Security Code Review Advantages over Black-Box/Grey-Box Application Security Assessment

Kiran Maraju, CISSP, CEH, ITIL, SCJP

Abstract:
Typically, it is a belief that web application security assessment by black-box/grey-box testing technique is sufficient enough to identify vulnerabilities within the web applications. But, there is a possibility of missing out some critical vulnerability during these types of Application Security assessments. In addition to the black-box/grey-box testing technique security code review is an important piece of security testing by which it is possible to identify weakness in the source code as well as in the application. This paper gives an insight on the benefits of security code review than black-box/grey-box security assessment and some instances detailing the importance of security source code review.

Introduction:
Importance of Application Security growing rapidly with the increase of business needs of online transactions. Protecting online applications from hacking attempts is the prime concern to run a successful online business. Failing to protect web applications from malicious attempts leads to financial loss, legal complications and reputation loss. There are several legal requirements like PCI-DSS, Sarbanes-Oxley, GLBA and HIPAA enforced on the online business to help to protect company's customer's sensitive data from data theft.

Web Application Security:
Web Application security assessment involves identification web application vulnerabilities to protect the application from malicious hacking attempts. It is used as a baseline to implement proper security control of the online web applications.
OWASP (open web application security project) and WASC (Web Application Security Consortium) are the widely known application security initiatives involved in developing web application security awareness to the public and private held companies running business on web transactions.

Application Security Assessment Techniques:
Black-box Testing: Testing the application without the knowledge on the application. This testing process involves simulating the attack as a normal user without having access to the source code.
Grey-Box Testing: Testing the application with limited knowledge on the application. This testing process involves simulating the attack with the use of user credentials or limited access to the application.
Security Code Review: Testing will be done with the availability of the source code of the application. Testing process involves automatic and/or manual source code analysis to identify the vulnerabilities of the application source code.

Security code analysis is used to find security vulnerabilities in the source code of the application. It involves automated/manual assessment on the entire source code of the application. Automated assessment involves execution of code analysis tools. Automated code analysis requires validation of the results and elimination of false positives.

Manual assessment can be performed on the entire application source code or selection of critical source files that are prone to vulnerable attacks. Selected Manual assessment involves identification of vulnerable components like input validation, calls to native interfaces, session implementation, unreleased resources, and application configuration files identified from the design/architecture diagrams. Vulnerable source files are selected and manual analysis is performed on the selected files

Code Review used to identify the vulnerabilities of the following areas where black-box/ grey-box testing techniques fail to identify.

- Insecure coding practices
- Cryptography issues
- Improper configuration issues
- Improper exception handling
- Insecure error handling
- Access Control issues
- Application logic issues

Insecure coding practices:
Insecure coding practices in applications, which can lead to critical loss to the companies, will not be identified in Black-box/Grey-box web application assessment as there will be no availability of source code during testing process.

Coding level issues like class, method, variable access scope
Code level issues related to the scope of the class, method, and variable with public access provides a potential entry point for an attacker. Every class, method and variable scope should be limited to private access

Improper Inner class usage
JVM can't treat an inner class any differently from a regular class. An inner class with private member will be treated as package scope, and the inner classes defined as private are accessible to any class in the package.

Obsolete API usage
Application Developers uses obsolete or deprecate APIs which can lead to security issues are not identified in black box application security assessment. The obsolete API may satisfy functional requirements but can causes security issues.

Serialization Issues
Serialization is used to understand the internal state of your objects. This allows reading internal state of objects and sensitive information in the serialized objects declared as private. These kinds of issues can be easily identified in security code review process.

Synchronization/ Thread Issues
Synchronization and thread usage in source code leads to race conditions and deadlocks especially in static methods and constructors. Code review process identifies these types of loopholes at the source level.

Native Interface calls/ Unmanaged code usage
During code review process Native Interface calls/ unmanaged code usage calls can be identified. Native interface calls using JNI API to communicate c/c++ code can leads to command injection attacks can be identified at source code level.

Cryptography issues
Code review identifies vulnerabilities related to weak implementation cryptographic algorism used in the application code. Weak cryptographic algorithms like DES where key length is too small, broken algorithms like MD4 and MD5 etc. usage will be figured out during code review process.

Improper configuration issues
Sensitive information like passwords and database connections strings are hard coded in plain text in the application specific configuration web.xml/web.config file, source code leads to critical security issues in the application.

Improper exception handling
Improper implementation of exception handling brings security issues in the application. Code review process identifies improper exception handling issues that miss out in black-box/grey-box application security assessment process.

Null pointer exceptions
Root-cause identification of null pointer exception is not possible without the availability of source code. Source code review process explicitly checks whether an operation will return null before executing it. A null-pointer dereference takes place when a pointer with a value of NULL is used.

Unreleased resources leading to DOS
Unreleased resources lead to DOS due to memory consumption of the resources. Unreleased database resources, I/O resources, sockets should be closed in finally block. The reason for using the finally block is, any unreleased resources can be released and the memory can be freed. Security code review process identifies these types of loopholes in the source code.

Insecure error handling
Insecure error handling techniques like error messages directly reflecting back to the end user using printstacktrace or system.out and without implementing proper logging mechanism (log4j) etc. leads to information disclosure to end user. Using code review process either by automatic assessment with tools or with manual assessment such types of vulnerabilities can be identified.

Access Control issues
Thorough and Manual verification of declarative (web.xml) or programmatic access control implementation can prevent privilege escalation attacks like access of unauthorized modules of the applications. Declarative access control can be configured to restrict access to resources via the deployment descriptor (web.xml). During manual code review it is possible to identify proper implementation of programmatic access control that is access provided with the help of API like 'isUserInRole' etc based on the roles.

Application logic issues

Application Business Logic issues like guessable random number usage in the algorithms and sequence number generation of tokens are difficult to figure out from black-box or grey-box application assessment techniques. Thorough Manual code review process can identify such types of issues to protect the web application

Conclusion:

Organizations doing online business using web applications should perform security code review assessment in addition to black-box or grey-box security assessment testing to identify vulnerabilities in their public facing applications to protect from the hacking attacks and data theft. Automated and manual security code review process identifies the vulnerabilities at the source level so that there is no possibility of leaving the vulnerabilities unidentified.

References:

1) Twelve rules for developing more secure Java code
http://www.javaworld.com/javaworld/jw-12-1998/jw-12-securityrules.html

2) How To: Perform a Security Code Review for Managed Code
http://msdn2.microsoft.com/en-us/library/ms998364.aspx

3) Open Web Application Security Project
http://www.owasp.org/index.php/Main_Page

4) We Application Security Consortium
http://www.webappsec.org/