Ch.10 summary By @MHazazi

What Is SDLC

- Systems development lifecycle (SDLC) provides a standard process for any system development
- **There are five phases in the SDLC:**

  1) **Initiation phase**
     Establishes the need for a system and documents its purpose

  2) **Development /acquisition phase**
     The system is designed, purchased, programmed, or developed

  3) **Implementation phase**
     The system is tested and retested, and any modifications are applied until it is accepted

  4) **Operational phase**
     The system in put into production

  5) **Disposal phase**
     Ensure the orderly termination of the system

- SDLC principles apply to commercial off-the-shelf software (COTS) and open source software
  - Development is not done in-house but should be evaluated to ensure it meets or exceeds the organization's security requirement
  - Only stable and tested software should be deployed

- **Software Releases:**
  - Alpha phase
    - Initial release of software for testing
    - Can be unstable
  - Beta phase
    - Software is complete and ready for usability testing
  - Release candidate (RC)
    - Hybrid of beta and final release version
    - Has the potential of being final release unless significant issues are identified
  - General availability or go live
    - Software has been made commercially available

Software Updates

- Updates are different from security patches
- **Security patches** are designed to address a specific vulnerability
- **Updates** include functional enhancements and new features
- Updates should be thoroughly tested
- A documented rollback strategy should exists before applying any updates
- If update required a system reboot, it should be delayed until the reboot has the least impact on business operations

Testing Environment Concerns

- Companies SHOULD have a test environment
- The closer to the live environment the test environment is, the more expensive it is, but the more accurate the testing will be
- The cost of setting up the test environment should be compared to the cost of a loss of data confidentiality, integrity, and/or availability because of a patch-related reason
- Testing environment should be 100% segregated from the live network
- Live data should NEVER be used in a test environment
- The test servers may not be as well secured as the live, production servers
- De-identified or dummy data should be used in place of live data

Secure Code

Two types of code

- Insecure code (referred as “sloppy code”)
- Secure code
    - Deploying secure code is responsibility of the systems’ owner

The Open Web Application Security Project (OWASP)

- Open community dedicated to enabling organizations to develop, purchase, and maintain ap that can be trusted
- Every 3 years releases the top 10 most critical web application security flaws

- **Other issue :**
  - Injection = untrusted data is sent to an interpreter as part of a command or query
  - Input validation = validating all the input to an application before using it
  - Dynamic data verification = data that changes as updates become available
  - Output validation = validating (masking) the output of a process before show it to the user
  - Broken authentication and session management= hijacked or taken over by a malicious intruder

Cryptography

- **Cryptography** = **Encryption** = The process that takes plain text and turns it into cipher text
- **Ciphertext**: Text cannot be read unless apply the correct algorithm and predetermined value
- The predetermined value is also referred to as a **key**
- The key must be securely stored and strong enough to resist brute force cracking attempts.
- **Goals of cryptography** = Confidentiality , Integrity , Authenticity

- **Hashing**
  - The process of creating a numeric value that represents the original text
  - It is a one-way process
  - Provides integrity but not confidentiality and authentication

- **Digital signature:**
  - A hash value that has been encrypted with the sender's private key
  - Insures nonrepudiation and data integrity
  - Does not insure data confidentiality

- **What Is a "Key"?**
  - **Key** is a secret code that is used by a cryptographic algorithm
  - **Keyspace** , is the number of possible keys that can be used with an algorithm
  - **Symmetric key(shared key) algorithm**
    uses a single secret key, which must be shared in advance and kept private by both the sender and the receiver.
  - **Asymmetric key(public key) algorithm**
    uses two different but mathematically related keys known as public and private keys

- **Public Key Infrastructure (PKI)**
  - Framework and services used to create, distribute, manage, and revoke public keys
    - **Components:**
      - **Certification Authority (CA)** = issues and maintains digital certificates.
      - **Registration Authority (RA)** = verifying the identity of users and organizations
      - **Client nodes** = interfaces for users, devices, and applications to access PKI functions,
      - **Digital certificate** = associate a public key with an identity

- **Protecting the encryption keys**
  - Compromised keys mean that the confidential data is not safe anymore

  - Worse if the company does not *know* that the key has been compromised as it will continue to rely on it and use it to send confidential data, thinking that it is secure

  - Someone must be officially responsible for the security of the keys (senior IT employee, in correlation with the information security officer)

- **Digital certificates can be revoked**
  - Usually a bad sign! It means there is a chance that the key has been compromised

  - If there's the slightest chance that a key may have been compromised, the digital certificate MUST be revoked

  - Revocation lists are kept to verify that a given certificate has not been revoked
  - Certificates can be suspended when it is known that it won't be used for a period of time
  - Key destruction must occur before a hard drive is reused