
SOFTWARE SECURITY ASSURANCE:

A FRAMEWORK FOR SOFTWARE VULNERABILITY MANAGEMENT AND AUDIT

CHARLES H. LE GRAND, CIA, CISA

EXECUTIVE OVERVIEW

Internet-facing systems represent significant opportunity as well as risk to any organization using them. They help meet customer and competitive needs, but they also provide a primary avenue for attackers to evade protective system barriers. Once an attack has exploited a vulnerability in a Web application, the application's server loses its reliability, subjects data to compromise or destruction, and can become a base for launching attacks against other systems within the organization's network or against other Internet systems.

This guide provides information needed to identify, measure, remediate, and manage specific security vulnerabilities in online systems. It identifies the source of the problem, recommends specific techniques to assess the extent and severity of the problem, and explains how the control environment can be structured to manage software security risks efficiently within the organization's risk appetite.

Software security is also a significant element of compliance with the laws, regulations, and policies that govern an organization and its data. Weak software security can represent, for example, a significant control deficiency in terms of compliance with the Sarbanes-Oxley Act; potentially compromising the reliability of financial information and reporting. The appendixes of this guide provide references to example laws and regulations related to information security, and cross-reference sources of guidance for assuring effective compliance practices.

Many positions within an organization have responsibilities for ensuring the security of online applications – from the programmer writing the source code all the way through the audit committee of the board that must assess the reliability of assurance regarding information reliability and security. As audit represents an essential element for controls assurance, this guide also provides guidance for audits of software security vulnerability management as well as an example audit program that can be modified to fit an organization's specific needs.

Many organizations and individuals participated in the global project team that helped develop and review this guide. We are grateful for their support and their professional commitment to relevance, accuracy, and the efficient delivery of information we believe the guide provides. We are also grateful to Ounce Labs for providing the sponsorship necessary to produce the guide.

As the author, I welcome questions, comments, or any input on the guide and its usability. I hope you will find the guide highly usable by the many people in your organization that have a role in providing software security assurance.

Charles H. Le Grand
CHL Global Associates

ABOUT THE AUTHOR:

Charles H. Le Grand, CIA, CISA, the founder of CHL Global Associates, has more than 30 years experience addressing the most critical technology issues facing the auditing profession. For many years he led the IIA headquarters staff in addressing IT issues and applying solutions.

Author of IIA's first seminars addressing auditing and personal computers, Introduced PCs to the IIA headquarters staff, Pioneered IIA's information systems auditing seminars and conferences programs, Directed the landmark 1991 and 1994 Systems Auditability and Control (SAC) research projects, Directed IIA's first research in Environmental Auditing and Improving Audit Committee Performance, As CIO, architected IIA's systems and network environment for global communications, Established IIA's first two Web sites: TheIIA.org, and ITAudit.org, Directed development of the "Information Security Management and Assurance" series, Co-authored the PC Management Guide sponsored by Intel, Pioneered the Global Technology Audit Guides (GTAG) series and co-wrote the first guide, Co-wrote the Information Security Program Elements for the Corporate Information Security Working Group (CISWG) supporting the Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, Government Reform Committee of the U.S. House of Representatives.

CHL Global Associates provides information security and reliability services in association with the best available technology management, security, control, risk management, auditing, assurance, and governance advisers and experts. To learn more, visit us at: www.CHLGlobalAssociates.com.

CHL GLOBAL ASSOCIATES
www.chlglobalassociates.com

ABOUT THE SPONSOR:

Ounce Labs™, the leader in software security assurance, delivers products that allow customers to verify that software meets their defined security requirements. Ounce Labs' enterprise-level source code vulnerability analysis provides reliable metrics necessary to manage software risk, enforce security policies, enhance audit capabilities, and track compliance efforts. Based on patents-pending Contextual Analysis technology, Ounce Labs' products also pinpoint specific software design errors and coding flaws to simplify remediation during any phase of the development lifecycle. Founded in 2002, Ounce Labs is located in Waltham, Massachusetts. For more information, please visit www.ouncelabs.com.



CONTENTS:

I. Managing Software Risk: An Executive Call to Action	1
A. Risky Internet Business.....	1
B. Software Security Assurance: Responsible Business Practices.....	1
C. The Executive's Role in Software Security Assurance	1
D. Call to Action: An Executive Checklist for Software Security Assurance:.....	2
1. Policies.....	2
2. Assessment and Monitoring.....	2
3. Assurance	2
4. Audits	3
F. Next Steps.....	3
G. Summary	3
II. Introduction	4
A. Statement of the Problem	4
B. Available Solutions.....	4
III. Audience for this Guide	5
IV. Purpose, Objectives, and Scope	5
V. The Issues and Consequences	6
A. Managing Risks from Vulnerable Software	6
B. Software Security Vulnerabilities.....	6
1. Sources of Source Code Security Vulnerabilities	7
2. Why Software Vulnerabilities Exist	7
3. Where Software Vulnerabilities Exist	8
VI. The Solution: A Framework for Software Security Assurance	9
A. What to Do	9
1. Risk Assessment	9
2. Vulnerability Management and Remediation	12
3. Security Standards for Development and Deployment	13
4. Ongoing Assessment and Assurance.....	14
B. Roles and Responsibilities.....	15
1. Board of Directors	16
2. Audit and Assurance.....	16
3. CEO	16
4. CFO	16
5. CIO.....	16
6. Executive(s) Responsible for Systems Development and Change Management.....	16
VII. Closing Summary	17
Appendixes	
Appendix A: Audit Program and Internal Control Questionnaire for Source Code Vulnerability Management.....	18
1. Scope.....	18
2. System Design, Development and Testing.....	19
3. Questions	20
Appendix B: Roles and Responsibilities for Software Security Assurance.....	21
1. The Importance of Software Risk Management.....	21
2. The Parties to Software Security Assurance	21
Appendix C: Control Objectives and Practices	29
1. Security, Reliability, and Compliance Frameworks.....	29
2. Software Security Assurance and Related Control Frameworks, Requirements, Standards and Guidance: COSO, Sox, COBIT, and ISO/IEC 17799.....	29
3. Key Issues	30

4. Assessing and Applying Compliance Guidance in Software Security Assurance	30
5. COSO	30
6. Sarbanes-Oxley	32
7. The COBIT Framework	35
8. ISO/IEC 17799.....	39
9. Cross-Reference Matrix for Private Sector Guidance Applicable to Software Security Assurance.....	41
10. Cross-Reference Matrix for Public Sector Guidance Applicable to Software Security Assurance.....	46
11. Software Security Assurance: A Management Compliance Checklist.....	48
Appendix D: Identifying Vulnerabilities in Web Applications: The Top Sources of Exposure to Locate and Remediate	52
1. Unvalidated Sources of Input.....	52
2. Use of Unvalidated Input.....	52
3. Unvalidated Output Streams.....	52
4. Flawed Authorization and Access Control	52
5. Flawed Authorization and Session Management.....	53
6. Native Code and Buffer Overflows.....	53
7. Dynamic Code	53
8. Weak Encryption	53
9. Application Configuration	53
10. Denial of Service.....	53
11. Network Communications	54
12. Unsupported Application Interfaces	54
13. Improper Administrative and Exception Handling	54
Appendix E: References	55
1. Bibliography and Web References	55
2. Organizations.....	58
Appendix F: Closely Related Issues to Consider.....	60
1. You Just Cannot Say it Enough...	60
a. Education	60
b. Passwords.....	60
c. Separation of Duties.....	60
d. Employee Terminations	60
2. CA SB 1386	60
3. Your Company Name Here.....	60
4. New York Times, Monday, May 9, 2005	61
5. Definition: Software Security Vulnerability	61
Acknowledgements: The Project Team.....	62
1. Advisory Council	62
2. Reviewers of the Draft Guide.....	63
3. Survey Responses.....	64

I. MANAGING SOFTWARE RISK: AN EXECUTIVE CALL TO ACTION

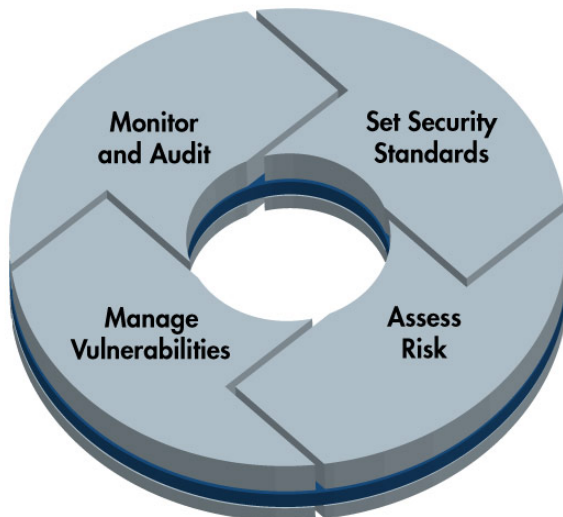
A. Risky Internet Business

The list of recent, high-profile security breaches is daunting; headlines have exposed major leaks among the country's largest organizations, resulting in loss of customer trust, potential fines and lawsuits. Vulnerable systems pose a serious risk to successful business operations, so managing that risk is therefore a necessary board-level and executive-level concern. Executives must ensure appropriate steps are being taken to audit and address IT flaws that may leave critical systems open to attack.

One of the greatest – but least understood – sources of IT risk lies within software applications. As the engines that power today's global enterprises, they process, calculate, transmit, and store the data that are an organization's primary asset. Gartner states that 70% of attacks come at the application layer, yet most critical software applications are never audited to identify vulnerabilities that may expose critical data and operations to hackers¹. Increasing consequences caused by regulations, targeted attacks and consumer awareness mandate an enterprise-wide approach for auditing, measuring, and addressing the risk to operations from vulnerable software.

B. Software Security Assurance: Responsible Business Practices

Elements for effective governance and management of software risk include:



- **Risk Assessment:** to determine the extent of vulnerabilities and estimate probability of losses from exploits
- **Vulnerability Management:** to identify and remediate specific security vulnerabilities
- **Security Standards for Development and Deployment:** to prevent the introduction of security vulnerabilities
- **Assessment and Assurance:** to provide ongoing auditing to monitor that risk levels remain within acceptable thresholds.

C. The Executive's Role in Software Security Assurance

Software security assurance is a broad management responsibility. Because vulnerabilities represent significant control deficiencies in terms of secure and reliable information, processes, and reporting, they fall within the direct purview of the CEO, CFO, and audit committee of the board. Vulnerabilities may also result in the disclosure of personal and other sensitive information, and therefore also impact the roles and responsibilities of management positions throughout the enterprise.

¹ Pescatore, John, Gartner, quoted in Computerworld, February 25, 2005, <http://www.computerworld.com/printthis/2005/0,4814,99981,00.html>

Two important elements for executives to consider and balance:

- **Assurance:** Software security assurance is driven primarily by the management processes that ensure effective controls. Secondary, independent assurance comes from auditors who perform control assessments and attest to management's assertions about the reliability of controls.
- **Cost/Value of Control:** The costs of software vulnerability management must be balanced against expectable losses from exploits of control weaknesses. It may be difficult to quantify expectable losses from vulnerability exploits, but the costs of controls must be balanced against values such as protection of customer information, business continuity and the organization's reputation.

D. Call to Action: An Executive Checklist for Software Security Assurance

Any enterprise-wide program for managing software risk requires executive-level sponsorship and leadership. The checklist below provides a guide for working with the management stakeholders across IT, audit, risk, development, and outsourced providers to outline and implement a comprehensive and effective assurance program. For more information, interested executives may consult the complete Software Security Assurance Framework, which outlines in detail the processes, stakeholders, and metrics required for an enterprise approach to software security assurance. It also provides audit guidance and control objectives aligned with the key regulatory regimes. The Framework is available online at the research sponsor's site: www.ouncelabs.com/audit.

Policies:

- ✓ Information security policies, procedures, and standards specifically address security vulnerabilities in Internet-facing applications.
- ✓ System development and maintenance processes and standards specifically provide for preventing the introduction of security vulnerabilities in new or changed systems and programs.
- ✓ Security standards for system design and program code apply equally to outsourced as well as internal design and programming.

Assessment and Monitoring:

- ✓ Intrusion protection systems specifically monitor attempts to attack Internet-facing applications.
- ✓ Risk management includes assessment of risks related to attacks against Internet-facing systems and cost/benefit evaluation of control effectiveness.
- ✓ Security vulnerabilities in software supporting Internet-facing applications are routinely measured and determined to be within the acceptable level of risk for such systems.
- ✓ Internet-facing applications are specifically assessed for their ability to enforce privacy requirements for personal and other sensitive information.

Assurance

- ✓ Responsibilities are communicated to management with specific roles in assuring software security vulnerabilities are efficiently controlled.
- ✓ Management provides metrics and other relevant information to the CFO, CEO, and audit committee of the board concerning the effectiveness of software security controls related to legal and regulatory compliance.

Audits:

- ✓ Compliance audits and other audits of information security specifically address management of security vulnerabilities in source code to include:
 - ✓ Measurement of vulnerabilities against prescribed standards for security and risk management.
 - ✓ Testing of software applications for the existence of security vulnerabilities.
 - ✓ Management of software security vulnerabilities in the system design, development, maintenance, and change management processes.
 - ✓ Management of software security in all outsourced systems and programming processes.

F. Next Steps

- If you only have the “*Executive Call to Action*” download and review the entire guide “*Software Security Assurance: A Framework for Software Vulnerability Management and Audit*” authored by Charles H. Le Grand, CIA, CISA, available at www.ouncelabs.com/audit.
- Share the guide with stakeholders on the information assurance team.
- Review the Executive Checklist and formulate your Software Security Assurance action plan.

*For a description of what kinds of questions to ask in a software security audit, reference the **Audit Checklist** in **Appendix A** on **page 23**.*

*For a description of pertinent **Roles and Responsibilities**, reference **Appendix B** on **page 26**.*

G. Summary

To maintain reliable operations, protect sensitive data, and comply with regulations, enterprises must institute a process for managing and auditing software risks. Executives should galvanize stakeholders from throughout the organization to identify the standards, processes, and technologies necessary to answer critical software security assurance questions across their software portfolio. The result will be a repeatable, consistent, and measurable process for addressing significant risk to corporate operations, data, and reputation.

II. INTRODUCTION

A. Statement of the Problem

Organizations today routinely connect mission-critical systems and data to the Internet and use the World Wide Web to create efficiencies and meet customer expectations and competitive demands. Building security into these systems is a challenge many organizations have not handled well – evidenced by the continuing rash of data theft, impersonation frauds (also called identity theft), and hacker intrusions into sensitive systems.

The very features that make Web browsers so convenient make Internet-facing systems² insecure. Internet-facing systems also have interfaces to legacy systems and databases that were never designed to consider Internet threats. As a result, hackers find it relatively easy to use Internet-facing business applications to penetrate enterprise systems and access sensitive and private information. New techniques to exploit systems and their users are prevalent in the daily news³ monitored by security professionals and hackers alike.

B. Available Solutions

The processes and techniques for security and control of information networks are reasonably mature and well documented. While networks and systems software are still subject to compromise, the means

Steps to Protect Against Internet Attacks

- *Identify and measure key vulnerabilities and threats*
- *Establish control objectives and norms*
- *Identify the key players and their roles*
- *Ensure effective tools and practices are in place*
- *Educate personnel concerning their role, the tools and practices in place, and the importance of their active participation in maintaining security*
- *Provide continuous assurance that controls are followed and remain effective*

for prevention, detection, recovery, monitoring, and minimization of harm can be made effective and reliable by any organization that makes a serious effort to implement security⁴. But vulnerabilities in business applications on the Internet continue to provide the best available avenues to compromise an organization's information and systems.

It is time for every organization to take decisive steps to protect against Internet attacks. Responsible entities must assess and manage the risks for Internet-facing systems. Effective information security and protection is not only good business practice, but in many cases it is a legal requirement. In recent years, legal and regulatory compliance requirements have

increased dramatically, and can be expected to continue increasing until security becomes the de facto standard for all electronic commerce and communications. The steps for instituting this change include:

- Identify and measure key vulnerabilities and threats
- Establish control objectives and norms

²Internet-facing systems are simply those systems that can be accessed via the Internet. The most familiar examples include email and web sites. Files can be transmitted using file transfer protocol (FTP), there is instant messaging (IM) and phone conversations can be held using voice over internet protocol (VOIP), but there are many more. Many organizations recognize the security weaknesses inherent in Internet contact, and take steps to isolate Internet-facing systems from other systems and data. But when the application calls for data from legacy systems or data warehouses, the ability to segregate starts to disappear. An exploit of a web site may provide the attacker access to other systems that serve data – possibly sensitive personal data – to the web application thus opening an avenue for compromise of data in systems that were never designed to compensate for Internet security requirements.

³New York Times, <http://www.nytimes.com/2005/05/10/technology/10cisco.html?pagewanted=1&th&emc=th>

⁴An excellent example of the maturity of practices for security networks and systems is found within the security benchmarks from the Center for Internet Security (www.CISecurity.org)

- Identify the key players and their roles
- Educate personnel concerning their roles, the tools and practices in place, and the importance of their active participation in maintaining security
- Ensure effective tools and practices in place provide continuous monitoring and assurance that controls remain effective.

The responsibility to provide secure information and systems must go beyond individual organizations and their stakeholders to become a universal and collective requirement, as indicated in a report presented in the White House in April 2000:

“In the modern world, everything business or government does with their information technology becomes part of the global information infrastructure. We must build infrastructure to a very high standard. Attaching weak components to the infrastructure puts your organization as well as your neighbors at risk. Responsible citizens will contribute only sound components to that cooperative infrastructure.”⁵

III. AUDIENCE FOR THIS GUIDE

This guide is for professionals in information management, systems development, information security, risk management, and auditing. It addresses the interfaces with executives, auditors, and governance as well as the roles and responsibilities of the CEO, CFO, CIO, and others in managing the risks and practices associated with software security vulnerabilities in Internet-facing systems. Government agencies, regulated industries, and publicly traded companies will appreciate the specific references to requirements to manage software security risks and provide appropriate assurance of effective controls and regulatory compliance.

Practitioners in systems management, security, auditing, risk management, consulting, and compliance will find a straight-forward presentation of the risk and management issues as well as guidance in assuring the presence and sustainability of controls that protect stakeholder interests and meet executive and organizational responsibilities.

IV. PURPOSE, OBJECTIVES, AND SCOPE

This guide is presented to help organizations implement and maintain a strong system to:

- Address the impacts of security vulnerabilities on risk management and monitoring
- Identify where software security vulnerability management fits within the system of internal controls
- Identify security vulnerabilities in source code and measure their extent and severity
- Mitigate and remediate existing security vulnerabilities
- Keep vulnerabilities out of new or changed software
- Provide reliable and sustainable monitoring and assurance that software security vulnerabilities remain within the organization’s specified risk appetite and tolerances, and
- Provide evidence of compliance with requirements.

⁵Information Security Management and Assurance: A Call to Action for Corporate Governance, by The Institute of Internal Auditors, National Association of Corporate Directors, American Institute of Certified Public Accountants, and Information Systems Audit and Control Association. See: http://www.theiia.org/?doc_id=3061#Books

V. THE ISSUES AND CONSEQUENCES

A. Managing Risks from Vulnerable Software

Software vulnerabilities provide the avenues that allow attackers to break through a system's protection to illegally access private information and system resources. Successful attacks can result in disclosure, corruption, or destruction of data or software and expose an organization to:

- Disruption of operations – impacting customers, employees, and business partners
- Loss of integrity in information and systems as attackers install unauthorized programs or program changes, make unauthorized use of computer or network services including Internet access, and corrupt or even destroy sensitive data
- Harm to reputation and consequent loss of trust, market value, and customer base
- Litigation, regulatory sanction, and personal liability for executives and directors

Efficient risk management addresses the likelihood of adverse events, their potential impacts, and effective allocation of resources to avoid them. Analysis of risks related to software vulnerabilities must identify the risk level, assess whether it is acceptable, and determine the measures needed to contain risks at an acceptable level.

Historical approaches to manage software security vulnerabilities are not adequate.

Measurement of software risks includes risks based on system uses, data access, and the type and extent of vulnerabilities in systems. Historically the level of risk from software vulnerabilities has not been measured. Instead, risk was

estimated through techniques such as system penetration testing and scanning of processing environments for the existence of unauthorized software or malicious code. That approach is not sufficient, and with the tools now available responsible management must take a more active approach to prevention of software and other security vulnerabilities.

B. Software Security Vulnerabilities

Software security is a crucial element of the information security management program for any organization. Software vulnerabilities enable external attacks and allow trusted insiders to exploit their access privileges to gain unauthorized access to information, systems, and services. Since insiders must have access to systems and data, an important control objective is to track their access and maintain records of their actions. But the more important point of this guide is to eliminate the vulnerabilities that allow inappropriate access from the outside or inside.

Source Code
When humans write programs, they write them in "source code" using a programming language like C, C++, Java and others. Source code is compiled into object code that can be installed and processed on a computer. Common errors in programming result in security vulnerabilities.

Security breaches result when an attacker exploits a flaw or feature in a program that causes the program to act in a manner for which it was not designed. Programs have normal interfaces with other programs, operating systems, databases, and system and network components that allow them to process transactions, exchange information, and deliver other services. A program can become corrupted when a vulnerability is exploited, and can take unexpected advantages of (abuse) the interfaces with other information, system, and network components resulting in undesirable consequences. Common examples of critical Web-application vulnerabilities are summarized in Appendix D.

1. Sources of Source Code Security Vulnerabilities

While much of an organization's software may be outside of its direct control for the management of vulnerabilities (i.e. purchased software maintained, upgraded, and patched by the vendor), this guide focuses on software developed and managed by and for the organization. This would include: software developed internally; outsourced and offshored software; open source software; and software acquired through mergers and acquisitions. For purchased software, an organization will provide a wide range of protective controls: from perimeter defenses to intrusion detection, patch management, keeping up with news of vulnerability discoveries, and more. These are addressed in other publications and are outside the scope of this guide as it focuses on software that can be controlled at the source code level.

Systems development has always presented significant management challenges, and today those challenges are increased by the need for strong security to defend against threats from the Internet. Secure coding standards and practices are now recognized as a necessary solution to the plague of online system exploits. Though an Internet search will reveal millions of references to the subject, there are no generally accepted "standards for secure programming." So each organization must establish and manage its own secure coding requirements.

The outsourcing of system and program development has also proven to be challenging as organizations can outsource the work, but not the liability for system vulnerabilities. Fortunately, the tools and techniques for managing software vulnerabilities have matured to the point that they offer strong capabilities for development of secure code and to ensure high-impact vulnerabilities do not exist in systems being implemented.

2. Why Software Vulnerabilities Exist

a) *We Put Them There*

Vulnerabilities exist in programs because the developers failed to prevent or detect them during the initial development cycle or in program upgrades or maintenance. Programmers may inadvertently introduce coding flaws that allow attacks such as buffer overflows or cross-site scripting, which can provide an attacker with unauthorized access. Or developers may fail to implement appropriate security mechanisms, such as encryption, and thereby allow sensitive information to be disclosed. Appendix D provides a summary of common types of vulnerabilities in source code.

Common reasons why vulnerabilities are introduced in source code include:

- Improper training of programmers
- Improper use of programming languages
- Inadequate security specifications or standards for program quality
- Inadequate review and testing of programs
- Improper use of software, and more.

These reasons are exacerbated by:

- Scarcity of programmers and their management skilled in security awareness
- Lack of generally accepted standards for secure program coding or for the stability of operating systems in which the programs operate
- Emphasis on speed rather than security during development resulting in ineffective change management or other project management practices or processes
- Decision to use the low cost source of programming without providing effective quality management or security requirements

- Or just plain ignorance of the need for security management during the development process.

Vulnerabilities may also be introduced intentionally by programmers during development or changes if they are not prevented by security management and quality practices and techniques. This may be motivated by programmers' dissatisfaction and desire to "punish" the organization or by programmers' intentions to exploit the vulnerability for profit. Vulnerabilities can also be introduced as a result of other exploits – as when a worm, virus, or hacker plants a Trojan horse or Zombie inside an existing system. Vulnerabilities will continue to exist in software as long as the environments in which software is developed, resides, operates, and is administered are unstable.

b) We Have Conflicting Objectives

Organizations have conflicting objectives in keeping information and systems secure while making information and services available for customers, potential customers, business partners, employees, and others via the Internet.

Secure systems provide no access. Opening a system to access creates vulnerabilities. The objective is to maintain the proper balance of security "and" (not "or") availability through risk management including access protection, minimizing vulnerabilities, monitoring known weaknesses and threats, and providing the structure and means for individual accountability.

Management of system development processes also involves conflicting objectives as the need for security and controls is weighed against the budget, schedule, functional requirements, and benefits of quick deployment. Only an irrefutable requirement that defined security levels be maintained can save an organization from such pressures. It is also important to note that when weighing the cost versus benefits for security and controls an organization should consider the cost of non-compliance as part of the overall analysis (e.g. US Federal Sentencing Guidelines).

3) Where Software Vulnerabilities Exist

Security vulnerabilities can exist in virtually any program accessible via the Internet or other networks. Web applications provide a popular avenue for delivering information and services, which makes them attractive targets for attack. These applications can contain vulnerabilities, that, unless identified by some reliable means, can remain undetected until an exploit is discovered and the damage has been done.

Many organizations neglect to monitor system activity at the Web application level, so intrusion attempts can easily go unnoticed. Since a carefully crafted exploit may leave little evidence, a significant lag may result between the exploit and its detection.

Newer programming languages and tools can provide improved security over older techniques. But many new systems continue to rely on older, or "legacy," systems to provide behind-the-scenes access to databases and program logic. Because these legacy systems and database management tools were not designed to contemplate threats from the Internet, they may be vulnerable to exploits relayed to them by the Internet-facing systems with which they interface.

VI. THE SOLUTION: A FRAMEWORK FOR SOFTWARE SECURITY ASSURANCE

Every organization involved in commerce or information exchange via the Internet must be accountable for secure systems and operations. Insecure systems put the organization and its stakeholders at risk. Insecure systems can harbor the means by which other systems are attacked. And the security, reliability, and privacy of sensitive information is mandated by legislation, regulations, and agreements between interactive parties. This increased recognition of the need for security, reliability, and protection from fraud and other threats has led to progressively more stringent compliance requirements.

A. What to Do

There are four main activities in software security assurance:

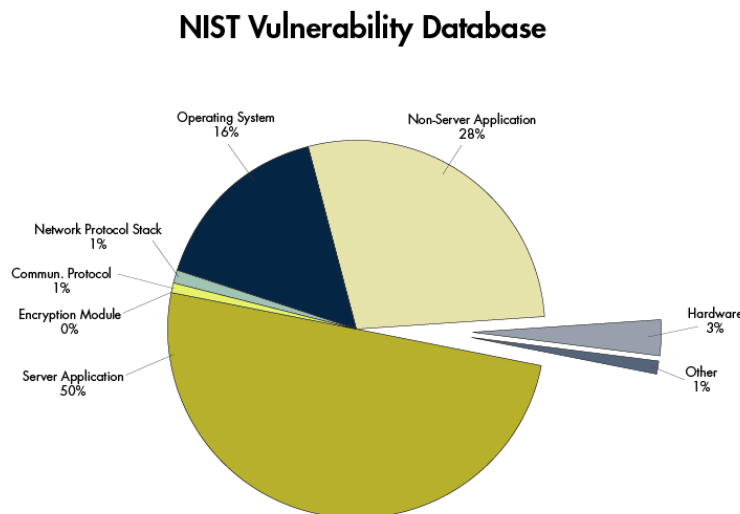
- **Perform Risk Assessment:** Determine the extent of vulnerabilities and their potential impacts
- **Provide Vulnerability Management and Remediation:** Identify and fix the flaws
- **Set Security Standards for Development and Deployment:** Prevent the introduction of vulnerabilities
- **Ensure Ongoing Assessment and Assurance:** Provide monitoring, and auditing

The fact that security vulnerabilities exist is a more immediate concern than how they got there. As soon as a threat is introduced that exploits a software vulnerability, an organization is faced with potentially costly damage control. Activities 1 and 2 are immediate priorities, while 3 and 4 provide the means for ongoing effective practices. Setting security standards for development and deployment may well be the most important step in preventing the introduction of vulnerabilities, but information from the risk assessment and vulnerability management processes will help to set realistic standards.

1. Risk Assessment:

The risks posed by software vulnerabilities can be measured and factored into the organization's overall risk management program. The U.S. National Institute of Standards and Technology

(NIST) maintains the National Vulnerability Database (NVD), a searchable index of information on computer vulnerabilities. As of July 2005, this Metabase contained 10,619 vulnerabilities, exposing organizations to risk of attack.⁶ Analysis of data from the NVD data indicates software accounts for more than 94% of vulnerabilities.



Protection begins with analyzing existing applications for security vulnerabilities and establishing priorities to eliminate them or mitigate their potential impacts. The range of technology tools now available significantly improves the organization's ability to assess the security state of its applications.

⁶ NIST National Vulnerability Database, <http://nvd.nist.gov>

A critical element of any internal control framework is the performance of enterprise risk assessment and management. An organization's risk model should identify its risk appetite and the key elements of risk (threats, vulnerabilities, probability, and mitigation) that impact the organization's ability to manage its risks within an acceptable range (risk tolerance).⁷

“Businesses should conduct a ‘business impact analysis’ as part of the process of evaluating vulnerabilities. Without this process (which would inherently be designed to put true business risks in context) it is difficult to get the attention of the executives who drive resource and capital allocation. This is the real reason that these issues don’t get attention – because security people talk in code and business people make business decisions based on operational and financial implications and expectations. Unless someone links these, the problems won’t get the attention they need.”

-J. Russell Gates, Dupage Consulting

Tools for software vulnerability risk assessment include penetration testing, manual review of code, and automated code scanning.

a) Penetration Testing

Network and system scanning and penetration testing (pen-test) tools can provide a variety of useful information, and these tools are steadily increasing in their sophistication. Penetration testing and scanning are techniques to analyze networks for faulty and poorly configured services, applications, and operating systems. Techniques include “ethical hacking” to determine vulnerability to an external attack invading externally visible servers or devices such as the domain name server (DNS), e-mail server, Web server, or firewall. Such tests may also include “social engineering” and simulated internal hacks that mimic network attacks by a disgruntled employee or a visitor with authorized access privileges.

The downside of scanning and pen-test tools is that they are also in the hands of attackers. So whether or not you deploy them to identify and remediate vulnerabilities, it is likely someone else will apply them to your systems to identify and exploit those vulnerabilities. (This guide does not describe attack methodologies or techniques, but they are well documented in the resources identified in the bibliography.)

Gartner estimates there are only 500 software engineers worldwide with the skill and knowledge necessary to efficiently scan code for security problems.

Scanning and pen-testing can be expensive yet may not deliver sufficient information to isolate and resolve security vulnerabilities in systems. Further, it can be difficult to determine how frequently scans and pen-

tests should be performed to assess the impacts of changes and/or newly discovered threats or vulnerabilities. Since these techniques do not scan the program code, they do not get to the heart of the vulnerabilities. So while penetration testing remains a valuable tool to test the security of the software in deployment, it alone cannot address the in-depth, ongoing requirements of a software security assurance program.

b) Manual Review of Program Code

Manual review of program code is an important step in the development process. It has been recognized as good programming practice since the earliest days of programming. However, even the best programmers and reviewers have typically not been educated to recognize the myriad security vulnerabilities that may inadvertently be written into code. In fact, Gartner

⁷This guide is not a treatise on risk management. For guidance on risk management see the bibliography. For guidance specifically related to internal assessment and auditing of risk management, visit The Institute of Internal Auditors web site at www.theiia.org.

estimates there are only 500 software engineers worldwide with the skill and knowledge necessary to scan code for security problems efficiently and effectively⁸. Further, manual review is laborious and time-consuming, difficult to manage, and not a viable solution for identifying and assessing the seriousness of vulnerabilities in large bodies of program code.

Cost to Repair:

The cost to repair a security vulnerability during the early stages of source code program development is about 2% of the cost to repair that same flaw in a production environment. And the repair cost does not take into account the potential costs associated with the exploit of security vulnerabilities.

The use of manual review for the products of outsourced program development is counter to two of the main reasons for outsourcing – reducing overall cost, and reducing dependencies on highly skilled technical staff.

No matter who performs the manual code reviews, or how extensive the quality and security measures for source code may be, manual review activities are still subject to human error and variability of results.

c) Automated Code Scanning Tools

Automated source code vulnerability scanning tools have recently emerged in an environment where they are sorely needed. These tools can be deployed during development where the cost to repair a vulnerability is about 2% of the cost of repairing that same vulnerability in a production system.⁹ And the costs of recovering from the exploit of that same vulnerability in a production system, including the impacts on reputation, customers, business partners, and potential regulatory sanction defy realistic measurement. The use of automated source code vulnerability scanning tools can also be included as a condition of contracts for the outsourced development of software.

Source code vulnerability scanning tools can be used as a discovery device to measure the extent and seriousness of vulnerabilities in production systems. Without such discovery, the organization has no reliable means for measuring vulnerabilities or planning protective, monitoring, and mitigative actions to reduce these security risks.

Code scanning tools can be used during program development to identify vulnerabilities as soon as they are created rather than later when their correction may impact other dependent code or multiple iterations of the flawed code.

Multiple Iterations of Flawed Code
A popular method of system design is to produce reusable segments of code, or “objects” that are placed in a library to be used by any program or system needing the function or process performed by that object. Proliferation of flawed code can greatly increase vulnerability.

They can perform an essential security and quality management process during acceptance testing, including the testing of code from outsourced programming services. And they can be deployed by security management, auditors, and even outside testing services to assess production systems to determine their reliability in the context of the system of internal controls.

Some advantages to automating the source code security analysis process include:

- **Speed:** Greater coverage is available through use of a tool that can reliably accomplish in minutes or even seconds what would otherwise be a tedious and less reliable manual process carried out over many days by skilled technicians.
- **Objectivity:** Automated tools apply known, reliable algorithms that can be reliably enhanced as new threats and vulnerability types are identified. Manual reviews can produce a wide range of results depending on the person(s) performing the review. Automated scans reliably produce consistent results across a wide range of programs, and are not subject to human limitations such as availability, fatigue, or distraction.

⁸Press Release: Gartner Debunks Six Information Security Myths, Victor Wheatman, managing VP Security, September 20, 2004

⁹Gartner: Pescatore, John, “Sanctum Buy Shows Security Is Key to Application Development”, FirstTake FT-23-5794, Gartner Research, July 30, 2004.

- **Depth of Analysis:** Automated tools can address all the various resources, options, and entry points within or pertaining to an application or business process. Parts will not be overlooked due to deadlines, fatigue, or judgment errors.
- **Measurable Results:** Automated scanning can support the establishment of minimum baselines and targeted thresholds for vulnerability management. And repeated use of the tools can provide reliable evidence of progress toward meeting objectives, and complying with policies or standards.

Software Security Metrics

Measuring the extent of software security vulnerabilities involves not only occurrence but also the severity of potential consequences of exploits. Location and type of vulnerability contribute to the seriousness more so than number of vulnerabilities.

Automated scanning of source code can, with minimum impact on resources, provide a set of metrics identifying the extent of source code vulnerabilities, the potential impact level of those vulnerabilities, identification of the most vulnerable systems or applications, and the information to assess the extent and priority of remediation required.

2. Vulnerability Management and Remediation

a) Fix the Flaws

Vulnerability assessment should identify the systems representing the greatest risks and establish tolerances for acceptable level of risk. Likelihood of exploit and value of assets threatened will determine severity. Risk severity, value of remediation, and availability of resources, will determine the remediation plan and schedule.

The software security metrics and remediation plan should also target the most efficient means to mitigate risks. Not every vulnerability can or should be fixed. Flawed code may be repaired or rewritten, or it may be “wrapped” within other protective code. The remediation plan should identify the specifics of problems identified as well as remediation approach.

- Specific identification of each problem’s location, including file, line, and column will increase remediation efficiency.
- Clear descriptions of problems including potential impacts and severity of abuse will provide the added benefit of educating developers on secure programming concepts and improve performance on current and future projects.
- Conclusive recommendations for alternate programming structures or more secure routines will minimize the time investment to resolve vulnerabilities.
- Aggregation of issues according to location, problem type, and vulnerable routine, will allow resolution efforts to be mapped into other development or maintenance processes, and help guide future development and change management.

For some sensitive legacy applications rewriting code is not a feasible option. Analysis in these cases may direct remediation toward wrapping – providing secure interfaces that validate transactions without disturbing the sometimes fragile and outdated application itself. Other applications may be too insecure for remediation or wrapping, and must simply be replaced.

Baseline security metrics will establish affordable and achievable remediation objectives, and the remediation plan will determine how objectives are met and measured.

3. Security Standards for Development and Deployment

Organizations must establish appropriate standards for application security and ensure all processes work together in accordance with those standards.

a) Set Security Standards

The absence of mandatory or even generally accepted standards for system and program security should not stop an organization from establishing secure coding requirements and standards. As previously mentioned, millions of references to secure coding standards are available via a Web search. Refining the search and discovery techniques will help identify those practices and techniques most applicable to the organization and its objectives.

Default standards can also be adopted as a by-product of implementing an automated code scanning tool. These tools include libraries of common design and coding flaws as well as information about techniques and practices necessary to prevent or remediate them.

A process to establish and maintain secure coding standards could begin concurrently with the project to assess available scanning tools and select the one most suited to the organization's needs. The same knowledge needed for tool selection can contribute toward the establishment of ongoing requirements and standards for processes and practices. A key ingredient is efficiency. Efficient security practices and standards are effective, affordable, and tailored to the organization's needs and activities.

b) Stop Writing Insecure Code

The default responsibility for preventing security vulnerabilities in source code often falls to the systems development team. The marketplace of the last decade focused on features over security, which has resulted in the problems of today's security-conscious world. Developers are actually in a difficult position, balancing security requirements and delivery deadlines against the market forces and internal demands that drive them. Another issue is that "secure" code has not been a key priority simply because, until recently, it has not been practical to achieve.

No matter how hard you try to write or approve only secure code, we cannot forget that security holes, or "bugs," can be introduced even when secure programming is practiced. Still it is important, to the extent possible, to minimize the introduction of security vulnerabilities in code.

In pursuit of secure source code, tools and services now available enable organizations to evaluate security in the design and coding of applications and to identify potential areas of vulnerability as soon as they appear. Whether these tools are used individually or in combination, security managers now have a more effective arsenal to demonstrably manage and measure their software security.

c) Build Security Requirements into Change Management, QA, and Testing

When system and program changes occur as a result of problem resolution or maintenance processes, an opportunity is provided to implement security vulnerability assessment and remediation into the change process. Quality assurance and acceptance testing can also be enhanced to include assessment of security vulnerabilities.

As security vulnerability measurement and remediation becomes standard practice, the state of security in online applications will rapidly advance toward the desired level of acceptable risk. Security vulnerability measurement and remediation must become integral for all processes related to program design, development, incident response, and change management.

In March of 1928, a tug boat called The T.J. Hooper encountered bad weather off the coast of New Jersey and lost the barge of coal it had been pulling. Had the ship had a working radio, it would have become aware of the storm ahead and might have saved its load. But radios were a relatively new invention and it was not the custom in the shipping industry at the time to equip boats with them.

Custom carries great weight before the law and is very often the source of law. The social norms and preferred practices that a specific community or industry has developed shapes behavior and affects legal expectations. If commercial custom in an area has for a long time held that debts are not delinquent until a day after they are due, then a court will treat that as law and will not penalize a debtor who took advantage by paying a day after the date on his note.

Despite the primacy of custom, Judge Learned Hand found the owners of The T.J. Hooper liable for the loss of the coal citing the lack of a radio as negligent. He wrote: "Courts must in the end say what is required; there are precautions so imperative that even their universal disregard will not excuse their omission." Even though it was not the norm for ships to not have radios, Hand was saying, the norm should be the opposite and that is the law he applied.

"Norms in a Wired World" by Steven A. Hetcher

d) The Need for Automated Controls

The availability of effective vulnerability measurement and management tools and techniques presupposes their use to the extent that failure to apply them could be regarded as negligence. Customers are developing expectations that errors will be immediately corrected when

identified. Similarly, the need to correct vulnerabilities exists within a small window. News of vulnerabilities spreads quickly in the hacker world, and exploits that used to take days or weeks can be prepared and launched in a matter of minutes. The increasing expectation is that automated incident protection mechanisms will immediately respond to attacks and alert humans as needed.

Change Management

Change management is an important subject in its own right, but too broad to cover effectively here. See "Visible Ops" and the "Change Management" GTAG in the bibliography for more information.

4. Ongoing Assessment and Assurance

a) Monitoring

Today, any reliable risk assessment of an organization engaged in electronic commerce via the Internet will identify cyber attacks as a key threat, software weaknesses as a key vulnerability, and a high and the probability that such attacks will occur is high, and will continue to increase. The likelihood of successful cyber attacks is influenced by the attractiveness of the target, and the ability of the enterprise to prevent, detect, and recover from cyber incidents.

Every organization with Internet-facing systems must maintain preventive, detective, and corrective controls to mitigate the risks of cyber attacks.

In recent years the application of continuous monitoring, assessment, auditing, and reporting (collectively called continuous assurance) has increased dramatically because of the increasing incidence of risks and cyber threats. Coincidentally, software tools to support periodic or continuous assurance have also improved dramatically. Although continuous measurement and assurance applications are not yet regarded as the norm, they make use of readily available tools, and should certainly be regarded as effective business practice.

Change is certain. Change management is a choice. Ideally an organization's change management process will be sufficient to prevent the introduction of vulnerabilities in new or changing systems. But some changes may evade even the most stringent change

management controls, and new types of exploits may take advantage of coding practices not previously thought to represent security vulnerabilities. Changes in purposes and uses of online applications can also result in new vulnerabilities.

In any well managed environment, ongoing assessment is a critical element of assurance practice. Security vulnerability management must be assessed to ensure it incorporates up-to-date data about vulnerability types, that program libraries are routinely scanned for vulnerabilities, and that vulnerability scans remain required practice for all changes.

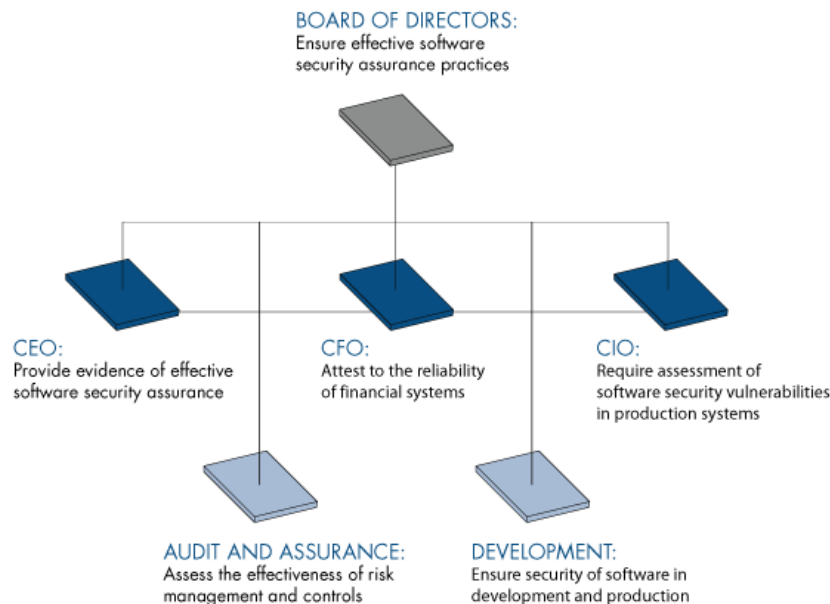
b) Audit

Audit review provides an independent assessment and attestation to management's assurance of an effective system of security vulnerability management. Audit analysis and reporting on the effectiveness of significant controls is mandated by the Sarbanes-Oxley Act. Other legislation and regulations around the world are also increasing their recognition of the value of audit assessment and assurance regarding the effectiveness of significant internal controls – particularly in the realm of information and technology reliability and security.

Internal auditing is an important element of the overall system of internal controls. Internal audit is a control that functions by evaluating the effectiveness of other controls. For more information about internal auditing, see The Institute of Internal Auditors (www.theiia.org). For specific coverage of internal audits roles in information technology security and assurance, see the IIA Global Technology Audit Guide (GTAG) series. Appendix B of this guide provides an example audit program and internal control questionnaire for assessment of source code vulnerability management.

B. Roles and Responsibilities

Current legislation – including Sarbanes-Oxley (SOx), Gramm-Leach-Bliley, HIPAA, CA SB 1386, PIPEDA (Canada), the EU Data Protection Directive and similar or related laws from around the world – places responsibility for effective internal controls squarely in the hands of senior management and the board of directors. Practices can be delegated, but not responsibility, as illustrated in the Federal Sentencing Guidelines (U.S.). As SOx offers the most pressing current requirements for management responsibility regarding internal controls, it is used here as the basis for the following synopsis of management, governance, and audit roles and responsibilities. Additional details and references to related requirements and guidance are provided in appendixes B and C.



1. Board of Directors

- Ensure management practices and reports provide evidence of effective software security assurance practices.
- Assess management's determination of acceptable risk levels for the organization in light of stakeholder interests and legal obligations.
- Ensure adequate resources, including competent human resources, are provided for software security assurance.
- Ask trenchant questions – for example: “Do security policies include the requirement that no high-severity vulnerabilities be harbored in any systems accessible via the Internet or interfacing with Web-based systems?” See also “Information Security Management and Assurance: A Call to Action for Corporate Directors” in Appendix E: References.

2. Audit and Assurance

- Assess the effectiveness of risk management and control practices related to software security assurance.
- Ensure information security policy specifically addresses software vulnerability management and provides for compliance with applicable laws and regulations.
- Assess the effectiveness of processes to manage software vulnerabilities within the tolerances of the organization's risk appetite.
- For further guidance on the audit role and an example audit program, see Appendix B.

3. CEO

- Ensure management structures, practices, and reports provide evidence of effective software security assurance practices.
- Personally attest to the reliability of controls related to financial information and its processing and reporting – including security assurance for Internet-facing applications.

4. CFO

- Personally attest to the reliability of controls related to financial information and its processing and reporting – including security assurance for Internet-facing applications.
- Seek specific assurance that the level of risk associated with software security vulnerabilities is within prescribed risk tolerances for the organization (i.e. no high-severity vulnerabilities, and no excessive spending to mitigate minor risks).

5. CIO

- Ensure effective management structure, practices, and resources to manage software security vulnerabilities within the organization's prescribed risk appetite.
- Ensure vulnerability assessments express actual and potential consequences in terms of “business impact” rather than only expressing technical consequences.
- Assess software security vulnerabilities in production systems and provide evidence to senior management of effective software security assurance.
- Sponsor the development and implementation of secure coding standards within the development process.
- Ensure the development process requires (automated) application security testing before systems are deployed.

6. Executive(s) Responsible for Systems Development and Change Management

- Ensure systems development and deployment processes provide applications that meet defined security standards.
- Ensure change management prevents introduction of software security vulnerabilities.
- Ensure security requirements apply equally to in-house and outside software.

VII. CLOSING SUMMARY

The measurement and management of security vulnerabilities in source code is, to date, typically not handled well. But with the increasing scrutiny of the problems associated with cyber attacks, personal information privacy, critical infrastructure protection, and the reliability of Internet-facing systems, it is only a matter of time before the utilization of automated tools to manage software security assurance becomes accepted practice. The early implementers of software security management tools will reap tremendous benefits including operational efficiency, customer and business partner confidence, competitive advantage, regulatory compliance, the ability to take reliable advantage of the reduced costs of outsourced and offshore system and program development and maintenance, and much more.

This guide explains the risks, responsibilities, and opportunities associated with software security assurance. It points out the dynamic rate of change in the subject of risk management and associated controls. The following sections provide appendixes with greater detail than the body of the report. Of particular note are:

- **Appendix A: Audit Program and Internal Control Questionnaire for Source Code Vulnerability Management** – which provides detailed guidance for internal audits of software security assurance, and
- **Appendix C: Control Objectives and Practices** – which describes and cross references legal and regulatory requirements with available security, control, auditability, and governance guidance to support the case for application of effective software security assurance practices.

If you have any questions or concerns about the guide, please contact the author.

Charles H. Le Grand, www.chlglobalassociates.com

APPENDIX A: AUDIT PROGRAM AND INTERNAL CONTROL QUESTIONNAIRE FOR SOURCE CODE VULNERABILITY MANAGEMENT

1. Scope:

The management of risks associated with security vulnerabilities in source code for applications accessible via the Internet is a significant element of overall software risk management. The assessment of management practices for security of online applications begins with identification of key business dependencies and related risks. It addresses the reliability of general controls for protecting access to systems and data, but focuses on the specific controls for preventing, detecting, and correcting vulnerabilities in source code.

The context for vulnerability analysis must be “enterprise impact.” While the audit may focus on technical issues and controls, any opportunities to improve as noted in the audit should be based on improvement to the organization or its processes resulting from any technical improvements.

An important audit perspective is compliance with laws and regulations. To the extent that vulnerabilities found in software have the potential to impact compliance, that potential should be expressed. Potential impacts on the reliability or privacy of financial or other sensitive information, and/or the potential to disrupt important business processes or significant controls should be explained from a management and governance perspective.

General controls assessment begins with the “tone” for security, control, and assurance set at the highest management and governance levels of the organization. Applicability to source code security vulnerability management includes ensuring management policies, risk management, and security objectives are sufficiently comprehensive to include protection against security vulnerabilities in online systems. It concludes with determining effective practices are in place to ensure: that risks associated with online system security will be assessed and monitored; that the full extent of risks are communicated to a level of management appropriate to make decisions about the level of risk to be maintained; and ensuring incidents and/or changes to the ongoing level of risk are duly reported to an appropriate level of authority.

a) Systems Subject to Assessment

All systems that provide for access via the Internet are subject to online security assessment. Those applications specifically designed for browser-based access by customers, business partners, employees, etc. are subject to security assessment. Those applications that interface with Internet-facing systems may also be subject to assessment depending on the nature of the interface. If they are called and/or passed instructions, parameters, data, data requests, etc. from Internet facing systems, they are subject to assessment.

b) Key Assessment Areas:

The three main process areas related to source code security vulnerability management include:

- System Design, Development, and Testing
- System Implementation, Quality Assurance, and Change Management
- Vulnerability Assessment for Operational Systems

2. System Design, Development, and Testing:

a) Objective:

- Ensure a secure design, development, and programming environment. (See related operational, security, and audit assessments.)
- Ensure program development procedures include provisions for protection against the introduction of security vulnerabilities into source code.
- Ensure procedures specify how designers and programmers are held accountable for keeping systems free from security vulnerabilities.
- Ensure procedures are in place to maintain awareness of and protection from new security vulnerabilities and threats as they are identified.
- Ensure systems and program code from providers outside the organization are subject to the same requirements for protection from security vulnerabilities.
- Ensure code acceptance testing includes testing for security vulnerabilities.

3. Questions:

a) General Controls, Policies and Documentation

- What evidence exists of assurance that general controls for system design, development, and programming are sufficient and adequately monitored?
(Note: control weaknesses in key areas such as separation of duties, access controls, authentication, monitoring and reporting, system development, change management, etc. may call for increasing the scope of the source code security vulnerability assessment.)
- How are the responsibilities of designers and programmers to ensure protection against security vulnerabilities in source code documented and communicated to responsible individuals?
- How are security and vulnerability protection requirements communicated to outside providers of systems and programs?
- How are designers and programmers trained and kept informed of secure design and coding practices and techniques?

b) Metrics, Preventive and Detective Controls

- What metrics exist to assess and monitor the extent of security vulnerabilities in source code?
- How are security vulnerability metrics maintained and enhanced as new threats and vulnerabilities are identified?
- What procedures exist to protect against the introduction of security vulnerabilities in program code as it is written?
 - Peer, quality, and management review of design and code?
 - Independent review of design and code?
 - Use of automated tools to identify security vulnerabilities?
- How are the requirements to ensure protection against security vulnerabilities in source code documented and communicated to outside parties developing systems and program code for the organization?
- How does the acceptance testing process for both programs and systems ensure the level of security vulnerabilities in systems and programs is within the level of tolerance established by the organization?

- How does the acceptance testing process simulate the operation and protection of the system in a production environment?
 - How does such simulation include testing for vulnerability to external or internal attacks?
 - How are penetration testing procedures assessed to ensure they cover the full range of potential threats and vulnerabilities for each system tested?
- As part of the acceptance testing process and a condition of approval, how are system owners (also called users) apprised of the:
 - Level of residual security vulnerabilities in systems and programs?
 - Potential impacts on the organization if security vulnerabilities are exploited and the probability of such exploits?
 - Costs versus benefits of further reductions in security vulnerabilities?

c) System Implementation, Quality Assurance, and Change Management:

- How are Systems and Network Operations Management and Security Management apprised of the level of security vulnerability represented by new or changed systems before they are moved into the production systems environment?
- What quality assurance processes do operational and security management functions perform when moving new or changed systems into production?
- How do change management processes ensure the same set of security vulnerability management controls applicable to new systems development and implementation are applied to system changes? To outside-developed products?
- How are new and changed systems monitored after they are implemented to ensure they have not introduced vulnerabilities to the production environment?
- What testing (code scans, penetration testing, monitoring of transaction traffic, etc.) is applied to new or changed systems after they have been implemented?
- What post-implementation reviews are applied to new and changed systems and programs?
- How is the overall production environment monitored to assess the results of new and changed systems on processing and establish new norms and tolerance levels as needed?

d) Vulnerability Assessment for Operational Systems:

- What procedures are in place to assess systems already in production for security vulnerabilities?
 - Penetration testing?
 - Monitoring of transactions and processing for anomalous conditions?
 - Analysis of source code for security vulnerabilities?
 - Use of source code security vulnerability scanning tools?
 - Monitoring of security incidents for clues to new threats and vulnerabilities?
- What procedures exist to inform appropriate management of the ambient level of security vulnerabilities in production systems and the potential for successful exploits?

APPENDIX B: ROLES AND RESPONSIBILITIES FOR SOFTWARE SECURITY ASSURANCE

Responsibility and ownership of systems and the processes for effective systems management and assurance are now widely acknowledged as applying to many different positions within the organization and key outside parties. The responsibilities of governance, management, operational, and technical level positions should not only be clearly described, they should also be frequently reassessed as new challenges and opportunities arise due to changing business practices, technologies, threats, and vulnerabilities.

1. The Importance of Software Risk Management

The management of security vulnerabilities in source code is a key element of increasing importance in overall risk management. The significance of software security assurance increases as business functions and customer services are added to organizations' Web-based services. The Internet introduces new threats at an alarming pace, and the best security practices of some of the world's best organizations have been compromised by threats that materialized before effective protection could be implemented.

Evidence indicates any vulnerabilities harbored in online systems will eventually be exploited. That is why commercial software providers issue so many patches and encourage system operators to install them immediately. Exploits of vulnerabilities previously took months to appear once the vulnerability was known. Then it became weeks, then days... Now exploits are deployed in a matter of hours.

Effective management practices in information security tend to develop and become accepted at a slower pace than the vulnerability, threat, exploit, protection cycle. This means it takes a long time for new effective practices to become recognized and broadly applied – often too long. While management, auditors, and regulators seek to determine which practices are necessary, cost effective, and/or required, the attacks and compromises continue to expand. Consequently, the individuals responsible for risk management in any organization must ensure sufficient resources are provided to assess, measure, and monitor the threats to the organization resulting from vulnerabilities in this critical component of the overall system of internal controls. And they must ensure the responses to new threats are timely and sufficient to provide assurance of continuous and sustainable controls.

2. The Parties to Software Security Assurance

Organizational roles and responsibilities for information security can be classified as governance, management, and technical. Some frameworks include operational, but that category can generally be divided among management and technical.

The "governance" of IT and information security is broadly addressed in current publications because the roles of board members and executive management have been illuminated through the increased legislation, regulation, and monitoring that resulted from the financial scandals and steadily increasing cyber incidents. This increased attention has sharpened the focus on risks to be managed in protecting and ensuring the reliability of information (business and personal), financial management and reporting, and the protection of stakeholder interests. At risk, too, is the director's personal liability with regard to prudent practice and effective oversight.

The governance and management of security and reliability is impacted and improved through effective control of vulnerabilities in the software that controls online business processes. As indicated in the COSO Internal Control Integrated Framework, the proper environment for effective controls is established by the "tone at the top" or executive management's message to the organization about the importance (rather than merely the appearance) of effective controls.

The following sections address management roles and responsibilities as specifically related to software security assurance. Position descriptions and titles may be different within different countries, industries, and organizations, and some roles may be merged in smaller organizations, but someone in the organization must still address the function. References to broader scope and responsibilities are provided in the bibliography.

a) Board of Directors

The “governance” level of the organization is typified in the Board of Directors. The Audit Committee of the board is most likely the entity responsible for assessing information security and reliability, assuring compliance with laws and regulations, and interfacing with the assurance management elements within the organization. The board sets policy and maintains contact with the organization’s key executives in ensuring effective leadership, direction, and strategic alignment of resources and objectives.

While the desired governance role of the board is “noses in, fingers out,” audit committee members may find themselves applying “gentle direction” if they perceive certain organizational roles may need increased attention or resources, or they believe additional evidence is needed for adequate assurance. Because directors are expected to oversee the reliability of financial information and financial reporting (for compliance with the Sarbanes-Oxley act), they must have sufficient evidence that the parties responsible for controls continuously meet their responsibilities.

Audit committee members probably will not want any details about the management of security vulnerabilities in source code. They will want to know that competent individuals with adequate

Do security policies include the requirement that no high-severity vulnerabilities be harbored in any systems accessible via the Internet?

resources have examined the full set of information controls and found them to be appropriate, continuous, and effectively monitored. And they *will* want to know the details if the organization’s systems or data are compromised as a result of source code vulnerabilities. After such an event, or after learning of such an event in another organization, they may also want to know that security policies include the requirement that no high-severity vulnerabilities be harbored in any systems accessible via the Internet or interfacing with Web-based systems.

b) Chief Auditing Executive, CAE (or Chief Internal Auditing Officer)

“The internal auditor’s role in IT controls begins with a sound conceptual understanding, and culminates in providing the results of risk and control assessments. Internal auditing involves significant interaction with the people in positions of responsibility for controls, and requires continuous learning and reassessment as new technologies emerge and the organization’s opportunities, uses, dependencies, strategies, risks, and requirements change.” (From Information Technology Controls, in the Global Technology Audit Guides series, The Institute of Internal Auditors, March 2005)

An organization’s internal auditors may provide services including financial auditing, operational analysis, assurance assessments, consulting, governance support, automated testing and analyses, continuous monitoring and auditing, fraud or forensics assessments or investigations, and more. Frequently internal auditors will find or create a way to measure the impacts or potential impacts of a problem or risk. They may also pioneer the use of analytical tools and techniques that subsequently become part of management’s monitoring and assurance processes. Such tools themselves then become subject to audit assessments.

An effectively placed internal auditing function reports administratively to executive management at a level sufficient to ensure independence and objectivity in their work. Internal auditors build relationships throughout the organization to ensure concerns are identified and resolved

The internal auditor provides assurance regarding the organization's business risks, financial statements, system of internal control, and level of compliance with laws, regulations, and policies. And a professional internal auditing function complies with "International Standards for the Professional Practice of Internal Auditing" by The Institute of Internal Auditors (IIA).

The internal auditor will tailor the audit program and approach to best meet the organization's needs. Internal auditing is a control function that adds value to the organization by assessing the effectiveness and efficiency of the organization's overall set of controls, provides appropriate recommendations when controls should be improved, and provides assurance that management's assertions of effective controls are reliable.

The internal auditor is a key player in supporting the roles of the CEO and CFO who must personally attest to the reliability of internal controls. And the internal auditor must work closely with IT and security management as the parties most directly responsible for the quality and reliability of information and system security controls.

i. Internal Audit and Software Security Assurance

The internal audit program and allocation of audit resources are typically based on addressing the areas of greatest risk to the organization. Internal auditing will prepare a risk assessment, and ensure it is consistent with the enterprise-wide risk management function and governance views.

The information security policy should be reviewed to ensure it provides sufficient and timely requirements regarding security risk management and monitoring. Then the auditor should assess whether implementation of the policy, via procedures and techniques, is adequate to ensure the intent of the policy is met and that practices are appropriate for compliance with applicable laws and regulations.

Since threats to the organization via the Internet (also called cyber threats) are so prevalent today, and the potential consequences of the realization of those threats are tremendous, the internal auditor should assess the processes whereby these risks are measured and monitored. An essential part of the assessment is measuring the cost of risk reduction and determining the feasibility of practices, tools, and techniques.

ii. Measuring the Effectiveness of Vulnerability Reduction

Until recently it was not feasible to measure vulnerabilities in source code. Now tools are available to scan code while it is in development, and for systems already in production. The best tools provide an index of vulnerabilities categorized by their severity.

In building support for recommendations regarding source code vulnerabilities, the auditor may identify examples of the costs of software vulnerability exploits, determine or estimate the probability of such exploits against the organization's systems, and compare the expectable loss to the costs of mitigating these risks. For information on quantifying security risks to the organization, see Appendix A of "Information Security Management and Assurance: A Call to Action for Corporate Governance," identified in the Bibliography and Web References.

The auditor may also want to consider the reasons why vulnerabilities have been allowed to exist in code written by or for the organization. Often control problems have their bases in lack of awareness of the extent of risk unwittingly accepted on behalf of the organizations executives, board, and stakeholders. Bringing such risks to light could set in motion a plan to remediate the risks, or it could be only the first step in seeking the allocation of resources for remediation.

Appendix D of this guide provides an example audit program for assessing the

management of source code security vulnerabilities. The internal auditor should adjust and augment this program as appropriate to the environment subject to audit.

c) CEO (Chief Executive Officer)

The Chief Executive Officer has overall management responsibility for investments in and the use of technology. Since the passage of the Sarbanes-Oxley act, most CEOs now also have overall accountability for the system of internal controls including security, reliability, and compliance. Since SOx requires the CEO to personally accept responsibility for risk management and controls, the CEO will want to know the persons directly responsible for controls are held accountable and measured as to how well they meet their responsibilities.

As the CEO likely does not want to become an expert in risk management, assessment, or mitigation (let alone managing source code vulnerabilities), the internal auditor can play an important role in apprising the CEO on the effectiveness of these processes and related controls. However, the CEO cannot delegate the mandated responsibilities for controls and security of information including:

- The organization's objectives and performance measures
- Ownership of critical success factors
- Technology strategies
- Availability of appropriate resources
- Management and executive attention to new and emerging security issues
- Management reporting on the effectiveness of the system of internal controls.

d) CFO (Chief Financial Officer)

The Chief Financial Officer has specific and personal responsibility and liability for reliable financial information, processing, and reporting, and for management reporting on the system of internal controls relevant to financial information. Consequently, the CFO should:

- Obtain general and specific knowledge as necessary to understand how the organization's strategies are impacted by technology
- Understand how IT objectives and strategies are impacted by risk and security issues
- Seek reliable assurance that areas of high risk in IT and security are effectively managed, monitored, and audited.

The CFO should specifically seek assurance that the level of risk associated with software security vulnerabilities is within prescribed risk tolerances for the organization (i.e. no high-severity vulnerabilities, and no excessive spending to mitigate minor risks).

e) CIO (Chief Information Officer)

The CIO is responsible for structural (organizational) and procedural controls for the management and reliability of information technology (IT), information, and information systems (IS). S/he is responsible for general and technical management and controls as well as ensuring technology resources and controls are aligned with business objectives.

Some organizations may have a separate chief security officer or chief information security officer. Such positions may or may not report to the CIO. But typically they are responsible for assessment, strategy, design, development, and monitoring of security elements. They are not responsible for security activities as carried out in the context of routine processing. That responsibility falls to the CIO.

i. Responsibility for Information Security

The CIO is likely the position most directly responsible for information security as it is applied and managed within the organization on a regular basis. Some organizations may have a Chief Risk Officer, Chief Security Officer, and/or Chief Information Security Officer as appropriate to the organization's size, mission, or complexity. In such cases, the responsibilities for managing security and protecting against software security vulnerabilities may be shared by such positions.

Information security is a significant element of the overall system of internal controls. Information security is essential for ensuring the reliability of financial information and reporting. It is the key ingredient in protecting sensitive and private information. It is the tool for providing accountability among individuals authorized to act on behalf of or in cooperation with the organization. It is the basis for knowing information accurately recorded will remain accurate and that all changes will be recorded and traceable to a responsible person. It is a key component of recoverability from errors, omissions, corruption, disruptions, and attacks.

The controls that provide security of information and technology have evolved into a continuum with overlaps, redundancies, continuous monitoring, complementary control processes, and logging of any action or transaction that may provide evidence needed to resolve violations of controls and security. However, they are not bullet-proof, and must also provide for recovery from inappropriate actions. By design, weaknesses in one area of IT controls are compensated by strengths in another. An example is the monitoring of indicators that would reveal the compromise of a security vulnerability in an online system.

ii. Responsibility for Software Security Management:

The CIO is also the position most directly responsible for managing security vulnerabilities in software developed by or for the organization.

Software security vulnerabilities are a known (and perhaps the most frequent) avenue of cyber attack. The incidence of software security vulnerabilities puts additional burden on the already strained capabilities of intrusion protection controls. The availability of software tools to detect, measure, and mitigate the incidence of software security vulnerabilities creates a responsibility on the part of the CIO to assess the value of this control for mitigating the risks of cyber attacks within the organization's overall risk management process.

f) Management of Systems Development

The executive in charge of systems development has a multifaceted role, typically interfacing directly with the "Owners" of:

- Business processes and the systems that support them
- Information Systems and Networks – including operations management
- Outsourced (perhaps off-shore) developers of systems and programs
- Customer Support functions
- Information Security (and perhaps overall Security)
- Risk Management
- Financial Management

Systems development is where good or bad program code is put together to process, store, manage, and distribute data, transactions, and information. The best way to solve the problem of source code security vulnerabilities is to prevent them at their source. A programming group well-trained and skilled in efficiently producing secure and efficient code is the objective. But the results have historically been difficult to measure until the systems began to reveal their flaws and vulnerabilities during production processing.

Many books have been written, many educational courses delivered, and many approaches tried on the subject of systems design and development. Yet systems development and the “Systems Development Life Cycle” remain among the greatest challenges in the entire information technology realm. This guide does not attempt to solve those problems. It only addresses source code vulnerability management in the context of systems development.

i. Assess the Process

An important rule for effective process management is to identify problems and errors (issues) as close to their source as possible, and to use the information about these issues to provide incentives for the responsible persons to prevent or eliminate them at the source.

Logically, good rules for program coding, effective training of program designers and coders, and close monitoring of code as it is written will contribute to eliminating security vulnerabilities in new or modified code. (Libraries of secure code objects are another means for preventing introduction of vulnerabilities while increasing the efficiency of programming tasks.) But even with the best design and programming practices, it is important to scan new code at each step of its development to detect and remove any security vulnerabilities before they can be propagated or otherwise impact the design or coding of other programs.

Use of automated tools for scanning or new or changed code should be an essential task in each step of program development. Then as programming procedures and techniques mature, the frequency of code scans can be assessed to determine the most efficient practices.

ii. Quality Assurance

An essential step in quality assurance for systems development is acceptance testing. Acceptance testing is another broad topic, and cannot be addressed in-depth in this guide. However, acceptance testing is typically a final step before new or revised program code is approved for implementation in the production environment. Therefore, acceptance testing should include a final scan for security vulnerabilities in source code. (For more information on managing changes to systems and networks, see the Global Technology Audit Guide on Change Management at www.theiia.org/technology.)

Quality assurance (QA) involves the measurement of a process or product against a given standard. Depending on the needs of the organization and system users, it may not be practical to eliminate all security vulnerabilities in source code (particularly for large volumes of code from existing or legacy systems). However, the nature of QA should, at a minimum, include the measurement of vulnerabilities against a known scale, and identification of how each program fits within the range of acceptable values for that type of program.

Some programs may be sensitive enough that any known vulnerabilities must fit into a low probability and/or low impact category. Other programs may tolerate greater vulnerabilities because the systems themselves do not represent high risk to the organization or its stakeholders.

iii. Resource Constraints

Organizations with large libraries of program code, perhaps written before the availability of tools to detect vulnerabilities, may establish a project to scan those code libraries and set priorities for remediating the vulnerabilities found. The same rules that apply to assignment of resources to mitigate risks and improve controls must apply to reduction of source code security vulnerabilities – the benefit to the organization must be greater than the cost of the control.

Availability of tools to scan source code makes the cost side of the equation fairly simple. And the benefit of having a reliable metric of the level of vulnerabilities in systems also greatly simplifies the reliability of risk measurement.

The role of the systems development officer may vary depending on the organization, but among the interfaces identified at the beginning of this section lies responsibility for ensuring ongoing assessment of source code vulnerability is required for all new and changed systems, and that periodic automated scanning of production applications is also a required management practice.

g) External Auditor:

The primary role of the external auditor is to attest to management's assertions regarding the reliability of financial information, financial reporting, and the system of internal control. (This is an over-simplification, but is adequate for the purposes of this guide.) The external auditor generally reports to the audit committee of the board regarding financial systems and reporting, and internal control.

Independent external audits are a requirement for most organizations, and are normally performed annually. With regard to information security and management of source code vulnerabilities and the role of the organization's external auditors, the internal auditing department and the Audit Committee of the Board may wish to consider:

- The extent to which security vulnerabilities could impact the reliability of financial information and reporting,
- The overall reliability of information systems and related IT controls, and
- The scope of and responsibilities for examining the information systems and controls during any formal attestation that may be required by statute or regulation (e.g. internal controls over financial reporting and other regulatory requirements).

The external auditor may also provide updates on pending accounting pronouncements and their potential impact on the organization. Such pronouncements today in the USA most likely come from the Public Company Accounting Oversight Board (PCAOB, the governing body for U.S. based accounting and auditing firms). New PCAOB rules, typically based on interpretations from the Sarbanes-Oxley Act, may put the organization's management, board, and external auditors at risk for failure to identify or report significant control weaknesses. Further, privacy of personal or business information must also be subject to reliable internal controls.

The probability that security vulnerabilities in source code could actually impact the reliability of financial information and reporting, privacy, and the system of internal controls must be assessed by management and independently by the external auditor. Typically such probability would be remote, but neither the organization nor the auditor can assume that to be the case without first assessing the extent to which the organization and its data rely on the security of online and Internet-facing systems.

For example, if the computer or server that manages access to financial information and reporting also supports remote or Internet access, then a relatively simple flaw in assignment of access privileges can open financial information to unlawful disclosure, corruption, disruption, or destruction. Responsible management, the board, and the external auditor are all required to provide assurance of the reliability of financial information and reporting, and the reliability of the overall system of internal controls.

The availability of metrics regarding the state of software security vulnerabilities is an important element of online systems security. But it is not the only relevant metric. The security system must also ensure vulnerabilities in the network, operating systems, and systems software will prevent and detect unauthorized access to system-controlled resources.

APPENDIX C: CONTROL OBJECTIVES AND PRACTICES

1. Security, Reliability, and Compliance Frameworks

Wouldn't it be nice if you could buy a framework that would exactly fit your organization and provide all the right mechanisms to ensure compliance, manage risk within your risk appetite, provide ongoing evidence that information and its security are reliable, and ensure the protection of customer and business information and privacy? The problem is every organization is just different enough that no one-size fits all.

Some control elements are common to all organizations. Enterprise management must evaluate specific frameworks and guidance to determine the elements and details appropriate to management and measurement of key internal controls within the organization.

Example components of a framework for Sarbanes-Oxley and other regulatory compliance:

- *Governance and management processes for reliability of financial information and reporting*
 - *Performance of enterprise risk assessment and management*
 - *Internal control identification, documentation and ongoing assessment including:*
 - *The system(s) of internal control*
 - *Significant accounts*
 - *Significant controls*
 - *Management reporting on the effectiveness of internal controls*
 - *Identification and remediation of significant control deficiencies*
 - *Independent audits of internal control adequacy and the reliability of financial reporting*
 - *Sustainability of controls and information reliability*
-

2. Software Security Assurance and Related Control Frameworks, Requirements, Standards, and Guidance: COSO, SOx, COBIT, AND ISO/IEC 17799

- The **COSO** Internal Control Integrated Framework is recognized by the SEC and PCAOB as suitable for compliance with provisions of the Sarbanes-Oxley Act (SOx). The framework is high-level and general enough to accommodate the myriad variations of internal control frameworks needed by the individual organizations subject to SOx compliance. COSO also acknowledges that guidance for technology controls must be provided elsewhere and is subject to continuous change just as technology and its applications are subject to continuous change.

It is worth noting that in 2004 COSO released its Enterprise Risk Management framework. While the ERMF has not attained the recognition or status of the ICIF, it does provide specific guidance related to the risk assessment and risk management elements of Sarbanes-Oxley compliance.

- **SOx** compliance requires that an organization assess its risks and ensure the significant risks are well covered in the system of internal controls. Many controls identified in control frameworks represent effective business control practices, but are not necessarily the most significant controls relative to the risks addressed in SOx. That is why SOx specifies risk assessment and management as essential components of the management and control

environment. It is also worth noting that SOx says nothing about IT controls. It is the responsibility of management, governance, and auditors to assess the relative significance of IT controls.

- **ISO/IEC 17799** enjoys broad global acceptance and is another model suitable for building an organization's compliance framework for information security and controls. The 17799 framework also has sufficient flexibility that each organization must determine its own approach to details of information security and control that are not specified within 17799 guidance.
- **COBIT** provides a higher level of detail in many areas than ISO/IEC 17799, and is rapidly gaining ground as a framework for SOx compliance. But again, as no one framework can provide both adequate detail and sufficient flexibility to be universally applicable, it is still up to each organization to provide and document its own control framework for SOx compliance.

3. Key Issues:

Integration: Achieving compliance with key regulations, while optimizing operations by integrating an organizational approach to security, availability, and processing integrity. As a result, risk management competencies and prioritization of initiatives gain strategic importance.

Compliance Strategy: SOx compliance is based on the reliability of financial information and its processing and reporting. Clearly business controls are much broader than that. For example, business continuity and disaster recovery are not deemed relevant to SOx compliance because they address what might happen rather than the reliable recording and reporting of what has happened. But no viable organization would last for long doing business and electronic commerce today without reliable processes to protect and recover from interruptions and disruptions. The same is true for many key business controls that are not necessarily considered significant for SOx compliance assurance purposes. Consequently the organization's risk management and compliance strategies must go far beyond SOx.

4. Assessing and Applying Compliance Guidance in Software Security Assurance

The following sections describe COSO, ISO/IEC 17799, and COBIT as tools for managing SOx compliance. It is written specifically to address the risks of security vulnerabilities in source code for Internet-facing applications. Although this perspective is somewhat esoteric to this specific risk issue, the comparison of the frameworks will also be helpful in identifying and assessing the applicability of available guidance for other IT and business controls.

5. COSO

COSO refers to the Committee of Sponsoring Organizations for the Commission on Fraudulent Financial Reporting (also known as the Treadway Commission). See www.coso.org. The COSO "Internal Control Integrated Framework" is a recognized formal model for compliance with the Sarbanes-Oxley act. The PCAOB's audit standards indicate:

"Because of the frequency with which management of public companies is expected to use COSO as the framework for the assessment, the directions in the standard are based on the COSO framework. Other suitable frameworks have been published in other countries and likely will be published in the future. Although different frameworks may not contain exactly the same elements as COSO, they should have elements that encompass all of COSO's general themes."

COSO provides high-level guidance for managing internal controls including IT controls. Created by accountants and auditors (American Institute of CPA's, American Accounting Association, Financial Executives International, Institute of Internal Auditors, and Institute of Management Accountants), its focus is on internal controls relevant to financial information, processing, and reporting. The COSO Internal Control Integrated Framework (circa 1991) acknowledges the importance of IT controls, but provides only a few pages addressing their overall impacts. It indicates other sources of information on IT controls are needed, and that the IT control

environment is subject to dynamic change.

In 2004 the COSO model was refined and enhanced, resulting in the COSO Enterprise Risk Management – Integrated Framework. This ERM framework supplements (rather than replacing) the ICI framework.

COSO defines internal controls as follows:

“Internal control is a process, effected by an organization’s board of directors, management and other personnel, designed to provide reasonable assurance regarding the achievement of objectives in the following categories:

- Effectiveness and efficiency of operations
- Reliability of financial reporting
- Compliance with applicable laws and regulations

The first category addresses an entity's basic business objectives, including performance and profitability goals and safeguarding of resources. The second relates to the preparation of reliable published financial statements, including interim and condensed financial statements and selected financial data derived from such statements, such as earnings releases, reported publicly. The third deals with complying with those laws and regulations to which the entity is subject.

a) Internal Control Integrated Framework

The COSO framework describes internal control in five interrelated components:

i. Control Environment

Tone at the top sets the control environment for the organization, influencing the control consciousness of its people, and establishing a foundation, discipline, and structure for all other controls. It includes integrity, ethical values and individual competence; management's philosophies and style; assignment of authority and responsibility; and the attention and direction provided by the board of directors.

- Emphasize alignment of IT with business, not as a separate organization/ control environment
- IT may introduce additional risks requiring their own subset of control activities
- Ownership of IT controls may be unclear, especially for application controls

ii. Risk Assessment

A precondition to risk assessment is establishment of objectives, linked at different levels, and internally consistent. Risk assessment also includes measurement of risk factors (vulnerabilities, threats, probability, and expected impacts). Automated measurement of source code vulnerabilities provides important information for overall risk assessment.

- Identify and analyze the relevant risks to achieve predetermined objectives – this is the basis for determining control activities.
- Perform formal risk assessments throughout the systems development methodology, built into the infrastructure operation and change process, and built into the program change process

iii. Control Activities

Policies and procedures help ensure management directives and business objectives are accomplished. Prevention, detection, and mitigation of security vulnerabilities in source code are important control activities for any organization with online systems. General controls include (for software security assurance):

- Access security controls
- Application system development and maintenance controls, embedded within software programs to prevent or detect unauthorized transactions

iv. Information and Communication

Information must be identified, captured, and communicated in a form and timeframe that enables people to perform their responsibilities. Security weaknesses in systems can compromise an organization's ability to manage and control information and communications.

- Determine the quality and relevance of available information, and ensure it is transmitted to the appropriate parties.

v. Monitoring

Ongoing monitoring occurs in the course of operations. But it is neither simple nor automatic. The more control structures can be simplified and strengthened, the more meaningful management monitoring can be. Secure source code strengthens the control structure and simplifies the monitoring processes.

- Defect identification and management: establishing metrics and analysis of trends
- Security monitoring: building an effective IT security infrastructure
- Internal audits (including IT internal audit reviews)
- External audits
- Regulatory examinations
- Attack and penetration studies
- Independent performance and capacity analyses
- IT effectiveness reviews
- Independent security reviews

6. Sarbanes-Oxley

The Sarbanes-Oxley Act of 2002 (SOx) brought about sweeping changes within the accounting profession and for the management and governance of publicly traded companies. The most relevant sections of the act for software security assurance are sections 302 and 404.

Briefly, section 302 requires management to evaluate and report on the effectiveness of disclosure controls and procedures with respect to the quarterly and annual reports. The principal executive (CEO) and financial officers (CFO) must certify that financial information and reports are accurate, and that the system of internal controls is appropriate to ensure the reliability and security of financial information and reporting.

Section 404 of SOx requires management's development and monitoring of procedures and controls for making their required assertion about the adequacy of internal controls over financial

reporting, as well as the required attestation by an external auditor of management’s assertion.

More than two years after passage of the Act, the SEC, PCAOB, public accountants, and organization management continued to struggle with identifying those controls deemed “significant” in regards to their potential for materially impacting financial reporting. For many companies SOx compliance represents a major commitment of valuable resources.

Achieving SOx 404 compliance is frequently a major corporate initiative consisting of several phases and specific activities within each phase. The following table summarizes typical phases, activities, and person(s) responsible:

<i>Phase/Activity</i>	<i>Lead Responsibility</i>
Planning	
Plan	Project Sponsor
Scope	Project Team
Execution	
Documentation	Line Managers and/or Project Team and/or Specialists
Evaluation & Testing	Line Managers / Project Team / Specialists
Issues	Project Team and Line Managers
Corrective Action	Line Managers
Monitoring Systems	Senior Management
Reporting	
Management Reporting	Senior Management and Line Managers
External Audit Reporting	External Auditor
Monitoring	
Ongoing Monitoring	Senior Management
Periodic Assessment	Project Team and/or Line Managers

Because SOx specifically addresses financial information and all the processes related to managing this information, ensuring its reliability and security, and ensuring reliable financial reporting, it necessarily applies to information security and management controls. A breach in information security that could allow insiders or attackers to compromise financial information or systems would certainly be considered “significant” to SOx compliance, and would require management and auditors to disclose the breach and its possible consequences.

As the Securities and Exchange Commission (SEC) and the Public Company Accounting Oversight Board (PCAOB) continue to establish rules and standards to tighten the interpretation of SOx provisions, it remains clear that systems and software security are integral to compliance.

The following list summarizes those elements of a SOx compliance program relevant to the assurance of information and software security.

A Framework for SOx Compliance

1. Plan and Scope
 - Financial reporting process
 - Define supporting systems
2. Perform Risk Assessment
 - Probability and impact to business
3. Identify Significant Accounts/Controls
 - Application controls over initiating, recording, processing and reporting (COSO: design applications to prevent/detect unauthorized transactions. Combined with manual controls to ensure completeness, accuracy, authorization and validity)
 - IT general controls (COSO: those that support the quality and integrity of information and are designed to mitigate risks)
4. Document Control Design
 - Policy manuals
 - Procedures
 - Narratives
 - Flowcharts
 - Configurations
 - Assessment questionnaires
5. Evaluate Control Design
 - Mitigate control risk to an acceptable level
6. Evaluate Operational effectiveness
 - Internal Audit
 - Technical testing
 - Self-assessment
 - Inquiry
7. Identify and Remediate Deficiencies
 - Internal Control Deficiencies: A design or operating deficiency may exist when a necessary control is missing or badly designed such that the objective is not always met. An operating deficiency may exist when a well-designed control is not operating as designed or there is 'user error.'
 - Significant deficiency: an internal control deficiency that could have "more than inconsequential" results
 - Material weakness: significant deficiency or deficiencies that "preclude the entity's internal control from providing reasonable assurance that material misstatements in the financial statements will be prevented or detected on a timely basis by employees in the normal course of performing their assigned functions."
 - Remediation
8. Document Process and Results
 - Coordination with auditors
 - Internal signoff (includes section 404)
 - Independent signoff (404)
9. Build Sustainability
 - Internal and external evaluations

7. The COBIT framework:

COBIT (Control Objectives for Information and related Technology) is a widely recognized framework for information, systems, and technology controls, compliance, and auditing. Promulgated by the Information Systems Audit and Control Association, ISACA refers to COBIT as an “Open Standard.”

COBIT contains a set of 34 high-level control objectives and 318 specific control objectives for IT processes grouped into **Four Domains**:

- Plan and Organize
- Acquire and Implement
- Deliver and Support
- Monitor and Evaluate

a) Plan and Organize

The Plan and Organize domain covers strategy and tactics, and identification of the ways IT contributes to achievement of the business objectives. As Web-facing systems increase in importance within the business and systems plan, the importance of managing security vulnerabilities continues to increase.

- PO1: Define a strategic IT plan
- PO2: Define the information architecture
- PO3: Determine the technological direction
- PO4: Define the IT organization and relationships
- PO5: Manage the IT investment
- PO6: Communicate management aims and direction
- PO7: Manage human resources
- PO8: Ensure compliance with external requirements
- PO9: Assess risks
- PO10: Manage projects
- PO11: Manage quality

b) Acquire and Implement

Each organization must ensure it builds Internet-facing infrastructure to a high standard because it becomes part of the global information infrastructure. Weak components and vulnerabilities put your organization and your stakeholders at risk.

The availability of consistent, reliable, and affordable tools for measuring and managing security vulnerabilities in source code opens a new realm of “best practices” in acquiring and implementing systems and programs.

- AI1: Identify automated solutions
- AI2: Acquire and maintain application software
- AI3: Acquire and maintain technology infrastructure
- AI4: Develop and maintain procedures
- AI5: Install and accredit systems
- AI6: Manage changes

c) Deliver and Support

Delivery and support of online systems includes the measurement of performance features of those systems. Support also includes managing and tracing problems and incidents to their source and to the specific vulnerability exploited. Vulnerability remediation is another key element of support.

- DS1: Define and manage service levels
- DS2: Manage third-party services
- DS3: Manage performance and capacity
- DS4: Ensure continuous service
- DS5: Ensure systems security
- DS6: Identify and allocate costs
- DS7: Educate and train users
- DS8: Assist and advise customers
- DS9: Manage the configuration
- DS10: Manage problems and incidents
- DS11: Manage data
- DS12: Manage facilities
- DS13: Manage operations

d) Monitor and Evaluate

Monitoring and evaluating online systems includes the measurement of vulnerability as well as any exploits of those systems. Given that the typical organization with online systems implemented many of them before automated tools became available to support assessment of security vulnerabilities, it is now important to assess the extent of vulnerabilities in those systems and determine the steps needed to remediate vulnerabilities based on their potential impacts.

- M1: Monitor the processes
- M2: Assess internal control adequacy
- M3: Obtain independent assurance
- M4: Provide for independent audit

e) Control Objectives Relevant to Software Security Assurance

PO9: Assess Risks

9.1 Business Risk Assessment: Management should establish a systematic risk assessment framework. Such a framework will incorporate a regular assessment of information risks relevant to the achievement of the business objectives, forming a basis for determining how risks should be managed to an acceptable level. The process should provide for risk assessments at both the global level and system specific level, for new projects as well as on a recurring basis, and with cross-disciplinary participation. Management should ensure reassessments occur and that risk assessment information is updated with results of audits, inspections and identified incidents.

9.2 Risk Assessment Approach: Management should establish a general risk assessment approach that defines the scope and boundaries, the methodology to be adopted for risk

assessments, the responsibilities and the required skills. Management should lead the identification of the risk mitigation solution and be involved in identifying vulnerabilities. Security specialists should lead threat identification and IT specialists should drive the control selection. The quality of the risk assessments should be ensured by a structured method and skilled risk assessors.

9.3 Risk Identification: The risk assessment approach should focus on the examination of the essential elements of risk and the cause/effect relationship between them. The essential elements of risk include tangible and intangible assets, asset value, threats, vulnerabilities, safeguards, consequences and likelihood of threat. The risk identification process should include qualitative and, where appropriate, quantitative risk ranking and should obtain input from management brainstorming, strategic planning, past audits and other assessments. The risk assessment should consider business, regulatory, legal, technology, trading partner and human resources risks.

9.4 Risk Measurement: The risk assessment approach should ensure the analysis of risk identification information results in a quantitative and/or qualitative measurement of risk to which the examined area is exposed. The risk acceptance capacity of the organization should also be assessed.

9.5 Risk Action Plan: The risk assessment approach should provide for the definition of a risk action plan to ensure cost-effective controls and security measures mitigate exposure to risks on a continuing basis. The risk action plan should identify the risk strategy in terms of risk avoidance, mitigation or acceptance.

AI1: Identify Automated Solutions

1.1 Definition of Information Requirements: The organization's system development life cycle methodology should ensure the business requirements satisfied by the existing system and to be satisfied by the proposed new or modified system (software, data and infrastructure) are clearly defined before a development, implementation or modification project is approved. **The system development life cycle methodology should require that the solution's functional and operational requirements be specified including performance, safety, reliability, compatibility, security and legislation.**

1.9 Cost-Effective Security Controls: Management should ensure the costs and benefits of security are carefully examined in monetary and non-monetary terms to guarantee the costs of controls do not exceed benefits. The decision requires formal management signoff. **All security requirements should be identified at the requirements phase of a project and justified, agreed and documented as part of the overall business case for an information system.** Security requirements for business continuity management should be defined to ensure the planned activation, fallback, and resumption processes are supported by the proposed solution.

1.10 Audit Trails Design: The organization's system development life cycle methodology should require that adequate mechanisms for audit trails are available or can be developed for the solution identified and selected. **The mechanisms should provide the ability to** protect sensitive data (e.g., user ID's) against discovery and misuse.

AI5: Install and Accredit Systems

5.7 Testing of Changes: Management should ensure that changes are tested in accordance with the impact and resource assessment in a separate test environment by an independent (from builders) test group before use in the regular operational environment begins. Back-out plans should also be developed. Acceptance testing should be carried out in

an environment representative of the future operational environment (e.g., similar security, internal controls, workloads, etc.)

5.9 Final Acceptance Test: Procedures should provide, as part of the final acceptance or quality assurance testing of new or modified information systems, for a **formal evaluation and approval of the test results by management of the affected user department(s) and the IT function**. The tests should cover all components of the information system (e.g., application software, facilities, technology, user procedures).

5.12 Promotion to Production: Management should define and implement formal procedures to control the handover of the system from development to testing to operations. Management should require that system owner authorization is obtained before a new system is moved into production and that, before the old system is discontinued, the new system will have successfully operated through all daily, monthly and quarterly production cycles. The respective environments should be segregated and properly protected.

DS1: Define and Manage Service Levels

1.2 Aspects of Service Level Agreements: Explicit agreement should be reached on the aspects a service level agreement should have. **The service level agreement should cover at least the following aspects:** availability, reliability, performance, capacity for growth, levels of support provided to users, continuity planning, **security, minimum acceptable level of satisfactorily delivered system functionality**, restrictions (limits on the amount of work), service charges, central print facilities (availability), central print distribution and change procedures.

1.5 Review of Service Level Agreements and Contracts: Management should implement a regular review process for service level agreements and underpinning contracts with third-party service providers.

DS7: Educate and Train Users

7.3 Security Principles and Awareness Training: All personnel must be trained and educated in system security principles, including periodic updates with special focus on **security awareness** and incident handling. Management should provide an education and training program that includes: ethical conduct of the IT function, security practices to protect against harm from failures affecting availability, confidentiality, integrity and performance of duties in a secure manner.

M1: Monitor the Processes

1.2 Assessing Performance: Services to be delivered by the IT function should be measured (key performance indicators and/or critical success factors) by management and be compared with target levels. Assessments of the IT function should be performed on a continuous basis.

1.4 Management Reporting: Management reports should be provided for senior management's review of the organization's progress toward identified goals. Status reports should include the extent to which planned objectives have been achieved, deliverables obtained, performance targets met and risks mitigated. Upon review, appropriate management action should be initiated and controlled.

M3: Obtain Independent Assurance

3.3 Independent Effectiveness Evaluation of IT Services: Management should obtain independent evaluation of the effectiveness of IT services on a routine cycle.

3.4 Independent Effectiveness Evaluation of Third-Party Service Providers: Management should obtain independent evaluation of the effectiveness of IT service providers on a routine cycle.

3.5 Independent Assurance of Compliance with Laws and Regulatory Requirements and Contractual Commitments: Management should obtain independent assurance of the IT function's compliance with legal and regulatory requirements, and contractual commitments on a routine cycle.

3.6 Independent Assurance of Compliance with Laws and Regulatory Requirements and Contractual Commitments by Third-Party Service Providers: Management should obtain independent assurance of third-party service providers' compliance with legal and regulatory requirements and contractual commitments on a routine cycle.

3.7 Competence of Independent Assurance Function: Management should ensure the independent assurance function possesses the technical competence, skills, and knowledge necessary to perform such reviews in an effective, efficient and economical manner.

3.8 Proactive Audit Involvement: IT management should seek audit involvement in a proactive manner before finalizing IT service solutions.

8. ISO/IEC 17799

Although called an international standard, ISO/IEC 17799 is actually classified as a "Code of practice for information security management." Much of the material is high-level and open to broad interpretation. It is adopted by ISO/IEC from the British Standards Institute where it is Part 1 of the two-part BS 7799. ISO/IEC 17799 consists of 12 sections. Pertinent "Standards" start at section 3. (Note the ISO/IEC draft adaptation of BS 7799 Part 2 was released while this guide was being prepared.)

- Scope
- Terms and Definitions
- Security Policy
- Organizational Security
- Asset Classification and Control
- Personnel Security Management
- Physical and Environmental Security
- Communications and Operations
- Information Access Management Control
- Systems Development and Maintenance
- Business Continuity Management
- Compliance Management

The standards within ISO/IEC17799 most relevant to software security assurance include:

Section 8. Communications and Operations

- 8.1 Establish operational procedures
 - 8.1.2 Control changes to facilities and systems
- 8.3 Protect against malicious software
 - 8.3.1 Detect and prevent malicious software

Section 10. Systems Development and Maintenance

- 10.1 Identify system security requirements
 - 10.1.1 Specify security controls and requirements that new information systems must meet
- 10.2 Build security into your application systems
 - 10.2.1 Build input data validation controls into your application systems
 - 10.2.4 Build output data validation into your systems
- 10.3 Use cryptography to protect information
- 10.5 Control development and support
 - 10.5.1 Establish change control procedures
 - 10.5.2 Review changes to operating system
 - Review and test application systems whenever OS changes
 - Make sure OS changes do not adversely effect applications
 - 10.5.4 Safeguard against covert channels and Trojans
 - Purchase programs from reputable sources
 - Inspect all source code before you use it
 - 10.5.5 Control outsourced software development

Section 12. Compliance Management

- 12.2 Perform security compliance reviews
 - 12.2.2 Review technical security compliance
 - Carry out penetration tests to detect information security vulnerabilities

9. Cross-Reference Matrix for Private Sector Guidance Applicable to Software Security Assurance

Software Assurance Requirements	Pertinent Activities	COBIT	ISO/IEC 17799	COSO Component	FFIEC Information Security IT Examination Handbook (GLBA)	IT Control Objectives for Sarbanes-Oxley
RISK ASSESSMENT						
<p>Assess level of risk: Identify, prioritize, and develop remediation strategy to mitigate or accept relevant software risks</p>	<ol style="list-style-type: none"> 1. Prepare risk management plan to address most significant risks. 2. Determine "risk appetite" and define acceptable levels of system security risk 3. Consider cost-effective means to identify and manage the identified security risks through security practices 4. Develop risk management strategy to mitigate software risk and establish security controls 	<p>PO9: 9.1 Business Risk Assessment, 9.3 Risk Identification DS5: 5.8 Data Classification</p> <p>PO9: 9.5 Risk Action Plan AI1: 1.9 Cost-Effective Security Controls DS7: 7.3 Security Principles and Awareness Training</p>	<p>4.1 Establish a security infrastructure 4.2 Coordinate information security implementation 5.2 Use an information classification system 10.1 Identify system security requirements</p> <p>3.1 Establish an information security policy 4.1 Establish a security infrastructure 6.3 Respond to information security incidents</p>	Risk Assessment	<p>A risk assessment provides a foundation for the remainder of the security process by guiding the selection and implementation of security controls and the timing and nature of testing those controls. Testing results, in turn, provide evidence to the risk assessment process that the controls selected and implemented are achieving their intended purpose. Testing can also validate the basis for accepting risks.</p> <p>Management may decide that since some risks do not meet the threshold set in their security requirement, they will accept those risks and not proceed with a mitigation strategy. Other risks may require immediate corrective action. Still others may require mitigation, either fully or partially, over time.</p>	<p>Management prepares strategic plans for IT that align business objectives with IT strategies.</p> <p>The IT organization's risk assessment framework measures the impact of risks according to qualitative and quantitative criteria, using inputs from different areas including, but not limited to, management planning, past audits and other assessments.</p> <p>The IT organization's risk assessment framework is designed to support cost-effective controls to mitigate exposure to risks on a continuing basis, including risk avoidance, mitigation or acceptance.</p>
VULNERABILITY MANAGEMENT AND REMEDIATION						
<p>Test software and technology infrastructure: Review acceptance criteria and evaluate code to determine acceptable security thresholds prior to deployment</p>	<p>5. Test software against functional and operational system requirements, which should include business value and security threshold requirements.</p>	<p>AI5: 5.7 Testing of Changes 5.11 Operational Test 5.12 Promotion to Production AI5: 5.9 Final Acceptance Test, 5.13 Evaluation of Meeting User Requirements, 5.14 Management's Post-Implementation Review</p>	<p>4.1 Establish a security infrastructure, 5.1 Make information asset owners accountable</p> <p>8.2 Develop plans to provide future capacity (8.2.1 use acceptance criteria to test systems)</p>	Control Activities	<p>Application and operating system source code can have numerous vulnerabilities due to programming errors or misconfiguration. Where possible, financial institutions should use software that has been subjected to independent security reviews of the source code, especially for Internet facing systems.</p>	<p>The organization has a system development life cycle methodology that considers security, availability and processing integrity requirements of the organization.</p> <p>Procedures exist to ensure that system software is installed and maintained in accordance with the organization's requirements.</p>

Software Assurance Requirements	Pertinent Activities	COBIT	ISO/IEC 17799	COSO Component	FFIEC Information Security IT Examination Handbook (GLBA)	IT Control Objectives for Sarbanes-Oxley
Identify automated technology solutions. Consider security when identifying automated solutions	<p>6. Determine security of the selected technology through code analysis</p> <p>7. Include security thresholds as part of acceptance criteria in requirements documents.</p>	<p>AI1: 1.1 Definition of Information Requirements</p>	—	—	—	—
MANAGE OUTSOURCED PROVIDERS						
<p>Include software security as an acceptance criteria in service level agreements with third parties: Define and manage service levels.</p>	<p>8. Ensure that management establishes security requirements and regularly reviews compliance of internal SLAs and contracts with 3rd party service providers.</p>	<p>DS1: 1.2 Aspects of Service Level Agreements 1.5 Review of SLAs and Contracts</p> <p>DS2: 2.3 Third-Party Contracts 2.8 Monitoring</p> <p>AI4: 4.1 Operational Requirements and Service Levels</p>	<p>4.3 Control outsourced information processing (4.3.1 Use contracts to control outsourced services)</p> <p>6.1 Control your personnel recruitment process</p> <p>10.5 Control development and support (10.5.5 control outsourced software development)</p> <p>12.2 Perform security compliance reviews (12.2.2 Carry out penetration tests to detect information security vulnerabilities)</p>	<p>Control Environment</p> <p>Control Activities</p> <p>Monitoring</p>	<p>Establish security requirements, acceptance criterion and test plans</p>	<p>Procedures exist and are followed to ensure that a formal contract is defined and agreed to for all third party services before work is initiated, including definition of internal control requirements and acceptance of the organization's policies and procedures.</p> <p>A regular review of security, availability and processed integrity is performed for service level agreements and related contracts with third-party service providers.</p>

Software Assurance Requirements Manage third party services: Require evidence of security acceptance criteria.	Pertinent Activities 9. Mitigate security and confidentiality risk from third party providers by defining source code security acceptance criteria in Service Level Agreements. 10. Require objective evidence of compliance with security acceptance criteria.	COBIT DS2: 2.6 Continuity of Services 2.7 Security Relationships	ISO/IEC 17799 4.2 Control Third party access to facilities 4.3 Control outsourced information processing (4.3.1 use contracts to control outsourced services) 6.3 Respond to information security incidents (6.3.2 Report security threats and weaknesses) 8.1 Establish operational procedures 8.7 Control interorganizational exchanges 10.5 Control development and support (10.5.5 control outsourced software development)	COSO Component Control Environment Risk Assessment Control Activities Monitoring	FFIEC Information Security IT Examination Handbook (GLBA) Review and test source code for security vulnerabilities, including covert channels or backdoors that might obscure unauthorized access into the system. Perform security tests to verify that the security requirements are met before implementing the software in production.	IT Control Objectives for Sarbanes-Oxley —
---	--	--	--	---	---	--

Software Assurance Requirements	Pertinent Activities	COBIT	ISO/IEC 17799	COSO Component	FFIEC Information Security IT Examination Handbook (GLBA)	IT Control Objectives for Sarbanes-Oxley
MONITOR AND AUDIT						
Manage changes: Ensure continuing security and stability of software by enforcing a strict change management procedure that includes patches and updates, scheduled or emergency.	11. Evaluate all changes, including patches, to establish the impact on the integrity, exposure or loss of sensitive data, availability of critical services, and validity of important transactions by analyzing source code prior to rollout.	AI5: 5.7 Testing of changes AI6: 6.4 Emergency Changes	8.1 Establish operational procedures (8.1.2 Control changes to facilities and systems.) 10.5 Control development and support (10.5.4 Inspect all source code before you use it)	Control Activities Monitoring	The source code reviews should be repeated after the creation of potentially significant changes. Procedures exist to ensure that system software changes are controlled in line with the organization's change management procedures. IT management ensures that the setup and implementation of system software do not jeopardize the security of the data and programs being stored on the system.	[Expect] risk assessments built into the program change process. Procedures exist to ensure that system software changes are controlled in line with the organization's change management procedures. IT management ensures that the setup and implementation of system software do not jeopardize the security of the data and programs being stored on the system.
Regularly evaluate the performance of information security. Review software for compliance with requirements and current security conditions.	12. Assess adequacy of defined security controls. 13. Evaluate software for weaknesses.	M1: 1.2 Assessing Performance 1.3 Assessing Customer Satisfaction 1.4 Management Reporting M2: 2.1 Internal Control Monitoring	9.7 Monitor system access and use 12.2 Perform security compliance reviews	Control Activities Information and Communication Monitoring	Risk assessments should be updated as new information affecting information security risks are identified. At least once a year, senior management should review the entire risk assessment to ensure relevant information is appropriately considered.	The IT organization monitors its progress against the strategic plan and reacts accordingly to meet established objectives.

<p>Software Assurance Requirements</p> <p>Software Audit: Obtain assurance of compliance with regulations and frameworks pertinent to critical systems, either through a third party or internal audit team.</p>	<p>Pertinent Activities</p> <p>14. Review security controls; assess compliance with laws, regulations and contracts.</p>	<p>COBIT</p> <p>M3: 3.3 Independent Effectiveness Evaluation of IT Services 3.4 Independent Effectiveness Evaluation of Third-Party Service Providers 3.5 Independent Assurance of Compliance with Laws and Regulatory Requirements and Contractual Commitments. 3.6 Independent Assurance of Compliance with Laws and Regulatory Requirements and Contractual Commitments by Third Party Service Providers. 3.7 Competence of Independent Assurance Function</p>	<p>ISO/IEC 17799</p> <p>—</p>	<p>COSO Component</p> <p>—</p>	<p>FFIEC Information Security IT Examination Handbook (GLBA)</p> <p>Institutions should design tests to produce results that are logical and objective. Results that are reduced to metrics are potentially more precise and less subject to confusion, as well as being more readily tracked over time.</p>	<p>IT Control Objectives for Sarbanes-Oxley</p> <p>The organization monitors changes in external requirements for legal, regulatory or other external requirements related to IT practices and controls. Control activities are in place and followed to ensure compliance with external requirements, such as regulatory and legal rules.</p>
---	---	--	--------------------------------------	---------------------------------------	---	--

10. Cross-Reference Matrix for Public Sector Guidance Applicable to Software Security Assurance

Compliance Category	NIST Special Publication 800-53 (NIST Implementation Guide for FISMA)	Department of Defense Instruction Number 8500.2, Information Assurance Implementation	DISA Application Security Checklist	ISO/IEC 17799
Identification and Authentication	<p>IA-7 Cryptographic Module Authentication</p> <p>SC-13 Use of Validated Cryptography</p>	<p>IAKM-1 Key Management</p> <p>IA TS-1 Token and Certificate Standards</p> <p>DCNR-1 Non repudiation</p> <p>ECCR-1 Encryption for Confidentiality (Data at Rest)</p> <p>ECCR-2 Encryption for Confidentiality (Data at Rest) (classified non-SAMI)</p>	<p>APP0140 An application user or client authentication process is inadequate (must inventory all client authentication processes in the application)</p> <p>APP0330 The application utilizes an unapproved cryptographic module</p> <p>APP0580 Application users can circumvent the intended user interface to access resources in its supporting infrastructure.</p>	<p>9.1 Control access to information</p> <p>10.3 Use cryptography to protect information</p> <p>10.3.1 Develop a policy on the use of cryptography</p>
Risk Assessment	<p>RA-3 Risk Assessment: Conduct assessments of the risk and magnitude of harm that could result from the unauthorized access, use, disclosure, disruption, modification, or destruction of information and information systems. Risk assessment take into account vulnerabilities, threat sources, and security controls.</p> <p>RA-4 Risk Assessment Update: Update the risk assessment whenever there are significant changes to the information system, facility, or other conditions</p>	<p>DCDS-1 Dedicated IA Services</p> <p>DCII-1 Impact Assessment</p> <p>E3.3.10 DoD IA program shall regularly and systematically assess information systems, services and supporting infrastructures</p>	<p>N/A</p>	<p>4.0, Organizational Security</p> <p>4.1 Establish a security infrastructure</p> <p>4.2 Control third party access to facilities</p> <p>6.2.1 Control your information security training</p>
Vulnerability Management	<p>RA-5 Vulnerability Scanning: Using appropriate vulnerability scanning tools and techniques, scan for vulnerabilities in the information system or when significant new vulnerabilities are identified and reported.</p>	<p>ECMT-1 Conformance Monitoring and Testing</p> <p>VIVM-1 Vulnerability Management</p>	<p>APP1020 The application does not adequately validate user inputs before processing them</p> <p>APP1030 The application is vulnerable to buffer overflows</p>	<p>8.2.2 Use acceptance criteria to test systems</p> <p>8.3 Protect against malicious software</p> <p>10.2.1 Build input data validation into your systems</p> <p>10.2.4 Build output data validation into your systems</p> <p>10.5.4 Safeguard against covert channels and Trojans (inspect all source code before you use it)</p>

Compliance Category	NIST Special Publication 800-53 (NIST Implementation Guide for FISMA)	Department of Defense Instruction Number 8500.2, Information Assurance Implementation	DISA Application Security Checklist	ISO/IEC 17799
Outsourced Information System Services	SA-9 Outsourced Information System Services: Third party providers employ adequate security controls, and the organization monitors security control compliance. Third party providers are subject to the same information system security policies and procedures of the organization. Service level agreements define the expectations of performance, describe measurable outcomes, and identify remedies for non-compliance.	DCDS-1 Dedicated IA Services DCIT-1 IA for IT Services	N/A	10.5.4 Safeguard against covert channels and Trojans (inspect all source code before you use it) 10.5.5 Control outsourced software development
Security Testing	SA-11 Developer Security Testing: Developer creates a security test and evaluation plan, implements the plan and documents results. Developmental security test results may be used in support of the security certification and accreditation process for the delivered information system. SA-3 Life Cycle Support: The organization manages the information system using a system development lifecycle methodology that includes information security considerations.	E3.4.4 All applications shall employ Information System Security Engineering (ISSE) as part of acquisition process or development to ensure IA is built in 5.8.1 Ensure that IA is incorporated as an element of DoD information system life cycle management processes	N/A	12.2 Perform security compliance reviews 12.2.2 Review technical security compliance (carry out penetration tests to detect information security vulnerabilities)
Vulnerability Remediation	SI-2 Flaw Remediation: The organization identifies, reports, and corrects system flaws. The organization identifies information systems containing proprietary or open source software affected by recently announced software flaws. Flaws discovered during security assessments should also be addressed.	DCSQ-1 Software Quality DCCT-1 Compliance Testing	APP1020 The application does not adequately validate user inputs before processing them APP1030 The application is vulnerable to buffer overflows	10.2.1 Build input data validation into your systems 10.2.4 Build output data validation into your systems 10.5.4 Safeguard against covert channels and Trojans (inspect all source code before you use it)

11. Software Security Assurance – A Management Compliance Checklist

a) *Monitoring Activities:*

Quality management:

- Does a quality plan exist for significant IT functions (e.g. system development and deployment)?
- Does the quality plan prescribe the type(s) of quality assurance activities (such as reviews, audits, inspections) to be performed to achieve the objectives of the quality plan?

Monitoring:

- Have performance indicators (e.g. benchmarks) from both internal and external sourced been defined, and are data being collected and reported regarding achievement of these benchmarks?
- Has IT management established appropriate metrics to effectively manage the day-to-day activities of the IT department?
- Are internal control assessments performed periodically, using self assessments or independent audits, to examine whether or not internal controls are operating satisfactorily?

b) *For Managers:*

Conditions to Check:

- Ensure sufficient resources and skills sets to exercise security responsibilities
- Consider security in job performance appraisals
- Integrate security in SDLC and explicitly addressed at each stage
- Ensure applicable security measures have been identified and implemented
- Establish rules for authorizing changes and for evaluating their security impact
- Ensure security aspects have been considered in all service level agreements and the security competence of the service providers has been assessed
- Ensure the security baseline and vulnerabilities have been constantly assessed through monitoring system weakness
- Ensure a measurable and management-transparent security strategy exists based on benchmarking, maturity models, gap analysis, and continuous performance monitoring and reporting
- Ensure all staff are aware they may be held legally responsible for a serious security breach

c) *For Executives:*

Questions to ask:

- When was the last risk assessment completed on the criticality of information security assets?
- Is the information security risk assessment a regular agenda item at IT management meetings and does management follow through with improvement initiatives?
- When was the latest policy statement issued on information security? Does it cover the identified risks and the control mechanisms established to address those risks? What are the monitoring and feedback procedures?
- What safeguards have been established over systems connected to the Internet to protect the entity from viruses and other attacks?

Action List:

- Set up and execute a risk management program that identifies threats, analyses vulnerabilities, assesses criticality and uses industry best practices for due care.
- Ensure a measurable and management-transparent security strategy is created based on benchmarking, maturity models, gap analysis, and continuous performance monitoring and reporting
- Regularly assess vulnerabilities through monitoring system weaknesses using CERT bulletins, intrusion and stress testing, and testing of contingency plans.
- Establish security baselines and rigorously monitor compliance

d) For Senior Executives:

“CIO’s must now take on the challenges of (1) enhancing their knowledge of internal control, (2) understanding their company’s overall Sarbanes-Oxley compliance plan, (3) developing a compliance plan to specifically address IT controls and (4) integrating this plan into the overall Sarbanes-Oxley compliance plan.” Note, the senior executives’ responsibilities in this area are much broader than SOx requirements, and were there before the act was passed. However, now that they are law, they are less subject to dispute

Action List:

- Ensure written policies, guidelines, and applicable standards have been documented and communicated across the organization.
- Develop and introduce clear and regular reporting on the organization’s information security status to the board of directors based on established policies and guidelines and applicable standards. Report on compliance with these policies, important weaknesses and remedial actions, and important security projects.
- Ensure information security audits are conducted based on clear process and accountabilities, with management tracking the closure of recommendations.

e) For Board of Directors:

Questions to Ask:

- Is security considered an afterthought or a prerequisite?
- Has management set up an independent audit of information security? Does management track its own progress on recommendations?

Action List:

- Insist that management make security investments and security improvements measurable, and monitor and report on program effectiveness.
- Ensure that the audit committee clearly understands its role in information security and how it will work with management and auditors.
- Ensure that internal and external auditors agree with the audit committee and management on how information security should be covered in the audit.
- Require a report of security progress and issues for the audit committee.

f) Risk Assessments: Important issues to consider (SOx, but applicable elsewhere)

- Integration: Is the IT department’s risk assessment process integrated with the company’s overall risk assessment process including financial reporting related risks?
- Process: Does the IT dept. document, evaluate and remediate IT controls related to financial reporting on an annual basis? (Or more frequently for SOx-related issues.)

- Response: Does the IT dept have a formal process in place to identify and respond to IT control deficiencies?
- Communication: Does the CIO have an adequate knowledge of the types of IT controls necessary to support reliable financial processing?

Risk Assessment Activities:

- Does the IT organization have an entity- and activity-level risk assessment framework that is used periodically to assess information risk to achieving business objectives? (Note: It is not just about IT risks, but includes risks occasioned by IT that impact the entire entity, and therefore should be integrated with entity-wide risk management.)
- Does the IT organization's risk assessment framework measure the impact of risks according to qualitative and quantitative criteria?
- Is a comprehensive security assessment performed for critical systems and locations based on their relative priority and importance to the organization?

What to include in a risk assessment:

Systems that process large volumes of transactions, process large dollar-value items, and/or are used to process complex transactions or support highly sensitive financial data repositories.

- Impact (effect of possible events)
 - Security failure on the reporting of financial info
 - Implementation of an unapproved change
 - Lack of system/application availability
 - Failure to maintain the system/application
 - Failure in the integrity of information managed by the system/application
- Probability (potential that they'll occur)
 - Volume of transactions
 - Complexity of technology/application
 - Volume and complexity of changes
 - Age of the system/application
 - Past history issues
 - Custom in-house programming vs. COTS

Computer Misuse Security Risks:

Ensure software security risk management practices specifically address:

- Trojan Horses
- Back door and remote administration programs
- DOS attacks
- Being an intermediary for another attack
- Unprotected Windows networking shares

- Mobile code (Java/JavaScript/ActiveX)
- Cross-site scripting
- E-mail spoofing
- Email borne viruses
- Hidden file extensions
- Chat clients
- Packet sniffing
- Identity theft
- Tunneling
- Zombies
- Spyware

(Note, this list is ad-hoc and intended only as an example of the types of issues to be addressed in software security assurance. Also, the list should be subject to continuous update and enhancement.)

APPENDIX D: IDENTIFYING VULNERABILITIES IN WEB APPLICATIONS: THE TOP SOURCES OF EXPOSURE TO LOCATE AND REMEDIATE

1. Unvalidated Sources of Input

A Web application should perform validation of all user input passed into the application. Security reviews must identify where systems might be vulnerable by pinpointing the following sources of input:

- URL parameters
- Form fields
- Cookies
- HTTP headers
- Database queries

User input gets passed into an application from these sources through methods that are grouped together into classes. For example, consider the use of request objects in a Web application. A request object retrieves the values the client browser passes to the server during an HTTP request such as headers, cookies or parameters associated with the request.

2. Use of Unvalidated Input

Web applications are designed to execute tasks based on a request delivered by the client browser, including accessing files or databases, invoking a new program, or initiating a program action. Tasks are executed by passing a user request from the client browser to a resource controlled by the server-side application. Problems may occur when the unvalidated user request is passed into an application, introducing an opportunity for an attacker to trick the application into doing something for which it was not intended. Input must be validated before allowing it to execute an operation.

3. Unvalidated Output Streams

Many Web applications are designed to generate dynamic content based on a specific user request. Dynamic content is generated at the server and contains both text and HTML markup. If the output has not been validated properly, there is a risk that the server could be tricked to insert malicious code hidden within the dynamic Web content returned to the client browser in the output stream. There are several methods used to generate output based on a user request executed by the server. To prevent malicious content from being passed back to the user, it is imperative to review these output streams to ensure content has been validated and encoded to protect the user.

4. Flawed Authorization and Access Control

The improper use of access control (also known as authorization) mechanisms can allow attackers to have unauthorized access to data and services and gain privileges to manipulate content or perform functions not available to normal users. Secure programming practices recommend access controls be defined in a formal policy and applied consistently throughout the application. It is important to note authorization and access controls may be applied in many locations, and their sensitivity should be managed in each. For example database access controls exist in conjunction with the applications that access the database as well as within the database management system. Weak authorization controls in either environment can result in vulnerabilities to both.

5. Flawed Authorization and Session Management

Insecure authorization and session management can expose account credentials and session tokens to compromise by an attacker. Flawed implementations include the use of weak credentials for authentication, exposed or unencrypted credentials during login, and failure to change session ID after login. If compromised, an attacker can circumvent authentication restrictions and assume another user's identity.

6. Native Code and Buffer Overflows

Web applications may invoke native methods, libraries or drivers that are written in C and C++, introducing security risks that would otherwise not be present in Web code. Native code is not protected by the built-in security model unique to Web application languages. This means native methods may allow for untrusted, malicious code to access local system resources, either by providing access to new resources and failing to secure them properly or by bypassing existing security checks.

To mitigate the security risk, methods or libraries that indicate the use of native code throughout the program must be identified. All input passed to these calls should be validated for content to prevent an attacker from injecting malicious commands into the application. Similarly, the length of input should be limited to mitigate the risk of a buffer overflow.

Native code is particularly vulnerable to buffer overflow attacks. In order to ensure the application does not pass string parameters longer than the maximum allowable string length, input passed to native code must be checked for both content and length to prevent a malicious or unintentional buffer overflow.

While some operating systems provide exploit mitigation techniques, it is best to avoid flaws in the programming environment and regard the execution environment controls as complementary, compensating, or even redundant.

7. Dynamic Code

Web application languages provide specific methods that dictate whether or not a program can load dynamic libraries, which are necessary for invoking native methods. If untrusted code passed into the application is allowed to load a dynamic library, then that code could maliciously invoke native methods that expose the system to a security risk. Calls that enable the use of dynamic libraries should be reviewed to determine whether they are appropriate and to ensure proper validation is performed to prevent execution of malicious code.

8. Weak Encryption

Web applications frequently use cryptography to protect confidential data and credentials necessary to gain access to this data. Cryptography is difficult to use correctly and poor implementation often results in weak protection. Based on current standards, encryption keys should be at least 128 bits long. The key should be generated using a strong random number generator, or another commercial or open-source cryptographic library.

9. Application Configuration

Application configuration details, including property files, XML data, and other storage information, must be protected. Access to these details can be used by an attacker to exploit the application, so they must be securely stored.

10. Denial of Service

Denial of service attacks cause a Web application to fail by causing the application to shut down unintentionally or by consuming the available resources so legitimate users can no longer access the application. Extraneous exit calls also expose the application to denial of service risks. It is

critical to identify calls that could cause the application to shut down if an attacker gains unauthorized access including System.exit.

11. Network Communications

Web applications support a range of network communication interfaces, including CORBA, servlets, email, remote method invocation (RMI) and socket communication. These interfaces could enable an attacker to gain unintended access to vulnerable applications or to eavesdrop on application communications. All network interfaces should be examined to ensure proper authentication occurs, content is properly encrypted and all input is carefully validated.

12. Unsupported Application Interfaces

Web applications use a variety of lower level application interfaces, as part of their core packages. These lower level application interfaces are not intended to be called directly by the application. Applications that make direct calls to these internal interfaces should be investigated to ensure the calls are necessary and adequately protected. A security risk may result if these interfaces are left unsupported in a publicly accessible program.

13. Improper Administrative and Exception Handling

Improper error messages can provide critical information about an application which may aid an attacker in exploiting the application. The most common problem occurs when detailed error codes are displayed to the user. Security analysts view logging and error handling as potential areas of risk that must be considered as part of a security review. Calls should be reviewed to determine whether the appropriate details are displayed to the user.

Similarly, logging is a critical security safeguard. All errors, exceptions, and relevant business and security events should be logged in order to detect and determine the events that lead up to an attack. Best practices suggest logging should be implemented using a centralized logging system to ensure all relevant information is captured system-wide in a common format and stored in a central location.

APPENDIX E: REFERENCES

1. Bibliography and Web References

1. Basel II: Revised international capital framework – Basel Committee on Banking Supervision, Bank for International Settlements, <http://www.bis.org/publ/bcbsca.htm>
2. BITS Framework: Managing Technology Risk for Information Technology (IT) Service Provider Relationships – Financial Services Roundtable (FSR), <http://www.bitsinfo.org>
3. BS 7799 – Parts 1 & 2, Code of Practice for Information Security Management (British Standards Institute), <http://www.bsi.org.uk>
4. Building Web Application Security into Your Development Process, by Kevin Heineman, SPI Dynamics, Inc. <http://www.spidynamics.com>
5. CA SB 1386 (the “You’ve Been Hacked” Act), http://info.sen.ca.gov/pub/01-02/bill/sen/sb_1351-1400/sb_1386_bill_20020926_chaptered.html
6. Change and Patch Management Controls: Critical for Organizational Success, Global Technology Audit Guide, The Institute of Internal Auditors, Inc. http://www.theiia.org/index.cfm?doc_id=4706
7. CISSP and SSCP Open Study Guides web site, <http://www.cccure.org>
8. COBIT – Control Objectives for Information and Related Technologies (ISACA), <http://www.isaca.org>
9. Common Criteria, <http://www.commoncriteriaportal.org>
10. Consensus Benchmark Scoring Tools, <http://www.cisecurity.org>
11. The Corporate and Auditing Accountability, Responsibility, and Transparency Act of 2002, Public Law 107-204 – 107th Congress, the “Sarbanes-Oxley Act of 2002”. http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ204.107.pdf
12. Corporate Information Security Working Group, Best Practices and Metrics Team, report to the U.S. House of Representatives, Technology Subcommittee, November 17, 2004, <http://www.educause.edu/ir/library/pdf/CSD3661.pdf>
13. The Dirty Dozen: The Top Web Application Vulnerabilities and How to Hunt Them Down at the Source, Ounce Labs, Inc. <http://www.ouncelabs.com>
14. EU Data Protection Directive - Part 1 & Part 2 available in separate PDFs, <http://aspe.os.dhhs.gov/datacncl/eudirect.htm>, http://europa.eu.int/comm/internal_market/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf, http://europa.eu.int/comm/internal_market/privacy/docs/95-46-ce/dir1995-46_part2_en.pdf
15. Federal Financial Institutions Examination Council (FFIEC) - FFIEC “Audit IT Examination Handbook,” and “FFIEC Audit Examination Procedures”, <http://www.ffiec.gov>
16. Federal Information Security Management Act of 2002 (FISMA) – U.S. Congress, 2002, <http://www.fedcirc.gov/library/legislation/FISMA.html>
17. Federal Sentencing Guidelines (US), <http://www.ussc.gov/GUIDELIN.HTM>
18. GAISP – Generally Accepted Information Security Principles, Currently available: Generally Accepted Systems Security Principles (GASSP) consisting of Pervasive Principles (PP), & Broad Functional Principle (BFP), June, 1999, <http://www.issa.org/gaisp.html>
19. GAPP – “Generally Accepted Principles and Practices” NIST SP 800-18, “Guide for Developing Security Plans for Information Technology Systems” December 1998 (Marianne Swanson & Barbara Guttman), <http://csrc.nist.gov/publications/nistpubs/index.html>

20. A Guide to Building Secure Web Applications, The Open Web Application Security Project (OWASP) <http://www.owasp.org>
21. Gramm, Leach, Bliley Act (GLBA) – The Financial Modernization Act of 1999, <http://www.ftc.gov/privacy/glbact/>
22. Health Information Portability and Accountability Act – HIPAA, <http://www.hhs.gov/ocr/hipaa>
23. ICAT Metabase of Common Vulnerabilities and Exposures – National Institute of Standards and Technology (NIST) http://icat.nist.gov/icat_documentation.htm Changed July 2005 to the National Vulnerability Database. See: <http://nvd.nist.gov>
24. Improving Security Across the Software Development Lifecycle, National Cyber Security Partnership, <http://www.cyberpartnership.org/SDLCFULL.pdf>
25. Information Assurance Technical Framework, Information Assurance Task Force (IATF) National Security Agency Outreach, http://www.iatf.net/framework_docs/version-3_1/index.cfm
26. Information Security Governance: Guidance for Boards of Directors and Executive Management”, 2001 – IT Governance Institute, <http://www.itgi.org>
27. Information Security Management and Assurance: A Call to Action for Corporate Governance, The Institute of Internal Auditors, Inc., April 2000, Part 1 of a 3 volume set of board and executive level guidance on information security and what the leaders are doing about it. Appendix A of this guide is a board-level description of effective risk management practices featuring quantitative analysis. http://www.theiia.org/index.cfm?doc_id=3061
28. Information Security Oversight: Essential Board Practices, National Association of Corporate Directors, (NACD), <http://www.nacdonline.org/publications/pubDetails.asp?pubID=138&user=6158BBEB9D7C4EE0B9E4B98B601E3716>
29. Information Security Program Elements and Supporting Metrics (sections V-VIII of the Corporate Information Security Working Group, Best Practices and Metrics Team, report to the U.S. House of Representatives, Technology Subcommittee, November 17, 2004) http://www.educause.edu/content.asp?page_id=666&ID=CSD3661&bhcp=1
30. The Information Technology Baseline Protection Manual, Federal Office for Information Security (BSI) Germany, <http://www.bsi.bund.de/english/publications/index.htm>
31. Information Technology Controls, Global Technology Audit Guide, The Institute of Internal Auditors, Inc. http://www.theiia.org/index.cfm?doc_id=4706
32. Information Technology Security Evaluation Criteria (ITSEC) - Harmonised Criteria of France, Germany, the Netherlands, the United Kingdom, Printed and published by the Department of Trade and Industry, London
33. IFAC International Guidelines on Information Technology Management—Managing Information Technology Planning for Business Impact: International Federation of Accountants, New York, 1999, <http://www.ifac.org>
34. International Standards for the Professional Practice of Internal Auditing, The Institute of Internal Auditors, Inc., http://www.theiia.org/index.cfm?doc_id=124
35. ISO 17799 – IT – Code of Practice for Information Security Management, <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=33441&ICS1=35&ICS2=40&ICS3>
36. National Vulnerability Database - National Institute of Standards and Technology (NIST) <http://nvd.nist.gov>
37. NIST 800-14 Generally Accepted Principles and Practices for Securing IT Systems, 1996, <http://csrc.nist.gov/publications/nistpubs/index.html>

38. NIST 800-27 Engineering Principles for IT Security, <http://csrc.nist.gov/publications/nistpubs/index.html>
39. NIST 800-53 - Recommended Security Controls for Federal Info Systems, <http://csrc.nist.gov/publications/nistpubs/index.html>
40. NoticeBored—Information security awareness content service, <http://www.noticebored.com>
41. OpenSourceTesting.org, “Open source tools for software testing professionals. <http://opensourcetesting.org>
42. Open Web Application Security Project (OWASP), OWASP Guide to Building Secure Web Applications, http://www.owasp.org/documentation/guide/guide_about.html
43. The Organization for Economic Cooperation and Development, OECD Guidelines for the Security of Information Systems and Networks (9 pervasive principles for information security upon which several other guides are based.) http://www.oecd.org/document/42/0,2340,en_2649_33703_15582250_1_1_1_1,00.html
44. Personal Information Protection and Electronic Documents Act (PIPEDA), Canada, http://www.privcom.gc.ca/legislation/02_06_01_01_e.asp
45. Policy statement regarding implementation of auditing standard No. 2, an audit of internal control Over financial reporting performed in Conjunction with an audit of financial Statements, PCAOB Release No. 2005-009, May 16, 2005
http://www.pcaob.com/Standards/Standards_and_Related_Rules/PCAOB%20Release%20No.%202005-009%20-%20AS2%20Policy%20Statement%20-%20May%2016,%202005.pdf
46. Processes to Produce Secure Software, National Cyber Security Partnership, <http://www.cyberpartnership.org/Software%20Pro.pdf>
47. Security at the Next Level – Are your web applications vulnerable, by Caleb Sima, SPI Dynamics, Inc. <http://www.spidynamics.com>
48. Seven Steps to Security Awareness, Gary Hinson, <http://www.noticebored.com>
49. Staff Statement on Management’s Report on Internal Control Over Financial Reporting, U.S. Securities and Exchange Commission, May 16, 2005, <http://sec.gov/info/accountants/stafficreporting.pdf>
50. Standard of Good Practice for Information Security (Information Security Forum), http://www.isfsecuritystandard.com/index_ie.htm
51. The Ten Most Critical Web Application Security Vulnerabilities, 2004 Update, The Open Web Application Security Project (OWASP) <http://www.owasp.org>.
52. Tescom, “The Global Software Assurance Company” <http://www.tescom.co.il>
53. Trusted Computer System Evaluation Criteria (TCSEC), U.S Department of Defense, <http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html>
54. Trust Services Criteria; including SysTrust/WebTrust (AICPA), <http://www.aicpa.org/trustservices>
55. The Visible Ops Handbook, Information Technology Process Institute, <http://www.itpi.org>

2. Organizations:

AICPA – The American Institute of Certified Public Accountants, www.aicpa.org

ANSI – American National Standards Institute, www.ansi.org

ASBDC-US – The Association of Small Business Development Centers, www.asbdc-us.org

BITS - The Technology Group for The Financial Services Roundtable, www.bitsinfo.org

BR – Business Roundtable, www.businessroundtable.org

BSA – Business Software Alliance, www.bsa.org/usa

BSI – British Standards Institute, www.bsi.org.uk

BSI - Bundesamt für Sicherheit in der Informationstechnik, Federal Office for Information Security (BSI) Germany, www.bsi.bund.de

CERT – Computer Emergency Response Team, www.cert.org

CIAO – Critical Infrastructure Assurance Office (formerly U.S. Dept. of Commerce, now Information Analysis and Infrastructure Protection of the Department of Homeland Security)

CICA – Canadian Institute of Chartered Accountants www.cica.ca

CIS – The Center for Internet Security, www.cisecurity.org

CMU SEI – Carnegie Mellon University, www.sei.cmu.edu

COSO – Committee of Sponsoring Organizations for the Commission on Fraudulent Financial Reporting (Treadway Commission), www.coso.org

DHS – Department of Homeland Security, www.dhs.gov

DISA - Defense Information Systems Agency www.disa.mil

FFIEC – Federal Financial Institutions Examination Council (USA), www.ffiec.gov

FSR – Financial Services Roundtable, www.fsround.org

FTC - Federal Trade Commission (USA), www.ftc.gov

GAISPC – Generally Accepted Information Security Principles Committee, www.issa.org/gaisp.html

IAIP – Information Assurance and Infrastructure Protection Directorate of the U.S. Department of Homeland Security (DHS), www.dhs.gov

IATF – Information Assurance Task Force, National Security Agency Outreach, www.iatf.net

ICAEW – Institute of Chartered Accountants in England & Wales, www.icaew.co.uk

ICC – International Chamber of Commerce, www.iccwbo.org

IFAC – International Federation of Accountants, www.ifac.org

IIA – The Institute of Internal Auditors, Inc. (and IIA Research Foundation), www.TheIIA.org

ISECOM – The Institute for Security and Open Methodologies, <http://www.isecom.org>

ISA – Internet Security Alliance, www.isalliance.org

ISACA – The Information Systems Audit and Control Association, www.isaca.org

ISF – Information Security Forum, www.securityforum.org

ISO – International Organization for Standardization, www.iso.org

ISSA – Information Systems Security Association, www.issa.org

NACD – National Association of Corporate Directors, www.nacdonline.org

NCSA – National Cyber Security Alliance, www.staysafeonline.info

NCSP – National Cyber Security Partnership, www.cyberpartnership.org

NERC – North American Electric Reliability Council www.nerc.com

NIST – National Institute for Standards and Technology, www.nist.gov
NSA – National Security Agency, www.nsa.gov
NVD—National Vulnerability Database, NIST (replaced ICAT) <http://nvd.nist.gov>
OWASP – Open Web Application Security Project, <http://www.owasp.org>
OECD – Organization for Economic Cooperation and Development, www.oecd.org
PCAOB – Public Company Accounting Oversight Board, www.pcaobus.org
SANS – Systems Administration, Audit, and Network Security Institute, www.sans.org
SEC – Securities & Exchange Commission, www.sec.gov
SEI – Carnegie Mellon University Software Engineering Institute, www.sei.cmu.edu
SNAC – Systems and Network Attack Center (NSA), www.nsa.gov/snac
US-CERT – U.S. Computer Emergency Readiness Team, www.us-cert.gov
WB – World Bank, www.worldbank.org

APPENDIX F: CLOSELY RELATED ISSUES TO CONSIDER

A. You Just Cannot Say It Enough...

While some of these items are not software specific, they can open entry holes that impair software security.

1. Education

It is specifically relevant to emphasize the importance of educating everyone in the organization and others with access to the systems such as consultants and some supply-chain partners on their role in maintaining security. One of the biggest security holes tends to be in “social-engineering” and people not following the procedures in place.

2. Passwords

The importance of setting, and guidelines for establishing and changing, “secure” passwords should be a management priority item. This is an area often discussed but also often overlooked or ignored. The importance of resetting default passwords, access codes, etc. should be made explicit.

3. Separation of Duties

The importance of establishing and adhering to the division of duties should be stressed. Access should be limited to those who truly need it in the performance of their assigned duties. That access should not be shared even for convenience or some other “emergency” reason. Policies should be in place to appropriately deal with such situations.

4. Employee terminations

The importance of immediately denying access when someone leaves the organization should be noted. There is often a delay in this process or even a failure to follow-through. This creates a dangerous “window-of-opportunity” for corporate espionage or sabotage by a disgruntled former employee.

B. CA SB 1386

Attorneys have suggested the courts would not look kindly upon a company that treats its California customers differently from others just because of the California Acts modified by SB 1386. As more people become aware of the risks posed by security vulnerabilities in source code and the tools available to remediate them, the laggards in implementing this management tools will have less standing in court (or responding to a SOx compliance issue) by saying nobody else was using source code scanning. Further, the wording from NIST SP 800-53 does not say network scanning. It says “Using appropriate vulnerability scanning tools and techniques, the organization scans for vulnerabilities in the information system or when significant new vulnerabilities are identified and reported.”

C. Your Company Name Here

“A hacker who broke into the computer system of (your company name here?) earlier this year might have stolen employees’ personal data, including Social Security numbers and bank deposit information, the company said this week.”

D. New York Times - Monday, May 9, 2005

<http://www.nytimes.com/2005/05/10/technology/10cisco.html?pagewanted=1&th&emc=th>

Internet Attack Called Broad and Long Lasting by Investigators

... "Now federal officials and computer security investigators have acknowledged that the Cisco break-in last year was only part of a more extensive operation - involving a single intruder or a small band, apparently based in Europe - in which thousands of computer systems were similarly penetrated." ...

"Shortly after being stolen last May, a portion of the Cisco programming instructions appeared on a Russian Web site. With such information, sophisticated intruders would potentially be able to compromise security on router computers of Cisco customers running the affected programs." ...

"The intruder probed computers for vulnerabilities that allowed the installation of the corrupted program, known as a Trojan horse, in place of the legitimate program." ...

"In many cases the corrupted program is distributed from a single computer and shared by tens or hundreds of users at a computing site, effectively making it possible for someone unleashing it to reel in large numbers of log-ins and passwords as they are entered. Once passwords to the remote systems were obtained, an intruder could log in and use a variety of software "tool kits" to upgrade his privileges - known as gaining root access. That makes it possible to steal information and steal more passwords. The operation took advantage of the vulnerability of Internet-connected computers whose security software had not been brought up to date." ...

"Last May, the security investigators were able to install surveillance software on the University of Minnesota computer network when they discovered that an intruder was using it as a staging base for hundreds of Internet attacks. During a two-day period they watched as the intruder tried to break into more than 100 locations on the Internet and was successful in gaining root access to more than 50. When possible, they alerted organizations that were victims of attacks, which would then shut out the intruder and patch their systems.

"As the attacks were first noted in April 2004, a researcher at the University of California, Berkeley, found that her own computer had been invaded. The researcher, Wren Montgomery, began to receive taunting e-mail messages from someone going by the name Stakkato - now believed by the authorities to have been the primary intruder - who also boasted of breaking in to computers at military installations." ...

"Ms. Montgomery, a graduate student in geophysics, said that in a fit of anger, Stakkato had erased her computer file directory and had destroyed a year and a half of her e-mail stored on a university computer. She guessed that she might have provoked him by referring to him as a "quaint hacker" in a communication with system administrators, which he monitored. "It was inconvenient," she said of the loss of her e-mail, "and it's the thing that seems to happen when you have malicious teenage hackers running around with no sense of ethics.""

E. Definition: Software security vulnerability

From Wikipedia, the free encyclopedia:

In computer software, a security vulnerability is a software bug that can be used deliberately to violate security.

Such vulnerabilities are of significant interest when the program containing the vulnerability operates with special privileges, performs authentication or takes action on behalf of a user (such as a network server or RDBMS).

Well known vulnerabilities include (but are not limited to)

- stack smashing and other buffer overflows
- symlink races
- input validation errors, such as:
- format string bugs
- improperly handling shell metacharacters so they are interpreted
- SQL injection
- cross-site scripting (in web applications)
- directory traversal

See also: Exploit (computer science), computer security

Retrieved from "http://en.wikipedia.org/wiki/Software_security_vulnerability"

ACKNOWLEDGEMENTS. THE PROJECT TEAM

Any worthwhile effort in information security today involves a range of perspectives as well as specific knowledge in technical areas — too much for one person to accomplish alone. I am grateful for the counsel and advice provided by my friends and mentors as this guide was contemplated, planned, drafted, reviewed, and completed. A phone conversation, an email message, a word of encouragement, an insight to cultural differences, pointing out the absence of information or the need for further explanation; all were essential to producing the final guide.

Many people responded to the survey that substantiated many of the statements in the guide, and while I cannot identify them I greatly appreciate the concern and care they took to answer and explain their answers. You know who you are, and I thank you.

Advisory Council

- Carolee Birchall, Bank of Montreal, Canada
- Lawrence Capuder, Aramco, Saudi Arabia
- Gerardo Carstens, Mexico
- Richard E. Crawford, National Association for the Self-Employed, USA
- Jerry E. Durant, Certifiable Technologies, Ltd., USA
- Alexey Guriev, Corporate Internal Audit Head, SeverStal Group, Russia
- Ulrich Hahn, Switzerland
- Clint Kreitner, President/CEO, The Center for Internet Security, USA
- Alexandra Lajoux, National Association of Corporate Directors, USA
- Dr. Cynthia LeRouge, MS, CPA, Ph.D., St. Louis University, USA
- Warren E. Malmquist, Coors Brewing Company, USA
- Alan S. Oliphant, MAIR International, Scotland
- Will Ozier, OPA Inc., USA
- Frederick B. Palmer, Palmer Associates, USA
- Bernard K. Plagman, TechPar Group, USA
- Sri Ramamoorthy, Ernst & Young, USA
- Mark Salamasick, CIA, CISA, University of Texas at Dallas, USA
- Kyoko Shimizu – PwC, Japan

Reviewers of the Draft Guide:

- Dr. Denise Guithues Amrhein, Associate Professor of Accounting, Saint Louis University, USA
- Lawrence P. Brown, Vice President, Chief Audit Executive, The Options Clearing Corporation, USA
- Philip L. Campbell, Technical Staff, Sandia National Laboratories, USA
- Richard Cascarino, CIA, CISM, MBA, CEO of Compact Business Services, South Africa
- P.J. Corum, Managing Director, Quality Assurance Institute, Middle East and Africa, United Arab Emirates
- Jim Dillon, IT Audit Manager, University of Colorado System, USA
- J. Russell Gates, Dupage Consulting LLC, USA
- Howard Glavin, CPP - CISM – OPSEC, Internet Security Systems, USA
- Glen L. Gray, PhD, CPA, California State University, Northridge, USA
- Michael A. Gwynne, CA, CISA, CISSP, Audit Principal, IT Audit Services, Office of the Auditor General, Manitoba, Canada
- Richard Hefele, Internal Audit, Bank of Tokyo-Mitsubishi, USA
- Michael S. Hines, CIA, CISA, CFE, CDP, President, Administrative Business Consultants, Inc., USA
- Barry F. Jones, Tribridge, Inc., USA
- Susan Kennedy, MBA, CISA, CIW, Director, Information Technology Audit, University of Pennsylvania, USA
- John C. Lazarine, CISA, CIA, IT Audit Director, Raytheon, USA
- Michael B. Legary, CISSP, CISA, President, Seccuris, Inc., Canada
- Tom Le Grand, Convergys, USA
- Jagdish Pathak, MComm, PhD; Associate Professor of Accounting & Systems, University of Windsor, Canada
- Tom Patterson, CPA, CISA, Group Director, Corporate IT Audit, Delhaize Group, Belgium
- Sue Paulsen, Community College of Vermont, USA
- Christian S.J. Peron, Seccuris, Inc., Canada
- Kevin Smith, MCSE, Information Technology Manager, NuCO², USA
- Gib Sorebo, JD, CISSP, PMP, Senior Information Security Analyst, SAIC, USA
- George Spafford, President, Spafford Global Consulting, Inc., USA
- Brian Spindel, CPA CIA CISA GSEC, Senior Information Technology Auditor, WPS Insurance Corporation, USA
- Wyatt Starnes, Chairman & CEO, SignaCert, Inc., USA
- Dan Swanson, CIA, AVP Professional Practices, The Institute of Internal Auditors, Inc., USA
- Bill Swirsky, Vice President, Knowledge Development, Canadian Institute of Chartered Accountants, Canada
- Jason B. Taule CMC, CPCM, CISM, CHS-III, NSA-IAM, Director, Corporate Information Security, ViPS, USA
- William T. Tener, CIA, CISA, CISSP/ISSMP, Nevada System of Higher Education, USA
- Akitomo Yamamoto, The IIA, Japan

Survey Responses:

Survey responses came from professionals in a variety of industry and government positions representing the following countries:

- Argentina
- Australia
- Belgium
- Canada
- England
- Germany
- India
- Israel
- Japan
- Lebanon
- Malaysia
- Mexico
- Norway
- Pakistan
- Qatar
- Russia
- South Africa
- Spain
- Sultanate of Oman
- Sweden
- Switzerland
- United Arab Emirates
- United Kingdom
- United States of America

