veracode.com

Why Even Google Is Susceptible to the Most Basic Website Vulnerabilities

By John Zorabedian

6-7 minutes

This week's National Cyber Security Awareness Month theme of "recognizing and combating cybercrime" brings up an elementary but crucial point about why our efforts to fight cybercrime seem inadequate for the challenge: it can be really difficult to fix what's broken even when we know exactly what the problem is.

Here's an example. When a sick patient comes to a doctor complaining about pain, it's important to immediately address the patient's suffering. A good doctor will want to understand what caused the pain in the first place. With abdominal pain, for example, is it temporary – such as gas caused by an unhealthy diet? Or is it a chronic condition like Crohn's disease? When the problem is identified, the doctor can prescribe the right medication and advise the patient on preventive measures to promote ongoing good health.

But then the real challenge begins. The patient has to go forth and actually follow doctor's orders, make health-conscious decisions and change lifestyles.

Cybercrime is a huge and growing problem, with many types of attacks – from financially-motivated crime like 419 email scams and insidious ransomware, to politically-motivated hacktivist attacks and state-sponsored advanced persistent threat schemes. One of the most persistent and widespread problems in cybersecurity is application vulnerabilities – application-layer attacks are the most common source of confirmed breaches, according to the 2016 Verizon Data Breach Investigation Report.

Although we have diagnosed the source of the application-layer threat – vulnerabilities that result from coding weaknesses – we still fail in our efforts to address the threats with preventive measures. Veracode research shows the enormous scale of the problem. More than 60 percent of applications fail to pass the OWASP Top 10 security policies on initial assessment, according to the latest Veracode State of Software Security report.

If addressing failures in application security were simply a matter of throwing enough money and resources at the problem, you might expect big, resource-rich companies to have it under control. But that is simply not the case. Even the biggest and brightest stars in the high-tech galaxy, like Facebook and Google, struggle with vulnerabilities in their applications. Just recently, a security researcher discovered that the French version of Google's website had a Cross-Site Scripting vulnerability that bad actors could have exploited to steal private information or take over a victim user's browser.

The cause of common but preventable application vulnerabilities like Cross-Site Scripting (or XSS) is deep-seated and systemic – symptomatic of the way applications are built and how developers are trained. The vast majority of applications are built using third-party and open source components, so when there is defective code in commonly-used components, vulnerabilities are widely distributed. Developers may not be aware that they are using components with defective code and organizations may not know what components they are using and where.

In addition, developers aren't trained in secure coding. Even though XSS is an OWASP Top 10 vulnerability, just 11 percent of developers know how to prevent XSS, according to a 2014 Denim Group study. Is it really surprising that Veracode research shows 50% of applications have at least one XSS vulnerability on initial assessment?

Given the systemic nature of these problems, the solution to application risks must be a systematic approach to security, including:

- A comprehensive application security (AppSec) program that includes static and dynamic testing of all code, including software composition analysis of open-source, third-party components and vendor-supplied software.
- A developer-friendly training program, such as online courses and video tutorials, to help developers recognize common coding weaknesses and learn preventive, secure coding practices.

Of course, it's not possible to eliminate all risk. People and programs are fallible. Mistakes happen. Applications we think are secure today may turn out to have built-in defects we won't recognize until tomorrow. But there are some things we can control, particularly with developer training. It's possible to get that low-hanging fruit and cut down on easily preventable errors.

Online learning is proven to work: according to the 2016 State of Software Security report, development teams with an eLearning program reduce flaw density by 55 percent from initial to subsequent scans, on average. That's opposed to just a 9 percent reduction in flaw density for teams without an eLearning program.

If you have a developer training program, how could you make it better? And if you don't, what better time to begin your focus on developer training than National Cyber Security Awareness Month?

If you want to know more about how Veracode is helping developers learn secure coding skills, watch a short video demonstration of our eLearning platform. And you can register to see a free sample course in XSS, which demonstrates a basic XSS attack and provides remediation training for Java and .NET developers.

John Zorabedian is a blogger, content marketer, and research editor. He has a background in marketing and journalism, writing about IT security, technology, business, politics and culture. He lives and works in the Boston area.