

Secure Software Concepts

Designing secure software is based on the application of secure software design principles. These principles will be discussed in this chapter and form the fundamental basis for software assurance. Software assurance has been given many definitions, and it is important to understand the concept. The Software Security Assurance Report¹ defines *software assurance* as follows, “The basis for gaining justifiable confidence that software will consistently exhibit all properties required to ensure that the software, in operation, will continue to operate dependably despite the presence of sponsored (intentional) faults. In practical terms, such software must be able to resist most attacks, tolerate as many as possible of those attacks it cannot resist, and contain the damage and recover to a normal level of operation as soon as possible after any attacks it is unable to resist or tolerate.”

The U.S. Department of Defense (DoD) Software Assurance Initiative² defines software assurance as “the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software.”

The Data and Analysis Center for Software (DACs)³ requires that software must exhibit the following three properties to be considered secure:

- **Dependability**—Software that executes predictably and operates correctly under a variety of conditions, including when under attack or running on a malicious host.

¹Information Assurance Technology Analysis Center (IATAC), Data and Analysis Center for Software (DACs), *Software Security Assurance, State-of-the-Art Report (SOAR)*, July 31, 2007.

²Komaroff, M., and Baldwin, K., DoD Software Assurance Initiative, September 13, 2005 (<https://acc.dau.mil/CommunityBrowser.aspx?id=25749>).

³Goertzel, K., Winograd, T., et al., *Enhancing the Development Life Cycle to Produce Secure Software*, Draft Version 2.0. Rome, New York: United States Department of Defense Data and Analysis Center for Software, July 2008.



2 Chapter 1 ■ Secure Software Concepts

- **Trustworthiness**—Software that contains a minimum number or no vulnerabilities or weakness that could sabotage the software’s dependability. It must also be resistant to malicious logic.
- **Survivability (Resilience)**—Software that is resistant or tolerant of attacks and has the ability to recover as quickly as possible with as little harm as possible.

Chapter 1 explores the fundamentals of software assurance through basic design principles, risk management, supporting software architectures, legal issues, standards, acquisition methods, and information security models.

Seven complementary principles that support information assurance are confidentiality, integrity, availability, authentication, authorization, auditing, and accountability. These concepts are summarized in the following sections.

Confidentiality, Integrity, and Availability

Confidentiality, integrity, and availability are sometimes known as the C-I-A triad of information system security.

Confidentiality

Confidentiality refers to the prevention of intentional or unintentional unauthorized disclosure of information. Confidentiality in information systems is related to the areas of intellectual property rights, covert channels, traffic analysis, encryption, and inference.

- **Intellectual property rights**—*Intellectual property* (IP) includes inventions, designs, and artistic, musical, and literary works. Rights to intellectual property are covered by copyright laws, which protect creations of the mind, and patents, which are granted for new inventions.
- **Covert channels**—A *covert channel* is an unauthorized and unintended communication path that provides for exchange of information. Covert channels can be accomplished through timing of messages or inappropriate use of storage mechanisms.
- **Traffic analysis**—*Traffic analysis* is a form of confidentiality breach that can be accomplished by analyzing the volume, rate, source, and destination of message traffic, even if it is encrypted. Increased message activity and high bursts of traffic can indicate a major event is occurring. Countermeasures to traffic analysis include maintaining a near constant rate of message traffic and disguising the source and destination locations of the traffic.
- **Encryption**—*Encryption* involves scrambling messages so that they cannot be read by an unauthorized entity, even if they are intercepted. The amount of effort (*work factor*) required to decrypt the message is a function of the strength of the encryption key and robustness and quality of the encryption algorithm.



- **Inference**—*Inference* is usually associated with database security. Inference is the ability of an entity to use and correlate information protected at one level of security to uncover information that is protected at a higher security level.

Integrity

The concept of *integrity* requires that the following three principles are met:

- Modifications are not made to data by unauthorized personnel or processes.
- Unauthorized modifications are not made to data by authorized personnel or processes.
- The data is internally and externally consistent—in other words, the internal information is consistent among all sub-entities and that the internal information is consistent with the real-world, external situation.

Availability

Availability ensures the reliable and timely access to data or computing resources by the appropriate personnel. Availability guarantees that the systems are functioning properly when needed. In addition, this concept guarantees that the security services of the system are in working order. A denial-of-service attack is an example of a threat against availability.

The reverse of confidentiality, integrity, and availability is disclosure, alteration, and destruction (D-A-D).

Authentication, Authorization, Auditing, and Accountability

There are additional factors that directly affect information system and software assurance. These factors include authentication, authorization, auditing, and accountability, as summarized in the following sections.

Authentication

Authentication is the testing or reconciliation of evidence of a user's identity. It establishes the user's identity and ensures that users are who they claim to be. For example, a user presents an identity (user ID) to a computer login screen and then has to provide a password. The computer system authenticates the user by verifying that the password corresponds to the individual presenting the ID.

Authorization

Authorization refers to rights and privileges granted to an individual or process that enable access to computer resources and information assets. Once a user's identity

and authentication are established, authorization levels determine the extent of system rights a user can hold.

Auditing

To maintain operational assurance, organizations use two basic methods: system audits and monitoring.

- A *system audit* is a one-time or periodic event to evaluate security.
- *Monitoring* refers to an ongoing activity that examines either the system or the users, such as intrusion detection.

Information technology (IT) auditors are often divided into two types: internal and external. Internal auditors typically work for a given organization whereas external auditors do not. External auditors are often certified public accountants (CPAs) or other audit professionals who are hired to perform an independent audit of an organization's financial statements. Internal auditors, on the other hand, usually have a much broader mandate: checking for compliance and standards of due care, auditing operational cost efficiencies, and recommending the appropriate controls.

IT auditors typically audit the following functions:

- System and transaction controls
- Systems development standards
- Backup controls
- Data library procedures
- Data center security
- Contingency plans

In addition, IT auditors might recommend improvements to controls, and they often participate in a system's development process to help an organization avoid costly reengineering after the system's implementation.

An *audit trail* or *log* is a set of records that collectively provide documentary evidence of processing used to aid in tracing from original transactions forward to related records and reports, and/or backward from records and reports to their component source transactions. Audit trails may be limited to specific events, or they may encompass all of the activities on a system.

The audit logs should record the following:

- The transaction's date and time
- Who processed the transaction
- At which terminal the transaction was processed
- Various security events relating to the transaction

In addition, an auditor should examine the audit logs for the following:

- Amendments to production jobs
- Production job reruns

- Computer operator practices
- All commands directly initiated by the user
- All identification and authentication attempts
- Files and resources accessed

Accountability

Accountability is the ability to determine the actions and behaviors of a single individual within a system and to identify that particular individual. Audit trails and logs support accountability and can be used to conduct a postmortem study to analyze acts that previously occurred and the individuals or processes associated with those acts. Accountability is related to the concept of *nonrepudiation*, wherein an individual cannot successfully deny the performance of an action.

Security Design Principles

Historically, computer software was not written with security in mind. Because of the increasing frequency and sophistication of malicious attacks against information systems, modern software design methodologies include security as one of the primary objectives. With any system that seeks to meet multiple objectives such as cost, performance, reliability, maintainability, and security, trade-offs have to be made. A completely secure system will exhibit poor performance characteristics or might not function at all.

Technically competent hackers can usually find a way to break into a computer system, given enough time and resources. The goal is to have a system that is secure enough for everyday use while exhibiting reasonable performance and reliability characteristics.

In a 1974 paper⁴, Saltzer and Schroeder of the University of Virginia addressed the protection of information stored in a computer system by focusing on hardware and software issues that are necessary to support information protection. The paper presented the following 11 security design principles:

- Least privilege
- Separation of duties
- Defense in depth
- Fail safe
- Economy of mechanism
- Complete mediation
- Open design
- Least common mechanism

⁴Saltzer, J. H., and Schroeder, M. D., "The Protection of Information in Computer Systems," Fourth ACM Symposium on Operating Systems Principles, October 1974.

- Psychological acceptability
- Weakest link
- Leveraging existing components

The fundamental characteristics of these principles are summarized below.

Least Privilege

The principle of *least privilege* maintains that an individual, process, or other type of entity should be given the minimum privileges and resources for the minimum period of time required to complete a task. This approach eliminates capabilities that are not needed for the assigned task and, thus, reduces the opportunity for unauthorized access to sensitive information.

Separation of Duties

Separation of duties requires that completion of a specified sensitive activity or access to sensitive objects is dependent on the satisfaction of multiple conditions. For example, an authorization would require signatures of more than one individual or the arming of a weapon system would require two individuals with different keys. Thus, separation of duties forces collusion among entities in order to compromise the system.

Defense in Depth

Defense in depth is the application of multiple layers of protection wherein a subsequent layer will provide protection if a previous layer is breached.

The Information Assurance Technical Framework Forum (IATFF), an organization sponsored by the National Security Agency (NSA), has produced a document entitled the “Information Assurance Technical Framework” (IATF) that provides excellent guidance on the defense-in-depth concepts.

The IATFF encourages and supports technical interchanges on the topic of information assurance among U.S. industry, U.S. academic institutions, and U.S. government agencies. Information on the IATFF document can be found at the Web site, www.niap-ccs.org/cc-scheme/IATF_3.1-Chapter_03-ISSEP.pdf.

The IATF document 3.1⁵ stresses the importance of the *people* involved, the *operations* required, and the *technology* needed to provide information assurance and to meet the organization’s mission.

The defense-in-depth strategy as defined in IATF document 3.1 promotes application of the following information assurance principles:

- **Defense in multiple places**—Information protection mechanisms placed in a number of locations to protect against internal and external threats

⁵National Security Agency, “Information Assurance Technical Framework (IATF),” Release 3.1, September 2002.

- **Layered defenses**—A plurality of information protection and detection mechanisms employed so that an adversary or threat will have to negotiate multiple barriers to gain access to critical information
- **Security robustness**—An estimate of the robustness of information assurance elements based on the value of the information system component to be protected and the anticipated threats
- **Deploy KMI/PKI**—Use of robust key management infrastructures (KMI) and public key infrastructures (PKI)
- **Deploy intrusion detection systems**—Application of intrusion detection mechanisms to detect intrusions, evaluate information, examine results, and, if necessary, to take action

Fail Safe

Fail safe means that if a system fails it should fail to a state where the security of the system and its data are not compromised. One implementation of this philosophy would be to make a system default to a state where a user or process is denied access to the system. A complementary rule would be to ensure that when the system recovers it should recover to a secure state and not permit unauthorized access to sensitive information. This approach is based on using permissions instead of exclusions.

In the situation where system recovery is not done automatically, the failed system should permit access only by the system administrator and not by users, until security controls are reestablished.

Economy of Mechanism

Economy of mechanism promotes simple and comprehensible design and implementation of protection mechanisms, so that unintended access paths do not exist or can be readily identified and eliminated.

Complete Mediation

In *complete mediation*, every request by a subject to access an object in a computer system must undergo a valid and effective authorization procedure. This mediation must not be suspended or become capable of being bypassed, even when the information system is being initialized, undergoing shutdown, being restarted, or is in maintenance mode. Complete mediation entails the following:

1. Identification of the entity making the access request
2. Verification that the request has not changed since its initiation
3. Application of the appropriate authorization procedures
4. Reexamination of previously authorized requests by the same entity

Open Design

There has always been a continuing discussion on the merits and strength of security of designs that are kept secret versus designs that are open to scrutiny and evaluation by the community at large. A good example is an encryption system. Some feel that keeping the encryption algorithm secret makes it more difficult to break. The opposing philosophy believes that exposing the algorithm to review and study by experts at large while keeping the encryption key secret leads to a stronger algorithm because the experts have a higher probability of discovering weaknesses in the algorithm. In general, the latter approach has proven more effective, except in the case of organizations such as the National Security Agency (NSA), which employs some of the world's best cryptographers and mathematicians.

For most purposes, an open access control system design that has been evaluated and tested by a myriad of experts provides a more secure authentication method than one that has not been widely assessed. Security of such mechanisms depends on protecting passwords or keys.

Least Common Mechanism

This principle states that a minimum number of protection mechanisms should be common to multiple users, as shared access paths can be sources of unauthorized information exchange. Shared access paths that provide unintentional data transfers are known as *covert channels*. Thus, *least common mechanism* promotes the least possible sharing of common security mechanisms.

Psychological Acceptability

Psychological acceptability refers to the ease of use and intuitiveness of the user interface that controls and interacts with the access control mechanisms. The user must be able to understand the user interface and use it without having to interpret complex instructions.

Weakest Link

As in the old saying, "A chain is only as strong as its weakest link," the security of an information system is only as good as its weakest component. Thus, it is important to identify the weakest mechanisms in the security chain and layers of defense and improve them so that risks to the system are mitigated to an acceptable level.

Leveraging Existing Components

In many instances, the security mechanisms of an information system are not configured properly or used to their maximum capability. Reviewing the state and settings of the extant security mechanisms and ensuring that they are operating at their optimum design points will greatly improve the security posture of an information system.

Another approach that can be used to increase system security by leveraging existing components is to partition a system into defended subunits. Then, if a security mechanism is penetrated for one subunit, it will not affect the other subunits and damage to the computing resources will be minimized.

Risk Management

This section examines risk management for information system security and, after providing this base of knowledge, develops the concepts of software security risk management, which are more directly focused on application software risk and security.

Information System Risk Management

The U.S. National Institute of Standards and Technology (NIST) Special Publication (SP) 800-30 defines *risk management* as comprising three processes: risk assessment, risk mitigation, and evaluation. These three processes are summarized as follows:

- **Risk assessment**—Identification and evaluation of risks and risk impacts and recommendation of risk-reducing measures. It is the process that allows IT managers to balance the operational and economic costs of protective measures and obtain improvements in the security of a system.
- **Risk mitigation**—Prioritizing, implementing, and maintaining the appropriate risk-reducing measures recommended from the risk assessment process.
- **Evaluation**—A continuous appraisal process and application of key methods for implementing a successful risk management program. In this step, the system authorizing official is responsible for determining whether the remaining risk is at an acceptable level or whether additional security controls should be implemented to further reduce or minimize the residual risk.

The Risk Assessment Process

As defined in NIST SP 800-30, “Risk is a function of the likelihood of a given threat-source’s exercising a particular potential vulnerability, and the resulting impact of that adverse event on the organization.” Risk assessment comprises the following steps:

1. System characterization
2. Threat identification
3. Vulnerability identification
4. Control analysis
5. Likelihood determination
6. Impact analysis

10 Chapter 1 ■ Secure Software Concepts

7. Risk determination
8. Control recommendations
9. Results documentation

Each of these steps is summarized in the following sections.

System Characterization

NIST SP 800-30 describes and defines the scope of the risk assessment process. During this step, information about the system has to be gathered. This information, taken from SP 800-30, includes:

- Software
- Hardware
- Data
- Information
- System interfaces
- IT system users
- IT system support personnel
- System mission
- Criticality of the system and data
- System and data sensitivity
- Functional system requirements
- System security policies
- System security architecture
- Network topology
- Information storage protection
- System information flow
- Technical security controls
- Physical security environment
- Environmental security

This information can be obtained using questionnaires, on-site interviews, review of documents, and automated scanning tools. The outputs from this step are:

- Characterization of the assessed IT system
- Comprehension of the IT system environment
- Delineation of the system boundary

Threat Identification

This step identifies potential threat-sources and compiles a statement of the threat-sources that relate to the IT system under evaluation. A *threat* is defined in NIST SP 800-30 as “the potential for a threat-source to exercise (accidentally trigger or intentionally exploit) a specific vulnerability.” A *threat-source* is defined in the same document as “either (1) intent and method targeted at the intentional exploitation of a vulnerability or (2) a situation and method that may accidentally trigger a vulnerability.” Common threat-sources include *natural threats* such as storms and floods, *human threats* such as malicious attacks and unintentional acts, and *environmental threats* such as power failure and liquid leakage. A *vulnerability* is defined as “a flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security breach or a violation of the system’s security policy.”

Sources of threat information include the Federal Computer Incident Response Center (FedCIRC), intelligence agencies, mass media, and Web-based resources. The output from this step is a statement that provides a list of threat-sources that could exploit the system’s vulnerabilities.

Vulnerability Identification

This activity results in a list of system vulnerabilities that might be exploited by potential threat-sources. Vulnerabilities can be identified through vulnerability analyses, including information from previous information assessments, audit reports, the NIST vulnerability database (<http://icat.nist.gov/icat.cfm>), FedCIRC and DOE security bulletins, vendor data, commercial computer incident response teams, and system software security analyses. Testing of the IT system will also yield important results. This testing can be accomplished using penetration testing techniques, automated vulnerability scanning tools, and security test and evaluation (ST&E) procedures.

This phase also involves determining whether the security requirements identified during system characterization are being met. Usually, the security requirements are listed in a table with a corresponding statement about how the requirement is or is not being met. This *security requirements checklist* addresses management, operational, and technical information system security areas. Some useful references for this activity are:

- The Computer Security Act of 1987
- The Privacy Act of 1974
- The organization’s security policies
- Industry best practices
- Federal Information Processing Standard (FIPS) 200/NIST Special Publication 800-53, *Recommended Security Controls for Federal Information Systems, Rev.2*, December 2007
- NIST Special Publication 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems*, June 2008

12 Chapter 1 ■ Secure Software Concepts

NIST SP 800-53 and NIST SP 800-53A have superseded NIST SP 800-26, *Security Self-Assessment Guide for Information Technology Systems*.

The output from this step is a list of system vulnerabilities that could be exploited by the potential threat-sources.

Control Analysis

The control analysis step analyzes the controls that are in place or in the planning stage to minimize or eliminate the probability that a threat will exploit a vulnerability in the system.

Controls can be implemented through technical means such as computer hardware or software, encryption, intrusion detection mechanisms, and identification and authentication subsystems. Other controls such as security policies, administrative actions, and physical and environmental mechanisms are considered nontechnical controls. Both technical and nontechnical controls can further be classified as preventive or detective controls. As the names imply, *preventive* controls attempt to anticipate and stop attacks. Examples of preventive technical controls are encryption and authentication devices. *Detective* controls are used to discover attacks or events through such means as audit trails and intrusion detection systems.

Changes in the control mechanisms should be reflected in the security requirements checklist.

The output of this step is a list of current and planned control mechanisms for the IT system to reduce the likelihood that a vulnerability will be exercised and to reduce the impact of an attack or event.

Likelihood Determination

This activity develops a rating that provides an indication of the probability that a potential vulnerability might be exploited based on the defined threat environment. This rating takes into account the type of vulnerability, the capability and motivation of the threat-source, and the existence and effectiveness of information system security controls. The likelihood levels are given as high, medium, and low, as illustrated in Table 1-1.

Table 1-1: Likelihood Levels

LEVEL OF LIKELIHOOD	DEFINITION OF LIKELIHOOD
High	A highly motivated and capable threat-source and ineffective controls to prevent exploitation of the associated vulnerability
Medium	A highly motivated and capable threat-source and controls that might impede exploitation of the associated vulnerability
Low	Lack of motivation or capability in the threat-source or controls in place to prevent or significantly impede the exploitation of the associated vulnerability

Impact Analysis

If a threat does exploit a vulnerability in an IT system, it is critical to know the negative impact that would result to the system. Three important factors should be considered in calculating the negative impact:

- The mission of the system, including the processes implemented by the system
- The criticality of the system, determined by its value and the value of the data to the organization
- The sensitivity of the system and its data

The information necessary to conduct an impact analysis can be obtained from existing organizational documentation, including a *business impact analysis* (BIA), sometimes called a mission impact analysis report. This document uses either quantitative or qualitative means to determine the impacts caused by compromise or harm to the organization's information assets. An attack or adverse event can result in compromise or loss of information system confidentiality, integrity, and availability. As with the likelihood determination, the impact on the system can be qualitatively assessed as high, medium, or low, as shown in Table 1-2.

Table 1-2: Definitions of Likelihood

IMPACT MAGNITUDE	DEFINITION OF IMPACT
High	Possibility of costly loss of major tangible assets or resources; might cause significant harm or impedance to the mission of an organization; might cause significant harm to an organization's reputation or interest; might result in human death or injury
Medium	Possibility of costly loss of tangible assets or resources; might cause harm or impedance to the mission of an organization; might cause harm to an organization's reputation or interest; might result in human injury
Low	Possibility of loss of some tangible assets or resources; might noticeably affect an organization's mission; might noticeably affect an organization's reputation or interest

Qualitative analysis is more easily accomplished and provides identifiable areas for immediate improvement. However, it does not provide specific magnitudes of measures and thus makes a cost-benefit analysis difficult. *Quantitative analysis* does provide magnitudes of measurements but may take more time. It is sometimes very difficult or impossible to place quantitative values on abstract items such as reputation.

Other items that should be included in the impact analysis are the estimated frequency of the threat-source's exploitation of a vulnerability on an annual basis, the approximate cost of each of these occurrences, and a weight factor based on the relative impact of a specific threat exploiting a specific vulnerability.

The output of this step is the magnitude of impact: high, medium, or low.

Risk Determination

This step, the seventh step in the risk assessment process, determines the level of risk to the IT system. The risk is assigned for a threat/vulnerability pair and is a function of the following characteristics:

- The likelihood that a particular threat-source will exploit an existing IT system vulnerability
- The magnitude of the resulting impact of a threat-source successfully exploiting the IT system vulnerability
- The adequacy of the existing or planned information system security controls for eliminating or reducing the risk

Mission risk is calculated by multiplying the threat likelihood ratings (the probability that a threat will occur) by the impact of the threat realized. A useful tool for estimating risk in this manner is the risk-level matrix. An example risk-level matrix is shown in Table 1-3. In the table, a high likelihood that the threat will occur is given a value of 1.0; a medium likelihood is assigned a value of 0.5; and a low likelihood of occurrence is given a rating of 0.1. Similarly, a high impact level is assigned a value of 100, a medium impact level 50, and a low impact level 10.

Table 1-3: A Risk-Level Matrix Example

LIKELIHOOD OF THREAT	LOW IMPACT (10)	MEDIUM IMPACT (50)	HIGH IMPACT (100)
High (1.0)	Low $10 \times 1.0 = 10$	Medium $50 \times 1.0 = 50$	High $100 \times 1.0 = 100$
Medium (0.5)	Low $10 \times 0.5 = 5$	Medium $50 \times 0.5 = 25$	High $100 \times 0.5 = 50$
Low (0.1)	Low $10 \times 0.1 = 1$	Medium $50 \times 0.1 = 5$	High $100 \times 0.1 = 10$

Using the risk level as a basis, the next step is to determine the actions that senior management and other responsible individuals must take to mitigate estimated risk. General guidelines for each level of risk are:

- **High risk level**—At this level, there is a high level of concern and a strong need for a plan for corrective measures to be developed as soon as possible.
- **Medium risk level**—For medium risk, there is concern and a need for a plan for corrective measures to be developed within a reasonable period of time
- **Low risk level**—For low risk, the approving authority of the system must decide whether to accept the risk or implement corrective actions.

The output of the risk determination step is a risk level of high, medium, or low.

Control Recommendations

With the risks identified and general guidelines provided for risk mitigation in the previous step, this step specifies the controls to be applied for risk mitigation. In

order to specify appropriate controls, issues such as cost-benefit, operational impact, and feasibility have to be considered. Other factors such as applicable legislative regulations, organizational policy, safety, reliability, and the overall effectiveness of the recommended controls should also be taken into account.

The output of this step is a recommendation of controls and any alternative solutions to mitigate risk.

Results Documentation

The last step in the risk assessment process is the development of a risk assessment report. This report is directed at management and should contain information to support appropriate decisions on budget, policies, procedures, management, and operational issues.

The output of this step is a risk assessment report that describes threats and vulnerabilities, risk measurements, and recommendations for implementation of controls.

Risk Mitigation

Risk mitigation prioritizes, evaluates, and implements the controls that are an output of the risk assessment process. Risk mitigation is the second component of the risk management process.

Because risk can never be completely eliminated and control implementation must make sense under a cost-benefit analysis, a least-cost approach with minimal adverse impact on the IT system is usually taken.

Risk Mitigation Options

Risk mitigation can be classified into the following options:

- **Risk assumption**—Accept the risk and keep operating
- **Risk avoidance**—Forgo some functions
- **Risk limitation**—Implement controls to minimize the adverse impact of threats realized
- **Risk planning**—Develop a risk mitigation plan to prioritize, implement, and maintain controls
- **Research and development**—Research control types and options
- **Risk transference**—Transfer risk to other sources, such as purchasing insurance

NIST SP 800-30 emphasizes the following guidance on implementing controls: Address the greatest risks and strive for sufficient risk mitigation at the lowest cost, with minimal impact on other mission capabilities.

The control implementation approach from the risk mitigation methodology recommended by NIST SP 800-30 is given in Figure 1-1.

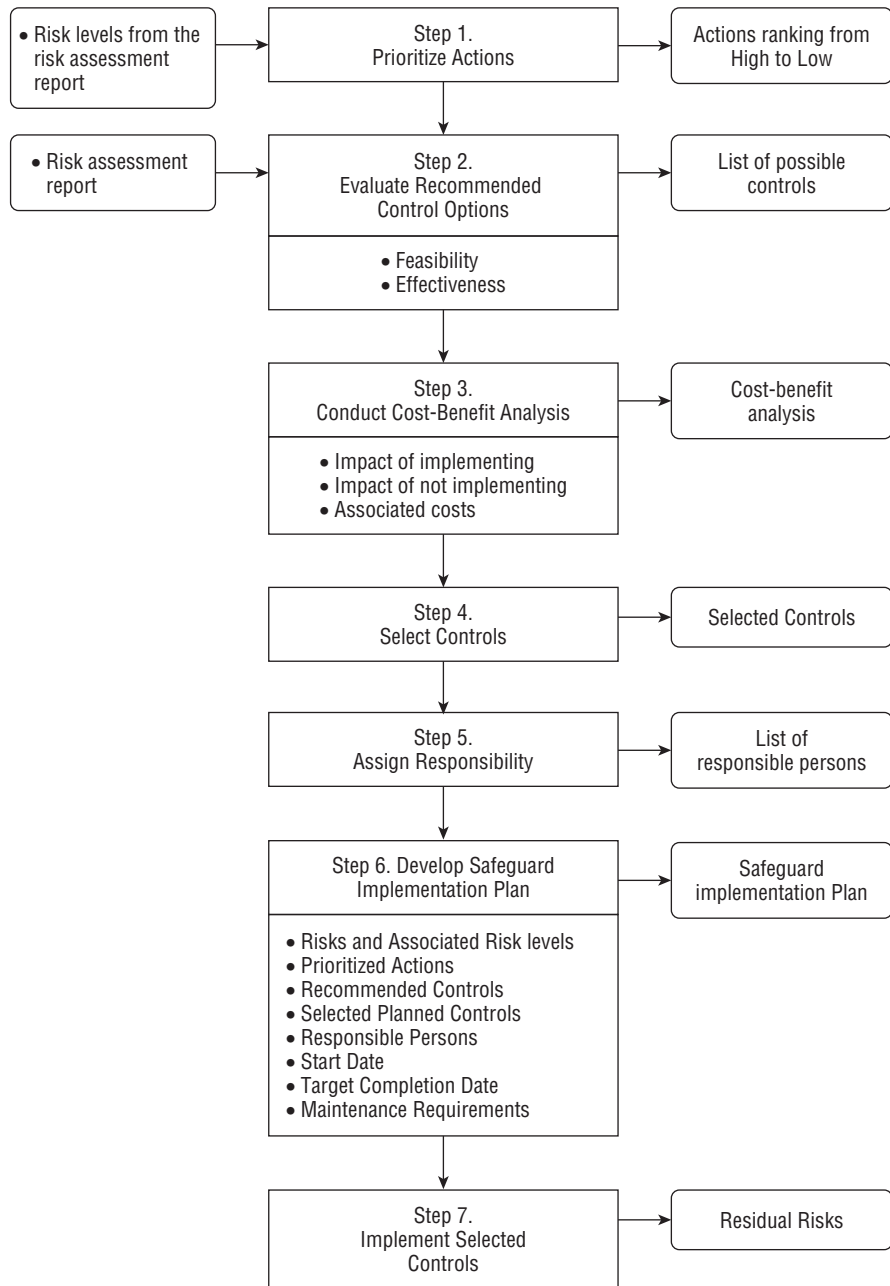


Figure 1-1: A control implementation approach

Figure source: from NIST SP 800-30

Categories of Controls

Controls to mitigate risks can be broken into the following categories:

- Technical
- Management
- Operational
- A combination of the above

Each of the categories of controls can be further decomposed into additional subcategories.

Technical controls can be subdivided into:

- **Supporting controls**—These controls implement identification, cryptographic key management, security administration, and system protections.
- **Preventive controls**—Preventive technical controls include authentication, authorization, access control enforcement, nonrepudiation, protected communications, and transaction privacy.
- **Detection and recovering controls**—These technical controls include audit, intrusion detection and containment, proof of wholeness (system integrity), restoration to a secure state, and virus detection and eradication.

Management controls comprise:

- **Preventive controls**—Preventive management controls include assigning responsibility for security, developing and maintaining security plans, personnel security controls, and security awareness and technical training.
- **Detection controls**—Detection controls involve background checks, personnel clearance, periodic review of security controls, periodic system audits, risk management, and authorization of IT systems to address and accept residual risk.
- **Recovery controls**—These controls provide continuity of support to develop, test, and maintain the continuity of the operations plan and establish an incident response capability.

Operational security controls are divided into preventive and detection types. Their functions are listed as follows:

- **Preventive controls**—These operational controls comprise control of media access and disposal, limiting external data distribution, control of software viruses, securing wiring closets, providing backup capability, protecting laptops and personal computers, protecting IT assets from fire damage, providing an emergency power source, and control of humidity and temperature.
- **Detection controls**—Detection operational controls include providing physical security through the use of items such as cameras and motion detectors and ensuring environmental security by using smoke detectors, sensors, and alarms.

Determination of Residual Risk

The risk that remains after the implementation of controls is called the *residual risk*. All systems will have residual risk because it is virtually impossible to completely eliminate risk to an IT system. An organization's senior management or the designated approving authority is responsible for authorizing/accrediting the IT system to begin or continue to operate. The authorization/accreditation must take place every three years in federal agencies or whenever major changes are made to the system. The approving authority signs a statement accepting the residual risk when accrediting the IT system for operation. If the approving authority determines that the residual risk is at an unacceptable level, the risk management cycle must be redone with the objective of lowering the residual risk to an acceptable level.

Figure 1-2 shows the relationship between residual risk and the implementation of controls.

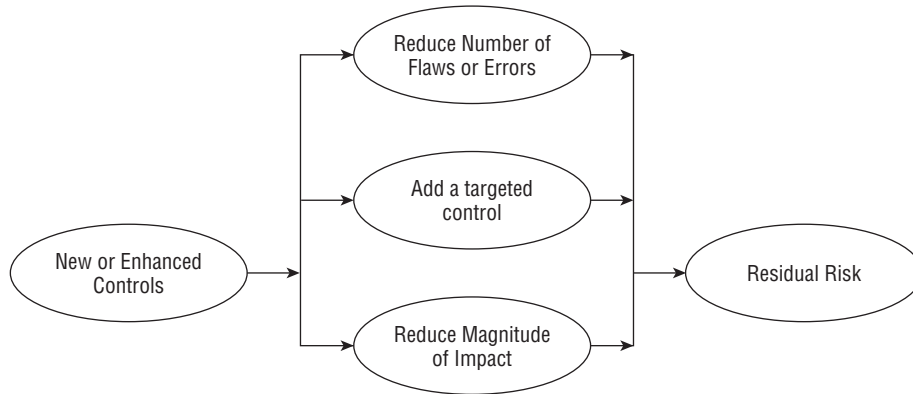


Figure 1-2: The relationship between residual risk and implementation of controls

Figure source: from NIST SP 800-30

Personnel Involved in the Risk Management Process

As with any enterprise endeavor, risk management must have the support of senior managers and the commitment of appropriate and qualified personnel. These personnel and their functions as outlined in NIST SP 800-30 are:

- **Senior management**—Provides the required resources and meets responsibilities under the principle of due care
- **Chief information officer (CIO)**—Considers risk management in IT planning, budgeting, and meeting system performance requirements
- **System and information owners**—Ensure that controls and services are implemented to address information system confidentiality, integrity, and availability
- **Business and functional managers**—Make trade-off decisions regarding business operations and IT procurement that affect information security

- **Information system security officer (ISSO)**—Participates in applying methodologies to identify, evaluate, and reduce risks to the mission-critical IT systems
- **IT security practitioners**—Ensure the correct implementation of IT system information system security requirements
- **Security awareness trainers**—Incorporate risk assessment in training programs for the organization’s personnel

Software Security Risk Management Concepts

A number of risk management concepts for the security of software have evolved from the IT approach to risk management. These methodologies include the Microsoft Security Risk Management Discipline (SRMD),⁶ vulnerability-oriented risk management proposed by Charles LeGrand,⁷ Morana⁸ risk management activities, and Cigital risk management⁹ methods. These approaches are summarized in the following sections.

Microsoft Security Risk Management Discipline (SRMD)

The SRMD comprises the following steps:

1. **Assessment**
 - a. **Asset assessment and valuation**—The value placed on an information asset and the cost to maintain or recover an asset value or the value of the asset to another entity
 - b. **Identifying security risks**—Acquisition of data on vulnerabilities, threats, and countermeasures and discovery of possible risks to security
 - c. **Analyzing and prioritizing security risks**—Identification of risks, determining the impact of potential threats, and making technical and economic decisions on the cost of countermeasures versus the impact of threats materialized
 - d. **Security risk tracking, planning, and scheduling**—Application of information derived from security risk analysis to plan and schedule mitigation approaches

⁶Microsoft Corporation, “Understanding the Security Risk Management Discipline,” revised May 31, 2006, Chapter 3 in *Securing Windows 2000 Server* (Redmond, WA: Microsoft Corporation, November 17, 2004). Available from: www.microsoft.com/technet/security/prodtech/windows2000/secwin2k/swin2k03.msp.

⁷LeGrand, Charles H. (CHL Global Associates), “Managing Software Risk: an Executive Call to Action” (Waltham, MA: Ounce Labs, September 21, 2005).

⁸Morana, Marco M. (Foundstone Professional Services), “Building Security into the Software Life Cycle: A Business Case,” paper presented at BlackHat USA, Las Vegas, NV, August 2–3, 2006.

⁹Viega, John and McGraw, Gary, *Building Secure Software: How to Avoid Security Problems the Right Way* (Boston: Addison-Wesley, 2001).

2. Development and implementation

- a. **Security remediation development**—Creation of policies and procedures addressing strategies for patch management, configuration management, auditing, monitoring, and operations using the results of the risk assessment phase
- b. **Security remediation testing**—Determination of the means to deploy the remediation strategies into an operational environment and evaluation of the countermeasures' ability to mitigate risk as planned
- c. **Capturing security knowledge**—Continuous determination of the methods used to acquire knowledge concerning the securing of information assets and to document known vulnerabilities and exploits

3. Operating

- a. **Reassessing new and changed assets and security risks**—Application of change management and security configuration management to modified and new assets and potential new risks
- b. **Stabilizing and deploying new or changed countermeasures**—The assessment of modified and new assets and the deployment of corresponding countermeasures on a day-to-day operational basis

LeGrand Vulnerability-Oriented Risk Management

Charles LeGrand has proposed a risk management method based on vulnerability analysis. The four principal steps in this approach are:

1. **Risk assessment**—Identification of vulnerabilities, estimation of possible losses caused by threats materialized, cost-benefit examination of countermeasures, and assessment of attacks
2. **Vulnerability management**—Identification, measurement, and remediation of specific vulnerabilities
3. **Adherence to security standards and policies for development and deployment**—Formal implementation methods to minimize or, possibly, eliminate the introduction of vulnerabilities
4. **Assessment, monitoring, and assurance**—Determination of required compliance provided by risk management practices and evaluation of risk levels to ensure they are within desired limits

Morana Risk Management Activities

Marco Morana postulates the following risk management procedure that maps onto the software development life cycle.

NOTE System and software development life cycles are covered in detail later in this chapter.

1. **Requirements**—Conducting security requirements engineering, incorporating appropriate standards, ensuring compliance requirements are met, performing vulnerability and threat analyses, determining technical security needs, and evaluating extant system security
2. **Architecture and design**—Performing security architecture and design reviews, modeling threat patterns, preparing security tests, and conducting measurements of security posture
3. **Development**—Analyzing code and eliminating/reducing coding errors, testing code units for security, revising threat models based on new information, and evaluating system security posture
4. **Testing**—Use of attack patterns, application of automated testing (including black box and white box testing)
 - *Black box* testing is also known as “zero knowledge” testing, where the testing team is provided no knowledge of the resources to be tested and has to acquire information on its own.
 - In *white box* or “full knowledge” testing, the testing team has as much knowledge as possible about the network and computing resources to be evaluated.Additional testing such as regression testing is also performed along with developing new threat models and conducting additional evaluations of the systems’ security posture.

Cigital Risk Management Method

Gary McGraw of Cigital and John Viega of Stonewall Software proposed the following software security risk management steps:

1. Security requirements derivation/elicitation and specification
2. Security risk assessment
3. Secure architecture and design
4. Secure implementation
5. Security testing
6. Security assurance

Risk Management and Assurance Activities in the SDLC

The term *SDLC* is used to refer to a number of different development life cycles. In developing systems, *SDLC* refers to the system development life cycle. When you are focusing on the software development portion of a system, *SDLC* represents the software development life cycle. However, in the context of developing secure software, *SDLC* or *SDL* stands for the security development life cycle.

SOFTWARE LIFE CYCLE (SLC) AND SYSTEM LIFE CYCLE

In the context of the CSSLP certification, the software life cycle (SLC) is also used. The SLC processes are defined in ISO/IEC Standard 12207, “Software Life Cycle Processes,” as a grouping of the following three classes:

- **Primary processes**—Principal elements of the SLC, which are acquisition, supply, development, operation, and maintenance.
- **Supporting processes**—Complementary items such as documentation, configuration management, quality assurance, joint review, audit, verification, validation, and problem resolution, which support other processes. A supporting process supports another process in performing a specialized function.
- **Organizational processes**—Components including management, infrastructure, process improvement, and training.

ISO/IEC 12207 defines the system life cycle as covering “the phases of needs determination and demonstration, development, production, use, and disposal or retirement.” The system life cycle is sometimes used interchangeably with the system development life cycle.

As discussed in this chapter, risk management provides the means to reduce the negative impact of threats on an information system. To be effective, risk management must not be an “add-on” process, but must be integrated into the system development life cycle. Because the concepts of the system development life cycle are applicable to software and form a basis for software security, this section addresses risk management in the system development life cycle. The software security development life cycle is covered later in this chapter.

System Development Life Cycle

The five phases of the SDLC as described in IATF document 3.1 are:

1. **Initiation**—Documentation of the need for the system and its mission. A sensitivity assessment, which evaluates the sensitivity of the IT system and the information to be processed, is included in this phase.
2. **Development/acquisition**—Comprises the system acquisition and development cycles. In this phase, the system is designed, developed, programmed, and acquired.
3. **Implementation**—Installation, testing, security testing, and accreditation are conducted.

4. **Operation/maintenance**—In this phase, the system performs its designed functions. Administration, security operations, modification/addition of hardware and/or software, operational assurance, monitoring, and audits are also conducted in this phase.
5. **Disposal**—This final phase involves the disposition of system components and products, such as hardware, software, and information; disk sanitization; archiving files; and moving equipment.

Incorporating Risk Management into the SDLC

The risk management processes of risk assessment, risk mitigation, and evaluation should be conducted in each phase of the SDLC. Table 1-4, from NIST SP 800-30, details the risk management activities that should be performed for each SDLC phase.

Incorporating Assurance in the SDLC

The Data and Analysis Center for Software¹⁰ defines 13 key elements of a secure SDLC process, summarized as follows:

1. Incorporation of security criteria in each SDLC phase checkpoint at the exit of the phase
2. Application of secure software principles and practices
3. Employment of adequate requirements
4. Use of secure coding
5. Integration of secure software
6. Performance of secure testing
7. Practicing secure distribution and deployment
8. Practicing secure sustainment and maintenance
9. Deployment of supportive development tools
10. Practicing secure configuration management
11. Employment of security-knowledgeable developers
12. Practicing secure project management and obtaining upper management commitment
13. Robust design and architecture

¹⁰Goertzel, K., Winograd, T. et al., "Enhancing the Development Life Cycle to Produce Secure Software," Draft Version 2.0. (Rome, New York: United States Department of Defense Data and Analysis Center for Software, July 2008).

Table 1-4: Risk Management in the SDLC

SDLC PHASE	DESCRIPTION	RISK MANAGEMENT ACTIVITIES
Phase 1— Initiation	The need for an IT system is expressed and the purpose and scope of the IT system is documented.	Identified risks are used to support the development of the system requirements, including security requirements, and a security concept of operations (strategy).
Phase 2— Development or acquisition	The IT system is designed, purchased, programmed, developed, or otherwise constructed.	The risks identified during this phase can be used to support the security analyses of the IT system that may lead to architecture and design trade-offs during system development.
Phase 3— Implementation	The system security features should be configured, enabled, tested, and verified.	The risk management process supports the assessment of the system implementation against its requirements and within its modeled operational environment. Decisions regarding risks identified must be made prior to system operation.
Phase 4— Operation or maintenance	The system performs its functions. Typically the system is being modified on an ongoing basis through the addition of hardware and software and by changes to organizational processes, policies, and procedures.	Risk management activities are performed for periodic system reauthorization (or reaccreditation) or whenever major changes are made to an IT system in its operational, production environment (for example, new system interfaces).
Phase 5— Disposal	This phase may involve the disposition of information, hardware, and software. Activities may include moving, archiving, discarding, or destroying information and sanitizing the hardware and software.	Risk management activities are performed for system components that will be disposed of or replaced to ensure that the hardware and software are properly disposed of, that residual data is appropriately handled, and that system migration is conducted in a secure and systematic manner.

NIST SP 800-30, "Risk Management Guide for Information Technology Systems," July 2002.

The following list, taken from NIST SP 800-64, summarizes the information system security steps to be applied to the SDLC. An organization will use the general SDLC described in SP 800-64 document or will have developed a tailored SDLC that meets its specific needs. In either case, NIST recommends that organizations incorporate the associated IT security steps of this general SDLC into their development process:

■ **Initiation phase:**

- **Security categorization**—Defines three levels (low, moderate, or high) of potential impact on organizations or individuals should there be a breach of security (a loss of confidentiality, integrity, or availability). Security categorization standards assist organizations in making the appropriate selection of security controls for their information systems.
- **Preliminary risk assessment**—Results in an initial description of the basic security needs of the system. A preliminary risk assessment should define the threat environment in which the system will operate.

■ **Acquisition/development phase:**

- **Risk assessment**—Analysis that identifies the protection requirements for the system through a formal risk assessment process. This analysis builds on the initial risk assessment performed during the Initiation phase, but will be more in depth and specific.
- **Security functional requirements analysis**—Analysis of requirements that may include the following components: (1) system security environment (that is, enterprise information security policy and enterprise security architecture) and (2) security functional requirements.
- **Assurance requirements analysis security**—Analysis of requirements that address the developmental activities required and assurance evidence needed to produce the desired level of confidence that the information security will work correctly and effectively. The analysis, based on legal and functional security requirements, will be used as the basis for determining how much and what kinds of assurance are required.
- **Cost considerations and reporting**—Determine how much of the development cost can be attributed to information security over the life cycle of the system. These costs include hardware, software, personnel, and training.
- **Security planning**—Ensures that agreed-upon security controls, planned or in place, are fully documented. The security plan also provides a complete characterization or description of the information system as well as attachments or references to key documents supporting the agency's information security program (for example, configuration management plan, contingency plan, incident response plan, security awareness and training plan, rules of behavior, risk assessment, security test and evaluation results, system interconnection agreements, security authorizations/accreditations, and plan of action and milestones).
- **Security control development**—Ensures that security controls described in the respective security plans are designed, developed, and implemented. For

information systems currently in operation, the security plans for those systems may call for the development of additional security controls to supplement the controls already in place or the modification of selected controls that are deemed to be less than effective.

- **Developmental security test and evaluation**—Ensure that security controls developed for a new information system are working properly and are effective. Some types of security controls (primarily those controls of a nontechnical nature) cannot be tested and evaluated until the information system is deployed—these controls are typically management and operational controls.
- **Other planning components**—Ensure that all necessary components of the development process are considered when incorporating security into the life cycle. These components include selection of the appropriate contract type, participation by all necessary functional groups within an organization, participation by the certifier and accreditor, and development and execution of necessary contracting plans and processes.
- **Implementation phase:**
 - **Inspection and acceptance**—Ensure that the organization validates and verifies that the functionality described in the specification is included in the deliverables.
 - **Security control integration**—Ensures that security controls are integrated at the operational site where the information system is to be deployed for operation. Security control settings and switches are enabled in accordance with vendor instructions and available security implementation guidance.
 - **Security certification**—Ensures that the controls are effectively implemented through established verification techniques and procedures. Gives organization officials confidence that the appropriate safeguards and countermeasures are in place to protect the organization's information system. Security certification also uncovers and describes the known vulnerabilities in the information system.
 - **Security accreditation**—Provides the necessary security authorization of an information system to process, store, or transmit information that is required. This authorization is granted by a senior organization official and is based on the verified effectiveness of security controls to some agreed-upon level of assurance and an identified residual risk to agency assets or operations.
- **Operations/maintenance phase:**
 - **Configuration management and control**—Ensure adequate consideration of the potential security impacts due to specific changes to an information system or its surrounding environment. Configuration management and configuration control procedures are critical to establishing an initial baseline of hardware, software, and firmware components for the information system and subsequently controlling and maintaining an accurate inventory of any changes to the system.

- **Continuous monitoring**—Ensures that controls continue to be effective in their application through periodic testing and evaluation. Security control monitoring (that is, verifying the continued effectiveness of those controls over time) and reporting the security status of the information system to appropriate agency officials are essential activities in a comprehensive information security program.
- **Disposition phase:**
 - **Information preservation**—Ensures that information is retained, as necessary, to conform to current legal requirements and to accommodate future technology changes that may render the retrieval method obsolete.
 - **Media sanitization**—Ensures that data is deleted, erased, and written over as necessary.
 - **Hardware and software disposal**—Ensures that hardware and software are disposed of as directed by the information system security officer.

After discussing these phases and the information security steps in detail, the guide provides specifications, tasks, and clauses that can be used in a request for proposal (RFP) to acquire information security features, procedures, and assurances.

The CSSLP candidate should also understand the relationship between the SDLC phases and the acquisition process for the corresponding information system. This relationship is illustrated in Table 1-5, also taken from NIST SP 800-64.

Table 1-5: Relationship Between Information Systems Acquisition Cycle Phases and the SDLC

	ACQUISITION	CYCLE	PHASES	
Mission and Business Planning	Acquisition Planning	Acquisition	Contract Performance	Disposal and Contract Close-Out
Initiation	Acquisition/Development	Implementation	Operation/Maintenance	Disposition
		SDLC	Phases	

NIST SP 800-64, "Security Considerations in the System Development Life Cycle," October 2008.

NIST SP 800-64 also defines the following acquisition-related terms:

- **Acquisition**—Includes all stages of the process of acquiring property or services, beginning with the process for determining the need for the property or services and ending with contract completion and closeout.
- **Acquisition initiator**—The key person who represents the program office in formulating information technology requirements and managing presolicitation activities.
- **Acquisition technical evaluation**—A component of the selection process, defined as the examination of proposals to determine technical acceptability and merit.

ACQUISITION SPIRAL MODEL

An additional, valuable tool in the acquisition process is the *spiral model of the acquisition management process*. This approach is known as an evolutionary acquisition strategy. This model depicts the acquisition management process as a set of phases and decision points in a circular representation. The model illustrates the concept that a mission need is defined and translated into a solution that undergoes a continuous circle of improvement and evolution until it is no longer required.

NIST SP 800-64 also lists the key personnel associated with system acquisition and development as follows:

- **Chief information officer (CIO)**—The CIO is responsible for the organization’s information system planning, budgeting, investment, performance, and acquisition. As such, the CIO provides advice and assistance to senior organization personnel in acquiring the most efficient and effective information system to fit the organization’s enterprise architecture.
- **Contracting officer**—The contracting officer is the person who has the authority to enter into, administer, or terminate contracts and make related determinations and findings.
- **Contracting officer’s technical representative (COTR)**—The COTR is a qualified employee appointed by the contracting officer to act as his or her technical representative in managing the technical aspects of a particular contract.
- **Information technology investment board (or equivalent)**—The information technology (IT) investment board, or its equivalent, is responsible for managing the capital planning and investment control process defined by the Clinger-Cohen Act of 1996 (Section 5).
- **Information security program manager**—The information security program manager is responsible for developing enterprise standards for information security. This individual plays a leading role in introducing an appropriate, structured methodology to help identify, evaluate, and minimize information security risks to the organization. Information security program managers coordinate and perform system risk analyses, analyze risk mitigation alternatives, and build the business case for the acquisition of appropriate security solutions that help ensure mission accomplishment in the face of real-world threats. They also support senior management in ensuring that security management activities are conducted as required to meet the organization’s needs.
- **Information system security officer**—The information system security officer is responsible for ensuring the security of an information system throughout its life cycle.
- **Program manager (owner of data)/acquisition initiator/program official**—This person represents programmatic interests during the acquisition process. The program manager, who has been involved in strategic planning initiatives of

the acquisition, plays an essential role in security and is, ideally, intimately aware of functional system requirements.

- **Privacy officer**—The privacy officer is responsible for ensuring that the services or system being procured meet existing privacy policies regarding protection, dissemination (information sharing and exchange), and information disclosure.
- **Legal advisor/contract attorney**—This individual is responsible for advising the team on legal issues during the acquisition process.

CSSLP candidates who are interested in additional information contained in NIST SP 800-64 can obtain the document from the NIST Web site: <http://csrc.nist.gov/publications/nistpubs/>.

Regulations, Privacy, and Compliance

In addition to focusing on assurance in software applications, organizations are required to comply with regulations and laws designed to provide protections in the consumer and financial arenas. Some of the important compliance requirements and acts are:

- The U.S. Federal Information Security Management Act (FISMA) has data security management requirements for United States federal government organizations.
- Privacy laws
- Sarbanes-Oxley (SOX) is the U.S. mandate that requires adequate controls to be in place to protect sensitive data and assets for publicly-traded companies.
- The Gramm-Leach-Bliley Act (GLB) Act includes provisions to protect consumers' financial information is mandatory for financial institutions.
- The Health Insurance Portability and Accountability Act (HIPAA) requires protection of personal health information.
- The Payment Card Industry Data Security Standard (PCI DSS) encompasses credit card transaction protection.

FISMA

In order to increase the security of federal information systems, the Federal Information Security Management Act (FISMA), which is Title III of the E-Government Act of December 2002 (Public Law 107-347), was passed. FISMA was enacted to:

- Provide a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support federal operations and assets
- Recognize the highly networked nature of the current federal computing environment and provide effective government-wide management and oversight of the related information security risks, including coordination of information

security efforts throughout the civilian, national security, and law enforcement communities

- Provide for development and maintenance of minimum controls required to protect federal information and information systems
- Provide a mechanism for improved oversight of federal agency information security programs

FISMA; the Paperwork Reduction Act (PRA) of 1980, as amended by the Paperwork Reduction Act of 1995 (44 U.S.C., Chapter 35); and the Clinger-Cohen Act (also known as “Information Technology Management Reform Act of 1996”) (Pub. L. 104-106, Division E) promote a risk-based policy for cost effective security. The Clinger-Cohen Act supplements the information resources management policies contained in the PRA by establishing a comprehensive approach for executive agencies to improve the acquisition and management of their information resources. FISMA also specifies that national security classified information should be handled in accordance with the appropriate national security directives as provided by DoD and NSA.

FISMA charges the director of the Office of Management and Budget (OMB) with the responsibility of overseeing the security policies and practices of all agencies of the executive branch of the federal government, including “coordinating the development of standards and guidelines between NIST and the NSA and other agencies with responsibility for national security systems.” Agencies of the executive branch of the U.S. government are defined as:

- An executive department specified in 5 U.S.C., Section 101
- Within the executive office of the president, only OMB and the Office of Administration
- A military department specified in 5 U.S.C., Section 102
- An independent establishment as defined in 5 U.S.C., Section 104(1)
- A wholly owned government corporation fully subject to the provisions of 31 U.S.C., Chapter 91

OMB Circular A-130, Appendix III, “Security of Federal Automated Information Resources,” specifies that federal government agencies perform the following functions:

- Plan for security
- Ensure that appropriate officials are assigned security responsibility
- Review the security controls in their information systems
- Authorize system processing prior to operations and periodically thereafter

OMB Circular A-130, Appendix III, also requires that each agency perform security accreditation, which is considered “a form of quality control and challenges managers and technical staffs at all levels to implement the most effective security controls possible in an information system, given mission requirements, technical constraints, operational constraints, and cost/schedule constraints. By accrediting an information

system, an agency official accepts responsibility for the security of the system and is fully accountable for any adverse impacts to the agency if a breach of security occurs.”

FISMA Performance Requirements

The actions that FISMA requires each government agency to perform in developing and implementing an agency-wide information security program are specified in NIST Special Publication 800-37, “Guide for the Security Certification and Accreditation of Federal Information Systems,” May 2004. FISMA specifies that the program must include:

- Periodic assessments of risk, including the magnitude of harm that could result from the unauthorized access, use, disclosure, disruption, modification, or destruction of information and information systems that support the operations and assets of the agency
- Policies and procedures that are based on risk assessments, cost-effectively reduce information security risks to an acceptable level, and ensure that information security is addressed throughout the life cycle of each agency information system
- Subordinate plans for providing adequate information security for networks, facilities, information systems, or groups of information systems, as appropriate
- Security awareness training to inform personnel (including contractors and other users of information systems that support the operations and assets of the agency) of the information security risks associated with their activities and their responsibilities in complying with agency policies and procedures designed to reduce these risks
- Periodic testing and evaluation of the effectiveness of information security policies, procedures, practices, and security controls to be performed with a frequency depending on risk, but no less than annually
- A process for planning, implementing, evaluating, and documenting remedial action to address any deficiencies in the information security policies, procedures, and practices, of the agency
- Procedures for detecting, reporting, and responding to security incidents
- Plans and procedures to ensure continuity of operations for information systems that support the operations and assets of the agency

Identification of Information Types

FISMA assigned to NIST the responsibility for developing the following information system-related standards and guidelines:

1. Standards to be used by all federal agencies to categorize all information and information systems collected or maintained by or on behalf of each agency based on the objectives of providing appropriate levels of information security according to a range of risk levels



2. Guidelines recommending the types of information and information systems to be included in each category
3. Minimum information security requirements (that is, management, operational, and technical controls)

In order to satisfy item 1, NIST developed FIPS Publication 199, “Standards for Security Categorization of Federal Information and Information Systems.” FIPS 199 and the recently developed FIPS 200 standard, entitled “Minimum Security Requirements for Federal Information and Information Systems,” are two mandatory standards specified in the FISMA legislation.

FIPS 199 is used to identify and categorize information and information systems and, as cited in the standard, should be used “[t]o provide a common framework and understanding for expressing security that, for the Federal government promotes: (i) effective management and oversight of information security programs, including the coordination of information security efforts throughout the civilian, national security, emergency preparedness, homeland security, and law enforcement communities; and (ii) consistent reporting to the Office of Management and Budget (OMB) and Congress on the adequacy and effectiveness of information security policies, procedures, and practices.”

Information Privacy and Privacy Laws

Privacy is the right of an individual to protection from unauthorized disclosure of the individual’s personally identifiable information (PII). For example, the Health Insurance Portability and Accountability Act (HIPAA) lists the following 16 items as a person’s individual identifiers:

- Names
- Postal address information, other than town or city, state, and zip code
- Telephone numbers
- Fax numbers
- Electronic mail addresses
- Social security numbers
- Medical record numbers
- Health plan beneficiary numbers
- Account numbers
- Certificate/license numbers
- Vehicle identifiers and serial numbers, including license plate numbers
- Device identifiers and serial numbers
- Web Universal Resource Locators (URLs)



- Internet Protocol (IP) address numbers
- Biometric identifiers, including finger- and voiceprints
- Full face photographic images and any comparable images

FUNDAMENTAL PRINCIPLES OF PRIVACY

An individual's right to privacy is embodied in the following fundamental principles of privacy:

- **Notice—Regarding collection, use and disclosure of PII**
- **Choice—To opt out or opt in regarding disclosure of PII to third parties**
- **Access—By consumers to their PII to permit review and correction of information**
- **Security—To protect PII from unauthorized disclosure**
- **Enforcement—Of applicable privacy policies and obligations**

Privacy Policy

Organizations develop and publish privacy policies that describe their approach to handling PII. Web sites of organizations usually have their privacy policies available to read online and these policies usually cover the following areas:

- Statement of the organization's commitment to privacy
- The type of information collected, such as names, addresses, credit card numbers, phone numbers, and so on
- Retaining and using email correspondence
- Information gathered through cookies and Web server logs and how that information is used
- How information is shared with affiliates and strategic partners
- Mechanisms to secure information transmissions, such as encryption and digital signatures
- Mechanisms to protect PII stored by the organization
- Procedures for review of the organization's compliance with the privacy policy
- Evaluation of information protection practices
- Means for the user to access and correct PII held by the organization
- Rules for disclosing PII to outside parties
- Providing PII that is legally required

Privacy-Related Legislation and Guidelines

The following list summarizes some important legislation and recommended guidelines for privacy:

- The Cable Communications Policy Act provides for discretionary use of PII by cable operators internally but imposes restrictions on disclosures to third parties.
- The Children’s Online Privacy Protection Act (COPPA) is aimed at providing protection to children under the age of 13.
- Customer Proprietary Network Information rules apply to telephone companies and restrict their use of customer information both internally and to third parties.
- The Financial Services Modernization Act (Gramm-Leach-Bliley) requires financial institutions to provide customers with clear descriptions of the institution’s policies and procedures for protecting the PII of customers.
- The Telephone Consumer Protection Act restricts communications between companies and consumers, such as in telemarketing.
- The 1973 U.S. Code of Fair Information Practices states that:
 1. There must not be personal data recordkeeping systems whose very existence is secret.
 2. There must be a way for a person to find out what information about them is in a record and how it is used.
 3. There must be a way for a person to prevent information about them, which was obtained for one purpose, from being used or made available for another purpose without their consent.
 4. Any organization creating, maintaining, using, or disseminating records of identifiable personal data must ensure the reliability of the data for their intended use and must take precautions to prevent misuses of that data.
- The Health Insurance Portability and Accountability Act (HIPAA), Administrative Simplification Title, includes Privacy and Security Rules and standards for electronic transactions and code sets.

Sarbanes-Oxley

The Sarbanes-Oxley Act was enacted in July of 2002 to regulate corporate financial practices and reporting. The act has 11 titles and includes deadlines for compliance. The titles include sections, where Sections 302, 401, 404, 409, 802, and 906 are considered the most important relative to compliance. The 11 titles are:

1. Establishment of the Public Company Accounting Oversight Board (PCAOB)
2. Standards for External Auditor Independence
3. Corporate Responsibility by Senior Executives for Accuracy and Completeness of Corporate Financial Reports

4. Enhanced Financial Disclosures and Reporting
5. Code of Conduct for Securities Analysts and Disclosure of Analyst Conflicts of Interest
6. Commission Resources and Authority that Defines Practices for Securities Analysts and the Authority to Censure Securities Professionals.
7. Studies and Reports for Conducting Research Relevant to Enforcement of Violations by the Securities and Exchange Commission (SEC) Registered Companies
8. Corporate and Criminal Fraud Accountability, which Describes Criminal Penalties for Fraud
9. White Collar Crime Penalty Enhancement, which Recommends Stronger Sentencing Guidelines for White Collar Crimes
10. Corporate Tax Returns Specifies that the Corporate CEO Sign the Company Tax Return.
11. Corporate Fraud Accountability Established Tampering with Records and Committing Fraud as Criminal Offenses.

Gramm-Leach-Bliley

The Gramm-Leach-Bliley Act, “Financial Services Modernization Act,” Public Law 106-102, was passed in 1999. It comprises seven titles as follows:

1. **Facilitating Affiliation among Banks, Securities Firms, and Insurance Companies**—Permits banks to affiliate with securities firms and to have financial subsidiaries
2. **Functional Regulation**—Modifies federal securities laws to include functional regulation of bank securities activities
3. **Insurance**—Includes functional regulation of insurance activities of banks and bank subsidiaries
4. **Unitary Savings and Loan Holding Companies**—Unitary thrift holding companies may be sold only to financial organizations
5. **Privacy**—All financial organizations must provide clear disclosure of their privacy policies regarding sharing of PII
6. **Federal Home Loan Bank System Modernization**—Establishes rules of operation for federal home loan banks
7. **Other Provisions**—Covers a number of other items including ATM fees, foreign banks establishing units in the United States, and the administration of Community Reinvestment Act (CRA) loan agreements

HIPAA

The Health Insurance Portability and Accountability Act (HIPAA) addresses the security and privacy of protected health information (PHI). Covered entities under

HIPAA include health plans, health care providers, health care clearinghouses, and any employer that stores, manages, or communicates protected health information. HIPAA addresses the protection of health information through two specific rules: the Final Security Rule and the Final Privacy Rule.

HIPAA Final Security Rule

The HIPAA Final Security Rule was adopted in February 2003 and specified compliance by April 21, 2005, and April 21, 2006 for small health plans. The Final Security Rule was designed to be compatible with the HIPAA Final Privacy Rule and is structured into four overlapping categories:

- Protection for transmitted data
- Protection for data at rest
- Physical protection
- Administrative procedures

The standards in the Final Security Rule fall into the categories of administrative, technical, and physical safeguards. The Final Security Rule requires the covered entity to:

- Protect against any reasonably anticipated threats or hazards to the security of protected health information (PHI)
- Protect the confidentiality, integrity, and availability of all electronic PHI that is created, transmitted, received, or maintained by the covered entity
- Protect against any reasonably anticipated uses or disclosures of (PHI) that are not permitted under HIPAA
- Ensure that the employees of the covered entity are trained in and comply with the Final Security Rule

The Final Security Rule also requires covered entities to enter into agreements with business associates with whom they exchange electronic information. In these agreements, the business associates must be willing and able to safeguard the PHI.

HIPAA Final Privacy Rule

Under the Final Privacy Rule, a covered entity may use or disclose protected health information for its own treatment, payment, or health care operations. Additional provisions of the Final Privacy Rule are:

- A covered entity must obtain an individual's prior written authorization to use his or her protected health information for marketing purposes except for a face-to-face encounter or a communication involving a promotional gift of nominal value.
- A covered entity is prohibited from selling lists of patients and enrollees to third parties or from disclosing protected health information to a third party for the marketing activities of the third party, without the individual's authorization.

- Covered entities must provide patients with notice of the patient's privacy rights and the privacy practices of the covered entity.
- Direct treatment providers must make a good faith effort to obtain patient's written acknowledgement of the notice of privacy rights and practices. (The rule does not prescribe a form of written acknowledgement; the patient may sign a separate sheet or initial a cover sheet of the notice.)
- Covered entities may disclose protected health information, without authorization, to a person subject to the jurisdiction of the FDA for public health purposes related to the quality, safety, or effectiveness of FDA-regulated products or activities (collecting or reporting adverse events, dangerous products, and defects or problems with FDA-regulated products.)
- State law, or other applicable law, governs in the area of parents and minors.
- A covered entity is permitted to disclose PHI to a business associate who performs a function or activity on behalf of the covered entity that involves the creation, use, or disclosure of PHI, provided that the covered entity enters in a contractual relationship with the business associate detailing specific safeguards.
- A researchers' use of a single combined form to obtain informed consent for research and authorization to use or disclose protected health information for such research is permitted.
- The creation and dissemination of a limited data set (that does not include directly identifiable information) for research, public health, and health care operations is permitted.

PCI Data Security Standard

The Payment Card Industry (PCI) group was formed in 2004 to address the security of credit card transactions and to apply to all cardholder associations, such as MasterCard and Visa. The results are documented in the most recent PCI Data Security Standard (DSS) document, Version 1.2, released October 1, 2008. The DSS document can be found at the PCI Council Web site at www.pcisecuritystandards.org/.

The standard addresses card and cardholder authentication and is organized as 12 requirements under 6 logically consistent control objectives. Compliance requires that all 12 requirements are met, with noncompliance penalties up to a maximum of \$500,000 per incident. Merchants and member banks were required to meet the DSS starting on June 30, 2005.

The 6 control objectives of the DSS and their corresponding 12 requirement areas as listed in "Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures," version 1.2, October 2008, are summarized as follows:

1. Build and maintain a secure network

- **Requirement 1**—Install and maintain a firewall configuration to protect cardholder data

- **Requirement 2**—Do not use vendor-supplied defaults for system passwords and other security parameters
- 2. **Protect cardholder data**
 - **Requirement 3**—Protect stored cardholder data
 - **Requirement 4**—Encrypt transmission of cardholder data across open, public networks
- 3. **Maintain a vulnerability management program**
 - **Requirement 5**—Use and regularly update antivirus software or programs
 - **Requirement 6**—Develop and maintain secure systems and applications
- 4. **Implement strong access control measures**
 - **Requirement 7**—Restrict access to cardholder data by business need-to-know
 - **Requirement 8**—Assign a unique ID to each person with computer access
 - **Requirement 9**—Restrict physical access to cardholder data
- 5. **Regularly monitor and test networks**
 - **Requirement 10**—Track and monitor all access to network resources and cardholder data
 - **Requirement 11**—Regularly test security systems and processes
- 6. **Maintain an information security policy**
 - **Requirement 12**—Maintain a policy that addresses information security for employees and contractors

Software Architecture

In general, a *software architecture* is a high-level design structure comprising abstract-level components of the required software system functionality and descriptions of their interactions. It is a design plan that assigns and portrays roles and behavior among all IT assets.

Software Architecture Definitions

Some additional definitions of software architecture are as follows:

- Software architecture is concerned with global organization as a composition of components, global control structures, communication protocols, and physical locations.¹¹

¹¹Dijkstra, E. W., “Notes on Structured Programming,” *Structured Programming* (London: Academic Press, 1972).

- Software architecture is a level of design that involves the description of elements from which systems are built, interactions among those elements, patterns that guide their composition, and constraints on those patterns.¹²
- The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.¹³

A software architecture can be considered a composition system comprising the following elements:

- Components (modules with interfaces)
- Connectors (abstraction of communication)
- Operators that create systems from subsystems

The software architecture employs *abstraction*, which is the neglect of unnecessary details.

Software architecture systems are represented by a *component model* that incorporates the following concepts:

- Connector; components attached to ports
- Binding point or port; abstract interface points that specify transfers to and from components
- Glue code generated from connectors
- Separation of application and communication
- Use of a composition language such as an architecture description language (ADL)
- Components and connectors bound together to form a configuration

These concepts are illustrated in Figure 1-3 and Figure 1-4.

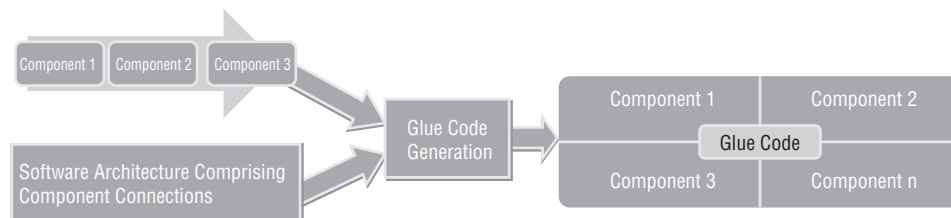


Figure 1-3: Software architecture overview

¹²Shaw, M., and Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline* (Upper Saddle River, NJ: Prentice Hall, 1996).

¹³Bass, L., Clements, P., and Kazman, R., *Software Architecture in Practice* (Reading, MA: Addison-Wesley, 1998).

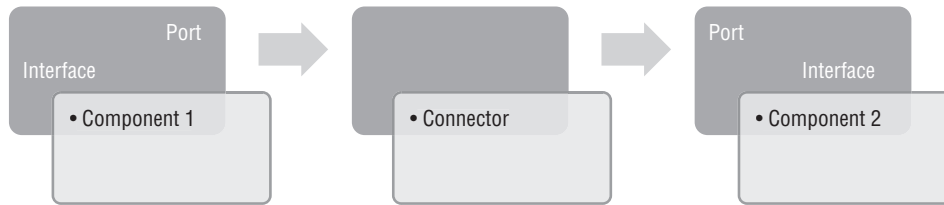


Figure 1-4: Configuration example

Software Architecture Styles

An *architecture style* defines a vocabulary of components and connector types, and a set of constraints on how they can be combined. For many styles there may also exist one or more semantic models that specify how to determine a system's overall properties from the properties of its parts.¹⁴ Some typical architecture styles include:

- **Pipes and filters**—A component receives inputs from connectors (pipes) and produces a transformed (filtered) set of outputs.
- **Layered**—Multiple layers exist in which a layer acts as a client to the layer below and a server to the layer above.
- **N-tiered**—This type enforces a strict separation of concerns that is not demanded in the layered architecture style. Each tier is responsible for a specific functional area.
- **Heterogeneous**—These systems use a number of architecture styles.

Software Architecture Assurance

The Software Assurance Common Body of Knowledge (CBK) developed by the U.S. Department of Homeland Security Software Initiative lists a number of important architectural design assurance objectives. These objectives, summarized from the DHS CBK document, are listed as follows:¹⁵

- The architectural design should ease creation and maintenance of an assurance case.
- The architecture should provide predictable execution behavior.
- The architectural design should ease traceability, verification, validation, and evaluation.
- The architecture should eliminate possibilities for violations.
- The architectural design should help ensure certification and accreditation of the operational system.

¹⁴Shaw, M., and Garlan, D., *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, Upper Saddle River, NJ, 1996.

¹⁵DHS, "Secure Software Assurance, A Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software," Draft Version 0.9, January 9, 2006.

- The design should avoid and work around any security-endangering weaknesses in the environment or development tools.
- The number of components to be trusted should be minimized.
- The system should be designed to do only what the specification calls for and nothing else.
- The system should be designed to tolerate security violations.
- The designer should make weak assumptions.
- The system should not cause security problems for other systems in the environment.

Software Development Methodologies

A number of software development life cycles (SDLCs) have been developed over the years to define the software development process and the life cycle of the resultant software. A typical SDLC comprises the following phases:

1. Requirements
2. Design
3. Implementation
4. Testing
5. Deployment
6. Operations/maintenance
7. Decommissioning

A number of the popular and effective software development methodologies are presented in the following sections.

Software Development Life Cycle Characteristics

One characterization of the software development life cycle process is by the “weight” of the process. A *heavyweight* process employs extensive reviews, evidence, and formal methods. Some elements of a heavyweight process that include software assurance are:

- Security policy
- Test specifications
- Formal specifications
- Formal design
- Coding
- Proof of formal design
- Proof of functional properties

- Static analysis
- Proof of consistency

A *lightweight* process is less restrictive and builds on elements of an extant software development process. The Trustworthy Security Development Life Cycle conceived by Microsoft is a good example of a lightweight process that will be discussed in following section. Examples of software development processes are:

- **Waterfall or linear sequential**—Assumes software development progresses in a linear sequence from one step to another without any backtracking or iteration.
- **Iterative and incremental**—Assumes software development is nonlinear and development might have to recycle to the previous step, if required.
- **Evolutionary or rapid prototyping**—Operates by developing prototypes that are tested and evaluated and then go back to continue the development process incorporating feedback obtained during the prototyping phase.
- **Spiral**—Depicts the software development process as a set of phases and decision points in a circular representation. The model illustrates the concept that software development undergoes a continuous circle of improvement and evolution until the improvement and evolution is no longer required.
- **Concurrent release or cascade model**—Performs iterative development cycles (sprints) and ensures schedules, objectives, quality, and competition issues are met.
- **Unified process**—Comprehensive processes that include items such as maintain business rules, find business actors and use cases, find business workers and entities, and define automation requirement (Rational Unified Process).
- **Agile**—Characterized by early and frequent delivery of workable and usable software, customer involvement in the development process, iterative development, acceptance of late requirements changes, employment of self-managing teams with appropriate expertise, and delivery of multiple releases.

Microsoft Trustworthy Security Development Life Cycle

In 2002, Microsoft developed the Trustworthy Security Development Life Cycle paradigm to increase the security of the resultant software. The objective of the process was to reduce the impact severity of problems in the code as well as minimizing the security-related design and coding “bugs” in the software.

An organization that employs the Microsoft SDL is expected to have a central security entity or team that performs the following functions:

- Develop security best practices
- Serve as a source of security expertise to the organization
- Conduct a final review of the completed software prior to its release
- Consult as needed with the software development organization as a whole

Microsoft Software Development Baseline Process

The basic software development process at Microsoft is a spiral approach in that the software requirements and design are reviewed and updated, if necessary, during the implementation phase. The phases of the Microsoft software development process are summarized as follows:

1. Requirements
 - a. Schedules
 - b. Quality guidelines
 - c. Feature lists
 - d. Architecture documentation
2. Design
 - a. Functional specifications
 - b. Design specifications
3. Implementation
 - a. Development of new code
 - b. Testing and verification
4. Verification
 - a. Testing and verification
 - b. Bug fixes
5. Release
 - a. Code signing and signoff
 - b. Release
6. Support and servicing
 - a. Product support
 - b. Service packs
 - c. Quick Fix Engineering (QFE)
 - d. Security updates

Microsoft Trustworthy Security Development Life Cycle (SDL) Principles and Model

The Microsoft SDL for software is based on the following set of principles expounded in the 2005 Microsoft document, “The Trustworthy Computing Security Development Lifecycle,” by Lipner and Howard. The following summarizes the principles presented in the paper:

- **Secure by design**—The product should be specified, designed, and developed to safeguard against attacks.

- **Secure by default**—Recognizing that there will always be vulnerabilities in real-world software, software should be designed to default to a secure state when problems occur, such as reducing privileges or disabling infrequently used features. Secure by default attempts to minimize the software vulnerability exposure or *attack surface*, as it is sometimes called.
- **Secure in deployment**—Recommendations and guidance should be provided with software in the field to assist system administrators and users in deploying the software in the most secure fashion.
- **Communications**—Software developers should be cognizant of vulnerabilities discovered in the software and should readily communicate this information to users and system administrators along with appropriate patches and updates.

These principles are known as SD³ + C. Incorporating SD³ + C into the basic software development life cycle yields the following spiral model as presented in the “The Trustworthy Computing Security Development Lifecycle”:

1. Requirements
 - a. Security kickoff and coordinate with central security team
 - b. Consideration of security feature requirements
2. Design
 - a. Review security design best practices
 - b. Security architecture and attack surface review
 - c. Threat modeling
3. Implementation
 - a. Application of security development tools
 - b. Application of security best development practices
 - c. Application of security best test practices
 - d. Creation of security documentation and product tools
 - e. Application of static analysis tools
 - f. Conduct of code reviews
4. Verification
 - a. Preparation of security response plan
 - b. Security push (additional security code reviews beyond those performed in the implementation phase)
 - c. Penetration testing
5. Release
 - a. Final security review (FSR)
 - b. Release

6. Support and servicing
 - a. Security servicing
 - b. Evaluation of vulnerability reports
 - c. Response execution

CLASP

The Comprehensive, Lightweight Application Security Process (CLASP) is an open source application of the Open Web Application Security Project (OWASP) that can be referenced at www.owasp.org.

CLASP ORIGINS

CLASP was developed originally by Secure Software, Inc., with major contributions by John Viega and Pravir Chandra. Additional input was provided by Jerry Epstein and IBM. Secure Software was acquired by Fortify Software, Inc., in 2007, and in the transaction, Fortify Software obtained the rights to CLASP. Fortify Software then donated CLASP to the Open Web Application Security Project (OWASP). One of the valuable features of CLASP is that it defines roles that can affect the security of a software system and assigns activities to those roles.

CLASP is designed to support the incorporation of information security processes into each phase of the software development life cycle. OWASP members include educational institutions, corporations, and security-conscious individuals. Its mission is to develop methodologies, technologies, and publications focusing on application security.

CLASP is based on the following seven key best practices for application security listed on the OWASP Web site:

- Institute awareness programs
- Perform application assessments
- Capture security requirements
- Implement secure development practices
- Build vulnerability remediation procedures
- Define and monitor metrics
- Publish operational security guidelines

In addition, CLASP defines 30 security activities, which are organized as discrete process components, and linked to one or more specific project roles. The CLASP security activities as listed on the OWASP Web site are presented as follows:

1. Institute security awareness program
2. Monitor security metrics integrator

3. Manage certification process
4. Specify operational environment
5. Identify global security policy
6. Identify user roles and requirements
7. Detail misuse cases
8. Perform security analysis of requirements
9. Document security design assumptions
10. Specify resource-based security properties
11. Apply security principles to design
12. Research and assess security solutions
13. Build information labeling scheme
14. Design UI for security functionality
15. Annotate class designs with security properties
16. Perform security functionality usability testing
17. Manage system security authorization agreement
18. Specify database security configuration
19. Perform security analysis of system design
20. Integrate security analysis into build process
21. Implement and elaborate resource policies
22. Implement interface contracts
23. Perform software security fault injection testing
24. Address reported security issues
25. Perform source-level security review
26. Identify and implement security tests
27. Verify security attributes of resources
28. Perform code signing
29. Build operational security guide
30. Manage security issue disclosure process

CLASP also provides a means to assist developers in avoiding specific design and code errors that might create vulnerabilities. This mechanism is the CLASP *Vulnerability Lexicon*, which provides a classification structure that enables development teams to find Lexicon vulnerability information acquired from different historical views and sources of data.

To provide guidelines for employing information assurance processes into the SDL, CLASP is structured into *views*, *resources*, and *vulnerability use cases*.

CLASP Views

This component of the CLASP methodology is organized in a hierarchical fashion with a view being the top level, activities being the second level, and process components being the third level. There are five CLASP views as presented on the OWASP Web site:

- I. **Concepts view**—Develop understanding of the interaction among CLASP process components and the application of following views II through V.
- II. **Role-based view**—Develop rules needed by the security-related project and apply these rules in views III through V.
- III. **Activity-assessment view**—Evaluate 24 security-related CLASP activities for possible application in view IV.
- IV. **Activity-implementation view**—Conduct subset of 24 security-related CLASP activities chosen in view III.
- V. **Vulnerability view**—Incorporate solutions to problem types into activities III and IV.

CLASP Resources

CLASP resources, which are provided as part of the CLASP package and labeled alphabetically, provide references to objects that support automation tools for CLASP process elements. Table 1-6 identifies the resources, their alphabetical location reference, and the views they are designed to support.

Table 1-6: CLASP Resources and Supported Views

CLASP RESOURCES	LOCATION
Basic Principles in Application Security (all views)	Resource A
Example of Basic Principle: Input Validation (all views)	Resource B
Example of Basic-Principle Violation: Penetrate-and-Patch Model (all views)	Resource C
Core Security Services (all views, especially III)	Resource D
Sample Coding Guideline Worksheets (views II, III, and IV) * System Assessment Worksheets (views III and IV) *	Resource F
Sample Road Map: Legacy Projects (view III)	Resource G1
Sample Road Map: New-Start Projects (view III)	Resource G2
Creating the Process Engineering Plan (view III)	Resource H
Forming the Process Engineering Team (view III)	Resource I
Glossary of Security Terms (all views)	Resource J

From OWASP Web site, www.owasp.org/index.php/CLASP_Concepts

Vulnerability Use Cases

CLASP vulnerability use cases serve as a connection between the CLASP concepts view and the CLASP Vulnerability Lexicon of the vulnerability view. The vulnerability use cases provide examples of situations where security services such as those designed to protect confidentiality, availability, and integrity and to provide authorization and authentication are subject to exploitation because of vulnerabilities. The role of CLASP use cases in the overall CLASP SDL is shown in Figure 1-5.

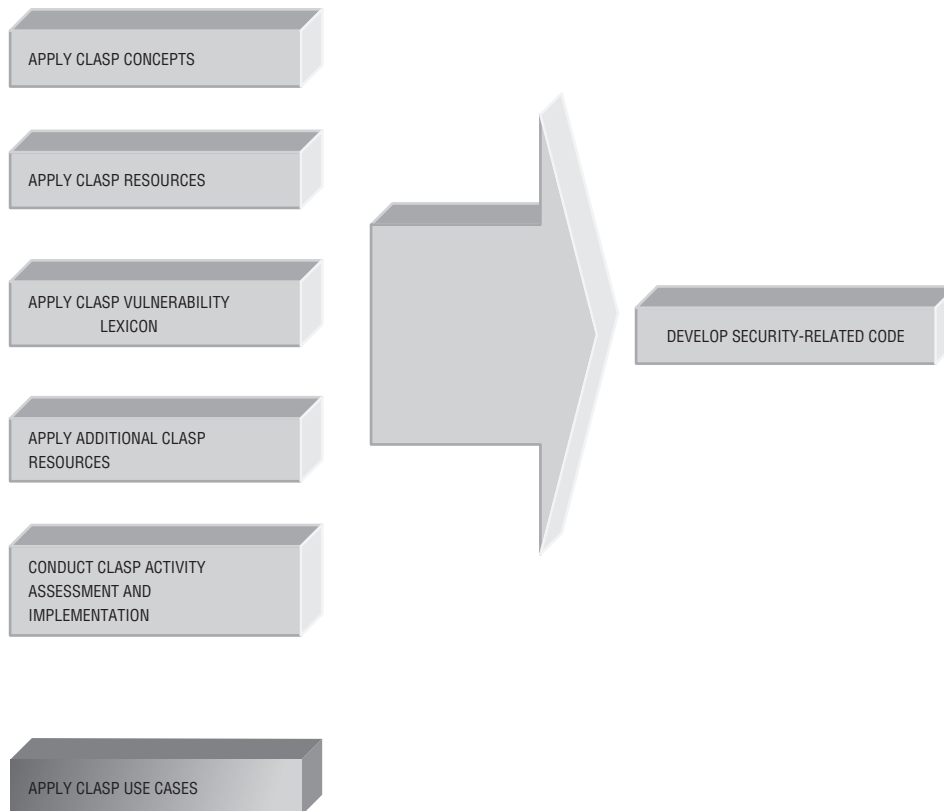


Figure 1-5: Use cases in the CLASP SDL

From OWASP Web site, www.owasp.org/index.php/CLASP_Concepts

Seven Touchpoints for Software Security

In addition to the 30 security activities defined in CLASP, Gary McGraw of Cigital developed seven additional security activities, which he called touchpoints.¹⁶ These touchpoints are summarized in the following list:

1. Employ static analysis of source code

¹⁶McGraw, Gary, "The 7 Touchpoints of Secure Software," *Software Development*, September 2005.

2. Conduct risk analysis of architecture and design
3. Perform penetration testing
4. Perform security functionality testing and risk-based security testing
5. Categorize abuse cases by analyzing system behavior when the system is under attack
6. Specify security requirements for functional security and emergent security characteristics
7. Monitor post deployment security operational behavior

McGraw has since identified an additional touchpoint of having independent outside analysts conduct a review of the security of the software design and implementation.

TSP-Secure

The Carnegie Mellon University Software Engineering Institute (SEI) and CERT Coordination Center (CERT/CC) have developed a method designed to estimate the probability of vulnerabilities in production software and to minimize or eliminate software vulnerabilities. The approach is based on the SEI Team Software Process (TSP) and is called TSP-Secure. TSP-Secure purposely directs engineers through the development process and requires engineers and managers to institute and support a teamwork environment.

TSP-Secure provides the means to incorporate security practices in the SDLC through the following processes:

- Establishment of operational procedures, organizational policies, management oversight, resource allocation, training, and project planning and tracking, all in support of secure software production
- Vulnerability analysis by defect type
- Establishment of security-related predictive process metrics, checkpoints, and measurement
- Risk management and feedback, including asset identification, development of abuse/misuse cases, and threat modeling
- Secure design process that includes conformance to security design principles, use of design patterns for avoiding common vulnerabilities, and design security reviews
- Quality management for secure programming, including use of secure language subsets and coding standards, and code reviews using static and dynamic analysis tools
- Security review, inspection, and verification processes that include development of security test plans, white and black box testing, and test defect reviews/vulnerability analyses by defect type
- Removal of vulnerabilities from legacy software.

Intellectual Property and Privacy Legal Issues

Intellectual property is one of the most valuable assets of an organization. It is what distinguishes the organization from others in the field and provides a competitive advantage in the marketplace. As such, intellectual property must be protected from compromise. The means to protect intellectual property are varied and depend on the type of information that is to be safeguarded. The types of intellectual property law are summarized in the following section.

Intellectual Property Law

The following categories fall under intellectual property law:

Patent

A patent provides the owner of the patent with a legally enforceable *right to exclude* others from practicing the invention covered by the patent for a specified period of time. It is of interest to note that a patent does not necessarily grant the owner the right to make, use, or sell the invention. A patent obtained by an individual might build on other patents, and thus, the individual must obtain permission from the owner(s) of the earlier patent(s) to exploit the new patent.

There are four criteria that an invention must meet in order to be patentable. These criteria are:

- The invention must fall into one of the following five classes:
 - Processes
 - Machines
 - Manufactures (objects made by humans or machines)
 - Compositions of matter
 - New uses of any of the above
- The invention must be *useful*. One aspect of this test for utility is that the invention cannot be only a theoretical phenomenon.
- The invention must be *novel*; it must be something that no one has developed before.
- The invention must be *obvious* to “a person having ordinary skill in the art to which said subject matter pertains.”

Patent law protects inventions and processes (utility patents), ornamental designs (design patents), and new varieties of plants (plant patents). In the United States, as of June 8, 1995, utility patents are granted for a period of 20 years from the date the application was filed. For patents in force prior to June 8, 1995, and patents granted on applications pending before that date, the patent term is the greater of 17 years from the date of issue (the term under prior law) or 20 years from the date of filing. Design patents are granted for a period of 14 years and a plant patent has a term of 17 years.

Once the patent on an invention or design has expired, anyone is free to make, use, or sell the invention or design.

Copyright

A copyright protects “original works of authorship” and protects the right of the author to control the reproduction, adaptation, public distribution, and performance of these original works. Copyrights can also be applied to software and databases. The copyright law has two provisions that address uses of copyrighted material by educators, researchers, and librarians. These provisions:

- Codify the doctrine of fair use, under which limited copying of copyrighted works without the permission of the owner is allowed for certain teaching and research purposes
- Establish special limitations and exemptions for the reproduction of copyrighted works by libraries and archives

The Sonny Bono Copyright Term Extension Act, signed into law on October 27, 1998, amends the provisions concerning duration of copyright protection. The act states that the terms of copyright are generally extended for an additional 20 years. Two specific example provisions of the Sonny Bono Copyright Term Extension Act are as follows:

- Works originally created on or after January 1, 1978, are protected from the time of their creation and are usually given a term of the author’s life plus an additional 70 years after the author’s death.
- Works originally created before January 1, 1978, but not published or registered by that date are covered by the statute, also with a duration of the author’s life plus an additional 70 years after the author’s death. In addition, the statute provides that in no case will the term of copyright for these types of works expire before December 31, 2002. For works published on or before December 31, 2002, the term of copyright will not expire before December 31, 2047.

Materials might fall into other copyright categories depending on the age of the work, if the copyright was renewed, if it was developed as work for hire, and so on. Detailed information can be found in the following publications of the U.S. Copyright Office:

- Circular 15, “Renewal of Copyright”
- Circular 15a, “Duration of Copyright”
- Circular 15t, “Extension of Copyright Terms”

Trade Secret

A trade secret secures and maintains the confidentiality of proprietary technical or business-related information that is adequately protected from disclosure by the owner. Corollaries to this definition are that the owner has invested resources to develop this information, it is valuable to the business of the owner, it would be valuable to a competitor, and it is not obvious.

Trademark

A trademark establishes a word, name, symbol, color, sound, product shape, device, or combination of these that will be used to identify goods and to distinguish them from those made or sold by others.

Warranty

A warranty is a contract that commits an organization to stand behind its product. There are two types of warranties, implied and express. An *implied* warranty is an unspoken, unwritten promise created by state law that goes from a manufacturer or merchant to the customer. Under implied warranties, there are two categories—the implied warranty of fitness for a particular purpose and the implied warranty of merchantability. The implied warranty of *fitness for a particular purpose* is a commitment made by the seller when the consumer relies on the advice of the seller that the product is suited for a specific purpose. The implied *warranty of merchantability* is the seller's or manufacturer's promise that the product sold to the consumer is fit to be sold and will perform the functions that it is intended to perform. An *express* warranty is a warranty that is explicitly offered by the manufacturer or seller to the customer at the time of the sales transaction. This type of warranty contains voluntary commitments to remedy defects and malfunctions that some customers may encounter in using the product. An express warranty can be made orally or in writing. If it is in writing, it falls under the Magnuson-Moss Warranty Act.

The Magnuson-Moss Warranty Act is the 1975 U.S. federal law that governs warranties on consumer products. The Act requires manufacturers and sellers of consumer products to provide consumers with detailed information concerning warranty coverage. In addition, the FTC adopted three rules under the Act. These rules are the *Rule on Disclosure of Written Consumer Product Warranty Terms and Conditions* (the *Disclosure Rule*), the *Rule on Pre-Sale Availability of Written Warranty Terms* (the *Pre-Sale Availability Rule*), and the *Rule on Informal Dispute Settlement Procedures* (the *Dispute Resolution Rule*.) These rules and the act detail three basic requirements that apply to a warrantor or seller. These requirements are:

1. A warrantor must designate, or title, the written warranty as either *full* or *limited*.
2. A warrantor must state certain specified information about the coverage of the warranty in a single, clear, and easy-to-read document.
3. The warrantor or seller must ensure that warranties are available at the site of sale of the warranted consumer products so that consumers can read them before purchasing a product.

Regarding used products, an implied warranty can be disclaimed if a written warranty is not provided. This disclaimer must be made in a conspicuous manner, preferably in writing, so that the consumer is aware that there is no warranty on the product. Terms such as "this product is being sold with all faults" or "as is" should be used. Some states do not permit disclaiming of the implied warranty."

Information Privacy Principles

Privacy is the right of an individual to protection from unauthorized disclosure of the individual's personally identifiable information (PII).

As stated earlier in this chapter, an individual's right to privacy is embodied in the following fundamental principles of privacy:

- **Notice**—Regarding collection, use and disclosure of PII
- **Choice**—To opt out or opt in regarding disclosure of PII to third parties
- **Access**—By consumers to their PII to permit review and correction of information
- **Security**—To protect PII from unauthorized disclosure
- **Enforcement**—Of applicable privacy policies and obligations

The European Union has established privacy principles that serve as guidelines for a number of privacy policies. These principles are reviewed in the next section.

European Union (EU) Principles

The protection of information on private individuals from intentional or unintentional disclosure or misuse is the goal of the information privacy laws. The intent and scope of these laws vary widely from country to country. The European Union (EU) has defined privacy principles that in general are more protective of individual privacy than those applied in the United States. Therefore, the transfer of personal information from the EU to the United States, when equivalent personal protections are not in place in the United States, is prohibited. The EU principles include the following:

- Data should be collected in accordance with the law.
- Information collected about an individual cannot be disclosed to other organizations or individuals unless authorized by law or by consent of the individual.
- Records kept on an individual should be accurate and up to date.
- Individuals have the right to correct errors contained in their personal data.
- Data should be used only for the purposes for which it was collected, and it should be used only for a reasonable period of time.
- Individuals are entitled to receive a report on the information that is held about them.
- Transmission of personal information to locations where equivalent personal data protection cannot be assured is prohibited.

Health Care–Related Privacy Issues

An excellent example of the requirements and application of individual privacy principles is in the area of health care. The protection from disclosure and misuse of a

private individual's medical information is a prime example of a privacy law. Some of the common health care security issues are as follows:

- Access controls of most health care information systems do not provide sufficient granularity to implement the principle of least privilege among users.
- Most off-the-shelf applications do not incorporate adequate information security controls.
- Systems must be accessible to outside partners, members, and some vendors.
- Providing users with the necessary access to the Internet creates the potential for enabling violations of the privacy and integrity of information.
- Criminal and civil penalties can be imposed for the improper disclosure of medical information.
- A large organization's misuse of medical information can cause the public to change its perception of the organization.
- Health care organizations should adhere to the following information privacy principles (based on European Union principles):
 - An individual should have the means to monitor the database of stored information about him or her and should have the ability to change or correct that information.
 - Information obtained for one purpose should not be used for another purpose.
 - Organizations collecting information about individuals should ensure that the information is provided only for its intended use and should provide safeguards against the misuse of this information.
 - The existence of databases containing personal information should not be kept secret.

Platform for Privacy Preferences (P3P)

The Platform for Privacy Preferences was developed by the World Wide Web Consortium (W3C) to implement privacy practices on Web sites. The W3C P3P specification states, "P3P enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents will allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit."

The W3C P3P document can be found at www.w3.org/TR/P3P/. With P3P, an organization can post its privacy policy in machine-readable form (XML) on its Web site. This policy statement should include:

- Who has access to collected information
- The type of information collected

- How the information is used
- The legal entity making the privacy statement

The P3P specification contains the following items:

- A standard vocabulary for describing a Web site's data practices
- A set of data elements that Web sites can refer to in their P3P privacy policies
- A standard schema for data a Web site may wish to collect, known as the "P3P base data schema"
- A standard set of uses, recipients, data categories, and other privacy disclosures
- An XML format for expressing a privacy policy
- A means of associating privacy policies with Web pages or sites and cookies
- A mechanism for transporting P3P policies over HTTP

A useful consequence of implementing P3P on a Web site is that Web site owners are required to answer multiple-choice questions about their privacy practices. This activity will cause the organization sponsoring the Web site to think about and evaluate their privacy policy and practices in the event that they have not already done so. After answering the necessary P3P privacy questions, an organization can then proceed to develop their policy. A number of sources provide free policy editors and assistance in writing privacy policies. Some of these resources can be found at www.w3.org/P3P/ and <http://p3ptoolbox.org/>.

P3P also supports user agents that allow a user to configure a P3P-enabled Web browser with the user's privacy preferences. Then, when the user attempts to access a Web site, the user agent compares the user's stated preferences with the privacy policy in machine-readable form at the Web site. Access will be granted if the preferences match the policy. Otherwise, either access to the Web site will be blocked or a pop-up window will appear notifying the user that he or she must change the privacy preferences.

Standards and Guidelines

Two of the most popular standards and guidelines for information and software assurance are the International Organization for Standardization (ISO) 27000 series and the Open Web Application Security Project (OWASP) guidelines outlining the "top ten" Web application security vulnerabilities.

ISO 27000 Series

The ISO series 27000 (2700X) standards are dedicated to the field of information system security. The relevant standards are ISO 27001, 27002, 27003, 27004, 27005, and 27006, which are summarized in the following sections.

ISO 27001

The British Standards Institution (BSI) 7799-2 standard was the predecessor and basis for ISO 27001, which is the specification for an information security management system (ISMS). According to ISO, the standard is designed to “provide a model for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving an Information Security Management System.”

ISO 27001 comprises the following topics:

- Management responsibility
- Internal audits
- ISMS improvement
- Annex A—Control objectives and controls
- Annex B—Organization for Economic Cooperation and Development (OECD) principles and this international standard
- Annex C—Correspondence between ISO 9001, ISO 14001, and this standard

ISO 27001 emphasizes developing an ISMS through an iterative plan-do-check-act (PDCA) cycle. The activities in each cycle component are summarized from the 27001 document as follows:

1. Plan
 - Establish scope
 - Develop a comprehensive ISMS policy
 - Conduct risk assessment
 - Develop a risk treatment plan
 - Determine control objectives and controls
 - Develop a statement of applicability describing and justifying why the specific controls were selected and others not selected
2. Do
 - Operate selected controls
 - Detect and respond to incidents properly
 - Conduct security awareness training
 - Manage resources required to accomplish security tasks
3. Check
 - Intrusion detection operations
 - Incident handling operations
 - Conduct internal ISMS audit
 - Conduct a management review

4. Act

- Implement improvements to the ISMS in response to items identified in Check phase
- Take corrective actions in response to items identified in Check phase
- Take preventive actions in response to items identified in Check phase

ISO 27002

ISO 27002, the “Code of Practice for Information Security Management,” is a repackaged version of (ISO) 17779:2005. It is designed to serve as a single source for best practices in the field of information security and presents a range of controls applicable to most situations. It provides high level, voluntary guidance for information security management.

ISO 27002 presents requirements for building, maintaining, and documenting ISMSs. As such, it lists recommendations for establishing an efficient information security management framework. ISO 27002 is also used as the basis of a certification assessment of an organization. It lists a variety of control measures that can be implemented according to practices outlined in ISO 27001. The areas covered in ISO 27002 are:

- Structure
- Risk assessment and treatment
- Security policy
- Organization of information security
- Asset management
- Human resources security
- Physical security
- Communications and operations management
- Access control
- Information systems acquisition, development, maintenance
- Information security incident management
- Business continuity
- Compliance

ISO 27003

ISO 27003, “Information Technology – Security Techniques – Information Security Management System Implementation Guidance,” is in draft form as of this writing and uses the PDCA paradigm to provide recommendations and guidance in developing an ISMS.

The draft table of contents is as follows:

1. Introduction
2. Scope
3. Terms and Definitions
4. CSFs (Critical Success Factors)
5. Guidance on Process Approach
6. Guidance on Using PDCA
7. Guidance on Plan Processes
8. Guidance on Do Processes
9. Guidance on Check Processes
10. Guidance on Act Processes
11. Inter-Organization Co-operation

ISO 27004

ISO 27004, “Information Technology – Security Techniques – Information Security Management – Measurement,” is in second final committee draft form as of this writing. According to ISO, the standard “provides guidance on the specification and use of measurement techniques for providing assurance as regards the effectiveness of information security management systems. It is intended to be applicable to a wide range of organizations with a correspondingly wide range of information security management systems.”

ISO 27005

ISO 27005:2008, “Information Technology – Security Techniques – Information Security Risk Management,” provides guidelines for information security risk management (ISRM) according to the requirements outlined in ISO 27001.

The main headings of ISO 27005 are:

- Terms and Definitions
- Structure
- Background
- Overview of the ISRM Process
- Context Establishment
- Information Security Risk Assessment (ISRA)

- Information Security Risk Treatment
- Information security Risk Acceptance

ISO 27006

ISO 27006, “Information Technology – Security Techniques – Requirements for Bodies Providing Audit and Certification of Information Security Management Systems,” provides guidelines for the accreditation of organizations that are concerned with certification and registration relating to ISMSs.

The main elements covered in the standard document are:

- Scope
- References
- Terms
- Principles
- General requirements
- Structural requirements
- Resource requirements
- Information requirements
- Process requirements
- Management system requirements
- Information security risk communication
- Information security risk monitoring and review
- Annex A: Defining the scope of the process
- Annex B: Asset valuation and impact assessment
- Annex C: Examples of typical threats
- Annex D: Vulnerabilities and vulnerability assessment methods
- Annex E: Information security risk assessment (ISRA) approaches

OWASP Top Ten Project

The Open Web Application Security Project (OWASP) Top Ten Project provides a minimum standard for Web application security. It summarizes the top ten Web application security vulnerabilities based on input from a variety of information system security experts. The results provide guidance to standards that can be used to address these security weaknesses. The Top Ten vulnerabilities are summarized in Table 1-7.

Table 1-7: Summary of OWASP Top Ten Web Application Vulnerabilities

A1—Cross Site Scripting (XSS)
A2—Injection Flaws
A3—Malicious File Execution
A4—Insecure Direct Object Reference
A5—Cross Site Request Forgery (CSRF)
A6—Information Leakage and Improper Error Handling
A7—Broken Authentication and Session Management
A8—Insecure Cryptographic Storage
A9—Insecure Communications
A10—Failure to Restrict URL Access

From OWASP Top Ten 2007 Web site, www.owasp.org/index.php/Top_10_2007

OWASP Development Guide

Another document that addresses application security is the OWASP Development Guide, version 3.0, which focuses on Web application security. The guide describes how to make Web applications self-defending. The chapters in the guide are organized into the following three sections:

- **Best practices**—Key features that should be included in applications
- **Secure patterns**—Optional security patterns that can be used as guides
- **Anti-patterns**—Patterns in code that increase vulnerability

Some of the topics addressed by the guide include:

- Secure coding principles
- Threat risk modeling
- Phishing
- Ajax and other “rich” interface technologies
- Session management
- Data validation
- Error handling, auditing, and logging
- Distributed computing
- Buffer overflows

- Cryptography
- Software quality assurance

OWASP Code Review Guide

The OWASP Code Review Guide defines secure code review as “the process of auditing code for an application on a line by line basis for its security quality. Code review is a way of ensuring that the application is developed in an appropriate fashion so as to be self defending in its given environment (www.owasp.org/index.php/Category:OWASP_Code_Review_Project).” Review code for security is usually a manual effort, although some tools have been developed to assist in the process.

Secure code review comprises the following phases, according to the OWASP Code Review Guide:

- Discovery
- Transactional analysis
- Post-transaction analysis
- Procedure peer review
- Reporting and presentation
- Laying the groundwork

In secure code review, the important items that have to be considered are:

- **Code**—The language and associated features used
- **Context**—Knowledge of the application
- **Audience**—The users of the application
- **Importance**—Criticality of the availability of the application

The OWASP Code Review Guide has also identified the following “top nine” source code flaw categories:

- Input validation
- Source code design
- Information leakage and improper error handling
- Direct object reference
- Resource usage
- API usage
- Best practices violation
- Weak session management
- Using HTTP GET query strings

NIST SP 800-95

NIST Special Publication 800-95, "Guide to Secure Web Services," provides guidance on security Web services. It address the following issues:

- Functional integrity of Web services during transactions
- Confidentiality and integrity of data transmitted during Web services protocols
- Availability in the event of attacks, such as denial of service

The security techniques covered in NIST SP 800-35 are:

- Confidentiality of Web services messages using XML Encryption
- Integrity of Web services messages using XML Signature
- Web service authentication and authorization using XML Signature
- Web Services (WS) Security
- Security for Universal Description, Discovery, and Integration (UDDI)

NIST SP 800-95 recommends that organizations consider the following security actions where applicable:

- Replicate data and services to improve availability
- Use logging of transactions to improve nonrepudiation and accountability
- Use threat modeling and secure software design techniques to protect from attacks
- Use performance analysis and simulation techniques for end-to-end quality of service and quality of protection
- Digitally sign UDDI entries to verify the author of registered entries
- Enhance existing security mechanisms and infrastructure

OWASP Testing Guide

The OWASP Testing Guide 2008, V3.0 (www.owasp.org/index.php/Category:OWASP_Testing_Project) defines testing as "a process of comparing the state of a system/application against a set of criteria." The testing techniques described in the guide are:

- Manual inspections and reviews
- Threat modeling
- Code review
- Penetration testing

Testing is used to determine if security controls are functioning as desired, validate security requirements, and determine threats and the root causes of vulnerabilities.

Common security tests that should be performed to evaluate security controls include:

- Authentication and access control
- Input validation and encoding
- Encryption
- User and session management
- Error and exception handling
- Auditing and logging

Information Security Models

Models are used in information security to formalize security policies. These models might be abstract or intuitive and will provide a framework for the understanding of fundamental concepts. In this section, three types of models are described: access control models, integrity models, and information flow models.

Access Control Models

Access control philosophies can be organized into models that define the major and different approaches to this issue. These models are the access matrix, the Take-Grant model, the Bell-LaPadula confidentiality model, and the state machine model.

Access Matrix

The *access matrix* or *access control matrix* is a straightforward approach that provides access rights to subjects for objects.

- *Access rights* are of the type read, write, and execute
- A *subject* is an active entity that is seeking rights to a resource or object. A subject can be a person, a program, or a process.
- An *object* is a passive entity, such as a file or a storage resource.

In some cases, an item can be a subject in one context and an object in another. A typical access control matrix is shown in Figure 1-6.

The columns of the access matrix are called *access control lists* (ACLs), and the rows are called *capability lists*. The access matrix model supports discretionary access control because the entries in the matrix are at the discretion of the individual(s) who have the authorization authority over the matrix. In the access control matrix, a subject's capability can be defined by the *triple* (object, rights, and random #). Thus, the triple defines the rights that a subject has to an object along with a random number used to prevent a replay or spoofing of the triple's source.

Subject \ Object	File Object	File Salaries	Process Deductions	Print Server A
Joe	Read	Read/Write	Execute	Write
Jane	Read/Write	Read	None	Write
Process Check	Read	Read	Execute	None
Process Tax	Read/Write	Read/Write	Call	Write

Figure 1-6: Access control matrix

Take-Grant Model

The Take-Grant model uses a directed graph to specify the rights that a subject can transfer to an object or that a subject can take from another subject. For example, assume that Subject A has a set of rights (S) that includes Grant rights to Object B. This capability is represented in Figure 1-7a. Then assume that Subject A can transfer Grant rights to Subject C and that Subject A has another set of rights (Y) to Object D. In some cases, Object D acts as an object, and in other cases, it acts as a subject. Then, as shown by the heavy arrow in Figure 1-7b, Subject C can grant a subset of the Y rights to Subject/Object D because Subject A passed the Grant rights to Subject C.

The Take capability operates in an identical fashion as the Grant illustration.

Bell-LaPadula Model

The Bell-LaPadula model was developed to formalize the U.S. Department of Defense (DoD) multi-level security policy. The DoD labels materials at various levels of security classification. These levels are Unclassified, Confidential, Secret, and Top Secret—ordered from least sensitive to most sensitive. An individual who receives a clearance of Confidential, Secret, or Top Secret can access materials at that level of classification or below. An additional stipulation, however, is that the individual must have a need-to-know for that material. Thus, an individual cleared for Secret can access only the Secret-labeled documents that are necessary for that individual to perform an assigned job function. The Bell-LaPadula model deals *only with the confidentiality* of classified material. It does not address integrity or availability.

The Bell-LaPadula model is built on the *state machine* concept. This concept defines a set of allowable states (A_i) in a system. The transition from one state to another upon receipt of input(s) (X_j) is defined by transition functions (f_k). The objective of this model is to ensure that the initial state is secure and that the transitions always result in a secure state. The transitions between two states are illustrated in Figure 1-8.

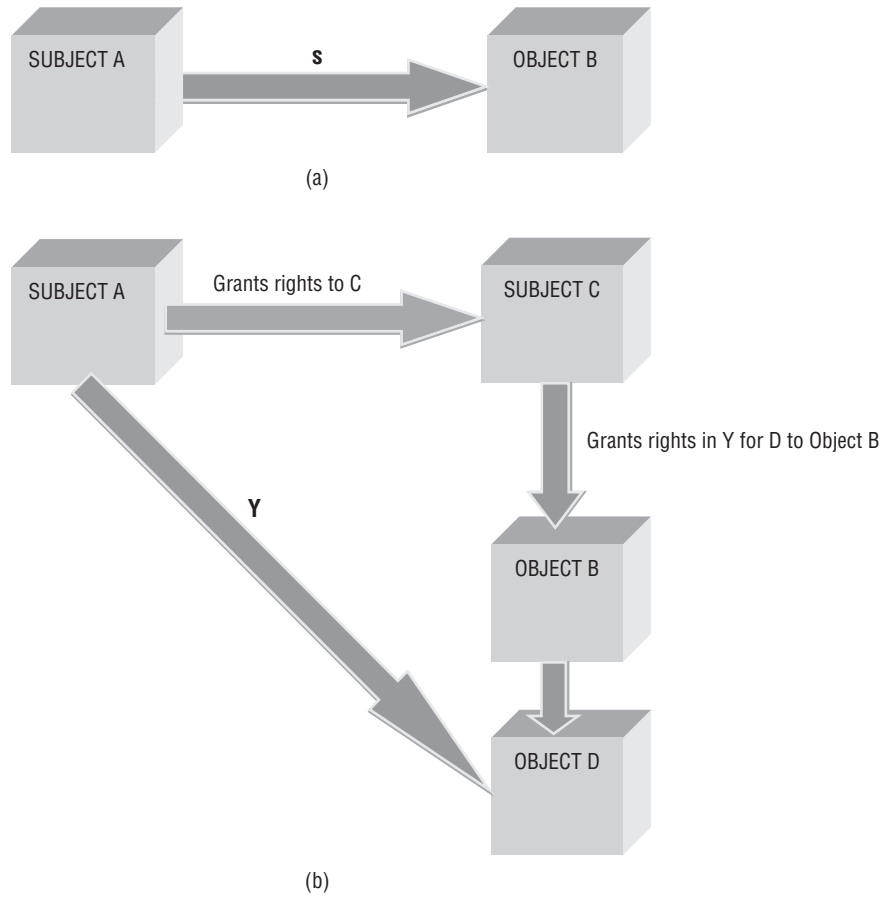


Figure 1-7: Take-Grant model illustration

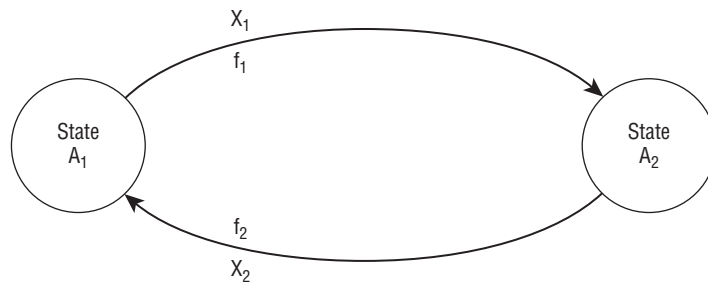


Figure 1-8: State transitions defined by the function f with an input X

The Bell-LaPadula model defines a *secure state* through three multi-level properties. The first two properties implement mandatory access control, and the third one permits discretionary access control. These properties are defined as follows:

- **Simple security property (ss property)**—States that the reading of information by a subject at a lower sensitivity level from an object at a higher sensitivity level is not permitted (no read-up). In formal terms, this property states that a subject can read an object only if the access class of the subject dominates the access class of the object. Thus, a subject can read an object only if the subject is at a higher sensitivity level than the object.
- *** (star) security property**—States that the writing of information by a subject at a higher level of sensitivity to an object at a lower level of sensitivity is not permitted (no write-down). Formally stated, under * property constraints, a subject can write to an object only if the access class of the object dominates the access class of the subject. In other words, a subject at a lower sensitivity level can only write to an object at a higher sensitivity level.
- **Discretionary security property**—Uses an access matrix to specify discretionary access control

There are instances where the * (star) property is too restrictive and it interferes with required document changes. For instance, it might be desirable to move a low-sensitivity paragraph in a higher-sensitivity document to a lower-sensitivity document. The Bell-LaPadula model permits this transfer of information through a *trusted subject*. A trusted subject can violate the * property, yet it cannot violate its intent. These concepts are illustrated in Figure 1-9.

In some instances, a property called the *strong * property* is cited. This property states that reading or writing is permitted at a particular level of sensitivity but not to either higher or lower levels of sensitivity.

This model defines requests (R) to the system. A request is made while the system is in the state v_1 ; a decision (d) is made upon the request, and the system changes to the state v_2 . (R, d, v_1 , v_2) represents this tuple in the model. Again, the intent of this model is to ensure that there is a transition from one secure state to another secure state.

The discretionary portion of the Bell-LaPadula model is based on the access matrix. The system security policy defines who is authorized to have certain privileges to the system resources. *Authorization* is concerned with how access rights are defined and how they are evaluated. Some discretionary approaches are based on context-dependent and content-dependent access control. *Content-dependent* control makes access decisions based on the data contained in the object, whereas *context-dependent* control uses subject or object attributes or environmental characteristics to make these decisions. Examples of such characteristics include a job role, earlier accesses, and file creation dates and times.

As with any model, the Bell-LaPadula model has some weaknesses. These are the major ones:

- The model considers normal channels of the information exchange and does not address covert channels.

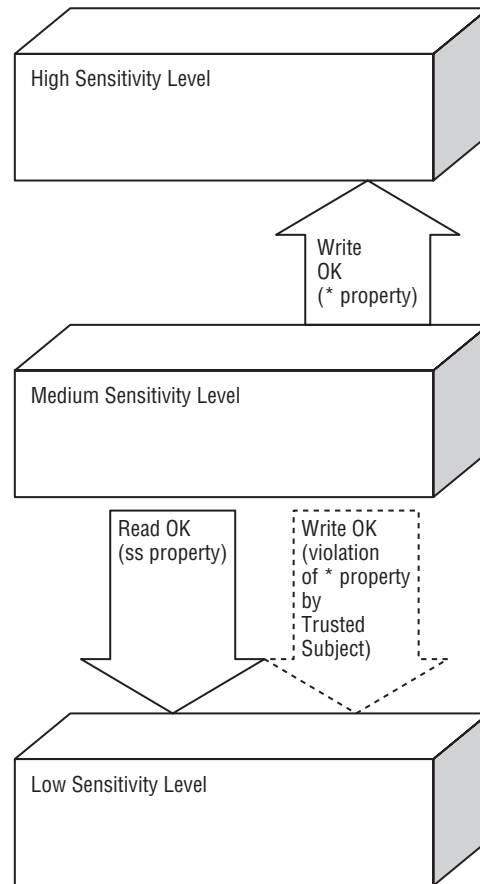


Figure 1-9: The Bell-LaPadula simple security and * properties

- The model does not deal with modern systems that use file sharing and servers.
- The model does not explicitly define what it means by a secure state transition.
- The model is based on a multi-level security policy and does not address other policy types that might be used by an organization.

Integrity Models

In many organizations, both governmental and commercial, the integrity of the data is as important or more important than confidentiality for certain applications. Thus, formal integrity models evolved. Initially, the integrity model was developed as an analog to the Bell-LaPadula confidentiality model and then became more sophisticated to address additional integrity requirements.

Biba Integrity Model

Integrity is usually characterized by the three following goals:

- The data is protected from modification by unauthorized users.
- The data is protected from unauthorized modification by authorized users.
- The data is internally and externally consistent; the data held in a database must balance internally and correspond to the external, real-world situation.

To address the first integrity goal, the Biba model was developed in 1977 as an integrity analog to the Bell-LaPadula confidentiality model. The Biba model is lattice-based and uses the less-than or equal-to relation. A *lattice structure* is defined as a partially ordered set with a *least upper bound* (LUB) and a *greatest lower bound* (GLB). The lattice represents a set of *integrity classes* (ICs) and an ordered relationship among those classes. A lattice can be represented as (IC, \leq , LUB, GUB).

Similar to the Bell-LaPadula model's classification of different sensitivity levels, the Biba model classifies objects into different levels of integrity. The model specifies the three following integrity axioms:

- **Simple integrity axiom**—States that a subject at one level of integrity is not permitted to observe (read) an object of a lower integrity (no read-down). Formally, a subject can read an object only if the integrity access class of the object dominates the integrity class of the subject.
- *** (star) integrity axiom**—States that an object at one level of integrity is not permitted to modify (write to) an object of a higher level of integrity (no write-up). In formal terms, a subject can write to an object only if the integrity access class of the subject dominates the integrity class of the object.
- **Invocation property**—Prohibits a subject at one level of integrity from invoking a subject at a higher level of integrity. This property prevents a subject at one level of integrity from invoking a utility such as a piece of software that is at a higher level of integrity. If this invocation were possible, the software at the higher level of integrity could be used to access data at that higher level.

These axioms and their relationships are illustrated in Figure 1-10.

Clark-Wilson Model

The approach of the Clark-Wilson model (1987) was to develop a framework for use in the real-world, commercial environment. This model addresses the three integrity goals and defines the following terms:

- **Constrained data item (CDI)**—A data item whose integrity is to be preserved
- **Integrity verification procedure (IVP)**—Confirms that all CDIs are in valid states of integrity
- **Transformation procedure (TP)**—Manipulates the CDIs through a *well-formed transaction*, which transforms a CDI from one valid integrity state to another valid integrity state

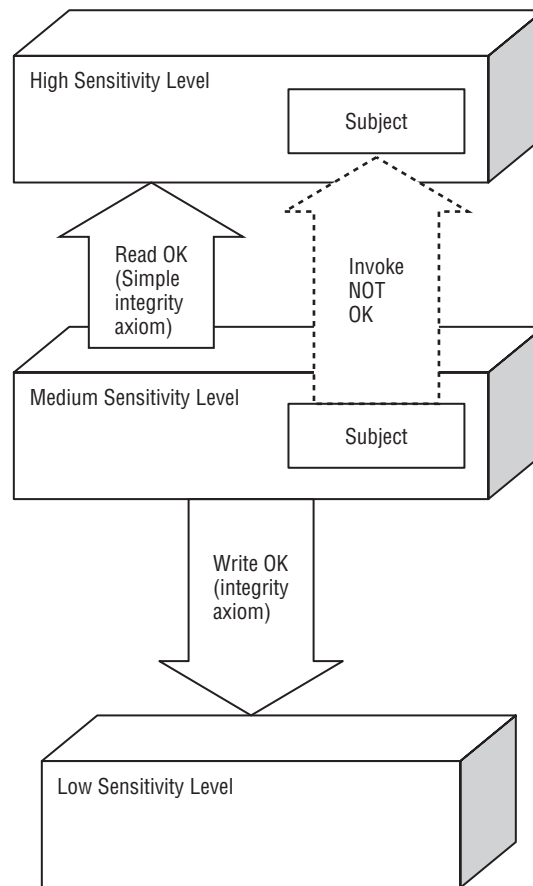


Figure 1-10: The Biba integrity model

- **Unconstrained data item**—Data items outside the control area of the modeled environment, such as input information

The Clark-Wilson model requires integrity labels to determine the integrity level of a data item and to verify that this integrity was maintained after an application of a TP. This model incorporates mechanisms to enforce internal and external consistency, a separation of duty, and a mandatory integrity policy.

Information Flow Models

An information flow model is based on a state machine, and it consists of objects, state transitions, and lattice (flow policy) states. In this context, objects can also represent users. Each object is assigned a security class and value, and information is constrained to flow in the directions that are permitted by the security policy. An example is shown in Figure 1-11.

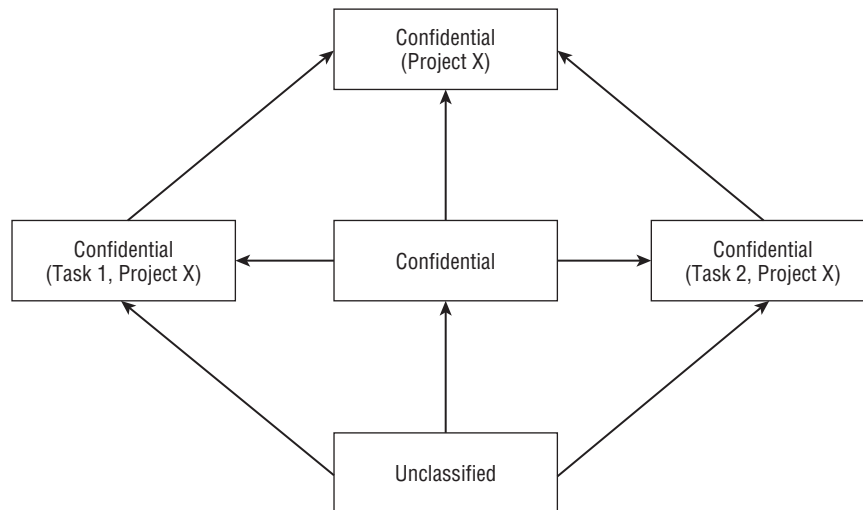


Figure 1-11: An information flow model

In Figure 1-11, information flows from Unclassified to Confidential in tasks in Project X and to the combined tasks in Project X. This information can flow in only one direction.

Non-interference Model

This model is related to the information flow model with restrictions on the information flow. The basic principle of this model is that a group of users (A) who are using the commands (C) do not interfere with the user group (B) who are using commands (D). This concept is written as $A, C \mid B, D$. Restating this rule, the actions of Group A who are using commands C are not seen by users in Group B using commands D.

Chinese Wall Model

The Chinese Wall model, developed by Brewer and Nash, is designed to prevent information flow that might result in a conflict of interest in an organization representing competing clients. For example, in a consulting organization, analyst Joe might be performing work for the Ajax corporation involving sensitive Ajax data while analyst Bill is consulting for a competitor, the Beta corporation. The principle of the Chinese Wall is to prevent compromise of the sensitive data of either or both firms because of weak access controls in the consulting organization.

Composition Theories

In most applications, systems are built by combining smaller systems. An interesting situation to consider is whether the security properties of component systems are maintained when they are combined to form a larger entity.

John McClean studied this issue in 1994.¹⁷ He defined two compositional constructions: external and internal. The following are the types of external constructs:

- **Cascading**—One system's input is obtained from the output of another system.
- **Feedback**—One system provides the input to a second system, which in turn feeds back to the input of the first system.
- **Hookup**—One system communicates with another system as well as with external entities.

The internal composition constructs are intersection, union, and difference.

The general conclusion of this study was that the security properties of the small systems were maintained under composition (in most instances) in the cascading construct, yet are also subject to other system variables for the other constructs.

Systems Security Engineering Capability Maturity Model (SSE-CMM)

The Systems Security Engineering Capability Maturity Model (SSE-CMM¹⁸; copyright © 1999 by the Systems Security Engineering Capability Maturity Model [SSE-CMM] Project) is based on the premise that if you can guarantee the quality of the processes that are used by an organization, then you can guarantee the quality of the products and services generated by those processes. It was developed by a consortium of government and industry experts and is now under the auspices of the International Systems Security Engineering Association (ISSEA) at www.issea.org. The SSE-CMM has the following salient points:

- Describes those characteristics of security engineering processes essential to ensure good security engineering
- Captures the industry's best practices
- Presents an accepted way of defining practices and improving capability
- Provides measures of growth in capability of applying processes

The SSE-CMM addresses the following areas of security:

- Operations security
- Information security
- Network security
- Physical security
- Personnel security
- Administrative security

¹⁷McLean, J. "A General Theory of Composition for Trace Sets Closed Under Selective Interleaving Functions," *Proceedings of 1994 IEEE Symposium on Research in Security and Privacy*, IEEE Press, 1994.

¹⁸"The Systems Security Engineering Capability Maturity Model v3.0," 2003.

- Communications security
- Emanations security
- Computer security

The SSE-CMM methodology and metrics provide a reference for comparing existing systems' security engineering best practices against the essential systems security engineering elements described in the model. It defines two dimensions that are used to measure the capability of an organization to perform specific activities. These dimensions are domain and capability.

- The domain dimension consists of all the practices that collectively define security engineering. These practices are called base practices (BPs). Related BPs are grouped into process areas (PAs).
- The capability dimension represents practices that indicate process management and institutionalization capability. These practices are called generic practices (GPs) because they apply across a wide range of domains. The GPs represent activities that should be performed as part of performing BPs.

For the domain dimension, the SSE-CMM specifies 11 security engineering PAs and 11 organizational and project-related PAs, each consisting of BPs. BPs are mandatory characteristics that must exist within an implemented security engineering process before an organization can claim satisfaction in a given PA. The 22 PAs and their corresponding BPs incorporate the best practices of systems security engineering. The PAs are as follows:

- Security Engineering
 - PA01—Administer Security Controls
 - PA02—Assess Impact
 - PA03—Assess Security Risk
 - PA04—Assess Threat
 - PA05—Assess Vulnerability
 - PA06—Build Assurance Argument
 - PA07—Coordinate Security
 - PA08—Monitor Security Posture
 - PA09—Provide Security Input
 - PA10—Specify Security Needs
 - PA11—Verify and Validate Security
- Project and Organizational Practices
 - PA12—Ensure Quality
 - PA13—Manage Configuration
 - PA14—Manage Project Risk
 - PA15—Monitor and Control Technical Effort
 - PA16—Plan Technical Effort

- PA17—Define Organization’s Systems Engineering Process
- PA18—Improve Organization’s Systems Engineering Process
- PA19—Manage Product Line Evolution
- PA20—Manage Systems Engineering Support Environment
- PA21—Provide Ongoing Skills and Knowledge
- PA22—Coordinate with Suppliers

The GPs are ordered in degrees of maturity and are grouped to form and distinguish among five levels of security engineering maturity. The attributes of these five levels as given in the SSE-CMM model document are as follows:

- Level 1
 - 1.1 BPs Are Performed
- Level 2
 - 2.1 Planning Performance
 - 2.2 Disciplined Performance
 - 2.3 Verifying Performance
 - 2.4 Tracking Performance
- Level 3
 - 3.1 Defining a Standard Process
 - 3.2 Perform the Defined Process
 - 3.3 Coordinate the Process
- Level 4
 - 4.1 Establishing Measurable Quality Goals
 - 4.2 Objectively Managing Performance
- Level 5
 - 5.1 Improving Organizational Capability
 - 5.2 Improving Process Effectiveness

The corresponding descriptions of the five levels are given as follows:

- Level 1, “Performed Informally,” focuses on whether an organization or project performs a process that incorporates the BPs. A statement characterizing this level would be, “You have to do it before you can manage it.”
- Level 2, “Planned and Tracked,” focuses on project-level definition, planning, and performance issues. A statement characterizing this level would be, “Understand what’s happening on the project before defining organization-wide processes.”
- Level 3, “Well Defined,” focuses on disciplined tailoring from defined processes at the organization level. A statement characterizing this level would be, “Use the best of what you’ve learned from your projects to create organization-wide processes.”

- Level 4, “Quantitatively Controlled,” focuses on measurements being tied to the business goals of the organization. Although it is essential to begin collecting and using basic project measures early, measurement and use of data is not expected organization-wide until the higher levels have been achieved. Statements characterizing this level would be, “You can’t measure it until you know what ‘it’ is” and “Managing with measurement is only meaningful when you’re measuring the right things.”
- Level 5, “Continuously Improving,” gains leverage from all the management practice improvements seen in the earlier levels and then emphasizes the cultural shifts that will sustain the gains made. A statement characterizing this level would be, “A culture of continuous improvement requires a foundation of sound management practice, defined processes, and measurable goals.”

A SOFTWARE SECURITY FRAMEWORK

In 2008, Gary McGraw and Brian Chess introduced a software security framework (SSF) that incorporates a maturity model. The model, summarized at www.informit.com/articles/article.aspx?p=127138, defines and provides means to measure software security initiatives.

The SSF comprises 4 domains that encompass 12 practices. These domains are:

- **Governance**—Managing, measuring, and organizing the software security project
- **Intelligence**—Collecting corporate knowledge and modeling threats
- **SDL touchpoints**—Analyzing software development processes
- **Deployment**—Interfacing with network security and software maintenance organizations

Table 1-8 summarizes the 12 practices of the framework. As of this writing, maturity models are being developed for each practice, including a Software Assurance Maturity Model called OpenSAMM (www.opensamm.org).

Table 1-8 summarizes the 12 practices of the framework.

Table 1-8: Practices of the Software Security Framework

GOVERNANCE	INTELLIGENCE	SDL TOUCHPOINTS	DEPLOYMENT
Strategy and metrics	Attack models	Architecture analysis	Penetration testing
Compliance and policy	Security features and design	Code review	Software environment
Training	Standards and requirements	Security testing	Configuration management and vulnerability management

Additional related efforts include the Building Security In Maturity Model (BSIMM) (<http://www.bsi-mm.com/>) and the Software Assurance Maturity Model (OpenSAMM) (<http://www.opensamm.org>). The BSIMM comprises 110 activities that are based on the SSF framework. It is designed to assist an organization in developing a software security program. In general, there are a number of activities for each of the twelve practices, with each activity divided into levels of maturity for that particular practice. The Software Assurance Maturity Model provides guidance for developing a software security approach customized to the identified risks to the organization. It is an open model and is available at no cost to users.

Trusted Computing

The threats to computers and networks are becoming increasingly sophisticated and adaptable. In many instances, software alone is not robust enough to protect information systems from attack. In order to counter malicious attacks, a not-for-profit organization comprising a variety of industrial members from around the globe was formed in 2003. This entity is the Trusted Computing Group (TCG), and its mission is to develop and promote vendor-independent, open specifications addressing the security of computing platforms, including hardware and software. These specifications are designed to provide the following capabilities:

- Authentication using two or more factors
- Secure file storage and privacy protections
- Access control to networks based on an organization's security policy
- Access to operating system-provided security capabilities

TCG Elements

Key elements of the TCG are:

- The Trusted Platform Module (TPM), an embedded hardware-based protection mechanism designed to protect the security and privacy of individual computer users. It refers to both a specification and a hardware device.
- The TCG Software Stack (TSS), which developers can use as a foundation for various applications.
- The Trusted Network Connect (TNC) specifications for network security implementations.

Trusted Platform Modules

TPMs are usually mounted on PC motherboards and are designed to protect cryptographic keys and authentication processes. The TPM is an element that can securely generate, store, and manage cryptographic keys. Encrypted data cannot be decrypted unless the key is provided by the secure TPM after appropriate authentication. Each TPM chip contains a secret and unique RSA key built into the chip during its production and can be used to verify the authenticity of other systems with TPM chips.

TCG Software Stack

The TSS is a specification for the software interface to the TPM and provides support for developers in creating interfaces to a number of cryptographic application programming interfaces (APIs). These APIs allow developers to gain access to TPM functions.

Trusted Network Connect

The TNC specification provides the means for network managers to employ network security policies, monitor end point devices, and provide authorized network access based on the security policy.

Trusted Computing Base (TCB)

Another fundamental component of trusted computing is the *trusted computing base* (TCB). The TCB is the total combination of protection mechanisms within a computer system, which includes the hardware, software, and firmware that are trusted to enforce a security policy. The TCB components are responsible for enforcing the security policy of a computing system and, therefore, these components must be protected from malicious and untrusted processes. The TCB must also provide for memory protection and ensure that the processes from one domain do not access memory locations of another domain.

The *security perimeter* is the boundary that separates the TCB from the remainder of the system. A *trusted path* must also exist so that a user can access the TCB without being compromised by other processes or users. A *trusted computer system* is one that employs the necessary hardware and software assurance measures to enable its use in processing multiple levels of classified or sensitive information. This system meets the specified requirements for reliability and security.

Acquisition Assurance Issues

A large amount of the software produced historically did not take into account assurance issues. The resultant vulnerabilities in the software expose their computing platforms to malicious and dangerous attacks. This situation is of particular concern in software that is used in the nation's critical infrastructure systems. Software assurance is aimed at reducing the risks to these sensitive computer systems.

In addition to suppliers, the responsibility for software assurance must also be assumed by acquirers and included in the acquisition process. According to NIST Special Publication 800-64, "Security Considerations in the System Development Life Cycle," acquisition is defined as "all stages of the process of acquiring property or services, beginning with the process for determining the need for the property or services and ending with contract completion and closeout." In many instances, the acquiring elements are not trained to ensure that software assurance is delivered by contractors, particularly in the U.S. defense establishment. To meet this need, the U.S. Department of Defense and the U.S. Department of Homeland Security (DHS) joined in an effort to address software assurance issues in the acquisition process.

A part of the joint DoD and DHS project was the Acquisition and Outsourcing Working Group, which produced a document “that provides information on how to incorporate software assurance (SwA) considerations in key decisions and how to exercise due diligence throughout the acquisition process relative to potential risk exposures that could be introduced by the supply chain.” This document is a prepublication version as of this writing and is entitled “Software Assurance in Acquisition: Mitigating Risks to the Enterprise, A Reference Guide for Security-Enhanced Software Acquisition and Outsourcing: Building Security in Software Assurance.”

The Working Group recognized that software procurement can take a multitude of paths and all the pertinent elements of the supply chain have to be considered. Figure 1-12, taken from the prepublication document, illustrates these paths.

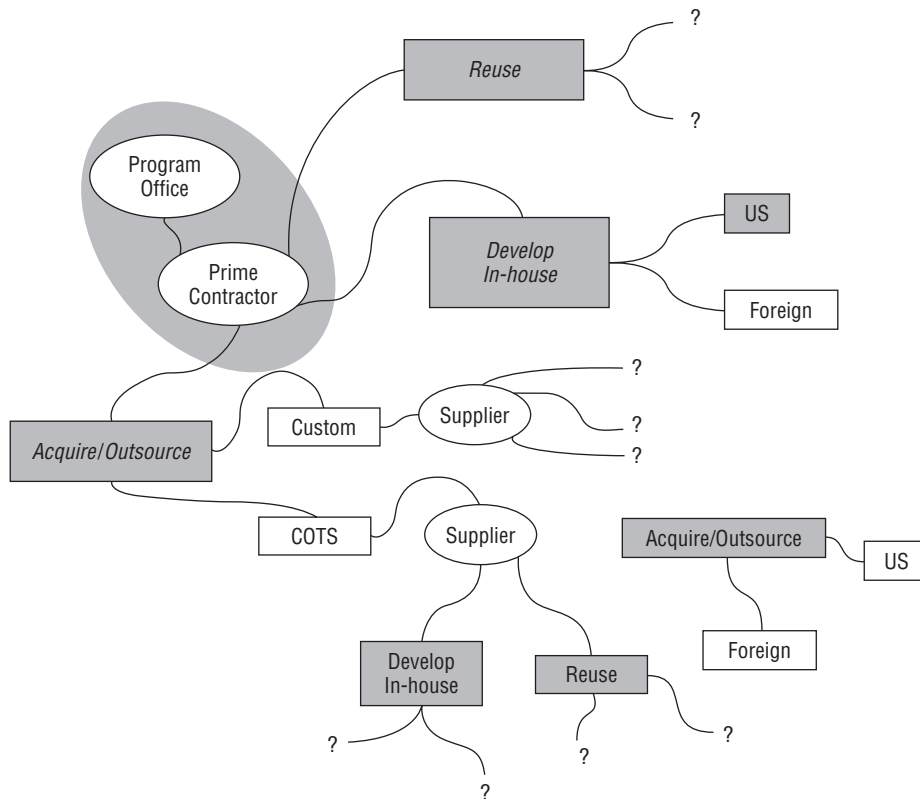


Figure 1-12: Potential software acquisition paths

From “Software Assurance in Acquisition: Mitigating Risks to the Enterprise, A Reference Guide for Security-Enhanced Software Acquisition and Outsourcing: Building Security in Software Assurance,” October 22, 2008.

Some of the issues addressed in the Working Group document are:

- Example contract provisions
- An outline of a general acquisition process and associated acquisition phases
- Recommended purchasing practices for SwA in acquisition

- Support of risk mitigation efforts by providing due diligence questionnaires to obtain information concerning the software supply chain
- Language samples to include in statements of work

The scope of the Working Group effort, as presented in the document, is given in Figure 1-13.

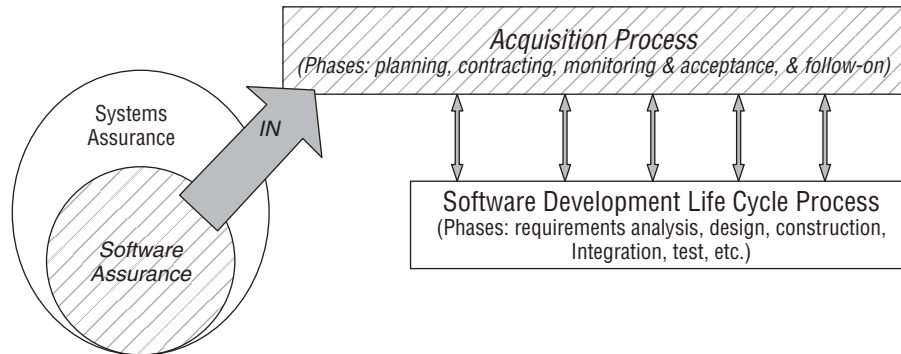


Figure 1-13: Scope of the SwA acquisition effort

Phases of the Acquisition Process

The major phases of the software assurance acquisition process are summarized in Table 1-9.

Figure 1-14 summarizes these phases and their relationship to other relevant processes.

Measures in the Software Assurance Acquisition Process

As with any acquisition effort, it is important to establish metrics and measures in the software acquisition process in order to evaluate the performance of the contractor. Some typical measures include the following:

- The number of software vulnerabilities created and found.
- Service level agreements (SLA) in which the supplier contractually agrees to specific levels of performance requirements.
- Use of earned value management systems (EVMS) that provide insight into the performance, progress, technical quality, cost, scheduling, and planning of suppliers. EVMS are defined in ANSI/EIA 748-1998.

Additional Software Acquisition Assurance Issues

In the development/acquisition phase of the system development life cycle (SDLC) described earlier in this chapter, the following security considerations should be addressed:

- Performance of a risk assessment
- Security requirements analysis

- Conduct of functional testing
- Conduct of security testing
- Development of security architecture
- Preparation of first draft documents for system certification and accreditation
- Determination if the proposed system is capable of performing in a manner expected by the acquiring agency
- Supply chain assurance goals of predictable execution, trustworthiness, and conformance to requirements and standards

The following acquisition planning considerations for software assurance during the development/acquisition phase of the SDLC are excerpted from Appendix F of NIST Special Publication 800-64.

- **Type of contract**—The type of contract (e.g., firm fixed price, time and materials, cost plus fixed fee) can have significant security implications. The information security technical representative developing the specifications and the contracting officer should work together to select the contract type that will be most advantageous to the organization.

	Planning		Contracting		Monitoring & Acceptance		Follow-on
IEEE 1082 1888	Planning		Contracting		Product Implementation	Product Acceptance	Follow-on
PMBOK 3.0	Initiating				1. Planning 2. Executing	3. Monitoring & controlling	Closing
NIST SP 800-84 Rev. 1 2004	Mission & Business Planning	Acquisition Planning	Acquisition		Contract Performance	Contract Closeout	Follow-on Contracts & Disposal
DoD Instruction 5000.2 2003	Pre-Systems Acquisition		Systems Acquisition				Sustainment
ISO/IEC 12207 2008 (E)	Acquisition Preparation		Acquisition Advertisement	Supplier Selection & Contract Agreement	Agreement Monitoring	Acquirer Acceptance & Closure	

Figure 1-14: Software assurance acquisition phases

Figure source: "Mitigating Risks to the Enterprise, Software Assurance in Acquisition," October 22, 2008.

Table 1-9: Phases in the Software Assurance Acquisition Process

PHASE	ACTIVITY
Planning	Software product or service needs determination; identification of associated risks; developing software requirements; creating acquisition strategy; developing evaluation criteria and evaluation plan; development and use of SwA due diligence questionnaires.
Contracting	Creating and issuing the solicitation with a work statement, instructions to offerers, terms and conditions, and certifications; evaluating supplier proposals submitted in response to the solicitation; finalizing contract negotiation to include changes in terms and conditions; awarding the contract. (Software risks are addressed and mitigated through terms and conditions, certifications, evaluation factors for award, and risk mitigation requirements in the work statement.)
Monitoring and Acceptance	Establishing and consenting to the contract work schedule; implementing change (or configuration) control procedures; evaluating risk management and assurance case deliverables to determine compliance in accepted risk mitigation strategies as stated in the requirements; reviewing and accepting software deliverables.
Follow-on	Sustainment (maintenance) of the software, including risk management, assurance case management, and change management; (risks must be managed through continued analysis of the assurance case and should be adjusted to mitigate changing risks); disposal or decommissioning.

From "Mitigating Risks to the Enterprise, Software Assurance in Acquisition," October 22, 2008.

- **Review by other functional groups**—Depending on the size and scope of the system, a review of the system by participants from various functional groups (e.g., legal, human resources, physical security, records management) may be useful. These functional groups should have insight into the confidentiality, integrity, and availability requirements. Involving these groups early in the planning process is important because it may result in reduced life-cycle costs, and it is easier to change requirements in the early stages.
- **Review by certification agent and authorizing official**—OMB Circular A-130, Appendix III, requires that systems be approved, or authorized, to process data in specific environments. Management, operational, and technical controls must be employed to adequately protect the information system. Management and operational security controls can sometimes be outside the scope of the contract,

as the developer, in most cases, cannot be responsible for the organization's implementation of these security controls. The technical security control functional and assurance specifications must be contained in the contract with the developer. These security controls should be factored into the development of the technical specifications. The authorizing official (AO) can take these assumptions into account when deciding on the adequacy of the total set of security controls for reducing the residual risks to an acceptable level.

- **Cyclical nature of the process**—Security steps in the development/acquisition phase may need to be addressed cyclically. These security steps interrelate and build on each other. Depending on the size and complexity of the system, these steps may be performed often as ideas are refined.
- **Evaluation and acceptance**—The system evaluation plan and appropriate acceptance criteria are developed in the development/acquisition phase. The solicitation should be designed for evaluation, which should include testing and analysis. Specifications should be written in a way to make it easy to clearly determine if the implemented system complies with the specification. In general, two separate activities require security testing—contract acceptance and certification and accreditation (C&A).
- **Request for proposal (RFP) development**—An RFP enables an organization to make a best-value decision based on an offeror's proposal. One strength of the RFP process is the flexibility it provides the government and the offeror to negotiate a contract that best meets the government's needs. The organization can identify needed information security features, procedures, and assurances in many ways. An RFP can be a flexible document. Guidance on acquisition alternatives should be obtained from the organization's acquisition office or the contracting officer.
- **Security specifications and statement of work development**—Security specifications and the statement of work (SOW) are based on the requirements analysis. The specifications provide details of what the system is supposed to do. Specifications should also be written independently of the implementation mechanisms, strategy, and design. In other words, the specifications should state what the system is to do, not how. The developer's implementation of the system in conformance with the specifications can and should be tested. This implies that well-written specifications are those that can be tested. The SOW details what the developer must do in the performance of the contract. Documentation developed under the contract, for example, is specified in the SOW. Security assurance requirements, which detail many aspects of the processes the developer follows and what evidence must be provided to assure the organization that the processes have been conducted correctly and completely, may also be specified in the SOW.

Summary

Chapter 1 defined the basic assurance terms and developed the security design principles for software assurance. Risk management was defined and the basics of risk management were reviewed based on NIST SP 800-30. With this foundation, the

system and software development life cycles were reviewed and a number of the prominent software assurance development life cycle methodologies were presented. The chapter concluded with a review of relevant assurance standards, legal issues, security models, and acquisition guidelines for software assurance projects.

Assessment Questions

1. The level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software, is the definition of:
 - A. Software risk
 - B. Software impact
 - C. Software assurance
 - D. Software accountability
2. Seven complementary elements that support information assurance are confidentiality, integrity, availability, authentication, authorization, accountability, and:
 - A. Repudiation
 - B. Auditing
 - C. Operations
 - D. Acquisition
3. A form of confidentiality breach that is accomplished by studying the volume, rate, source, and destination of transmitted messages is:
 - A. Inference analysis
 - B. Covert channel analysis
 - C. Messaging analysis
 - D. Traffic analysis
4. An unauthorized and unintended communication path that provides for exchange of information is a:
 - A. Secret link
 - B. Covert channel
 - C. Covert encryption
 - D. Communication pipe
5. The ability of an entity to use and correlate information protected at one level of security to uncover information that is protected at a higher security level is called:
 - A. Inference
 - B. Knowledge acquisition
 - C. Covert channeling
 - D. Cryptanalysis

6. Confidentiality, integrity, and availability are the principal components of information system security. The reverse of these concepts is:
 - A. Disclosure, authentication, and destruction
 - B. Disclosure, alteration, and destruction
 - C. Encryption, alteration, and destruction
 - D. Disclosure, alteration, and disposal
7. The testing or reconciliation of evidence of a user's identity is:
 - A. Authorization
 - B. Accountability
 - C. Auditing
 - D. Authentication
8. An ongoing activity that examines either the system or the users, such as intrusion detection, is:
 - A. Auditing
 - B. Monitoring
 - C. Accounting
 - D. Eavesdropping
9. A set of records that collectively provides documentary evidence of processing used to aid in tracing from original transactions forward to related records and reports, and/or backward from records and reports to their component source transactions, is called:
 - A. Data library
 - B. Data dictionary
 - C. Audit trail
 - D. Monitor data
10. The ability to determine the actions and behaviors of a single individual within a system and to identify that particular individual is:
 - A. Authentication
 - B. Accountability
 - C. Authorization
 - D. Nonrepudiation
11. Economy of mechanism, separation of duties, fail-safe, open design, and psychological acceptability are 5 of the 11:
 - A. Security design principles
 - B. Security bible principles
 - C. Defense-in-depth principles
 - D. Security trade-off principles

12. What principle maintains that an individual, process, or other type of entity should be given the minimum privileges and resources for the minimum period of time required to complete a task?
 - A. Separation of duties
 - B. Complete mediation
 - C. Least privilege
 - D. Limited access
13. The principle that requires that completion of a specified sensitive activity or access to sensitive objects is dependent on the satisfaction of multiple conditions is:
 - A. Defense-in-depth
 - B. Fail-safe
 - C. Economy of mechanism
 - D. Separation of duties
14. The application of multiple layers of protection wherein a subsequent layer will provide protection if a previous layer is breached is:
 - A. Defense-in-depth
 - B. Weakest link
 - C. Fail-safe
 - D. Control analysis
15. The condition that every request by a subject to access an object in a computer system must undergo a valid and effective authorization procedure, and must not be suspended or become capable of being bypassed, is known as:
 - A. Fail-safe
 - B. Complete mediation
 - C. Economy of mechanism
 - D. Accountability
16. According to the National Institute for Standards and Technology (NIST) Special Publication (SP) 800-30, the three major components of risk management are:
 - A. Risk assessment, risk analysis, and evaluation
 - B. Risk assessment, risk mitigation, and risk reduction
 - C. Risk assessment, risk determination, and evaluation
 - D. Risk assessment, risk mitigation, and evaluation
17. Which of the following steps are components of the NIST SP 800-30 risk assessment process?
 - A. System characterization, threat elimination, vulnerability identification, and control analysis
 - B. System characterization, threat identification, vulnerability identification, and control implementation

- C. System characterization, threat identification, vulnerability identification, and control analysis
 - D. System characterization, threat identification, vulnerability reduction, and control analysis
18. In the NIST SP 800-30 risk assessment process, which one of the following is the likelihood rating in which a highly motivated and capable threat-source will exploit an existing vulnerability? In addition, there are also ineffective controls to prevent exploitation of the associated vulnerability.
- A. Medium
 - B. High
 - C. Low
 - D. Intermediate
19. In the NIST SP 800-30 risk assessment process, the type of analysis that is more easily accomplished and provides identifiable areas for immediate improvement, but does not provide specific magnitudes of measures, is called:
- A. Qualitative analysis
 - B. Measurable analysis
 - C. Quantitative analysis
 - D. Qualified analysis
20. NIST SP 800-30 risk mitigation options include which one of the following groups?
- A. Risk assumption, risk avoidance, risk planning, and research and development
 - B. Risk assumption, risk avoidance, risk planning, and risk elimination
 - C. Risk assumption, risk avoidance, risk deterrence, and research and development
 - D. Risk calculation, risk avoidance, risk planning, and research and development
21. The categories of controls to mitigate risks are:
- A. Administrative, management, operational, and a combination of these
 - B. Technical, logical, operational, and a combination of these
 - C. Technical, management, qualitative, and a combination of these
 - D. Technical, management, operational, and a combination of these
22. The Microsoft Security Risk Management Discipline (SRMD) comprises which one of the following groups of steps?
- A. Aggregation, development and implementation, operating
 - B. Assessment, development and implementation, operating
 - C. Assessment, evaluation, operating
 - D. Assessment, development and implementation, maintenance
23. The LeGrand Vulnerability-Oriented Risk Management method includes which one of the following groups of steps?

- A. Risk mitigation, vulnerability management, adherence to security standards and policies
 - B. Risk assessment, vulnerability management, adherence to operational standards and policies
 - C. Risk assessment, vulnerability elimination, adherence to security standards and policies
 - D. Risk assessment, vulnerability management, adherence to security standards and policies
24. The Morana Risk Management Activities include which one of the following groups of steps that map onto the software development life cycle?
- A. Requirements, architecture and design, development, and testing
 - B. Assessment, architecture and design, development, and testing
 - C. Requirements, vulnerability management, development, and testing
 - D. Requirements, architecture and design, vulnerability management, and testing
25. In black box testing of information systems:
- A. The testing team is provided full knowledge of the resources to be tested.
 - B. The testing team is provided partial knowledge of the resources to be tested and has to acquire some information on its own.
 - C. The testing team is provided no knowledge of the resources to be tested and has to acquire information on its own.
 - D. The testing team is not permitted direct access to the resources to be tested.
26. The Information Assurance Technical Framework (IATF) document 3.1 defines the five phases of the system development life cycle (SDLC) as:
- A. Initiation, development/acquisition, implementation, operation/maintenance, upgrade
 - B. Initiation, evaluation, implementation, operation/maintenance, disposal
 - C. Initiation, development/acquisition, assessment, operation/maintenance, disposal
 - D. Initiation, development/acquisition, implementation, operation/maintenance, disposal
27. The IATF document 3.1 stresses the importance of which one of the following groups to provide information assurance?
- A. People, operations, and technology
 - B. Assessment, operations, and technology
 - C. People, security, and technology
 - D. People, operations, and management
28. In NIST SP 800-64, the process for determining the need for property or services and ending with contract completion and closeout is defined as:
- A. Initiation

- B. Acquisition
 - C. Subcontracting
 - D. Development
29. The U.S. federal act that was passed to “provide a comprehensive framework for ensuring the effectiveness of information security controls over information resources that support Federal operations and assets” and “to provide for development and maintenance of minimum controls required to protect Federal information and information systems” is known as the:
- A. Gramm-Leach-Bliley Act (GLB) Act
 - B. Sarbanes-Oxley (SOX) Act
 - C. Federal Information Security Management Act (FISMA)
 - D. Health Insurance Portability and Accountability Act (HIPAA)
30. Which one of the following specifies that U.S. federal government agencies plan for security, ensure that appropriate officials are assigned security responsibility, review the security controls in their information systems, and authorize system processing prior to operations and periodically thereafter?
- A. Sarbanes-Oxley (SOX)
 - B. U.S. Office of Management and Budget (OMB) Circular A-130
 - C. Health Insurance Portability and Accountability Act (HIPAA) Final Security Rule
 - D. Gramm-Leach-Bliley
31. Which NIST publication is used to identify and categorize U.S. federal information and information systems?
- A. Federal Information Processing Standard (FIPS) 200
 - B. Federal Information Processing Standard (FIPS) 300
 - C. Federal Information Processing Standard (FIPS) 199
 - D. Federal Information Processing Standard (FIPS) 198
32. The right of an individual to protection from unauthorized disclosure of personally identifiable information (PII) is the definition of:
- A. Security
 - B. Confidentiality
 - C. Authorization
 - D. Privacy
33. Notice, choice, access, security, and enforcement are fundamental principles of:
- A. Privacy
 - B. Assurance
 - C. Access
 - D. Authorization

34. The U.S. act that was passed to regulate corporate financial practices and reporting is:
- A. Sarbanes-Oxley (SOX)
 - B. U.S. Office of Management and Budget (OMB) Circular A-130
 - C. Health Insurance Portability and Accountability Act (HIPAA)
 - D. Gramm-Leach-Bliley
35. The U.S. Financial Services Modernization Act, Public Law 106-102, is also known as:
- A. Sarbanes-Oxley (SOX)
 - B. U.S. Office of Management and Budget (OMB) Circular A-130
 - C. FISMA
 - D. Gramm-Leach-Bliley
36. Which U.S. health care–related law addresses protection for transmitted data, protection for data at rest, physical protection, and administrative procedures?
- A. Health Insurance Portability and Accountability Act (HIPAA) Final Privacy Rule
 - B. Health Insurance Portability and Accountability Act (HIPAA) Final Security Rule
 - C. Health Insurance Portability and Accountability Act (HIPAA) Transactions and Code Sets Rule
 - D. Health Insurance Portability and Accountability Act (HIPAA) Final Authorization Rule
37. Which standard addresses credit card and cardholder authentication and is organized as 12 requirements under 6 logically consistent control objectives?
- A. Payment Card Industry (PCI) Data Privacy Standard (DPS)
 - B. Payment Card Industry (PCI) Data Confidentiality Standard (DCS)
 - C. Payment Card Industry (PCI) Data Security Standard (DSS)
 - D. Payment Card Industry (PCI) Data Authorization Standard (DAS)
38. A high-level design structure comprising abstract-level components of the required software system functionality and descriptions of their interactions is the definition of:
- A. Software specification
 - B. Software assurance
 - C. Software requirements
 - D. Software architecture

39. A software architecture can be considered a composition system comprising the following elements:
- A. Components, connectors, operators
 - B. Components, requirements, operators
 - C. Specifications, connectors, operators
 - D. Components, connectors, links
40. Software architecture systems can also be represented by a component model that includes the following concepts:
- A. Binding point, operators, glue code, separation of application and communication
 - B. Binding point, connector, glue code, separation of application and communication
 - C. Binding point, connector, authorization, separation of application and communication
 - D. Binding point, connector, glue code, requirements
41. Some typical software architecture styles include:
- A. Pipes and filters, layered, P-level, heterogeneous
 - B. Matrix, layered, N-tiered, homogeneous
 - C. Pipes and filters, layered, N-tiered, heterogeneous
 - D. Matrix, layered, N-tiered, heterogeneous
42. The document that lists a number of important architectural design assurance objectives—including: the architecture should provide predictable execution behavior; the architectural design should ease traceability, verification, validation, and evaluation; and the architecture should eliminate possibilities for violations—is entitled:
- A. The Software Assurance Common Body of Knowledge (CBK), developed by the U.S. Department of Homeland Security Software
 - B. The Microsoft Trustworthy Security Development Life Cycle Paradigm
 - C. The Microsoft Software Development Baseline Process
 - D. The Comprehensive, Lightweight Application Security Process (CLASP)
43. Which type of software development life cycle process employs extensive reviews, evidence, and formal methods?
- A. Lightweight
 - B. Agile
 - C. Spiral
 - D. Heavyweight

44. Which software development process is characterized by early and frequent delivery of workable and usable software, customer involvement in the development process, iterative development, acceptance of late requirements changes, employment of self-managing teams with appropriate expertise, and delivery of multiple releases?
- A. Agile
 - B. Evolutionary
 - C. Waterfall
 - D. Unified
45. Which software development process operates by first developing prototypes that are tested and evaluated and then going back to continue the development process and incorporating feedback obtained during the prototyping phase?
- A. Waterfall
 - B. Spiral
 - C. Evolutionary
 - D. Concurrent release
46. The basic Microsoft Software Development Baseline Process is a spiral approach in that the software requirements and design are reviewed and updated, if necessary, during the implementation phase. The phases of this process are:
- A. Requirements, review, implementation, verification, release, support and servicing
 - B. Requirements, design, implementation, verification, release, support and servicing
 - C. Evaluation, design, implementation, verification, release, support and servicing
 - D. Requirements, design, implementation, verification, release, disposal
47. The Microsoft Security Development Life Cycle (SDL) is based on which one of the following sets of principles?
- A. Secure in requirements, secure by default, secure in deployment, communications
 - B. Secure by design, secure by evaluation, secure in deployment, communications
 - C. Secure by design, secure by default, secure in deployment, verification
 - D. Secure by design, secure by default, secure in deployment, communications
48. The Microsoft Security Development Life Cycle (SDL) set of principles is known as:
- A. $SD^3 + C$
 - B. $SD^3 + V$
 - C. $SD^2 + E + C$
 - D. $SD^2 + R + C$

49. Which security process includes 7 key best practices for application security and 30 security activities?
- A. Microsoft Security Development Life Cycle (SDL)
 - B. The Comprehensive, Lightweight Application Security Process (CLASP)
 - C. Touchpoints for Software Security
 - D. Team Software Process (TSP)-Secure
50. What mechanism in the Comprehensive, Lightweight Application Security Process (CLASP) provides a classification structure that enables development teams to find vulnerability information acquired from different historical views and sources of data?
- A. Vulnerability View
 - B. Vulnerability Resource
 - C. Vulnerability Module
 - D. Vulnerability Lexicon
51. To provide guidelines for employing information assurance processes into the security development life cycle (SDL), CLASP is structured into:
- A. Views, resources, and vulnerability use cases
 - B. Views, principles, and vulnerability use cases
 - C. Views, resources, and security use cases
 - D. Views, activities, and vulnerability use cases
52. The five CLASP views are:
- A. Concepts view, resource-based view, activity-assessment view, activity-implementation view, vulnerability view
 - B. Concepts view, role-based view, resource-assessment view, activity-implementation view, vulnerability view
 - C. Concepts view, role-based view, activity-assessment view, activity-implementation view, vulnerability view
 - D. Requirements view, role-based view, activity-assessment view, activity-implementation view, vulnerability view
53. When a system is under attack, one of the following includes these activities: Employ static analysis of source code; conduct risk analysis of architecture and design; perform penetration testing; and categorize abuse cases by analyzing system behavior. Which one?
- A. Comprehensive, Lightweight Application Security Process (CLASP)
 - B. Microsoft Security Development Life Cycle
 - C. Team Software Process (TSP)-Secure
 - D. Seven Touchpoints for Software Security

54. The Carnegie Mellon University Software Engineering Institute (SEI) and CERT Coordination Center (CERT/CC) have developed a method designed to estimate the probability of vulnerabilities in production software and to minimize or eliminate software vulnerabilities. This method is called:
- A. Team Software Process (TSP)-Secure
 - B. Digital Design Process
 - C. Design Capability Maturity Model (CMM)
 - D. Seven Touchpoints for Software Security
55. Which one of the following provides the owner with a legally enforceable right to exclude others from practicing a covered invention for a specified period of time?
- A. Copyright
 - B. Patent
 - C. Warranty
 - D. Trade Secret
56. In the U.S., patent law protects which one of the following groups of items?
- A. Original works of authorship, ornamental designs, and new varieties of plants
 - B. Inventions and processes; words, names, or symbols; and new varieties of plants
 - C. Inventions and processes, ornamental designs, and words, names, or symbols
 - D. Inventions and processes, ornamental designs, and new varieties of plants
57. Which one of the following “secures and maintains the confidentiality of proprietary technical or business-related information that is adequately protected from disclosure by the owner”?
- A. Trade secret
 - B. Trademark
 - C. Warranty
 - D. Patent
58. The International Organization for Standardization (ISO) series 27000 (2700X) standards are dedicated to the field of information system security. Which one of the ISO 27000 series is designed to “provide a model for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving an Information Security Management System (ISMS)”?
- A. 27002
 - B. 27004
 - C. 27001
 - D. 27006
59. The ISO 27000 series emphasizes the PDCA cycle. PDCA stands for:
- A. Plan-Do-Check-Act
 - B. Prepare-Do-Check-Act

- C. Plan-Develop-Check-Act
 - D. Plan-Do-Certify-Act
60. The Code of Practice for Information Security Management is a repackaged version of (ISO) 17779:2005. It is designed to serve as a single source for best practices in the field of information security and presents a range of controls applicable to most situations. It is:
- A. ISO 27001
 - B. ISO 27002
 - C. ISO 27003
 - D. ISO 27006
61. What provides a minimum standard for Web application security and summarizes the primary Web application security vulnerabilities based on input from a variety of information system security experts?
- A. The British Standards Institute (BSI) 7799
 - B. ISO 27002
 - C. The Open Web Application Security Project (OWASP) Top Ten Project
 - D. TSP-Secure
62. In an access matrix, an active entity that is seeking rights to a resource is called:
- A. Object
 - B. Subject
 - C. Capability
 - D. Grantor
63. The columns of the access matrix are called:
- A. Access control lists (ACLs)
 - B. Capability lists
 - C. Triples
 - D. Properties
64. What model uses a directed graph to specify the rights that a subject can transfer to an object or that a subject can acquire from another subject?
- A. Access matrix
 - B. Information flow
 - C. Take-Grant
 - D. Bell-LaPadula
65. What model was developed to formalize the U.S. Department of Defense (DoD) multi-level security policy?
- A. Biba
 - B. Take-Grant

- C. Clark-Wilson
 - D. Bell-LaPadula
66. What model was developed to address the first integrity goal of protecting data from modification by unauthorized users?
- A. Bell-LaPadula
 - B. Biba
 - C. Clark-Wilson
 - D. Information flow
67. What model is built on the state machine concept and addresses only the confidentiality of classified material?
- A. Bell-LaPadula
 - B. Biba
 - C. Clark-Wilson
 - D. Information flow
68. The * (star) property of which one of the following models states that writing of information by a subject at a higher level of sensitivity to an object at a lower level of sensitivity is not permitted (no write-down)?
- A. Take-Grant
 - B. Biba
 - C. Clark-Wilson
 - D. Bell-LaPadula
69. Which one of the following is the * (star) integrity axiom of the Biba model?
- A. A subject at one level of integrity is not permitted to observe (read) an object of a lower integrity (no read-down).
 - B. An object at one level of integrity is not permitted to modify (write to) an object of a higher level of integrity (no write-up).
 - C. A subject at one level of integrity is prohibited from invoking a subject at a higher level of integrity.
 - D. Integrity labels are required to verify the integrity of an object.
70. In the Clark-Wilson model, an activity that confirms that all constrained data items are in valid states of integrity is:
- A. Invocation procedure
 - B. Unconstrained data item procedure
 - C. Transformation procedure
 - D. Integrity verification procedure
71. What model describes those characteristics of security engineering processes essential to ensure good security engineering?
- A. Systems Security Engineering Capability Maturity Model (SSE-CMM)

- B. Capability Maturity Model Integration (CMMi)
 - C. Bell-LaPadula model
 - D. Systems Engineering Capability Maturity Model (SE-CMM)
72. The Trusted Platform Module (TPM), the Software Stack (TSS), and the Trusted Network Connect (TNC) specifications are key elements of the:
- A. Trusted Computing Base (TCB)
 - B. Trusted Computing Group (TCG)
 - C. Security Boundary
 - D. Security Perimeter
73. The total combination of protection mechanisms within a computer system, which includes the hardware, software, and firmware that are trusted to enforce a security policy, defines the:
- A. Trusted platform module (TPM)
 - B. Security perimeter
 - C. Trusted path
 - D. Trusted computing base (TCB)
74. According to the U.S. Department of Defense (DoD) and Department of Homeland Security (DHS) Acquisition and Outsourcing Working Group, the major phases of the software assurance acquisition project are:
- A. Planning, contracting, monitoring and acceptance, and follow-on
 - B. Requirements, contracting, monitoring and acceptance, and follow-on
 - C. Planning, contracting, implementation, and follow-on
 - D. Planning, contracting, monitoring and acceptance, and auditing
75. According to NIST Special Publication 800-64, "Security Considerations in the System Development Life Cycle," "all stages of the process of acquiring property or services, beginning with the process for determining the need for the property or services and ending with contract completion and closeout" is the definition of:
- A. Contracting
 - B. Acquisition
 - C. Planning
 - D. Purchasing

