

The Importance of a Source Code Review

Introduction

With the maturation of IT infrastructure and improvements in network security, Internet-connected applications have become a favorite target of hackers and fraudsters. SQL injection and cross-site scripting (XSS) are two examples of application-layer attacks that have been devastating for a number of unlucky organizations. Securing applications is fundamentally different than securing a network. Applications tend to suffer because network security gurus don't understand the inner workings of applications, while programmers and other software development pros aren't trained on security concepts.

Experts agree that building security into the application code is the ultimate solution to having secure software applications. Gartner goes so far as to say that developers are responsible for application security (Gartner report #143844). If that is true, how can an organization know that developers are creating secure applications? The answer is a Source Code Review, performed by experts who are familiar with application vulnerabilities and understand application architecture and code.

Performing a code review is one of the best ways to verify that secure coding practices were used (or are being used) during the development of your applications. This is true whether the application was built in-house or by an outside contractor.

Benefits

There is a financial incentive to perform a Source Code Review on an application. A high-quality, security-focused review of the programming code can significantly reduce the chances of a security breach and its ensuing financial repercussions. Finding and fixing software vulnerabilities in the development cycle prior to release is much less expensive than correcting them after code has been deployed to production. Furthermore, getting a Source Code Review from an external party shows due diligence on the part of an organization.

Code reviews also provide benefit in terms of meeting compliance initiatives, especially the Payment Card Industry Data Security Standard (PCI-DSS). Section 6.3.7 of the PCI-DSS requires custom application code to be reviewed for potential vulnerability. For Internet-facing Web applications, a Source Code Review is one option to satisfy Section 6.6. For payment applications sold by software vendors, Section 5.1.7 of the Payment Application Data Security Standard (PA-DSS) states that a vendor's payment application code must be reviewed prior to release to customers. The PCI-DSS also states that Web applications must be developed according to guidelines such as the Open Web Application Security Project (OWASP).

One of the chief benefits of a Source Code Review is the ability to have 100% visualization of an application. Run-time assessments or black-box penetration testing can certainly be valuable to help identify vulnerabilities, but there is often some question as to whether or not all aspects of the application were tested. For example, different configuration settings could expose or hide entire modules of the application. What configuration was used during the testing? Or, certain user roles with different levels of access might not have been used by the testers. A review of the code can theoretically "see" all of the different configuration options and user roles.

A code review can identify whether an application is susceptible to many different types of attack, including:

- SQL Injection – does the code use safe constructs like parameterized queries and prepared statements? Or, is string concatenation incorporating user-controllable data used to build SQL statements?
- Cross-Site Scripting (XSS) – does the code output untrusted user-supplied data to HTML pages without proper encoding (e.g., HTML entity references or JavaScript escapes).
- Cross-Site Request Forgery (CSRF) – does the application validate a unique token with each HTTP request that causes an action or a change to occur? Or, is the request trusted simply by the presence of a valid session cookie?
- Parameter Tampering – does the code validate input data and check user permissions to prevent a user from elevating privilege or gaining access to unauthorized information?

- Forced Browsing – are specific controls in place to prevent users from accessing sensitive resources or bypassing required application logic and/or steps?
- As mentioned previously, some benefits are unique to Source Code Review. These include:
- The ability to identify backdoors left by a malicious or well-meaning developer. A Web application may have a backdoor that surreptitiously opens up administrative functions when a cryptic parameter like "N8K9TaV42mq" is present in an HTTP request. Black-box testing (even with automated "fuzzing" tools) won't find that type of backdoor.
- The ability to identify race conditions among multiple threads. Coding flaws involving race conditions are notoriously difficult to find (and reproduce) with run-time testing methods. This is because the coding bug surfaces only if the timing is right or when a very specific series of events occurs.
- The ability to identify improper error handling that could leak technical information useful to an attacker. A code review can determine if error/exception handling is missing, and find places where inappropriate debug information is output to users. A run-time analysis may not test in such a way that produces the error or debug information.
- The ability to identify sensitive files that are Web-accessible. An application may use or create certain files while it is running and interacting with users. These might include configuration files, temporary/work files, PDF statements, uploaded user files, or server pages containing source code. If these files are located in a Web-accessible directory, then remote, unauthenticated users may be able to download them.

As an example of the last item, in a previous Source Code Review engagement FishNet Security identified a vulnerability that allowed anyone on the Internet to view part of the source code for our client's application. This security hole was found during a review of the application's JSP code. Certain JSPs loaded JSP fragment, aka ".jspxf" files. JSP fragment files were a convention promoted by Sun Microsystems (the inventor of Java) years ago, and the recommendation was to use a ".jspxf" extension when the file is loaded directly into other JSPs (in other words, not a stand-alone JSP). The developers of this application had chosen to adopt this convention, but failed to locate the ".jspxf" files under the WEB-INF directory, which by default is inaccessible to remote users. The fragment files were located in the same directory as the regular JSPs, and the web server returned them without processing the JSP code since they weren't recognized as being JSPs. FishNet Security confirmed this leakage of source code by run-time testing in the live, production environment. The ".jspxf" files were downloadable by anyone on the Internet. The only barrier to their discovery and misuse was knowledge of the correct file names.

FishNet Security Advantages

At FishNet Security, many of our application security consultants have software development backgrounds. That experience, combined with our expert knowledge of application vulnerabilities and attack techniques, enables us to identify security flaws in your application source code. We have the ability to analyze code written in almost any programming language and Web platform or technology, including Java/JEE, JSF, Microsoft ASP.NET, classic ASP, PHP, ColdFusion, and AJAX. Additionally, we use best-of-breed, commercial static analysis tools in our Source Code Review practice. We also employ and continually monitor the progress of leading free/open source software (FOSS) tools that are available in the public domain.

References

1. Gartner Research Publication, "Application Developers Should Assume Responsibility for Application Security", ID #G00143844, November 16, 2006.

ABOUT FISHNET SECURITY

We Focus on the Threat so You can Focus on the Opportunity.

Committed to security excellence, FishNet Security is the #1 provider of information security solutions that combine technology, services, support, and training. FishNet Security solutions have enabled 3,000 clients to better manage risk, meet compliance requirements, and reduce cost while maximizing security effectiveness and operational efficiency.

For more information on FishNet Security, Inc., visit www.fishnetsecurity.com.