



An Innovative Way to do Security Code Reviews

By Brad C. Johnson and Jonathan Gossels

brad.johnson@systemexperts.com; jon.gossels@systemexperts.com

Web applications are being deployed at fast rates. Unfortunately, exploiting vulnerabilities in these business transaction sites has become one of the highest security risks on the Internet today. Why is that? The need for rapid development and deployment of Web-based business functionality has caused many organizations to put aside their time-tested application design and development methodologies used in pre-Internet environments. The result is a high percentage of business applications being deployed on the Internet before they have been scrutinized for security-related problems. This affords hackers and other determined intruders ample opportunity to access or even compromise sensitive information.

An efficient and effective security code review methodology is needed to compensate for the control deficiencies in today's typical Web application development process.

Web Application Development

The typical business application is the result of a long and complex process—architecture, design, implementation, functional testing, quality assurance testing, production deployment, ongoing maintenance, and functional enhancement. During the last ten years, the need for security-oriented code reviews of business applications has increased significantly. The main reason for this is simple. Time-to-market competitive pressure to introduce enhanced functionality in the fast-paced Web world is compressing the development cycle. Most development organizations have their hands full just trying to perform function, unit and feature testing. They don't have the time to put their applications through the rigors of an assessment focused on finding and resolving security-related problems. As a consequence, security issues are often unrecognized or ignored. Too often when security problems are found, the project plan calls for them to be addressed in a later release, after the application has already gone into production.

While security is rarely a priority of the application development teams, it is increasingly recognized by senior management as an important aspect of sound business practice. This creates a dynamic tension that results in many organizations understanding the need to proactively review their policies, procedures and applications to help prevent security breaches. To fully understand the overall security and business risk, one needs to consider several different aspects of the environment (for example, networks, hosts, software infrastructure, and applications). There are a variety of techniques that are commonly used to assess these areas, including conducting penetration analysis, scanning and hardening hosts and networks, running exploit tools, and per-

forming third-party security assessments of applications and its supporting infrastructure.

For any Web-based application, however, there is no substitute for a hands-on review of the actual source code. The problem lies in the fact that with object-oriented code, reusable modules, outsourced development, and with the time pressures of the marketplace, these often huge code-bases are deployed without a qualified review of the key features and functions that might be subverted by a hacker to gain direct access to data or systems. Some type of code review process is needed.

While a thorough evaluation of every line of code would be helpful, it is usually an impractical option. Unfortunately, because most organizations can't afford (both time and money) a line-by-line code review, they perform no code review. Many organizations take this course of inaction because they are unaware that viable alternative approaches exist.

This problem is compounded by a lack of security skills. Most business application developers are hired because of their skills in design, implementation and testing of specific programming languages and not on writing secure code. Therefore, many of these applications have a myriad of inherent security flaws that make them vulnerable when deployed in an intranet, extranet, or Internet environment. Significant benefit can be gained by identifying these problems before the applications are released into production.

Recognizing this reality, over the past year, we worked with a number of leading financial firms to develop and refine an effective and practical alternative to the unaffordable line-by-line code review process. Our approach combines interactive discussions (this is called the workshop phase) and line-by-line review of a carefully selected "security-important" set of code modules (this is called the review phase) to quickly and cost effectively get to the heart of not only architectural and design flaws, but also specific coding problems.

This code workshop and review consists of two distinct parts—an interactive review of the application design and implementation and a hands-on analysis of selected portions of code. Over the past year, it has been clear that this process consistently generates well-grounded recommendations to make the networked-based application more secure and less susceptible to attack. It is a methodology you should consider adopting within your own organization.

Key Considerations

This methodology minimizes the burden it places on an organization's budget and resources. For example, when done properly, there is no need to prepare detailed documentation (something that is often missing for

many business applications). Similarly, the process does not burden your staff. Typically, a handful of your people will only need to dedicate one day to the effort.

The highly interactive, collegial nature of the on-site workshop fosters a tighter relationship between the development, deployment and security teams. The developers experience the "ahahs" when design or coding problems are uncovered in the workshop so they know exactly what needs to be fixed and the agreed-upon way to do it.

Proper staffing of the project is critical to efficiency and success. This process is most effective when staff who are familiar with your overall application architectures, design models and development practices participate. Similarly, the review process is most efficient when programmers who are intimately familiar with the details of the specific project application participate as well. Broad participation in the code workshop phase is essential because creating effective and secure business applications is only secondarily about the technical details of coding in C++, JavaScript, XML, or whatever language is being used. It is primarily about ensuring that the key business objectives are implemented and supported by the entire application environment.

Unlike the line-by-line approach which takes weeks or even months to complete, our code workshop and review methodology generates significant results within the first day and often complete results within a week or two.

The objective of most software quality efforts is to identify problems as early in the application life cycle as possible. Generally, the earlier a problem is found, the less it costs to remedy and the potential business impact is smaller. This methodology accelerates the shakeout of design and coding problems, thereby enabling organizations to solve the problems proactively, rather than waiting for hackers to stumble across them.

By being smart about which modules to analyze in depth, our code workshop and review methodology has a dramatically lower cost structure than traditional line-by-line code review.

Just like every organization should use host, network and exploit scanners to programmatically review its environment for unexpected problems, organizations should also use programmatic code analysis tools to look for common programming language-specific problems. This code workshop and review methodology is intended to complement the use of automated tools.

Let's look at how this methodology really works.

Code Workshop and Review

The code workshop and review methodology takes place in two phases. The first phase is a day or two of highly interactive discussions between the code designers and developers and the development/security experts. The second phase is an intensive independent review of a carefully selected subset of the application's code (this second phase requires virtually no participation from the original developers).

Workshop: Phase 1

The first part of the workshop is focused on understanding the business requirements that drive the functional design and architecture of the application. This approach allows the team to put the code in proper business context. Generally, the workshop discussions progress from high level (for example, what the business is trying to accomplish with the application) to medium level (for example, what is the application's architecture

and key touch points with other applications or system) to low level (for example, how the code is actually constructed and how it works). The technical discussions usually include several interactive walk-throughs of selected portions of the code.

During the remainder of the workshop, the team delves into the actual coding practices used to support the application and identifies the exact code modules that will be analyzed in detail during phase 2, the review phase. Typically, it is not necessary to go over each and every line of the application to understand its security strengths and weaknesses. However, it is vital that the most sensitive security code be properly critiqued.

One frequent unexpected benefit of the code workshop is that the highly interactive nature of the discussions provides a forum for the sponsor's staff to better understand the ideas behind their own design and how the application may be improved. It is surprising how often we've seen clients gain fresh insight into their environment just because they needed to explain something out loud to a team of independent experts!

Topics that are useful to cover in the workshop include the following:

- ▲ Business Context
 - ▼ Assumptions
 - ▼ Requirements
 - ▼ Constraints
 - ▼ Dependencies
 - ▼ Schedule and Future Releases
- ▲ Code Context
 - ▼ Application Architecture
 - ▼ Development Practices
 - ▼ Function Design
 - ▼ Logging Design
 - ▼ Deployment Requirements
 - ▼ Configuration Options
 - ▼ Module Review Selection

Review: Phase 2

Once the code workshop phase is completed, the code review team spends an agreed-upon amount of time performing a manual inspection of the selected code. During that review, the team looks for common security problems and issues such as unvalidated parameters, broken access control, broken account and session management, inappropriate state management, buffer overflows, error handling problems, and insecure use of cryptography. Problems like these are often the way Web-based applications are subverted, providing access to back-end systems and confidential or proprietary information.

It is impossible to articulate every type of problem that a code review might look for, but here are some general areas and specific issues that should be addressed.

- ▲ Functional hygiene
 - ▼ Unvalidated parameters
 - ▼ Unexpected parameter values (for example, special, ".../", local files)
 - ▼ Improper handling of function return values
 - ▼ Buffer overflows
 - ▼ Error handling
 - ▼ Unused code

- ▲ Base security handling
 - ▼ Access control (for example, separate from authentication)
 - ▼ Account and session management (for example, timeouts, segregation)
 - ▼ State management
 - ▼ Use of cryptography
 - ▼ Cookie entropy
 - ▼ Information leakage
 - ▼ Username and password quality

- ▲ Intrusion handling
 - ▼ Unexpected parameter value logging
 - ▼ Unexpected function-call logging
 - ▼ General events and logging (for example, escalation, timestamps)


- ▲ Exploit handling
 - ▼ Command injection
 - ▼ Improper/unauthenticated/unauthorized (SQL) calls

Final Word

While IT leaders and security experts alike agree that code reviews are necessary to ensure the security of critical applications, the cost (both time and money) has discouraged most organizations from performing them. Clearly there has to be a better solution than omitting critical reviews.

The code workshop and review methodology described in this article has proven to be an innovative solution to this problem for several leading financial institutions. They have found that this process is not only faster and less expensive than traditional code reviews, but its holistic approach frequently identifies architectural, design and deployment problems that a standard code review would not discover. Some organizations find that they can perform the methodology with internal staff; others recognize the value in bringing in outside expertise to drive the effort.

It is important to remember that participation of staff that genuinely understands the business drivers and the technology underlying the application is essential to a successful outcome for the project. It is equally important to spend the time necessary to identify the security-critical modules that will undergo detailed analysis.

Finally, the code workshop and review methodology is intended to complement an organization's own code analysis, quality assurance testing and programmatic assessments, rather than replace those vital activities. 

Brad Johnson is vice president and Jonathan Gossels is president of SystemExperts Corporation.