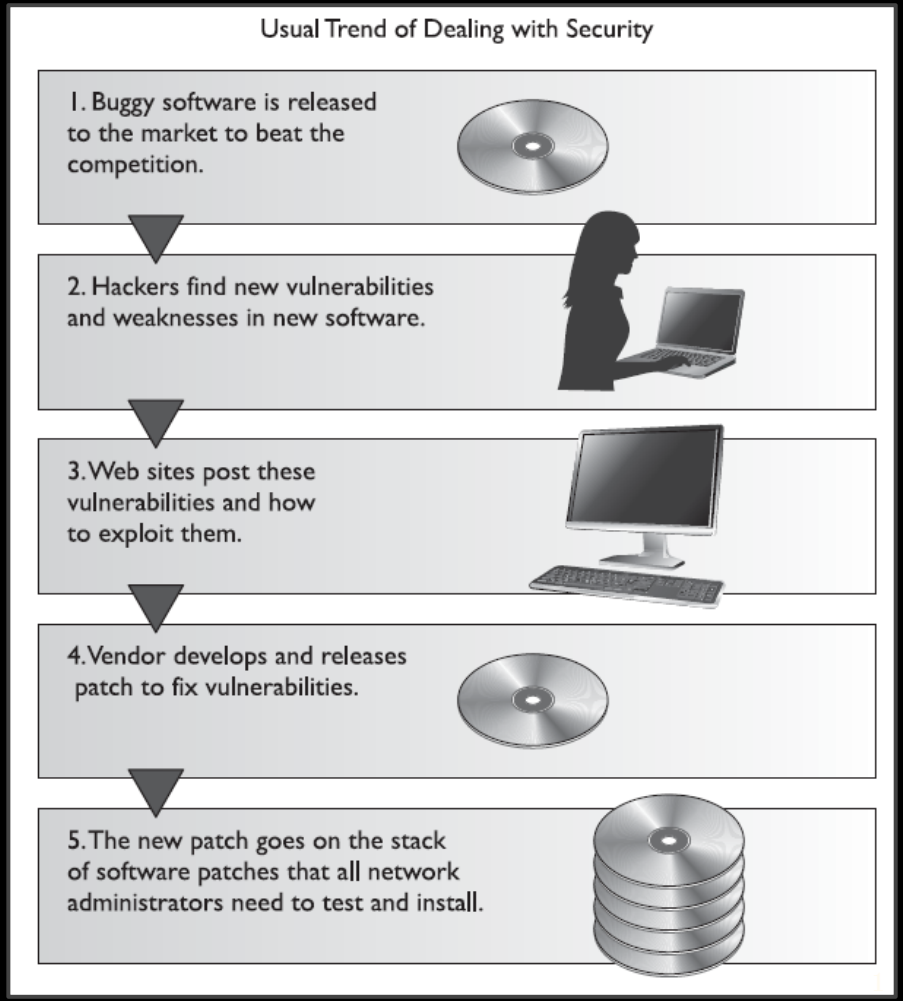
# Software Development Security

Kennedy Sanchez (MGP, PS. Auditor, Comptia Security+)



# This chapter present the following

- Common software development issues.
- Software development life cycles.
- Secure software development approaches.
- Change control and configuration management.
- Database concepts and security issues.
- Web development.
- Web server security.

Building secure software is the responsibility of all the stakeholders involved with the system/software development lifecycle (SDLC).

| |
|---|
| **In the past, it was not crucial to implement security during the software development stages.** |
| **Most security professionals are not software developers**. |
| **Functionality is usually considered more important than security.** |
| Software vendors are trying to **get their products to market in the quickest possible time,** and thus do not take time for proper security architecture, design, and testing steps. |

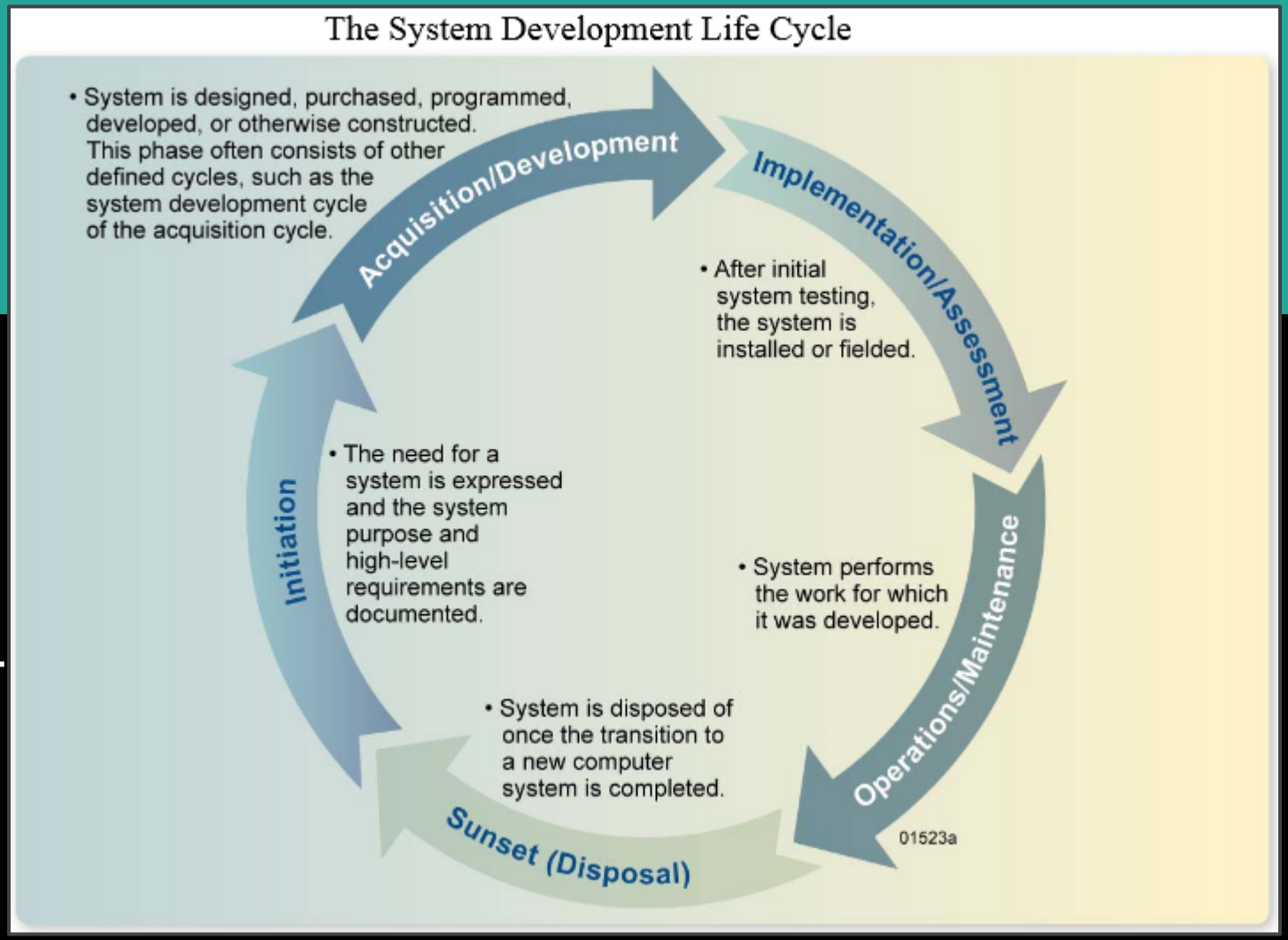
The computing community has gotten used to receiving **software with flaws and then applying patches.** This has become a common and seemingly acceptable practice.

Customers cannot control the flaws in the software they purchase, so they must depend upon perimeter protection.

# System/Software Development Life Cycle - SDLC

The acronym “SDLC” can represent **system** development life cycle or **software** development life cycle

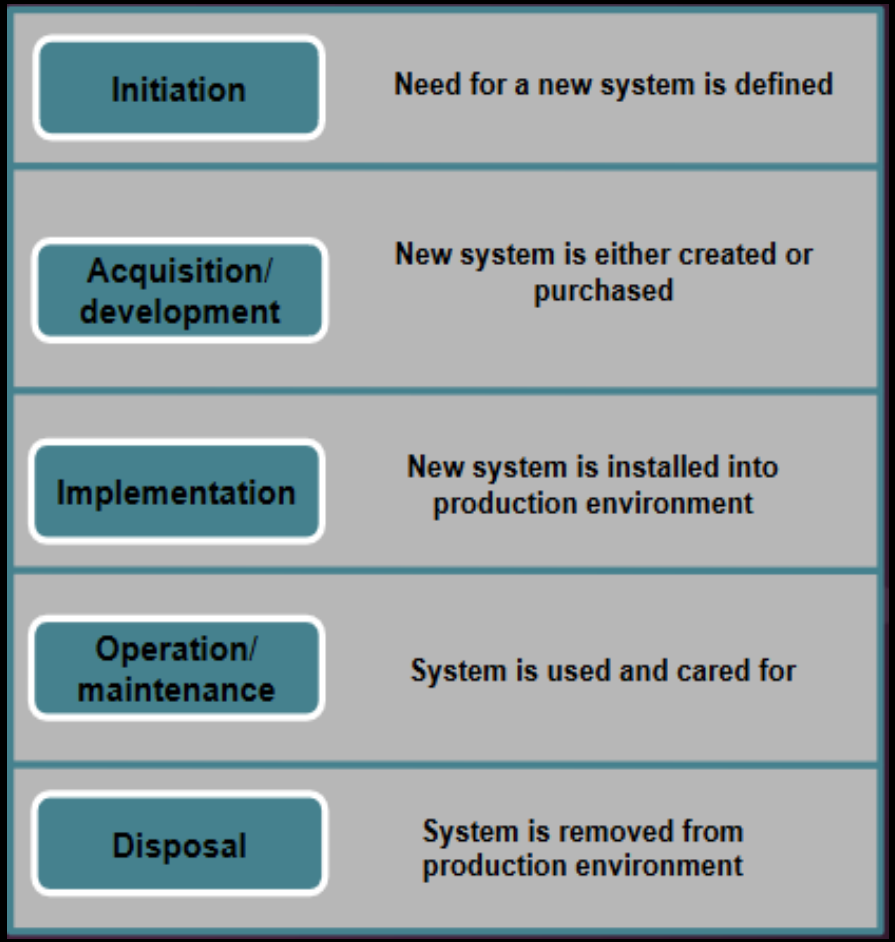
The system development life cycle is the **overall process of developing, implementing, and retiring information systems through a multistep process** from initiation, analysis, design, implementation, and maintenance to disposal.


The System Development Life Cycle
• System is designed, purchased, programmed, developed, or otherwise constructed. This phase often consists of other defined cycles, such as the system development cycle of the acquisition cycle.
Acquisition/Development
Implementation/Assessment
• After initial system testing, the system is installed or fielded.
Operations/Maintenance
• System performs the work for which it was developed.
Sunset (Disposal)
• System is disposed of once the transition to a new computer system is completed.
Initiation
• The need for a system is expressed and the system purpose and high-level requirements are documented.
01523a

Some of the benefits of integrating security into the system development life cycle include.

- Early identification and mitigation of security vulnerabilities and problems with the configuration of systems.
- Facilitation of informed executive decision making through the application of a comprehensive risk management process in a timely manner.
- Documentation of important security decisions made during the development process.
- Improved systems interoperability and integration that would be difficult to achieve if security is considered separately at various system levels.

A system has its own developmental life cycle, which is made up of the following phases



# Initiation

**Establishes the need for a specific system and documents its purpose.**

**Security planning** should begin in the initiation phase with the identification of key security roles to be carried out in the development of the system.

**The information to be processed, transmitted, or stored is evaluated for security requirements**, and all stakeholders should have a common understanding of the security considerations.

# From a security point of view, the type of questions are.

| | | |
|---|---|---|
| What level of protection does this system need to provide?” | Does it need to protect sensitive data at rest and in transit? | Does it need to provide two factor authentication? |

## The acquisition/development phase

**Pertains to the “buy” or “build” decision.**

An organization will need to evaluate the needs of the system, if the system can be developed in-house, or if the system needs to be purchased from a vendor.

**“What do we need and why do we need it?”**

Before the system is actually **developed or purchased.**

several things should take place to ensure the end result meets the company's true needs. Some of the activities are as follows.

- ➔ Requirements analysis.
- ➔ Formal risk assessment.
- ➔ Security functional requirements analysis.
- ➔ Security assurance requirements analysis.
- ➔ Third-party evaluations.
- ➔ Security plan.
- ➔ Security test and evaluation plan

**This activities would help in the results to supplement the baseline security controls.**

# Requirements analysis

Study of what functions the company needs the desired system to carry out.

# Formal risk assessment

Identifies vulnerabilities and threats in the proposed system and the potential risk levels as they pertain to confidentiality, integrity, and availability

**The results of this assessment help the team build the system's security plan.**

# Security functional requirements analysis

Identifies the protection levels that must be provided by the system to meet all regulatory, legal, and policy compliance needs.

# Security assurance requirements analysis

**Identifies the assurance levels the system must provide.**

The activities that need to be carried out to ensure the desired level of confidence in the system are determined.

# Third-party evaluations

Reviewing the level of service and quality a specific vendor will provide if the system is to be purchased.

# Security plan

**Documented security controls the system must contain to ensure compliance with the company's security needs.**

This plan provides a complete description of the system and ties them to key company documents, as in configuration management, test and evaluation plans, system interconnection agreements, security accreditations, etc.

# Security test and evaluation plan

Outlines how security controls should be evaluated before the system is approved and deployed.

Once all of these activities are carried out, the company can develop the necessary system in-house or purchase the system from a third-party vendor.

If a company chooses to develop their needed solution internally.

**Does the staff know how to carry out software architecture, design, development, testing, and deployment securely?**

# Implementation

The organization configures and enables system security features.

- ➔ Tests the functionality of these features.
- ➔ Installs or implements the system.
- ➔ Obtains a formal **authorization** to operate the system.

# Implementation

**It may be necessary to carry out certification and accreditation (C&A) processes.**

➔ **Certification** is the technical testing of a system.

➔ **Accreditation** is the formal authorization given by management to allow a system to operate in a specific environment.

**The accreditation decision is based upon the results of the certification process.**

Once the system is implemented into its environment, it must be used in a secure fashion and maintained securely.

# Operations/Maintenance

- In this phase, systems and products are in **place and operating.**
- Enhancements and/or modifications to the system are **developed and tested.**
- Hardware and software components are **added or replaced.**

The organization should continuously monitor performance of the system to ensure that it is consistent with pre-established user and security requirements.

**Configuration management (CM) and control** activities should be conducted to document any proposed or actual changes in the security plan of the system.

Information systems are in a constant state of evolution with upgrades to hardware, software, firmware, and possible modifications in the surrounding environment.

**Documenting information system** changes and assessing the potential impact of these changes on the security of a system are essential activities to assure continuous monitoring.

# Operations/Maintenance

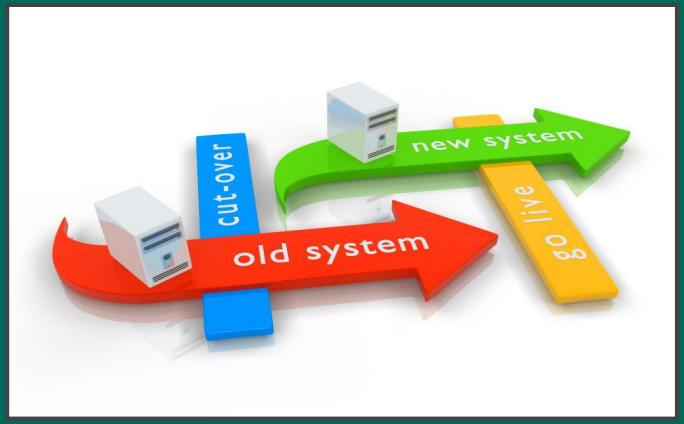
**Microsoft Baseline Configuration Analyzer 2.0**

**A system should have baselines** set pertaining to the system's hardware, software, and firmware configuration during the implementation phase.

In the operation and maintenance phase, **continuous monitoring** needs to take place to ensure that these baselines are always met.

**Vulnerability assessments and penetration testing** should also take place in this phase. These types of periodic testing allow for new vulnerabilities to be identified and remediated.

# Disposal




Disposal activities need to **ensure that an orderly termination of the system takes place** and that all necessary data are preserved.

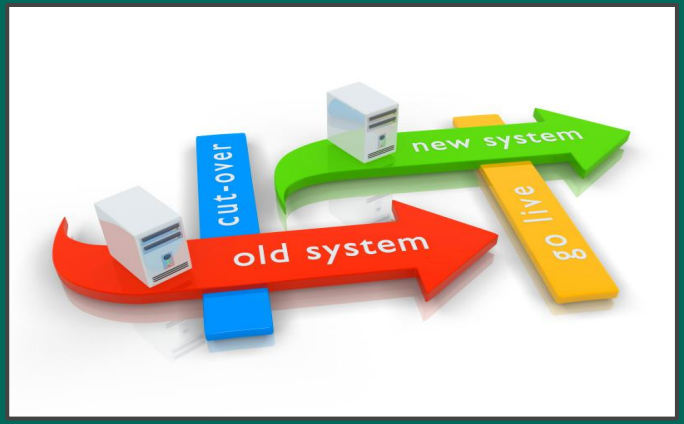
When a system no longer provides a needed function, plans for how the system and its data will make a transition should be developed.

**Data may need to be moved to a different system, archived, discarded, or destroyed.**

**Data Sanitization and Disposal Tools**

https://www.cmu.edu/iso/tools/data-sanitization-tools.html

# Disposal




- In this phase, plans are developed for discarding system information, hardware, and software and making the transition to a new system.
- The information, hardware, and software may be moved to another system, archived, discarded, or destroyed.
- If performed improperly, the disposal phase can result in the unauthorized disclosure of sensitive data.

**Data Sanitization and Disposal Tools**

https://www.cmu.edu/iso/tools/data-sanitization-tools.html

In the disposal phase, does the operation staff even know what zeroization and degaussing is, and if they do, do they know how to do it in a standardized manner?

# Ten Best Practices for Secure Software Development

# $(ISC)_2$

https://www.isc2.org/uploadedfiles/%28isc%292_public_content/certification_programs/csslp/isc2_wpiv.pdf

# Protect the Brand Your Customers Trust

# 1

It's the defenders and their organizations that need to stay a step ahead of cybercriminals or else they will be held responsible for security breaches.

In the event of cybercrimes, victims will look for someone to be held responsible, and it will not be the hackers but the brands that the victims trusted to protect them.

# Know Your Business and Support it with Secure Solutions

# 2

Understanding the business can help in the identification of regulatory and compliance requirements, applicable risk, architectures to be used, technical controls to be incorporated, and the users to be trained or educated.

need to have multi-factor authentication architecture, encryption, authentication, authorization, and auditing controls, as well as the need to educate employees on social engineering and phishing.

# Understand the Technology of the Software

# 3

But one must have a strong background in technology to be effective in building or buying secure software. A lack of understanding of the technology used to build or buy software can lead to insecure implementations of the software.

When it comes to building the software in-house, a thorough understanding of network segregation, hardened hosts, and public key infrastructure, is necessary to ensure that the deployment of the software will, first, be operationally functional and, second, not weaken the security of the existing environment.

## Ensure Compliance to Governance, Regulations, and Privacy

# 4

One must understand the internal and external policies that govern the business, its mapping to necessary security controls, the residual risk from post implementation of security controls in the software, and the compliance to regulations and privacy requirements.

# Know the Basic Tenets of Software Security

# 5

## These basic tenets are.

- Protection from disclosure (confidentiality).
- Protection from alteration (integrity).
- Protection from destruction (availability).
- Who is making the request (authentication).
- What rights and privileges does the requestor have (authorization).
- The ability to build historical evidence (auditing).
- The management of configuration, sessions, and exceptions.

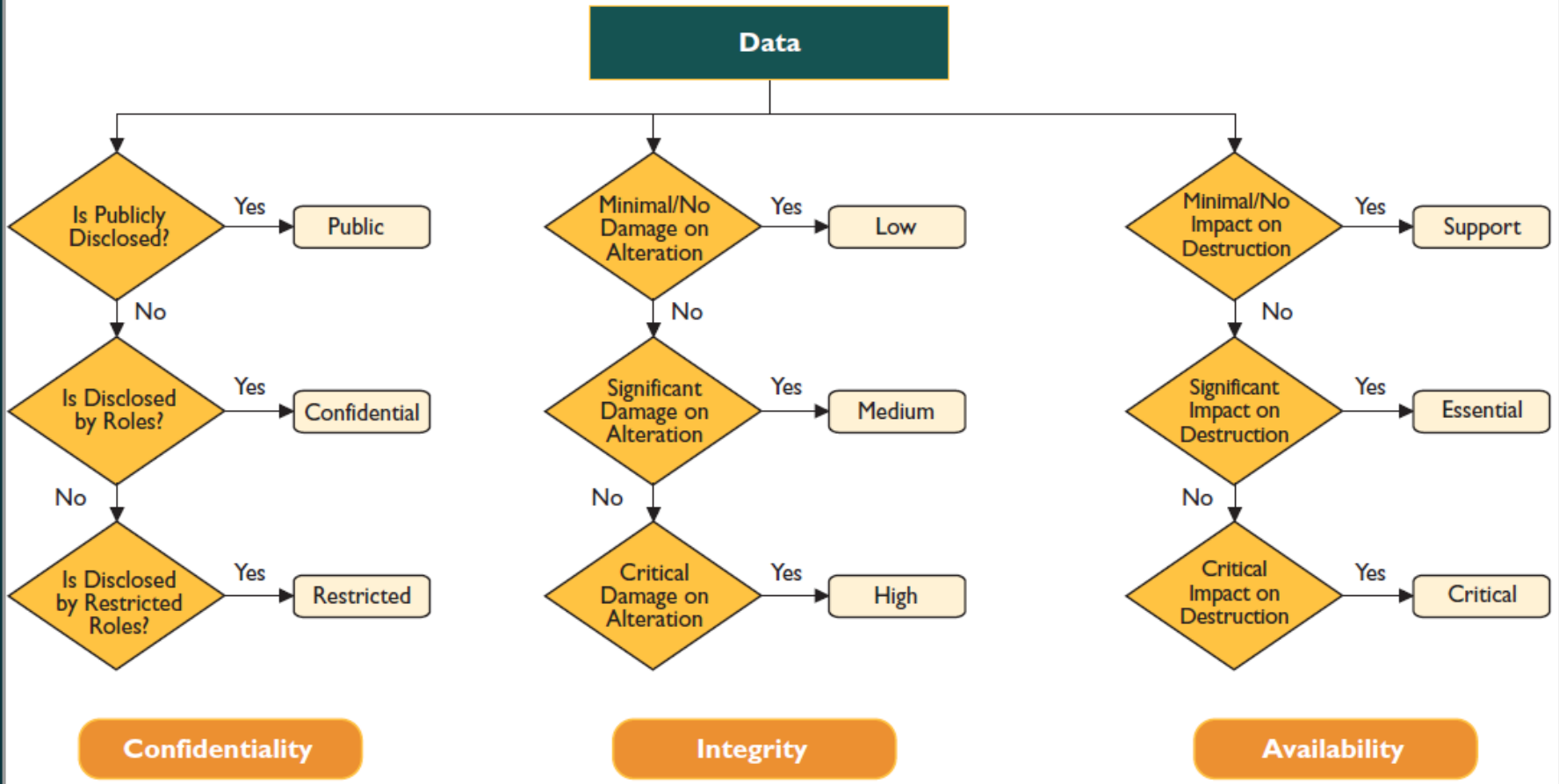
# Ensure the Protection of Sensitive Information

# 6

Data classification is the conscious decision to assign a level of sensitivity to data as it is being created, stored, transmitted, or enhanced.

Sensitive information refers to any information upon which the organization places a measurable value. By implication, this is information that is not in the public domain and would result in loss, damage, or even business collapse should the information be lost, stolen, corrupted, or in any way compromised.

*Figure 2*. Example of a data classification flowchart



# Design Software with Secure Features

# 7

One must be aware of how to implement secure design principles.

| Design Principle | What is it? | Example |
|---|---|---|
| Economy of mechanism | Keeping the design simple and less complex | Modular code, Shared objects, and Centralized services |
| Fail-safe defaults | Access denied by default and granted explicitly | Denied transaction |
| Complete mediation | Checking permission each time subject requests access to objects | Credentials not cached |
| Open design | Design is not a secret, implementation of safeguard is | Cryptographic algorithms |
| Separation of privilege | More than one condition is required to complete a task | Split keys, Compartmentalized functions |
| Least privilege | Rights are minimum and users granted access explicitly | Non-administrative accounts, Need to know |
| Least common mechanisms | Common mechanisms to more than one user/ process/role is not shared | Role based dynamic libraries and functions |
| Psychological acceptability | Security protection mechanism unbeknownst to the end user for ease of use and acceptance | Help dialogs, Visually appealing icons |

Table 1. Adapted from the Saltzer & Schroeder Protection of Information in Computer Systems

# Develop Software with Secure Features

# 8

One should not only ensure that the code written is secure, but also know how to write secure code, conduct, perform, and orchestrate code reviews and security testing.

It is recommended that security code review be performed while the code is reviewed for functionality, and before the software is released for testing. Security code reviews can be manual or automated.

# Deploy Software with Secure Features

# 9

Secure deployment ensures that the software is functionally operational and secure at the same time. It means that software is deployed with defense-in-depth, and attack surface area is not increased by improper release, change, or configuration management.

It also means that assessment from an attacker's point of view is conducted prior to or immediately upon deployment.

## Educate Yourself and Others on How to Build Secure Software

# 10

The National Institute of Standards and Technology (NIST) states that education should cause a change in attitudes, which in turn will change the organizational culture.

NIST SP 800-64
NIST SP 800-88, Guidelines for Media Sanitization
NIST SP 800-95, Guide to Secure Web Services.

People without proper knowledge of software security can circumvent even the most carefully thought-out security implementations.

# ACERCA DE MI

- Ingeniero en Sistemas Informáticos Universidad Central del Este (UCE)
- Profesor Universidad Dominicana O&M
- Maestria en Gerencia y Productividad - Universidad APEC
- Postgrado de Auditoria de Sistemas - Universidad O&M
- Comptia Security+ Certified


@ksanchez_cld


ksanchez_cld