Securing the web: New tool would automatically plug holes that hackers exploit

October 8 2009, by Larry Hardesty

Graphic: Christine Daniloff

(PhysOrg.com) -- More and more, malicious hackers are exploiting web site security holes to attack their victims' computers. Programmers try to identify those holes in advance and plug them with code that performs security checks; but if they find a hundred holes and miss one, their programs are still insecure. At next week's ACM Symposium on Operating Systems Principles, however, MIT researchers will present a new system called Resin, which automatically calls up security checks whenever they're required, even in unforeseen circumstances.

Typically, web programmers will associate security checks with particular application functions. If you belonged to a social-networking

site, for instance, you might be able to e-mail your friends, or post remarks on their pages, or comment on their own posts, or tag their pictures, and so on. Each of these operations executes its own chunk of code, and the developer will usually attach a security check to each chunk, to ensure that the user is authorized to invoke it. (These types of security checks operate in the background: they don't require you, for instance, to reenter your user name and password.) Many web applications also "sanitize" data posted by their subscribers: if a friend posts something to your social-network page, the application probably won't show you the post without inspecting it for malicious code.

"We've looked at a lot of these web applications, and there's literally hundreds of places where these checks happen," says Nickolai Zeldovich, an assistant professor in MIT's Computer Science and Artificial Intelligence Lab. Indeed, Zeldovich and his colleagues identified one popular web application that sanitized data in more than 1,400 places (but still had about 60 security holes).

They also, however, identified a feature that web application security checks usually had in common: "Namely," Zeldovich says, "it's that the same data is being handled in all these hundreds of places."

So Zeldovich, grad students Alexander Yip and Xi Wang, and Professor Frans Kaashoek developed a system that associates security checks with particular chunks of data rather than with particular chunks of code. Any attempt to access the data, by any imaginable route, invokes the check.

The researchers modified 12 existing applications written in the popular web programming languages Python and PHP so that they used the Resin system. In experiments, the modified applications repelled attacks that exploited known security holes. But the researchers also developed their own attacks, which Resin thwarted as well.

For programmers, the new system should be easy to adopt. They're already writing code for security checks and sanitization anyway; now, they'd have to write it only once, instead of pasting it into their programs in hundreds of different places.

But the MIT system relies on additional software that tracks data as they flow through an application, to make sure that security rules remain associated with the information wherever it's being stored and however it's being used. And the data tracker presents the biggest obstacle to commercial adoption.

Web applications need to run on any type of computer, regardless of the operating system or web browser being used, so web languages like Python and PHP require an extra layer of software called a "runtime" to translate code into the language spoken by a given machine. Generally, the organizations that develop new programming languages also maintain the runtimes, which undergo sequential releases, just like any commercial program. The MIT system's data tracker would have to be incorporated into several different languages' runtimes, which could be a hard sell.

"At least in PHP, the focus tends to be on performance," says Eddie Kohler, an assistant professor of computer science at UCLA. Resin, Kohler says, "shows that you can do it without too much of a performance loss," but "it's not zero; it's not a performance gain." Kohler points out, however, that Resin could gain traction with the runtime gatekeepers if it first proves itself in some particular, real-world instances. "A place like, maybe Facebook, say, that runs other people's code on their servers already has an environment where they're much more worried about people stealing data out of their servers than they are necessarily about getting the last two percent of performance," Kohler says. "I expect that as it gets deployed, it would get deployed by individual companies first."