CYBR515: Software Security
Additional topics for Module 2

Objectives

- Learn basic terminology associated with software requirements
- Examine functional requirements used to implement security in systems
- Understand the role of non-functional requirements such as operational and deployment requirements
- Learn about regulations and compliance requirements
- Examine data classification requirements
- Explore privacy requirements
- Learn how to develop misuse and abuse cases
- Understand the security requirements traceability matrix (RTM)
- See how to ensure that security requirements flow down to suppliers/providers

Introduction: Define Software Security Requirements

- Requirements are the blueprint by which software is designed, built, and tested.
- If they are not managed properly, adding new requirements later in the process can cause significant issues.

Functional Requirements

- Functional requirements describe how the software is expected to function.
- They are translated from business requirements, IT operations requirements, coding requirements, and security requirements into functional requirements that can be followed by designers, coders, and testers.

Business Environment -> Functional Environment (Example #1)

- Business Requirement: "The system should provide a secure and user-friendly registration process for new customers.“

- Functional Requirement: "The system shall include a registration module that ensures secure and user-friendly onboarding of new customers.

Registration System Features

From the previous slide, the system should have the following features:

1. User Registration: The system should allow new customers to create an account by providing necessary information such as name, email address, and password.
2. Validation: The system should validate the entered information to ensure its accuracy and completeness. It should also perform checks to prevent duplicate registrations.
3. Password Security: The system should enforce password requirements, such as a minimum length, combination of alphanumeric characters, and possibly additional security measures like two-factor authentication.
4. User-Friendly Interface: The registration process should be intuitive and easy to follow. Clear instructions and helpful error messages should be provided to guide users through each step.
5. Privacy and Data Protection: The system should handle customer data securely, following industry standards and legal regulations. It should incorporate measures such as encryption and secure data transmission.
6. Error Handling: The system should anticipate and handle any errors or exceptions that may occur during the registration process, providing informative error messages and a way to resolve issues.
7. Confirmation and Feedback: The system should provide a confirmation message or email to the user upon successful registration. It should also give feedback if any issues arise during the process, informing users of the necessary steps to rectify the problem.

Example #2

- Business Requirement: "The university needs a student information system that allows efficient course enrollment and registration."
- Functional Requirement: "The student information system shall provide a comprehensive course enrollment and registration module with the following functionalities:
- (see next slide)

Students Registration System Features

1. Course Search: The system should allow students to search for available courses based on criteria such as course code, title, department, instructor, and schedule.
2. Course Details: The system should display detailed information about each course, including prerequisites, course description, credit hours, and availability.
3. Enrollment Eligibility: The system should determine if a student meets the prerequisites and any enrollment restrictions for a specific course.
4. Course Selection: The system should enable students to select courses for enrollment, including the ability to add or remove courses from their schedule.
5. Seat Availability: The system should provide real-time updates on the number of available seats for each course and prevent enrollment when a course reaches its maximum capacity.
6. Waitlisting: If a course is full, the system should allow students to join a waitlist and prioritize their enrollment if seats become available.
7. Schedule Conflict Detection: The system should check for any scheduling conflicts when students attempt to enroll in courses, notifying them of any conflicts and offering alternative options.
8. Registration Confirmation: Upon successful enrollment, the system should provide a confirmation message or email to students, confirming their course registration.
9. Drop/Add Period: The system should support a designated drop/add period during which students can modify their course selections without any penalties.
10. Enrollment History: The system should maintain a record of students' enrollment history, including past and current courses, grades, and academic progress.

Role and User Definitions

- Role and user definitions are the statements of who will be using what functionality of the software.
- They are presented in generic form, and refinements detail specifics, such as which users are allowed which functionality as part of their job.
- The detailed listing of what users are involved in a system form part of the use-case definition.
- In computer science terms, users are referred to as subjects.
  - This term is important to understand the subject-object-activity matrix presented later in this chapter.

Objects

- Objects are items that users (subjects) interact with in the operation of a system.
  - File
  - Database records
  - Sub-system
  - Program element
- Anything that can be accessed is an object

Objects

- In order to control access, we use access control lists.
- Access control lists are assigned to objects, and defining the objects and their function in a system is an important part of the SDLC.

Activities/Actions

- Activities are permitted events that a subject can perform on an associated object.
- All possible activities should be defined and documented for each object in the system.
- Undocumented functionality has been the downfall of many a system.

Subject-Object-Activity Matrix

- *Subjects represent the who*, *objects represent the what*, and *activities or actions represent the how* of the subject-object-activity relationship.
- To assist designers and developers in correctly defining the subject-object-activity relationship, a matrix referred to as the subject-object-activity matrix is employed.
  - This matrix allows concise communication about allowed system interactions.

Subject-Object-Activity Matrix (Examples)

- Subject-Object-Activity Matrix for the Business Requirement of a Secure and User-Friendly Registration Process:

Subject	Object	Activity
Customer/User	Registration Module	Register on the system
System Administrator/IT Staff	Registration Module	Configure and maintain the registration module
Customer/User	Registration Module	Provide necessary information
System Administrator/IT Staff	Registration Module	Validate and store registration information
System Administrator/IT Staff	Registration Module	Implement password security measures
System Administrator/IT Staff	Registration Module	Ensure privacy and data protection
System Administrator/IT Staff	Registration Module	Handle error handling and exceptions
System Administrator/IT Staff	Registration Module	Confirm successful registration

- Subject-Object-Activity Matrix for the Business Requirement of an Efficient Course Enrollment and Registration System in a University Environment:

Subject	Object	Activity
Student	Student Information System	Search for available courses
Student	Student Information System	View course details
Student	Student Information System	Select courses for enrollment
Student	Student Information System	Check enrollment eligibility
Student	Student Information System	Join a course waitlist (if applicable)
Student	Student Information System	View class schedule
Faculty/Instructor	Student Information System	Create and manage course offerings
Faculty/Instructor	Student Information System	Set prerequisites for courses
Registrar/Academic Services Staff	Student Information System	Monitor course capacities and availability
Registrar/Academic Services Staff	Student Information System	Resolve schedule conflicts
System Administrator/IT Staff	Student Information System	Manage and maintain the system

Use Cases

- Use cases are a powerful technique for determining functional requirements in developer-friendly terms.
  - They are not intended for all subject-object interactions, but rather for complex or confusing or ambiguous situations associated with user interactions with the system.
- Use cases are constructed of actors representing users and intended system behaviors, with the relationships between them depicted graphically. They are typically presented in a graphical format and enable the construction of complex business processes in a simple-to-understand form.

Use Cases / Example

- When sequences of actions are important, another diagram can be added to explain this.
- Figure 3-1 illustrates a use-case model for a portion of an online account system.

- Users are depicted as stick figures, and the intended system functions as ellipses. Use-case modeling requires the identification of the appropriate actors, whether person, role, or process (nonhuman system), as well as the desired system functions.

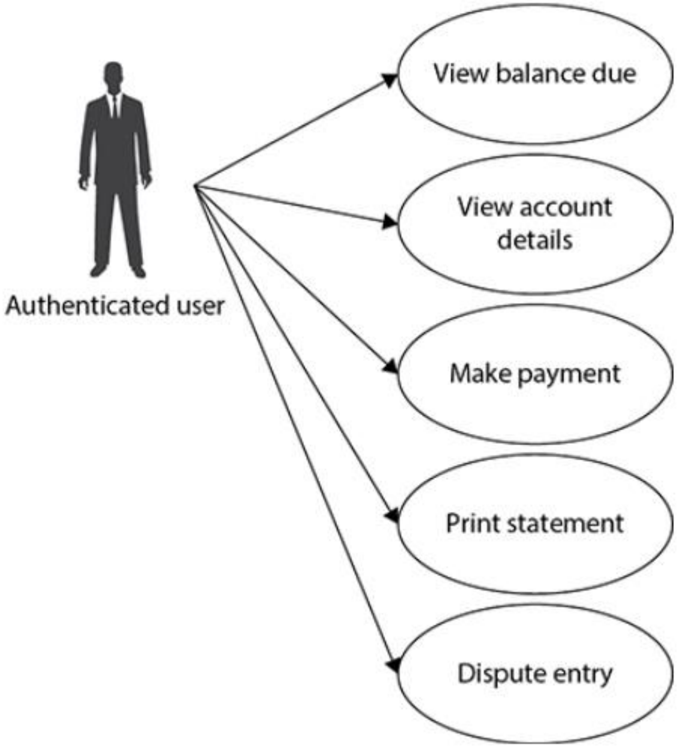


Figure 3-1 Use-case diagram

Sequencing and Timing

- In today's multithreaded, concurrent operating model, it is possible for different systems to attempt to interact with the same object at the same time.
- This is known as a race condition, or a TOC/TOU attack.
- Race conditions arise when one thread depends on a value from another function that is actively being changed by a separate process.
  - These conditions can be difficult to predict and find.

Sequencing and Timing

- Race conditions are defined by race windows, which can be avoided by designing the system so that they are not called concurrently.
- When program logic becomes complex, such as date processing for leap years, care should be taken to ensure that error and other loop-breaking mechanisms do not allow the program to enter a state where the loop controls will fail.
  - a process known as mutual exclusion

Secure Coding Standards

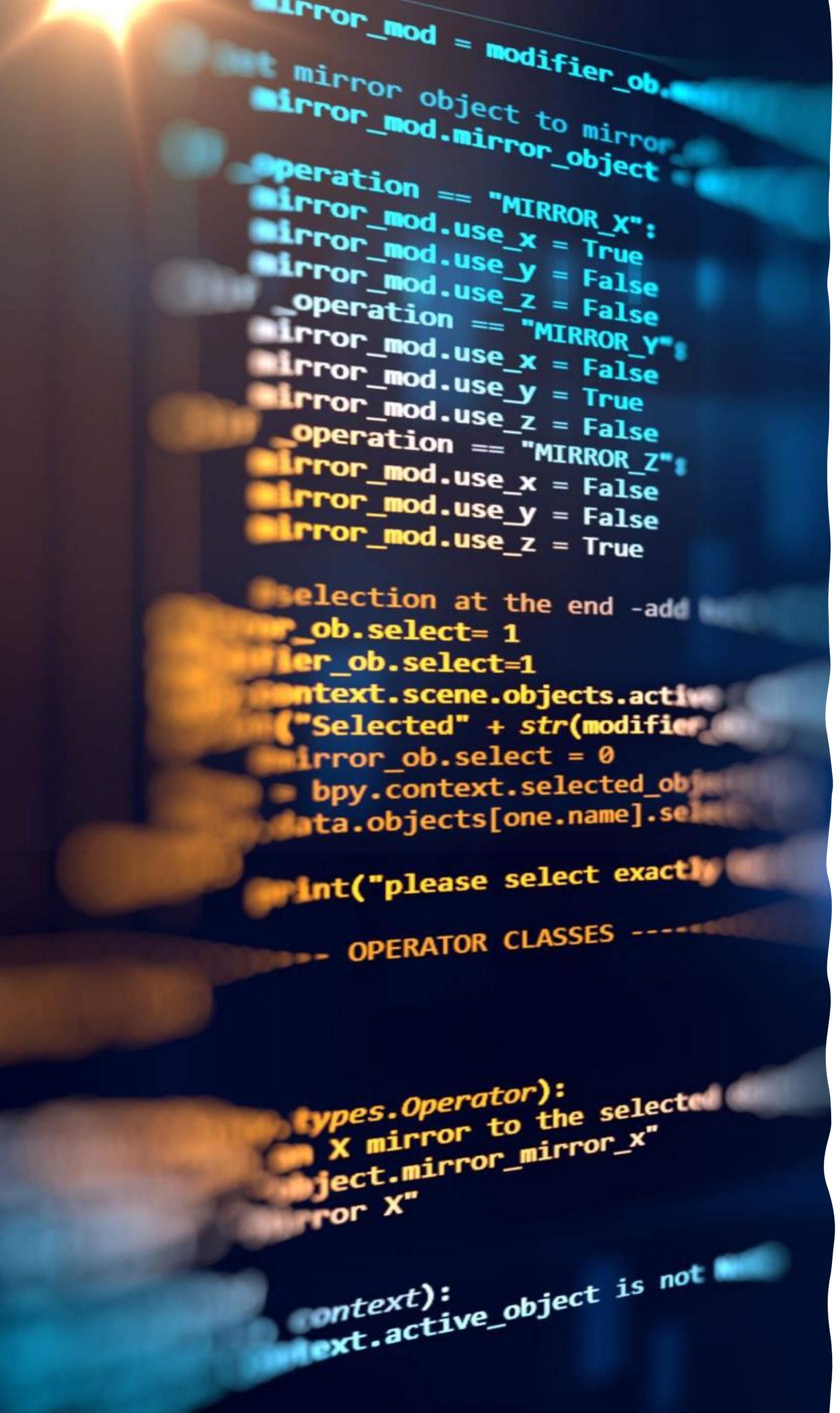
- Complex conditional logic with unhandled states can result in infinite loops.
- Secure coding standards prescribe forms that will preclude specific sets of vulnerabilities and exploitable conditions found in typical code.

Secure Coding Standards

- Application programming can be considered a form of manufacturing.
- Adopting secure coding standards, using a secure application development framework, and adapting and adopting industry best practices are important elements in the secure development lifecycle.
- Poor error trapping and handling is a common problem in many programs.
- An enterprise rule can solve this problem.

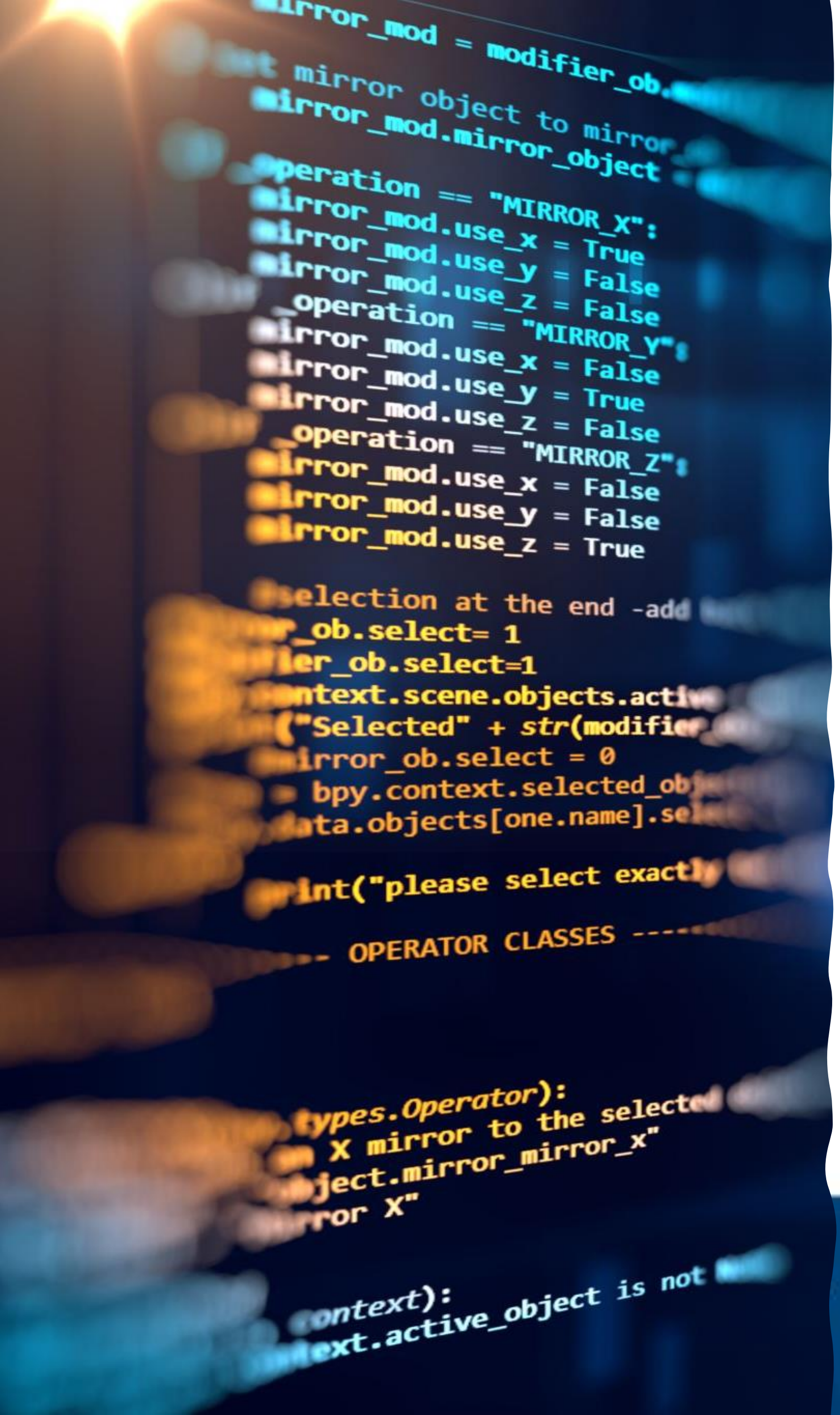
Secure Coding Standards

- Logging is another area that can benefit from secure coding standards.
- Standards can be deployed specifying what, where, and when issues should be logged.
- This serves two primary functions:
  - it ensures appropriate levels of logging,
  - and it simplifies the management of the logging infrastructure



NOTE

Secure coding standards have been published by the Software Engineering Institute/CERT at Carnegie Mellon University for C, C++, and Java. Each of these standards includes rules and recommended practices for secure programming in the specific language



NOTE

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

Operational and Deployment Requirements

- Software is deployed in an enterprise environment where it is rarely completely on its own.
- Enterprises have standards as to technology deployment, specifying platforms, operating systems, specific types and versions of database servers, web servers, and other infrastructure components.

- Enterprises have standards as to technology deployment, specifying platforms, operating systems, specific types and versions of database servers, web servers, and other infrastructure components.

Operational and Deployment Requirements

- Secure software development ensures systems are secure by design, secure by default, and secure in deployment.
  - This includes elements such as secure by default and secure when deployed.
- Software will be deployed in the environment that best suits its maintainability, data access, and access to needed services.
- Corporate standards dictate the selection of database servers and web servers.

Connecting the Dots

- Requirements are the foundational elements used in the development of any project.
- They come from many sources, and the challenge is to enumerate and document all of the related security, functional, and operational requirements that are not stated because they are "implied".

Connecting the Dots

- Requirements are the foundational elements used in the development of any project. They come from many sources, and the challenge is to enumerate and document all of the related security, functional, and operational requirements that are not stated because they are "implied".

Chapter 3: Review

- Chapter 3 of Part 2 examined the requirements associated with a system, including functional requirements, architecture group requirements, and security group requirements.
- It also covered the development of a subject-object-activity matrix.
- Use cases were presented as a means of communicating business requirements and security requirements across the SDLC.
- Enterprise-wide secure coding standards were discussed as a foundational element in defining an enterprise methodology.

Identify and Analyze Compliance Requirements

Chapter 4

Chapter 4: Introduction

- Software development methodologies have been in existence for decades, and security can be added to a development lifecycle to reduce defects that cause security bugs.

Regulations and Compliance

- Regulations and compliance drive many activities in an enterprise. CSSLPs need to understand the basis behind various rules and regulations and how they affect the enterprise in the context of their own development efforts.
- Compliance is not the same as security, because one can be compliant and still be insecure.
  - An "all hazards" approach is popular in many industries.

Regulations and Compliance

- Regulations can come from several sources, including industry and trade groups and government agencies, and the penalties for noncompliance can vary based on the severity of the violation and other factors.
- Many development efforts may have multiple regulatory impacts, and it is important to map the different requirements to the individual data flows that they each affect.

Regulations and Compliance

- When executing software development, it is important to understand the various sources of security requirements, and not to mistake security functionality for the objective of secure software development.

Note

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.

Security Standards

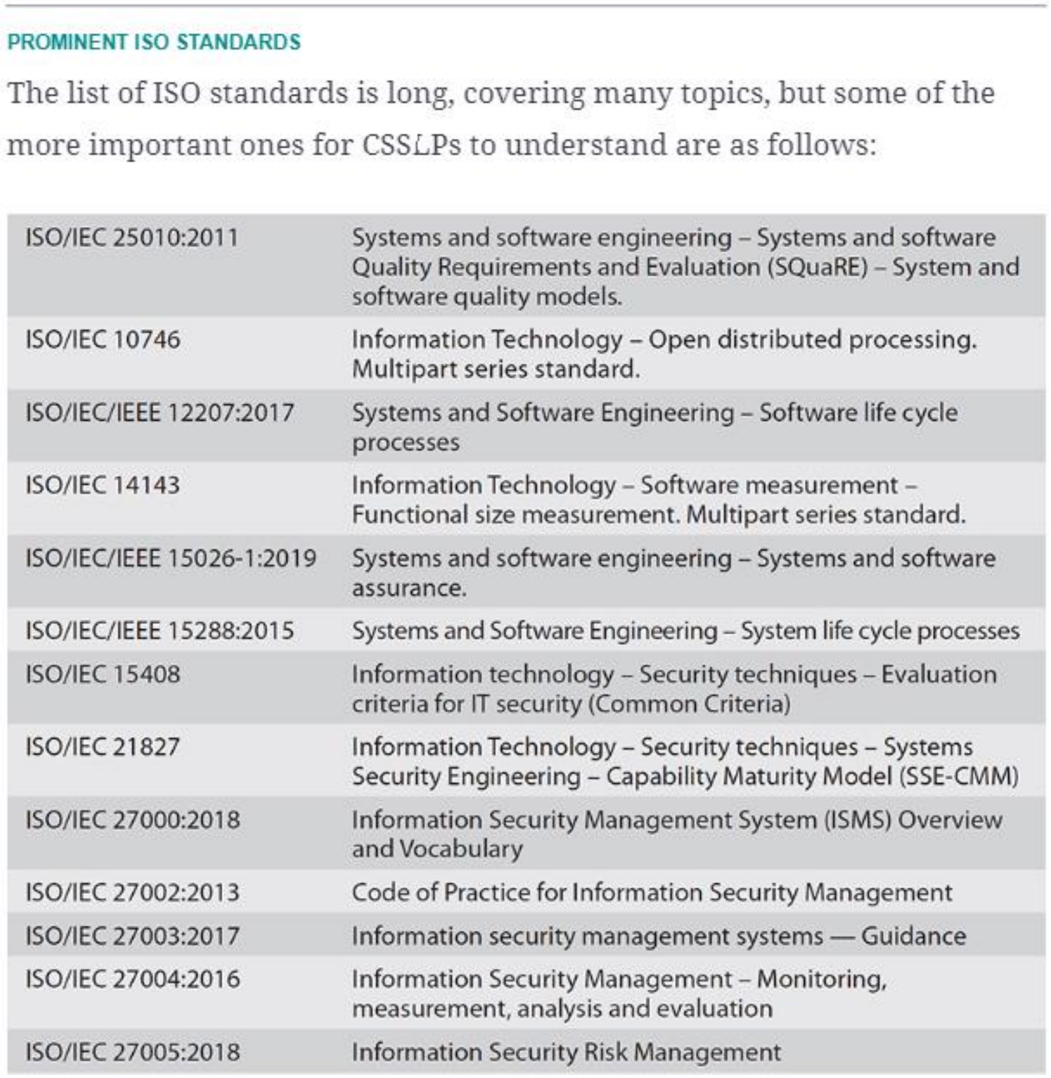
- Standards are a defined level of activity that can be measured and monitored for compliance by a third party.
  - They facilitate comparisons between organizations and enhance security in an enterprise.
  - Security standards serve a role in promoting interoperability. In software design and development, there will be many cases where modules from different sources will be interconnected. In the case of webservices, the WS-security standard provides a means of secure communication between web services.

International ISO

- ISO is the International Organization for Standardization, a group that develops and publishes international standards.
- The United States has an active relationship to ISO through the activities of the U.S. National Committee, the International Electrotechnical Commission (IEC), and the American National Standards Institute (ANSI).

ISO

- The American National Standards Institute (ANSI) has published a variety of standards covering the information security arena. These standards are on a five-year review cycle.



PROMINENT ISO STANDARDS

The list of ISO standards is long, covering many topics, but some of the more important ones for CSSLPs to understand are as follows:

ISO/IEC 25010:2011	Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
ISO/IEC 10746	Information Technology – Open distributed processing. Multipart series standard.
ISO/IEC/IEEE 12207:2017	Systems and Software Engineering – Software life cycle processes
ISO/IEC 14143	Information Technology – Software measurement – Functional size measurement. Multipart series standard.
ISO/IEC/IEEE 15026-1:2019	Systems and software engineering – Systems and software assurance.
ISO/IEC/IEEE 15288:2015	Systems and Software Engineering – System life cycle processes
ISO/IEC 15408	Information technology – Security techniques – Evaluation criteria for IT security (Common Criteria)
ISO/IEC 21827	Information Technology – Security techniques – Systems Security Engineering – Capability Maturity Model (SSE-CMM)
ISO/IEC 27000:2018	Information Security Management System (ISMS) Overview and Vocabulary
ISO/IEC 27002:2013	Code of Practice for Information Security Management
ISO/IEC 27003:2017	Information security management systems — Guidance
ISO/IEC 27004:2016	Information Security Management – Monitoring, measurement, analysis and evaluation
ISO/IEC 27005:2018	Information Security Risk Management

ISO 2700X Series

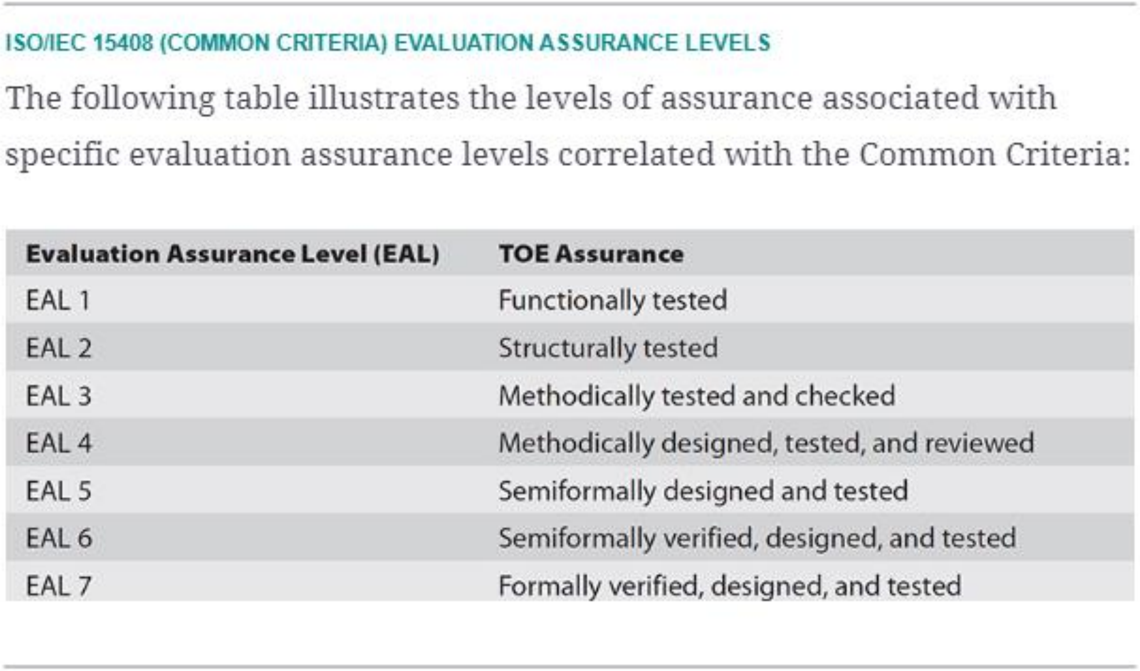
- The ISO 2700X series of standards does for information security what the ISO 900X series does for quality management.
  - This series defines the relevant vocabulary, a code of practice, management system implementation guidance, metrics, and risk management principles.
- The ISO 2700X series of standards covers information security management in all shapes and sizes of organizations, and is a growing family with more than 20 standards currently in place.

ISO/IEC 15408 Common Criteria

- The Common Criteria is a framework for defining security functional and assurance requirements, and testing laboratories can evaluate products to determine if they meet the requirements.
- The Common Criteria use specific terminology to describe activity associated with the framework. The Target of Evaluation, Security Target, and Protection Profile are used to compare products within product classes.

ISO/IEC 15408 Common Criteria

- The Common Criteria process produces an evaluation assurance level, from 1 to 7. Higher EAL does not indicate greater security.



ISO/IEC 15408 (COMMON CRITERIA) EVALUATION ASSURANCE LEVELS

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

Evaluation Assurance Level (EAL)	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

ISO/IEC 9126: Software Engineering – Product Quality

- Product quality is an international standard for the evaluation of software quality. It includes metrics that measure the quality of the software during operation.
- This four-part standard addresses some of the critical issues that adversely affect the outcome of a software development project. The standard provides a framework that defines a quality model for the software product

ISO/IEC 9126: Software Engineering – Product Quality

- ISO 9126 defines six quality characteristics that can be used to measure the quality of software:
  1. Functionality
  2. Reliability
  3. Usability
  4. Efficiency
  5. Maintainability
  6. Portability

ISO/IEC/IEEE 12207: Systems and Software Engineering – Software Life Cycle Processes

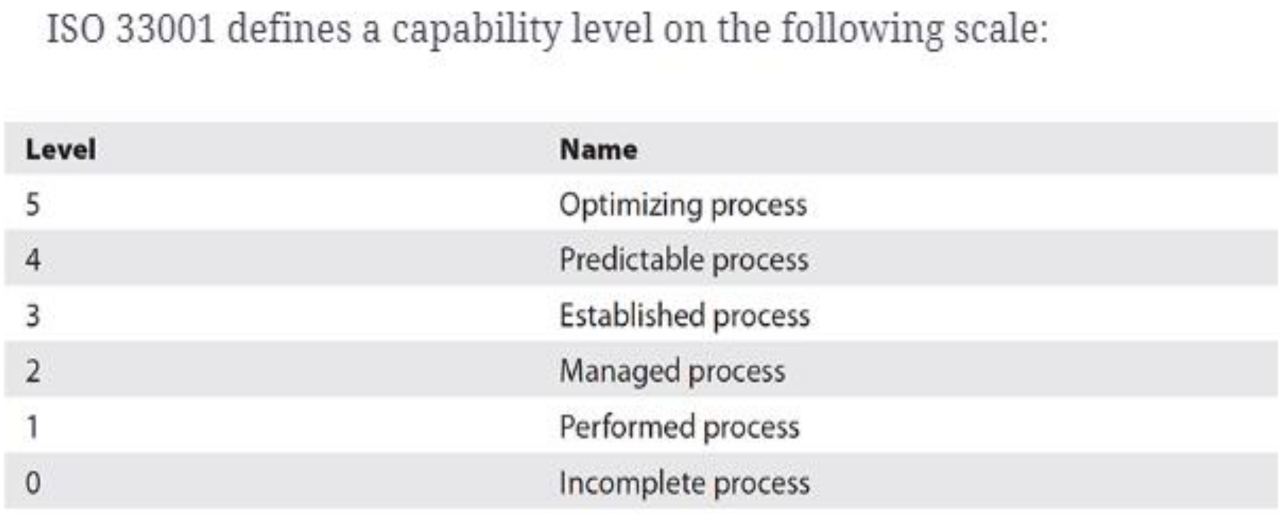
- This international standard establishes a set of processes covering the life cycle of the software. Each process has a defined set of activities, tasks, and outcomes associated with it.
- The standard acts to provide a common structure so all parties associated with the software development effort tcan communicate through a common vocabulary.

ISO/IEC 33001:2015 Information Technology – Process Assessment

- Process assessment is also known as SPICE. SPICE originally stood for Software Process Improvement and Capability Evaluation, but international concerns over the term evaluation has resulted in the substitution of the term determination (SPICD).
- ISO 33001 is a set of technical standards documents for the computer software development process.

ISO/IEC 33001:2015

- The standard was derived from ISO/IEC 12207 and ISO 15504, the process lifecycle standard, and from maturity models like the CMM. ISO33001 is used for process capability determination and process improvement efforts related to software development



ISO 33001 defines a capability level on the following scale:

Level	Name
5	Optimizing process
4	Predictable process
3	Established process
2	Managed process
1	Performed process
0	Incomplete process

The National Institute of Standards and Technology (NIST)

- The National Institute of Standards and Technology (NIST) develops and publishes several relevant document types associated with information security, including Federal Information Processing Standards (FIPS) and Special Publication (SP) 800 series from the NIST Information Technology Laboratory (ITL).
- NIST also publishes Interagency or Internal Reports (NISTIRs).

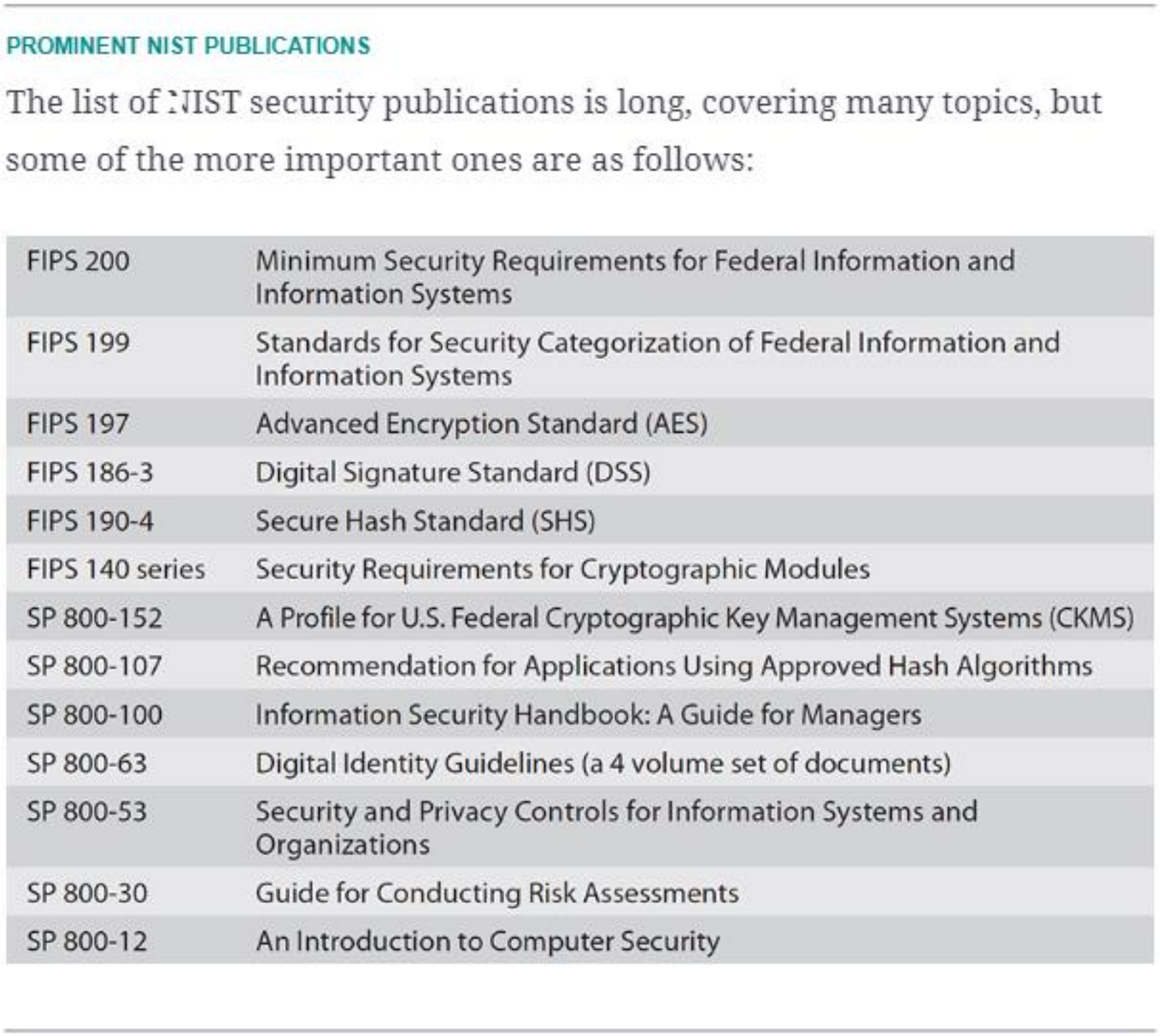
Federal Information Processing Standards (FIPS)

- FIPS are mandatory sets of requirements on federal agencies and specific contractors.
- Although limited in number, they are wide sweeping in authority and scope.
- Older FIPS had sections describing a waiver process, but since the passage of FISMA, all aspects of FIPS are now mandatory, and the waiver process is no longer applicable

NIST SP 800 Series

- The more common set of NIST publications utilized by industry is the 800series of Special Publications. These documents are designed to communicate the results of relevant research and guidelines associated with securing information systems. The 800 series has documents ranging from describing cryptographic protocols, to security requirements associated with a wide range of system elements, to risk management framework elements associated with information security governance

NIST Security Publications



PROMINENT NIST PUBLICATIONS

The list of NIST security publications is long, covering many topics, but some of the more important ones are as follows:

FIPS 200	Minimum Security Requirements for Federal Information and Information Systems
FIPS 199	Standards for Security Categorization of Federal Information and Information Systems
FIPS 197	Advanced Encryption Standard (AES)
FIPS 186-3	Digital Signature Standard (DSS)
FIPS 190-4	Secure Hash Standard (SHS)
FIPS 140 series	Security Requirements for Cryptographic Modules
SP 800-152	A Profile for U.S. Federal Cryptographic Key Management Systems (CKMS)
SP 800-107	Recommendation for Applications Using Approved Hash Algorithms
SP 800-100	Information Security Handbook: A Guide for Managers
SP 800-63	Digital Identity Guidelines (a 4 volume set of documents)
SP 800-53	Security and Privacy Controls for Information Systems and Organizations
SP 800-30	Guide for Conducting Risk Assessments
SP 800-12	An Introduction to Computer Security

Industry

- SAFECode is an industry-backed organization that is committed to increasing communication between firms on the topic of software assurance.
- This group was formed by members who voluntarily share their practices, which together form a best practice solution. SAFECode   to communicating best practices that have been used successfully by member firms.

Industry

- A sampling of SAFECode’s publications include:
  - Software Assurance: An Overview of Current Industry Best Practices
  - Fundamental Practices for Secure Software Development, 3rd Edition
  - Overview of Software Integrity Controls
  - Security Engineering Training

Note

The users' stories for agile can be a valuable resource for CSSLP agile developers to explore. See "SAFECode Releases Software Security Guidance for Agile Practitioners." This paper provides practical software security guidance to agile practitioners in the form of security-focused stories and security tasks they can easily integrate into their agile-based development environments.

You can find it at https://safecode.org/resource-secure-development-practices/guidance-for-agile-practitioners/

(Note that the link to this paper in the book is outdated)

OWASP

- The Open Web Application Security Project (OWASP) is a worldwide free and open community focused on improving the security of application software.
- OWASP publishes a series of Top Ten vulnerability lists highlighting the current state-of-the-art threats facing web application developers.
- OWASP maintains a website (www.owasp.org)with significant resources to help firms build better software and eliminate these common and pervasive problems

The Federal Information Security Management Act of 2002 (FISMA)

- The Federal Information Security Management Act of 2002 (FISMA) requires each federal agency to implement an agencywide information security program.
- The NIST was designated to develop implementation guidelines and published a risk management framework for compliance.

FISMA

- FISMA is a federal information security program that is mandated for federal agencies and, by extension, contractors that implement and operate federal information systems.
- Its effectiveness is directly related to the level of seriousness placed on it by senior management.

FISMA

- NIST has responded with a series of publications detailing a security lifecycle built around a risk management framework. This framework is a process-based methodology for achieving desired security levels in an enterprise.

Sarbanes-Oxley

- The Sarbanes-Oxley Act of 2002, which was a reaction to several major accounting and corporate scandals, mandates a specific level of internal control measures for financial reporting systems.

Gramm-Leach-Bliley

- The Financial Modernization Act of 1999, also known as the Gramm-Leach-Bliley Act (GLBA), contains elements designed to protect consumers' personal financial information (PFI). From a software perspective, it is important to understand that the act specifies rules as to the collection, processing, storage, and disposal of PFI. The three primary rules worth noting are:
  1. The Financial Privacy Rule, which governs the collection and disclosure of PFI, including companies that are nonfinancial in nature
  2. The Safeguards Rule, which applies to financial institutions and covers the design, implementation, and maintenance of safeguards deployed to protect PFI
  3. The Pretexting Protections, which addresses the use of pretexting (falsely pretending) to obtain PFI

HIPAA and HITECH

- The Healthcare Insurance Portability and Accountability Act (HIPAA) deals with personal health information (PHI), and the Health Information Technology for Economic and Clinical Health Act (HITECH Act) deals with electronic personal health information records.

Payment Card Industry Data Security Standard

- PCI stands for Payment Card Industry, and there are three main standards for securing cardholder data.
- PCI standards are contractual requirements for firms that accept payment cards, store payment card data, or make products associated with payment cards.
- There are three main standards:
  1. the Data Security Standard (PCI DSS),
  2. the Payment Application Data Security Standard (PADSS),
  3. and the PIN Transaction Security (PTS).
- Each of these is designed to provide a basic level of protection for cardholder data

Other Regulations

- There are many lesser known regulations that govern banking over the Internet.
- These regulations state that authentication must be multifactor in nature.

Legal Issues

- Legal issues frame a wide range of behaviors and work environments.
- This comes from the concept that when disputes between parties arise, the legal system is a method of resolving these disputes.
- Over time, a body of laws and regulations has been created to govern activities, providing a roadmap for behavior between parties

Intellectual Property

- Intellectual property is a legal term that recognizes that creations from the mind can be and are property to which exclusive control can be granted to the creator.
- A variety of different legal mechanisms can be used to protect the exclusive control rights.
- The association of legal mechanism to the property is typically determined by the type of property. The common forms of legal protection are patents, copy rights, trademarks, trade secrets, and warranties

Patents

- Patents are exclusive rights granted by a government to the inventor for a specified period of time.
- Patents allow the owner to prevent others from using a claimed invention, even if the other party claims they independently developed a similar item.

SOFTWARE PATENTS

- There is intense debate over whether software patents should be granted, if at all.
- There is some overlapping protection for software in the form of copyrights, which are covered in the next section.
- In the United States, patent law excludes issuing patents to abstract ideas, and this has been used to deny some patents involving software.
- In Europe, computer programs as such are typically excluded from patentability.

Copyrights

- Copyrights are a form of intellectual property protection applied to any expressible form of an idea or information that is substantive and discrete.
- They are governed internationally through the Berne Convention.

SOFTWARE COPYRIGHTS

- Patent protection and copyright protection cover the same subject matter, but each serves a different purpose.
- Copyright protection prevents another entity from copying a program, but does not prevent other developers from independently writing their own versions.

Trademarks

- Trademarks are recognizable qualities associated with a product or firm, and can be registered with the government for more legal protection.
- Amazon.com is a trademarked name because it projects the image of the firm. Common terms are not eligible for trademark protection.

Trade Secrets

- Trade secrets offer the ultimate in time-based protection for intellectual property, and are protected by a variety of laws. The most famous trade secrets revolve around food and taste, such as Coca-Cola's recipe or Kentucky Fried Chicken's recipe.
- Cryptographic algorithms are difficult to use in software, as the end user has access to much information. However, there is limited protection under the Digital Millennium Copyright Act.

Warranties

- Warranties represent an implied or contractually specified promise that a product will perform as expected, but do not guarantee that it will perform the tasks the user bought it for.
- The technical specification of software is typically considered by the end user to be fitness for use on the end user's problem. However, most software licenses specifically dismiss this measure.

Data Classification

- Data can be classified as to its state, its use, or its importance from a security perspective.
  - This is done as part of the development process to reduce the costs associated with protecting data.
- Data classification may be simple or fairly complex, depending on the size and scope of the enterprise, and may include specific compliance requirements.

Data States

- Data can be considered a static item that exists in a particular state at any given time.
- When considering data states, it is easy to expand this idea to the information lifecycle model (ILM), which includes generation, retention, and disposal

Data States

- For the purposes of development and security, these states are:
  - At rest, or being stored
  - Being created
  - Being transmitted from one location to another
  - Being changed or deleted
    - In addition, one should consider where the data is currently residing:
  - On permanent media (hard drive, CD/DVD/optical disc)
  - On remote media (USB, cloud/hosted storage)
  - In RAM on a machine

Data Usage

- Data can also be classified as to how it is going to be used in a system.
- This is meant to align the data with how it is used in the business and provide clues as to how it should be shared, if appropriate.

Data Usage Classification

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

Data Usage

- In addition, data can be considered security sensitive, marked as containing personally identifiable information (PII) or to be hidden.
- These categories include
  - Security-sensitive data A subset of data of high value to an attacker
  - PII data that contains PII elements
  - Hidden data that should be concealed to protect it from unauthorized disclosure using obfuscation techniques

Data Risk Impact

- Data that is labeled medium risk has serious consequences if disclosed, while low-risk data has limited consequences if lost or disclosed.
  - Each firm needs to determine the appropriate definitions of severe, serious, and limited.
- Data that is labeled medium risk has serious consequences if disclosed, while low-risk data has limited consequences if lost or disclosed. Each firm needs to determine the appropriate definitions of severe, serious, and limited.

Data Lifecycle

- Data in the enterprise has a lifecycle. It can be created, used, stored, and destroyed, and must be managed from a backup and business continuity/disaster recovery perspective.

Generation

- Data can be generated in the enterprise in many ways. If the data is going to be persistent, it needs to be classified and have the appropriate protection and destruction policies assigned.

Data Ownership

- Data is the property of the enterprise, but it is assigned to people in a form of an ownership or stewardship role.

Data Owner

- The data owner is the party that determines who has specific levels of access associated with specific data elements, such as who can read, who can write, who can change, delete, and so on.
- Data owners are responsible for defining data classification, defining authorized users and access criteria, and implementing appropriate security controls. Data custodians implement the desired actions.

Data Custodian

- Data custodians support the business use of data in the enterprise and ensure that processes safely transport, manipulate, and store the data.
- Data custodians may not require access to read the data elements, but they do need appropriate access to apply policies to the data elements.

Labeling

- Because data can exist in the enterprise for an extended period of time, itis important to label the data in a manner that can ensure it is properly handled.
- Data in the enterprise can exist for an extended period of time, and metadata fields can be used to support data labeling.

Sensitivity

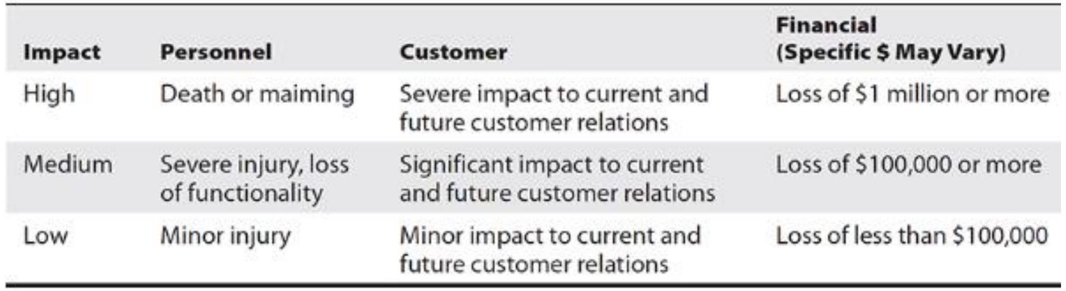
- Data can have different levels of sensitivity within an organization.
  Payroll data can be sensitive, with employees having restricted access.
- Some employees need specific access to this type of data, based on position, and HR personnel have business reasons to access this data, although the access may not be just by job title or position.

Impact

- Data can be classified by impact, which is a wider concern than sensitivity. NIST FIPS 199 and SP 800-18 provide a framework for classifying data.
- Data can be classified by the impact the organization would suffer in the event of data loss, disclosure, or alteration.
- The first step in impact analysis is to define the levels of high, medium, and low. These levels can be based on impact to people, impact on customers, and financial impact.

Impact

- The differentiators for the separation of high, medium, and low can be based on impact to people, impact on customers, and financial impact.
- Table 4-1 shows a typical breakdown of activity



Impact	Personnel	Customer	Financial (Specific $ May Vary)
High	Death or maiming	Severe impact to current and future customer relations	Loss of $1 million or more
Medium	Severe injury, loss of functionality	Significant impact to current and future customer relations	Loss of $100,000 or more
Low	Minor injury	Minor impact to current and future customer relations	Loss of less than $100,000

Table 4-1 Summary of Impact Level Definitions

Privacy

- Privacy is the principle of controlling information about one's self.
  - To buy something over the Internet, you need to enter a credit card or other payment method into a website, and to have the item delivered to your house, you need to provide an address..
  - If credit card information is stored for future orders, safeguards are needed to protect the data. Data can also be used to test systems, but anonymization can work.

PRIVACY AND SOFTWARE DEVELOPMENT

- Privacy may seem like an abstract issue for CSSLP, but there are many rules and regulations associated with software development and privacy.
- Development teams need to be aware of these rules and regulations.

Privacy Policy

- A privacy policy is a high-level document that describes the principles associated with the collection, storage, use, and transfer of personal information within the scope of business.
- A customer-facing privacy policy is provided to customers to inform them of how their data is protected, used, and disposed of.

Personally Identifiable Information

- Information that can be used to identify an individual is referred to as personally identifiable information (PII).
- The U.S. government defines PII using the following from an Office of Management and Budget Memorandum.

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

Personal Health Information

- Personal health information is a set of data elements associated with an individual's healthcare that can be used to identify a specific individual.
  - This information is protected by HIPAA and the HITECH Act.
- PHI and associated medical data are sought after by cybercriminals because they contain insurance information and financial responsibility information, including credit cards.

Breach Notifications

- When security fails to secure information and information is lost to parties outside of authorized users, a breach is said to have occurred.
  - The state of California was the first to address this issue with SB 1386, a data disclosure law that requires disclosure of unencrypted personal information.

General Data Protection Regulation

- Two factors led to a complete rewrite of EU data protection regulations: the Snowden revelations and the European Court of Justice invalidating the Safe Harbor provisions.
- The General Data Protection Regulation (GDPR)ushers in a brand-new world with respect to data protection and privacy. It makes the Safe Harbor provisions obsolete and requires specific programs to address the requirements.

General Data Protection Regulation

- The GDPR brings many changes, including the appointment of a data protection officer. The DPO must operate with significant independence.
- The GDPR specifies requirements regarding consent, which are significantly more robust than previous regulations. Consent must be specific to each data-processing operation, and the data subject can withdraw consent at any time.

General Data Protection Regulation

- The GDPR provides protections for new individual rights, and firms may be forced to adopt new policies to address these requirements. The GDPR also requires that data protection issues be addressed by means of appropriate safeguards.
- The EU expressed concern about the adequacy of data protection in the United States, forcing U.S. and other international companies to adapt their privacy protections.

GDPR PERSONAL DATA ELEMENTS

- Under GDPR, personal data is defined as any information relating to an identified or identifiable natural person.
- This includes the following if they are capable of being linked back to the data subject:
  - Online identifiers
  - IP addresses
  - Cookie

GDPR PERSONAL DATA ELEMENTS

- GDPR demands that individuals have full access to information on how their data is processed.
- The GDPR mandates that individuals must be able to withdraw consent at any time and have a right to be forgotten. If data is no longer required for the reasons for which it was collected, it must be erased.

GDPR PERSONAL DATA ELEMENTS

- Under GDPR, personal data is defined as any information relating to an identified or identifiable natural person.
- This includes the following if they are capable of being linked back to the data subject:
  - Online identifiers
  - IP addresses
  - Cookie

California Consumer Privacy Act 2018 (AB 375)

- California passed a sweeping privacy bill in June 2018 that mandates several key elements. Individuals have a right to know how their data is being used, to object to data being sold, and to access their data.

Privacy-Enhancing Technologies

- Information security is the ability to control information about oneself. Encryption is at the top of the list of privacy-enhancing technologies (PETs) for protecting privacy and anonymity, and is a prime choice for protecting information at any stage in its lifecycle.
- PETs can be used to prevent the transfer of cookies between browsers and web servers, block HTTP headers that reveal personal information, and block banner ads, pop-up windows, animated graphics, or other unwanted web elements.

Data Minimization

- Data minimization is one of the most powerful privacy-enhancing technologies.
- Data minimization involves keeping what you don't need, and in the EU, this means destroying unneeded personal information once the transaction has concluded.
  - This reduces the risk of future breaches and disclosures by not keeping "excess" data.

Data Masking

- Data masking involves changing data by substituting altered values, or by physically redacting elements.
- This makes reverse engineering or detection impossible.

Tokenization

- Tokenization is the use of a random value to take the place of a data element that has traceable meaning.
  - For example, a credit card approval code provides complete traceability to a transaction.
- Tokens are used in data transmission systems involving commerce to protect sensitive information from being reused or shared, yet maintain the desired nonrepudiation characteristics of the event.

Anonymization

- Data anonymization is the process of removing identifiers from data to keep the connection to the source anonymous.
- This is easier said than done, because data exists in many places in many forms.

Pseudo-anonymization

- Pseudo-anonymization replaces private identifiers with fake identifiers or pseudonyms, while maintaining statistical accuracy and data integrity.
- It can be used for training, development, testing, and analytics while protecting data privacy.

Chapter 4: Review

- Chapter 4 of Part 2 examined the regulatory and compliance elements associated with secure development, including security standards, laws and regulations, contractual schemes, intellectual property concerns, and data classification. It then pivoted to data privacy, and discussed technologies used to enhance privacy.

Chapter 5: Misuse and Abuse Cases

- Use cases are used to convey requirements for how software should be used, while misuse cases convey requirements for what should not be allowed.
- Misuse cases are a form of use case illustrating specifically prohibited actions. They serve a valuable role in communicating requirements to developers and testers.

Misuse/Abuse Cases

- Figure 5-1 illustrates a series of misuse cases associated with the online account management system.
- In this diagram, the actor is now labeled as unauthorized user. This is different from the previous authenticated user, as this misuse actor may indeed be authenticated.
- The misuse actor could be another customer or an internal worker with some form of access required to manage thes ystem.

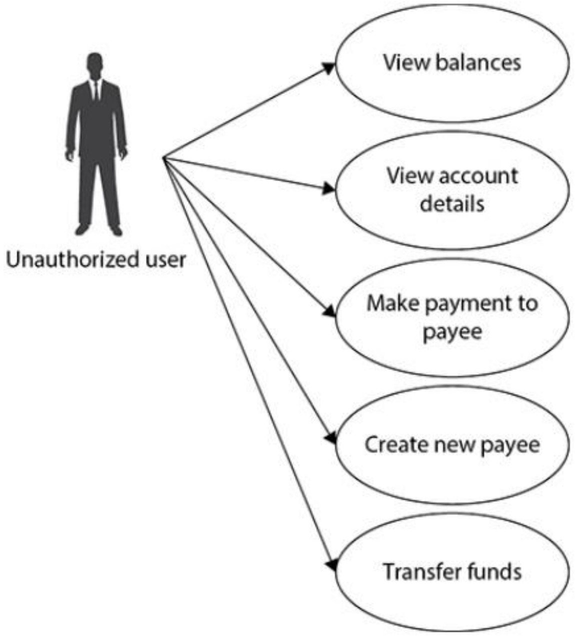


Figure 5-1 Misuse-case diagram

Misuse/Abuse Cases

- Through brainstorming exercises, the development team discovered that a system administrator could create a new payee on an account, which would enable them to put themselves or a proxy into the automatic bill-pay system. To mitigate this activity, a design of an out-of-band mechanism was introduced to make it significantly more difficult.

Misuse/Abuse Cases

- Misuse cases are used early in the development process to decide and document how the software should react to improper use. They are created through a process of informed brainstorming and are designed to facilitate communication among designers, developers, and testers.

Misuse/Abuse Cases

- Misuse cases can examine a system from an attacker’s point of view, whether the attacker is an inside threat or an outside one.
- Properly constructed misuse cases can trigger specific test scenarios to ensure known weaknesses have been recognized and dealt with appropriately before deployment.

Note

- SAFECode has made significant contributions to development and distribution use cases. It has published a useful document describing "Practical Security Stories and Security Tasks for Agile Development Environments," available for use and free download at https://safecode.org/publication/SAFECode_Agile_Dev_Security0712.pdf

Misuse/Abuse Cases

- Misuse cases can be used to help document nonfunctional or quality requirements, such as reliability, resiliency, maintainability, testability, and so on.
- Security failures occur whenever software performs a function that is not authorized, whether by bad input or environmental conditions.
- While security threats many times stem directly from genuinely hostile agents, the reliability requirements may be elicited and analyzed as threats caused by things that are not necessarily intelligent. Such causes include human error, storms, design errors (e.g., software bugs), and interference or noise across communication links.

Misuse/Abuse Cases

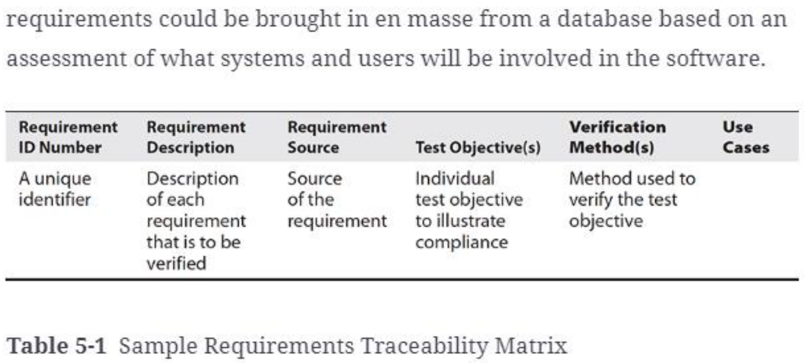
- When examining the threat model, misuse cases provide a means of communicating the implications of threats to function to the entire development team.
- When a misuse case intersects with a use case, this use case is said to mitigate the misuse case.

Requirements Traceability Matrix

- A requirements traceability matrix (RTM) helps the development team track and manage requirements and implementation details, and helps document the relationships between security requirements, controls, and test/verification efforts.

Requirements Traceability Matrix

- Table 5-1 illustrates a sample RTM.
- An RTM allows the automation of many requirements, providing the team to gather sets of requirements from centralized systems.
- The security requirements could be brought in en masse from a database based on an assessment of what systems and users will be involved in the software.



requirements could be brought in en masse from a database based on an assessment of what systems and users will be involved in the software.

Requirement ID Number	Requirement Description	Requirement Source	Test Objective(s)	Verification Method(s)	Use Cases
A unique identifier	Description of each requirement that is to be verified	Source of the requirement	Individual test objective to illustrate compliance	Method used to verify the test objective	

Table 5-1 Sample Requirements Traceability Matrix

Requirements Traceability Matrix

- Software with only internal users will have different requirements from that of a customer interface across the Web.
- An RTM acts as a management tool and documentation system, and helps ensure that all requirements are appropriately managed.

Software Acquisition

- Software is not always created by combining existing elements, but rather, by connecting separate modules.
  - Not all software elements will be created by the development team.
- Acquisition of software components has security implications that are covered in detail in Chapter 20.

Software Acquisition

- Acquisition has its own set of terms used throughout this technical/legal discipline, but a couple of them stand out in the secure software environment.
- Commercial off-the-shelf (COTS) software is readily available for purchase and integration into a system, whereas government off-the-shelf (GOTS) software is specifically developed for government use.

Build vs. Buy Decision

- Software acquisition can be accomplished in two manners, either by building it or buying it.
- In today's modular world of software, the line between build and buy is blurred, as some elements may be built and some purchased.
- Successful integration involves ensuring requirements include both the business process perspective and generic features and functions.

Outsourcing

- Software development is an expensive undertaking. Outsourcing of development to other countries can lower the cost of development, but not all costs can be lowered by shipping development to a cheaper group of coders based on geography.
- The geographic separation of the development team from the business team adds complexity, learning curves, and cost. Outsourcing can make sense, but you must understand the details.

Contractual Terms and Service Level Agreements

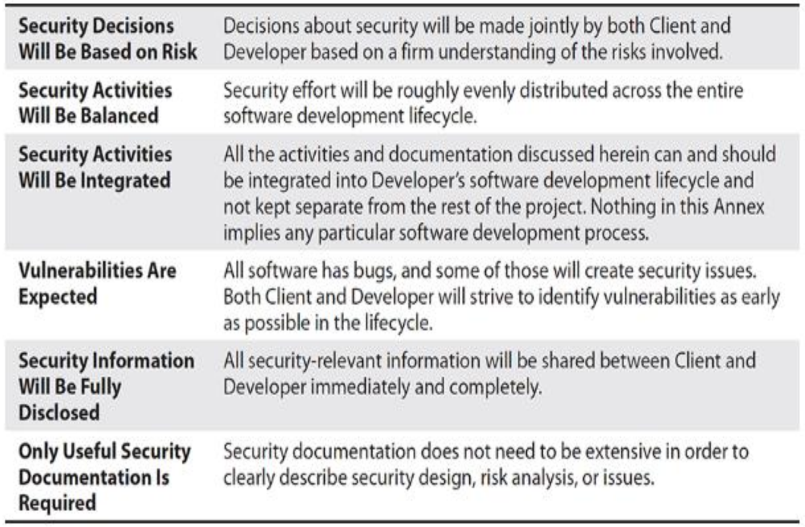
- Contractual terms and service level agreements establish expectations with respect to future performance, and can include security controls or standards that are expected to be implemented.
- Specific ISO standards or NIST standards that are desired by a supplier should be included in these mechanisms to ensure clear communication of expectations. Service level agreements can include acceptance criteria that software is expected to pass prior to integration.

Requirements Flow Down to Suppliers/Providers

- Software is seldom created without connections to suppliers or providers.
- Companies are incorporating language into their purchasing contracts that bring requirements with respect to many aspects of software development, including the release of a software bill of materials.

Requirements Flow Down to Suppliers/Providers

- Operationalizing the elements in Table 5-2 requires agreement on many subjects, including specific security requirements, source code control, including revision control and code escrow, and general security issues such as connection to external systems, input validation and encoding, authentication and session management, access control, logging, error handling, secure configuration, encryption, availability, libraries, testing procedures, and bug remediation processes.



Security Decisions Will Be Based on Risk	Decisions about security will be made jointly by both Client and Developer based on a firm understanding of the risks involved.
Security Activities Will Be Balanced	Security effort will be roughly evenly distributed across the entire software development lifecycle.
Security Activities Will Be Integrated	All the activities and documentation discussed herein can and should be integrated into Developer's software development lifecycle and not kept separate from the rest of the project. Nothing in this Annex implies any particular software development process.
Vulnerabilities Are Expected	All software has bugs, and some of those will create security issues. Both Client and Developer will strive to identify vulnerabilities as early as possible in the lifecycle.
Security Information Will Be Fully Disclosed	All security-relevant information will be shared between Client and Developer immediately and completely.
Only Useful Security Documentation Is Required	Security documentation does not need to be extensive in order to clearly describe security design, risk analysis, or issues.

Table 5-2 OWASP Secure Software Contract Annex

Chapter 5 Review

- In this chapter, we examined misuse/abuse cases, introduced a requirements traceability matrix, and discussed how security requirements will flow through a supply chain.