CSSLP Notes 2016

Michiel Kalkman

CSSLP Study Notes

Secure Software Concepts

1 - General Security Concepts

The term CIA is commonly used in the security industry to refer to confidentiality, integrity, and availability.

A key element in audit logs is the employment of a monitoring, detection, and response process. Without mechanisms or processes to "trigger" alerts or notifications to admins based on particular logged events, the value of logging is diminished or isolated to a post-incident resource instead of contributing to an alerting or incident prevention resource.

The Simple Security Rule is just that: the most basic of security rules. It basically states that in order for you to see something, you have to be authorized to see it.

Adversary Types

- Unstructured
- Structured (DDoS, Phishing, Scanning)
- Highly-structured (Criminal enterprise)
- Nation-State (APT)

Summary

- Information assurance and information security place the security focus on the information and not the hardware or software used to process it.
- The original elements of security were confidentiality, integrity, and availability - the "CIA" of security.
- Authentication, authorization, auditability, and non-repudiation have been added to CIA to complete the characterization of operational security elements.
- Systems have a set of characteristics; session management, exception management, and configuration management provide the elements needed to secure a system in operation.
- A series of secure design principles describe the key characteristics associated with secure systems.
- Security models describe key aspects of system operations with respect to desired operational characteristics, including preservation of confidentiality and integrity.
- The Bell-LaPadula security model preserves confidentiality and includes the simple security rule, the *-property, and the concept of "no read up, no write down."
- The Biba integrity model preserves integrity and includes the concept of "no write up and no read down."
- Access control models are used to describe how authorization is implemented in practice.
- Understanding the threat environment educates the software development team on the security environment the system will face in production.

2 - Risk Management

The purpose of risk management is to improve the future, not explain the past.

General Terms

The following terms are general terms associated with risk management.

Risk Risk is the possibility of suffering harm or loss.

Residual risk Residual risk is the risk that remains after a control is utilized and reduces the specific risk associated with a vulnerability. This is the level of risk that must be borne by the entity.

Total risk The sum of all risks associated with an asset, a process, or even a business is called the total risk.

Risk management Risk management is the overall decision-making process of identifying threats and vulnerabilities and their potential impacts, determining the costs to mitigate such events, and deciding what actions are cost effective for controlling these risks.

Risk assessment Risk assessment is the process of analyzing an environment to identify the risks (threats and vulnerabilities) and mitigating actions to determine (either quantitatively or qualitatively) the impact of an event that would affect a project, program, or business. It is also sometimes referred to as risk analysis.

Asset An asset is any resource or information an organization needs to conduct its business.

Vulnerability A vulnerability is any characteristic of an asset that can be exploited by a threat to cause harm. Your system has a security vulnerability, for example, if you have not installed patches to fix a cross-site scripting (XSS) error on your website.

Attack The instance of attempting to perform undesired or unauthorized activities via a vulnerability.

Impact Impact is the loss resulting when a threat exploits a vulnerability. A malicious hacker (the threat) uses an XSS tool to hack your unpatched website (the vulnerability), stealing credit card information that is used fraudulently. The credit card company pursues legal recourse against your company to recover the losses from the credit card fraud (the impact).

Threat A threat is any circumstance or event with the potential to cause harm to an asset. For example, a malicious hacker might choose to hack your system by using readily available hacking tools.

Mitigate The term mitigate refers to any action taken to reduce the likelihood of a threat occurring.

Control A control is a measure taken to detect, prevent, or mitigate the risk associated with a threat. The term control is also called countermeasure or safeguard.

Qualitative risk assessment Qualitative risk assessment is the process of subjectively determining the impact of an event that affects a project, program, or business. Completing the qualitative risk assessment usually involves the use of expert judgment, experience, or group consensus to complete the assessment.

Quantitative Terms

Quantitative risk assessment Quantitative risk assessment is the process of objectively determining the impact of an event that affects a project, program, or business. Quantitative risk assessment usually involves the use of metrics and models.

Single loss expectancy (SLE) The single loss expectancy (SLE) is the monetary loss or impact of each occurrence of a threat.

Exposure factor Exposure factor is a measure of the magnitude of loss of an asset. Used in the calculation of single loss expectancy.

Annualized rate of occurrence (ARO) Annualized rate of occurrence (ARO) is the frequency with which an event is expected to occur on an annualized basis.

Annualized loss expectancy (ALE) Annualized loss expectancy (ALE) is how much an event is expected to cost per year.

Security and privacy are often confused—while security describes the protective attributes of data in a system, privacy describes the attributes that define with whom the data within a system is shared (or not shared).

Types of Controls

Controls can be classified based on the types of actions they perform. Three classes of controls exist:

- Administrative
- Technical
- Physical

For each of these classes, there are four types of controls:

- Preventative (deterrent) (primary defensive controls)
- Detective (after-the-fact)
- Corrective (recovery) (after-the-fact)
- Compensating (after-the-fact)

Failure Mode Effects Analysis

Failure mode effects analysis (FMEA) is a structured methodology for the assessment of failure modes and their effects on the system. FMEAs allow engineers to rank risks in a system.

Quantitative Risk Management

Whereas qualitative risk assessment relies on judgment and experience, quantitative risk assessment applies historical loss information and trends in an attempt to predict future losses.

```
SLE = asset value * exposure factor
ARO = number of events / number of years
ALE = SLE * ARO

ROI (%) = (Avoided Loss - Control Cost) / (Control Cost) * 100
ROI (Time) = (Avoided Annual Loss) / (Annual Control Cost)
```

Software Engineering Institute Model (SEI Model Steps)

- *Identify* Examine the system, enumerating potential risks.
- *Analyze* Convert the risk data gathered into information that can be used to make decisions. Evaluate the impact, probability, and timeframe of the risks. Classify and prioritize each of the risks.
- *Plan* Review and evaluate the risks and decide what actions to take to mitigate them. Implement the plan.
- *Track* Monitor the risks and the mitigation plans. Trends may provide information to activate plans and contingencies. Review periodically to measure progress and identify new risks.
- *Control* - Make corrections for deviations from the risk mitigation plans. Correct products and processes as required. Changes in business procedures may require adjustments in plans or actions, as do faulty plans and risks that become problems.

Summary

- The vocabulary of risk management is an important element in communicating risks and controls to facilitate cross-cutting activities needed to manage risk in the enterprise.
- There are two primary forms of risk management methodology: qualitative and quantitative.
- The purpose of risk management is to influence the future and reduce future risk.
- The foundational element in determining value associated with risk is information criticality.

3 - Security Policies and Regulations

For a CSSLP, it is important to understand the various sources of security requirements, as they need to be taken into account when executing software development. It is also important to not mistake security functionality for the objective of secure software development. Security functions driven by requirements are important, but the objective of a secure development lifecycle process is to reduce the number and severity of vulnerabilities in software.

FISMA Risk Management Framework (RMF)

- Inventory of systems
- Categorize information and systems according to risk level
- Security controls
- Certification and accreditation of systems (including risk assessment and system security plans)
- Training

NIST SP 800-39 RMF

- Categorize information systems
- Select security controls
- Implement security controls
- Assess security controls
- Authorize information systems
- Monitor security controls

Gramm-Leach-Bliley Act (GLBA) is designed to protect consumers' personal financial information (PFI)

1. Financial Privacy Rule
2. Safeguards Rule
3. Pretexting Provision

HIPPA deals with PHI. Health Information Technology for Economic and Clinical Health Act (HITECH Act) deals with privacy enhancements for electronic PHI.

Payment Card Industry Data Security Standard (PCI DSS),

- Data Security Standard (PCI DSS)
- Payment Application Data Security Standard (PA DSS)
- PIN Transaction Security (PTS)

The privacy policy is the high-level document that describes the principles associated with the collection, storage, use, and transfer of personal information within the scope of business.

Common PII Elements

The following items are commonly used to identify a specific individual and are, hence, considered PII:

- Full name (if not common)
- National identification number (i.e., SSN)
- IP address (in some cases)

- Home address
- Motor vehicle registration plate number
- Driver's license or state ID number
- Face, fingerprints, or handwriting
- Credit card and bank account numbers
- Date of birth
- Birthplace
- Genetic information

A study by Carnegie Mellon University found that nearly 90 percent of the U.S. population could be uniquely identified with only gender, date of birth, and ZIP code.

PHI and associated medical data are sought after by cybercriminals because they contain both insurance information and financial responsibility information, including credit cards, both of which can be used in fraud. In addition, there is sufficient PII for an identity to be stolen, making health records a highly valued source of information for cybercriminals.

The term data protection is typically associated with the European Union Data Protection Directive (EUDPD). The EUDPD equates personal data protection as a basic human right and places strict rules on firms using personal data.

Safe Harbor Principles

The Safe Harbor principles require firms provide seven elements:

- *Notice* Customers must be informed that their data is being collected and how it will be used.
- *Choice* Customers must have the ability to opt out of the collection and forward transfer of the data to third parties.
- *Onward Transfer* Transfers of data to third parties may only occur to other organizations that follow adequate data protection principles.
- *Security* Reasonable efforts must be made to prevent loss of collected information.
- *Data Integrity* Data must be relevant and reliable for the purpose it was collected for.
- *Access* Customers must be able to access information held about them and correct or delete it if it is inaccurate.
- *Enforcement* There must be effective means of enforcing these rules.

Common ISO standards

ISO/IEC	
9126	Software Engineering Product Quality. Multipart series standard.
10746	Information Technology – Open distributed processing. Multipart series standard.
12207	Systems and Software Engineering – Software lifecycle processes.
14143	Information Technology – Software measurement – Functional size measurement. Multipart series standard.
15026	Systems and Software Assurance. Multipart series standard.
15288	Systems and Software Engineering – System lifecycle processes.
15408	Evaluating Criteria for IS Security (Common Criteria).
21827	Information Technology – Security techniques – Systems security engineering – Capability Maturity Model (SSE-CMM).
27001	Information Security Management System (ISMS) Overview and Vocabulary.
27002	Code of Practice for Information Security Management.
27003	Information Security Management System Implementation Guidance.
27004	Information Security Management – Measurement.

ISO/IEC	
27005	Information Security Risk Management.

15408 Common Criteria

ISO/IEC 15408 (Common Criteria) Evaluation Assurance Levels (EALs)

The following table illustrates the levels of assurance associated with specific evaluation assurance levels correlated with the Common Criteria.

Level	TOE Assurance
EAL 1	Functionally tested
EAL 2	Structurally tested
EAL 3	Methodically tested and checked
EAL 4	Methodically designed, tested, and reviewed
EAL 5	Semiformally designed and tested
EAL 6	Semiformally verified, designed, and tested
EAL 7	Formally verified, designed, and tested

Security Frameworks

COBIT

Control Objectives for Information and Related Technology (COBIT) is a framework designed to assist management in bridging the gap between control requirements, technical issues, and business risks.

- Principle 1: Meeting Stakeholder Needs
- Principle 2: Covering the Enterprise End to End
- Principle 3: Applying a Single, Integrated Framework
- Principle 4: Enabling a Holistic Approach
- Principle 5: Separating Governance from Management

COSO

Committee of Sponsoring Organizations of the Treadway Commission (COSO) is a joint initiative of five private-sector organizations. Enterprise Risk Management,

- Control environment
- Risk assessment
- Control activities
- Information and communication
- Monitoring

ITIL

- ITIL Service Strategy
- ITIL Service Design
- ITIL Service Transition
- ITIL Service Operation
- ITIL Continual Service Improvement

Frameworks

Zachman - The Zachman Framework is a highly structured and formal method of defining an enterprise.

SABSA - The Sherwood Applied Business Security Architecture (SABSA) is a framework and methodology for developing risk-driven enterprise information security architectures and for delivering security infrastructure solutions that support critical business initiatives.

SDLC - Software development lifecycle (SDLC) is a generic term describing a process imposed on the development of software

SEI CMMI - Software Engineering Institute (SEI), the Capability Maturity Model Integration (CMMI) is a process metric model that rates the process maturity of an organization on a 1 to 5 scales. CMMI addresses three primary areas: product and service development, service establishment and management, and product and service acquisition

OWASP - Open Web Application Security Project

OCTAVE Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) is a suite of tools, techniques, and methods for risk-based information security assessment. OCTAVE is designed around three phases: build asset-based threat profiles, identify infrastructure vulnerabilities, and develop a security strategy.

BSI - Build Security In

Trusted Computing

Trusted Platform Module (TPM). The TPM can hold an encryption key that is not accessible to the system except through the TPM chip. Principles,

- Security
- Privacy
- Interoperability
- Portability of data
- Controllability
- Ease of use

The ring model was devised to provide a system-based method of protecting data and functionality from errors and malicious behavior. The ring model is composed of a series of hierarchical levels based on given security levels.

A reference monitor is an access control methodology where a reference validation mechanism mediates the interaction of subjects, objects, and operations. In a computer system architecture, a subject is either a process or a user, and an object is an item on the system, typically in the form of a file or socket. Subjects interact with objects via a set of operations.

For a reference validation mechanism to be a reference monitor, it must possess three qualities:

- It must always be invoked and there is no path around it. This is called complete mediation.
- It must be tamper-proof.
- It must be small enough to be verifiable

A protected object is one whose existence may be known but cannot be directly interacted with. Specifically, any interaction must be done through a protected subsystem.

Trusted Computing Base

The term trusted computing base (TCB) is used to describe the combination of hardware and software components that are employed to ensure security. The Orange Book, which is part of the U.S. government's Trusted Computer System Evaluation Criteria (TCSEC), provides a formal definition for TCB:

The totality of protection mechanisms within it, including hardware, firmware, and software, the combination of which is responsible for enforcing a computer security policy.

The trusted computing base should not be confused with trustworthy computing or trusted computing.

The Trusted Platform Module (TPM) is an implementation of specifications detailing secure cryptostorage on a chip.

The Microsoft Trustworthy Computing Initiative is a company-wide effort to address concerns over security and privacy. From a white paper in 2002, Microsoft CTO Craig Mundie established four pillars of the company's Trustworthy Computing Initiative. In the years since, Microsoft has internalized these objectives into all of their processes and products. The four key pillars are security, privacy, reliability, and business integrity.

Summary

- Regulations and compliance form the basis of many security efforts.
- FISMA is the federal law governing information security for government systems in the United States.
- Sarbanes-Oxley dictates internal controls for public firms in the United States.
- HIPAA and HITECH Act govern information security with respect to medical records in the United States.
- PCI DSS is a set of standards that apply to the credit card issuers, including processors.
- Intellectual property is protected through patents, copyrights, trademarks, and trade secrets.
- Privacy is the principle of controlling information about one's self.
- Personally identifiable information (PII) should be protected in systems at all times.
- There are numerous standards from NIST and ISO applicable to software security.
- There are a wide variety of frameworks covering both process and product security that can be employed in the development effort.
- Common process frameworks include COBIT, ITIL, CMMI, and SDLC.
- Trusted computing is the set of technologies to improve computer security.
- Computer security models such as the ring model, reference monitor, and protected objects provide concepts to implement security.
- Software acquisition can have an effect on system security, with procurement and contractual implications.

4 - Software Development Methodologies

Security Tenets and Design Principles

Security Tenets

- Confidentiality
- Integrity
- Availability
- Authentication
- Authorization
- Auditability
- Session Management
- Exception Management
- Configuration Management

Security Design Principles

- Least Privilege
- Separation of Duties
- Defense in Depth

- Fail to a Secure State
- Economy of Mechanism
- Complete Mediation
- Open Design
- Least Common Mechanism
- Psychological Acceptability
- Leverage Existing Components
- Weakest Link
- Single Point of Failure

The acronym DREAD refers to a manner of classifying bugs: Damage potential, Reproducibility, Exploitability, Affected user base, and Discoverability.

The term STRIDE refers to sources of threats: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege.

Summary

- A secure development lifecycle is one where security tenets and design principles are built into the process of the development model.
- Adding security features to software is not secure development.
- All development team members should have appropriate security training.
- The inclusion of security processes in the development process occurs at security gates.
- Bugs should be measured and tracked as part of the secure development process.
- Threat modeling is a major tool for understanding threats and mitigation actions in the development process.
- Fuzz testing should be used against all inputs in the software.
- Security tenets can be included in all development models, including agile methods.
- Microsoft has a well-documented and mature SDL that it freely shares to assist others in pursuit of secure development practices.

5 - Policy Decomposition

Three general factors are used in authentication. In order to verify your identity, you can provide something you know, something you have, or something about you (something that you are). When two different factors of this list are used together it is referred to as two-factor authentication.

Most authentication involves one party authenticating the other, as in a bank authenticating an account holder. Mutual authentication involves both parties authenticating each other at the same time, an essential process to prevent some forms of man-in-the-middle attacks.

Common access control models include Mandatory access control (MAC), Discretionary access control (DAC), Role-based access control (RBAC), and Rule-based access control (RBAC).

Summary

- Confidentiality requirements relate to the protection of data from unauthorized disclosure.
- Integrity requirements relate to the protection of data from unauthorized alteration.
- Availability requirements relate to the protection from disruption of authorized access.
- Authentication requirements relate to the identification of a user.
- Authorization requirements relate to controlling the subject-object-activity model.
- Auditing requirements are used to ensure controls are operational and effective.

6 - Data Classification and Categorization

Data states

- At rest, or being stored
- Being created
- Being transmitted from one location to another
- Being changed or deleted

Data usage

- Internal data - Data initialized in the application, used in an internal representation, or computed within the application itself
- Input data - Data read into a system and possibly stored in an internal representation or used in a computation and stored
- Output data - Data written to an output destination following processing

Data owners are responsible for defining data classification, defining authorized users and access criteria, defining the appropriate security controls and making sure they are implemented and operational.

Data custodians are responsible for maintaining defined security controls, managing authorized users and access controls, and performing operational tasks such as backups and data retention and disposal.

NIST FIPS 199 and SP 800-18 provide a framework for classifying data based on impacts across the three standard dimensions: confidentiality, integrity, and availability.

Summary

- Data classification is a risk management tool, with the objective of reducing the costs associated with protecting data.
- Data is in typically one of four states: being stored (at rest), being created, being transmitted from one place to another, and being processed (changed or deleted).
- Data management responsibilities are typically split between data owners and data custodians.
- Data can be characterized and labeled based on its relative sensitivity and business impact.
- Data can be categorized by its formatting and structure,

7 - Requirements

Use cases are constructed of actors representing users and intended system behaviors, with the relationships between them depicted graphically.

A time of check/time of use attack is one that takes advantage of a separation between the time a program checks a value and when it uses the value, allowing an unauthorized manipulation that can affect the outcome of a process.

Complex conditional logic with unhandled states, even if rare or unexpected, can result in infinite loops. It is imperative that all conditions in each nested loop be handled in a positive fashion.

To prevent error conditions from cascading or propagating through a system, each function should practice complete error mitigation, including error trapping and complete handling, before returning to the calling routine.

Secure Coding Standards have been published by the Software Engineering Institute/CERT at Carnegie Mellon University for C, C++, and Java.

A complete SDLC solution ensures systems are secure by design, secure by default, and secure in deployment. A system that is secure by design but deployed in an insecure configuration or method of deployment can render the security in the system worthless.

The requirements traceability matrix (RTM) is a grid that assists the development team in tracking and managing requirements and implementation details. The RTM assists in the documentation of the relationships between security requirements, controls, and test/verification efforts.

Summary

- Functional requirements are those that describe how the software is expected to function.
- Business requirements must be translated into functional requirements that can be followed by designers, coders, testers, and more to ensure they are met as part of the SDLC process.
- Role and user definitions are the statements of who will be using what functionality of the software.
- Objects are items that users (subjects) interact with in the operation of a system.
- An object can be a file, a database record, a system, or a program element.
- Activities or actions are the legal events that a subject can perform on an associated object.
- The subject-object-activity matrix is a tool that permits concise communication about allowed system interactions.
- A use case is a specific example of an intended behavior of the system.
- Misuse or abuse cases can be considered a form of use case illustrating specifically prohibited actions.
- Sequence and timing issues, such as race conditions and infinite loops, influence both design and implementation of data activities.
- Secure coding standards are language-specific rules and recommended practices that provide for secure programming.
- A complete SDLC solution ensures systems are secure by design, secure by default, and secure in deployment.
- The requirements traceability matrix (RTM) is a grid that allows users to track and manage requirements and implementation details.

8 - Design Process

Attack Surface Measurement / Minimization

- Open sockets
- Open remote procedure call (RPC) endpoints
- Open named pipes
- Services
- Services running by default
- Services running as SYSTEM
- Active web handlers (ASP files, HTR files, and so on)
- Active Internet Server Application Programming Interface (ISAPI) filters
- Dynamic webpages (ASP and such)
- Executable virtual directories
- Enabled accounts
- Enabled accounts in admin group
- Null sessions to pipes and shares
- Guest account enabled
- Weak ACLs in the file system
- Weak ACLs in the registry
- Weak ACLs on shares

The attack surface is merely a representation of the potential vulnerability surface associated with the software. Once a baseline is known, the next step is to examine ways that this metric can be lowered.

Theat Modeling

DFD Elements for Threat Modeling (examples):

- External Entities
- Users (by type)
- Other systems
- Data Stores
- Files
- Database
- Registry
- Shared memory
- Queues/stack
- Trust Boundaries
- Users
- File systems
- Process boundaries
- Data Flows
- Function calls
- Remote procedure calls (RPCs)
- Network traffic

Threat Identification

The acronym STRIDE is used to denote the following types of threats:

Threat	Security Property
Spoofing	Authentication
Tampering	Integrity
Repudiation	Nonrepudiation
Information	Disclosure Confidentiality
Denial of Service	Availability
Elevation of Privilege	Authorization

Summary

- The attack surface of an application represents a measure of how many features/items are available for attack.
- The actual measure of an attack surface is useful across the development process to measure improvement on numbers of potential vulnerabilities.
- Threat modeling is a process used to identify and document all of the threats to a system.
- Threat modeling has five phases: define security objectives, system decomposition, threat enumeration, mitigation analysis, and validation of the model.
- STRIDE represents spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege.
- Attack trees provide causal information with respect to attack vectors.
- DREAD represents damage potential, reproducibility, exploitability, affected users, and discoverability.
- When securing vulnerabilities, priority should be afforded to any security control that exists in the

enterprise.
- Attack surface and threat modeling documentation are living documents to communicate security information across the development team.

9 - Design considerations

Confidentiality is the concept of preventing the disclosure of information to unauthorized parties. Keeping secrets secret is the core concept of confidentiality. One of the key elements in the design phase is determining what elements need to be kept secret.

Integrity refers to protecting data from unauthorized alteration. Alterations can come in the form of changing a value or in deleting a value. Integrity builds upon confidentiality.

Availability is defined as a system being available to authorized users when appropriate.

Authentication is the process used to verify to the computer system or network that the individual is who they claim to be.

Authorization - After the authentication system identifies a user to a system, the authorization system takes over and applies the predetermined access levels to the user.

Accounting (Audit) is the function of measuring specific IT activities.

When logging information associated with a failure or error, care must be taken to not disclose any sensitive information in logs. This includes elements such as personally identifiable information (PII) and programmatic elements such as paths and filenames.

The use of legacy code in current projects does not exempt that code from security reviews. All code should receive the same scrutiny, especially legacy code that may have been developed prior to the adoption of secure development lifecycle (SDL) processes.

Summary

- Including confidentiality, integrity, and availability concepts in an application and understanding what the specific requirements are for each of these elements help provide an essential foundation for software.
- Authentication needs to be designed into the system, as this function is a precursor to other aspects of security functionality.
- Whenever possible, the use of enterprise-level functionality for security functions, such as authentication, authorization, and logging, provide enterprise-level operational benefits.
- When logging information associated with a failure or error, care must be taken to not disclose any sensitive information in logs.
- During the design phase, the attack surface analysis and threat model provide information as to the elements that need to be secured in the application.
- Secure design principles can be employed to assist in the development of a comprehensive security solution for an application.
- Design attention should be paid to critical foundational elements such as configuration management, exception management, and session management.

10 - Securing Commonly Used Architecture

Service-Oriented Architecture characteristics,

- Platform neutrality
- Interoperability
- Modularity and reusability

- Abstracted business functionality
- Contract-based interfaces
- Discoverability

Cloud computing is marked by the following characteristics,

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

NIST Special Publication 800-145: The NIST Definition of Cloud Computing, defines four deployment models,

- Private cloud
- Public cloud
- Community cloud
- Hybrid cloud

Summary

- Client server architectures are characterized by a distributed application structure that partitions operations between the providers of a resource or service, called servers, and the requesters, called clients.
- A common utilization of the peer-to-peer model is in file sharing, where machines directly share files without an intermediate storage hub.
- Service-oriented architectures are distributed, modular applications that are platform neutral and have automated interfaces.
- SOAs can involve SOAP, XML, and REST protocols Web services are a form of SOA that use WSDL for interface definitions.
- Rich Internet applications replicate desktop functionality in a web-based form.
- Clients are susceptible to exploitation and remote code injections against the server.
- Radio frequency identification (RFID) is a radio frequency, noncontact means of transferring data between two parties.
- Near-field communication (NFC) is a protocol and set of standards for communication via radio frequency energy over very short distances.
- Cloud computing is a relatively new term in the computer field used to describe an architecture of scalable services that are automatically provisioned in response to demand.
- Software as a Service (SaaS) is a type of cloud computing where the software runs in the cloud on external hardware, and the user derives the benefit through a browser or browser-like interface.
- Platform as a Service (PaaS) is a form of cloud computing that offers a complete platform as a solution to a computing need.
- Infrastructure as a Service (IaaS) is a form of cloud computing that offers a complete platform as a provisioning solution to a computing need.

11 - Technologies

IAM - Identity and Access Management

Security controls are audited annually under Sarbanes-Oxley (SOX) section 404, and IAM controls are certainly security controls. Designing and building IAM controls to support this operational issue is a good business practice.

Passwords and other verification credentials, personal identification number (PIN), passphrases, token values, etc., are secrets and should never be accessible by anyone, including system administrators. Cryptography

allows secrets to remain secret and still be used. If a system can e-mail you your password, it is not stored properly; disclosure should be impossible.

X.509 Digital Certificate Fields

The following fields are included within an X.509 digital certificate,

- *Version number* Identifies the version of the X.509 standard that was followed to create the certificate; indicates the format and fields that can be used.
- *Serial number* Provides a unique number identifying this one specific certificate issued by a particular CA.
- *Signature algorithm* Specifies the hashing and digital signature algorithms used to digitally sign the certificate.
- *Issuer* Identifies the CA that generated and digitally signed the certificate.
- *Validity* Specifies the dates through which the certificate is valid for use.
- *Subject* Specifies the owner of the certificate.
- *Public key* Identifies the public key being bound to the certified subject; also identifies the algorithm used to create the private/public key pair.
- *Certificate usage* Specifies the approved use of the certificate, which dictates intended use of this public key.
- *Extensions* Allow additional data to be encoded into the certificate to expand its functionality. Companies can customize the use of certificates within their environments by using thes

Flow control

One of the requirements of the PCI Data Security Standard is for web applications to either have a web application firewall between the server and users or to perform application code reviews.

Digital Rights Management

- *ccREL* An RDF schema used by the Creative Commons project and the GNU project to express their general public license (GPL) in machine-readable form.
- *ODRL* Open Digital Rights Language, an open standard for an XML-based REL.
- *MPEG-21* Part 5 of this MPEG standard includes an REL.
- *XrML* eXtensible rights Markup Language. XrML began based on work at Xerox in the 1990s.

Databases

Primary keys are used to index and join tables and, as such, can not be obfuscated or encrypted. This is a good reason not to use personally identifiable information (PII) and personal health information (PHI) as keys in a database structure.

Regulations such as GLBA, HIPAA, and PCI DSS can impose protection requirements around certain data elements, such as personally identifiable information (PII) and personal health information (PHI). It is important for members of the design and development team to understand this to avoid operational issues later in the software lifecycle.

Summary

- Authentication is an identity verification process that attempts to determine whether users are who they say they are.

- Identity management is the comprehensive set of services related to managing the use of identities as part of an access control solution.
- There are two main parties in these systems: a relying party (RP) and an identity provider (IdP).
- OpenID was created for federated authentication, specifically to allow a third party to authenticate your users for you by using accounts that users already have.
- The OpenID protocol enables websites or applications (consumers) to grant access to their own applications by using another service or application (provider) for authentication.
- X.509 refers to a series of standards associated with the manipulation of certificates used to transfer asymmetric keys between parties in a verifiable manner.
- Single sign-on makes it possible for a user, after authentication, to have his credentials reused on other applications without the user re-entering the secret.
- Firewalls act as policy enforcement devices, determining whether to pass or block communications based on a variety of factors.
- Proxies act as middlemen and are similar to firewalls in that they can mediate traffic flows.
- Application firewalls use application-level information to make firewall decisions.
- Syslog is an IETF-approved protocol for log messaging.
- DLP solutions act by screening traffic, looking for traffic that meets profile parameters.
- Digital rights management is the series of technologies employed so that content owners can exert control over digital content on their systems.
- The trusted computing base (TCB) of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security.
- The Trusted Platform Module is a hardware implementation of a set of cryptographic functions on a computer's motherboard.
- Malware is a term used to describe software that has malicious intent.
- Code signing is the application of digital signature technology to computer code.
- Compilers convert the source code into a set of processor-specific codes, and linking involves the connecting of various program elements, including libraries, dependency files, and resources.
- Sandboxing is a term for the execution of computer code in an environment designed to isolate the code from direct contact with the target system.
- Embedded systems are dedicated systems where the hardware and software are coupled together to perform a specific purpose.
- Firmware is the name given to software code held in a device.

12 - Common Software Vulnerabilities and Countermeasures

CWE/SANS Top 25—2011 (Current)

1. CWE-89 - SQL Injection
2. CWE-78 - OS Command Injection
3. CWE-120 - Buffer Overflow
4. CWE-79 - Cross-Site Scripting (XSS)
5. CWE-306 - Missing Authentication for Critical Function
6. CWE-862 - Missing Authorization
7. CWE-798 - Hard-Coded Credentials
8. CWE-311 - Missing Encryption of Sensitive Data
9. CWE-434 - Unrestricted Upload of File with Dangerous Type
10. CWE-807 - Reliance on Untrusted Inputs in a Security Decision
11. CWE-250 - Execution with Unnecessary Privileges
12. CWE-352 - Cross-Site Request Forgery (CSRF)
13. CWE-22 - Path Traversal
14. CWE-494 - Download of Code Without Integrity Check
15. CWE-863 - Incorrect Authorization
16. CWE-829 - Inclusion of Functionality from Untrusted Control Sphere

17. CWE-732 - Incorrect Permission Assignment for Critical Resource
18. CWE-676 - Use of Potentially Dangerous Function
19. CWE-327 - Use of a Broken or Risky Cryptographic Algorithm
20. CWE-131 - Incorrect Calculation of Buffer Size
21. CWE-307 - Improper Restriction of Excessive Authentication Attempts
22. CWE-601 - URL Redirection to Untrusted Site ("Open Redirect")
23. CWE-134 - Uncontrolled Format String
24. CWE-190 - Integer Overflow or Wraparound
25. CWE-759 - Use of a One-Way Hash Without a Salt

OWASP Top 10—2013 (Current)

- A1 – Injection
- A2 – Broken Authentication and Session Management
- A3 – Cross-Site Scripting (XSS)
- A4 – Insecure Direct Object References
- A5 – Security Misconfiguration
- A6 – Sensitive Data Exposure
- A7 – Missing Function-Level Access Control
- A8 – Cross-Site Request Forgery (CSRF)
- A9 – Using Known Vulnerable Components
- A10 – Unvalidated Redirects and Forwards

Common XSS uses

- Theft of authentication information from a web application
- Session hijacking
- Deploy hostile content
- Change user settings, including future users
- Impersonate a user
- Phish or steal sensitive information

Only approved cryptographic libraries should be used for encryption. In addition, attention must be paid to algorithms and key lengths. At the time of writing, RSA keys should be >2048 bits.

Summary

- The CWE/SANS Top 25 and OWASP Top 10 lists can be used as a checklist of reminders and as a source for a custom "Top N" list that incorporates internal historical data.
- Injection attacks are some of the most common and severe attacks that are currently being seen in software.
- The SQL injection attack is performed by an attacker inputting a specific string to manipulate the SQL statement to do something other than that intended by the programmer or designer.
- Command injection attacks manipulate the input to cause additional command-level functionality.
- Cross-site scripting and cross-site request forgery attacks are web application attacks that use improperly validated input strings to result in unauthorized and undesired behaviors.
- Failures in the application of cryptography can result in failed protection for data.
- Credentials or other secret data should never be hard-coded in a program.
- Not encrypting all of the sensitive data is a common failure mode.
- All user input should be considered suspect and validated before use.
- The Common Weakness Enumeration (CWE) is a list of software weaknesses created by a community initiative.

- The Common Vulnerabilities and Exposures (CVE) is a list of standard identifiers for known software vulnerabilities that have been found in software.
- Social engineering refers to attacks against the people side of a system.

13- Defensive Coding Practices

Primary mitigations,

- Lock down your environment.
- Establish and maintain control over all of your inputs.
- Establish and maintain control over all of your outputs.
- Assume that external components can be subverted and your code can be read by anyone.
- Use libraries and frameworks that make it easier to avoid introducing weaknesses.
- Use industry-accepted security features instead of inventing your own.
- Integrate security into the entire software development lifecycle.
- Use a broad mix of methods to comprehensively find and prevent weaknesses.

Summary

- Declarative security refers to defining security relations with respect to the container.
- Programmatic security is where the security implementation is embedded into the code itself.
- Cryptographic agility is the ability to manage the specifics of cryptographic function that are embodied in code without recompiling, typically through a configuration file.
- Securing configuration parameters is an important issue when configuration can change programmatic behaviors.
- Memory management is a crucial aspect of code security.
- In managed code applications, the combination of managed code and the intermediate code execution engine takes care of memory management, and type safety makes the tasking easier.
- In unmanaged code situations, the responsibility for memory management is shared between the operating system and the application, with the task being even more difficult because of the issues associated with variable type mismatch.
- Type-safe code will not inadvertently access arbitrary locations of memory outside the expected memory range.
- Locality is a principle that, given a memory reference by a program, subsequent memory accesses are often predictable and are in close proximity to previous references.
- Exception management is the programmatic response to the occurrence of an exception during the operation of a program.
- APIs are significant in that they represent entry points into software.
- A set of primary mitigations have been established over time as proven best practices.

14 - Secure Software Coding Operations

Static code analysis is when the code is examined without being executed. Dynamic analysis is performed while the software is executed, either on a target or emulated system.

Error Mechanism	Review Tasking
Inefficient code	If code is obfuscated or overly complex, simplify it.
SANS Top 25/OWASP	Top 10 /Previously discovered errors All code should be reviewed to specifically prevent recurrence of previous errors. Common error patterns from SANS and OWASP lists should be checked as well.

Error Mechanism	Review Tasking
Errors and exception handling	All functions, all procedures, and all components should fully check and handle exceptions.
Injection flaws	Input validation checks.
Cryptographic calls	Approved libraries, good random numbers.
Unsafe and deprecated function calls	Use only approved functions.
Privilege levels	Least privilege violations.
Logging	Ensure proper logging of errors and conditions.
Secure key information	Handling of keys, PII, and other sensitive data.

Compilers can have flag options, such as Microsoft's /GS compiler switch, which enables stack overflow protection in the form of a cookie to be checked at the end of the function, prior to the use of the return address. Use of these options can enhance code security by eliminating common stack overflow conditions.

Code signing provides a means of authenticating the source and integrity of code. It cannot ensure that code is free of defects or bugs.

Steps to Code Signing

1. The code author uses a one-way hash of the code to produce a digest.
2. The digest is encrypted with the signer's private key.
3. The code and the signed digest are transmitted to end users.
4. The end user produces a digest of the code using the same hash function asthe code author.
5. The end user decrypts the signed digest with the signer's public key.
6. If the two digests match, the code is authenticated and integrity is assured.

Summary

- Code should be inspected during development for weaknesses and vulnerabilities.
- Static code analysis is performed without executing the code.
- Dynamic code analysis involves examining the code under production conditions.
- Code walkthroughs are team events designed to find errors using human-led inspection of source code.
- Software development is a highly automated task, with many tools available to assist developers in efficient production of secure code.
- Integrated development environments provide a wide range of automated functionality designed to make the development team more productive.
- Compilers and tools can be configured to do specific testing of code during the production process, and they need to be integrated into the SDL environment.
- Code can be cryptographically signed to demonstrate both authenticity and integrity.
- The management of the various elements of code, files, and settings requires a configuration

15 - Security Quality Assurance

Standards for Software Quality Assurance

- ISO 9216
- SSE-CMM - Systems Security Engineering Capability Maturity Model (SSE-CMM) is also known as ISO/IEC 21827
- OSSTMM

Functional Testing

Functional software testing is performed to assess the level of functionality associated with the software as expected by the end user. Functional testing is used to determine compliance with requirements in the areas of reliability, logic, performance, and scalability.

Reliability measures that the software functions as expected by the customer at all times. It is not just a measure of availability, but functionally complete availability.

Resiliency is a measure of how strongly the software can perform when it is under attack by an adversary.

Recoverability is the ability of an application to restore itself to expected levels of functionality after the security protection is breached or bypassed.

Steps for Functional Testing

1. Identifying the functions (requirements) that the software is expected to perform
2. Creating input test data based on the function's specifications
3. Determining expected output test results based on the function's specifications
4. .xecuting the test cases corresponding to functional requirements
5. Comparing actual and expected outputs to determine functional compliance

Unit Testing

One of the principle advantages of unit testing is that it is done by the development team and catches errors early, before they leave the development phase.

Integration or Systems Testing

Performance Testing

Security Testing

When testers have access to full knowledge of a system, including source code, it is referred to as white-box testing.

Comparison of Common Testing Types

White-Box Testing	Black-Box Testing
Full knowledge, including source code	Zero knowledge
Assesses software structure and design	Assesses software behavior
Low false positives	High false positives
Logic flaws are detected	Logic flaws are typically not visible

Bug Tracking

Remediation of bugs can take many forms, but typically four states are used:

- Removal of defect
- Mitigation of defect
- Transfer of responsibility
- Ignore the issue

Bug categories,

Category	Description
Bugs	Errors in coding
Flaws	Errors in design
Behavioral anomalies	Issues in how the application operates
Errors and faults	Outcome-based issues from other sources
Vulnerabilities	Items that can be manipulated to make the system operate improperly

The concept of a bug bar is an operational measure for what constitutes a minimum level of quality in the code. The bug bar needs to be defined at the beginning of the project as a fixed security requirement.

Summary

- ISO 9216 details quality in software products.
- ISO 21827 (SSE-CMM) details the processes of secure engineering of systems.
- OSSTMM is a scientific methodology for assessing operational security built upon analytical metrics.
- Reliability is not just a measure of availability, but functionally complete availability.
- Resiliency is a measure of how strongly the software can perform when it is under attack by an adversary.
- Functional testing is used to determine compliance with requirements in the areas of reliability, logic, performance, and scalability.
- Unit testing is the first level of testing and is essential to ensure that logic elements are correct and that the software under development meets the published requirements.
- Systems, or integration, testing is needed to ensure that the overall system is compliant with the system-level requirements.
- White-box testing is performed on a system with full knowledge of the working components.
- Black-box testing is where the attacker has no knowledge of the inner workings of the software under test.
- Software defects can be classified as bugs, flaws, anomalies, errors, and vulnerabilities.
- Bug bar is a term for classifying the severity of bugs and having rules as to what levels must be remediated before release.
- During testing, it is important to revisit and confirm the attack surface of the project to note any creep.

16 - Security Testing

Penetration testing is a structured test methodology. The following are the basic steps employed in the process:

1. Reconnaissance (discovery and enumeration)
2. Attack and exploitation
3. Removal of evidence
4. Reporting

Fuzz testing is a staple of SDL-based testing, finding a wide range of errors with a single test method.

Fuzz testing works well in white-, black- or grey-box testing, as it can be independent of the specifics of the application under test.

Simulation testing involves testing the application in an environment that mirrors the associated production environment.

Summary

- Scanning is automated enumeration of specific characteristics of an application or network.
- Penetration testing is an active form of examining the system for weaknesses and vulnerabilities.
- Fuzz testing is a brute-force method of addressing input validation issues and vulnerabilities.
- Simulation testing involves testing the application in an environment that mirrors the associated production environment.
- Although most testing is for failure, it is equally important to test for conditions that result in incorrect values, even if they do not result in failure.
- Only approved cryptographic algorithms should be used; creating your own cryptography is a bad practice.
- Testing various versions of software is referred to as regression testing.
- Bugs are measured in terms of their impact on the system, and this impact can be used to prioritize corrective action efforts.

17 - Secure Software Acceptance

Qualification or acceptance testing is the formal analysis that is done to determine whether a system or software product satisfies its acceptance criteria.

The formal analysis that is performed to determine whether a system or software product satisfies its acceptance criteria is called qualification or acceptance testing.

Qualification Testing Plan - The required features to be tested,

- Requisite load limits
- Number and types of stress tests
- All necessary risk mitigation and security tests
- Requisite performance levels
- Interfaces to be tested
- The test cases to address each of aforementioned questions

Elements of a Qualification Testing Plan

The qualification testing plan should answer the following nine practical questions:

1. Who's responsible for generating the test designs cases and procedures?
2. Who's responsible for executing the tests?
3. Who's responsible for building/maintaining the test bed?
4. Who's responsible for configuration management?
5. What are the criteria for stopping the test effort?
6. What are the criteria for restarting testing?
7. When will source code be placed under change control?
8. Which test designs/cases will be placed under configuration management?
9. What level will anomaly reports be written for?

The Qualification Testing Hierarchy

- Software design traceability analysis (e.g., trace for correctness)
- Software design evaluation
- Software design interface analysis
- Test plan generation (by each level)
- Test design generation (by each level)

Types of testing,

- Black-box Testing
- White-box Testing

- Load Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Alpha Testing
- Beta Testing

ISO 9126 Criteria

The six generic criteria for judging the suitability of a product are

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

Functionality can be decomposed into measures of: 1) suitability to requirements, 2) accuracy of output, 3) interoperability, 4) functional compliance, and 5) security.

Reliability can be decomposed into measures of: 1) product maturity, 2) fault tolerance, 3) frequency of failure, and 4) recoverability.

Usability, or ease of use, can be decomposed into measures of understandability. Usability can be evaluated by measuring the length of time that it takes to learn the product, as well as how long it takes for the user to operate it at an acceptable level of proficiency.

Efficiency can be measured in two terms: time behavior and resource behavior. Time behavior is characterized by factors like response and processing times and throughput rates. Resource behavior can be measured in terms of the amount of resources used and the duration of such use in performing a given function.

Maintainability can be measured in terms of the time it takes to analyze and address a problem report or change request and the ease of the change if a decision is made to alter it.

Portability can be judged in terms of the ability of the product to be adapted to a given situation, its ease of installation, and its conformance with the organization's general requirements and any applicable regulations. Moreover, the ability to replace the product if an improved version is available also should be considered (e.g., the product's extensibility).

Risk Acceptance properties include,

- Implicit and explicit safety requirements
- Implicit and explicit security requirements
- Degree of software complexity
- Performance factors
- Reliability factors

Risk assessment requires detailed knowledge of the risks and consequences associated with the software under consideration. This information is contained in a properly executed threat model, which is created as part of the development process.

The assessment should maintain continuous knowledge of three critical factors:

1. The interrelationships among all of the system assets
2. The specific threats to each asset
3. The precise business and technological risks associated with each vulnerability

Post-Release Audit of the installed configuration

The post-release plan should describe the procedures required to administer the post-release process.

Validation means that the software meets the specified user requirements. Verification describes proper software construction. Barry Boehm clarifies the difference in this simple fashion: Validation: Are we building the right product? Verification: Are we building the product right?

18 - Secure Software Installation and Deployment

The term bootstrapping is also known as booting. For a PC, typical boot processes are the power on self-test (POST) followed by the initial program load (IPL). Integrity of these processes must be mandatory, or all subsequent processes on the machine cannot be trusted.

A common way of keeping track of changes is through a configuration management database (CMDB). A CMDB contains details of the configuration and all subsequent changes in an organization. Another, more generic term is the configuration management system (CMS). Entities seeking ISO/IEC 15408 (Common Criteria) accreditation are expected to maintain a comprehensive CMS.

The term version control is commonly used in the security industry to refer to the process of labeling different releases of software such that the end user has the ability to determine which specific release they are using. Version control is the means by which operations can manage their software deployments to specific configurations.

Summary

- Keep in mind that installation is not necessarily a part of the product development lifecycle, so plans for installation and deployment should be developed to ensure a smooth transition from development to use.
- Customer support requests need to be coordinated by a single entity. This entity is normally called a configuration manager.
- Installation and deployment activities that involve the supplier should be planned as early as possible in order to be written into the contract. This is the only way they can be enforced.
- The interface between the user community and the configuration management process is a critical point of failure, so the establishment of a configuration manager is a critical role.
- Audits are essential to the configuration management process because they confirm that the process continues to be effective.
- Baselines and baseline managers are also critical elements of the configuration management process. Without a good baseline, it is impossible to know the current status.
- All changes have to be verified for correctness prior to reintegrating the changed code into the current controlled baseline.
- The archive of former controlled baselines provides a wealth of quality and security assurance information. Therefore, it has to be kept safe from tampering.
- Configuration management decisions can be strategic. Therefore, upper-level management has to be involved in the decision making process where a major change is concerned.
- Besides being planned, configuration management should also be managed in a disciplined fashion. Simply planning the process does not guarantee its proper execution.
- Configuration management decisions need to be driven by in-depth analysis.
- Therefore, the analyst role is not trivial. Analysts must be able to communicate with top-level decision makers.
- Installation and deployment are significant stages in the lifecycle, even if they are relatively short periods. Therefore, the installation and deployment process should be planned.

19 - Secure Software Operations and Maintenance

A CSSLP should be familiar with the security aspects of the following ongoing activities associated with operations/maintenance of software: monitoring, incident management, problem management, and change

management (patching/updating).

Connecting the operational dots between what needs to be monitored to ensure proper and secure operation relies upon the necessary information being exposed by the software in some form, typically either alerts or logs.

When developing software, it is important to take a holistic view toward security. There are many opportunities to learn crucial pieces of information that can be used at other points of the development process. Properly communicating them across team and development boundaries can result in stronger system performance and security. Many items that are useful in the secure operation of software as part of an enterprise need to be developed in earlier portions of the SDLC process. Beginning with the end in mind can assist in the creation of a securable system.

Metrics can provide management information as to the effectiveness of and trends associated with security processes. Measuring items such as the times between the following items can provide insight into the effectiveness of the response function: Incident occurrence and detection; detection and response; response and containment; and containment and resumption.

The use of management tools and techniques such as root cause analysis can assist in secure operations. Bug and vulnerability tracking and end-user support are inexplicably linked, and proper data logging can improve the ability to use this information in correcting issues in software across releases.

Patch management is a crucial element of a secure production environment. Integrating the patching process in a structured way within the change management process is important to ensure stability and completeness.

Summary

- The operations process is not, strictly speaking, part of the development lifecycle. Instead, operation starts when the product is released from development.
- Customer support requests need to be coordinated by a single entity. This entity is normally part of the operations process.
- Operations activities are often written into the contract to provide post-release sustainment services.
- The interface between the user community and the system operation is a critical point of failure, so the establishment of a problem reporting process is a critical role.
- Operations' day-to-day function is to perform routine tests and reviews to ensure that the system continues to function as built and meets stakeholder requirements.
- Maintenance and operations work together to ensure that any problem or request for modification is responded to appropriately by the organization's management.
- All changes have to be verified prior to releasing the changed software for operational use.
- Incident management is a critical function that has to be planned. Planning creates responses to incidents that can be foreseen, as well as procedures for incidents that are not foreseen.
- Because incidents can impact the entire organization, incident response planning is strategic. As a result, upper-level management has to be involved in the development of the incident response manual.
- Besides being planned, incident response should also be executed as a specialized function of the organization, with particular staffing and equipment requirements.
- Secure disposal is an often-overlooked process because it involves a post-lifecycle period. However, because of magnetic remanence, it is essential to ensure that all retired products have been disposed of properly; otherwise, their information can be stolen.
- Secure disposal ensures an effective retirement of the product. All stakeholders have to be kept in the loop with the change, and the continuation of all user functions must be assured.

20 - Supply Chain and Software Acquisition

A supplier risk assessment is used to identify specific threats to the organization's supply chain, evaluate how likely those threats are to occur, and determine the potential consequences of each threat should it

materialize.

Software can contain intellectual property in many forms: processes, algorithms or coding; thus, the intellectual property can represent business value and hence requires appropriate protections.

The fact that software is procured via a supply chain does not eliminate the need to understand and verify the necessary testing regimens employed as part of the development process. These aspects are best handled via process validation vice product testing.

Summary

- Perform a supplier risk assessment for all potential suppliers prior to engaging in any contractual relationship for a product.
- Ensure that common coding errors are mitigated, but also check for functions that you don't expect to be there, such as malicious code, when performing a risk management exercise.
- Ensure desired practices by locking them into a contract. Every desired consideration and response has to be stipulated in the contract in order to ensure their performance.
- Don't forget post-delivery assurance, particularly patch management. The process of retrospectively assuring product integrity through patches and fixes is a critical part of good supply chain risk management practice.
- Configuration management is perhaps the most important conventional process in the assurance of continuing supply chain integrity. That is because configuration management establishes product baselines and ensures rational change as the product evolves over time.