



2014 Open Source Development and Application Security Survey Analysis

Version 1.0

Released: July 9, 2014

Author's Note

The content in this report was **developed independently of any sponsors or licensees**. It is based on material originally posted on [the Securosis blog](#), but has been enhanced, reviewed, and professionally edited.

Special thanks to Chris Pepper for editing and content support.

Licensed by Sonatype



Every day, developers rely on millions of third party and open source building blocks – known as components – to build the software that runs our

world. Sonatype ensures that only the best components are used throughout the software development lifecycle so that organizations don't have to make the tradeoff between going fast and being secure. Policy automation, ongoing monitoring and proactive alerts makes it easy to have full visibility and control of components throughout the software supply chain so that applications start secure and remain that way over time. Sonatype is privately held with investments from New Enterprise Associates (NEA), Accel Partners, Bay Partners, Hummer Winblad Venture Partners and Morgenthaler Ventures. Visit: www.sonatype.com

Contributors

The following individuals contributed significantly to this report through comments on the Securosis blog and follow-on review and conversations:

Marco Tietz

Derek Weeks

Copyright

This report is licensed under Creative Commons Attribution-Noncommercial-No Derivative Works 3.0.



<http://creativecommons.org/licenses/by-nc-nd/3.0/us/>

Table of Contents

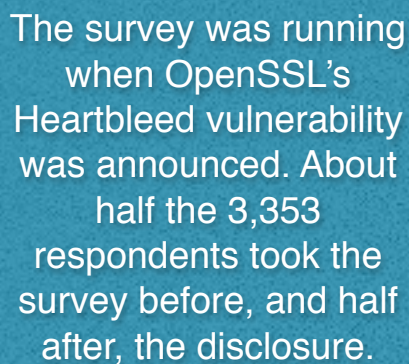
Introduction	4
Analysis of Application Security	6
Who are we hearing from?	6
Is open source important?	6
Are developers worried about security	7
What are they doing about it?	8
Development Trends	11
Are open source components more trustworthy?	11
Banning components	11
Open source policies	12
Summary	14
About Securosis	16

Introduction

Earlier this year I responded to the 2014 Open Source Development and Application Security Survey, something I have done for the last couple years. As a developer and former development manager who has benefited from open source for a couple decades I am always interested in adding my viewpoint to these inquiries, even as just one voice among thousands. But I have also directly benefited from these surveys — seeing the stuff my peers are using, and even selecting open source distributions based on these shared data points. They are yet another way to leverage the open source community.

But this year was different. Sonatype conducts this survey, and 2014 was their 4th annual review of open source development. The first thing I noticed was their name change to embrace “Application Security”. Sure enough there were several new questions on security and vulnerabilities. As security becomes more important to the craft of software development this data will be increasingly valuable to the community.

But the survey’s most interesting aspect is that it was running when [OpenSSL’s Heartbleed vulnerability](#) was announced. About half the respondents took the survey before, and half after, the disclosure. It takes a lot for a security vulnerability to make mainstream news, but Heartbleed managed it. For any of you reading this who were *not* aware of it, OpenSSL is an open source implementation of the SSL/TLS protocol. The disclosure simultaneously illustrated that open source components are in use just about everywhere — across industries and organizations of all sizes — and disrupted IT practitioners’ blind faith in this ubiquitous cryptographic module. But Heartbleed is not the story here — the fascinating question is how it affected people’s understanding of open source software and security. I wondered, “Did the vulnerability affect the survey results?”



The survey was running when OpenSSL’s Heartbleed vulnerability was announced. About half the 3,353 respondents took the survey before, and half after, the disclosure.

About this time Sonatype reached out to ask if we would like a pre-briefing on the survey results, just as they had in years past. But as we went through the data and discussed what it all meant, Sonatype suggested an independent analysis of the data, and asked if I would be interested. You don’t have to ask me twice — I jumped at the chance! As a security practitioner who has built software and managed development teams for a couple decades, I have some perspective to offer. I have seen at conferences and discussion forums how developer attitudes are changing towards

security; they don't exactly *embrace* security, but they accept it as a necessary part of the job. Additionally, we are seeing disruption yet again from development approaches, — from Agile to DevOps — affecting how we build security into software and deploy security measures into infrastructure. This research paper offers an analysis of the survey results with a focus on software security, and what it means for development teams and the open source industry in general.

Finally, for those of you in security who are not familiar with Sonatype, think Apache Maven and Nexus. Their founder built Maven, which is probably the most widely used build automation tool out there. Today the company develops the Nexus component manager, used by over 40,000 organizations for storing and organizing binary software components, including management of policies for use and automated health checks for security vulnerabilities.

As the steward of the Central Repository, which handled over 13 billion requests for open source components last year, they are in a unique position to monitor use of open source development components — including version management, license characteristics, update frequencies, and known security vulnerabilities. This perspective helped them formulate the survey and reach the 3,300+ development professionals who participated.

Taken together, that all gives Sonatype credibility in the open source community, and means they have genuine visibility into what the community is doing with open source components, libraries and frameworks.

Analysis of Application Security

Several questions in the 2014 Open Source Development and Application Security Survey focused on security practices within open source development, including vulnerability tracking and who is responsibility for security. I will dive into the results in detail, offering my perspective on where things are getting better, which results surprised me, and where I believe improvements and attention are still needed. Here we go...

Who are we hearing from?

When analyzing a survey I always start by asking: "Who is taking the survey?" That question frames many of the survey's answers. Understanding who is responding also helps illuminate the perspectives expressed and challenges discussed.

When asked **What is your role in the organization?** the respondents were largely developers, at 43% of those surveyed. Considering that most architects, DevOps types, and build managers perform some development tasks, it is a safe bet that over 50% of respondents have their hands on open source components and projects in some way. A full 79% (when we include development managers) are in a position to understand the nuances of open source development, judge security, and reflect on policy issues.

At least 75% of organizations rely on open source as the foundation of their applications.

Is open source important?

The short answer is "Hell yes, it's important!" The (Maven) Central Repository — the largest source of open source components for developers — handled thirteen billion download requests last year. For an idea of the popularity of open source components used to assemble software applications today, that is over a billion downloads each month.

Sonatype's data shows open source component usage is on the rise, growing 62% over 2012 in 2013, more than double 2011.

When asked **What percentage of a typical application in your organization is comprised of open source components?** at least 75% of organizations rely on open source as the foundation of their applications.

While '0-20%' was an option, I am willing to bet very few participants are really at 'zero' because people not using any open source would be highly unlikely to participate in this survey. I'll suggest to Sonatype that they adjust this question in future surveys to remove the ambiguity.

The survey looked at use of open source components across verticals — with more than 100 respondents working in each of the major industries including banking, insurance, technology/ISV, and government. Open source component usage is not relegated to a few target industries — it is widespread.

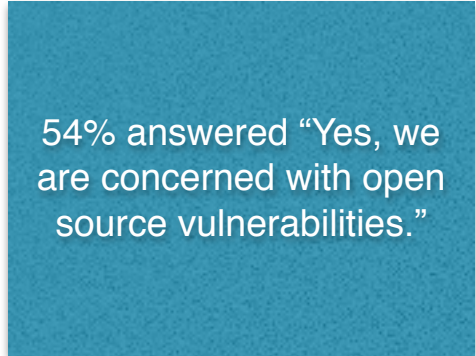
The survey also asked **How many developers are in your organization?** to which almost 500 participants answered 1,000 or more. Small firms don't employ 1,000 developers, so at least 15% of responses were from large enterprises. That is a strong showing, given that only a few years ago large enterprises did not trust open source and generally refused to officially endorse its use on corporate systems. That's not to say Apache and other open source tools were not being used — they often were — but against policy. But with nearly 700 responses from organizations with 26-100 developers, the survey reflects a good balance across organizational sizes.

Adoption continues to climb because open source has proven its worth — in terms of both quality and accelerated progress when you don't try to build everything from scratch. These statistics show that more software than ever leverages contributions from the open source community, and widespread adoption makes open source software incredibly important.

Are developers worried about security?

Questions around software security were a theme this year, which is why Sonatype changed the name to the “Open Source Development and *Application Security* Survey”.

A central question was **Are open source vulnerabilities a top concern in your position?**, to which 54% answered “Yes, we are concerned with open source vulnerabilities.” Concern among more than half of respondents is a good sign — security is seldom part of product design specifications, and has only recently become integrated into the testing phases of development. Respondents' concern with vulnerabilities is a positive sign. Considered another way, 10 years ago that number was close to zero, so we see a dramatic change in awareness.



54% answered “Yes, we are concerned with open source vulnerabilities.”

Security practitioners — basically security professionals outside of application development looking in — get annoyed that *only* about 50% responded “Yes” to this question. They zealously believe that when it comes to software development, everyone from the most senior software architect to the new guy in IT needs to make security practices a priority. As we have seen in breaches over decades, failure only requires one weak link.

And supporting the argument that software development has a long way to go when it comes to security, 47% of respondents said “Developers know it (Security) is important, but they don't have time to spend on it.” The feeling that “I am interested in security but the organization is not.” is very common among development teams. Most developers know security is an open issue. But fixing

security typically does not make its way up the list of priorities while there are important features to build — at least not until there is a problem like a data breach.

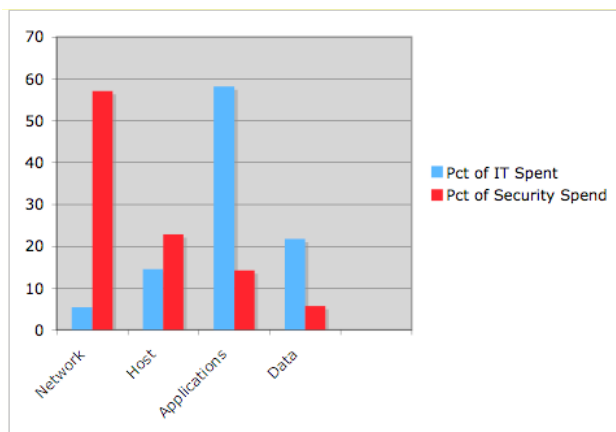
Developers' growing interest in security practices is a good sign, but allocation of resources and prioritization remains an issue.

What are developers doing about it?

This year's results offer a mixed impression of how development organizations are actually providing security.

41% of responses indicated that developers are responsible for **tracking and resolving newly discovered vulnerabilities in components included in their production applications**. As a developer this looks legitimate. And the [2014 Verizon Data Breach Investigations Report](#) makes clear that the application stack is where the main security issues are being exploited. But security buying behavior does not match the survey results. Understanding that survey participants were mostly developers with an open source perspective, this number is still surprisingly high because the vast majority of security expenditures are for *network and endpoint* security devices. Security, including application security, is generally bolted on after the coding phase, rather than built into the development lifecycle.

It is not uncommon for IT or security teams, rather than developers, to decide *how* to fix a specific application vulnerability. It might be a code fix, but more often it's a non-code workaround, firewall rule, etc. While nearly 41% consider application development responsible for tracking and resolving newly discovered vulnerabilities, about 18% of respondents said this responsibility falls to IT Operations, and another 18% said it is the security team's task. From buying behavior we know that means network devices. Most organizations opt for the quicker firewall approach when they can, rather than code fixes which they view as potentially destabilizing. But it is not always possible to address open source vulnerabilities in this manner; patches are required. It's well documented that organizations are slow to patch vulnerabilities for fear of destabilizing the application stack.



Gunnar Peterson, 2011 Security vs IT Spending

[Jeremiah Grossman](#), [Gunnar Peterson](#), and others have all discussed the ineffectiveness of gearing security toward the network rather than applications. There is a genuine mismatch between where IT spends its money — on applications — and where it deploys security controls. The [Whitehat Website Security Statistics](#) show a long-term cost benefit from fixing problems within applications, but we see that network security controls remain the standard. The point is to highlight the companies don't focus their security

investments on problem areas, and have adopted a short sighted ‘band-aid’ approach application security. I would hope to see the percentage of application developers responsible for resolving security issues increase in future surveys.

Developers I speak with say they would like to do more for application security but cannot. They

53% of organizations do not have a policy governing open source vulnerabilities.

break their limitations down into three problem areas: how much latitude they have to fix security flaws, their ability to patch given the complexity of their operating environment, and limited resources to train people on security.

These points are supported by other survey results. For example 29% of respondents monitor open source components for changes in vulnerabilities — almost one in three are tracking emerging issues.

At the same time only 16% said they *must prove* there are no *known* vulnerable components in their products. That is a tiny percentage. A full 53% do not have a policy governing open source vulnerabilities (**How does your open source policy address security vulnerabilities?**) at all. As developers we know that without a policy it is not assigned to development, and will not get fixed. Most industries demand validation for critical software. You hope that the software that runs your local power station, or the software built for your bank, is part of the 16% minority. Regardless, open source development needs to play ‘catch-up’ with vulnerability management.

Policies and practices are often accompanied by training. Only over the last few years have security testing and practices have worked their way into the software development lifecycle — whether Waterfall, Agile, or something else. For organizations which practice secure code development, only a handful of people are trained because *security training is expensive* — often thousands of dollars per person per class. Security training is generally not budgeted in development shops — and the limited budget generally goes for cheaper options: less specialized on-line tutorials and self-paced study courses. Responses to **What application security training is available to you?** were overwhelmingly e-learning (60%), while 26% responded that no training was available. Of course it isn’t necessary to fully train every developer in security — not every person in development needs to be a security expert — but each team member should understand security as it pertains to their role.


We can see here that changes are coming faster than companies can react — they have yet to understand or address the disconnect between their security problems and resource expenditures.

What Heartbleed Tells Us

[Heartbleed](#) was an extremely serious OpenSSL vulnerability that allowed attackers to remotely view portions of server memory, leaking sensitive information. It is incredibly rare for an open source component vulnerability to make national headlines, but given the severity of the issues — and the fact that thousands of very large companies were running known vulnerable versions of OpenSSL —

I am very glad Heartbleed received sufficient attention to prompt immediate action. Without accelerated action prompted by viral word-of-mouth, catastrophic issues could have resulted.

Coincidentally, when this vulnerability was discovered, Sonatype was a week into the survey. Roughly 45% of respondents took the survey before the announcement, and 55% after. Sonatype showed me the survey results in three different sets: responses before Heartbleed drew international media attention, afterwards, and combined results. I was amazed at the consistency of the before and after data. Most before-and-after answers were within 3% standard deviation, with one exception: **Has your company had a breach that can be attributed to a vulnerability in an open source component or dependency in the last 12 months?** 31% responded “Yes” after the news, compared to 19% before.



One in five firms
experienced some type of
breach in the prior 12
months — *before*
Heartbleed.

To be honest, and this is probably a minority opinion, the fact that almost 1 in 3 reported a breach in the last 12 months due to open source is the story here, not Heartbleed. Don't get me wrong — Heartbleed was a big deal. But the fact that about 1 in 5 developers were *already* aware of a breach is important. Many firms don't disclose breaches at all, so we are stuck speculating about what's really going inside companies without surveys like this. It also underscores the need, discussed above, to track open source vulnerabilities and have a policy or process in place to remediate important security issues in a timely fashion. Those of you who work in Agile or DevOps environments know that without a task card to track and assign issues, work does not get done. That includes fixing vulnerable components.

So we know open source is incredibly important, but open source development is still struggling to address security. Overall the trends are positive — awareness, concern, and commitment to security are generally improving. But the lack of tracking and security education, and insufficient number of people following policies, show development teams are not yet adequately prepared to deal with security as part of the development process.

Development Trends

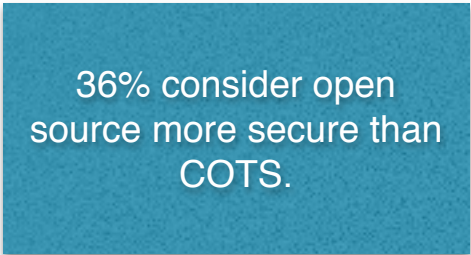
Here we examine how security and is affecting development, altering what teams track and how they address vulnerabilities.

Are open source components more trustworthy?

The survey asked, unambiguously, **Do you believe software assembled with open source is as secure as commercial off-the-shelf (COTS)?** Under 9% of respondents consider software assembled with open source less secure, while over 36% consider open source more secure than COTS. The remainder, around 55%, said that it was about even.

Even more interesting: 34.83% of survey participants who responded before Heartbleed believed applications assembled using open source components were more secure than COTS. After Heartbleed the number rose slightly, to 36.06%.

Yes, after a widely-reported major vulnerability in an open source component used in millions of systems around the globe, confidence in open source security did not suffer. In fact, it ticked up a point. Ironic? Amazing? I was surprised and impressed.



36% consider open source more secure than COTS.

What people believe is not necessarily fact. And we can't really perform a quantitative head-to-head comparison between applications assembled with open source components and COTS security to verify the belief behind these answers. But these survey respondents deal with open source and commercial software on a daily basis — they are offering informed professional opinions. The net is that for every person who felt COTS was more secure, four felt open source was more trustworthy. In any sort of popular vote that qualifies as a landslide.

Banning components

Has your company ever banned the use of an open source component, library or project?

The majority of respondents, some 78%, said "No". Still, I single this question out as a development practice issue, which I hear organizations talking about more and more.

Software organizations ban components for a number of reasons. Licensing terms might be egregious. Or they might simply no longer trust a component's reliability or security. For example, virtually all released Struts components have severe security exploits, described in critical CVE

warnings. Poorly written code has reliability and security issues. The two tend to go hand in hand. You can verify this by looking at bug tracking reports: issues clump together around one or two problematic pieces of software. Banning a module is often politically messy because it can be difficult to find or build a suitable replacement. But it is an effective, focused way to improve security and reliability.

The net is that for every person who felt COTS was more secure, 4 felt open source was more trustworthy.

Post-Snowden we have seen increased discussion around trust and whether or not to use certain libraries because of potential subversion by the NSA. This is more of a risk perception issue than more tangible issues such as licensing, but nonetheless an important topic of discussion. Regardless of your motivation, banning modules is an option to consider for suspect elements of your stack.

Open source policies

Policies were a major focus area for the survey, and

Does your company have an open source policy?

was the lead-in for several policy-related questions. The good news is 47% of respondents have an open source policy. The bad news is 43% do not have a policy, and only 68% follow the policy that is in place. The next survey question sheds some light on adoption rates.

When asked, **What are the top three challenges with your open source policy?**, the top three responses were: 39% believed a top challenge is that it does not deal with security vulnerabilities, 41% stated policy has little enforcement so workarounds are common, and 35% said expectations are not clear.

This begs the question: What is in an open source policy? The answer dovetails nicely with an early survey question: **When selecting components, what characteristics would be most helpful to you?** How you answer that question is how you determine what should be in your policy, and functionality, licensing, compatibility and security topped the survey results by a wide margin. Most open source policies include a licensing component, specifying which licenses (or types) are permitted. And most specify versioning and quality controls, such as no beta software. More often than not policies address security — perhaps requiring components with critical vulnerabilities to be patched or avoided altogether; while that sounds obvious, remember that 29% do not have a security policy for open source, and another 24% of respondents don't

The two key takeaways are this: 43% of firms don't have a policy, and only 68% follow the policy that is in place.

have a policy at all. Organizations are becoming more aware of risks with open source, with a growing trend to tracking licensing and security issues.

Who in your organization is primarily responsible for open source policy / governance?

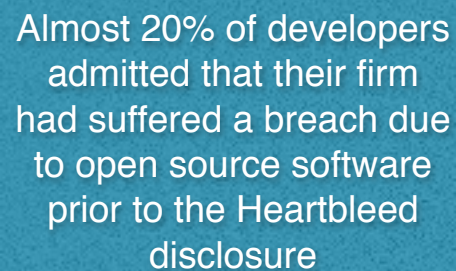
While the bulk of responsibility fell on development managers (34%) and IT architects (24%), much of it landed outside development. Legal, risk, and executive teams are unlikely to craft policies which development can implement easily. So development needs to either take ownership of policies or work with outside groups to define feasible goals and the easiest route to them.

We could spend many pages on policies, but the underlying issue is simple: Policies are supposed to make your life easier. If they don't you need to work on them. Yes, I know those of you who deal with regulatory compliance in your daily jobs scoff at this, but it's true. Policies are *supposed to* help avoid large problems or failures down the road, which cost serious time and resources to fix. The rule is simple: policies written without regard for how they will be implemented, or a clear path to make open source use easier and better, are likely to be bypassed. If your policies being ignored, we suggest you find out why, and see if some simple alterations can help make adherence easier and raise adoption rates.

Summary

To wrap up our analysis of the 2014 survey, let's recap key points:

- Open source software is almost universally embraced, even within enterprises which banned it a few years ago.
- Almost 20% of developers admitted their firms had suffered some breach due to open source software *prior* to the Heartbleed disclosure. That number climbed over 30% after.
- Open source software licensing remains a top concern and directly influences which open source distributions companies use.
- At a ratio of 4:1, developers and IT practitioners consider open source software more secure than commercial off-the-shelf solutions.



Almost 20% of developers admitted that their firm had suffered a breach due to open source software prior to the Heartbleed disclosure

The full survey results can be found online at: <http://www.sonatype.com/about/2014-open-source-software-development-survey>

If you have any questions on this topic, or want to discuss your situation specifically, feel free to send us a note at info@securosis.com or ask via the Securosis Nexus <<http://nexus.securosis.com/>>.

About the Analyst

Adrian Lane, Analyst and CTO

Adrian Lane is a Senior Security Strategist with 25 years of industry experience. He brings over a decade of C-level executive expertise to the Securosis team. Mr. Lane specializes in database architecture, data security and secure code development. With extensive experience as a member of the vendor community (including positions at Ingres and Oracle), in addition to time as an IT customer in the CIO role, Adrian brings a business-oriented perspective to security implementations. Prior to joining Securosis, Adrian was CTO at database security firm IPLocks, Vice President of Engineering at Touchpoint, and CTO of the secure payment and digital rights management firm Transactor/Brodia. Adrian also blogs for Dark Reading and is a regular contributor to Information Security Magazine. Mr. Lane is a Computer Science graduate of the University of California at Berkeley with post-graduate work in operating systems at Stanford University.

About Securosis

Securosis, LLC is an independent research and analysis firm dedicated to thought leadership, objectivity, and transparency. Our analysts have all held executive level positions and are dedicated to providing high-value, pragmatic advisory services. Our services include:

- **The Securosis Nexus:** The Securosis Nexus is an online environment to help you get your job done better and faster. It provides pragmatic research on security topics that tells you exactly what you need to know, backed with industry-leading expert advice to answer your questions. The Nexus was designed to be fast and easy to use, and to get you the information you need as quickly as possible. Access it at <https://nexus.securosis.com/>.
- **Primary research publishing:** We currently release the vast majority of our research for free through our blog, and archive it in our Research Library. Most of these research documents can be sponsored for distribution on an annual basis. All published materials and presentations meet our strict objectivity requirements and conform to our Totally Transparent Research policy.
- **Research products and strategic advisory services for end users:** Securosis will be introducing a line of research products and inquiry-based subscription services designed to assist end user organizations in accelerating project and program success. Additional advisory projects are also available, including product selection assistance, technology and architecture strategy, education, security management evaluations, and risk assessment.
- **Retainer services for vendors:** Although we will accept briefings from anyone, some vendors opt for a tighter, ongoing relationship. We offer a number of flexible retainer packages. Services available as part of a retainer package include market and product analysis and strategy, technology guidance, product evaluation, and merger and acquisition assessment. Even with paid clients, we maintain our strict objectivity and confidentiality requirements. More information on our retainer services (PDF) is available.
- **External speaking and editorial:** Securosis analysts frequently speak at industry events, give online presentations, and write and/or speak for a variety of publications and media.
- **Other expert services:** Securosis analysts are available for other services as well, including Strategic Advisory Days, Strategy Consulting engagements, and Investor Services. These tend to be customized to meet a client's particular requirements.

Our clients range from stealth startups to some of the best known technology vendors and end users. Clients include large financial institutions, institutional investors, mid-sized enterprises, and major security vendors.

Additionally, Securosis partners with security testing labs to provide unique product evaluations that combine in-depth technical analysis with high-level product, architecture, and market analysis. For more information about Securosis, visit our website: <http://securosis.com/>.