



Open Source Security Study

How Are Open Source Development Communities
Embracing Security Best Practices?

Open Source Security Study

How Are Open Source Development Communities Embracing Security Best Practices?

Table of Contents

3	Introduction
4	Study Methodology
5	Key Findings
9	Conclusions
10	About Larry Suto
11	About Fortify
11	References

Executive Summary

Open source now permeates more than 50 percent of enterprises, and its use is growing rapidly.¹ This trend underlies an assumption held by many IT and business leaders that open source is enterprise class in terms of functionality and scalability. But is it secure? How much business risk is introduced with open source?

As a provider of software security assurance tools, Fortify has often been drawn into the center of the debate over this question. The use of Fortify tools to identify vulnerabilities in open source software has demonstrated that risk exists. Fortify has made attempts to reduce the risk by sharing vulnerability reports with the open source community. Yet the risk remains. In an effort to ascertain why open source development seems resistant to information on security, Fortify surveyed the open source community. (See Figure 1.) Our research revealed that open source projects lack the three essential elements of security: people, process, and technology, thereby introducing significant application security risk. The study showed that many open source projects fail to:

- 1. Provide Access to Security Expertise:** Few open source projects provide documentation that covers the security implications and secure deployment of the software they develop, a dedicated email alias for users to report security vulnerabilities, or easy access to internal security experts to discuss security issues.



The European Commission's Competition Commissioner, Neelie Kroes, recently stated that open standards, and open source, are preferable to traditional closed source software.⁶

2. Adopt a Secure Development Process: Not only did every project that we scanned contain significant security issues, but in all but one, the total number of security issues remained constant or increased between successive releases. This demonstrates that the projects have not adopted a successful secure development process.

3. Leverage Technology to Uncover Security Vulnerabilities: Well-known security vulnerabilities, such as Cross-Site Scripting (XSS) and SQL Injection, were among the most common and serious problems identified, which is consistent with OWASP findings.² These classes of vulnerabilities can be identified by enrolling in the free Fortify Java Open Review (JOR) project or with open source tools, such as FindBugs.³ This indicates that the projects do not make use of technology to identify and resolve security issues.⁴

These findings provide a call-to-action for organizations that rely on open source software. Specifically, Fortify recommends:

- Government and commercial organizations that leverage open source should use open source applications with great caution. Risk analysis and code review should be performed on any open source code running in business-critical applications, and these processes should be repeated before new versions of open source components are approved for use.
- Open source projects should adopt robust security practices from their commercial counterparts. Open source development can benefit from private industry practices — notably those created by financial services organizations and larger independent software vendors (ISVs). Open source communities can then advertise and substantiate effective security practices that blend process and technology.

Introduction

Fortify recently conducted a study designed to better understand the overall security of popular open source projects and the role of security in their development processes. This work was motivated by the:

Rapid growth in the adoption of open source among enterprises

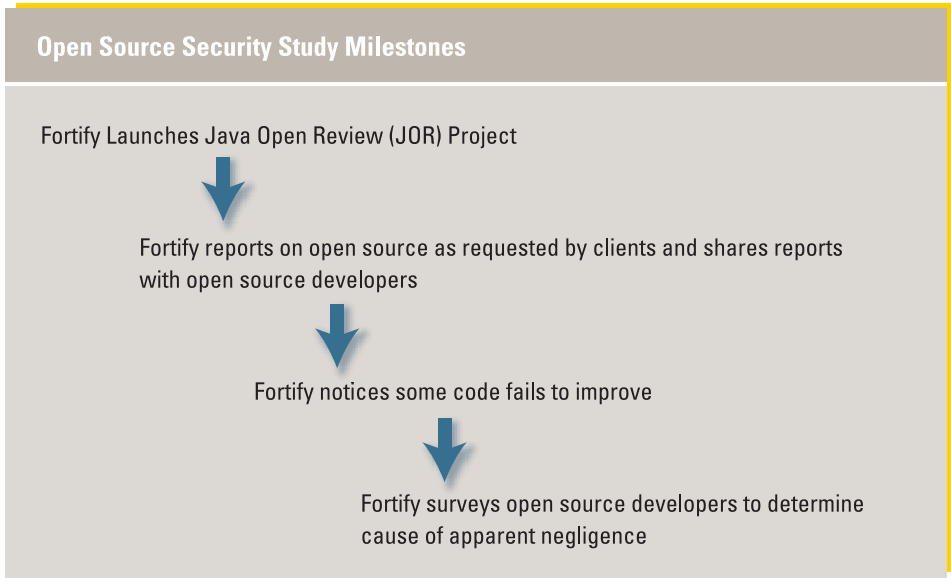
- An April 2008 survey by CIO.com showed that more than half of the respondents (53 percent) are using open source applications in their organization today, and an additional 10 percent plan to do so in the next year. For nearly half (44 percent), open source applications are considered equal to closed-source solutions during the acquisition process.⁵
- The European Commission's Competition Commissioner, Neelie Kroes, recently stated that open standards, and open source, are preferable to traditional closed source software.⁶

It is no longer acceptable for businesses to ignore the risk that open source software can introduce, nor is it adequate for open source projects to disregard the role security plays in their development process.

Rising number of attacks targeted at the application layer

- Today’s hackers are going after applications. Gartner has reported that 75 percent of hacks take place at the application layer.⁷
- The US Air Force, for example, has seen application layer attacks increase from 2 percent in 2004 to 33 percent in 2006.⁸

Fortify software has worked with organizations around the world that are committed to business software assurance as demonstrated through improved software development processes. A few years ago, this was limited to the elite, which typically included members of the global financial services and military communities. Today, we are witnessing a surge in adoption of secure development practices across organizations, both large and small, in a wide range of industries. This widespread adoption of security processes has helped to raise the bar for security. *It is no longer acceptable for businesses to ignore the risk that open source software can introduce, nor is it adequate for open source projects to disregard the role security plays in their development process.*



Study Methodology

The projects included in this study were selected because they are implemented in Java (the most common programming language for enterprise development⁹), represent a wide range of application functionality, and are used extensively to build and deploy enterprise applications. Freeware projects that are not open source were excluded. Table I lists the projects included in the study and their respective roles.



“It’s easier to break things than make them unbreakable.”

— John Pescatore,
Gartner analyst

Table I: Included Projects

Application	Description
Derby	Relational database
Geronimo	Application server
Hibernate	Object relational mapping tool
Hipergate	CRM web application
JBoss	Application server
JOAnAS	Application server
OFBiz	E-Business solution web application
OpenCMS	Content management solution
Resin	Application server
Struts	Web application framework
Tomcat	Application server

Two to four versions of each project were downloaded and analyzed with the standard capabilities of Fortify SCA, a static analysis tool targeted at security. Security-sensitive areas of the code and operations that are often associated with security vulnerabilities, such as database queries and dynamic user interface elements, were reviewed manually to ensure consistency with the Fortify findings. Major security issues discovered by Fortify SCA were also manually verified for accuracy. Fortify SCA groups its findings into high-, medium, and low-impact groups. Due to the extensive number of vulnerabilities identified in the medium and low categories, this paper includes only the high-impact findings. In accordance with responsible disclosure guidelines, this report does not include any detailed vulnerability information.

As Gartner analyst John Pescatore observed, “It’s easier to break things than make them unbreakable.”¹⁰ Finding and fixing vulnerabilities in existing code, while necessary, needs to be complemented with an effective security process to truly have impact. In order to evaluate the security expertise open source development organizations provide their users and measure the secure development processes that are in place for securing their software, Fortify interacted with open source maintainers and examined documented open source security practices.

Key Findings

When the Fortify Java Open Review (JOR) project was initially launched, it uncovered numerous flaws in a variety of open source packages. Our subsequent review of several additional code bases indicates that serious security threats stemming from numerous application vulnerabilities are a direct result of poor or nonexistent security processes.

This follow-up survey found that security best practices are a low priority to the open source projects surveyed/community. Yet open source packages often claim enterprise-class capabilities but are not adopting — or even considering — industry best security practices.

We cannot overemphasize the importance of a dedicated security expert. Security, unlike quality, requires a unique perspective.

Only a few open source development teams are moving in the right direction. For example, in July 2008, Mozilla announced a security initiative to improve the browser’s security, hiring independent security consultant Rich Mogul as an advisor.¹¹ This action is a role model for all open source communities.

Thus, though some open source packages can rightfully boast strong security teams and solid development practices, many still lack the people, process, and technology to get security right. The remainder of this section outlines evidence that supports this conclusion.

Failure to Provide Access to Security Expertise

We sought out three security resources that users of open source might rely on: documentation that covers the security implications and secure deployment of the software they develop, a dedicated email alias for users to report security vulnerabilities, or easy access to internal security experts to discuss security issues. We found that few open source projects provide any of these resources, as summarized in Table II.

Table II: Access to Security Expertise

Package	Security-Specific Email Alias	Prominent Link to Security Info ¹²	Easy Access to Security Experts
Derby	No	No	No
Geronimo	No	No	No
Hipergate	No	No	No
Hibernate	No	No	No
JBoss	No	Yes	Yes
JOnAS	No	No	No
Ofbiz	No	No	No
OpenCMS	No	No	No
Resin	No	No	Yes
Tomcat	Yes	Yes	Yes
Struts	No	No	No

In 2005, Gartner invoked security process as a core for application security: “Managers of application development and testing organizations must be willing to identify security specialists on their staffs and invest in training, education and processes for their respective teams.”¹³ The same prescription should apply for open source. These resources are indicative of the level of security expertise developed within an organization, and their absence leaves the project’s users with little insight into the security practices of the open source project and little recourse when they experience a problem related to security.

We cannot overemphasize the importance of a dedicated security expert. Security, unlike quality, requires a unique perspective. As security expert Bruce Schneier noted in his blog, “Security requires a particular mindset... [Security professionals] can’t vote without trying to figure out how to vote twice.”¹⁴

Failure to Adopt a Secure Development Process

In virtually every project analyzed, there were a significant number of security issues that went unaddressed over three generations of releases, sometimes spanning more than a year. Additionally, security vulnerabilities increased significantly or remained relatively constant over successive releases, as summarized in Table III.

Table III: Remediation Trends Across Releases

Package	Total Issues	Lines of Code
Cayenne 1.14	0	35373
Cayenne 1.24	3	43741
Cayenne 2.04	3	43751
Derby 10.2.2.0	398	224712
Derby 10.3.1.4	312	251659
Derby 10.3.2.1	313	253347
Geronimo 2.0	41	87375
Geronimo 2.01	103	85001
Geronimo 2.02	106	85483
Hibernate 3.12	18	62143
Hibernate 3.13	18	62865
Hibernate 3.25	23	74834
Hipergate 2.1.20	10734	108276
Hipergate 3.0.26	14425	80941
Hipergate 3.0.30	14423	83875
JOnAS 4.84	193	132013
JOnAS 4.85	198	132396
JOnAS 4.86	196	133145

The continued presence of a large number of security issues not only represents an immediate security risk, it also demonstrates that the projects have not adopted a successful secure development process. If a secure development process existed, we would expect to see the number of security issues decrease noticeably over time.

Failure to Leverage Technology to Uncover Security Vulnerabilities

The number of security issues identified in the study — especially in the most popular open source packages — was surprising. In particular, the two most prevalent classes of vulnerabilities, Cross-Site Scripting and SQL Injection, represent a daunting challenge for open source developers. Table IV summarizes the number of issues found in these categories.



Fortify, through the Java Open Review (JOR) project, has worked with over one hundred open source development teams to identify common classes of security vulnerabilities, including Cross-Site Scripting and SQL Injection.

Table IV: Cross-Site Scripting and SQL Injection Issues

Vulnerability	Total Found
Cross-Site Scripting	22,828
SQL Injection	15,612

Fortify, through the Java Open Review (JOR) project, has worked with over one hundred open source development teams to identify common classes of security vulnerabilities, including Cross-Site Scripting and SQL Injection. However, many open source development teams have not leveraged JOR, causing them to lose a key opportunity to quickly identify and remediate security issues. Beyond JOR, a variety of open source tools, such as FindBugs, can be leveraged as part of a secure development lifecycle.

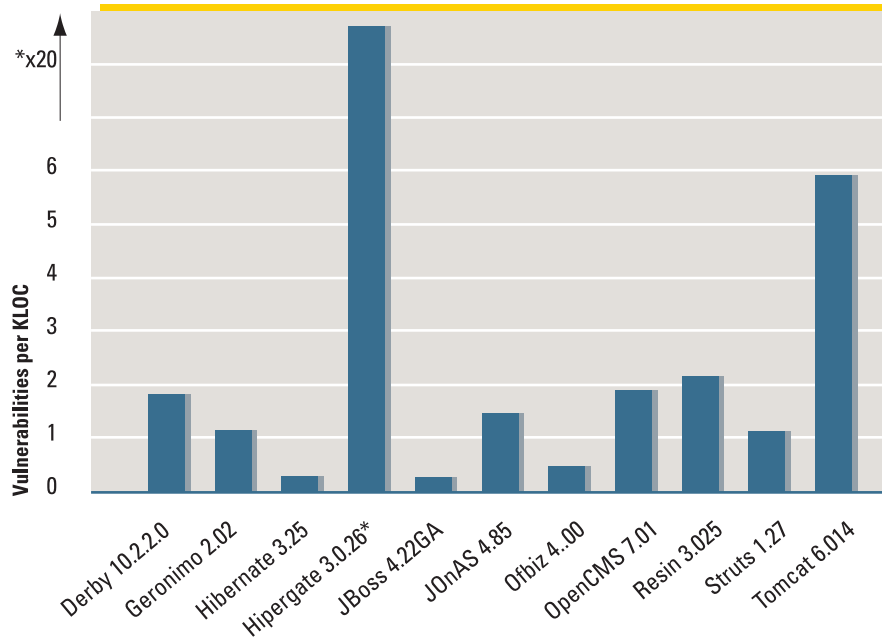
The state of affairs in which we are left is disappointing. Table V summarizes the collected results for the high-impact security issues discovered by Fortify SCA for the most current version of the projects.

Table V: Summary Statistics of Packages Scanned

Package	Total Issues	Total KLOC	Issues per KLOC
Derby 10.2.2.0	398	224.7	1.8
Geronimo 2.02	106	85.48	1.2
Hibernate 3.25	23	74.834	0.31
Hipergate 3.0.26	14425	80.94	178.2
JBoss 4.22GA	72	264.03	0.27
JOnAS 4.85	198	132.4	1.5
Ofbiz 4.00	56	110.63	0.51
OpenCMS 7.01	241	130.25	1.9
Resin 3.025	204	92.56	2.2
Struts 1.27	41	34.3	1.2
Tomcat 6.014	469	80.2	5.8
Average			17.72
Issues	Total Lines of Code (LOC)		
44233	4,247,318		

Risk analysis and code review should be performed on any open source code running in business-critical applications, and these processes should be repeated before new versions of open source components are approved for use.

Vulnerability Density



* Hipergate's vulnerability numbers were divided by 20 to avoid presentation issues due to scale.

Conclusions

These findings provide a call-to-action for organizations that rely on open source software. Specifically, Fortify recommends:

- **Government and commercial organizations that leverage open source should use open source applications with great caution.** Risk analysis and code review should be performed on any open source code running in business-critical applications, and these processes should be repeated before new versions of open source components are approved for use. Organizations considering open source software must thoroughly evaluate open source security practices. We recommend using the standards we recommend below to open source communities as a checklist. In addition, enterprises should:
 - Raise security awareness within open source development communities and emphasize the importance of preventing vulnerabilities upstream. Enterprise security teams should articulate their security requirements to open source maintainers to accelerate the adoption of secure development lifecycles.
 - Perform assessments to understand where your open source deployments and components stand from a security perspective.

– Remediate vulnerabilities internally or leverage Fortify’s JOR, which provides audited versions of several open source packages.

- **Open source projects should adopt robust security practices from their commercial counterparts.** Open source development can benefit from private industry practices — notably those created by financial services organizations and larger ISVs. Open source communities can then advertise and substantiate effective security practices that blend process and technology. Best practices to consider include:

People — Appointment of a security expert with the power to veto releases from getting into production — known as a gate model. Develop the expertise to conduct security activities and get security right.

Process — Build security in by mandating processes that integrate security proactively throughout the software development lifecycle. Include relevant non-coding activities, such as threat modeling and the development of abuse cases.

Technologies — Leverage technologies to get security right, which include static analysis in development and dynamic analysis during security testing in quality assurance.

Specifically, open source projects can leverage advances already made in the commercial software community. With regards to developing an encompassing security development lifecycle, open source projects may want to reference Dr. Gary McGraw’s book, *Software Security: Building Security In*. The book covers everything from threat modeling and abuse cases to code reviews and security testing. Microsoft has also made their threat modeling methodology and supporting tools available online.¹⁵ Going further, developers can utilize the OWASP Code Review guide to help guide their own internal security reviews.¹⁶ Fortify has made their static analysis suite available to open source projects through the Fortify Open Review project to further supplement open source security efforts.¹⁷

With these actions in place, open source communities can then advertise and substantiate effective security practices that will encourage strong adoption. As highlighted earlier in this report, the steps taken by Mozilla are a prototype for open source security.

About Larry Suto

Larry Suto is an independent consultant who has consulted for Fortune 500 companies, such as Charles Schwab, Cisco, Kaiser Permanente, Pepsico and Wells Fargo. He specializes in enterprise security architecture, risk management, and software code security. Larry can be reached at larry.suto@gmail.com.

About Fortify

Fortify Software's Business Software Assurance solutions protect companies and organizations from today's greatest security risk: the software that runs their businesses. Fortify reduces the threat of catastrophic financial loss and damage to reputation as well as ensuring timely compliance with government and industry mandates. Fortify's customers include government agencies and Global 2000 leaders in financial services, healthcare, e-commerce, telecommunications, publishing, insurance, systems integration and information technology. For more information, please visit us at www.fortify.com.

References

- 1 <http://www.networkworld.com/news/2008/053008-survey-open-source-is-entering.html>
- 2 http://www.owasp.org/index.php/OWASP_Top_Ten_Project
- 3 <http://findbugs.sourceforge.net/>
- 4 The authors would like to thank Dmitri Polejaev for his help in looking at countless lines of code.
- 5 <http://www.networkworld.com/news/2008/053008-survey-open-source-is-entering.html>
- 6 <http://www.theinquirer.net/gb/inquirer/news/2008/06/10/ec-backs-open-source>
- 7 *Now Is the Time for Security at the Application Level*, Theresa Lanowitz, Gartner December 2005.
- 8 *Colonel Kevin Foley, Best Practices in Cyber Defense from the US Air Force*, Federal Computer Week webinar, 12 February 2008.
- 9 *TIOBE Programming Community Index for July 2008*, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- 10 From Mr. Pescatore's presentation, *Gartner's Threat Forecast Timeline* delivered at Gartner IT Security Summit, June 2008.
- 11 http://www.theregister.co.uk/2008/07/08/mozilla_security_metrics/
- 12 Example: <http://www.mozilla.org/security/>
- 13 *Now Is the Time for Security at the Application Level*, Gartner Research, December 2005.
- 14 http://www.schneier.com/blog/archives/2008/03/the_security_mi.html
- 15 <http://msdn.microsoft.com/en-us/security/aa570413.aspx>
- 16 http://www.owasp.org/index.php/OWASP_Code_Review_Guide_Table_of_Contents
- 17 <http://opensource.fortify.com>



FORTIFY SOFTWARE INC.

MORE INFORMATION IS AVAILABLE AT WWW.FORTIFY.COM

2215 BRIDGEPOINTE PKWY.
SUITE 400
SAN MATEO, CALIFORNIA 94404

TEL: (650) 358-5600
FAX: (650) 358-4600
EMAIL: CONTACT@FORTIFY.COM