Giving web developers tools to protect their sites and users

February 6 2015, by Josie Pipkin

Most Internet users know that practicing good online hygiene – never clicking on spam, choosing strong passwords and setting up two-factor authentication – is essential for protecting their personal information. They typically don't know, however, that unless the developers of the websites they visit take similar precautions, they could still be at risk.

That's because Web developers increasingly embed strings of codes, or scripts, in their websites that have been developed by third parties, some reputable and others less so.

These scripts serve a variety of purposes. Some make browsing the Web more convenient. Single sign-on services, for instance, offer users the ability to log in to a variety of websites using their accounts from a provider such as Facebook or Twitter. Other services inject ads for the site to earn revenue or analyze who visits the page and what they do. Developers rely on third-party scripts because they make website design much more efficient.

University of Virginia computer science professor David Evans and graduate student Yuchen Zhou are developing tools that can help busy developers deploy these services safely.

"In many cases, developers lack the expertise, time or incentive to identify and correct the vulnerabilities caused by third-party scripts," Evans said. "We focus on ways to automate the detection of these vulnerabilities and limit the damage they can do."

Left uncorrected, vulnerabilities can open the way for hackers to impersonate a user and gain access to their personal information.

In the case of single sign-on services, there are at least two causes for the vulnerabilities. The first are flaws in the software development kits that these organizations distribute to developers. Zhou and Evans and collaborators at Microsoft took an exhaustive look at three such kits, identifying bugs that could produce vulnerabilities. Facebook awarded them with three $1,500 "bug bounties" for their work.

More commonly, the fault lies with the developers, who lack the knowledge to securely integrate the services. "Either the documentation is not clear or the developers fail to follow it correctly," Zhou said.

Evans and Zhou's response was to build a tool – SSOScan – that can automatically review a site for vulnerabilities caused by using Facebook single sign-on. A developer simply enters the website URL and receives an analysis within a short period of time.

These vulnerabilities are common. Zhou ran SSOScan on the top 20,000 U.S. websites. Of the sites that used Facebook's single sign-on, 20 percent had at least one type of vulnerability.

"The challenge is to convince busy developers to act on this information," Evans said. "In an ideal world, they would have to use something like SSOScan to validate their site during the development process."

Zhou also devised a tool that developers could use to constrain the activity of the hundreds of scripts they typically embed in a website. In the process of performing analytics or placing ads, these scripts gather user-generated information on a page and send it to their servers. There is no guaranteeing, however, that these servers are secure or that the

companies offering these services are trustworthy. In other cases, scripts ostensibly offering a benign service like analytics could take over the page, replacing the site-owner's ads with their own ads or stealing private user information from the page.

Rather than try to develop rules for every page on a site, Zhou's tool generates a set of policies that can be applied to the site as a whole. It catalogs the elements of the site based on their content and their relationship to the structure of the page and specifies which elements can be accessed by specific scripts. "We've found that using a white list approach – specifying which elements a script can use rather than those it can't – is more effective because it is easier to automatically identify public elements like an ad placeholder than sensitive information in the page," Zhou said. The site owner can then review the policies and grant permissions accordingly.

These projects are funded by a grant from the National Science Foundation and a gift from Google and are necessary because, as Evans noted, the business side of the Web has evolved much faster than the methods to ensure its security.

"In the absence of a more comprehensive, browser-based approach to security, the tools we're developing are useful ways of addressing the issues of complexity and trust that developers face," he said.

Provided by University of Virginia