Domain 5: Security Operations

▼ Data Governance

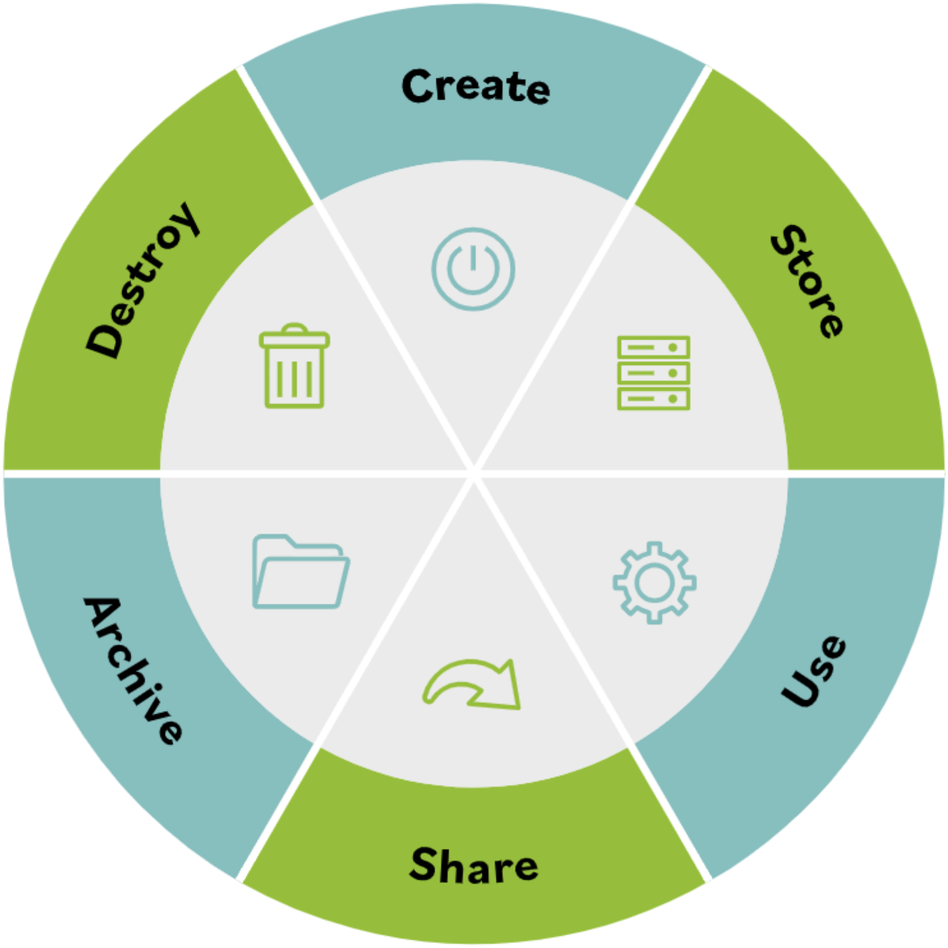
1. Data Lifecycle Overview

- **Definition:** The data lifecycle describes the sequence of stages data goes through from its creation to its final destruction or deletion.
- **Purpose:** Understanding and managing this lifecycle is crucial for ensuring data security, regulatory compliance, and efficient data management.

2. Core Phases of the Data Security Lifecycle

Phase	Description	Example
Create	Data is generated or acquired. This is often tacit knowledge or raw data.	A user fills out a registration form, generating new user data.
Store	Data is recorded and stored, making it explicit and accessible.	Saving customer information in a database.
Use	Data is accessed, processed, or modified for business or operational purposes.	Analyzing sales data to generate reports.
Share	Data is distributed or made accessible to others, either internally or externally.	Sending a report to a business partner or uploading files to a shared drive.
Archive	Data is moved to long-term storage when it is not actively needed but must be retained for compliance or future reference.	Moving old financial records to an archive server.
Destroy	Data is permanently deleted or physically destroyed when it is no longer needed.	Shredding paper files or wiping hard drives after retention period ends.

Note: Some models include additional steps like "Capture," "Analyze," or "Publish," but the above six are most commonly referenced in security frameworks.



3. Data States

- **In Use:** Data actively being processed or accessed (e.g., editing a document).
- **At Rest:** Data stored on physical or cloud storage and not actively used (e.g., files on a server).
- **In Motion:** Data being transmitted between locations or users (e.g., sending an email).

Understanding these states helps determine appropriate security controls for each phase.

4. Data Handling Practices

A. Classification and Labeling

- **Classification:** Determining the sensitivity and value of data to the organization, often based on confidentiality, integrity, and availability requirements. This informs how data should be handled and protected.
    - *Example classifications:*
        - **Highly Restricted:** Loss could threaten the organization’s existence.
        - **Moderately Restricted:** Loss could cause financial or operational harm.
        - **Low Sensitivity:** Loss would cause minor disruption.
        - **Unrestricted/Public:** No harm from disclosure.
- **Labeling:** Assigning visible tags or markers to data indicating its classification, guiding users on appropriate handling and access controls.

B. Retention

- **Definition:** Policies that define how long data should be kept, based on legal, regulatory, or business requirements.
- **Examples:**
    - Medical records may need to be kept for 10 years (HIPAA).
    - OSHA may require certain records to be kept for over 30 years.
- **Best Practice:** Only retain data as long as needed; excessive retention increases risk and storage costs.

C. Destruction

- **Definition:** Securely erasing or physically destroying data when it is no longer needed.

- **Methods:**
  - **Clearing:** Overwriting storage with random data.
  - **Purging:** More thorough than clearing, reducing chances of data recovery.
  - **Physical Destruction:** Shredding, burning, or degaussing media.
- **Remanence:** Residual data left after deletion; must be addressed to prevent unauthorized recovery.

5. Regulatory and Compliance Considerations

- **Multiple Regulations:** Data may be subject to overlapping laws (e.g., HIPAA, OSHA, PCI DSS, GDPR), each with specific requirements for retention, protection, and destruction.
- **Jurisdictional Challenges:** Data handled in multiple regions must comply with all applicable local, national, and international regulations.
- **Audit and Enforcement:** Failure to comply can result in audits, fines, or legal action.

6. Common Security Policies

Policy Type	Purpose	Key Elements
Data Handling Policy	Defines how data is used, shared, and protected.	Classification, access controls, legal restrictions.
Password Policy	Ensures secure authentication.	Complexity, change frequency, enforcement responsibility.
Acceptable Use Policy (AUP)	Sets rules for use of organizational IT assets.	Usage guidelines, prohibited actions, BYOD considerations.
BYOD Policy	Governs use of personal devices for work.	Security requirements, audit provisions, user agreements.
Privacy Policy	Outlines handling of personal and sensitive information.	Definitions, handling procedures, legal references (GDPR, HIPAA).

Policy Type	Purpose	Key Elements
Change Management Policy	Manages changes to IT systems to prevent vulnerabilities.	Approval processes, documentation, rollback procedures.

- **Enforcement:** Policies must include consequences for noncompliance, from warnings to termination, and require employee acknowledgment during onboarding.

7. Supporting Procedures

- **Procedures:** Step-by-step instructions for implementing policies (e.g., how to classify data, securely destroy media).
- **Alignment:** Procedures must align with organizational risk tolerance and regulatory requirements.
- **Customization:** Policies and procedures should be tailored to the organization’s needs, data types, and regulatory landscape.

8. Key Concepts Explained with Examples

- **Data Classification Example:** A bank classifies customer account data as "Highly Restricted" and public marketing material as "Unrestricted."
- **Retention Example:** A hospital keeps patient records for a legally mandated period, then destroys them securely.
- **Destruction Example:** A company uses degaussing equipment to erase hard drives before disposal, ensuring no data can be recovered.
- **BYOD Challenge Example:** An employee’s personal laptop used for work must meet security standards and may be subject to audit if a breach occurs.

Summary:

Effective data handling requires understanding the data lifecycle, classifying and labeling data appropriately, applying retention and destruction policies, and ensuring compliance with all relevant regulations. Security policies and supporting procedures must be clear, enforceable, and tailored to organizational needs, with regular training and enforcement to maintain compliance and protect data throughout its lifecycle.

▼ Change Management

Concepts Covered:

- Logging and Monitoring Security Events
- Configuration Management Overview
- The Risks of Change
- Change Management Components
- Change Management Components in the Workplace

Logging and Monitoring Security Events

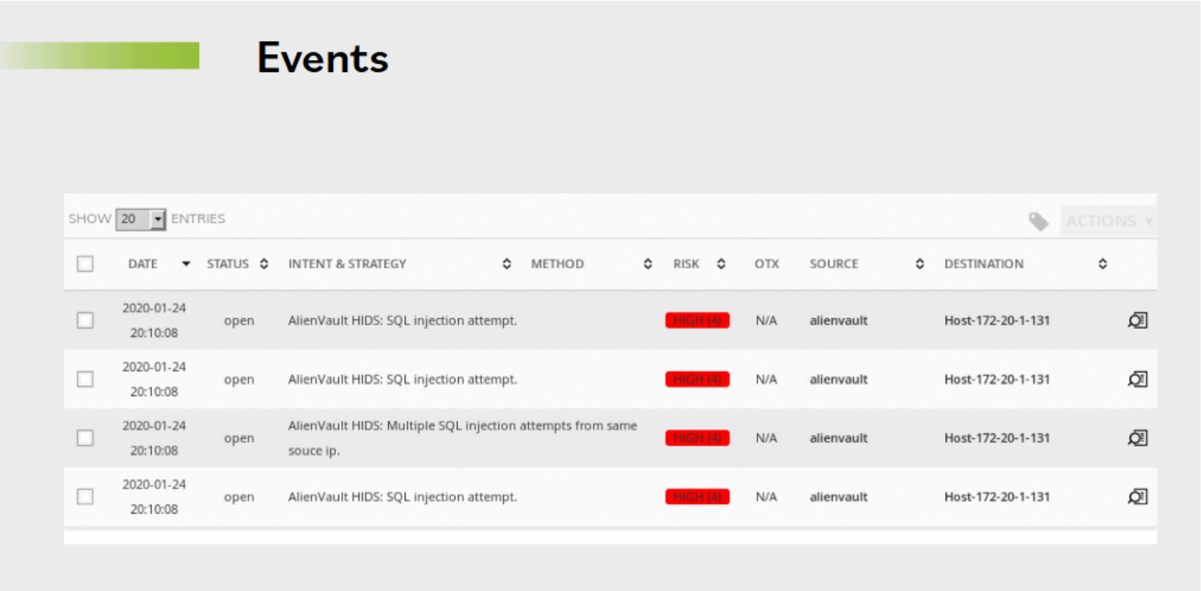
1. Logging

- **Definition:** Logging is the process of recording events and signals generated by actions within a system.
- **Purpose:** It provides a record of activities, which is crucial for accountability, troubleshooting, and security.
- **Example:** When a user logs into a system, the event (user login) is recorded in a log file.

2. Events

- **Definition:** Any action within a system that causes a measurable or observable change in system elements or resources.
- **Examples:**
  - User login/logout
  - File access or modification

- System configuration changes

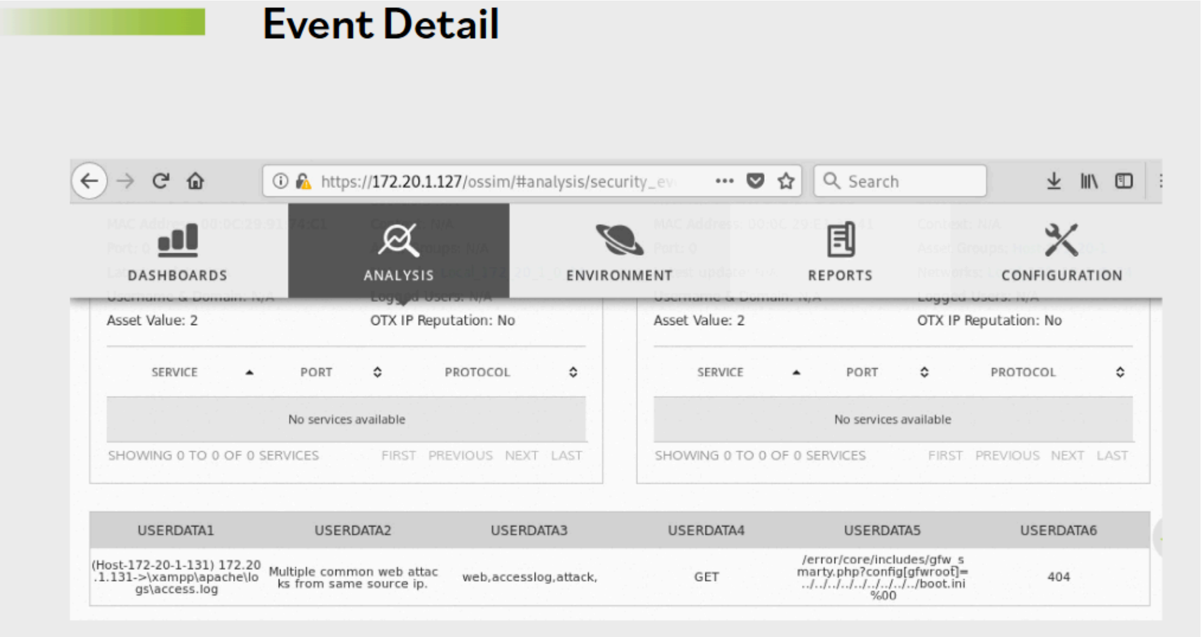


Events

SHOW 20 ENTRIES

ACTIONS

□	DATE	STATUS	INTENT & STRATEGY	METHOD	RISK	OTX	SOURCE	DESTINATION
□	2020-01-24 20:10:08	open	AlienVault HIDS: SQL injection attempt.		HIGH (4)	N/A	alienvault	Host-172-20-1-131
□	2020-01-24 20:10:08	open	AlienVault HIDS: SQL injection attempt.		HIGH (4)	N/A	alienvault	Host-172-20-1-131
□	2020-01-24 20:10:08	open	AlienVault HIDS: Multiple SQL injection attempts from same souce ip.		HIGH (4)	N/A	alienvault	Host-172-20-1-131
□	2020-01-24 20:10:08	open	AlienVault HIDS: SQL injection attempt.		HIGH (4)	N/A	alienvault	Host-172-20-1-131



Event Detail

https://172.20.1.127/ossim/#analysis/security_ev

Asset Value: 2

OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES

FIRST PREVIOUS NEXT LAST

Asset Value: 2

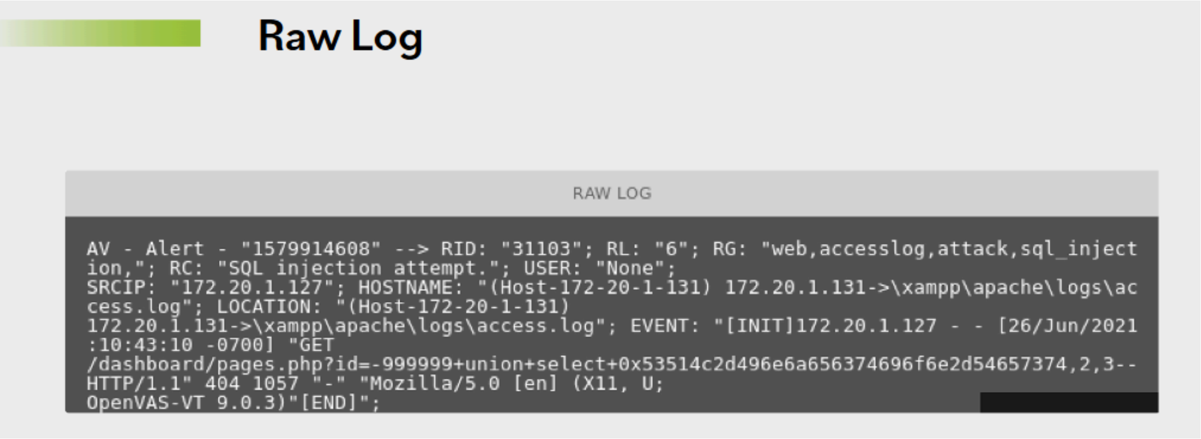
OTX IP Reputation: No

SERVICE	PORT	PROTOCOL
No services available		

SHOWING 0 TO 0 OF 0 SERVICES

FIRST PREVIOUS NEXT LAST

USERDATA1	USERDATA2	USERDATA3	USERDATA4	USERDATA5	USERDATA6
(Host-172-20-1-131) 172.20.1.131->\xampp\apache\logs\access.log	Multiple common web attacks from same source ip.	web,accesslog,attack,	GET	/error/core/includes/gfw_smarty.php?config[gfwroot]=../../../../../../../../../boot.ini%00	404



Raw Log

```
AV - Alert - "1579914608" --> RID: "31103"; RL: "6"; RG: "web,accesslog,attack,sql_inject
ion,"; RC: "SQL injection attempt."; USER: "None";
SRCIP: "172.20.1.127"; HOSTNAME: "(Host-172-20-1-131) 172.20.1.131->\xampp\apache\logs\ac
cess.log"; LOCATION: "(Host-172-20-1-131)
172.20.1.131->\xampp\apache\logs\access.log"; EVENT: "[INIT]172.20.1.127 - - [26/Jun/2021
:10:43:10 -0700] "GET
/dashboard/pages.php?id=-999999+union+select+0x53514c2d496e6a656374696f6e2d54657374,2,3--
HTTP/1.1" 404 1057 "-" "Mozilla/5.0 [en] (X11, U;
OpenVAS-VT 9.0.3)"[END]";
```

3. Information to Log

- **User IDs:** Identifies who performed the action.
- **System Activities:** What actions were performed.
- **Dates/Times:** When the actions occurred.
- **Device and Location Identity:** Where the action originated.
- **Access Attempts:** Both successful and failed attempts to access resources.
- **Configuration Changes:** Any changes to system settings or protections.

4. Importance of Logging

- **Computational Cost:** Logging uses system resources but is essential for security and accountability.
- **Best Practices:** Properly designed logging environments and regular log reviews are recommended.
- **Frameworks:** Major security frameworks require robust logging practices.

5. Benefits of Logging and Monitoring

- **System Health:** Helps identify inefficient or failing systems.
- **Security:** Detects compromises and provides a record of system use.

- **Correlation:** Allows analysis of activities across multiple systems to understand relationships between events.

6. Log Reviews

- **Purpose:** Identify security incidents, policy violations, fraud, and operational issues.
- **Support:** Useful for audits, forensic analysis, and maintaining security baselines.
- **Historical Analysis:** Reviewing old logs can reveal past exploitation of vulnerabilities.

7. Log Management Infrastructure

- **Components:** Systems and processes for collecting, storing, and protecting logs.
- **Integrity and Confidentiality:** Prevents unauthorized changes or deletion and protects sensitive information in logs.

8. Controls and Risks

- **Protection:** Prevent unauthorized changes to logs.
- **Operational Issues:** Risks include log alteration, deletion, or storage capacity issues.
- **Retention Policies:** Logs must be kept according to laws and regulations.
- **Attackers:** May attempt to delete or alter logs to hide evidence.
- **Sensitive Data:** Logs often contain valuable information and must be protected.

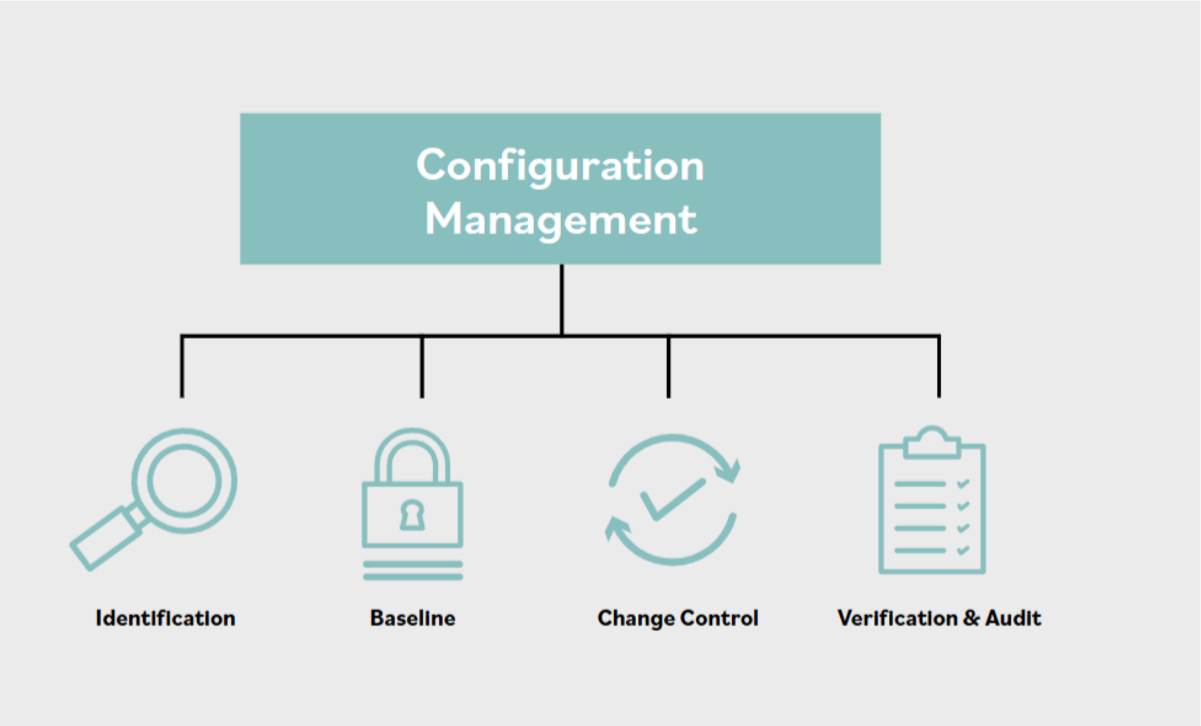
Configuration Management Overview

1. Definition

- **Configuration Management:** A discipline ensuring only authorized and validated changes are made to systems.

2. Key Components

- **Identification:** Cataloging all system components, interfaces, and documentation.
  - *Example:* Listing all servers, software versions, and network devices.
- **Baselines:** Establishing a minimum level of security and functionality as a reference point.
  - *Example:* A secure configuration for all company laptops.
- **Change Control:** Formal process for requesting, reviewing, approving, and implementing changes.
  - *Example:* Submitting a request to update antivirus software, which is reviewed and approved before deployment.
- **Verification and Audit:** Testing and validating that changes do not break the system.
  - *Example:* Regression testing after a software update to ensure existing features still work.
- **Inventory:** Keeping an up-to-date registry of all information assets.
  - *Example:* Asset management tools that track all devices and software in use.



3. Baselines in Detail

- **Definition:** The complete inventory of all system components and their configurations.
- **Purpose:** Used for comparison and to ensure a minimum protection level.
- **Classification:** Baselines can be tailored for different asset classifications (high, medium, low value).

4. Updates

- **Definition:** Repairs, maintenance, or improvements to systems.
- **Testing:** Must be acceptance and regression tested to ensure they work and don’t introduce new problems.
- **Security Assessment:** Ongoing evaluation to ensure updates maintain system security.

5. Patches

- **Definition:** Updates to fix vulnerabilities or improve functionality.

- **Patch Management:** Process of acquiring, testing, and deploying patches.
    - *Example:* Microsoft releases a security patch for Windows, which must be tested and applied.
- **Risks:**
    - Flawed patches can break systems.
    - Testing environments may not match production.
    - Unattended patching can cause unexpected outages.
- **Rollback:** Ability to revert to a previous state if a patch causes issues.
    - *Example:* Uninstalling a problematic update and restoring the system to its previous configuration.

The Risks of Change

- **Change Management:** Must have a robust process and test changes in a non-production environment.
- **Rollback Plan:** Essential to restore systems to a known good state if changes fail.
- **Testing Challenges:** Many organizations lack separate test environments, increasing reliance on vendor testing.
- **Criticality:** Rollback plans are especially important where full testing isn't possible.

Change Management Components

1. Documentation

- **Process:** Every change is documented from request to implementation.
- **Logs:** Each step produces records for accountability.

2. Approval

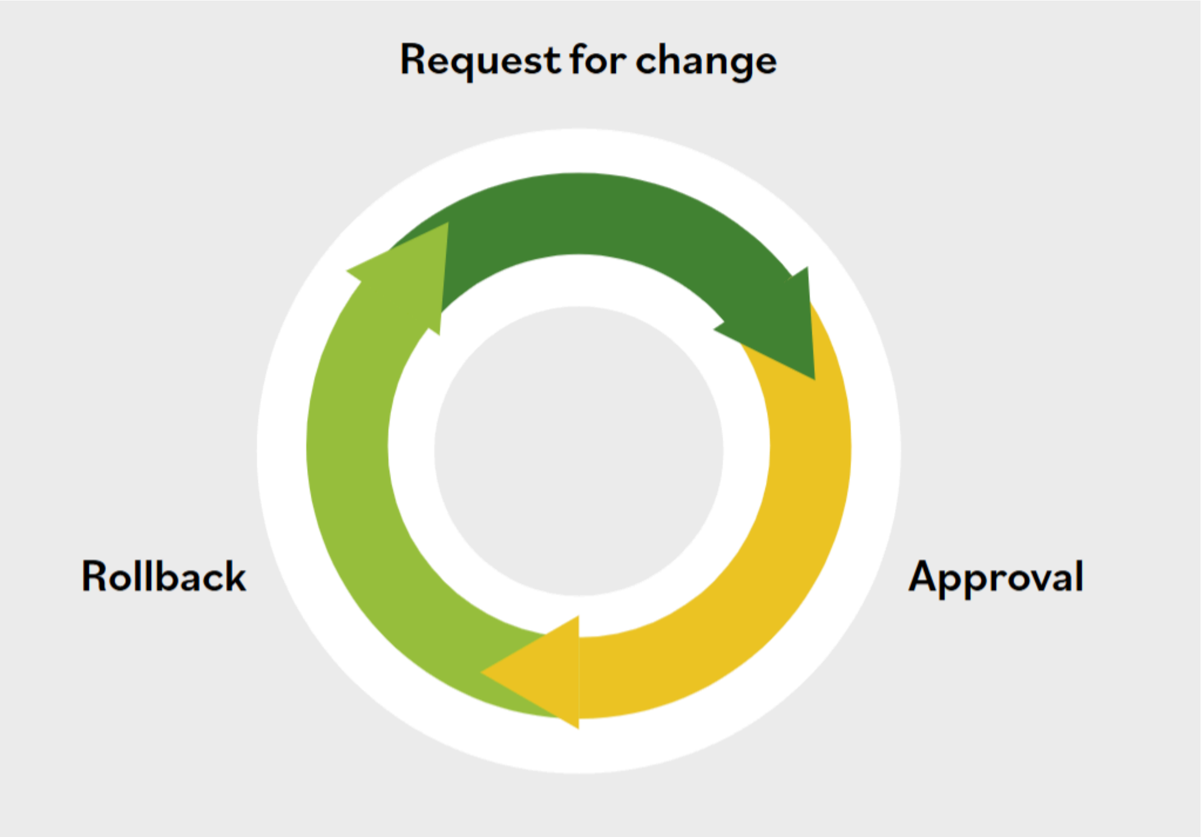
- **Evaluation:** Requests for change (RFCs) are reviewed for completeness and risk.
- **Authorization:** Changes are assigned to appropriate processes and approved or rejected.
- **Stakeholders:** Involvement from relevant parties ensures thorough review.

3. Rollback

- **Procedures:** Plans for reverting changes if they fail.
- **Authority:** Defined in advance, specifying who can initiate a rollback.

Change Management in the Workplace

- **Continuous Cycle:** Change management is ongoing, with continuous monitoring.
- **Approvals:** All changes must be reviewed and approved.
- **Rollback Preparedness:** Must be able to revert to previous systems if changes fail.
- **Responsibility:** Often led by information security, IT, or risk management departments.
- **Collaboration:** Involves end users, IT, development, security, and management to ensure changes are tested, approved, and communicated.



Summary Table

Concept	Description & Example
Logging	Recording system events (e.g., user login) for accountability and security
Events	Actions causing system changes (e.g., file access, configuration change)
Log Management	Collecting, storing, and protecting logs; ensuring integrity and confidentiality
Configuration Management	Ensuring only authorized system changes; includes identification, baselines, change control, and audit
Baselines	Reference configurations for security and functionality (e.g., secure laptop setup)
Updates & Patches	System improvements and vulnerability fixes; must be tested and managed
Rollback	Reverting to previous system state if changes fail

Concept	Description & Example
Change Management	Formal process for requesting, approving, implementing, and documenting changes
Continuous Monitoring	Ongoing review of systems and changes to maintain security and functionality

Key Takeaways

- **Logging and monitoring** are essential for security, accountability, and system health.
- **Configuration management** ensures systems are only changed in controlled, authorized ways.
- **Baselines** provide a reference for security and functionality.
- **Change management** is a structured, ongoing process requiring documentation, approval, testing, and rollback planning.
- **Collaboration and oversight** are critical to successful change management and maintaining organizational security.

If you need further breakdowns or specific examples for any concept, let me know!

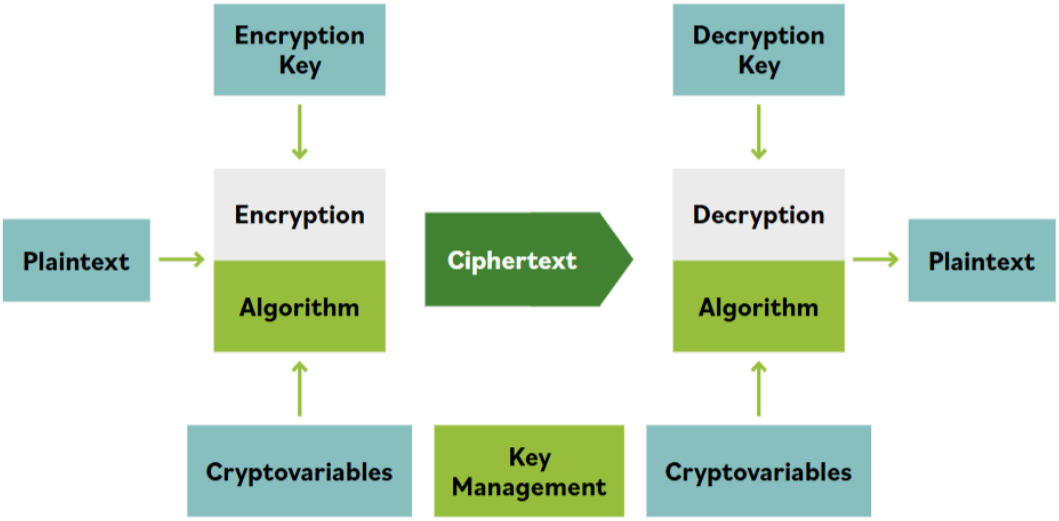
▼ Hashing & Encryption

Concepts Covered:

- Encryption Overview
- Encryption Deep Dive
- Hashing
- Hashing Deep Dive
- Asymmetric Encryption
- Symmetric Encryption

1. Encryption Overview

- **Definition & Importance**
    - **Encryption** is a core part of cryptography, used everywhere in the digital world to protect personal and business transactions, verify authenticity of software, and secure digital contracts.
    - **Example:** When you send a confidential email or make an online transaction, encryption ensures only the intended recipient can read the information.
- **Purpose**
    - **Protects information** by making it unreadable (ciphertext) to unauthorized users, only allowing those with the decryption key to access the original message (plaintext).
- **Plaintext**
    - **Plaintext** is the original, unencrypted data.
    - **Examples:** Human-readable text, images, audio, video files, database records, or any digital data.
    - *Note:* Not all plaintext is human-readable (e.g., binary files).
- **Ciphertext**
    - **Ciphertext** is the unintelligible, encrypted version of plaintext.
- **Encryption System**
    - Comprises hardware, software, algorithms, control parameters, and operational methods to provide encryption services.



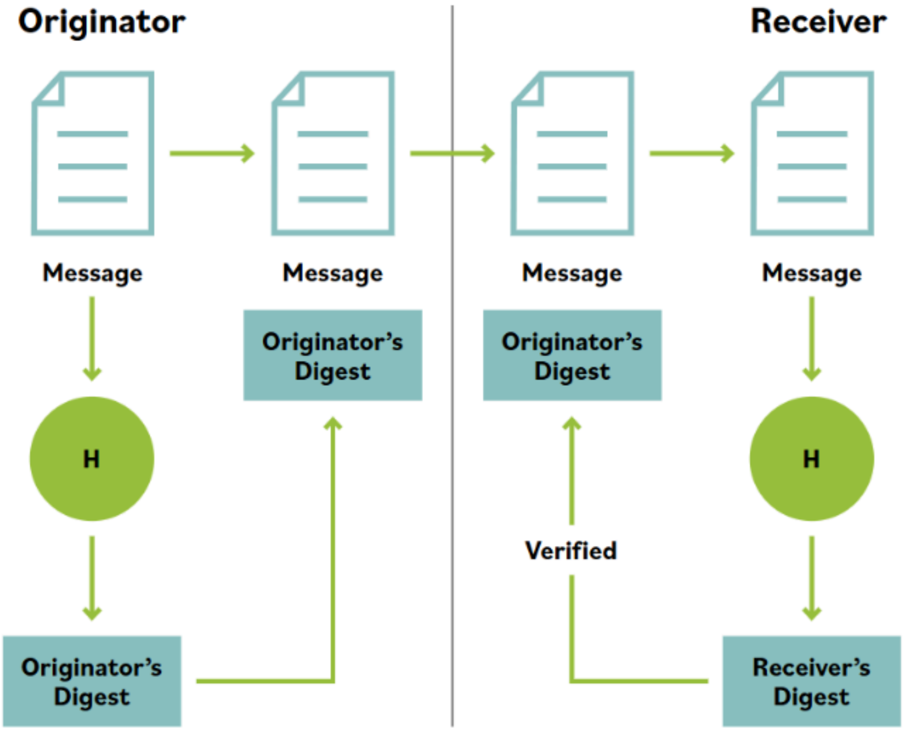
2. Core Cryptographic Services

Service	Description	Example
Confidentiality	Keeps information secret from unauthorized users.	Only the recipient with the correct key can read an encrypted message.
Integrity	Ensures data has not been altered in transit (accidentally or maliciously).	Hash functions and digital signatures verify that a message hasn't changed.
Non-repudiation	Prevents sender from denying authorship or sending of a message.	Digitally signed contracts exchanged via email cannot be repudiated later by the sender.

3. Hashing

- **Definition**
  - **Hashing** transforms data of any size into a fixed-length string called a **hash value** or **digest** using a hash function (algorithm).
- **Properties of Cryptographic Hash Functions**
  - **Easy to compute:** Fast to generate a hash for any input.

  - **Nonreversible:** Impossible to recover original input from the hash.
  - **Content integrity:** Any change in input yields a different hash.
  - **Unique:** Two different inputs should not produce the same hash (collision resistance).
  - **Deterministic:** Same input always results in the same hash.
- **Applications**
  - **Integrity verification:** Ensures files or messages haven't been tampered with.
  - **Authentication:** Used in digital signatures and password storage.
  - **Checksums:** Detect accidental data corruption.
  - **Fingerprinting:** Identify duplicate files or data.
- **Example:**
  - If you hash the phrase "The red fox jumps over the blue dog" and then change a single letter, the resulting hash will be completely different, indicating a change in the input.
- **Limitations**
  - Hashing alone does not prevent malicious tampering, as an attacker could alter both the message and hash. Secure systems combine hashing with digital signatures or encryption.



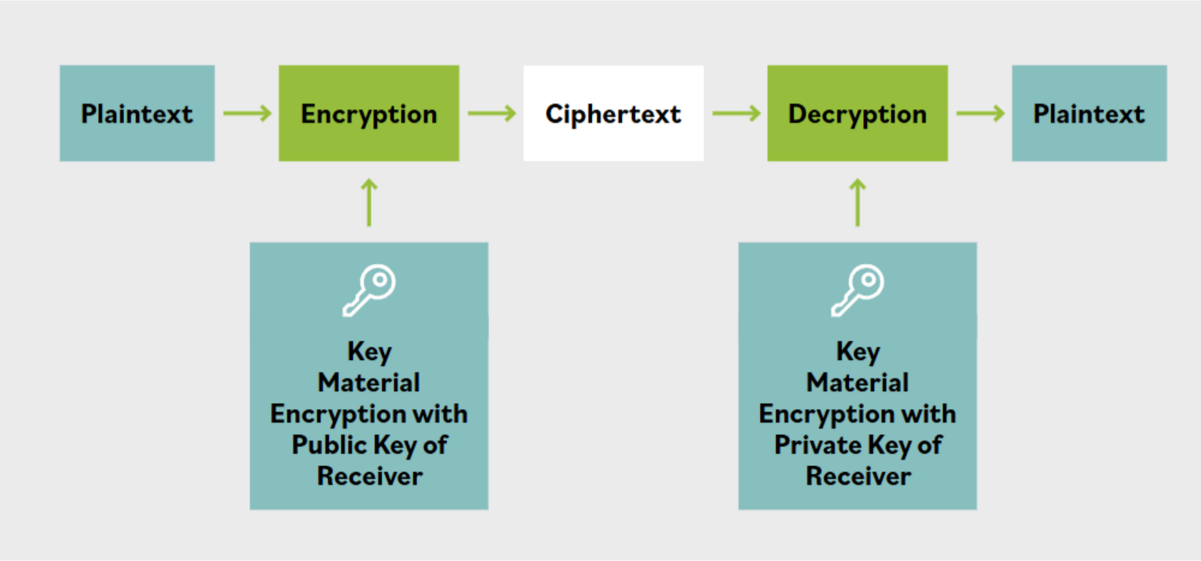
4. Hashing in Practice

- **Message Digest Workflow**
  1. Sender hashes the original message.
  2. Sender sends both message and hash to recipient.
  3. Recipient hashes received message and compares to the received hash.
  4. If hashes match, message integrity is confirmed.
- **Real-world Example:**
  - When downloading software, the publisher provides a hash of the original file. After download, you hash your copy and compare; a mismatch could indicate tampering.
- **Case Study:**

  - At the University of Florida, compromised software CDs were detected because the digests did not match the originals.

5. Asymmetric Encryption (Public Key Cryptography)

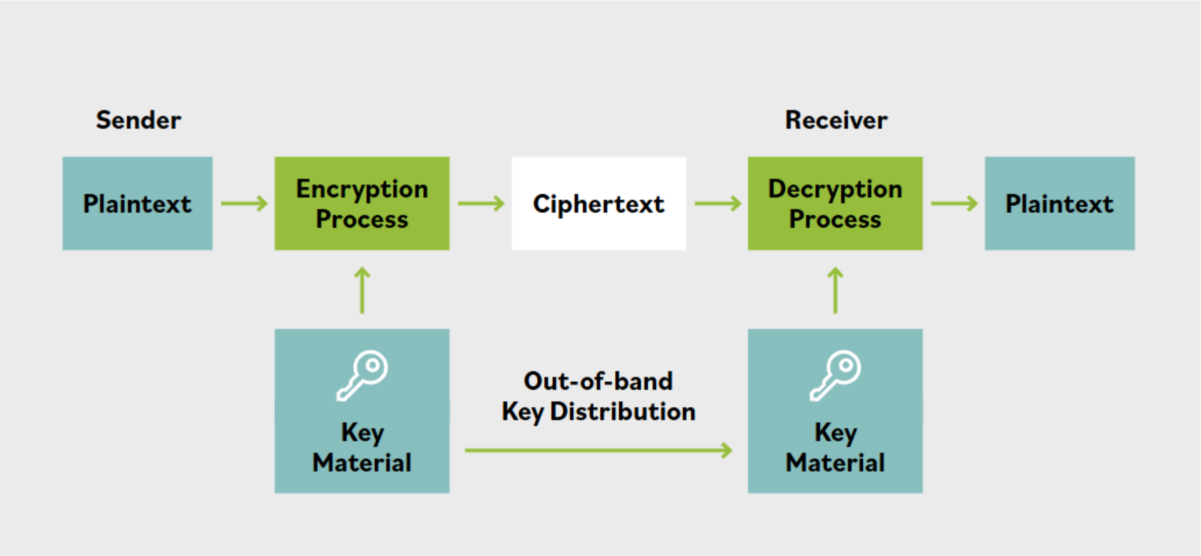
- **Definition**
  - Uses a **key pair**: one **public key** (shared openly) and one **private key** (kept secret). Data encrypted with one key can only be decrypted with the other.
- **How It Works**
  - To send a confidential message: Encrypt using recipient's public key; only recipient's private key can decrypt.
  - For digital signatures: Sender signs with their private key; anyone can verify using sender's public key.
- **Advantages**
  - **Solves key distribution problem:** No need to share secret keys in advance.
  - **Scalability:** Each user needs only one key pair, reducing the total number of keys compared to symmetric encryption.
  - **Non-repudiation and integrity:** Supports digital signatures.
- **Disadvantages**
  - **Slower** than symmetric encryption due to complex mathematical operations.
  - Not suitable for encrypting large amounts of data; often used to encrypt symmetric keys instead.
- **Example:**
  - Secure email (e.g., PGP): You encrypt a message with the recipient's public key; only they can decrypt it.


Plaintext
Encryption
Ciphertext
Decryption
Plaintext
Key Material Encryption with Public Key of Receiver
Key Material Encryption with Private Key of Receiver

6. Symmetric Encryption

- **Definition**
  - Uses the **same key** for both encryption and decryption.
- **How It Works**
  - Both sender and recipient must possess the same secret key.
  - **Example:** Substitution cipher, where each letter in the plaintext is replaced by another letter using a secret pattern.
- **Challenges**
  - **Key distribution:** Securely sharing the key is difficult, especially over insecure channels.
  - **Scalability:** Number of keys grows rapidly as more users are added (for n users, need n(n-1)/2 keys).
  - **Key compromise:** Anyone with the key can decrypt and alter messages.
- **Primary Uses**
  - **Bulk data encryption:** Backups, hard drives, portable media.
  - **Streaming data:** Audio/video, gaming.

    - **Communication channels:** IPsec, TLS.
- **Other Names**
    - Same key, single key, shared key, secret key, session key.



7. Key Management

- **Importance**
    - Secure storage and management of encryption keys is critical; poor key management can undermine all cryptographic protections.
    - **Example:** In businesses, keys may be stored on external servers and protected with additional security measures (e.g., hashing).
- **Asymmetric vs. Symmetric Key Management**
    - **Asymmetric:** Easier to manage, as only key pairs are needed.
    - **Symmetric:** Key management becomes complex as the number of users increases.

8. Summary Table: Symmetric vs. Asymmetric Encryption

Feature	Symmetric Encryption	Asymmetric Encryption
Keys Used	Same key for encryption/decryption	Public and private key pair
Speed	Fast	Slow
Key Distribution	Difficult (must be shared securely)	Easier (public key can be shared)
Scalability	Poor (many keys needed)	Good (one pair per user)
Use Case	Bulk data, streaming	Key exchange, digital signatures
Example	AES, DES	RSA, ECC

Key Takeaways:

- **Encryption** and **hashing** are fundamental to digital security, each serving different purposes: encryption for confidentiality, hashing for integrity.
- **Symmetric encryption** is efficient for large data but suffers from key management issues.
- **Asymmetric encryption** solves key distribution and scalability problems but is computationally intensive.
- **Hashing** is vital for verifying data integrity and is widely used in authentication and checksums.
- Effective **key management** underpins the security of all cryptographic systems.

▼ Password Security Awareness

Concepts Covered:

- Data Security Event Example
- Event Logging Best Practices
- How Passwords Work
- Security Awareness Training
- Security Awareness Training Example

- Password Advice and Examples
- Password Protection
- Best Practices of Security Awareness Training
- Phishing
- Social Engineering

---

1. Data Security Events and Incidents

- **Data Security Event:**

  An observable occurrence in a network or IT system that may indicate a security policy violation or failure of a safeguard.

  - *Example:* A user repeatedly failing to log in, or a system log showing an attempt to access a secure file.
  - *Explanation:* Not every event is harmful; some may be routine (e.g., a user logging in), while others may require investigation (e.g., multiple failed logins could indicate a brute-force attack).

- **Security Incident:**

  A specific event (or sequence of events) that results in an actual or suspected breach, such as unauthorized access, use, modification, or disclosure of sensitive data.

  - *Example:* Help Desk finds a virus on a system, or ransomware encrypts files and demands payment.
  - *Difference:*

Security Event	Security Incident
Any observable activity (routine or suspicious)	Activity that causes harm or policy violation
E.g., login attempt	E.g., data breach, malware infection

2. Event Logging Best Practices

- **Purpose:**

  To monitor and analyze both incoming (ingress) and outgoing (egress) network traffic for security threats.

- **Ingress Monitoring:**

  Surveillance of all inbound traffic and access attempts.

  - *Tools:*
    - Firewalls: Block/allow traffic based on rules.
    - IDS/IPS (Intrusion Detection/Prevention Systems): Detect and prevent unauthorized access.
    - Gateways: Control data flow between networks.
    - SIEM (Security Information and Event Management): Aggregates and analyzes logs for suspicious activity.
    - Remote Authentication Servers: Manage user authentication.
    - Anti-malware: Detects and blocks malicious software.

- **Egress Monitoring:**

  Controls and inspects data leaving the organization to prevent data leaks.

  - *Also known as:* Data Loss Prevention (DLP).
  - *Monitored Channels:*
    - Email (including attachments)
    - Web postings
    - Copying to portable media (USB drives)
    - Application APIs
    - File transfers (FTP)

3. Password Management and Security

- **How Passwords Work:**

  Passwords are typically stored as *hashes*—a cryptographic representation that cannot be reversed to reveal the original password.

  - *Hash Example:* "password123" hashed with SHA-256 produces a unique string of characters.
  - *Security Note:* If attackers obtain the hashed password file and know the algorithm, they can attempt to "brute force" or guess the password by generating hashes from possible passwords and comparing them.

- **Password Strength:**
  - *Short/simple passwords* (e.g., 10 digits) can be cracked in seconds.
  - *Long/complex passwords* (e.g., 16 characters with upper/lowercase and symbols) can take thousands of years to crack.
  - *Example:*

Password Type	Estimated Time to Crack
10-digit numeric	5 seconds
8-character mixed	35 days
16-character, upper/lower/special	152,000 years

- **Password Protection:**
  - Use different passwords for different systems.
  - Avoid sharing or writing down passwords.
  - Use password managers, but secure them with strong master passwords.

4. Security Awareness Training

- **Purpose:**

  To ensure all personnel understand their security responsibilities and can recognize and avoid risky behaviors.

- **Types of Learning Activities:**

- **Education:** Broad understanding of concepts (e.g., how fire suppression systems interact in a server room).
- **Training:** Skill-building for specific tasks (e.g., what to do during a fire alarm).
- **Awareness:** Ongoing reminders and alerts (e.g., signage, floor markings, or simulated phishing emails).

- **Example Application:**
    - *Fire Safety:*
        - Education: How fire systems work.
        - Training: Steps to take during an alarm.
        - Awareness: Signs and reminders.
    - *Anti-Phishing:*
        - Education: How phishing works.
        - Training: Practice identifying phishing emails.
        - Awareness: Alerts about new phishing tactics.

5. Best Practices for Security Awareness

- Regularly communicate about current threats.
- Use positive reinforcement and feedback for good security behavior.
- Simulate attacks (e.g., phishing emails) and reward correct responses.
- Make training engaging and non-punitive unless necessary.

6. Phishing and Social Engineering

- **Phishing:**

  Deceptive attempts to trick users into revealing sensitive information via email, phone, or other channels.

    - *Whaling:* Phishing targeting high-level executives or wealthy individuals, often to authorize large transfers.

- **Social Engineering:**

  Manipulating people into divulging confidential information or performing actions that compromise security.

  - *Common Tactics:*
    - **Vishing (Voice Phishing):** Fake phone calls or IVR systems imitating banks.
    - **Quid Pro Quo:** Offering something in exchange for credentials.
    - **Pretexting:** Impersonating authority figures to gain trust.
    - **Tailgating:** Following authorized personnel into restricted areas.
  - *Defense:*
    - Education, training, and awareness are key to helping employees recognize and resist social engineering attacks.

7. Data Security Concepts

- **Data Security:**

  Protecting digital data from unauthorized access, corruption, or loss throughout its lifecycle.

  - *Techniques:*
    - **Encryption:** Converts data to unreadable form; only those with the key can access.
    - **Data Masking:** Hides sensitive data by replacing it with fake but realistic values.
    - **Tokenization:** Substitutes sensitive data with non-sensitive tokens.
    - **Data Erasure:** Permanently deletes data so it cannot be recovered.
    - **Data Resiliency:** Ensures data can be restored after incidents (e.g., backups, disaster recovery).
    - **Access Control:** Restricts data access to authorized users only.
    - **Authentication:** Verifies user identities before granting access.

Summary Table: Key Concepts and Examples

Concept	Definition/Explanation	Example
Security Event	Observable occurrence, may or may not be harmful	Failed login attempt
Security Incident	Event causing harm or violating policy	Data breach, malware infection
Ingress Monitoring	Watching incoming traffic for threats	Firewall blocking suspicious IP
Egress Monitoring (DLP)	Monitoring outbound data to prevent leaks	Blocking sensitive data in email
Password Hashing	Storing passwords as irreversible digests	SHA-256 hash of "password123"
Strong Passwords	Longer, complex passwords are harder to crack	"P@ssw0rd!2025"
Security Awareness Training	Educating and training users on security	Simulated phishing emails
Phishing	Deceptive attempts to steal information	Fake bank email asking for credentials
Social Engineering	Manipulating people to gain access	Impersonating IT support
Data Security	Protecting data from unauthorized access/loss	Encryption, access controls