

Domain 5: Security Operations

▼ Data Governance

Concepts Covered:

- Data Handling
 - Data Handling Deep Dive
 - Data Handling Practices
 - Common Security Policies
 - Common Security Policies Deeper Dive
 - Supporting Security Policies with Procedures
-

1. Data Handling

Data handling encompasses the full *life cycle* of data, following it from the moment of creation to its eventual destruction. The typical stages are:

- **Create:** Data is generated, often starting as tacit (unwritten) knowledge before being recorded and formalized.
 - *Example:* An employee brainstorming a process improvement, then writing a report.
- **Store:** Data is saved using various storage methods (physical or digital), making it explicit and retrievable.
 - *Example:* Saving documents on a secure company server.
- **Use:** Data is accessed, reviewed, and possibly modified or updated.
 - *Example:* A manager editing a spreadsheet with financial data.
- **Share:** Data is distributed internally or externally, either by copying or moving it to new locations.
 - *Example:* Sending files via encrypted email to authorized partners.

- **Archive:** Data not needed for immediate use is moved to long-term storage but may be needed in the future.
 - *Example:* Moving completed project files to a backup server.
- **Destroy:** Data is permanently deleted or physically demolished when no longer needed.
 - *Example:* Shredding hard copies, or erasing and degaussing hard drives.

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



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).

Alignment with Roles: The data security life cycle helps define responsibilities at each stage for individuals or organizations.

2. Data Handling Deep Dive

Recognizing Assets:

Identify valuable assets needing protection based on their value, as assessed by the data owner. Determine associated risks and vulnerabilities (likelihood of compromise).

Regulatory Requirements:

Data handling is governed by multiple standards and regulations:

- **OSHA**: Medical records for workplace injury may need to be retained for over 30 years in the US.
- **HIPAA**: Specific periods for healthcare records.
- **PCI DSS**: Rules for handling payment card information.
- **GDPR**: Strict data handling and privacy rules in the EU.

Best Practices:

- Data must be protected at every life cycle stage.
- **Classification and labeling**: Assign sensitivity levels and restrict access accordingly.
- **Retention Policies**: Define how long data is stored and where, based on internal and regulatory requirements.

- **Defensible Destruction:** Follow sound methods for secure deletion, both physically (e.g., shredding) and digitally (e.g., degaussing, overwriting).

Destruction Techniques:

- Simply emptying a virtual trash bin is not enough—data remnants ("remanence") may persist.
- Use specialized methods like overwriting (clearing), purging, or physical destruction.

Multiple Jurisdictions:

Be aware that data may be subject to overlapping legal requirements based on its type and location.

3. Data Handling Practices

A. Classification

Classifying data answers, "How damaging would loss or misuse of this data be?" Classification is typically based on confidentiality, integrity, and availability impacts:

- **Highly Restricted:** Loss could threaten the organization's existence.
- **Moderately Restricted:** Loss could mean competitive or financial harm.
- **Low Sensitivity:** Loss causes only minor disruptions.
- **Unrestricted/Public:** No harm from public disclosure.

Benefits: Uniform application of controls and efficient response planning.

B. Labeling

Labels communicate sensitivity (e.g., "Confidential" or "Public"). Labels guide how data is handled and protected, and access is controlled accordingly.

- Too many categories are confusing; two to three are usually effective.

C. Retention

Retention ensures data is kept only as long as required by law, regulation, or business need.

- Maintain inventories detailing what data exists, where, and for how long it's kept.
- Periodic reviews help remove unneeded data, reducing storage costs and exposure risk.

A common error: Applying the longest retention period to all data types increases costs and risks.

D. Destruction

Ensure that data at the end of its retention period is destroyed securely:

- **Clearing:** Overwriting storage media multiple times.
- **Purging:** Advanced erasure or degaussing, reducing the chance of data remanence.
- **Physical Destruction:** Shredding or incinerating storage devices.

Security needs dictate the required level of destruction; sensitive environments may require purging or outright physical destruction.

4. Common Security Policies

All policies must ensure compliance with legal and contractual obligations. Common policies include:

A. Data Handling Policy

Defines internal and external data usage. Specifies classifications and restrictions, often referencing legal requirements.

- *Example:* PCI DSS requires credit card data be encrypted.

B. Password Policy

Details password creation and management standards, enforcement, and the commitment to data access security.

C. Acceptable Use Policy (AUP)

Defines how IT systems and data may be used. Employees must sign this, acknowledging understanding and consequences. Covers topics like data access, system usage, retention, and internet practices.

- **Bring Your Own Device (BYOD) Policy:** Allows or restricts personally owned devices. Managing BYOD presents unique security challenges, particularly when auditing or securing devices containing a mix of personal and business data.

D. Privacy Policy

Covers handling of personally identifiable information (PII) and, for healthcare, electronic protected health information (ePHI). Outlines definitions, handling protocols, legal responsibilities, and penalties for violations, referencing relevant laws (e.g., HIPAA, GDPR, PIPEDA).

E. Change Management Policy

Describes processes for instituting changes in systems or environments to ensure security and prevent introduction of new vulnerabilities.

5. Common Security Policies Deeper Dive

Policies must be aligned to the organization's goals and specify consequences for violations:

- Penalties may range from warnings to termination.
- Clarity during onboarding is crucial. Employees must acknowledge and sign the policies.
- Enforcement responsibilities must be clear, and employee understanding should be validated (e.g., quizzes).
- Sound policies are the foundation of security posture and must back all procedures.

6. Supporting Security Policies with Procedures

Security policies vary by organizational culture and risk tolerance:

- Some organizations allow extensive personal use of assets, increasing morale but also risk.
- Organizations with more sensitive data (e.g., healthcare, defense) enforce stricter restrictions.

- All security policies must strike a balance between operational flexibility and risk, while fulfilling *regulatory* requirements.
- The acceptable use policy should be shaped by the sensitivity of the data the organization handles.

Summary Table: Data Life Cycle & Practices

Life Cycle Stage	Key Practice	Example or Note
Create	Data generation, initial confidentiality	Writing a proposal or idea
Store	Secure recording/storage	Encrypting files on a server
Use	Controlled access and modification	Editing business plans, version control
Share	Secure sharing or transmission	Sending encrypted files to collaborators
Archive	Long-term, secure storage	Moving closed projects to backup servers
Destroy	Secure deletion/destruction	Shredding hard drives, overwriting data

Best Practices:

- Use classification and labeling to control access.
- Apply retention and destruction policies tailored to legal, regulatory, and business requirements.
- Back procedures with clear, enforceable policies.
- Remain aware of *all* applicable regulations, standards, and the organization's risk tolerance at every life cycle stage.

▼ Change Management

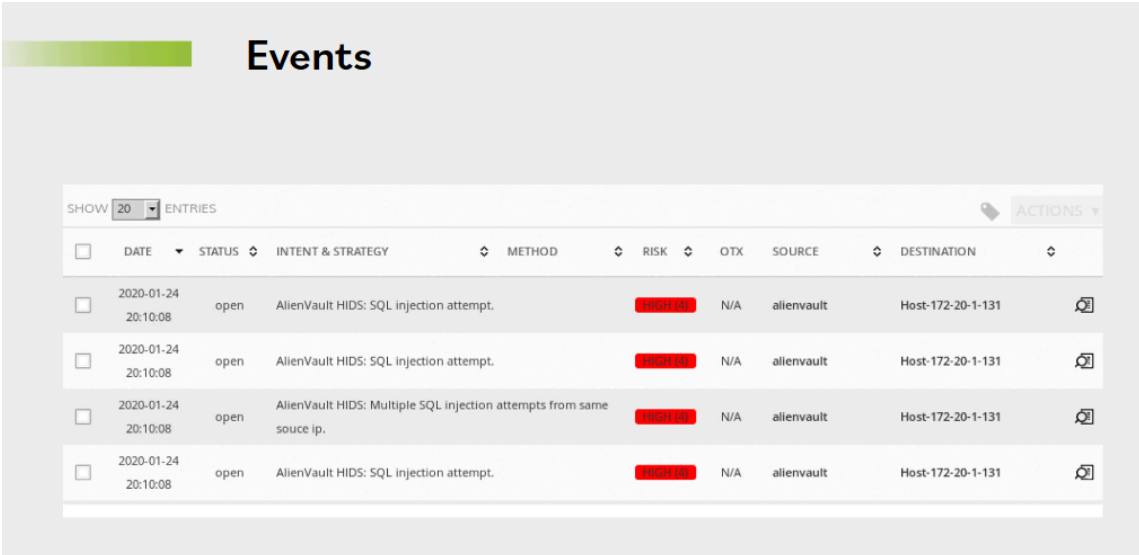
Concepts Covered:

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

1. Logging and Monitoring Security Events

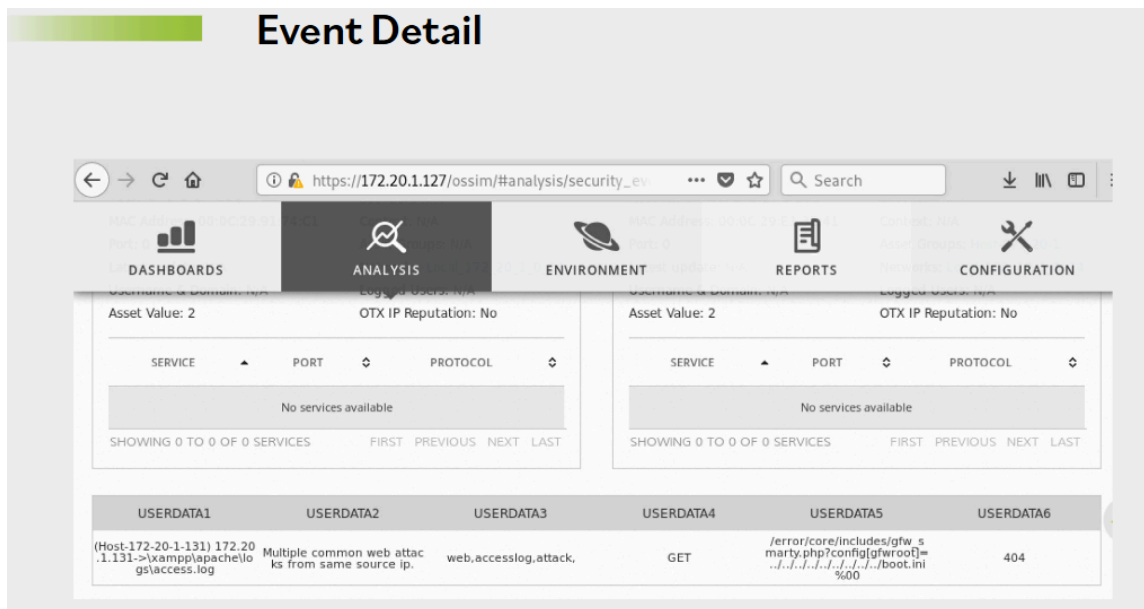
What is Logging?

- **Logging** is the primary method of instrumenting a system to capture signals (data) generated by **events**.
- **Events** refer to any actions occurring within the system environment that produce measurable or observable changes to system elements or resources.



The screenshot shows a web interface titled "Events". At the top left, there is a green header bar. Below it, the word "Events" is displayed in a large, bold font. To the right of the title, there is a search bar and a "SHOW 20 ENTRIES" dropdown menu. Below the search bar, there is a table with columns: DATE, STATUS, INTENT & STRATEGY, METHOD, RISK, OTX, SOURCE, and DESTINATION. The table contains four rows of data, all with a status of "open" and a risk level of "HIGH". The first three rows are for "AlienVault HIDS: SQL injection attempt." and the fourth row is for "AlienVault HIDS: Multiple SQL injection attempts from same source ip.".

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



Relevant Information to Log

Logging should capture data points relevant to security and operations:

- **User Identifiers (User IDs)**
- **System Activities** (e.g., file access, commands executed)
- **Dates and Times** of key events like logon and logoff
- **Device and Location Identity** (IP addresses, terminal details)
- **Access Attempts** (both successful and failed)
- **System Configuration Changes**
- **System Protection Events** (activation/deactivation of security controls)

Importance of Logging

- Logging imposes some computational cost but is **invaluable for accountability** and forensic activities.
- **Regular log review** is best practice across all computer systems to detect anomalies, policy violations, or breaches.
- Enables **correlation across systems** to understand the sequence and relationship of events.
- Supports **security assessment, incident detection, fraud detection, and operational troubleshooting**.
- Crucial for **audits, forensic investigations**, and validating if vulnerabilities were exploited historically.

Log Management Infrastructure

- Organizations should develop a structured log management system comprising:
 - Collection, storage, and analysis components.
 - Controls to maintain **log integrity** and prevent unauthorized modification or deletion.
 - Retain logs per **retention policies and legal/regulatory requirements**.
- **Log preservation policies** ensure attackers cannot erase evidence.
- Logs often contain sensitive information, so they need to be protected against malicious use.

Challenges

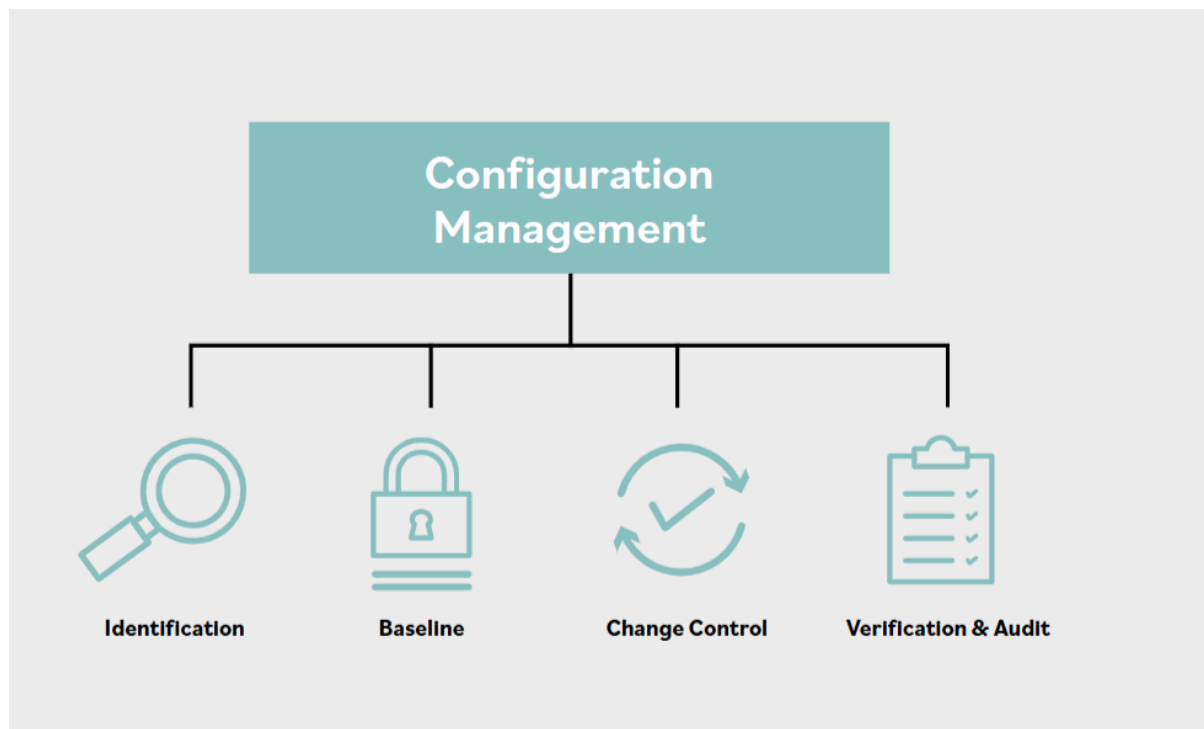
- Risks like log file editing, deletion, or storage limits exceeded can create gaps.
- Requires continuous operational monitoring to avoid loss or corruption of logs.

2. Configuration Management Overview

What Is Configuration Management?

- A **process and discipline** to ensure that only **authorized and validated changes** occur on a system.
- Combines **decision-making** and **control processes** for effective system management.

Core Components



1. Identification

- Establish a baseline by identifying the full system, including:
 - Hardware, software, interfaces, and documentation.

2. Baseline

- A **security baseline** is the minimal, agreed-upon level of protection.
- It serves as a reference standard to ensure all technological updates conform to organizational security expectations.

3. Change Control

- Formal procedure to **request, review, approve, and implement changes**.
- Applies to updates, patches, system configuration modifications.

4. Verification and Audit

- Regression and validation testing ensures **no disruptions or new vulnerabilities** occur due to changes.
- Audits ensure the current system state matches the initial baseline plus all authorized changes.

5. Inventory Management

- Catalogue all **information assets**, including hardware, software, and data.
- "You can't protect what you don't know you have."
- Maintaining accurate, up-to-date inventory is challenging but essential.

Updates and Patches

- **Updates:** Repairs, maintenance, and functional improvements that must be tested before deployment.
- **Patches:** Software/hardware fixes addressing vulnerabilities or bugs.
- Patch management requires:
 - Timely deployment, prioritization of critical patches.
 - Testing in environments that mirror production to avoid disruptions.
 - Rollback plans if a patch causes problems.
- Automated or unattended patching can conflict with stability and downtime concerns.

3. The Risks of Change

Challenges of Making Changes

- Even with thorough testing, **unintended consequences** can occur.

- A robust **change management process** includes:
 - Testing in model environments separate from production.
 - Having a **rollback plan** ready to restore previous stable system states.
- Maintaining separate test environments is logistically and financially difficult for many organizations.
- Organizations without test environments may rely on **vendor or third-party certification** but this is less reliable.
- Rollback plans are especially critical for organizations without full testing capacity to quickly recover from faulty changes.

4. Change Management Components

Change management consists of managing the lifecycle of changes from request to implementation. The main components include:

1. Documentation

- Initiates with a **Request for Change (RFC)**.
- Each step along the change path must be **formally documented** with logs and approvals.

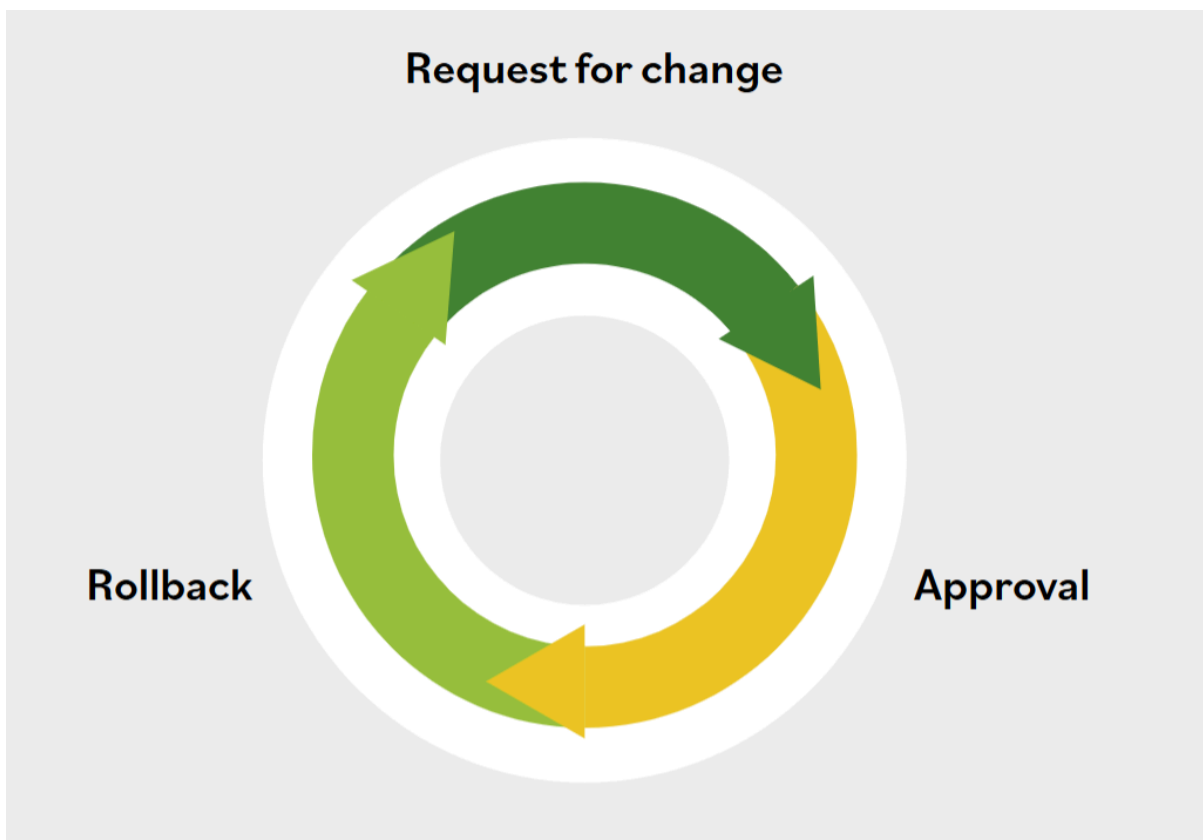
2. Approval

- Evaluate RFCs for completeness and assign to appropriate approval workflows.
- Includes stakeholder reviews, resource assignments.
- Formal approval or rejection documented.
- Risk-based evaluation ensures changes reflect organizational priorities and compliance requirements.

3. Rollback

- Prepare rollback procedures before implementation.
- Rollback activities include:

- Scheduling the change.
- Testing the change.
- Verifying rollback methods.
- Implementing and monitoring the change.
- Documenting results.
- Rollback may be immediate or scheduled depending on the impact of the change.



5. Change Management Components in the Workplace

Continuous Cycle of Change Management

- Change management is an ongoing, cyclical process requiring continuous monitoring.

- Every change request must:
 - Pass through **formal approval**.
 - Have a well-defined **rollback plan** if problems occur.
 - Be **communicated** clearly to all relevant stakeholders.

Organizational Roles

- Although organization-wide, change coordination often falls under:
 - **Information security professionals** (oversight and governance).
 - Or **IT/development teams** potentially in collaboration with **quality or risk management**.
- Input from end-users, IT, security, development, and management is vital.
- The goal is to ensure all changes are:
 - Properly tested,
 - Approved,
 - Communicated before implementation,
 - And monitored after deployment for effectiveness.

Summary

Concept	Key Takeaways
Logging & Monitoring	Capture and review relevant events; essential for accountability, incident detection, and audits.
Configuration Management	Control system changes via identification, baselines, change control, verification, and inventory.
Risks of Change	Changes can introduce unintended problems; robust testing and rollback plans are critical.
Change Management Components	Formal documentation, approvals, controlled implementations, and tested rollback procedures.
Workplace Change Management	Continuous process involving multiple organizational roles with strong communication and monitoring.

Example Scenario (Change Management)

- An organization wants to deploy a security patch to a critical web server.
- **Step 1:** The IT team submits an RFC detailing the change.
- **Step 2:** The security team reviews and approves after assessing risks.
- **Step 3:** The patch is tested in a separate staging environment.
- **Step 4:** Rollback procedures are documented and tested.
- **Step 5:** The patch is deployed in production.
- **Step 6:** Monitor system for any adverse effects. If any issue arises, rollback is executed promptly.

▼ Hashing & Encryption

Concepts Covered:

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

1. Encryption Overview

Encryption is a fundamental component of security in the digital age, playing a crucial role in everything from protecting online transactions to verifying the authenticity of digital files and messages.

Key Concepts

- **Cryptography** uses algorithms to convert readable data (plaintext) into an unreadable form (ciphertext), ensuring data remains private and secure.

- **Digital Signatures** verify the authenticity of data, such as software updates or contracts, ensuring the sender's identity cannot be repudiated.
- **Confidentiality:** Encryption ensures only authorized users can access the content, keeping sensitive data like personal information, business transactions, or emails secure.
- **Integrity:** Hash functions and digital signatures can detect if a message or data was altered—either by accident or maliciously—by comparing digests (hashes) before and after transmission.

Example

- Encrypted emails, online banking transactions, and digital contracts all use cryptography to keep content secure and authentic.

2. Encryption Deep Dive

Encryption is not a new concept—it has historical roots stretching from ancient tribal symbols to modern algorithms. The essence is always to keep information hidden from unauthorized viewers.

How Encryption Works

- **Plaintext:** Readable data, such as a PDF your accountant sends.
- **Encryption Algorithm:** A recipe for turning plaintext into unreadable ciphertext.
- **Encryption Key:** The secret needed to run the algorithm.
- **Ciphertext:** The scrambled output, unreadable without the key.

Key Management is crucial, especially at commercial scale. Improper handling of keys can jeopardize all protections—dedicated servers or third-party management often help.

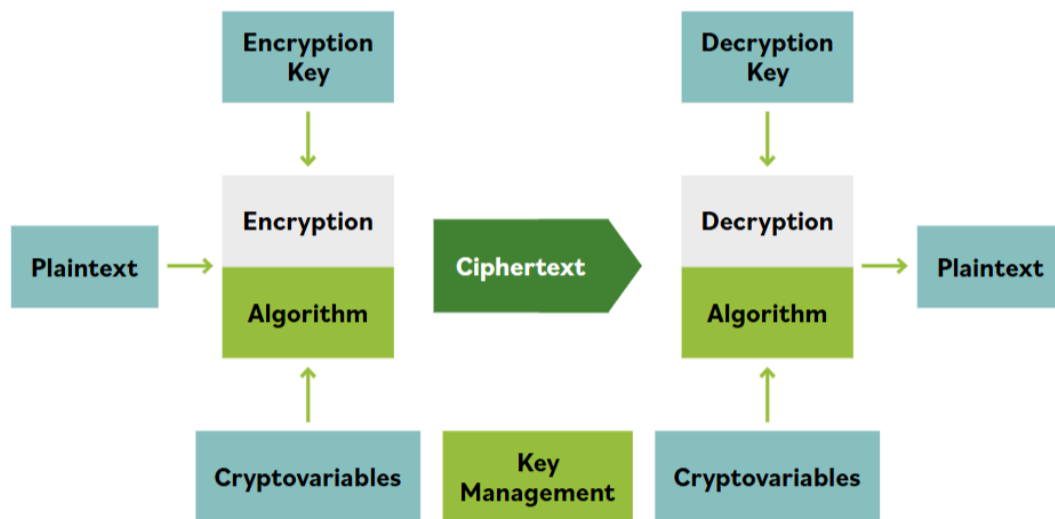
Types

- **Symmetric Encryption:** Same key for encryption and decryption.

- **Asymmetric Encryption:** Different keys for encryption and decryption (public/private key pairs), often using certificates to confirm identities.

Example

- Encrypted PDFs require the correct key to unlock. In businesses, keys are stored in protected locations to prevent all keys being compromised.



3. Hashing

Hashing is a process by which data of any size is converted into a fixed-size output ("hash digest") using a hash algorithm. While similar to encryption, hashing is **non-reversible**: you cannot get the original data back from the hash.

Properties of Cryptographic Hash Functions

- **Easy to compute** for any input.
- **Nonreversible:** Cannot reconstruct the original message from the hash.
- **Integrity Assurance:** Changing even a single bit in the input drastically changes the result.
- **Unique:** Each message produces a unique hash; collisions (different data with same hash) should be infeasible.

- **Deterministic:** Same input always produces the same output.

Uses

- Ensuring message integrity (detecting accidental or malicious change).
- Digital signatures, password verification, file fingerprinting, and checksums for data verification.

Example

Suppose you transmit the text "Fox." A hash of this text is created and sent with the message. When the receiver gets the message, they hash what they received and compare the result. If it matches, the message is intact.

- "Fox" → DFCD 3454 BBEA 788A...
- "The red fox jumps over the blue dog" → 0086 46BB FB7D...
- Changing even a single letter yields a completely different hash.

4. Hashing Deep Dive

Hash functions turn data into a digest that is **meaningless** to humans and remains the same length regardless of input size. Even minor changes yield utterly different digests, making them reliable for integrity checks.

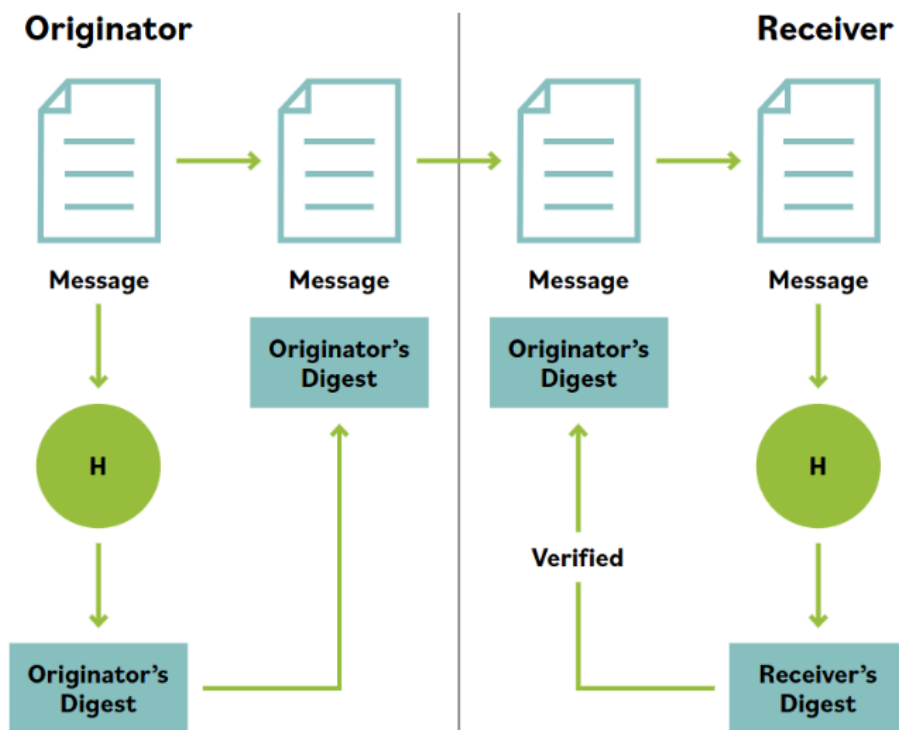
Real-world Applications

- **Banking:** Catching unauthorized changes in payment amounts (e.g., detecting if an extra zero is added).
- **Software Distribution:** Ensuring software received matches what was sent. Any mismatch in hash digests signals possible tampering.

Example

Before deploying new software, compare the hash digest from the vendor with the original. Discrepancies require careful investigation.

- **Case study:** The University of Florida distributed compromised Windows CDs to students. Mismatched hashes exposed the issue.



5. Asymmetric Encryption

Asymmetric encryption uses **two keys**—a public key for encryption and a private key for decryption—offering greater flexibility and security than symmetric encryption.

Features

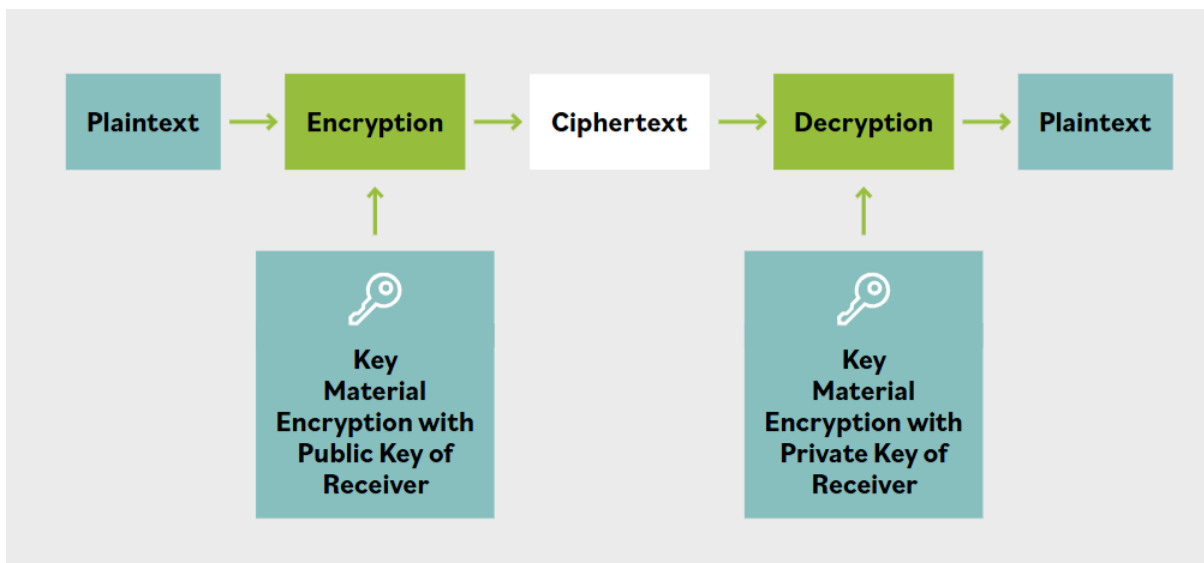
- **Key Pair:** Each user has a public key (shared freely) and a private key (kept secret).
- **Confidential Messaging:** Anyone can encrypt a message with the recipient's public key, but only the recipient's private key can decrypt it.
- **Non-repudiation:** Senders cannot deny sending a message if it was signed with their private key.
- **Scalability:** Only two keys per user, compared to the much larger number required for symmetric systems as users increase.

Drawbacks

- **Performance:** Asymmetric algorithms are computationally intensive and slow—unsuitable for encrypting large amounts of data quickly.
- **Usage:** Typically used for exchanging small secrets or establishing secure sessions, not for bulk encryption.

Example

- Alice wants to send Bob a confidential message. She encrypts it with Bob's public key—only Bob can decrypt with his private key.



6. Symmetric Encryption

Symmetric encryption algorithms use a **single secret key** for both encrypting and decrypting data. The process is simple and fast, making it ideal for bulk data encryption.

Features

- **Same Key Shared:** Both parties must agree on and safeguard the key.
- **Key Distribution Problems:** Keys must be exchanged securely, often via separate ("out-of-band") channels.

- **Scalability Issues:** Number of keys grows rapidly as number of users increases ($n(n-1)/2$ keys for n users).

Applications

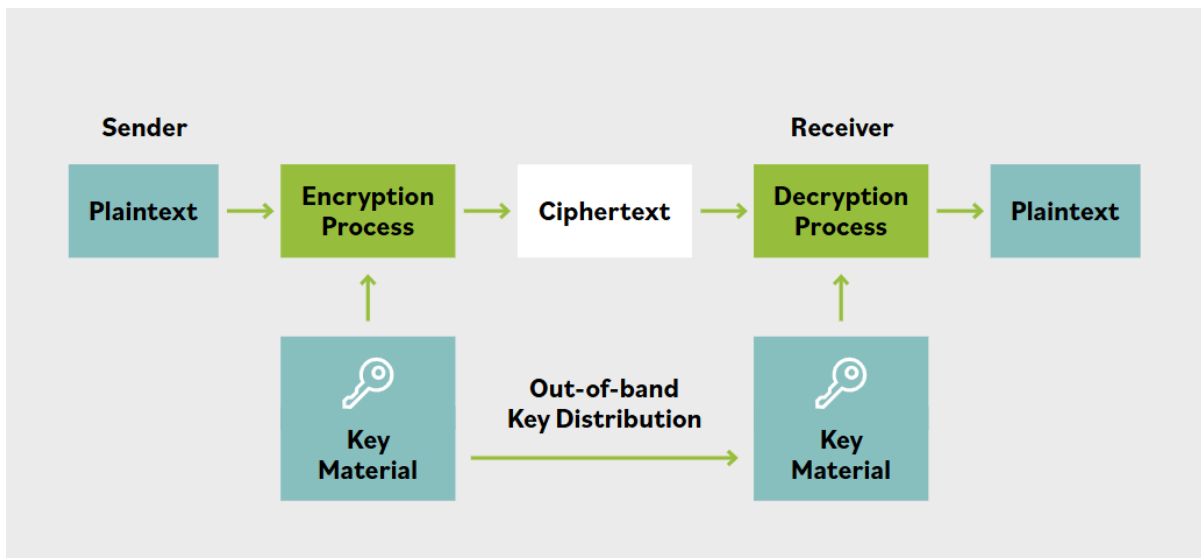
- **Bulk Data Encryption:** Encrypting files, hard drives, network traffic (e.g., VPN data).
- **Streaming Data:** Encrypted voice/video communication, online gaming.

Other Names

- Single key, shared key, secret key, session key.

Example

- **Caesar Cipher:** A substitution cipher that replaces each letter with another based on a fixed shift.



Summary Table: Symmetric vs Asymmetric Encryption

Feature	Symmetric Encryption	Asymmetric Encryption
Key Usage	Same key for encryption/decryption	Public/private key pair

Feature	Symmetric Encryption	Asymmetric Encryption
Speed	Very fast	Slower
Key Distribution	Difficult, requires secure channel	Public key can be shared openly
Scalability	Poor for large groups	Scales efficiently
Common Uses	Bulk data, streaming, backups	Secure communication, signatures
Vulnerabilities	Key exposure, key management	Computational inefficiency
Example	AES, DES	RSA, ECC

▼ 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 Event Example

A data security event log records activities that might indicate unauthorized attempts to access or compromise a secure system. For example, such a log may capture events where an individual tries to break into a protected file or hijack a server.

Security engineers analyze these logs to determine:

- **Who attempted access:** Usernames, IP addresses, or device identifiers.
- **Port usage:** Whether the attempt was on a standard secure port (like 443 for HTTPS) or a suspicious one, which could indicate attempted exploitation.
- **Timing and frequency:** Multiple failed logins over a short time could suggest brute-force attacks.

While many modern security systems present logs in user-friendly dashboards, deep security work often involves reading raw log files, which may look cryptic but contain crucial forensic information.

Key Principle:

Information security cannot be added as an afterthought. It must be planned from the beginning, embedded into system architecture before data even enters the network. Patching or adding defenses to an already insecure system yields limited protection.

Example log snippet:

```
textJul 24 13:22:10 server1 sshd[1523]: Failed password for invalid user admin from 192.168.1.20 port 45231 ssh2
```

This log shows a failed attempt to log in as "admin" from a particular IP and port—key details for investigation.

2. Event Logging Best Practices

Two Major Monitoring Categories:

- **Ingress monitoring:** Supervises all inbound traffic and access attempts.
- **Egress monitoring:** Controls information leaving the organization.

Ingress Monitoring Devices:

- **Firewalls:** Block/filter incoming malicious traffic.
- **IDS/IPS (Intrusion Detection/Prevention Systems):** Detect/block suspicious activities.
- **Gateways:** Manage access between internal/external networks.

- **SIEM (Security Information and Event Management):** Aggregate and analyze event logs in real time.
- **Remote authentication servers:** Verify user credentials.
- **Anti-malware tools:** Scan inbound content for threats.

Egress Monitoring:

- **DLP (Data Loss/Leak Prevention) Solutions:**

Inspect all outgoing data for sensitive information, covering:

- Emails and attachments
- Website postings
- Transfers to USB or other portable media
- Application data/API calls
- FTP uploads

Best Practice:

Deploy solutions that can log, alert, and, if necessary, block suspicious behavior at both entry and exit points of the organization's digital environment.

3. How Passwords Work

When you create a password for a system:

- It's usually stored not as plain text but as a **hash digest**—a fixed-length encrypted output generated by a mathematical hash function.
- When you log in, your password is hashed using the same algorithm, and the system compares the hashes. If they match, access is granted.

Why Hashing?

It ensures the original password isn't stored or transmitted, reducing exposure risk.

Hash Attacks:

If attackers steal the hash file and know the hash algorithm, they can try many password guesses (brute force or "dictionary" attacks) offline until they find a

match.

Modern Practices:

- Requiring passwords with minimum lengths and complexity (numbers, uppercase, symbols) to increase hash unpredictability.
- Systems often indicate password strength, guiding users to create secure passwords.
- However, hashing alone is becoming less sufficient—multi-factor authentication and biometrics are now common for extra layers.

Example Hash:

A simple password like "password1" might hash to

5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8 (using SHA-1, for illustration).

4. Security Awareness Training

Purpose:

Ensure everyone in the organization understands their security responsibilities and eliminates risky behavior due to ignorance or complacency.

Three Learning Types in Organizations:

- **Education:** Develops conceptual understanding, enabling learners to relate security concepts to their work and apply them.
- **Training:** Focuses on practical skills—knowing **what** to do, **when**, and **how** (e.g., handling a suspicious email).
- **Awareness:** Engages employees' attention, making them conscious of security topics by providing reminders or introducing issues.

Example:

A new executive with no prior compliance background must first become aware of specific security requirements before progressing to education or training.

5. Security Awareness Training Example

Fire Safety Context:

- **Education:** Explains how fire detection and suppression systems interact with building utilities.
- **Training:** Teaches staff correct response steps for alarms, system failures, or emergencies.
- **Awareness:** Uses signs, floor markings, and recurring reminders to reinforce knowledge and keep procedures top of mind.

Anti-Phishing Campaign:

- **Education:** Guides employees to understand different phishing tactics and encourages strategy development for defense.
- **Training:** Involves practice (e.g., fake phishing emails) to build proficiency in identifying and reporting attacks.
- **Awareness:** Maintains high visibility of threats, informs staff of new tactics, and keeps alertness active via frequent reminders.

6. Password Advice and Examples

Strength by Length and Complexity:

- **10-digit numeric password:** Cracked in about 5 seconds using brute-force software.
- **8-character mixed password:** Takes about 35 days to crack.
- **16-character password with uppercase and special symbols:** Estimated cracking time is around 152,000 years.

Examples:

- **Weak:** "password1"
- **Stronger:** "J#m9\$e7pqW\$Z8*ua"

Best Practice:

Favor longer passwords with combinations of upper/lowercase, numbers, and symbols. Follow password policies strictly to vastly improve security.

7. Password Protection

Password managers simplify secure access by storing many passwords, so users only remember one master password.

Risks:

- If the password manager (or its master password) is compromised, all stored passwords could be accessed.
- Incidents have occurred where poorly chosen master passwords led to all of a user's data being stolen via cloud providers.

Recommended Practices:

- Different passwords for every system.
- Use organization-approved password managers.
- Never:
 - Reuse passwords for business and personal use.
 - Leave passwords written in unprotected areas.
 - Share credentials with colleagues or tech support.

8. Best Practices of Security Awareness Training

- **Continuous threat communication:** Regular updates keep staff aware of emerging threats.
- **Friendly competition:** Encourage departments to spot phishing attempts, creating engagement.
- **Reminders:** Use branded items (stress balls, posters) and automatic screen locks to cultivate vigilance.
- **Feedback:** Nurture a positive environment by rewarding correct responses to simulations (like reporting phishing emails).
- **Leadership involvement:** Organization's leaders should champion security training and enable staff to practice with simulations and exercises.

- **Positive experience:** Training is most effective when seen as supportive, not punitive (unless required).

9. Phishing

Phishing is a major cyber threat where attackers try to trick targets into revealing sensitive data (passwords, financial info, etc.) by posing as legitimate entities.

Delivery Methods:

- Email (most common)
- Phone calls
- Spam/instant messages
- Malicious attachments
- Fake websites

Whaling:

A subtype of phishing targeting high-level executives or wealthy individuals to trick them into large money transfers or surrendering sensitive credentials.

10. Social Engineering

Social engineering uses psychological manipulation to trick people into divulging confidential information or granting system access.

Key Techniques:

- **Phone phishing (vishing):** Automated or live calls impersonating banks or institutions, extracting account info via deception.
- **Quid pro quo:** Offers a benefit in exchange for sensitive information (e.g., gift cards for passwords).
- **Pretexting:** Pretending to be an authority (like IT support) to justify asking for credentials or system access.
- **Tailgating:** Physically following a permitted user into a secure area or borrowing devices under false pretenses to install malicious software.

Why It Works:

Social engineering exploits human nature—trust, helpfulness, or fear. The best defense is ongoing education, training, and awareness that security is everyone's job, not just IT's.