



BUGS

[TechNewsWorld](#) > [Security](#) > [Bugs](#) | [Read Next Article in Bugs](#)

July 15, 2010 01:02:21 PM

Please note that this material is copyright protected. It is illegal to display or reproduce this article without permission for any commercial purpose, including use as marketing or public relations literature. To obtain reprints of this article for authorized use, please call a sales representative at (818) 461-9700 or visit <http://www.ectnews.com/about/reprints/>.

Taking FOSS Security Seriously



By Jack M. Germain
[LinuxInsider](#)
Part of the ECT News Network
08/07/09 4:00 AM PT


[Back to Online Version](#)
[E-Mail Article](#)
[Reprints](#)

Developers of open source software projects should be just as concerned about security as anyone developing a proprietary app. However, the nature of the two development processes can be very different at times, and debate still rages about which is inherently more secure -- a secret code kept by a company, or a public one that all eyes can see. Just as important is how each community reacts once a problem is spotted.

▼ advertisement

Bandwidth Monitoring w/ NetFlow Analyzer

Monitor Bandwidth, Identify top talkers, do better capacity planning. Try Now!

Code hunters are spotting with greater frequency defective coding that could open security  holes in free and [open source](#) (FOSS) software.

The [Open Source Report 2008](#) and the Architecture Library Report, conducted by [Coverity](#) for the U.S. Department Homeland Security Cybersecurity Open Source Hardening Project, shows more than 10,000 defects fixed since project launch in March 2006.

The report, delivered in July at the OSCON 2009 (Open source Convention) gathering, used the same analysis tools and configurations as the Scan Benchmark 2006. The results are based on analysis of over 55 million lines of code from more than 250 open source projects that represent 14,238 individual project analysis runs. All totaled, nearly 10 billion lines of code were analyzed.

By understanding possible code execution paths, defects are identified and eliminated by open source developers, according to the report's author, David Maxwell, Open Source Strategist for Coverity. The code analysis used Coverity Prevent, a static code analysis tool that delivers path simulation, data flow analysis and false path pruning.

"It is the responsibility of everyone who works in software system to investigate testing and security issues. People with an engineer's mindset want to break down security. Security is a physical problem by nature. You have to analyze the whole thing," Maxwell told LinuxInsider.

Report Findings

Overall, code testers found that defect density dropped 16 percent over the past two years. Defect density is the number of defects per 1,000 lines of code. For example, a defect density of 1.0 means one defect in 1,000 lines of code. A defect density of 0.5 means one defect in 2,000 lines of code.

As many as 314 defects were found in one particular code base. How often the same code defect occurs may be directly related to the frequency of the type of operations the code runs. For example, a NULL Pointer Dereference was tracked in 6,448 incidents for 27.95 percent frequency. A resource leak occurred in 5,852 defects for a frequency of 25.73 percent.

False positives involved a reasonably small percentage of the results. Currently, false positives are below 14 percent.

Security View Varied

Not everyone involved in building software weighs security factors with the same intensity. In fact, there is quite a variety of how seriously security is received, according to Maxwell.

For instance, some project leaders restrict access to security staff only. Others have a wide-open review process. Some projects use huge tests of the software before releasing, he explained.

"When dealing with open source projects, some security issues are handled free-form. Others are based on maintaining a built-up community reputation," said Maxwell.

Fluid Standards

In order to spot defective code that can lead to security issues, those checking the code have to be intent on finding a problem. Many similarities in security exist with both open source and proprietary software products.

Engineers who follow one set of standards during their day jobs for proprietary firms might follow those same principles at night while developing their own software. The major difference stems from the case manager who has to follow a set company line, said Maxwell.

"The 'more eyes' theory is often valid. More people can participate, but not all do it. There needs to be enough people with a level of interest to look for security flaws," said Maxwell.

Testing Key?

The issue of software security is present on both sides of the software industry -- proprietary and open source. However, the amount of testing done and who does it tends to be more manifest in the open source community.

"[Testing's] obviously critical, and it's growing in importance. What's changed is that testing used to be almost exclusively the domain of testers who, by definition, aren't that close to the code. Now, developers are seen to have equal responsibility for security and are expected to pursue rigorous verification of their code before it's ever given to a test team. That's a big paradigm shift for many development teams, but definitely a healthy change where security becomes an organizational responsibility and not just the purview of the test team," Gwyn Fisher, CTO of [Klocwork](#), told LinuxInsider. Klocwork develops static code analysis technology used by software developers and quality assurance (QA) organizations

Security testing should not be the sole approach to getting better code writing. In fact, security testing should not take the place of good security-focused design techniques, noted Dave Roberts, vice president of strategy for [Vyatta](#). The company develops open source router and security products.

"It's easier to design more secure software using a better design methodology than it is to avoid thinking about security up front and try to find problems through testing. There are a variety of libraries and tools in common use that make it easier for developers to write secure software from the start and avoid issues altogether. The libraries and tools are not perfect, however, and so you still need to look for subtle problems once the software is complete, but they do avoid the blatant gaffes," Roberts told LinuxInsider.

Apples Vs. Oranges?

Security experts still bicker over whether open source or proprietary code is more secure. The answer depends on some guess work, as well as a measure of religious fervor.

"The honest answer is that nobody knows, and if anybody tells you otherwise, they're just guessing. There have been some studies that attempt to characterize one being more secure than another. But most of those are provided by security vendors with an agenda," suggested Fisher.

Of course, security is important for both open source and proprietary software. But with proprietary software, there may be a little more control because people can be held accountable, noted Mandeep Khera, CMO for Web application security vendor [Cenzic](#).

"You can also provide security training for your developers, but for open source, it's a wild game. You have to be extra careful," Khera told LinuxInsider.

However, the more-eyes-on-the-code reasoning carries considerable influence in the debate. Open source produces more secure software than proprietary development, proffers Chander Kant, the CEO of [Zmanda](#). Zmanda is an open source backup and recovery software developer.

"The fact that any security issue can be seen by thousands of eyes, in fact, makes it easy to find and fix security issues. If you got proprietary software, just because the security vulnerability may not be seen in the open doesn't make the code more secure," Kant told LinuxInsider.

Wrong Question

Asking security experts to debate the merits of security between the species may be missing the point. In fact, Roberts thinks asking which one is more secure is the wrong question, period. A better question to ask, he said, is how the community handles things once a problem is discovered. On that one point you see a big difference between the open source community and proprietary companies.

"There is no reason to think that either open source software or proprietary software is better than the other when it comes to the fundamental development process. While everybody likes to pick on [Microsoft's](#) (Nasdaq: MSFT) security problems on the proprietary side of things, the reality is that developers have intellects that look like a bell curve. The developers working on open source are no smarter, on average, than the proprietary developers, and both sets of developers will introduce unintended security flaws into the respective code bases," said Roberts.

Tell-Tale Difference

The primary distinguishing criteria between proprietary and open source software is the latter's commitment to finding, fixing and discussing security issues with its user base, Roberts said. There is a sense that there is nothing to hide. Problems happen, you fix them, you make users aware that a fix exists, and then you move on, he said. Not so for proprietary code.

"The same thing can't be said of proprietary software manufacturers, on average. While proprietary companies are finding that they must get better about dealing with security issues, there are many cases we have seen where the companies will wait months after an exploit is brought to their attention to develop and adequate fix," he said.

Other than that, the topic of which software flavor is more secure is oftentimes enough to start quite a heated argument, agreed Sampo Nurmentaus, technical director at [Movial](#). The company develops Linux-based mobile devices.

For him, the openness of open source leads to better, more secure software. Open source developers have a totally different attitude toward the program code.

"An open source developer is like a sculptor that is hired to create a statue to be placed the middle of the city. Since everybody will see it, it needs to be something he or she can be proud of. This attitude toward code quality reduces the traditional overflow vulnerabilities dramatically," Nurmentaus told LinuxInsider. [EGT](#)

Next Article in Bugs

Apple Seals iPhone's SMS Security Leak

July 31, 2009



Security researchers have apparently motivated Apple to kick out a patch that plugs an SMS hole in the iPhone's operating system. As demonstrated in a Thursday presentation at Black Hat, an iPhone flaw allowed hackers to launch malicious attacks through text messages. On Friday, Apple served up a fix.

Copyright 1998-2010 ECT News Network, [Terms of Service](#) | [Privacy Policy](#) | [How To Advertise](#)
Inc. All Rights Reserved.