities, but that's not enough. Couple tools with developer-focused security programs, draft clear security standards, and deliver frequent training sessions to drive security-focused behavior among developers.

Mitigation Costs Skyrocket When Security Issues Survive The Development Process
Mitigation costs grow exponentially the longer it takes a development team to identify and fix a security issue during the development process. Defining a solid app security architecture during the design phase and coupling it with accurate development against that design are critical components of the secure code delivery process.

Watching The OWASP Top 10 Doesn't Help When Your App Is Attacked With No. 11
Most of the companies we spoke to trained developers on the Open Web Application Security Project (OWASP) Top 10 and stopped there. We agree that you have to cover the top 10, but developers must mitigate still more risks through secure code.

Brief: App Security Can't Happen Without Developers

Use A Combination Of People And Tools To Deliver Secure Apps

by John M. Wargo
with Martin Gill, Jeffrey S. Hammond, Michael Facemire, Christopher McClean, Jaclyn Galan, and Peter Harrison

May 26, 2016

Avoid Adverse Headlines Through A Solid App Security Plan

We all know the stories: A security flaw makes it into an app, and suddenly your brand is in the headlines and your customers are scurrying toward safer pastures. App security tool vendors are happy to share stories about the stupid developer tricks that lead to spectacular breaches of personal data as they pitch their products. There are well-documented strategies and tools that AD&D pros employ to help secure their apps; you know this, and you've implemented many of them in your development process, right?[1] To reduce the risk of security issues making it into the wild, many software development organizations:

› **Define reasonable security standards developers can live with.** Security standards are important, but they have to be both comprehensible and reasonable to apply in real-world situations. Developers must be capable of understanding and implementing these standards consistently across all apps that your organization delivers. As a security architect at a major bank said, "Never underestimate developers to take the easy [insecure] route."

› **Verify security implementation using app and code scanners.** A Reagan era "trust-but-verify" approach ensures that developers have implemented the security standards as you've documented and catch common vulnerabilities with suspect code sent back to development for repair.[2] Unfortunately, these tools are only a patch to the problem. App scanning only applies to the last mile of a long journey — after the development team invested time and energy to create the vulnerability in the first place.

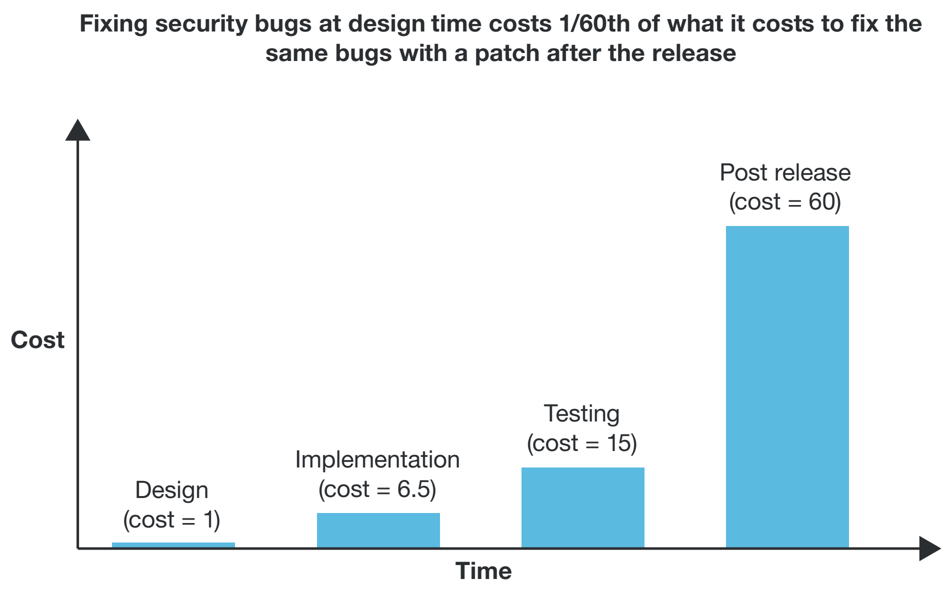
› **Deliver relevant security training for developers.** Don't assume that developers working for you understand the business impact of the security vulnerabilities in their code. As Ty Rollin, chief technology officer (CTO) of Mobiquity, said, "Most of our customers do scans and nothing else," and that's not enough. Code scanners will identify possible vulnerabilities, but they don't teach developers how to code securely. Provide directed education to ensure that you've set the proper expectations and associated knowledge with developers.

FORRESTER®
Forrester Research, Inc., 60 Acorn Park Drive, Cambridge, MA 02140 USA
+1 617-613-6000 | Fax: +1 617-613-5000 | forrester.com

Catch Security Issues Early — It Costs More To Fix Them Later

There's measurable benefit in catching issues earlier, as research from IBM has shown (see Figure 1).[3] The cost of implementing security fixes increases exponentially the longer an issue lives through the development process. This correlates directly with overall development effort and resource cost to fix the specific vulnerability plus any work that you must do to mitigate any issues that the team created in a cascading effect through the fix. Remediation efforts are more effective when you apply them during development than when you catch them later through security scans. The good news is that an Agile development process helps reduce the time-to-fix. As a vice president of security at a large insurance company told us, "Moving from waterfall to Agile delivers defects in much smaller code units; our implementation of continuous integration enables us to address security issues sooner."

FIGURE 1 The Cost Of Fixing Security Vulnerabilities Grows Exponentially Through The Development Process



Source: Mark G. Graff and Kenneth R. van Wyk, *Secure Coding: Principles & Practices*, O'Reilly Media, 2003

Cultivate A Security Culture Among Your Developers

Security tools are a critical part of any security architecture, but there's a human side to the equation as well. Developers have their hands deep in the code, so it's their skills (or lack thereof) being tested by these tools. Help push security to the left by enhancing developers' security skills and increasing their awareness of security issues and their fixes.

Get A Leg Up On Security Issues With A Solid Security Methodology

The way to implant security in a developer's brain is to make security part of the complete, end-to-end development process. As Aristotle said, "It is frequent repetition that produces a natural tendency," and developers are no different from anyone else in this respect. Eli Feldman, CTO of advanced technology at EPAM Systems, told us, "Security should never be an afterthought." Start with security during the design phase, reinforce it throughout the development process, and:

› **Lock down security architecture before you do anything else.** Define security architecture at the beginning of the project. Involve all of the stakeholders to ensure you understand all of the security requirements for the app — front end and back. After development has started, architecture changes could stall the project. Ty Rollin, CTO of Mobiquity, said, "Changes to security requirements could force a rearchitecture of the app and changes in underlying technologies," and that means wasted effort and project delays.

› **Make security requirements part of user stories — put everything in one place.** Avoid surprises, and lay the groundwork for all of the work that follows. Developers use user stories to guide their development efforts, and they need the complete story as they're planning development; otherwise, they'll spin their wheels trying to get everything right. OWASP advises developers to include evil user stories in their backlogs — stories that describe all the ways evil users (hackers) may try to break into your app — to drive them to address all security risks when delivering new features.[4]

› **Maintain a close working relationship with the security team.** Involve security and risk professionals in the initial app security architecture definition, security feature design sessions, and subsequent sprint reviews to streamline the process. With this approach, teams address security requirements more efficiently rather than collaborating only when things break. The security team's involvement shouldn't interrupt development; as a security architect at a major bank told us, "The development team learned that delivery timelines would not be in jeopardy if they worked closely with my team."

Forge Critical Developer Security Skills — Rinse, Then Repeat

Don't hope; ensure that your developers have the secure development skills they need for the environment they're working in. Even if the developer has strong security skills, you can't know if they'll apply accurately to the technologies you use. A security architect at a major bank told us, "Assume every time you change technology, the newness of the tech blinds you to the sameness of the patterns. As a result, you see the same flaws all over again." To foster the appropriate security capabilities in your development organization, you should:

› **Deliver developer security training more than once yearly.** Each development organization we interviewed for this report requires some sort of developer security training — often when developers first join the organization — and only some repeat yearly. In a dynamic threat landscape with developers regularly adopting new technologies in their work, developers need more frequent training. Professional services firm Mobiquity recognized that its yearly approach wasn't imparting the necessary security skills on its development team, so it requires attendance in a refresher course before the start of each project.

› **Recognize that the level of security awareness varies based on experience.** Expecting all developers to execute at the same level won't work. Acknowledge skill differences, assign work accordingly, and then implement a structured plan for promotion of lower-level developers to the next level. A large insurance company implemented a belt system for highlighting developer security capabilities with a clear path between each; a yellow belt has completed three security training classes, a brown belt made a measurable difference at the team level, while black belts made a difference at the enterprise level.

"Assume every time you change technology, the newness of the tech blinds you to the sameness of the patterns. As a result, you see the same flaws all over again."

› **Foster security competition between development teams.** There's nothing like a bit of competition to get developers motivated. Use security success as an incentive for developers to compete against other development teams. One AD&D professional at a US insurance company created a cross-team security dashboard highlighting vulnerabilities found and published it internally for all dev teams to see; the result was teams that were self-motivated to stay on the right end of the scorecard.

› **Trust but verify — periodically assess developer security knowledge.** Just because a developer has sat through the requisite training classes doesn't mean he or she can apply the concepts in practice. Verify developer knowledge either through periodic assessment tests or, to make things more interesting, by starting a "find-the-security-bug" competition between developers or development teams. Be sure to publish the correct answer later!

Create A Self-Sustaining Environment By Helping Developers Help Themselves

Formal developer education ensures coverage of the specific topics that are critical to your company's app security requirements. There's a wealth of knowledge in your developers' heads that would simply be too hard to collect into a training class, and, since the knowledge continues to grow, courses would never be complete. The security landscape and threat mitigation strategies change regularly, so developers have to keep up somehow. To accommodate this, enable free outflow of your existing developer knowledge and:

› **Spread the wealth — promote security best practices across development teams.** Structured training has its place, but variety is important as well. More experienced developers or security team personnel have insights that won't be available in formal training classes. Create a forum for information sharing that is available to all developers such as internal wiki pages, brown-bag lunches, and newsletters. One service provider, for example, tracks when developers implement interesting solutions and asks them to write up case studies and present them to a wide audience of developers.

› **Pair new developers with experienced ones when they join a team.** Streamline the onboarding of new developers by pairing them with more experienced ones to show them the ropes. For example, open source project leader and app server provider Nginx must ensure consistency in the code it delivers to market and uses developer pairing to accomplish this. New developers work side by side with more experienced developers until they're ready to stand on their own.

› **Implement code reviews to get an extra set of eyes on all new code.** It never hurts to get an extra set of eyes on your code. Many development organizations mandate code review at some part of the development cycle, but, at a minimum, you should require code reviews for vulnerable areas of your apps' code.[5] Not only does this help you identify issues before they make it further through the process, but it also educates the source developer every time these reviews identify a potential problem in the code. Consider code reviews as a relatively inexpensive continuing education program for your developers.

Recommendations

Focus On Developers To Reduce The Cost Of Fixing Security Flaws

Developers are a core component of any app security implementation. Use code and app scanners to validate their efforts, or, for native apps, use security wrappers to provide an extra level of security in the wild. But tools alone aren't the complete solution. When attention to security starts at the beginning of a project, security is more consistent, and it's less expensive to fix issues that arise. As an application development leader, you can ensure the continued success of your app security implementations when you:

› **Don't focus solely on the top vulnerabilities.** Most developer training programs focus on the OWASP Top 10 Project's list of the most critical web application security flaws.[6] If you're going to start anywhere, that's a great place, but recognize that hackers might still hit you using No. 11. Use commercial or home-grown courses on the top 10 to create a baseline of security knowledge among your developers. Next, use a periodic risk assessment or industry-specific research to identify additional attack vectors and train accordingly.

› **Educate developers on platforms' specific vulnerabilities.** Each target platform has specific architecture or implementation flaws and, therefore, its own set of vulnerabilities. Beyond requisite base security training for developers, add courses for individual development areas that deal with platform-specific scenarios. With mobile, for example, the OS vendor may have built a secure environment for your app to run in, but the extra apps and services that device manufacturers added to differentiate themselves against other vendors may add new vulnerabilities you'll have to deal with.[7]

› **Make implementing security as easy as possible for developers.** Beyond the strategies we outline in this report, there is still more you can do to streamline application security implementation. Use commercial or home-grown security wrappers that limit what an app can do in the wild or let you know when attacks happen. Deliver standard libraries that solve common app security problems for developers, and consider using a resource like InnerSource to create and maintain them in-house to drive better adoption.[8]

Engage With An Analyst

Gain greater confidence in your decisions by working with Forrester thought leaders to apply our research to your specific business and technology initiatives.

Analyst Inquiry

Ask a question related to our research; a Forrester analyst will help you put it into practice and take the next step. Schedule a 30-minute phone session with the analyst or opt for a response via email.

Learn more about inquiry, including tips for getting the most out of your discussion.

Analyst Advisory

Put research into practice with in-depth analysis of your specific business and technology challenges. Engagements include custom advisory calls, strategy days, workshops, speeches, and webinars.

Learn about interactive advisory sessions and how we can support your initiatives.

Supplemental Material

Companies Interviewed For This Brief

EPAM Systems	Mobiquity
IBM	Nginx

Endnotes

[1] Application security is a multidimensional and multivariate problem — one that's not amenable to simplistic and one-dimensional tools or techniques. Technologies like static application security testing, dynamic application security testing, and hybrid analysis promise to streamline the software development life cycle, making application security process and information analysis more effective. In isolation, none of these tools is a cure-all for application security

problems. Traditional solutions to application security no longer work, and technologies with new techniques and tools are cropping up to better protect applications from external threats. See the "TechRadar™: Application Security, Q2 2015" Forrester report.

[2] Security pros are looking to work with vendors to detect vulnerabilities more efficiently and effectively, to avoid exposing business-critical security issues, as well as to improve overall business and development efficiency. The vendors we've assessed meet the growing demand across all businesses without compromising the security needs of their clients' operations. After examining the current trends in the market, user needs assessments, and vendor and expert interviews, we developed a comprehensive set of evaluation criteria for application security. See the "The Forrester Wave™: Application Security, Q4 2014" Forrester report.

[3] This data is from the IBM Systems Sciences Institute's statistics. Source: Mark G. Graff and Kenneth R. van Wyk, Secure Coding: Principles & Practices, O'Reilly Media, 2003.

[4] In "Agile Software Development: Don't Forget EVIL User Stories," OWASP explains that each sprint must include specific coverage for "authentication, session management, access control, input validation, output encoding/escaping, cryptography, error handling and logging, data protection, communication security, and HTTP security features" for every backlog item. Source: "Agile Software Development: Don't Forget EVIL User Stories," OWASP (https://www.owasp.org/index.php/Agile_Software_Development:_Don't_Forget_EVIL_User_Stories).

[5] It's important to closely review security-related code (e.g., how an application authenticates users and systems, how it processes data, etc.), but there are also a lot of general application functions that are not specifically security-related that could introduce major security issues if you do them incorrectly (e.g., input validation).

[6] The Open Web Application Security Project maintains a list of the top 10 security threats facing web applications. Source: "Category:OWASP Top Ten Project," OWASP (https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project).

[7] Source: Lei Wu, Michael Grace, Yajin Zhou, Chiachih Wu, and Xuxian Jiang, "The Impact of Vendor Customizations on Android Security," Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS 2013), November 2013 (https://www.csc.ncsu.edu/faculty/jiang/pubs/CCS13.pdf).

[8] InnerSource utilizes the best practices from open source development projects and applies them to internal development efforts. When developers are forced to use home-grown libraries for their development efforts, letting them have a hand in crafting and maintaining those libraries increases buy-in and helps the teams involved collaboratively deliver better code. Source: InnerSource Commons (http://paypal.github.io/InnerSourceCommons/).

FORRESTER®
CHALLENGE THINKING. LEAD CHANGE.

We work with business and technology leaders to develop customer-obsessed strategies that drive growth.

PRODUCTS AND SERVICES

› Core research and tools
› Data and analytics
› Peer collaboration
› Analyst engagement
› Consulting
› Events

Forrester's research and insights are tailored to your role and critical business initiatives.

ROLES WE SERVE

Marketing & Strategy Professionals	Technology Management Professionals	Technology Industry Professionals
CMO	CIO	Analyst Relations
B2B Marketing	› Application Development & Delivery	
B2C Marketing	Enterprise Architecture	
Customer Experience	Infrastructure & Operations	
Customer Insights	Security & Risk	
eBusiness & Channel Strategy	Sourcing & Vendor Management	

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866-367-7378, +1 617-613-5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.