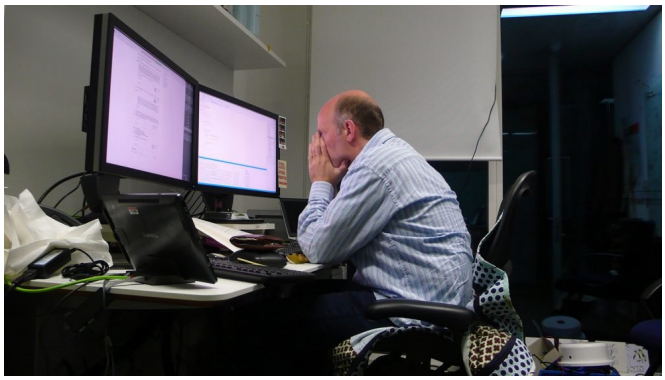
Study finds auto-fix tool gets more programmers to upgrade code

16 October 2017, by Matt Shipman



Credit: Tim Regan. Shared under a Creative Commons license.

Failure to make necessary upgrades to software code can have dire consequences, such as the major data breach at Equifax. A recent study finds that auto-fix tools are effective ways to get programmers to make the relevant upgrades - if programmers opt to use them.

"Most software programs rely, in part, on code in external 'libraries' to perform some of their functions," says Chris Parnin, an assistant professor of computer science at North Carolina State University and senior author of a paper on the work. "If those external libraries are modified to address flaws, programmers need to update their internal code to account for the changes. This is called 'upgrading an out-of-date dependency.' However, for various reasons, many programmers procrastinate, putting off the needed upgrades.

"This is what happened at Equifax," Parnin says. "An external library they relied on had made public that it contained a security flaw. And while the external library was patched, Equifax never got around to updating its internal code. So months after the problem was identified, Equifax was still vulnerable and got hacked.

"Our goal with this project was to assess tools designed to get more programmers to upgrade their out-of-date dependencies. Could they help prevent another Equifax?"

For this study, the researchers looked at thousands of open-source projects on GitHub, an online programming community that fosters collaboration on open-source software projects. Specifically, the researchers looked at different means projects used to incentivize or facilitate upgrades and whether those incentives made any difference.

One group consisted of 2,578 projects that utilized automated pull requests, which notified project owners of needed upgrades to out-of-date dependencies, proposed potential code changes, and ran a small battery of tests to determine if the replacement code was viable. These project owners were still required to approve the changes or modify updated code if it failed initial viability tests.

A second group consisted of 1,273 projects that did not utilize incentives to upgrade out-of-date dependencies.

The researchers found that projects with automated pull requests made 60 percent more of the necessary upgrades than projects that didn't use incentives.

"We also found that the majority of automated pull request projects were using the most up-to-date versions of dependent software, whereas the unincentivized projects were all over the map," Parnin says. "The take-home message here is that we have automated tools that can help programmers keep up with upgrades. These tools can't replace good programmers, but they can make a significant difference. However, it's still up to programmers to put these tools in place and make use of them."

The paper, "Can Automated Pull Requests Encourage Software Developers to Upgrade Out-of-Date Dependencies?", will be presented at the IEEE/ACM International Conference on Automated Software Engineering, Oct. 30-Nov. 3 at the University of Illinois at Urbana-Champaign, Ill.

More information: "Can Automated Pull Requests Encourage Software Developers to Upgrade Out-of-Date Dependencies?" Presented: IEEE/ACM International Conference on Automated Software Engineering, Oct. 30-Nov. 3 at the University of Illinois at Urbana-Champaign, Ill. chrisparnin.me/pdf/VersionBot17.pdf

Provided by North Carolina State University