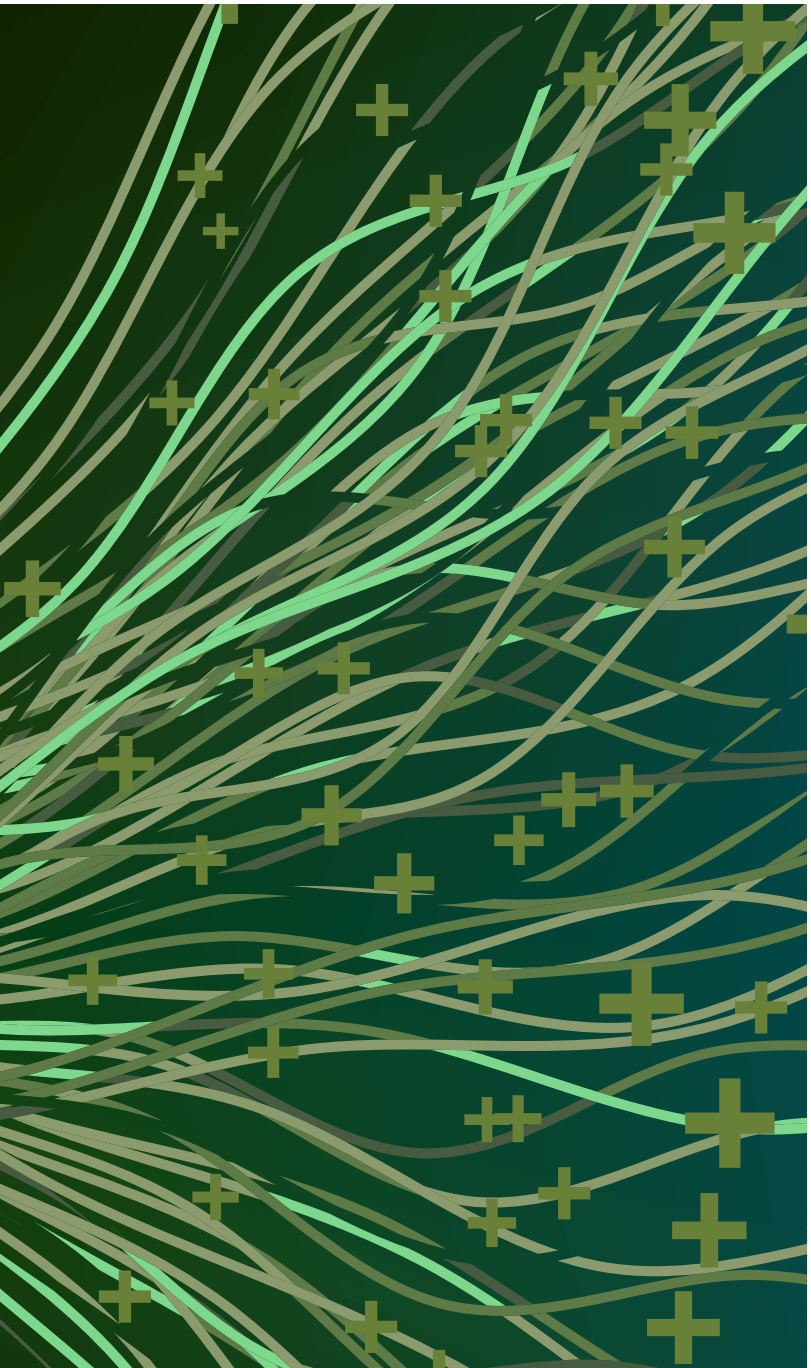
SearchSecurity



E-Guide

UPDATE YOUR APPLICATION SECURITY POLICY AFTER HEARTBLEED

TechTarget

SearchSecurity

ORRIED ABOUT THE stability of your software security? Lower your risk by rewriting policy and procedures for development with open source and third-party components.

SearchSecurity

UPDATE YOUR APPLICATION SECURITY POLICY AFTER HEARTBLEED

Michael Cobb

The CISO job is under the microscope. Security officers have to juggle the ongoing challenges of maximizing security initiatives across their organizations to ensure information assets are adequately protected. The majority focus on making certain human access to key data and resources is strictly controlled. But the same level of attention is rarely paid to the software that manages access to that data -- the code used in encryption, authentication and permission checks.

Faced with resource constraints, enterprises increasingly take advantage of open source libraries and third-party components to develop complex applications -- why rewrite functionality that already exists? But few vet this code with the same rigor that's demanded for internally produced software.

Applications and databases maintained by longtime employees who have since left the company and software systems inherited through mergers and acquisitions also pose significant security risks. Some IT teams may assume

that code is secure because someone else has completed the task of vetting it for flaws and bugs. As the Heartbleed flaw in the OpenSSL cryptographic software library showed, relying solely on others to correctly implement and deliver security can put enterprise and customer data at risk.

What is the security officer's role in managing the risks involved in using open source and third-party software? Application vulnerabilities ranked highest among the threat concerns of 72% of the 1,634 security executives surveyed in 2013, according to "The View From the Top: (ISC)2 Global Information Security Workforce CXO Report," and lowest in terms of time spent. Only 7% of respondents spent a significant amount of time on software security. Compliance with software security policies is distressingly low, similar to enterprise mobile and bring your own device requirements. What impact can revised security policies have on development practices that open the company up to liability and vulnerability issues?

RELEASE EARLY AND OFTEN

The dilemma CISOs face is ensuring code used within software projects is secure without incurring the wrath of business owners and development teams who are under pressure to deliver applications and updates on time and within

tight budgets. As the threats increasingly outweigh the benefits of insecure software, security officers need to reevaluate the risks involved in using open source software and third-party components and how best to manage them.

The security and quality of open source software -- two reasons enterprises choose to use these libraries and components -- is dependent on each project's developer base being large enough that any bug and fix is eventually obvious to someone. This "given many eyeballs, bugs will be shallow" concept is known as Linus's Law, after Linus Torvalds, the creator of Linux and an early proponent of the self-correcting, community-driven software development model. A problem with this philosophy in today's threat landscape is that the financial incentives for discovering and using exploits are much higher than the rewards for finding, publishing and fixing open source software vulnerabilities.

Financial rewards have been offered for open source security initiatives. The Internet Bug Bounty sponsored by Microsoft and Facebook rewards hackers who contribute to a more secure Internet by submitting vulnerabilities found in key open source software such as PHP, Perl and Apache httpd. Matthew Green, a computer science professor at Johns Hopkins University, and others behind the community-funded audit of the TrueCrypt disk-encryption utility, listed a bug bounty program as part of the planned security audit before

the open source encryption project was suddenly shut down by its developers because of unnamed “security vulnerabilities” in May.

Revelations about the National Security Agency’s surveillance activities has caused many -- including China and other nation states -- to suspect open source projects and U.S. technology companies of deliberately introducing bugs into popular security protocols and functions to provide the NSA with backdoors. RSA (EMC Corp.’s security division), Microsoft and TrueCrypt are among those that have faced scrutiny.

Enterprise development teams can easily use a hundred or more different open source libraries, frameworks and tools, along with code snippets copied off the Internet, when building an application. The 2014 Sonatype Open Source Development Survey found that 90% of a typical application is assembled with open source components, many of which contain known security flaws. The problem of vulnerable components incorporated into new applications has become so acute that it appears in the latest OWASP Top 10 List of Web application vulnerabilities.

SearchSecurity

BROKEN POLICIES, VULNERABLE SOFTWARE

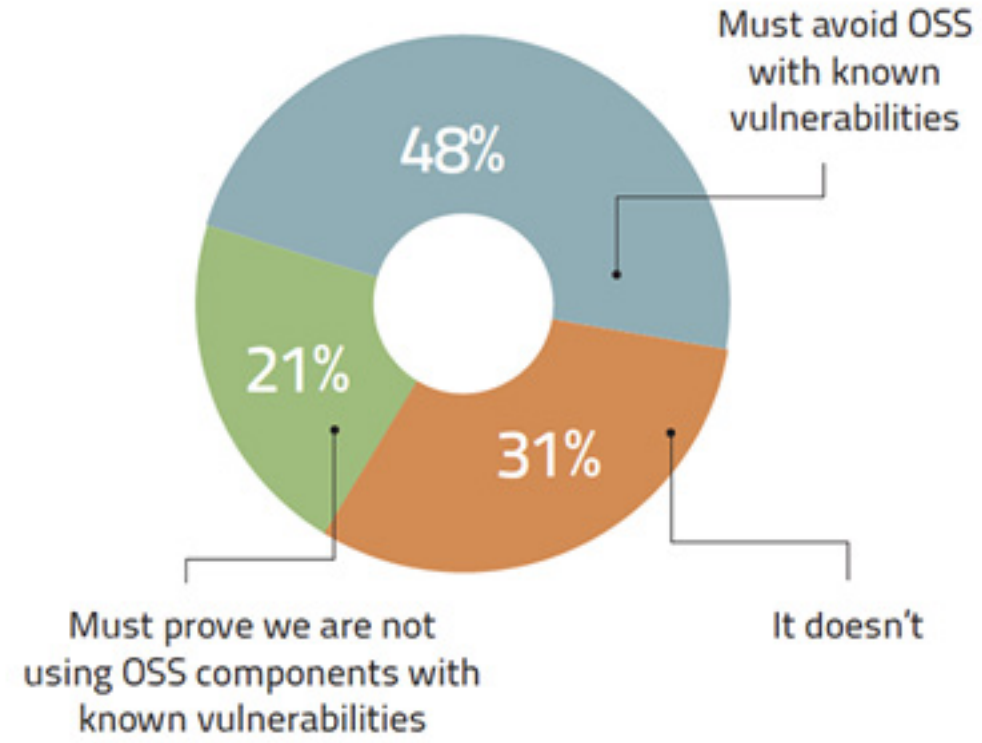
Research shows that few enterprises have or enforce policies regarding the use of third-party code. Sonatype's survey found 75% of 3,353 respondents said their organizations had a policy covering code and component use but only 68% of those surveyed -- managers, architects and developers -- followed it. In fact, 77% of respondents said their organizations had never banned an open source component, even though 31% were either victims or suspected a breach due to open source software. For more on open source security policies, see Figure 1.

SearchSecurity

FIGURE 1

Security Vulnerabilities

How does your open source security policy address security vulnerabilities?



SOURCE: 2014 SONATYPE OPEN SOURCE APPLICATION SECURITY SURVEY

Clearly, InfoSec executives responsible for developing the organization's software security posture need to revisit policies, procedures, and guidance governing code and component use to ensure that their security programs provide sufficient control over the use of open source code. Software development lifecycles should provide a framework for the pragmatic inclusion of security practices in the development process.

A Software Security Group (SSG), which falls under security and acts an intermediary between the often siloed security and development groups, should oversee application security, according to Gary McGraw, the chief technology officer of Cigital. The head of the SSG should be appointed by the board of directors to ensure that secure code is recognized as an integral part of the business; a necessary expense in the firm's governance processes; and that it's equal to other business drivers. Cigital's Building Security In Maturity Model study, based on data from 67 real software security initiatives at Bank of America, EMC, Fidelity, HSBC, Microsoft, McAfee, Salesforce and Zynga, among others, found that enterprises with mature software-development operations typically have a senior executive in charge of software security and an SSG to manage the development program. (Data from the BSIMM-V project and related documents are available under a Creative Commons Shared Attribution

3.0 License.)

The development team should be involved in formulating software security policies from the start; otherwise, the level of compliance will be low. The SSG and development team leaders need to agree on clear parameters for code and component selection, such as business case, the quality of support forums and documentation, acceptable licenses and, most importantly, the quality of code.

Giving developers responsibility in the code and component selection process puts their reputations on the line -- and could mean future liability. This level of engagement should help the development team appreciate that speed and fancy functions are not the most important coding factors: Proposed open source components and dependencies along with in-house code and software composition have to be assessed. The SSG, in conjunction with the development team, can assign an assurance level to each code section or component based on overall business risk to determine the extent of the security review required.

COST-EFFECTIVE CODE ANALYSIS

Development teams need to use both static and dynamic code analyzers. Static analysis of the code -- which takes place before the application is executed

-- provides a scalable capability for code review and can help validate that coding policies are being followed. Dynamic analysis of the code -- during runtime -- ensures it is correctly integrated and working as intended. Security managers need to make sure that adequate training is provided for developers and quality assurance testers, who are running these tools.

While analysis tools can do much of the work of finding and flagging vulnerabilities, they are not perfect, especially with messy and complex codebases. Be prepared to use manual code reviews for critical components of an application where sensitive data is processed or stored. If any code is too complex to understand, then reconsider its use or employ outside help. Outsourcing what is a highly skilled task to specialists can be cost-effective. Services that employ cloud-based scanning to test for vulnerabilities can potentially deliver a more in-depth review of vulnerabilities than organizations short on man power and resources in their security team.

Application security testing services are appearing, such as HP's Fortify Software Security Center, Checkmarx and Veracode's VAST on-demand service that analyzes code without requiring access to the source. However, relying on the assertions of third-party services or consultants means it's important to fully understand what has been tested and for which scenarios. OpenSSL,

for example, has a FIPS 140-2 certificate but FIPS validation only checks the crypto routines. The Heartbeat protocol is not part of the crypto module so it is outside the scope of FIPS. It's also important to remember that a one-off certification or review only covers the threat landscape at that point in time, so regular audits are key.

GOOGLE'S SINGLE CODE TRUNK

Once approved, code should be stored in an internal repository and developer tools configured to only retrieve code from the repository -- not the Internet. Google keeps the source code of all of its projects in a single code trunk, and all of its developers access the same repository. This process is an important aspect of version control. It reduces the risk of cross-build injection attacks, in which attackers compromise the server that hosts the components and replace them with malicious copies.

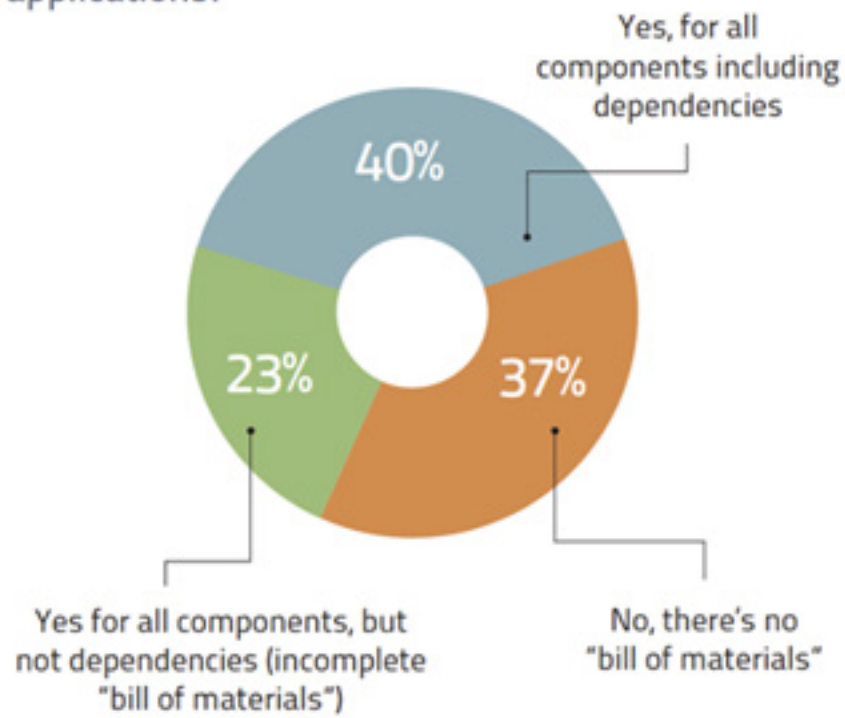
Enterprises should maintain a list of all third-party code, including all dependencies and sources, held in the repository and designate a point person who monitors all relevant security mailing lists and obtains, tests, and distributes any updates and fixes. Many companies lack basic security practices as shown in Figures 2 and 3.

SearchSecurity

FIGURE 2

Open Source Inventory

Does your organization maintain an inventory of open source components used in production applications?

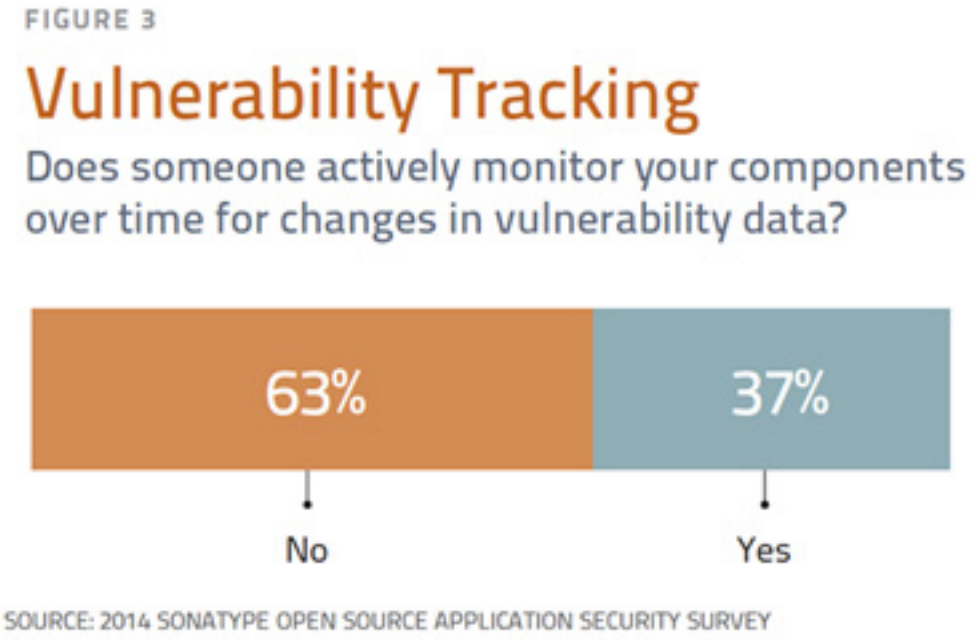


SOURCE: 2012, 2013, 2014 SONATYPE OPEN SOURCE APPLICATION SECURITY SURVEY

SearchSecurity

Open source framework Ruby on Rails was hit by several security vulnerabilities in 2013 that allowed remote code execution. Development teams unaware of these critical alerts and updates to the popular Web application framework left their clients and users at serious risk of attack.

FIGURE 3



SearchSecurity

Inevitably bugs will still make it through to production code, so all pertinent information such as source code, binaries, documentation, emergency response plans and license terms for any third-party software must be archived to allow for post-release servicing of the application. An emergency response plan should be put in place to deal with critical patches. Any application sitting on the Internet will need a rapid response to prevent an attack exploiting a newly discovered vulnerability from succeeding.

Enterprises today depend on reliable, secure software. Using open source code during application development makes sense for efficiency, cost and security reasons, but reviewing this code requires more realistic project timetables and budgets to cover tools and training. Automating policy enforcement with a well-maintained repository will enable developers to remain agile enough to keep pace while reducing the introduction and occurrence of flaws in today's complex applications. Enterprises that have outdated, unenforceable software security policies, which fail to reward developers for doing a good job of maintaining control over code, will face higher risk going forward.

MICHAEL COBB, CISSP-ISSAP, is a renowned security author with over 20 years of experience in the IT industry. He co-authored the book IIS Security and has written many technical articles for SearchSecurity.com and other leading IT publications. He was formerly a Microsoft Certified Database

SearchSecurity

Manager and a registered consultant with the CESG Listed Advisor Scheme (CLAS).

SearchSecurity

FREE RESOURCES FOR TECHNOLOGY PROFESSIONALS

TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific Web sites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more —drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily. Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

WHAT MAKES TECHTARGET UNIQUE?

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise IT professionals across all industries and markets.