Web Application Security: Don't Bolt It On; Build It In

How secure are your Web applications? Unless you conduct application vulnerability testing throughout the lifespan of your applications, there's no way for you to know about your web application security. That's not good news for your security or regulatory compliance efforts.

Companies make significant investments to develop high-performance Web applications so customers can do business whenever and wherever they choose. While convenient, this 24-7 access also invites criminal hackers who seek a potential windfall by exploiting those very same highly available corporate applications.

The only way to succeed against Web application attacks is to build secure and sustainable applications from the start. Yet, many businesses find they have more Web applications and vulnerabilities than security professionals to test and remedy them — especially when application vulnerability testing doesn't occur until after an application has been sent to production. This leads to applications being very susceptible to attack and increases the unacceptable risk of applications failing regulatory audits. In fact, many forget that compliance mandates like Sarbanes-Oxley, the Health Insurance Portability and Accountability Act, Gramm-Leach-Bliley, and European Union privacy regulations, all require demonstrable, verifiable security, especially where most of today's risk exists — at the Web application level.

In an attempt to mitigate these risks, companies use firewalls and intrusion detection/prevention technologies to try to protect both their networks and applications. But these web application security measures are not enough. Web applications introduce vulnerabilities, which can't be blocked by firewalls, by allowing access to an organization's systems and information. Perhaps that's why experts estimate that a majority of security breaches today are targeted at Web applications.

One way to achieve sustainable web application security is to incorporate application vulnerability testing into each phase of an application's lifecycle — from development to quality assurance to deployment — and continually during operation. Since all Web applications need to meet functional and performance standards to be of business value, it makes good sense to incorporate web application security and application vulnerability testing as part of existing function and performance testing. And unless you do this — test for security at every phase of each application's lifecycle — your data probably is more vulnerable than you realize.

Neglecting Application Vulnerability Testing: Risks and Costs of Poor Security

Consider supermarket chain Hannaford Bros., which reportedly now is spending billions to bolster its IT and web application security — after attackers managed to steal up to 4.2 million credit and debit card numbers from its network. Or, the three hackers recently indicted for stealing thousands of credit card numbers by inserting packet sniffers on the corporate network of a major restaurant chain.

The potential costs of these and related Web application attacks add up quickly. When you consider the expense of the forensic analysis of compromised systems, increased call center activity from upset customers, legal fees and regulatory fines, data breach disclosure notices sent to affected customers, as well as other business and customer losses, it's no surprise that news reports often detail incidents costing anywhere from $20 million to $4.5 billion. The research firm Forrester estimates that the cost of a security breach ranges from about $90 to $305 per compromised record.

Other costs that result from shoddy web application security include the inability to conduct business during denial-of-service attacks, crashed applications, reduced performance, and the potential loss of intellectual property to competitors.

What's so surprising, aside from all of the security and regulatory risks we've described, is that it's actually more cost effective to use application vulnerability testing to find and fix security-related software defects during development. Most experts agree that while it costs a few hundred dollars to catch such flaws during the requirements phase, it could cost well over $12,000 to fix that same flaw after the application has been sent to production.

There's only one way to ensure that your applications are secure, compliant, and can be managed cost-effectively, and that's to adapt a lifecycle approach to web application security.

The Web Application Security Lifecycle

Web applications need to *start secure* to *stay secure*. In other words, they should be built using secure coding practices, go through a series of QA and application vulnerability testing, and be monitored continually in production. This is known as the web application security lifecycle.

Remedying security problems during the development process via application vulnerability testing isn't something that can be achieved immediately. It takes time to integrate security into the various stages of software development. But any organization that has undertaken other initiatives, such as implementing the Capability Maturity Model (CMM) or even undergoing a Six Sigma program, knows that the effort is worth it because systematized application vulnerability testing processes provide better results, more efficiency, and cost savings over time.

Fortunately, application assessment and security tools are available today that will help you to get there — without slowing project schedules. But, in order to strengthen development throughout the application life cycle, it's essential to pick application vulnerability testing tools that aid developers, testers, security professionals, and application owners and that these toolsets integrate tightly with popular IDEs, such as Eclipse and Microsoft's Visual Studio.NET for developers.

And just as standardization on development processes — such as RAD (rapid application development) and agile — brings development efficiencies, saves time, and improves quality, it's clear that strengthening the software development life cycle, possessing the right security testing tools, and placing software security higher in the priority list are excellent and invaluable long-term business investments.

What types of web application security tools should you look for? Most companies are aware of network vulnerability scanners, such as Nessus, that evaluate the infrastructure for certain types of vulnerabilities. But fewer are aware of application vulnerability testing and assessment tools that are designed to analyze Web applications and Web services for flaws specific to them, such as invalid inputs and cross-site scripting vulnerabilities. These Web application security and vulnerability scanners are not only useful for custom-built applications but also to make sure that commercially acquired software is secure.

There are also web application security tools that help instill good security and quality control earlier and throughout development. For instance, these application vulnerability testing tools help developers find and fix application vulnerabilities automatically while they code their Web applications and Web services. There also are quality inspection applications that help QA

professionals incorporate Web application security and application vulnerability testing into their existing management processes automatically.

It's also important to know that technology alone won't get the job done. You need management support, too. And no matter how large or small your development efforts, all stakeholders — business and application owners, security, regulatory compliance, audit, and quality assurance teams — should have a say from the beginning, and benchmarks must be set for quality application vulnerability testing.

While it may seem like a daunting undertaking at first, the web application security lifecycle approach actually saves money and effort by establishing and maintaining more secure applications. Remedying security defects after an application is released requires additional time and resources, adding unanticipated costs to finished projects. It also diverts attention from other projects, potentially delaying time to market of new products and services. Moreover, you'll save on the excessive expense of having to fix flaws after the application has been deployed, and you've failed regulatory audits — and you'll avoid the embarrassment of being the next security breach news headline.

About the Author

Caleb Sima is the former co-founder and CTO of SPI Dynamics, which was acquired by HP Software in August 2007. He is now responsible for directing the lifecycle of the HP's Web application security solutions and is the Chief Technologist for the HP Application Security Center. Prior to joining HP, Caleb worked for the elite X-Force R&D team at Internet Security Systems and as a security engineer for S1 Corporation. Caleb is a frequent speaker and press resource on Internet attacks and has contributed to *Baseline Magazine* and *(IN)Secure Magazine* as well as being featured in the Associated Press. He is also a Microsoft Most Valuable Professional (MVP) in Visual Developer Security. For more details on enhancing web security, please visit www.HP.com.