Researchers tackle issues surrounding security tools for software developers

August 24 2015, by Matt Shipman

Credit: Yuri Samollov. Used under a Creative Commons license

For software programmers, security tools are analytic software that can scan or run their code to expose vulnerabilities long before the software goes to market. But these tools can have shortcomings, and programmers don't always use them. New research from National Science Foundation-funded computer science researcher Emerson Murphy-Hill and his colleagues tackles three different aspects of the issue.

"Our work is focused on understanding the developers who are trying to identify security vulnerabilities in their code, and how they use (or don't use) tools that can help them find those vulnerabilities," says Murphy-Hill, an associate professor of computer science at NC State University. "The one thing that ties all of our work together is that we want to help give programmers the best possible tools and help them use those tools effectively."

In the first of three related papers being presented next week at the Symposium on the Foundations of Software Engineering, a team of computer science and psychology researchers from NC State and Microsoft Research surveyed more than 250 developers on their experiences with security tools. The goal was to determine what influences a developer's use of these tools - and the findings were somewhat surprising.

For one thing, developers who said they worked on products in which security was important were not much more likely to use security tools than other programmers.

Instead, "the two things that were most strongly associated with using security tools were peer influence and corporate culture," Murphy-Hill says. Specifically, people who said they had seen what others do with security tools, and people whose bosses expected them to use security tools, were most likely to take advantage of the tools.

"This research gives software development companies and managers information they can use to effectively influence the adoption of security tools by developers," Murphy-Hill says.

But these tools aren't completely accurate. For example, they can tell programmers there's a problem where no problem actually exists. And the tools aren't always user-friendly. In short, the characteristics of the

tools themselves can affect whether programmers choose to use them.

How Tools Are Used

To shed light on how security tools support developers in diagnosing potential vulnerabilities, Murphy-Hill's team and collaborators from the University of North Carolina at Charlotte devised a separate study, effectively asking: do tools give developers the information they need to determine if there's a real problem and how to fix it?

In this study, the researchers gave 10 developers of varying backgrounds a specific security tool and a substantial chunk of open-source code to examine. The code contained known security vulnerabilities, which were identified by the security tool. Each of the study participants was asked to use the tool, inspect the source code, and say whether each security notification from the tool was real and how they would address the vulnerabilities.

"In many cases, the tool presented multiple possible fixes for a problem, but didn't give programmers much information about the relevant advantages and disadvantages of each fix," Murphy-Hill says. "We found that this made it difficult for programmers to select the best course of action."

The tool would also give developers multiple notifications that seemed to be related to each other - but the notifications didn't give developers information on exactly how the problems related to each other.

"This can be confusing for programmers, and lead to problems if developers don't fully understand how various problems are related to each other or how potential fixes might affect the overall code," Murphy-Hill says.

"More research is needed to really flesh these findings out - we need to expand this study to incorporate more programmers and more security tools," Murphy-Hill says. "But overall, we're hoping that this and related work can help programmers create more effective tools for use by the software development community."

'Bespoke' Tools

One concept that Murphy-Hill and colleagues from NC State propose in a third paper is the idea of "bespoke" tools. The basic idea is to create tools that developers use - including security tools - that are capable of evolving over time as they are used, adapting to each programmer's particular areas of expertise.

"For example, programmers with expertise in addressing security vulnerabilities won't need a security tool that offers extensive information on all of the potential fixes for a given vulnerability - wading through that might slow them down," Murphy-Hill says. "So a bespoke tool might learn to offer only basic information about potential fixes for them. But the tool could also recognize that it needs to leave in that additional information for less security-savvy programmers, who may need it to make informed decisions."

These bespoke tools could learn about a programmer's strengths through both the programmer's interactions with the tool and by analyzing the programmer's code itself, Murphy-Hill says.

More information: The Symposium on the Foundations of Software Engineering is being held Aug. 30 to Sept. 4 in Bergamo, Italy. Lead author of "Quantifying Developers' Adoption of Security Tools" is Jim Witschey, a former computer science graduate student at NC State. The paper was co-authored by Olga Zielinska, Allaire Welk, Murphy-Hill, and Chris Mayhorn of NC State and Thomas Zimmerman of Microsoft

Research. Lead author of "Questions Developers Ask While Diagnosing Potential Security Vulnerabilities with Static Analysis," is Justin Smith, a Ph.D. student at NC State. The paper was co-authored by Brittany Johnson and Murphy-Hill of NC State and Bill Chu and Heather Richter Lipford of UNC-Charlotte. Johnson is also lead author of "Bespoke Tools: Adapted to the Concepts Developers Know."

Provided by North Carolina State University