Effective Controls for Attaining Continuous Application Security Throughout the Web Application Development Life-cycle

Given the choice, every organization would want secure Web sites and applications from the Web application development phase all the way through the software development life-cycle. But why is that such a challenge to attain? The answer is in the processes (or lack thereof) that they have in place.

While individual and ad hoc Web application security assessments certainly will help you improve the security of that application or Web site, soon after everything is remedied, changes in your applications and newfound vulnerabilities mean new security problems will arise. So, unless you put into place continuous security and quality assurance controls throughout the software development life-cycle, from the initial phases of Web application development through production, you're never going to reach the high levels of ongoing security you need to keep your systems safe from attack - and your costs associated with fixing security weaknesses will continue to be high.

In the first two articles, we covered many of the essentials you need to know when conducting Web application security assessments, and how to go about remedying the vulnerabilities those assessments uncovered. And, if your organization is like most, the first couple of Web application assessments were nightmares: reams of low, medium, and high vulnerabilities were found and needed to be fixed by your web application development team. The process required that tough decisions be made on how to fix the applications as quickly as possible without affecting systems in production, or unduly delaying scheduled application rollouts.

But those first few web application assessments, while agonizing, provide excellent learning experiences for improving the software development life-cycle. This article shows you how to put the organizational controls in place to make the process as painless as possible and an integrated part of your Web application development efforts. It's a succinct overview of the quality assurance processes and technologies necessary to begin developing applications as securely as possible from the beginning, and keeping them that way. No more big surprises. No more delayed deployments.

Secure Web Application Development: People, Process, and Technology

Building highly secure applications begins early in the software development life-cycle with your developers. That's why instilling application security awareness through Web application development training is one of the first things you want to do. You not only want your developers armed with the latest knowledge on how to code securely – and how attackers exploit weaknesses – but you want them to know how important (and much more efficient) it is to consider security from the start. This awareness building shouldn't end with your Web application development team. It needs to include everyone who plays a part in the software development life-cycle: your quality and assurance testing teams, who need to know how to properly identify potential security defects, and your IT management team, who need to understand how to invest organizational resources most effectively to develop security applications, as well as how to successfully evaluate such essential technologies as Web application security scanners, Web application firewalls, and quality assurance toolsets.

By building awareness throughout the Web application development life-cycle, you're building one of the most central controls necessary to ensure the security of your Web applications. And while training is essential, you can't depend on it to make certain that your systems are built securely. That's why training needs to be reinforced with additional controls and technology. You need to begin to put into place the elements of a secure Software Development Life-cycle, or SDLC.

Essential Elements of Secure Software Development Life-cycle Processes

A secure software development life-cycle means having the policies and procedures in place that consider - and enforce - secure Web application development from conception through defining functional and technical requirements, design, coding, quality testing, and while the application lives in production. Developers must be trained to incorporate security best practices and checklists in their work: Have they checked their database query filtering, or validated proper input handling? Is the application being developed to be compliant with best programming practices? Will the application adhere to regulations, such as HIPAA or PCI DSS? Putting these types of procedures in place will dramatically improve security during the Web application development process. Having developers checked field inputs and looked for common programming mistakes as the application is being written also will make future application assessments flow much more smoothly.

While developers need to test and assess the security of their applications as they're being developed, the next major test of the software development life-cycle processes comes after the Web application development is completed. This is when the entire application, or a module, is ready to be sent to the formal testing phase that will be conducted by quality assurance and security assessors. It's during this phase of the software development life-cycle that quality assurance testers, in addition to their typical tasks of making sure performance and functional requirements are met, look for potential security problems.

Many companies make the mistake, during this phase, of not including members of the IT security team in this process. It's our opinion that IT security should have input throughout the software development life-cycle, lest a security issue surface later in the Web application development process - and what could have been a small problem is now a big problem.

Putting these types of processes in place is difficult work, and may seem onerous at first. But the truth is that the payoff can be huge: your applications will be more secure and your future security assessments won't feel like fire drills. There are software development life-cycle models and methodologies that could help direct you, such as the Application Security Assurance Program (ASAP), which puts a number of guiding principles in place necessary for building secure code, including executive commitment, considering security from the beginning of Web application development, and the adoption of metrics to measure coding and process improvements over time. A good primer is The Security Development Life-cycle by Michael Howard and Steve Lipner (Microsoft Press, 2006).

How Technology Helps Enforce and Maintain the Secure SDLC

Human nature being what it is, people tend to slip back into their old sloppy ways if new behaviors (the software development life-cycle processes we discussed earlier) are not enforced. That's where technology can play a role. The right tools not only help to automate the security assessment and secure coding process; they also can help keep in place the Web application development framework necessary for success.

As discussed in the first article of this series, at the very minimum you'll need a Web application security scanner to assess your custom-built as well as your commercially-acquired software. Depending on the size of your Web application development team, and how many applications you're working on at any given time, you'll want to consider other tools that will improve your software development life-cycle processes as well. For instance, quality and assurance tools are available that integrate directly into application performance and quality testing programs that many organizations already use, such as those from IBM and

HP. With this integration of security into quality and performance testing, quality assurance teams can concurrently manage functional and security testing from a single platform.

Put Baselines in Place (But Keep it Simple in the Early Days)

Now that security training is in place, and you have consistent, secure Web application development methodologies, along with the assessment and development tools you need, it's a good time to start measuring your progress.

At first, all of these changes in your software development life-cycle processes will feel disruptive and time consuming. So, executives and managers, as well as the Web application development team and auditors, are certainly going to want to see results from all the new work that they've put in place. Everyone will want metrics and baselines: Are our applications more secure? Are developers coding better? The only way to answer these questions is to start measuring progress. But, in the beginning, don't fall into the trap of measuring too much.

In the initial days of putting software development life-cycle processes in place, we strongly advise that you keep the measurements simple. Do not get overwhelmed with tracking too many types of vulnerabilities. In fact, you probably don't want to try to track and extinguish every class of vulnerability at once. We've seen this mistake made many times: enterprises try to fix vulnerabilities discovered in every part of the software development life-cycle in a big bang. Then, at the end of a year, they end up with a dozen completely vulnerable applications, and with no money in place to fix everything that needs to be fixed. They end up scrambling, disheartened, and getting nowhere. That's not the way to do it.

That's why, in the beginning, we've learned that a sensible - and attainable - approach to securing the Web application development process is to decide which are your most prevalent and severe vulnerabilities. If they include SQL Injection or logic errors that could provide unauthorized access to an application, then that's your initial focus. Pick the most critical vulnerabilities that will make significant differences, based on your assessment and the nature of your systems and business. These will be the first vulnerabilities you want to track during their march to extinction (at least from within your applications).

Once your Web application development team gets used to the process of fixing certain classes of vulnerabilities, you can add the next most pressing class (or two) of vulnerabilities to the mix. By slowing adding new classes of vulnerabilities into your formal software development life-cycle processes, you will have the opportunity to smooth any problems or kinks in the process. And your Web application development teams will grow increasingly accustomed to the process. There'll be no big shocks, and over the course of months, and years, you'll see dramatic improvement over your first few baselines.

By putting into place the essential controls and technologies outlined in this article, you're now well on the pathway to Web application development that is consistently secure. Your reward will be a software development life-cycle process that will flow much more smoothly and cost effectively; you'll have caught problems early in the development process, so your regulatory audits will flow more smoothly. And you'll have greatly reduced the chances of a successful attack against your Web sites.