**Domain 4: Application and Data Protection**

**Sub Domain: Administrative Application Security**

▪ **Application Security Administration**

Defines and enforces security policies for application use, access, and lifecycle. Focuses on secure SDLC, role-based access, and compliance with regulations.

▪ **Application Security Administration Practices**

Operational methods to enforce app security:

* Role-based access control (RBAC)
* Secure configurations
* Patch management
* Logging and monitoring
* Regular audits

**▪ Defense in Breadth**

A horizontal security approach that applies layered controls across all systems (e.g., servers, endpoints, apps). Ensures widespread protection across the environment.

▪ **Defense in Breadth vs Defense in Depth**

| **Defense in Depth** | **Defense in Breadth** |
| --- | --- |
| Layers within a single system | Controls across multiple systems/platforms |
| Focuses on internal resilience | Focuses on wide security coverage |
| Example: WAF + TLS + input validation | Example: MFA + antivirus on all endpoints |

**LO#01:** Discuss and implement Application Whitelisting and Blacklisting

▪ **Application Whitelisting**

**Definition:**  
A security approach that **allows only trusted applications** to run. Any application not explicitly allowed is blocked.

**Purpose:**

* Prevent execution of unauthorized or malicious software.
* Strengthen endpoint and system-level control.
* Common in high-security environments (govt., finance, ICS).

**Common Rule Types:**

* **Hash-based** (file signature)
* **Path-based** (file location)
* **Publisher-based** (signed apps)
* **File type** (extension control)

**▪ Application Blacklisting**

**Definition:**  
Blocks specific **known malicious or unwanted applications** while allowing all others.

**Use Case:**

* Corporate environments with moderate security requirements.
* Control usage of games, torrent apps, and unlicensed software.

**Limitation:**  
Cannot block unknown or zero-day threats.

▪ **Using Software Restriction Policies (SRP) for Application Whitelisting**

**Tool:** Built into Windows (Group Policy)

**How it works:**

* Configure through gpedit.msc under **Software Restriction Policies**.
* Rules based on:
  + **Hash**
  + **Path**
  + **Zone**
  + **Certificate**

**Use Case:**  
Suitable for basic application control in small-to-medium Windows environments.

▪ **Using AppLocker for Application Whitelisting**

**Platform:** Windows Enterprise / Education Editions

**Features:**

* More advanced than SRP.
* Supports multiple file types: EXE, MSI, Scripts, DLLs.
* Two modes:
  + **Audit** (logs violations)
  + **Enforce** (blocks unapproved apps)

**Benefits:**  
Better rule management and integration with Active Directory for centralized control.

**▪ Using McAfee Application Control for Application Whitelisting**

**Type:** Commercial enterprise solution

**Capabilities:**

* Dynamic allowlisting using trust models.
* Real-time enforcement without the need for frequent signature updates.
* Includes file integrity monitoring and policy rollback.

**Use Case:**  
High-security and offline environments like industrial control systems (ICS).

**▪ Using ManageEngine Desktop Central for Application Blacklisting**

**Type:** Unified Endpoint Management (UEM)

**Function:**

* Centralizes blacklisting policies.
* Prevents execution of blacklisted software across Windows, macOS, and Linux systems.

**Features:**

* Blacklist by app name, vendor, or file type.
* Push rules remotely.

**▪ Using Windows PUA (Potentially Unwanted Applications) Protection Feature**

**Tool:** Built into Microsoft Defender Antivirus

**What it blocks:**

* Adware
* Toolbars
* Unwanted software bundles

**Enable via PowerShell:**

Set-MpPreference -PUAProtection Enable

**Manage via:**  
Group Policy or Microsoft Endpoint Manager (Intune).

**▪ Using Group policies for Blocking Software Installation from Users**

**Tool:** Group Policy Editor (GPO)

**Location:**

Computer Configuration > Administrative Templates > Windows Components > Windows Installer

**Capabilities:**

* Block or restrict Windows Installer.
* Allow only digitally signed applications.
* Disable installation of new software entirely.

**▪ Using Registry for Blocking Certain Apps**

**Tool:** Windows Registry Editor

**Path:**

HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

**Steps:**

* Create a key named DisallowRun.
* Add string entries for apps to block (e.g., chrome.exe, notepad.exe).

**Caution:**  
Should be combined with Group Policy for better security, as advanced users can bypass it.

**▪ Application Whitelisting Tools**

1. Thycotic

**Type:** Privileged Access and Application Control  
**Use Case:**

* Controls both who can run apps and which apps can run.
* Ideal for environments needing both identity and application control.
* Integrates with Active Directory.

1. Kaspersky Whitelist

**Functionality:**

* Centralized allowlist management.
* Uses a trusted application database.
* Works with endpoint protection.

**Use Case:**  
Environments needing automated trust evaluation and strong malware defense.

**LO#02:** Discuss and implement application Sandboxing

* 1. **▪ Application Sandboxing**
  2. **Definition:**  
     Application sandboxing is a **security mechanism** that isolates applications in a controlled environment (sandbox) to prevent them from affecting the host system or accessing sensitive resources.
  3. **Purpose:**
  4. Prevent malware or untrusted code from impacting the system.
  5. Analyze suspicious programs safely.
  6. Enforce least privilege for applications.
  7. **Benefits:**
  8. Contains threats
  9. Protects OS and data
  10. Ideal for testing and secure browsing environments

1. **Application Sandbox Examples**

| * 1. **Platform** | * 1. **Example** |
| --- | --- |
| * 1. Windows | * 1. Windows Sandbox, Windows Defender Application Guard |
| * 1. Linux | * 1. Firejail, Flatpak, Snap |
| * 1. Cross-platform | * 1. Sandboxie, VMware/VirtualBox, Docker (container sandboxing) |

* 1. **▪ Run Applications in Windows Sandbox**
  2. **Tool:** *Windows Sandbox* (Windows 10/11 Pro & Enterprise)
  3. **Key Features:**
  4. Lightweight, disposable VM.
  5. Isolated environment to run untrusted apps.
  6. Automatically resets after closure (no persistence).
  7. **Use Case:**
  8. Safe testing of unknown applications.
  9. Temporary, clean Windows environment.
  10. **Enable via:**  
      Optional Features → Turn Windows Features On/Off → *Windows Sandbox*
  11. ▪ **Sandboxing in Linux: Firejail**

**Tool:** *Firejail*

**Function:**  
A SUID program that uses Linux namespaces, seccomp, and capabilities to sandbox applications.

**Features:**

* Restricts filesystem, network, and device access.
* Lightweight, with pre-configured profiles for common apps (e.g., Firefox, VLC).

**Example Usage:**

firejail firefox

**Use Case:**  
Run user applications with minimal privileges in isolated environments.

* 1. ▪ **Sandboxing Approaches in Linux**
  2. **Popular Techniques:**
  3. **Namespaces:** Isolate process IDs, users, mounts, etc.
  4. **Seccomp (Secure Computing Mode):** Restrict system calls.
  5. **AppArmor/SELinux:** Mandatory access control (MAC) systems.
  6. **cgroups:** Control and limit system resource usage.
  7. **Combined Use:**  
     These techniques often work together to form custom sandboxes or containers (e.g., Docker, LXC).
  8. **▪ Sandboxing Tool: Sandboxie**
  9. **Platform:** Windows
  10. **Function:**  
      Runs applications in an isolated sandbox on Windows, preventing them from making permanent changes.
  11. **Features:**
  12. Sandboxes browsers, email clients, installers.
  13. Blocks access to system files and registry.
  14. Allows controlled internet access.
  15. **Use Case:**  
      Lightweight sandbox for personal or small-scale use.
  16. ▪ Additional Sandboxing Tools

| * 1. **Tool** | * 1. **Description** |
| --- | --- |
| * 1. **Docker** | * 1. Container-based app isolation (mostly DevOps use) |
| * 1. **Qubes OS** | * 1. OS-level sandboxing with VM isolation |
| * 1. **Firetools** | * 1. GUI frontend for Firejail (Linux) |
| * 1. **Flatpak/Snap** | * 1. Linux app distribution systems with built-in sandboxing |

* 1. ▪ **Windows Defender Application Guard: Microsoft Edge**
  2. **Tool:** *Windows Defender Application Guard (WDAG)*
  3. **Function:**  
     Runs Microsoft Edge in a **Hyper-V-based sandbox**, isolated from the host OS.
  4. **Features:**
  5. Protects the system from browser-based threats.
  6. Sites run in a secure container with no access to local resources.
  7. Works with enterprise-controlled trusted sites.
  8. **Use Case:**  
     Safe web browsing in enterprise environments.

**LO#03:** Discuss and implement Application Patch Management

**▪ Application Patch Management**

**Definition:**  
Application Patch Management is the process of **identifying, acquiring, testing, and deploying software updates (patches)** to fix vulnerabilities, improve functionality, or ensure compliance.

**Purpose:**

* Address known vulnerabilities (security patches).
* Fix bugs and performance issues.
* Maintain system and software integrity.
* Ensure compliance with security standards and regulations.

**Key Phases:**

1. Patch identification
2. Risk assessment and prioritization
3. Testing in a controlled environment
4. Deployment
5. Verification and rollback (if needed)

**▪ Software Patch Management for third-party software using Patch Manager**

**Need for Third-Party Patch Management:**

* Operating systems may auto-update, but third-party software (e.g., Chrome, Adobe, Java) often lacks centralized updates.
* These apps are frequent targets of cyberattacks due to unpatched vulnerabilities.

**Patch Manager Tools:**  
Designed to automate patching for third-party apps, often integrated with endpoint or systems management platforms.

**Capabilities:**

* Auto-detection of outdated software
* Scheduled patch deployment
* Patch testing and approval workflows
* Reporting and compliance dashboards

**Example:**  
Using **SolarWinds Patch Manager** to update Java, Firefox, and Adobe Reader across the organization.

* 1. ▪ **Application Patch Management Solutions and Tools**
  2. Below are commonly used tools for enterprise-level patch management:

1. **Verismic CMS Patch Manager**

**Type:** Cloud-based Patch Management

**Features:**

* Agentless technology
* Patches both Windows and third-party applications
* Real-time compliance monitoring
* Easy-to-use web-based interface

**Best For:**  
Organizations looking for **cloud-native**, low-maintenance patching solutions.

1. **Shavlik Protect**

**Type:** On-premise and cloud-capable patch management tool

**Features:**

* Broad third-party application patching support
* Patch scanning, deployment, and rollback
* Integrates with SCCM (System Center Configuration Manager)

**Best For:**  
Businesses requiring **granular control** over both Microsoft and third-party software patching.

1. **IBM BIGFix**

**Type:** Enterprise-grade Endpoint Management and Patch Solution

**Features:**

* Real-time visibility and control of patch status
* Supports multiple OS platforms (Windows, macOS, Linux, UNIX)
* Automates patch compliance across distributed environments

**Best For:**  
Large enterprises managing **thousands of endpoints** with cross-platform requirements.

1. **Flexera Corporate Software Inspector**

**Type:** Vulnerability and Patch Management Tool

**Features:**

* Identifies unpatched software vulnerabilities
* Integrates with SCCM and other deployment systems
* Offers detailed threat intelligence

**Best For:**  
Organizations focused on **vulnerability management** alongside patching.

**LO#04:** Discuss and implement Web Application Firewall (WAF)

* **Web Application Firewall (WAF)**

**Definition:**  
A Web Application Firewall (WAF) is a security solution that **monitors, filters, and blocks** HTTP/S traffic to and from a web application. It operates at the **application layer (Layer 7)** of the OSI model.

**Purpose:**

* Protects web applications from **common attacks** such as SQL injection, XSS, CSRF, file inclusion, etc.
* Detects and blocks **malicious HTTP requests** based on predefined rules or behavioral patterns.

**Placement:**  
Usually deployed **in front of web servers** to inspect all incoming/outgoing traffic.

1. **Types of WAF**

| **Type** | **Description** |
| --- | --- |
| **Network-based WAF** | Hardware appliances placed inline to filter traffic at high speed. |
| **Host-based WAF** | Software integrated into the application server. Offers deep customization. |
| **Cloud-based WAF** | SaaS-based solution managed by a provider. Easy to deploy and scale. |

1. **Benefits of WAF**

* **Real-time protection** against application-layer attacks.
* **Zero-day threat mitigation** using behavioral analysis.
* **Compliance support** (PCI-DSS, HIPAA, etc.).
* **Custom rule creation** for application-specific protection.
* **Traffic monitoring** and logging for auditing.

1. **WAF Limitations**

* Cannot protect against attacks beyond the HTTP/HTTPS layer (e.g., DoS at network level).
* Might produce **false positives** or block legitimate traffic.
* Requires **ongoing tuning** and **rule updates**.
* Cannot fix insecure coding or poor backend logic.
* **Configuring URLScan to setup as WAF For IIS Server**

**Tool:** *URLScan* – Microsoft’s lightweight security tool for IIS.

**Purpose:**  
Restricts the types of HTTP requests that IIS will process, acting as a basic WAF.

**Key Configuration Options:**

* Block specific HTTP verbs (e.g., TRACE, DELETE).
* Limit request length (headers, query string, etc.).
* Deny URLs with certain characters or patterns.

**Steps:**

1. Install URLScan via IIS extension.
2. Configure urlscan.ini to define rules.
3. Monitor logs (urlscan.log) for violations.

**Use Case:**  
Basic protection for legacy or internal IIS-based web applications.

* **Open Source WAFs for Web Application Security**

| **Tool** | **Description** |
| --- | --- |
| **ModSecurity** | Most popular open-source WAF, works with Apache, Nginx, and IIS. Supports OWASP Core Rule Set (CRS). |
| **NAXSI** | Nginx Anti-XSS & SQL Injection WAF module. Lightweight and customizable. |
| **OpenWAF** | Modular open-source WAF focused on API security and microservices. |
| **Shadow Daemon** | PHP-based open-source WAF focused on code-level injection prevention. |

**Benefits of Open Source WAFs:**

* Free to use and modify
* Community-supported rule sets (e.g., OWASP CRS)
* Highly customizable and scriptable

**Sub Domain: Data Security**

**LO#01:** Understand Data Security and its Importance

▪ **What is Business Critical Data?**

**Definition:**  
Business Critical Data refers to information that is **essential to an organization’s operation, reputation, compliance, or revenue**. Loss, compromise, or unauthorized access to this data can result in severe financial, legal, and reputational damage.

**Examples Include:**

* Customer records
* Intellectual property
* Financial reports
* Trade secrets
* Employee PII (Personally Identifiable Information)

▪ **Examples of Critical Data**

| **Data Type** | **Example** |
| --- | --- |
| **Customer Data** | Names, credit card numbers, addresses |
| **HR Records** | Employee ID, salary, SSN |
| **Financial Data** | Invoices, balance sheets, tax records |
| **Healthcare Information** | Patient medical history, diagnosis |
| **Authentication Data** | Passwords, encryption keys |

▪ **The Need of Data Security**

**Reasons for Ensuring Data Security:**

* Prevent **data breaches** and **cyberattacks**.
* Ensure **regulatory compliance** (GDPR, HIPAA, PCI-DSS).
* Protect **customer trust** and business reputation.
* Maintain **business continuity** and resilience.
* Safeguard **competitive advantage** and proprietary information.

**▪ Data Security**

**Definition:**  
Data Security involves **protective digital and physical measures** used to prevent unauthorized access, corruption, theft, or loss of data throughout its lifecycle.

**Key Principles:**

* **Confidentiality:** Only authorized users access data.
* **Integrity:** Data remains accurate and unaltered.
* **Availability:** Data is accessible when needed.

**▪ Example: Data At Rest vs Data in Use vs Data in Transit**

| **State** | **Description** | **Example** |
| --- | --- | --- |
| **Data at Rest** | Data stored on disks, databases, or backups | Files on a hard drive |
| **Data in Use** | Data currently being processed or accessed | Data in RAM or an open document |
| **Data in Transit** | Data moving across networks or between devices | Emails, file uploads, web traffic |

**Protection Methods:**

* At Rest: Full-disk encryption
* In Use: Endpoint protection, RAM encryption
* In Transit: TLS/SSL, VPNs

**▪ Data Security Technologies**

1. Data Erasure

**Definition:**  
Data Erasure (also known as secure wipe) is a method of **permanently deleting data** from storage devices in a way that it **cannot be recovered**, even with forensic tools.

**Use Case:**  
Used during hardware disposal, repurposing drives, or meeting compliance standards.

**Tools:**  
Blancco, DBAN (Darik's Boot and Nuke)

1. Hardware-Based Security

**Definition:**  
Security mechanisms **built directly into physical hardware** (like hard drives, processors, or USBs) to enforce data protection.

**Examples:**

* Self-Encrypting Drives (SEDs)
* Trusted Platform Module (TPM) chips
* Hardware Security Modules (HSMs)

**Benefits:**

* Faster performance than software-based methods
* Reduced risk of software-based attacks
* Enhanced protection even if OS is compromised

**LO#02*:*** Understand Data Integrity and Its Importance

**▪ What is Data Integrity**

**Definition:**  
Data Integrity refers to the **accuracy, consistency, and reliability** of data throughout its lifecycle. It ensures that the data remains **unchanged** during transit, storage, and processing unless altered by authorized processes.

**Key Focus:**

* Preventing unauthorized data modification
* Detecting corruption or loss
* Maintaining trustworthy information

▪ **Types of Data Integrity**

| **Type** | **Description** |
| --- | --- |
| **Physical Integrity** | Ensures data is stored and retrieved correctly (e.g., no damage or hardware failure). |
| **Logical Integrity** | Ensures data remains logically accurate and consistent across systems or databases. |

Subtypes of Logical Integrity:

* **Entity Integrity** – Unique identifiers (primary keys)
* **Referential Integrity** – Consistent relationships between tables
* **Domain Integrity** – Valid entries (e.g., numeric in number fields)
* **User-defined Integrity** – Custom rules specific to applications

**▪ Data Integrity Checking**

**Definition:**  
The process of **verifying the correctness and consistency** of data using various tools, rules, and techniques.

**Common Methods:**

* Checksums and Hashing (e.g., MD5, SHA-256)
* Digital Signatures
* File Integrity Monitoring (FIM)
* Database constraints and validations

**▪ Checklist to Preserve Data Integrity**

1. Use **validated input** (sanitize data at entry points).
2. Implement **access control and authentication**.
3. Use **encryption** and **checksums** for data in transit/storage.
4. Perform **regular backups** and **data audits**.
5. Monitor logs and use **FIM tools**.
6. Apply **database constraints** and foreign keys.
7. Keep systems and applications **patched and secure**.

**▪ Data Integrity Checking Tools**

| **Tool** | **Functionality** |
| --- | --- |
| **Tripwire** | File integrity monitoring and alerting |
| **AIDE** | Linux-based integrity checker using database snapshots |
| **HashMyFiles** | Lightweight hashing tool for file integrity checks |
| **MD5/SHA Tools** | Verify data blocks and file transmission integrity |
| **DBMS Built-in Checks** | Primary key, NOT NULL, foreign key constraints |

**▪ Data Integrity vs. Data Quality vs. Data Security vs. Data Accuracy**

| **Concept** | **Focus Area** |
| --- | --- |
| **Data Integrity** | Consistency and trustworthiness across lifecycle |
| **Data Quality** | Relevance, completeness, timeliness, and correctness |
| **Data Security** | Protection from unauthorized access and breaches |
| **Data Accuracy** | Precision and correctness of data values |

**▪ Role of Data Integrity in Terms of GDPR Compliance**

**Importance in GDPR:**

* Article 5 of GDPR mandates **accuracy and integrity** of personal data.
* Data controllers must ensure that data is **not altered unlawfully**.
* Integrity supports **accountability, auditability, and trust** in handling personal data.

**GDPR Principles Supported by Data Integrity:**

* Lawfulness, fairness, and transparency
* Accuracy
* Integrity and confidentiality (security)

**LO#02:** Discuss the implementation of data access controls

* **Logical Implementation of Access Controls**

**Definition:**  
Logical access controls are **software-based mechanisms** that regulate who can access what data and what actions they can perform. They enforce **authentication**, **authorization**, and **accountability** across systems and applications.

**Key Elements:**

* User identification (e.g., username, ID)
* Authentication (e.g., passwords, biometrics)
* Authorization (e.g., roles, permissions)
* Logging and monitoring (audit trails)
* **Access Controls List (ACL)**

**Definition:**  
An ACL is a **table of permissions** attached to files, directories, or objects that specifies which **users or system processes** can access the object and what operations they can perform.

**Common Permissions:**

* Read
* Write
* Execute
* Modify
* Full Control

1. Setting Access Controls and Permission to Files and Folders in Windows

**How It Works:**

* Access permissions can be set via the file/folder **Properties → Security tab**.
* Users and groups are assigned specific rights.
* **NTFS permissions** provide detailed control.

**Steps:**

1. Right-click file/folder → Properties
2. Go to **Security tab**
3. Click **Edit** to add users and set permissions
4. Apply and confirm

**Example:**  
Allow "Read" permission to UserA, deny "Write" to UserB.

1. Setting Access Controls and Permission to Files and Folders in Linux

**How It Works:**

* Linux uses **file permission bits (rwx)** and **ownership (user/group/others)**.
* Permissions managed with chmod, chown, and setfacl.

**Basic Commands:**

* chmod 755 file.txt – Owner can read/write/execute, others can read/execute
* chown user:group file.txt – Change file owner and group
* getfacl / setfacl – For extended ACLs

**Example:**

chmod 700 secret.doc

chown alice:admin secret.doc

**▪ Group Policy**

**Definition:**  
Group Policy is a feature in Windows that allows **centralized management** of settings and permissions across user accounts and devices within an Active Directory (AD) domain.

**Uses for Access Control:**

* Restrict access to drives
* Disable USB ports
* Set password policies
* Enforce logon restrictions

**Tool:** Group Policy Management Console (GPMC)

**▪ Passwords /Access Token**

**Passwords:**

* First layer of authentication. Must be strong and regularly updated.

**Access Tokens:**

* A digital object issued after authentication that contains **user identity and permission claims**.
* Used in modern access control models (e.g., OAuth, Kerberos).

**Purpose:**  
Both serve as **proof of identity** and help enforce access rules in systems and applications.

**▪ Account Restrictions**

**Purpose:**  
Limit what users can do based on predefined rules, reducing the risk of misuse.

**Types of Restrictions:**

* Logon hours
* Workstation login restrictions
* Concurrent logins
* Time-bound accounts

1. Restricting Logon Hours for Linux Users

**Method:**  
Edit the /etc/security/time.conf file and configure PAM (Pluggable Authentication Module).

**Example Rule:**

login;\*;username;!SaSu

Denies username from logging in on Saturday and Sunday.

**Enable in PAM:** Add to /etc/pam.d/login:

account required pam\_time.so

**Purpose:**  
Control user activity during off-hours to **minimize insider threats** or unauthorized access.

**LO#03:** Discuss the implementation of encryption of

“Data at rest”

**Encrypting “Data-at-Rest”**

**Definition:**  
Data at rest refers to **inactive data stored** on physical media (hard drives, SSDs, backups, etc.). Encryption ensures this data is **unreadable without proper authorization**, even if physical access is gained.

**Goal:**  
Protect sensitive information against theft, unauthorized access, or device loss.

**Disk Encryption**

1. Implementing Built-in Disk Encryption for Windows 11

**Tool:** **BitLocker**

**Steps:**

1. Go to **Settings → Privacy & Security → Device Encryption**
2. Enable BitLocker for the system or individual drives
3. Backup the recovery key (Microsoft account or external storage)

**Note:** Requires TPM 2.0 for automatic unlocking.

**Enable Trusted Platform Module (TPM) in Windows**

**TPM:** A hardware chip that stores cryptographic keys securely.

**Steps to Enable TPM:**

1. Enter BIOS/UEFI settings on startup
2. Find **TPM** or **Security Chip**
3. Enable and save settings

**Purpose:** BitLocker uses TPM to protect encryption keys from being accessed offline.

1. **Implementing Built-in Disk Encryption for MacOS**

**Tool:** **FileVault**

**Steps:**

1. Go to **System Settings → Privacy & Security → FileVault**
2. Turn on FileVault
3. Store recovery key with Apple ID or manually
4. **Implementing Built-in Disk Encryption in Linux**

**Tool:** **LUKS (Linux Unified Key Setup)**

**Implementation via:** cryptsetup

**Example:**

cryptsetup luksFormat /dev/sdX

cryptsetup open /dev/sdX secure\_volume

Can be enabled during OS installation for full disk encryption.

1. **Implementing Built-in Disk Encryption in Android** Devices

**Tool:** **File-based Encryption (FBE)**

**Steps:**

1. Enable screen lock with PIN/password
2. Encryption activates automatically on supported devices (Android 10+)
3. Can check via **Settings → Security**
4. **Implementing Built-in Disk Encryption in iOS devices**
5. **Encryption is always enabled** by default once passcode is set.
6. **Uses:** Hardware-based encryption with **AES 256-bit** and Secure Enclave.
7. **Third Party Disk Encryption Tools**

| **Tool** | **Platform** | **Features** |
| --- | --- | --- |
| VeraCrypt | Windows/Linux | Open-source, hidden volumes |
| Symantec Endpoint Encryption | Windows/Mac | Enterprise-grade management |
| McAfee Complete Data Protection | Windows | Includes policy-based encryption |

* **File Level Encryption**
  + - Implementing Built-in File System-level Encryption on Windows

**Tool:** **Encrypting File System (EFS)**

**Steps:**

1. Right-click file/folder → **Properties → Advanced**
2. Check “Encrypt contents to secure data”
3. Backup encryption certificate if prompted
4. Third-Party Windows File Encryption Tools

| **Tool** | **Feature** |
| --- | --- |
| AxCrypt | Strong AES-256 encryption, simple interface |
| 7-Zip | File compression with built-in encryption |
| Folder Lock | Encrypts, hides, and password-protects files |

* + - Implementing Built-in File System-level Encryption on MacOS

**Tool:** **Disk Utility (Encrypted Disk Images)**

**Steps:**

1. Open Disk Utility → File → New Image → Blank Image
2. Set **Encryption: AES-128 or AES-256**
3. Mount as virtual encrypted drive
   * + Third-Party Linux File Encryption Tools

| **Tool** | **Description** |
| --- | --- |
| GnuPG | CLI-based, strong public/private key encryption |
| eCryptfs | Transparent file encryption (used by Ubuntu home folders) |
| CryFS | Encrypts files for cloud sync use (e.g., Dropbox) |

* + Removable Media Encryption
  1. Implementing Removable Media Encryption in Windows

**Tool:** **BitLocker To Go**

**Steps:**

1. Insert USB → Right-click → Turn on BitLocker
2. Set password and backup recovery key
3. Drive will be auto-encrypted
   1. Implementing Removable Media Encryption in Mac

**Steps:**

1. Right-click USB in Finder → **Encrypt "Drive Name"**
2. Set a password and hint
3. Drive becomes HFS+ encrypted (Mac-only readable)
   1. Implementing Removable Media Encryption in Linux

**Tool:** **LUKS on USB Drives**

**Steps:**

1. Format USB with LUKS using cryptsetup
2. Mount using passphrase or keyfile
   * + Database Encryption
     + MS SQL Server

Implementation of Transparent Database Encryption in MS SQL Server

Field level encryption

Application-level encryption

* + - Encryption: Implementation of Column-level Encryption in MS SQL Server
    - Implementation of Always Encrypted in MS SQL Server
    - Oracle
    1. Implementation of Transparent Data Encryption in Oracle

Encrypts sensitive data in tablespaces, columns, or backups.

**Implemented via:** Oracle Advanced Security

**Steps:**

* Create Wallet
* Enable TDE
* Apply encryption policy at column or tablespace level
* Data at Rest Encryption Best Practices
  1. Use **full disk encryption** for endpoints and servers
  2. Always **encrypt backups** and removable media
  3. Combine **file-level and disk-level** encryption for layered protection
  4. Use **strong key management** (HSMs, KMS)
  5. Ensure **compliance** with standards (FIPS 140-2, GDPR, HIPAA)
  6. Regularly **audit and test** encryption mechanisms
  7. Educate users on handling encrypted data and media

**LO#04:** Discuss the implementation of Encryption of

“Data at transit”

**Encryption of data in transit** protects data while it's being transferred across networks. It ensures that even if intercepted, the data remains unreadable to unauthorized users. Common methods include using SSL/TLS for websites, VPNs for secure connections, and encrypted email protocols. This prevents data breaches, maintains confidentiality, and supports compliance with security standards.

**LO#4.1:** Discuss the implementation of Encryption of

“Data at transit” between browser and web server

To encrypt data between a browser and web server, **HTTPS** is used via **SSL/TLS certificates**. This ensures all data sent (like passwords or payment info) is encrypted, preventing interception or tampering during transit.

* **Secure HTTP Connection using Digital Certificate**

**HTTPS (HTTP Secure)** encrypts communication between a **web browser and web server** using **TLS** and a **digital certificate**.

**How it works:**

1. Browser connects to server via HTTPS.
2. Server sends its **digital certificate**.
3. Browser validates certificate via **CA (Certificate Authority)**.
4. A **secure TLS session** is established (handshake).
5. Data exchanged is encrypted end-to-end.

* **Viewing a Digital Certificate**

In any browser (e.g., Chrome):

1. Click the padlock icon in the address bar.
2. Go to **"Certificate" → "Details"** to explore the following fields:
   * 1. Version 1 Fields

Core fields defined in X.509 v1:

* **Subject:** Entity to whom the certificate is issued.
* **Issuer:** Certificate Authority (CA).
* **Validity Period:** Start and expiry dates.
* **Serial Number:** Unique identifier.
* **Public Key Info:** Key algorithm and public key.
  + 1. Extensions

Add security and usage constraints:

* **Key Usage:** Defines allowed functions (e.g., digital signature, key encipherment).
* **Extended Key Usage:** Specifies application (e.g., TLS server auth, email).
* **Subject Alternative Name (SAN):** Lists domain names the certificate covers.
* **CRL Distribution Point:** Where to check certificate revocation.
  + 1. Certificate Path

Shows the **trust chain** from the **end-entity certificate** (website) to the **root CA**.

Includes:

* End-Entity Certificate (e.g., www.example.com)
* Intermediate CA (if any)
* Root CA (trusted authority)
  + 1. Root Certificate
* Installed in the browser/OS Trusted Root Certification Store.
* Root CAs are globally trusted, validating the authenticity of all child certificates.
* If not trusted, users will see security warnings.
* **Install and Configure SSL Certificate on Windows Server**

**For IIS (Internet Information Services):**

**Steps:**

1. Generate a Certificate Signing Request (CSR) in IIS.
2. Purchase or obtain SSL certificate from a CA.
3. Import the certificate into **IIS → Server Certificates**.
4. Bind HTTPS to your site:
   * Go to **Bindings → Add → Type: HTTPS**.
   * Select your installed certificate.

* **Backing Up and Exporting Digital Certificate in Windows Server**

**Using MMC (Microsoft Management Console):**

**Steps:**

1. Open MMC → Add Snap-in → Certificates → Local Computer.
2. Locate certificate under **Personal → Certificates**.
3. Right-click → **All Tasks → Export**.
4. Choose **Yes, export the private key** → .pfx format.
5. Set a password and export location.

* **Renew Certificate**
* Renewal is typically required **before expiration** (most CAs notify in advance).
* Generate a new CSR if needed.
* Rebind the new certificate to your site in IIS or server config.
* Always test the new cert before expiration.
* **Revoke Certificate**

Revocation is necessary when:

* Private key is compromised.
* Domain is no longer in use.
* Certificate info becomes invalid.

**Methods:**

1. CA publishes **CRL (Certificate Revocation List)**.
2. Or use **OCSP (Online Certificate Status Protocol)** for real-time validation.

Revoked certificates will appear **untrusted** to users and systems.

**LO#4.2:** Discuss the implementation of Encryption of

“Data at transit” between database server and web server

When a web application communicates with a **database server**, sensitive data like credentials, queries, and records are transferred over the network. If this traffic is **not encrypted**, it's vulnerable to interception and man-in-the-middle (MITM) attacks.

* **Enabling Encrypted connections for an instance of the SQL Server Database Engine**

SQL Server supports **SSL/TLS encryption** to protect data in transit.

**Steps to Enable Encryption in SQL Server:**

1. **Obtain and Install an SSL Certificate:**
   * Use a certificate from a trusted **Certificate Authority (CA)**.
   * The certificate must include the **fully qualified domain name (FQDN)** of the SQL Server.
2. **Assign the Certificate in SQL Server Configuration Manager:**
   * Open **SQL Server Configuration Manager** → SQL Server Network Configuration → Protocols for MSSQLSERVER.
   * Right-click **Properties** → **Certificate tab** → Select installed certificate.
3. **Force Encryption (Optional):**
   * In **Flags tab**, set **Force Encryption** to **Yes**.
   * Requires clients to use encrypted connections.
4. **Restart SQL Server Service** to apply changes.

**✅ Client-Side Encryption Option:**

* Modify the connection string:

Encrypt=True;TrustServerCertificate=False;

* If using TrustServerCertificate=True, encryption occurs, but the certificate is not validated (not recommended for production).
* **Enabling SSL/TLS encryption in Oracle Server**

Oracle supports **native network encryption** using **SSL/TLS** for secure communication between clients and servers.

**✅ Steps to Enable SSL/TLS in Oracle:**

1. **Create a Wallet (Oracle Wallet Manager):**
   * Store the server’s **private key** and **digital certificate**.
   * Oracle Wallet can be created using orapki or **Oracle Wallet Manager (OWM)**.
2. **Enable Oracle Advanced Security:**
   * Modify sqlnet.ora on the **server and client**.

Example configuration (on **server**):

SSL\_VERSION = 1.2

WALLET\_LOCATION = (SOURCE=(METHOD=FILE)(METHOD\_DATA=(DIRECTORY=/etc/oracle/wallet)))

SSL\_CLIENT\_AUTHENTICATION = FALSE

SSL\_CIPHER\_SUITES = (SSL\_RSA\_WITH\_AES\_256\_CBC\_SHA)

1. **Update Listener to Accept SSL:**
   * Modify listener.ora to add an **SSL-enabled listener**.

Example:

(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=hostname)(PORT=2484)))

1. **Connect Using TCPS (SSL-enabled protocol):**
   * The client must also have SSL configured.
   * Use a **TCPS connection string** instead of TCP.

**LO#4.3:** Discuss the implementation of Encryption of

“Data at transit” in Email Delivery

When emails are sent over the internet, they travel across various networks and servers. If not encrypted, this makes them **vulnerable to interception**, tampering, and unauthorized access. Encrypting **email data in transit** ensures **confidentiality and integrity**.

* **Email Encryption**

There are two primary layers of email encryption:

1. **Transport Layer Encryption (TLS)** – protects emails as they travel between mail servers.
2. **End-to-End Encryption** – protects the email contents from sender to recipient.
   * 1. MS Outlook

Outlook supports **S/MIME (Secure/Multipurpose Internet Mail Extensions)** and **Microsoft 365 Message Encryption (OME)**.

**🔐 S/MIME (Digital Certificate-Based Encryption)**

**Setup Steps:**

1. **Obtain a Digital Certificate** from a trusted Certificate Authority (CA).
2. Import the certificate into Outlook.
3. Go to:
   * File → Options → Trust Center → Trust Center Settings → Email Security
4. Enable **"Encrypt contents and attachments"** for all messages or per message.

**Sending Encrypted Email:**

* Compose new mail → Options tab → Click **"Encrypt"** or **"Sign"**.
* Recipient must also have S/MIME configured.

**🔐 Microsoft 365 Message Encryption (OME)**

* Available with Microsoft 365 subscriptions.
* Encrypts messages using **Azure Information Protection**.
* Recipients can open encrypted messages even without Outlook (via secure web portal).
  + 1. Gmail

Gmail uses **TLS by default** for encrypting messages in transit between mail servers—**if the recipient’s email provider also supports TLS**.

**🔐 Built-in Gmail TLS Encryption (Opportunistic TLS)**

* Automatically applied.
* No user action needed.
* A **gray or green padlock icon** shows if TLS is enabled between you and the recipient.

**🔐 End-to-End Encryption Options in Gmail:**

**A. S/MIME for Google Workspace Users**

* Only available for **Google Workspace Enterprise Plus**.
* Requires digital certificates.
* Admin must enable **S/MIME** in Google Admin Console.

**B. Confidential Mode (Not true encryption)**

* Sends emails with expiration dates and restrictions (e.g., no forwarding).
* Does **not use encryption**, but offers some access control.

**C. Third-party Encryption Tools**

* Use browser extensions or services like:
  + **FlowCrypt (PGP-based)**
  + **Virtru**
  + **ProtonMail Bridge** (for desktop clients)

**LO#5:** Discuss Data Masking Concepts

* **Data Masking**

**Data masking** is the process of hiding **sensitive data** by replacing it with **anonymized or scrambled data** that retains its format, making it useful for **testing, development**, or **training** without exposing actual sensitive information.

The main goal is to ensure **data privacy** and **security**, particularly in environments where real data is not needed but the data structure must be preserved.

* **Types of data masking**

There are several methods and strategies for **data masking**, depending on the desired outcome and use case.

1. Deterministic data masking

**Definition:**  
In **deterministic masking**, the same input data always produces the same masked result. This is critical for applications where the **relationship between data values** needs to be maintained across different datasets.

**Example:**

* Original Name: **John Doe**
* Masked Name: **A1B2 C3D4**  
  The same name will always be masked as **A1B2 C3D4**.

**Use Case:**

* When you need to maintain **referential integrity** between related data, such as in test environments with database relationships.

1. Statistical data obfuscation

**Definition:**  
This technique involves **altering data** such that it **resembles real data** statistically but is not traceable to the original source. It preserves the **distribution, range, and statistical properties** of the original data without revealing real values.

**Example:**

* Original Salary: **$80,000**
* Masked Salary: **$78,000**  
  The masked value stays within the same statistical range, but the exact salary is not exposed.

**Use Case:**

* When working with **aggregate data** or **statistical models** where individual data points are not important but trends or patterns must be preserved.
* **Data Masking Techniques**

There are various techniques to mask data, depending on the level of obfuscation and use case.

**1. Substitution**

* **Replace original values** with **random but valid values** (e.g., replacing a name with another, or a credit card number with a valid but fake one).
* **Example:**
  + Original: **John Smith**
  + Masked: **Jane Doe**

**2. Shuffling**

* **Shuffling data** involves randomly reordering values in a dataset.
* **Example:**
  + Original: **John Smith**, **Mary Johnson**, **James Lee**
  + Masked: **James Lee**, **Mary Johnson**, **John Smith**

**3. Nulling Out**

* **Null out** or remove sensitive data entirely by replacing it with **null values** or a generic placeholder.
* **Example:**
  + Original: **123-45-6789**
  + Masked: **NULL**

**4. Encoding**

* **Encode data** by transforming it into a format that's not meaningful, such as base64 or hexadecimal encoding.
* **Example:**
  + Original: **Sensitive Email**
  + Masked: **U29tZSBFbWFpbCBhZGRyZXNz** (Base64 encoding of email)

**5. Date Aging**

* Change the **date fields** so they reflect a realistic time period but not the actual date.
* **Example:**
  + Original: **2023-04-19**
  + Masked: **2020-06-15**
* **Implementing Dynamic Data Masking in SQL Server 2022**

**Dynamic Data Masking (DDM)** is a feature in **SQL Server** that **automatically masks sensitive data** when queried by users without appropriate privileges.

**Steps to Implement DDM in SQL Server:**

1. **Create a Database and Table:**

CREATE DATABASE TestDB;

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

FullName NVARCHAR(100) MASKED WITH (FUNCTION = 'default()'),

SSN CHAR(9) MASKED WITH (FUNCTION = 'partial(0,"XXX-XX-",4)'),

DateOfBirth DATE MASKED WITH (FUNCTION = 'default()')

);

1. **Define Masking Function:**
   * Use **partial**, **default**, or **email** masking functions to hide sensitive information.
   * partial(0, "XXX-XX-", 4) masks the SSN, showing only the last 4 digits.
2. **Test Masking:**
   * Users without the appropriate permissions will see the masked data, while privileged users can access the full data.
3. **Grant Permissions:**
   * The UNMASK permission is required to see unmasked data.

* **Implementing Data Masking in Oracle Database**

Oracle provides data masking capabilities through the **Oracle Data Masking and Subsetting** tool.

**Steps to Implement Data Masking in Oracle:**

1. **Install Oracle Data Masking:**
   * Ensure you have Oracle Enterprise Manager (OEM) and **Oracle Data Masking** installed.
2. **Create a Data Masking Plan:**
   * In OEM, navigate to **Data Masking** and create a new **masking plan**.
   * Choose a target database and select the columns for masking.
3. **Define Masking Formats:**
   * You can choose predefined formats or customize how data is masked (e.g., replace SSNs with random valid numbers).
4. **Execute the Masking Plan:**
   * Apply the plan to a **copy of the production database** or a **test environment** to protect sensitive data.

* **Data Masking Tools**

Several tools can help automate and manage data masking. Some of the popular ones are:

1. **Informatica Data Masking**
   * Provides data masking and subsetting capabilities to mask sensitive data across **databases, applications, and files**.
2. **Oracle Data Masking and Subsetting**
   * Integrated with Oracle databases to mask sensitive data and provide privacy for non-production environments.
3. **Delphix**
   * Provides data masking for **databases** and **files** in hybrid cloud and on-prem environments.
4. **DataVeil**
   * A tool specifically designed to mask data in **databases** like Oracle, SQL Server, MySQL, and PostgreSQL.
5. **Solix Data Masking**
   * Focuses on **non-production environments** to safely mask sensitive data, allowing analytics and testing without exposure.

**LO#6:** Discuss data backup and retention

**▪ Introduction to Data Backup**

**Data backup** refers to creating and storing copies of critical data to ensure its protection in case of **data loss** due to hardware failure, accidental deletion, corruption, or disasters. It is essential for **data recovery** and maintaining business continuity.

**▪ Data Backup Strategy/Plan**

A **data backup strategy** involves planning how and when data will be backed up, what tools and technologies will be used, and ensuring that backup policies meet organizational **RPO (Recovery Point Objective)** and **RTO (Recovery Time Objective)**.

* **RPO**: The maximum acceptable amount of data loss in case of a failure.
* **RTO**: The maximum acceptable time it takes to recover data after a failure.

**▪ Identify Critical Business Data**

Identify and prioritize **critical business data** that requires frequent and secure backups. Examples include:

* Customer information
* Financial records
* Intellectual property
* Legal documents
* Databases and application configurations

**▪ Selecting the Backup Media**

Choosing the right **backup media** is vital for ensuring the reliability, speed, and security of backups.

**▪ Examples of Data Backup Media Devices**

Common backup media include **disk-based storage, tape drives, cloud storage**, etc.

* 1. **▪ RAID (Redundant Array Of Independent Disks) Technology Advantages/Disadvantages of RAID Systems**
  2. RAID is a technology that uses multiple hard drives to improve performance, redundancy, and data storage reliability.
  3. **RAID Storage Architecture**
  4. RAID Storage Architecture combines multiple physical disks into one logical unit to improve performance, fault tolerance, or both. Data is distributed across drives using different techniques like striping, mirroring, and parity, depending on the RAID level. It helps ensure data availability even if one or more drives fail.
  5. **RAID Level 0: Disk Striping**

**Description:** Data is split into blocks and distributed across multiple drives.

**Advantages:** Improved performance.

**Disadvantages:** No redundancy, data loss if one drive fails.

* 1. **RAID Level 1: Disk Mirroring**

**Description:** Data is mirrored (duplicated) on two or more disks.

**Advantages:** Redundancy, fault tolerance.

**Disadvantages:** Higher cost due to extra storage requirements.

* 1. **RAID Level 3: Disk Striping with Parity**

**Description:** Data is striped across multiple disks, with one disk storing parity information for fault tolerance.

**Advantages:** Fault tolerance and improved performance.

**Disadvantages:** Performance issues on write-heavy workloads.

* 1. **RAID Level 5: Block Interleaved Distributed Parity**

**Description:** Data is striped across multiple disks, and parity information is distributed across the drives.

**Advantages:** Efficient storage and fault tolerance.

**Disadvantages:** Slower write performance due to parity calculation.

* 1. **RAID Level 10: Blocks Striped and Mirrored**

**Description:** Combines RAID 1 (mirroring) and RAID 0 (striping) for performance and redundancy.

**Advantages:** High performance and fault tolerance.

**Disadvantages:** Expensive due to doubled storage needs.

* 1. **RAID Level 50: Mirroring and Striping across Multiple RAID Levels**

**Description:** Combines RAID 5 and RAID 0, providing both striping and redundancy.

**Advantages:** High performance with fault tolerance.

**Disadvantages:** Expensive and complex.

* 1. **Selecting Appropriate RAID Levels**
  2. **Selecting Appropriate RAID Levels** depends on your needs for **performance**, **redundancy**, **storage capacity**, and **cost**.
  3. Choose **RAID 0** for high speed, no fault tolerance.
  4. Use **RAID 1** for full redundancy via mirroring.
  5. **RAID 5** balances speed, storage, and fault tolerance using parity.
  6. **RAID 10** combines striping and mirroring for high performance and reliability.
  7. **RAID 6** offers extra fault tolerance with double parity.
  8. The right RAID level depends on your **data criticality**, **budget**, and **performance requirements**.
  9. **Hardware and Software RAIDs**

**Hardware RAID** involves dedicated RAID controllers.

**Software RAID** is managed through the operating system.

* 1. **Using RAID Best Practices**

Always use **hot spare drives**.

Regularly **monitor RAID health** and perform **drive failure tests**.

Maintain **redundant power supplies**.

* 1. **Storage Area Network (SAN)**
  2. A **Storage Area Network (SAN)** is a high-speed network of storage devices that connects to servers, allowing multiple systems to access storage as if it were local.

1. SAN Advantages

* **Centralized storage** management.
* **High performance** due to direct connections.
* **Scalability** as data storage needs grow.

1. SAN Backup Best Practices

* Implement **data deduplication** to reduce storage costs.
* Use **snapshots** for quick backup and recovery.
* Regularly test **disaster recovery** procedures.

1. SAN Data Storage and Backup Management Tools

* Veritas NetBackup
* Commvault
* Dell EMC Data Domain
  1. **Network Attached Storage (NAS)**
  2. **NAS** is a file-level storage solution connected over a network, providing data access to multiple clients.

1. NAS Implementation Types: Integrated NAS System

* A complete storage solution with integrated hardware and software.

1. Examples of Integrated NAS System

* Example: Synology DiskStation

1. NAS Implementation Types: Gateway NAS System

Gateway NAS System: FreeNAS

* Provides NAS services using existing storage hardware.
* Example: **FreeNAS** (an open-source NAS system)
  1. **Selecting an Appropriate Backup Method**

Choose the backup method based on factors like recovery time, security, and storage costs. Common methods include:

**Full Backup:**

* 1. Backs up everything in the selected volume.
  2. **Advantages:** Fast recovery.
  3. **Disadvantages:** Time-consuming and storage-intensive.

**Incremental Backup:**

* 1. Only backs up data that has changed since the last backup.
  2. **Advantages:** Saves storage space and time.
  3. **Disadvantages:** Slower recovery due to multiple backup sets.

**Differential Backup:**

* 1. Backs up data changed since the last **full backup**.
  2. **Advantages:** Faster recovery than incremental.
  3. **Disadvantages:** Larger backups than incremental.
  4. **▪ Choosing the Backup Location**

Backup locations include:

* **On-site backups** (e.g., local storage, SAN)
* **Off-site backups** (e.g., cloud storage, remote servers)
* **Hybrid solutions** (combining both on-site and cloud)
  1. **▪ Types of Backup**
  2. **Cloud Backup:**
  3. **Advantages:** Off-site, scalable, and secure.
  4. **Disadvantages:** Requires internet access, potentially expensive for large volumes of data.
  5. **On-Site Backup:**
  6. **Advantages:** Faster recovery time.
  7. **Disadvantages:** Vulnerable to local disasters (e.g., fire, flood).
  8. **Tape Backup:**
  9. **Advantages:** Cost-effective for large data volumes.
  10. **Disadvantages:** Slower recovery times and high maintenance.
  11. **▪ Windows Data Backup: Disk, file and folders Backup**
  12. Disk, file, and folder backups can be performed using **Windows Backup** or third-party tools like **Acronis True Image** and **Veeam**.
  13. **▪ Third-Party Windows Data Backup Tools**

Third-party backup tools for Windows provide enhanced features beyond the built-in options. Popular tools include:

**Acronis True Image**: Full disk image backups and cloud backup.

**EaseUS Todo Backup**: Disk, file, and partition backup with cloud support.

**Macrium Reflect**: Disk imaging and cloning, with incremental backups.

* 1. **▪ Linux Data Backup: Disk, file and folders Backup**

Linux offers several built-in tools for backup:

* **rsync**: Command-line tool for efficient file synchronization and backups.
* **tar**: Used for archiving files and directories, often in combination with compression (e.g., .tar.gz).
* **dd**: For creating exact disk copies or backups of entire partitions.

  2. **▪ Third-Party Linux Data Backup Tools**

These tools offer additional functionality for Linux backups:

* **Clonezilla**: Open-source cloning tool for full disk backup and restore.
* **Duplicity**: Encrypted backup tool with cloud storage support.
* **Bacula**: Enterprise-grade backup software that supports multi-platform environments.
  1. **▪ Mac OS Data Backup: Disk, file and folders Backup**

macOS has several built-in options, but third-party tools provide additional features:

* **Time Machine**: Built-in tool for file and full system backups.
* **Carbon Copy Cloner**: Disk cloning and incremental backups.
* **SuperDuper!**: Simple, reliable disk cloning for macOS.
  1. **▪ Third-party MAC OS Data Backup Tools**
* **Acronis True Image:** Provides full disk image backups and cloud storage.
* **Backblaze:** Cloud-based backup solution with automatic file and folder backups.
* **Arq Backup:** Supports encrypted cloud backups for files and directories.
  1. **▪ Database Backup: MS SQL Server**

SQL Server offers various methods to back up data:

* **Full Backup**: Backs up the entire database.
* **Differential Backup**: Backs up changes made since the last full backup.
* **Transaction Log Backup**: Backs up changes made since the last log backup. Tools: SQL Server Management Studio (SSMS), PowerShell scripts.
  1. **▪ Database Backup: Oracle**

Oracle databases offer multiple backup types:

* **Cold Backup**: Backing up the database while it's offline.
* **Hot Backup**: Performing backups while the database is running.
* **RMAN (Recovery Manager)**: Oracle's tool for automated backup and recovery.
* **Data Pump**: For exporting and importing data in bulk.
  1. **▪ Email Backup: Outlook**
  2. **Outlook:** Use **PST files** for backup or configure backups via **Exchange Server**.
  3. **▪ Email Backup : Gmail**
  4. **Gmail:** Use **Google Takeout** for exporting emails or integrate with third-party tools like **Backupify**.
  5. **▪ Email Backup Tools**

1. MailStore Home / MailStore Server
2. SysTools Email Backup
3. Spinbackup
4. Gmvault
5. UpSafe Gmail/Office365 Backup
6. Altaro Office 365 Backup
7. EaseUS Todo Backup
   1. **▪ Web Server Configuration Backup: IIS**
   2. Use **IIS Manager** to export settings.
   3. **▪ Website Back Up**
   4. Regularly back up **files**, **databases**, and **configuration settings**.
   5. **▪ Data Backup Retention**

**Data retention policies** specify how long data backups should be kept. The length of time depends on **regulatory requirements** and business needs.

**Best Practices:**

* Keep **daily backups** for a short period (e.g., 30 days).
* Maintain **weekly backups** for longer periods (e.g., 6 months).
* Keep **monthly backups** for **long-term retention** (e.g., 7 years).
  1. **▪ Data Retention Policy Best practices**
* Regularly review and update retention policies to stay compliant with regulations (e.g., GDPR, HIPAA).
* Store backups in secure locations and ensure **encryption** of sensitive data.
* Periodically **test recovery** from backups to ensure effectiveness.

**LO#7:** Discuss Data Destruction Concepts

**▪ Data Destruction**

**Data destruction** is the process of permanently eliminating data from storage media to prevent unauthorized recovery or access. It is essential for ensuring **data privacy, compliance**, and **information security**, especially when disposing of old devices or decommissioning storage.

**▪ Data Destruction Policy**

A **Data Destruction Policy** defines how and when data should be destroyed within an organization. It ensures:

* Compliance with regulations like **GDPR**, **HIPAA**, or **SOX**.
* Protection against **data breaches** through improper disposal.
* Defined responsibilities, methods, timelines, and documentation procedures.

**▪ Data Destruction Techniques**

1. **Overwriting:**
   * Rewriting the data on a storage device with random characters multiple times.
   * Example: 3-pass or 7-pass overwrite (based on standards like DoD 5220.22-M).
2. **Degaussing:**
   * Exposing magnetic storage (like HDDs or tapes) to a strong magnetic field, rendering the data unrecoverable.
3. **Physical Destruction:**
   * Physically damaging the device (shredding, drilling, incineration).
   * Ensures total data elimination, even if software-based methods fail.
4. **Cryptographic Erasure:**
   * Deleting the encryption keys so the encrypted data becomes inaccessible.

**▪ Disk Wipe: Windows Diskpart Utility**

**Diskpart** is a command-line utility in Windows used to manage disks and partitions. It can be used to:

* **Clean** or **clean all** disks.
  + clean removes partition info.
  + clean all writes zeros across the entire disk (overwriting the data).
* Syntax:

**diskpart**

**list disk**

**select disk <number>**

**clean all**

**▪ Data Destruction Tools**

* **DBAN (Darik's Boot and Nuke):**  
  Open-source tool used to wipe entire hard drives.
* **Blancco Drive Eraser:**  
  Certified data erasure tool supporting various compliance standards.
* **CCleaner Drive Wiper:**  
  Tool for overwriting free space or entire drives on Windows.
* **Active@ KillDisk:**  
  Offers both software and hardware-based wiping methods.

**▪ Data Destruction Standards**

Several organizations have published standards for data destruction:

* **DoD 5220.22-M (U.S. Department of Defense):**  
  A standard for overwriting data multiple times.
* **NIST SP 800-88 (National Institute of Standards and Technology):**  
  Guidelines for media sanitization—includes clear, purge, and destroy.
* **ISO/IEC 27040:**  
  International standard covering storage security and data disposal.

▪ **Data Destruction Best Practices**

* Classify data and select appropriate destruction methods.
* Maintain logs and certificates of destruction for audits.
* Train employees on secure disposal methods.
* Ensure third-party destruction vendors follow compliance standards.
* Regularly review and update the data destruction policy.

**LO#8:** Data Loss Prevention (DLP) Concepts

**▪ What is Data Loss Prevention (DLP)?**

**Data Loss Prevention (DLP)** refers to a set of strategies and tools designed to prevent unauthorized access, leakage, or transfer of sensitive data—whether **in use**, **in motion**, or **at rest**.  
It helps organizations:

* Protect **intellectual property (IP)**
* Comply with **regulatory standards** (GDPR, HIPAA, etc.)
* Prevent **accidental or malicious data breaches**

**▪ Types of Data Loss Prevention (DLP) Solutions**

**Network DLP**

* Monitors and controls data being transmitted over the network (e.g., email, web traffic).
* Example: Blocking sensitive info from being emailed out.

**Endpoint DLP**

* Installed on user devices (laptops, desktops) to monitor local activities.
* Can block copying to USB drives, printing sensitive files, or screenshots.

**Storage DLP**

* Scans and protects data at rest on file servers, databases, or cloud storage.
* Identifies sensitive data stored insecurely.

**Cloud DLP**

* Protects data stored and processed in cloud environments like Google Workspace or Microsoft 365.

**▪ DLP Solution: Windows Information Protection (WIP)**

**Windows Information Protection (WIP)** is a Microsoft feature for managing and protecting data on Windows 10/11 devices.

**Key features:**

* Separates **personal and business data** on the same device.
* **Restricts data sharing** outside approved apps or containers.
* Prevents **corporate data leaks** via copy-paste, email, or cloud sync.

**Use Cases:**

* Protecting data on **BYOD (Bring Your Own Device)** systems.
* Enforcing **encryption and access control** on corporate apps.

**▪ DLP Solution: MyDLP**

**MyDLP** is an open-source Data Loss Prevention solution by Comodo.

**Features:**

* Monitors and filters **outgoing data** over email, web, and USB.
* Centralized management and reporting dashboard.
* Ideal for **small to mid-sized businesses** looking for affordable DLP.

**Limitations:**

* Limited support for cloud integrations compared to enterprise solutions.
* May require manual setup and fine-tuning.

**▪ Best Practices for a Successful DLP Implementation**

1. **Classify sensitive data** (e.g., PII, financial, health data).
2. Start with a **monitor-only** mode to understand data flow.
3. Set up **granular policies** based on user roles, data types, and behavior.
4. **Educate employees** on acceptable data handling practices.
5. Regularly **audit DLP logs** and refine rules.
6. Use **encryption, access controls, and endpoint hardening** alongside DLP.

**▪ DLP Solution Vendors**

1. Symantec DLP (Broadcom)
2. Forcepoint DLP
3. McAfee Total Protection for DLP
4. Digital Guardian
5. Trend Micro Integrated DLP
6. Microsoft Purview DLP
7. Cisco Cloudlock (Cloud DLP)
8. GTB Technologies DLP