Application Security

Walter S. Kobus, Jr.
Vice President, Security Consulting Services, Total Enterprise Security Solutions, LLC, Raleigh, North Carolina, U.S.A.

Abstract
The primary goal of application security is that it will operate with what the senior management has decided is a reasonable risk to the organization’s goals and its strategic business plans. Second, it will ensure that the application, once placed on the targeted platforms, is secure.

Application security is broken down into three parts: (1) the application in development; (2) the application in production; and (3) the commercial off-the-shelf software (COTS) application that is introduced into production. Each one requires a different approach to secure the application. As with the Common Criteria ISO 15408, one must develop a security profile or baseline of security requirements and level of reasonability of risk.

APPLICATION SECURITY IN THE DEVELOPMENT LIFE CYCLE

In an ideal world, information security starts when senior management is approached to fund the development of a new application. A well-designed application would include at least one document devoted to the application’s security posture and plan for managing risks. This is normally referred to as a security plan[1] However, many application development departments have worried little about application security until the latter-day advent of Web applications addressing E-commerce. Rather than a firewall guarding the network against a threat, poor coding of Web applications has now caused a new threat to surface: the ability of hacking at the browser level using a Secure Socket Layer encrypted path to get access to a Web application and, finally, into the internal databases that support the core business. This threat has required many development firms to start a certification and accreditation program or at least address security requirements during the development life cycle.

SECURITY REQUIREMENTS AND CONTROLS

Requirements that need to be addressed in the development cycle are sometimes difficult to keep focused on during all phases. One must remember that the security requirements are, in fact, broken down into two components: (1) security requirements that need to be in place to protect the application during the development life cycle and (2) the security requirements that will follow the application into the targeted platform in the production environment.

SECURITY CONTROLS IN THE DEVELOPMENT LIFE CYCLE

Security controls in the development life cycle are often confused with the security controls in the production environment. One must remember that they are two separate issues, each with its own security requirements and controls. The following discussion represents some of the more important security application requirements on controls in the development life cycle.

Separation of Duties

There must be a clear separation of duties to prevent important project management controls from being overlooked. For example, in the production environment, developers must not modify production code without going through a change management process. In the development environment, code changes must also follow a development change management process. This becomes especially important when a code is written that is highly sensitive, such as a cryptographic module or a calculation routine in a financial application. Therefore, developers must not perform quality assurance on their own code and must have peer or independent code reviews.

Responsibilities and privileges should be allocated in such a way that it prevents an individual or a small group of collaborating individuals from inappropriately controlling multiple key aspects of any process or causing unacceptable harm or loss. Segregation is used to preserve the integrity, availability, and confidentiality of...
information assets by minimizing opportunities for security incidents, outages, and personnel problems. The risk is when individuals are assigned duties in which they are expected to verify their own work or approve work that accomplishes their goals; hence, the potential to bias the outcome. Separation of duties should be a concern throughout all phases of the development life cycle to ensure there is no conflict of duties or interests. This security requirement should start at the beginning of the development life cycle in the planning phase. The standard security requirements should be that no individual is assigned a position or responsibility that might result in a conflict of interest to the development of the application. There are several integrated development tools available that help development teams improve their productivity, version control, maintain a separation of duties within and between development phases, create quality software, and provide overall software configuration management through the system’s life cycle.

**Reporting Security Incidents**

During the design, development, and testing of a new application, security incidents may occur. These incidents may result from people granted improper access or successful intrusion into both the software and hardware of a test environment and stealing a new code. All security incidents must be tracked and corrective action taken prior to the system being placed into production. The failure to document, assess, and take corrective action on security incidents that arise during the development cycle could lead to the deployment of an application containing serious security exposures. Included are potential damage to the system or information contained within it and a violation of privacy rights.

These types of incidents need to be evaluated for the possible loss of confidentiality, loss of integrity, denial of service, and the risk they present to the business goals in terms of customer trust.

Security incidents can occur at any time during the development life cycle. It is important to inform all development project team members of this potential in the planning phase.

**Security Awareness**

Security awareness training must be required for all team members working on the development project. If a particular team member does not understand the need for the security controls and the measures implemented, there is a risk that the particular member will circumvent or bypass these controls and weaken the security of the application. In short, inadequate security awareness training may translate into inadequate protection mechanisms within the application. The initial security briefing should be conducted during the planning phase, with additional security awareness, as appropriate, throughout the development life cycle. A standard for compliance with the security requirement is to review the security awareness training program to ensure that all project team members are aware of the security policies that apply to the development of the project.

**Access**

For each application developed, an evaluation must be made to determine who should be granted access to the application or system. A properly completed access form needs to be filled out by the development manager for each member who needs access to the development system and development software package. User identification and an audit trail are essential for adequate accountability during the development life cycle. If this security requirement has not been satisfied, there is a possibility that unauthorized individuals may access the test system and data, thereby learning about the application design. This is of special concern in applications that are sensitive and critical to the business operations of the organization. Access decisions for team personnel should be made at the assignment stage of the development project and no later than the planning stage of the development life cycle.

**Determination of Sensitivity and Criticality**

For every application that is going to be placed into the development and production environments, there must be a determination regarding the sensitivity of the information that will reside on that system and its criticality to the business. A formal letter of determination of sensitivity and criticality is required. This should be done prior to the approval stage of the application by the senior management because it will impact resources and money. The letter of determination of sensitivity is based on an analysis of the information processed. This determination should be made prior to any development work on the project and coordinated with the privacy officer or general counsel. The letter of criticality is used to evaluate the criticality of the application and its priority to the business operation. This document should be coordinated with the disaster and contingency officer. Both documents should be distributed to the appropriate information technology (IT) managers (operations, network, development, and security).

Applications that are sensitive and critical require more care and, consequently, have more security requirements than a nonsensitive or noncritical system. The improper classification of information or criticality in an “undetermined state” could result in the users not
properly safeguarding information, inadequate security controls implemented, and inadequate protection and recovery mechanisms designed into the application or the targeted platform system.

**Labeling Sensitive Information**

All sensitive documentation must be properly labeled to inform others of their sensitive nature. Each screen display, report, or document containing sensitive information must have an appropriate label, such as *Sensitive Information* or *Confidential Information*. If labeling is incorrect or has not been performed, there is a risk that sensitive information will be read by those without a need to know when the application moves into production. Labeling should begin at the time that reports, screens, and so on are coded and continue through the system life cycle.

**Use of Production Data**

If production data is used for developing or testing an application, a letter specifying how the data will be safeguarded is required; and permission is needed from the owner of the data, operations manager, and security. Sensitive production data should not be used to test an application. If, however, production data must be used, it should be modified to remove traceability and protect individual privacy. It may be necessary to use encryption or hash techniques to protect the data. When the development effort is complete, it is important to scrub the hardware and properly dispose off the production data to minimize security risk. The risk of using production data in a development and test environment is that there might be privacy violations that result in a loss of customer and employee trust or violation of law. Development personnel should not have access to sensitive information.

**Code Reviews**

The security purpose of the application code review is to deter threats under any circumstance; events with the potential to cause harm to the organization through the disclosure, modification, or destruction of information; or by the denial of critical services. Typical threats in an Internet environment include:

- **Component failure.** Failure due to design flaws or hardware/software faults can lead to denial of service or security compromises through the malfunction of a system component. Downtimes of a firewall or false rejections by authorization servers are examples of failures that affect security.

- **Information browsing.** Unauthorized viewing of sensitive information by intruders or legitimate users may occur through a variety of mechanisms.

- **Misuse.** The use of information assets for other than authorized purposes can result in denial of service, increased cost, or damage to reputations. Internal or external users can initiate misuse.

- **Unauthorized deletion, modification, or disclosure of information.** Intentional damage to information assets that result in the loss of integrity or confidentiality of business functions and information.

- **Penetration.** Attacks by unauthorized persons or systems that may result in denial of service or significant increases in incident handling costs.

- **Misrepresentation.** Attempts to masquerade as a legitimate user to steal services or information, or to initiate transactions that result in financial loss or embarrassment to the organization.

An independent review of the application code and application documentation is an attempt to find defects or errors and to assure that the application is coded in a language that has been approved for company development. The reviewer shall assure that the implementation of the application faithfully represents the design. The data owner, in consultation with information security, can then determine whether the risks identified are acceptable or require remediation. Application code reviews are further divided into peer code reviews and independent code reviews, as follows:

- **Peer code reviews** shall be conducted on all applications developed whether the application is nonsensitive, sensitive, or is defined as a major application. Peer reviews are defined as reviews by a second party and are sometimes referred to as *walk-throughs*. Peer code review shall be incorporated as part of the development life cycle process and shall be conducted at appropriate intervals during the development life cycle process.

- **The primary purpose** of an independent code review is to identify and correct potential software code problems that might affect the integrity, confidentiality, or availability once the application has been placed into production. The review is intended to provide the company a level of assurance that the application has been designed and constructed in such a way that it will operate as a secure computing environment and maintain employee and public trust. The independent third-party code review is initiated upon the completion of the application source code and program documentation. This is to ensure that adequate documentation and source code shall be available for the independent code review. Independent code reviews shall be done under the following guidelines:
Table 1  Confirmation that the application’s information, integrity, and availability are assured

As the development project manager of XYZ application, I will need the following number of (NT or UNIX) servers. These servers need to be configured to store and process confidential information and ensure the integrity and the availability of XYZ application. To satisfy the security of the application, I need assurance that these servers will have a minimum security configured as follows:
Password standards
Access standards
Backup and disaster plan
Approved banner log-on server
Surge and power protection for all servers
Latest patches installed
Appropriate shutdown and restart procedures are in place
Appropriate level of auditing is turned on
Appropriate virus protection
Appropriate vendor licenses/copyrights
Physical security of servers
Implementation of system timeout
Object reuse controls

Please indicate whether each security control is in compliance by indicating a “Yes” or “No.” If any of the security controls above is not in compliance, please comment as to when the risk will be mitigated. Your prompt reply would be appreciated not later than (date)

○ Independent third-party code reviews should be conducted for all Web applications, whether they are classified sensitive or nonsensitive, that are designed for external access (such as E-commerce customers, business partners, etc.). This

Table 2  Request for security

As the development project manager of XYZ application, I will need the assurance that the production network environment is configured to process confidential information and ensure the integrity and the availability of XYZ application to satisfy the security of the application. The network should have the following minimum security:
Inbound/outbound ports
Access control language
Password standards
Latest patches
Firewall
Configuration
Inbound/outbound services
Architecture provides security protection and avoids single point of failure

Please indicate whether each security control is in compliance by indicating a “Yes” or “No.” If any of the security controls above is not in compliance please comment as to when the risk will be mitigated. Your prompt reply would be appreciated not later than (date)


APPLICATION SECURITY IN PRODUCTION

When an application completes the development life cycle and is ready to move to the targeted production platform, a whole new set of security requirements must be considered. Many of the security requirements require the development manager to coordinate with other IT functions to ensure that the application will be placed into a secure production environment. Table 1 shows an example representing an E-mail message addressed to the group maintaining processing hardware to confirm that the application’s information, integrity, and availability are assured.

A similar E-mail message could also be sent to the network function requesting the items in Table 2.

COMMERCIAL OFF-THE-SHELF SOFTWARE APPLICATION SECURITY

It would be great if all vendors practiced application security and provided their clients with a report of the security requirements and controls that were used and validated. Unfortunately, that is far from the case, except when dealing with cryptographic modules. Every time an organization buys an off-the-shelf software application, it takes a risk—risk that the code contains major flaws that could cause a loss in revenue, customer, employee privacy information, and so on. This is why it is so important to think of protecting applications using the defense-in-depth methodology. With a tiny hole in Web application code, a hacker can reach right through from the browser to an E-commerce Web site. This is referred to as Web perversion, and hackers with a little determination can steal digital property, sensitive client information, trade secrets, and goods and services. There are two COTS packages available on the market today to protect E-commerce sites from such attacks. One software program on the market stops application-level attacks by identifying legitimate requests, and another software program automates the manual tasks of auditing Web applications.
OUTSOURCED DEVELOPMENT SERVICES

Outsourced development services should be treated no differently than in-house development. Both should adhere to a strict set of security application requirements. In the case of the outsourced development effort, it will be up to technical contract representatives to ensure that all security requirements are addressed and covered during an independent code review. This should be spelled out in the requirements section of the Request for Proposal. Failure to pass an independent code review then requires a second review, which should be paid for by the contractor as a penalty.

SUMMARY

The three basic areas of applications security—development, production, and commercial off-the-shelf software—are present in all organizations. Some organizations will address application security in all three areas, while others only in one or two areas. Whether an organization develops applications for internal use, for clients as a service company, or for commercial sale, the necessity of practice plays a major role in the area of trust and repeated business. In today’s world, organizations are faced with new and old laws that demand assurance that the software was developed with appropriate security requirements and controls. Until now, the majority of developers, pressured by senior management or by marketing concerns, have pushed to get products into production without any guidance of or concern for security requirements or controls. Security now plays a major role in the bottom line of E-commerce and critical infrastructure organizations. In some cases, it can be the leading factor as to whether a company can recover from a cyber-security attack. Represented as a major component in the protection of our critical infrastructure from cyber-security attacks, application security can no longer be an afterthought. Many companies have perceived application security as an afterthought, pushing it aside in order to get a product to market. Security issues were then taken care of through patches and version upgrades. This method rarely worked well, and in the end led to a lack of customer trust, reflecting negatively on the integrity of the development company. The practice of application security as an up-front design consideration can be a marketing advantage to a company. This can be marketed as an added feature so that, when the application is installed on an appropriately secure platform, it will enhance the customer’s enterprise security program—not help to compromise it.

REFERENCES