Implementing Effective Vulnerability Remediation Strategies Within the Web Application Development Lifecycle

Once you've completed a security assessment as a part of your web application development, it's time to go down the path of remediating all of the security problems you uncovered. At this point, your developers, quality assurance testers, auditors, and your security managers should all be collaborating closely to incorporate security into the current processes of your software development lifecycle in order to eliminate application vulnerabilities. And with your Web application security assessment report in hand, you probably now have a long list of security issues that need to be addressed: low, medium, and high application vulnerabilities; configuration gaffes; and cases in which business-logic errors create security risk. For a detailed overview on how to conduct a Web application security assessment, take a look at the first article in this series, *Web Application Vulnerability Assessment: Your First Step to a Highly Secure Web Site.*

First Up: Categorize and Prioritize Your Application Vulnerabilities

The first stage of the remediation process within web application development is categorizing and prioritizing everything that needs to be fixed within your application, or Web site. From a high level, there are two classes of application vulnerabilities: development errors and configuration errors. As the name says, web application development vulnerabilities are those that arose through the conceptualization and coding of the application. These are issues residing within the actual code, or workflow of the application, that developers will have to address. Often, but not always, these types of errors can take more thought, time, and resources to remedy. Configuration errors are those that require system settings to be changed, services to be shut off, and so forth. Depending on how your organization is structured, these application vulnerabilities may or may not be handled by your developers. Oftentimes they can be handled by application or infrastructure managers. In any event, configuration errors can, in many cases, be set straight swiftly.

At this point in the web application development and remediation process, it's time to prioritize all of the technical and business-logic vulnerabilities uncovered in the assessment. In this straightforward process, you first list your most critical application vulnerabilities with the highest potential of negative impact on the most important systems to your organization, and then list other application vulnerabilities in descending order based on risk and business impact.

Develop an Attainable Remediation Roadmap

Once application vulnerabilities have been categorized and prioritized, the next step in web application development is to estimate how long it will take to implement the fixes. If you're not familiar with web application development and revision cycles, it's a good idea to bring in your developers for this discussion. Don't get too granular here. The idea is to get an idea of how long the process will take, and get the remediation work underway based on the most time-consuming and critical application vulnerabilities first. The time, or difficulty estimates, can be as simple as easy, medium, and hard. And remediation will begin not only with the application vulnerabilities that pose the greatest risk, but those that also will take the longest to time correct. For instance, get started on fixing complex application vulnerabilities that could take considerable time to fix first, and wait to work on the half-dozen medium defects that can be rectified in an afternoon. By following this process during web application development, you won't fall into the trap of having to extend development time, or delay an application rollout because it's taken longer than expected to fix all of the security-related flaws.

This process also provides for excellent follow-up for auditors and developers during web application development: you now have an attainable road map to track. And this progression will reduce security holes while making sure development flows smoothly.

It's worth pointing out that that any business-logic problems identified during the assessment need to be carefully considered during the prioritization stage of web application development. Many times, because you're dealing with logic—the way the application actually flows—you want to carefully consider how these application vulnerabilities are to be resolved. What may seem like a simple fix can turn out to be quite complicated. So you'll want to work closely with your developers, security teams, and consultants to develop the best business-logic error correction routine possible, and an accurate estimate of how long it will take to remedy.

In addition, prioritizing and categorizing application vulnerabilities for remediation is an area within web application development in which consultants can play a pivotal role in helping lead your organization down a successful path. Some businesses will find it more cost effective to have a security consultant provide a few hours of advice on how to remedy application vulnerabilities; this advice often shaves hundreds of hours from the remediation process during web application development.

One of the pitfalls you want to avoid when using consultants during web application development, however, is failure to establish proper expectations. While many consultants will provide a list of application vulnerabilities that need to be fixed, they often neglect to provide the information that organizations need on how to remedy the problem. It's important to establish the expectation with your experts, whether in-house or outsourced, to provide details on how to fix security defects. The challenge, however, without the proper detail, education, and guidance, is that the developers who created the vulnerable code during the web application development cycle may not know how to fix the problem. That's why having that application security consultant available to the developers, or one of your security team members, is critical to make sure they're going down the right path. In this way, your web application development timelines are met and security problems are fixed.

Testing and Validation: Independently Make Sure Application Vulnerabilities Have Been Fixed

When the next phase of the web application development lifecycle is reached, and previously identified application vulnerabilities have (hopefully) been mended by the developers, it's time to verify the posture of the application with a reassessment, or regression testing. For this assessment, it's crucial that the developers aren't the only ones charged with assessing their own code. They already should have completed their verification. This point is worth raising, because many times companies make the mistake of allowing developers to test their own applications during the reassessment stage of the web application development lifecycle. And upon verification of progress, it is often found that the developers not only failed to fix flaws pegged for remediation, but they also have introduced additional application vulnerabilities and numerous other mistakes that needed to be fixed. That's why it's vital that an independent entity, whether an in-house team or an outsourced consultant, review the code to ensure everything has been done right.

Other Areas of Application Risk Mitigation

While you have full control over accessing your custom applications during web application development, not all application vulnerabilities can be fixed quickly enough to meet immovable deployment deadlines. And discovering a vulnerability that could take weeks to rectify in an application already in production is nerve-wracking. In situations like these, you won't always have control over reducing your Web application security risks. This is especially true for applications you purchase; there will be application vulnerabilities that go unpatched by the vendor for extended periods of time. Rather than operate at high levels of risk, we recommend that you consider other ways to mitigate your risks. These can include segregating applications from other areas of your network, limiting access as much as possible to the affected application, or changing the configuration of the application, if possible. The idea is to look at the application and your system architecture for other ways to reduce risk while you wait for the fix. You might even consider installing a web application firewall (a specially crafted firewall designed to secure web applications and enforce their security policies) that can provide you a reasonable interim solution. While you can't rely on such firewalls to reduce all of your risks indefinitely, they can provide an adequate shield to buy you time while the web application development team creates a fix.

As you have seen, remedying web application vulnerabilities during the web application development lifecycle requires collaboration among your developers, QA testers, security managers, and application teams. The associated processes can seem laborious, but the fact is that by implementing these processes, you'll cost-effectively reduce your risk of application-level attacks. Web application development is complex, and this approach is less expensive than reengineering applications and associated systems after they're deployed into production.

That's why the best approach to web application security is to build security awareness among developers and quality assurance testers, and to instill best practices throughout your Web application development life cycle—from its architecture throughout its life in production. Reaching this level of maturity will be the focus of the next installment, **Effective Controls For Attaining Continuous Application Security**. The third and final article will provide you with the framework you need to build a development culture that develops and deploys highly secure and available applications—all of the time.

About Caleb Sima

Caleb Sima is the co-founder of SPI Dynamics (www.spidynamics.com), a web application security products company. He currently serves as the CTO and director of SPI Labs, SPI Dynamics' R&D security team. Prior to co-founding SPI Dynamics, Caleb was a member of the elite X-Force R&D team at Internet Security Systems, and worked as a security engineer for S1 Corporation. Caleb is a regular speaker and press resource on web application security testing methods and is a co-author of the book titled, *Hacking Exposed Web Applications: Web Security Secrets & Solutions, Second Edition*.

About Vincent Liu

Vincent Liu, CISSP, CCNA, is the managing director at Stach & Liu (www.stachliu.com), a professional services firm providing advanced IT security solutions. Before founding Stach & Liu, Vincent led the Attack & Penetration and Reverse Engineering teams for the Global Security unit at Honeywell International. Vincent is an experienced speaker and has presented his research at conferences including BlackHat, ToorCon, and Microsoft BlueHat. He has been published in interviews, journals, and books with highlights including: *Penetration Tester's Open Source Toolkit*; *Writing Security Tools and Exploits*; *Sockets, Shellcode, Porting, and Coding*; and the upcoming *Hacking Exposed: Wireless*.