Armaan Shamsaasef

Encrypting Data

Notes: Encrypting Data (SY0-701 - 1.4)

Core Concept: Encryption is the process of converting plaintext (readable data) into ciphertext (unreadable, scrambled data) using a cryptographic algorithm and a key. Its primary purpose is to ensure confidentiality.

Key States of Data & Encryption Methods

Data needs to be protected in different states: while stored and while moving.

1. Encrypting Data at Rest

- **Definition**: Protecting data when it is stored on a physical or digital medium (e.g., a hard drive, SSD, database, or smartphone).
- **Purpose**: To prevent unauthorized access if the physical device is lost, stolen, or improperly decommissioned.
- **Common Technologies**:
  - Full Disk Encryption (FDE): Encrypts the entire storage drive (e.g., Windows BitLocker, macOS FileVault).
  - Database Encryption: Encrypts specific fields (like credit card numbers) or entire tables within a database.
  - File-level Encryption: Encrypts individual files or folders.

2. Encrypting Data in Transit

- **Definition**: Protecting data while it is actively moving across a network from one location to another.
- **Purpose**: To prevent eavesdropping (sniffing) or manipulation of data as it travels.
- **Common Technologies**:
  - TLS (Transport Layer Security): Used to secure web traffic (HTTPS), email, and instant messaging.

    - IPsec (Internet Protocol Security): Used to create secure VPN tunnels between networks or between a client and a network.
    - SSH (Secure Shell): Used to securely access and manage network devices and servers.

3. Encrypting Data in Process

- **Definition**: Data that is currently being used by a computer's CPU and is stored in temporary memory (RAM).
- **Challenge**: This data is typically in an unencrypted state so the processor can work with it. Protection at this stage often relies on overall system security.

Key Management: The Most Critical Part

The security of any encryption system depends entirely on the protection of the keys, not the secrecy of the algorithm.

- **Key Generation**: Creating a strong, cryptographically random key.
- **Key Exchange**: Securely sharing a key with the intended party (e.g., using asymmetric encryption like RSA to share a symmetric key).
- **Key Storage**: Protecting keys from unauthorized access, often using a dedicated Hardware Security Module (HSM).
- **Key Rotation**: The process of retiring an old encryption key and generating a new one at regular intervals to limit the amount of data protected by a single key.
- **Destruction**: Permanently and securely deleting keys when they are no longer needed, ensuring that old data encrypted with them can no longer be decrypted.

Symmetric vs. Asymmetric Encryption

- **Symmetric Encryption**:
    - **Single Key**: Uses the same secret key to both encrypt and decrypt data.
    - **Use Case:** Ideal for bulk encryption of data at rest because it is very fast.
    - Examples: AES (Advanced Encryption Standard), DES, 3DES.
- **Asymmetric Encryption**:
    - **Key Pair**: Uses a mathematically linked public key and private key.

    - **How it works**: Data encrypted with the public key can only be decrypted with the corresponding private key.
    - **Use Case**: Primarily used for key exchange (e.g., in TLS), digital signatures, and encrypting small amounts of data in transit.
    - Examples: RSA, Elliptic Curve Cryptography (ECC).

Key Takeaway

Encryption is a fundamental tool for protecting data confidentiality. To be effective, you must:

1. Choose the right type for the data's state (at rest vs. in transit).
2. Use strong, modern algorithms (like AES-256 and RSA-2048).
3. Implement a robust key management lifecycle, as the security of the entire system hinges on the protection of the keys.