Aligning Application Security and Compliance

A Security Innovation Whitepaper

SecurityInnovation
THE SOFTWARE SECURITY COMPANY

187 Ballardvale Street, Wilmington, MA 01887 +1.978.694.1008 **www.securityinnovation.com**

Table of Contents

Application Security: The Next Frontier of Compliance

APPLICATION SECURITY IS MOVING TO THE FOREFRONT

Enterprises have gone to great lengths to improve information security and document compliance with regulations and industry standards such as Sarbanes-Oxley, PCI DSS, HIPAA, FFIEC, FISMA and ISO 2700x.

But one critical area is only now coming into focus: application security.

Regulators, auditors and enterprises are increasingly emphasizing application security because of the growing recognition that *applications are the biggest source of data breaches*. Citing data compiled from the U.S. National Vulnerability Database, a recent Microsoft Security Intelligence Report found that application vulnerabilities were reported about *four times* as often as browser vulnerabilities and operating system vulnerabilities combined. Analyst firm Gartner stated: "Over 70% of security vulnerabilities exist at the application layer, not the network layer." Other researchers have estimated this figure at 90%.[i]

As a result, regulators and industry standards bodies are introducing requirements related to application development practices. For example:

- The PCI DSS standard calls for organizations to "develop applications based on secure coding guidelines" and prevent coding vulnerabilities such as injection flaws, buffer overflows, improper error handling and cross-site scripting.
- The latest version of PCI DSS (2.0) adds examples of secure coding standards that organizations should follow to be compliant.
- FISMA (Federal Information Security Act) and NIST (National Institute of Standards and Technology) guidelines require organizations to integrate security assessments into the software development life cycle, including code reviews, application scanning and regression testing.[ii]

WHY APPLICATION SECURITY IS DIFFICULT

Unfortunately, most executives, IT managers and compliance teams find it difficult to come to grips with application security for a number of reasons:

- Organizations usually have dozens or even hundreds of applications and systems that are exposed in some fashion to hackers, cybercriminals and potentially malicious "insiders" (business partners, contractors and employees).
- Enterprises don't understand how applications interact with their environment and exactly where these interactions can introduce risk.
- Regulations call for general security requirements like "developing according to industry best practices" when no authoritative source for industry best practices exists.
- Many regulatory requirements have non-obvious implications. For example, a requirement that protected information "should not be improperly altered or destroyed" implies that application developers need to provide extensive logging of transactions. This requirement is never stated explicitly, but an experienced auditor will recognize when inadequate logging renders an application non-compliant.
- Implementing application security is complex; it involves policies, processes and training the entire software development team.

HOW TO CONNECT APPLICATION SECURITY AND COMPLIANCE

While obstacles exist, reducing application-based risks and documenting the compliance of software applications can be easier than most organizations think — with the right knowledge and planning.

In this white paper we discuss how to:

1. Align software development processes with corporate policies.
2. Align software development activities with compliance requirements.
3. Create an action plan to identify and remediate gaps between current and best practices.

Aligning Software Development Processes with Corporate Policies

WHY INVOLVE MANAGEMENT?

Can't application security be left to the software development organization? Why involve management?

Software development groups typically concentrate primarily on delivering functionality and meeting schedules because they perceive that management priorities place these goals far ahead of application security and regulatory compliance.

To counteract this tendency, software development groups need guidance from management on topics such as:

- The importance management places on data security and compliance relative to other priorities.
- The direct impact software applications have on data security risks.
- The applicability and relative importance of the many federal, state and international regulations and industry standards.
- The business implications of not meeting compliance mandates.
- The potential impact on business of different types of data breaches and attacks on business systems.

In other words, enterprises need an explicit process for formulating corporate security and compliance standards and policies, and for translating these into terms specific enough for people to apply on their jobs. This process should reach not only the application development group, but also the IT operations teams that deploy applications and monitor them for vulnerabilities, and the risk and compliance teams that are on the hook for meeting compliance mandates.

THE CORPORATE APPLICATION COMPLIANCE FRAMEWORK

To align software development with management policies, enterprises need an approach such as the Corporate Application Compliance Framework, shown in **Figure 1**.

At the top level of the framework, the enterprise risk management (ERM), human resources and legal groups define the global scope, objectives and requirements for corporate governance, evaluating factors such as:

- Applicable legislation (Sarbanes-Oxley, HIPAA, California SB 1386), industry standards (ISO 2700x, FISMA/NIST standards), compliance mandates (PCI DSS), and legal and human resources requirements (data privacy laws).
- The potential impact of security breaches on customers, corporate reputation, regulatory bodies, and domestic & international governments.
- The costs of security breaches and attacks in terms of regulatory fines, customer notification, loss of revenue, interrupted operations, loss of business continuity and other expenses.

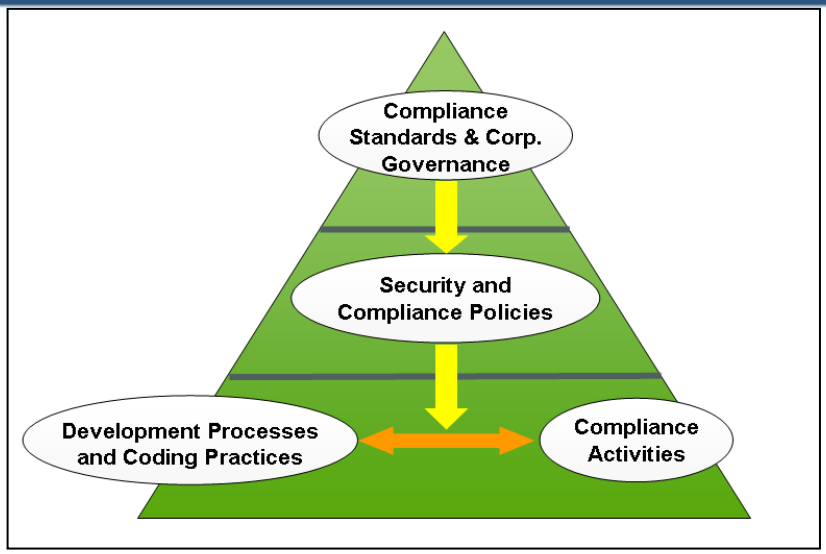

Figure 1: The Corporate Application Compliance Framework
Source: Security Innovation

These evaluations can be embodied in standards that provide guidelines and priorities for business units and form the basis of corporate compliance standards.

At the middle level of the process, an ERM group, together with representatives of the compliance and security teams, take the corporate security and compliance standards and add detail to create security policies for the company. These are high-level guidelines for operational security and compliance activities that can later be contextualized for specific business units and functional roles.

Typical tasks at this middle level include:

- Studying the applicable regulations and standards, to identify specific requirements that apply to the enterprise.
- Conducting a threat assessment to determine the security breaches potentially most damaging to the enterprise and the specific threats that might cause those breaches.
- Classifying data assets to define levels of data sensitivity and identify which business processes store and transmit sensitive data.
- Defining concrete application security objectives.

Ideally, the policies developed through these exercises will be specific enough to guide the operational teams, although in practice, reaching the right level of specificity can be very challenging.

At the base level of the process, the security and compliance teams help define and refine specific development processes, coding practices and procedures for documenting compliance, ensuring that they are relevant to local requirements and technology, as well as consistent with corporate security and compliance policies. These should address regional and business-unit-specific regulations and the technologies used by each development team.

Contextualizing the corporate security and compliance policies for each software development and assessment team can be a labor-intensive and deeply technical process, but the effort is justified. The more specific and practical the guidance, the more successful the team will be in complying with corporate governance requirements.

Aligning Software Development Activities with Compliance Requirements

SECURITY TRAINING

Security training is a critical prerequisite to a successful, and compliant, application security program.

While the appropriate type and amount of security training varies across organizations, in most cases:

- All members of the software development organizations and selected members of the compliance team should attend basic software security awareness training.
- Managers and architects should receive training on topics like threat modeling, architecture risk analysis and secure development life-cycle practices.
- Software engineers should study defensive coding for specific languages and environments such as C+, C#, J2EE and ASP.NET.
- Software engineers should learn how to operate tools such as code and web scanners. However, unless they understand how applications function and fail with respect to security, they will not get full utility out of these tools and spend countless hours weeding through false positives. Unless they understand the limits of the tools, they will be misled by a false sense of security.
- Software engineers, network/IT managers and quality assurance professionals should be trained on how software applications are attacked. They also need to learn how to code software to fail securely, and to handle the error conditions that attackers attempt to create in applications.
- Compliance professionals can benefit from courses on application security for specific standards like HIPAA and PCI DSS.

Compliance and Security Training

HIPAA requires that covered entities: Implement a security awareness and training program for all members of its workforce (including management).

PCI DSS requires that organizations: “Implement a formal awareness program to make all personnel aware of the importance of cardholder data security...[and] verify that personnel attend awareness training upon hire and at least annually.”

The Federal Financial Institutions Examination Council (FFIEC) states: “Financial institutions need to educate users regarding their security roles and responsibilities. Training should support security awareness and strengthen compliance with security policies, standards, and procedures.”

NIST SP 800-64 R2 suggests that: “System development training and orientation should include basic and specialized (to environment) security awareness, training, and education.”[iii]

CODING PRACTICES

Standardized coding practices are essential for any professional software development organization. They embody best practices, improve consistency, reduce errors and facilitate testing. Most important, they provide a benchmark against which compliance can be measured.

But "connecting the dots" between coding practices and compliance requirements can be extremely challenging.

First, a broad range of domain expertise is required. Several experts may need to pool their knowledge to triangulate between:

1. Applicable regulations and standards.
2. Probable threats and application-related vulnerabilities.
3. Application security techniques and coding practices.

Typically, this involves representatives from the compliance, software development and security teams. Often, outside experts are employed to catalyze the cross-training process and provide useful tools, mentoring and training.

Second, many high-level requirements imply that certain functionality be provided or specific practices be followed, without stating the details. To comply with a requirement for data integrity, customer-facing applications should include comprehensive input validation and error-handling routines. A general requirement for confidentiality implies (among other things) that the software development organization have an internal standard forbidding the use of custom or untrusted encryption routines. These details are not explicit in the original requirement, but experienced auditors will know that they are implicit.

Third, a number of implementation steps are required. Security coding practices and standards need to be documented, and software engineers, architects, database analysts and quality assurance engineers need to be trained in their use.

EXAMPLES OF APPLICATION SECURITY CODING PRACTICES

Coding practices and standards can run to hundreds of topics, and will vary based on factors like corporate priorities, application types, probable threats and technical environments. The Corporate Application Compliance Framework and the software development life cycle (SDLC) activities discussed later in this white paper will help narrow down and prioritize these topics.

But the coding practices necessary for compliance can still be voluminous. **Table 1** on the next page summarizes some of the more important, grouped by high-level requirements.

Table 1: Selected coding practices that contribute to compliance

High-Level Requirement	Standards (Partial List)	Selected Coding Practices
Confidentiality	SOX, PCI DSS, ISO 27002, HIPAA, GLBA, FFIEC, Basel II, CA SB 1386, FIPS 199, NIST SP 800-30/ 800-53/800-64	Appropriate use of strong encryption for data in databases. Encrypting confidential data in memory. Encrypting data in motion, especially for wireless transmissions. No custom or untrusted encryption routines. Masking confidential data that needs to be viewed in part (e.g., partial credit card or Social Security numbers).
Data integrity	SOX, PCI DSS, ISO 27002, HIPAA, GLBA, FIPS 199, NIST SP 800-30/ 800-53/800-64	Robust integrity checks to prevent tampering with data. Input validation and comprehensive error handling to prevent injection attacks, privilege escalation and other hacking techniques. Output encoding. Use of least privileges. Hashing for confidential data that needs to be validated (e.g., passwords).
Authentication and access control	SOX, PCI DSS, ISO 27002, HIPAA, Basel II, NIST SP 800-30/ 800-53/800-64	Support for strong passwords and two-factor authentication where appropriate. Role-based access control and revocation of rights, with clear roles mapped to permissions. Locked-down file access and database roles. No guest accounts. Passwords and keys encrypted before storage and transmission.
Logging and auditing	SOX, PCI DSS, ISO 27002, HIPAA, GLBA, CA SB 1386, NIST SP 800-30/ 800-53/800-64	Detailed audit trails of users accessing data and resources. Detailed logging of systems that process sensitive data, including shutdowns, restarts and unusual events. Event logs and audit trails available only to system administrators and protected from unauthorized modifications. No confidential data exposed in logs.
Availability	SOX, ISO 27002, HIPAA, Basel II, FIPS 199, NIST SP 800-30/ 800-53/800-64	Code reliability. Failover and redundancy built into applications. Applications resistant to denial-of-service attacks. Cleanup of confidential data in memory and in file systems during failures and shutdowns.
Change management	SOX, Basel II, NIST SP 800-53/ 800-64	Source control. Logging of application changes. Application change logs accessible only to authorized users and resistant to tampering.

The approach taken in assembling the table above is a useful tactic in mapping application security to compliance. Often, a spreadsheet is created with controls from the necessary compliance regulations and guidelines listed in rows and the application security activities necessary to meet each control documented in the columns. A simple refactoring that combines redundant activities will show how single application security changes can provide coverage for multiple regulatory and governance controls.

OWASP AND OTHER CODING STANDARDS

Several organizations publish documents and tools that can help enterprises develop and document their own secure coding practices. These include:

- The **Open Web Application Security Project** (OWASP) Top 10 listing of critical web application security threats and suggested preventative practices.

- The **Microsoft SDL** (Security Development Lifecycle).
- The **Common Weakness Enumeration** (CWE) list of most dangerous software weaknesses.
- The **CERT Secure Coding Standards**.
- Security Innovation's **TeamMentor™** secure development standards, an extensive collection of SSDLC practices and recommendations.[iv]

COMPLIANCE AND THE SDLC

Most enterprises have defined processes for software development. These help managers control functionality, quality and developer productivity. Unfortunately, they rarely place much emphasis on security or compliance activities.

Yet building security into the software development life cycle (SDLC) is critically important for both compliance and development productivity.

Regulations and industry standards are increasingly specifying requirements related to software development security best practices. For example, the PCI DSS standard specifies that compliant organizations should "incorporate information security throughout the software development life cycle." The Department of Defense and Defense Information Systems Agency (DISA) recently published the 114-page *Application Security and Development Security Technical Implementation Guide* with detailed recommendations for creating a secure SDLC.[iii]

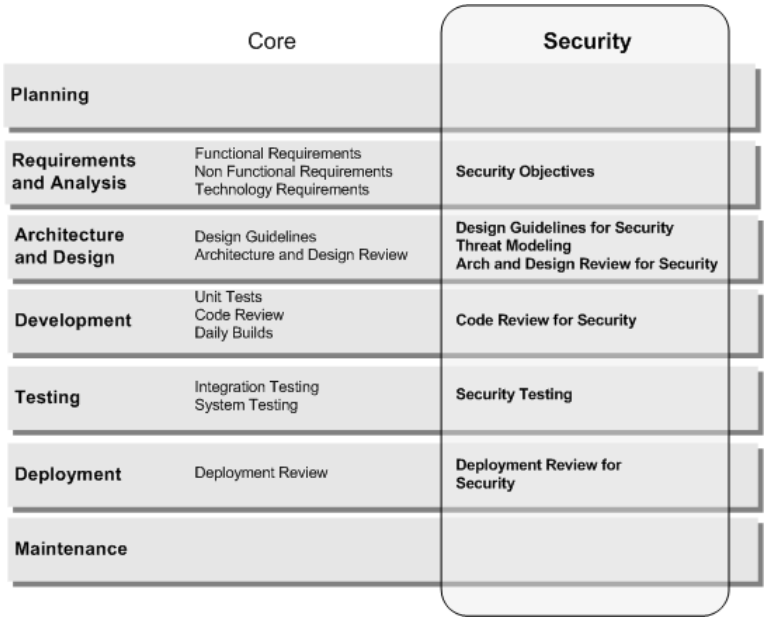

Figure 2: Integrating security into the software development life cycle (SDLC) *Source: Security Innovation*

There is also strong evidence that addressing security issues early can dramatically reduce software development costs and improve delivery schedules. Missing or inadequate security features are defects, and industry experts have determined that preventing defects in the design phase requires one-tenth the effort of catching and fixing those defects at the system test phase. Gartner estimates that removing 50 percent of software vulnerabilities prior to applications being put into production can reduce configuration management and incident response costs by 75 percent.[iv]

Figure 2 shows how security can be built into the SDLC. **Table 2** on the next page summarizes some activities that address compliance mandates.

Table 2: Addressing compliance requirements related to the SDLC

Activity	Examples of Requirements and Recommendations	Selected Best Practices
Security objectives and requirements analysis	*"Security planning should begin...by: identifying key security roles for the system development; identifying sources of security requirements, such as relevant laws, regulations, and standards; and ensuring all key stakeholders have a common understanding*." NIST SP 800-64	• Examine regulations and standards, potential exploits and attacks, and the business risks of those exploits and attacks. • Evaluate business requirements in terms of information confidentiality, integrity and availability. • Write "abuse cases" detailing how malicious users might interact with the system. • Define measurable goals for compliance and reducing vulnerabilities.
Threat modeling	*"[Covered entities must] conduct an accurate and thorough assessment of the potential risks and vulnerabilities to the confidentiality, integrity, and availability of electronic protected health information held by the covered entity*." HIPAA	Describe how adversaries might choose targets, locate entry points and conduct attacks. Identify what specific application defenses the attacker must defeat to be successful.
Architecture and design review for security	*"[Federal agencies should] employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems."* FIPS PUB 200	Identify design decisions that can mitigate risk; e.g., use languages such as Java and C# to reduce the risk from buffer overflow vulnerabilities, and validate all user input on servers to prevent attacks that manipulate variables on clients.
Code review for security	*"[Compliant organizations must conduct a] review of custom code prior to release to production or customers in order to identify any potential coding vulnerability...by knowledgeable internal personnel or third parties."* PCI DSS 2.0	Use automated code scanning tools and manual reviews to identify patterns such as non-constrained methods, unchecked return values, methods without exception handling, embedded passwords and sensitive information disclosed in logs.
Security testing	*"[Organizations must review] public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes [unless those applications are protected by a web-application firewall]."* PCI DSS 2.0	Rigorously test those application components that process confidential information, validate inputs, parse file formats and authenticate users. Check for sensitive information exposed in memory and in temporary files. Maintain a unit test library. Have developers cross-check each other's code. Conduct extensive penetration testing.
Deployment review for security	*"[Organizations must] develop configuration standards for all system components...Verify that system configuration standards are applied when new systems are configured... For a sample of system components, verify that all unnecessary functionality is removed."* PCI DSS 2.0	Develop checklists to ensure that web servers, databases, and storage and networking systems are properly configured and patched. Remove custom application accounts, user IDs and embedded passwords before applications are put into production.

Of course, in the real world not every best practice can be adopted, at least not all at once. Organizations need to identify which activities are most important to them based on corporate governance and priorities, then decide which to adopt first and in what sequence to introduce the rest.

For some enterprises, a "find and fix" approach using vulnerability scanners may suffice until their SDLC matures. Organizations with web-facing legacy applications may determine that the best use of their time is to plug a few gaping vulnerability holes. Companies with hundreds of small, internally facing applications might determine that code reviews are the best use of their time and skip penetration testing.

The bottom line is that an enterprise needs to understand the inherent risk level of the applications it is building and deploying. Applications need to be examined in terms of factors like applicable compliance requirements,

source (internal or outside third party), age, language (especially if written in an old or vulnerable language), environment in which they will be deployed, and the type and amount of sensitive information processed, stored and transmitted. The enterprise then needs to decide which activities in the software development life cycle will have the greatest impact on reducing risk and improving compliance.

Again, a simple spreadsheet can be very useful here. Each of the risk factors can be documented and weighted based on threats and the risks they pose to business operations and data protection. Threat modeling is an activity with a high return on investment, because it helps organizations chart and measure threats, translate them into tangible risks and document compensating controls that can be used to mitigate the threats.

Creating an Action Plan

So far we have discussed:

- How to align software development processes with corporate policies.
- How to align software development activities with compliance requirements.

But what is the best process for tackling these tasks?

Below is a four-phase process that has proved effective in helping enterprises improve application security and document compliance.

ASSESS EXISTING SOFTWARE PROCESSES AND PRACTICES

- Assess existing software development processes and practices in terms of security effectiveness and match with regulatory requirements.
- Review security policies and standards, organizational capabilities related to security, training, security attack response and patching processes.
- Examine the security-related components of the software development life cycle, such as threat modeling, security objectives and requirements analysis, abuse case modeling, architecture and design reviews, code reviews for security, static and penetration testing, and deployment reviews for security.
- Assess existing coding practices and standards related to objectives like confidentiality, privacy, data integrity and authentication.
- Evaluate current processes for documenting compliance with application security best practices.

IDENTIFY OBJECTIVES AND GAPS

The second phase of the process should center on goal setting. Investigate:

- The gaps between existing security practices and industry best practices.
- The technical and business risks associated with each security gap.
- The costs, benefits and expected value of addressing each gap.
- The extent to which current compliance activities successfully document application security processes and coding practices.

One output of this analysis can be a spreadsheet with compliance requirements on one axis and controls and actions on the other. This spreadsheet can be employed to identify overlap and opportunities to use a single control to satisfy multiple requirements.

The spreadsheet can also be used to highlight which current practices do not meet standards and which added security practices are desired but not yet in place in the SDLC. The sum of these two (practices needing improvement and practices desired but not yet in place) represents a comprehensive description of the activities the organization needs to adopt.

Then use the results of these investigations to create a set of security and security compliance goals.

PLAN A REMEDIATION ROADMAP

In the third phase of the process, use the assessments from the previous two phases to prioritize the identified threats and the areas most in need of improvement. For each high-priority area, review the major risk

management strategies (avoid, transfer, accept, remediate), identify appropriate control options and describe necessary modifications to compliance activities.

This process will show which activities and controls will produce the biggest “bang for the buck” by addressing the most important requirements or several requirements at once.

Use the results of this analysis to construct a phased software risk remediation and compliance roadmap.

IMPLEMENT THE REMEDIATION ROADMAP

In the final phase of the process, select tools and partners and implement the remediation roadmap. Tools and partners can be used to accelerate or outsource tasks like training, threat modeling, requirements analysis, code reviews, penetration testing and development of secure coding practices. For example, expert advice can pay large dividends in areas like selecting the sequence of activities, since the order in which new tools and procedures are introduced is often critical in ensuring the successful evolution of the application security management process.

Continue to measure progress relative to security and compliance objectives and requirements. Adjust the roadmap as corporate priorities, threat patterns and compliance standards change.

How Security Innovation Can Help

Security Innovation's solutions are built upon 15 years of application security experience and research. Organizations such as Barclay's, Symantec, ING, GoDaddy.com, MassMutual, Microsoft and HP depend on Security Innovation's solutions to ensure their software applications are secure and in compliance.

Improving application security and aligning it with compliance requirements involves a challenging set of activities. As detailed in this white paper, these range from defining standards and policies, to training large numbers of software engineers, testers and compliance professionals, to defining and documenting detailed coding practices, to adding new processes and compliance activities to the software development life cycle.

Security Innovation can provide expertise and solutions to accelerate all of these activities. Offerings include:

- **SDLC Compliance gap analysis and optimization service** to help IT and compliance managers map application security practices to compliance and corporate governance requirements. This service includes a complete analysis of people, processes and technology. It delivers a practical, specific roadmap to close gaps, and includes recommendations for security tools and training needed to achieve and maintain compliance.
- **TeamProfessor™ eLearning,** the industry's largest library of application security courses, covering each phase of the SDLC, all software application types, popular technologies like ASP.Net, Java, C/C++, .Net, Windows, C# and JRE, and industry standards/guidelines such as OWASP, PCI DSS, Microsoft SDL, HIPAA and NIST.
- **The TeamMentor™ Security Process Framework,** featuring an extensive collection of SDLC best practices and methodologies designed to ensure that development is aligned with corporate security policies and compliance requirements. TeamMentor includes dedicated, prescriptive guidance views for PCI DSS, OWASP and Security Engineering, and is an excellent complement to any security training initiative.
- **Compliance consulting services** to help managers evaluate current application security processes, assess staff training, create a Corporate Application Compliance Framework, and implement application security and compliance action plans.
- **Application assessment services** to perform threat modeling, code reviews and application penetration testing on a single application or across an entire portfolio of software applications.
- **High-performance encryption software** to ensure secure, efficient communications.

To Learn More:

For more information, please visit Security Innovation's web site at http://www.securityinnovation.com. To evaluate the company's eLearning products, please contact us at + 1.877.694.1008 x1 or getsecure@securityinnovation.com.

Additional Reading from Security Innovation

Application Security by Design: Security as a Complete Lifecycle Activity

Regulatory Compliance Demystified: An Introduction to Compliance for Developers

The Art of Threat Modeling for IT Risk Management

How to Conduct a Code Review: Effective Techniques for Uncovering Vulnerabilities in Your Code

19 Attacks for Breaking (all) Applications (Excerpts from the best-selling book *How to Break Software Security*)

Available at: http://www.securityinnovation.com/security-lab/whitepapers.html

Notes and Sources

[i] Microsoft Security Intelligence Report, Vol. 9, January through June 2010: http://www.microsoft.com/security/sir/keyfindings/default.aspx#section_3_1. Estimate from Gartner, quoted in *Computerworld*, February 25, 2005: http://www.computerworld.com/printthis/2005/0,4814,99981,00.html.

[ii] See https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf; https://www.pcisecuritystandards.org/pdfs/summary_of_changes_highlights.pdf; NIST Special Publication 800-53A Rev 1:Guide for Assessing the Security Controls in Federal Information Systems and Organizations: http://csrc.nist.gov/publications/nistpubs/800-53A-rev1/sp800-53A-rev1-final.pdf.

[iii] Department of Defense/DISA Application Security and Development Security Technical Implementation Guide, Vol. 3, Release 2: http://iase.disa.mil/stigs/downloads/zip/u_application_security_and_development_stig_v3r2_20101029.zip.

[iv] OWASP Top 10: http://www.owasp.org/index.php/OWASP_Top_Ten_Project
Microsoft SDL: http://www.microsoft.com/security/sdl/
CWE/SANS Top 25 Software Errors: http://www.sans.org/top25-software-errors/
CERT Secure Coding: http://www.cert.org/secure-coding/
Security Innovation TeamMentor™: http://www.securityinnovation.com/products/team-mentor/index.shtml.

[v] For estimates of the cost of finding defects at different stages of the development life cycle, see B. W. Boehm , P. N. Papaccio, Understanding and Controlling Software Costs, IEEE Transactions on Software Engineering, Vol. 14, No. 10, p.1462-1477, October 1988, or IDC and IBM Systems Sciences Institute, quoted in Microsoft Security Development Lifecycle: http://www.cert.uy/historico/pdf/MicrosoftSDL.pdf. The Gartner estimate can be found at: http://www.gartner.com/press_releases/asset_106327_11.html.