Partially Comprehensive Study Guide: Information Security*

April 25, 2025

1 Introduction

2 Cryptography

2.1 Principles

- **Kerckhoffs's Principle:** A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- **Shannons Maxim:** One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.

2.2 Cryptography Definition

Cryptography is the conversion of data into a scrambled code that is encrypted and sent across a network. It is used to protect confidential data like emails, chat sessions, web transactions, and personal/corporate data.

2.3 Objectives of Cryptography

Cryptography aims to provide:

- **Confidentiality:** Preventing unauthorized disclosure.
- **Integrity:** Ensuring data has not been altered.
- **Authentication:** Verifying the identity of parties.
- **Non-repudiation:** Preventing denial of having sent or received a communication.

2.4 Cryptographic Process

The process involves converting plaintext (readable format) into ciphertext (unreadable format) using encryption algorithms (like RSA, DES, AES) and a key. The ciphertext is then transmitted, and at the destination, it is converted back to plaintext using a decryption process and the appropriate key.

2.5 Computationally Secure

An encryption scheme is computationally secure if the ciphertext generated meets certain criteria, implying it is infeasible for an attacker to break the encryption with available computational resources.

2.6 Cipher vs Code

A cipher is distinct from a code. A code is a system of rules to convert information into another form, sometimes shortened or secret, and typically requires a codebook. Ciphers use algorithms and keys.

*For sufficiently small values of comprehensive

2.7 Types of Ciphers

- **Classical Ciphers:** The most basic type, operating on letters (A-Z), generally implemented by hand or simple devices. They provide only confidentiality and are generally unreliable due to ease of deciphering.
- **Block Ciphers:** Process plaintext input in fixed-size blocks, producing ciphertext blocks of equal size with an unvarying transformation specified by a symmetric key. Most modern ciphers are block ciphers and are widely used to encrypt bulk data. Padding is used if the block size is smaller than required by the cipher. Examples include DES, AES, and IDEA.

2.8 Attacks on Ciphers/Encryption

- **Brute-Force Attack:** Trying every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. Requires recognizing plaintext. On average, half of all possible keys must be tried.
- **Known-Plaintext Attack (KPA):** The cryptanalyst has access to at least a limited number of pairs of plaintext and corresponding ciphertext ("cribs").
- **Chosen-Plaintext Attack:** The cryptanalyst can choose plaintext and obtain the corresponding ciphertext to gather information. This technique was historically used in attacks like "gardening" against the Enigma machine.
- **Chosen-Ciphertext Attack (CCA):** An attack model where the cryptanalyst can obtain the decryptions of chosen ciphertexts to recover the secret key. Cryptographic smart cards are particularly vulnerable if under adversary control.

2.9 Enigma Machine

The Enigma machine was a cipher device used in the early to mid-20th century, notably by Nazi Germany in WWII, for protecting communications. It was considered highly secure, using a polyalphabetic substitution cipher with rotors. German Army versions had 3 rotors, while the Navy used a 4-rotor mechanism.

2.10 Cryptanalysis of the Enigma

Cryptanalysis efforts against the Enigma machine, like those led by Alan Turing using the Bombe, exploited patterns and known-plaintext attacks (such as recurring messages like "nothing to report") to help determine daily keys.

2.11 Symmetric Encryption

Symmetric encryption, also called secret-key or shared-key encryption, uses the same key for both encryption and decryption. Both sender and receiver must possess the same key. The sender encrypts plaintext, sends the ciphertext, and the receiver uses the same key to decrypt it. A strong algorithm is required so that an opponent knowing the algorithm and having ciphertext cannot decrypt without the key. A major challenge is the pre-sharing of the key, leading to key management issues.

2.12 Attacking Symmetric Encryption

- **Cryptanalysis:** Exploits algorithm characteristics to deduce plaintext or the key. Deducing the key is catastrophic as all messages using that key are compromised.
- **Brute-Force Attack:** Trying every possible key until plaintext is obtained.

2.13 Symmetric Block Encryption Parameters

Block ciphers have parameters like block size and key length. Larger block sizes offer greater security but reduce speed; 128 bits is a common size. Performance and ease of analysis are important considerations.

2.14 Feistel Cipher Structure

Defined by Horst Feistel, this structure forms the basis for most modern symmetric key algorithms. Plaintext is divided into two halves (left and right), which pass through multiple rounds of processing before being combined into ciphertext. Each round involves applying a subkey and an encryption function to one half and XORing the result with the other half.

2.15 Data Encryption Standard (DES)

Adopted by NIST in 1977 (FIPS PUB 46), DES is a symmetric cryptosystem using a 64-bit key (56 random bits, 8 for error detection), offering a large number of possible keys. Concerns arose over the strength of the algorithm (though no fatal weakness was found) and the small key size (56 bits) which became vulnerable to brute-force attacks. **DES is broken and should not be used.**

2.16 Triple Data Encryption Standard (3DES)

3DES, developed while searching for a new standard, involves repeating the basic DES algorithm three times using two or three unique keys, resulting in key sizes of 112 or 168 bits. **3DES has been replaced by AES and is no longer recommended for use.**

2.17 Advanced Encryption Standard (AES)

AES is a NIST specification for electronic data encryption, adopted as the replacement for 3DES. It is a symmetric-key algorithm efficient in both software and hardware. AES supports a block length of 128 bits and key lengths of 128, 192, and 256 bits. Longer messages are encrypted by breaking them into blocks and encrypting each separately, which can be done in various modes like Electronic Codebook (ECB).

2.18 Electronic Codebook (ECB)

ECB is a simple mode of operation for block ciphers where the message is divided into blocks, and each block is encrypted independently using the same key.

2.19 One-Time Pad

The One-Time Pad is an encryption scheme that, if used correctly, results in ciphertext that is impossible to decrypt or break (theoretically perfectly secure). It requires a single-use pre-shared key that is at least as long as the plaintext. Each bit/character of plaintext is combined with the corresponding key bit/character using modular addition.

2.20 One-Time Pad Conditions for Unbreakability

The theoretical unbreakability of OTP depends on four conditions:

1. The key must be at least as long as the plaintext.
2. The key must be truly random (from a non-algorithmic, chaotic source, with entropy equal to plaintext bits).
3. The key must never be reused.
4. The key must be kept completely secret by the communicating parties.

Ensuring these conditions, particularly true randomness, single-use, secrecy, and complete destruction after use, is difficult in practice.

2.21 Message Digest Functions (Hashing)

While encryption protects against passive attacks (eavesdropping), active attacks (data falsification) require message or data authentication, which hash functions help provide. Hash functions do not provide confidentiality but verify that message contents have not been altered and, when combined with other techniques, that the source is authentic. They calculate a unique fixed-size bit string (message digest) of any arbitrary block of information. Unlike HMAC, hashes don't require a secret key.

2.22 Message Authentication Code (MAC)

A MAC function takes a message and a secret key to produce a small fixed-size block. The receiver, sharing the secret key, can calculate the MAC of the received message and compare it to the received MAC. This assures the receiver that the message has not been altered and is from the alleged sender (who knows the key). Including a sequence number can assure proper sequence.

2.23 Role of Cryptographic Hash Function

The main role of a cryptographic hash function is document integrity. They are integral to digital signatures, used to calculate the signature of a document's hash value, which is smaller than the document itself. Hash functions are one-way, resistant to attack, mostly unique, and widely distributed.

2.24 Widely Used Message Digest Algorithms

- MD5: **MD5 is BROKEN and should not be used for anything serious.**
- SHA (Secure Hash Algorithm): Developed by NIST (FIPS 180). SHA-1 produces a 160-bit hash; **Do not use SHA-1 any more** due to collision vulnerabilities (an attack found two messages with the same SHA-1 hash using significantly fewer operations than expected). FIPS 180-2 defined SHA-256, SHA-384, and SHA-512. SHA provides integrity.

2.25 Length Extension Attack

Certain hash functions (including MD5 and SHA-1) are vulnerable to length extension attacks. If an attacker knows the hash of a message and the message's length, they can append additional data and compute the hash of the new, extended message without knowing the secret key used in the original hashing process.

2.26 Asymmetric Encryption

Asymmetric encryption, or public-key cryptography, uses a pair of mathematically related keys: a public key and a private key. The public key can be widely distributed, while the private key is kept secret by the owner. To send a confidential message, the sender encrypts it using the recipient's public key. Only the recipient, using their corresponding private key, can decrypt the message. Asymmetric encryption primarily provides confidentiality but can be combined with other techniques for non-repudiation, integrity, and authentication. A ciphertext generated using the private key can be decrypted by anyone using the public key.

2.27 Requirements for Public Key Cryptography

For a public-key cryptosystem to be effective, it must be computationally easy to:

- Generate a pair of keys (public and private).
- Encrypt a message using the public key.
- Decrypt a ciphertext using the corresponding private key.

It must be computationally infeasible for an attacker to determine the private key from the public key or decrypt ciphertext knowing only the public key.

2.28 RivestShamirAdleman (RSA)

RSA is a well-known public-key cryptosystem (asymmetric) used for Internet encryption and authentication. It relies on modular arithmetic and number theory involving two large prime numbers. RSA is used for public key encryption and digital signatures. It was the first technique used to generate digital signatures. RSA can provide confidentiality, authentication, integrity, and non-repudiation. The process involves selecting primes, calculating public and private key components based on them.

2.29 Security of RSA

The security of RSA relies on the difficulty of factoring large numbers. Brute force attacks (trying all possible private keys) are computationally infeasible with sufficiently large keys.

2.30 Comparison: Symmetric vs Asymmetric Encryption

Symmetric encryption algorithms are typically faster and simpler, using smaller keys and less processing power. Asymmetric encryption is generally slower, requires larger keys, and is computationally intensive. Symmetric keys must be pre-shared, which is a major key management challenge. Asymmetric encryption helps solve the key exchange problem for symmetric encryption.

2.31 Diffie-Hellman Key Exchange

Diffie-Hellman is a cryptographic protocol allowing two parties to establish a shared secret key over an insecure channel without having any prior shared secret. It was developed by Diffie and Hellman and independently by Williamson. Both parties contribute to generating the shared key, and an attacker who intercepts the public exchange cannot compute the final shared key. This allows parties to then use symmetric encryption for their communication.

2.32 Digital Signatures

A digital signature uses asymmetric cryptography and a one-way hash function to provide non-repudiation and integrity. To sign a document, the sender computes a hash of the document and encrypts this hash value using their private key. This encrypted hash is the digital signature. The recipient verifies the signature by computing a hash of the received document and decrypting the received signature using the sender's public key. If the two hash values match, it confirms the document has not been modified since it was signed and that the signature came from the holder of the private key (non-repudiation).

3 User Authentication

3.1 User Authentication

User authentication is the process of verifying the identity of a user. It is a critical security function.

3.2 Levels of Assurance (PIV)

The Personal Identity Verification (PIV) standard defines levels of assurance regarding an asserted identity's validity. These levels consider the confidence in the vetting process used to establish the identity and the confidence that the individual using a credential is the one to whom it was issued.

3.3 Four Levels of Assurance

- **Level 1:** Little or no confidence in the asserted identity's validity (e.g., a Reddit post).
- **Level 2:** Some confidence in the asserted identity's validity, requiring the use of a secure authentication protocol.
- Higher levels (3 and 4, details not provided) offer greater confidence.

3.4 Password-Based Authentication

A common method using a name or identifier (ID) and a password. The ID determines if the user is authorized and their privileges, often used in discretionary access control.

3.5 Vulnerability of Passwords

Passwords are vulnerable to various attacks:

- **Offline dictionary attack:** An attacker obtains a system password file and compares hashed passwords against hashes of commonly used passwords or dictionary words. If a match is found, access is gained.
- **Guessing attacks:** Trying variations based on knowledge of the account holder (e.g., pet names, notable dates) or system password policies. Countermeasures include training and enforcement of strong password policies (secrecy, minimum length, character set, avoiding well-known IDs, periodic changes).
- **Password re-use:** Using the same or similar password across multiple systems; a breach on one system compromises others. A policy forbidding re-use is a countermeasure. Using Two-Factor Authentication (2FA) and a Password Manager are recommended to mitigate this risk. **Electronic monitoring:** Passwords communicated over a network are vulnerable to eavesdropping.

3.6 Why Keep Using Passwords?

Despite vulnerabilities, passwords remain common. Automated password managers exist to relieve users of the burden but have limitations regarding roaming, synchronization across platforms, and usability research.

3.7 Salting Passwords

To counter dictionary and rainbow table attacks, a password should be combined with a unique, fixed-length salt value before hashing. This makes finding out if the same password is used across multiple systems nearly impossible for an attacker who obtains multiple password files.

3.8 Password Cracking

Traditional password guessing involves comparing hashes of dictionary words or variations against stored hashes in a password file. This process can be slow, especially with strong policies and salting.

3.9 Rainbow Tables

Rainbow tables trade space for time by precomputing potential hash values for a large dictionary of passwords combined with possible salt values. This can crack many password hashes quickly. Countermeasures include using a sufficiently large salt value and hash length.

3.10 Password File Protection

Denying an attacker access to the password file is a way to thwart attacks. Hashed passwords are often kept in a separate, more protected shadow password file. Access control measures must be complemented by policies forcing users to select difficult-to-guess passwords.

3.11 Password Selection

Left unconstrained, users choose weak passwords. Random passwords are hard to guess but also hard to remember. The goal is to eliminate guessable passwords while allowing memorable ones.

3.12 Password Selection Techniques

Four basic techniques are used:

1. User education.
2. Computer-generated passwords.
3. Reactive password checking.
4. Complex password policy (Proactive password checker).

3.13 Computer-Generated Passwords

If random, these are hard to remember. Even pronounceable ones can be difficult, tempting users to write them down. FIPS-181 defines a generator using pronounceable syllables.

3.14 Reactive Password Checking

The system periodically runs its own password cracker against stored passwords, canceling and notifying users of guessable ones. Drawbacks include resource intensiveness and passwords remaining vulnerable until found.

3.15 Complex Password Policy (Proactive Checker)

The system checks a user-selected password against a policy (e.g., minimum length, character types, dictionary lists using filters like Bloom filters) and rejects it if non-compliant.

3.16 Current Password Advice

Recent guidance on password policies, such as NIST 800-63, often emphasize memorability over strict complexity rules, combined with other factors like salting and 2FA.

4 Access Control

4.1 What is Access Control?

NIST IR 7298 defines access control as granting or denying specific requests to obtain/use information/processing services and enter physical facilities. RFC 4949 defines it as regulating the use of system resources according to a security policy, permitting access only by authorized entities (users, programs, processes) based on that policy.

4.2 Relationship Among Access Control and Other Security Functions

Access control is closely related to other security functions, including auditing, which reviews records and activities to check control adequacy and ensure compliance.

4.3 Access Control Policies

Access control models are based on policies that define how access is granted or denied:

- **Discretionary Access Control (DAC):** Based on the identity of the requestor and access rules defined by object owners.
- **Mandatory Access Control (MAC):** Based on comparing security labels (object classifications) with security clearances (subject levels). Access decisions are system-wide, overriding DAC. A key characteristic is that one with access cannot pass it to others if it violates policy.
- **Role-Based Access Control (RBAC):** Access decisions are based on the roles users hold within an organization. Users are assigned to roles, and permissions are assigned to roles.
- **Attribute-Based Access Control (ABAC):** Access decisions are based on attributes of the subject, object, and environment.

Policies can be based on authorizing all accesses except those prohibited, or prohibiting all accesses except those authorized. Administrative policies define who can add, delete, or modify access rules.

4.4 Access Control Elements

Access control involves three key elements:

- **Subject:** An active entity (e.g., a process representing a user/application) that accesses objects. Subjects often fall into classes like owner, group, and world.
- **Object:** A passive, access-controlled resource (e.g., files, directories, records, programs). The number and type of objects depend on the environment.
- **Access right:** The way in which a subject accesses an object (e.g., read, write, execute, delete, create, search).

4.5 Discretionary Access Control (DAC)

DAC is often represented using an access matrix, listing subjects in rows and objects in columns, with entries specifying access rights. The matrix can be sparse. It can be decomposed by column to form Access Control Lists (ACLs) or by row to form capability tickets. An alternative is a non-sparse table representation.

4.6 UNIX Access Control

Traditional UNIX access control uses owner, group, and others permissions (read, write, execute - rwx). Setuid and Setgid bits allow processes to run with the permissions of the file owner or group. The sticky bit on a directory limits rename/move/delete operations to the owner. The superuser (root) is exempt from usual access control restrictions.

4.7 UNIX Access Control Lists (ACLs)

Modern UNIX systems support ACLs, allowing specification of any number of additional users/groups and their rwx permissions. When access is required, the system selects the most appropriate ACL (owner, named users, owning/named groups, others) and checks for sufficient permissions.

4.8 Attribute-Based Access Control (ABAC)

ABAC makes access decisions based on attributes of the subject, object, and environment (context). It is highly flexible and expressive. There is considerable interest in applying ABAC to cloud services. Decisions can be based on rules like "Subjects with attribute X can perform action Y on objects with attribute Z if environmental condition W is true."

4.9 Types of Attributes (ABAC)

Attributes used in ABAC include:

- Subject attributes: Define identity and characteristics of the active entity (e.g., Name, Organization, Job title).
- Object attributes: Describe the resource being accessed (e.g., Title, Author, Date).
- Environment attributes: Describe the operational or situational context (e.g., Current date, current virus activity, Network security level). These are not associated with a specific resource or subject.

5 Database Security

Database systems are structured collections of data, often containing sensitive information requiring security. They provide a uniform interface through query languages.

5.1 Relational Databases

Relational databases are constructed from tables with columns (attributes/fields) holding data types and rows (tuples/records) containing specific values. Tables have a primary key (uniquely identifies a row) and foreign keys (linking to attributes in other tables). Views or virtual tables are results of queries selecting rows/columns from tables.

5.2 Structured Query Language (SQL)

SQL is a standardized language for defining, manipulating, and querying data in relational databases. Examples include 'CREATE TABLE' statements to define schema.

5.3 SQL Injection (SQLi)

SQL injection is an attack where an attacker inserts malicious SQL code into a database query. It typically works by prematurely terminating a text string input and appending a new command. Examples include:

- **Tautology:** Injecting code in conditional statements so they always evaluate to true, bypassing authentication (e.g., 'password = '...' OR 1=1 –').
- **End-of-line comment:** Injecting code and then using comments ('–') to nullify the legitimate code that follows in the original query.
- **Piggybacked queries:** Adding additional queries (e.g., 'DROP TABLE ...') after the intended query.

5.4 Inferential Attack (SQLi)

An inferential attack does not involve direct data transfer but allows the attacker to reconstruct information about the database (type, structure) by sending specific queries and observing the web application's/database server's behavior or error messages.

5.5 Out-band Attack (SQLi)

Data is retrieved through a different channel (e.g., email results of a query). This is used when information retrieval is limited but outbound connectivity from the database server is lax.

5.6 SQLi Countermeasures

Countermeasures include defensive coding (stronger data validation, using parameterized queries), detection (signature-based, anomaly-based, code analysis), and inference detection (at database design or query time by monitoring/altering/rejecting queries).

5.7 Statistical Databases

Statistical databases provide data of a statistical nature (counts, averages). They can be pure statistical or ordinary databases with statistical access enabled for some users. The security problem is inference: allowing statistical use without revealing individual entries.

5.8 Inference in Statistical Databases

Inference attacks aim to deduce sensitive information about individuals from statistical query results.

5.9 Perturbation (Statistical Databases)

Perturbation involves adding noise to statistics generated from data to prevent inference. Techniques include data perturbation (swapping data, generating statistics from probability distributions) and output perturbation (random-sample query, statistic adjustment). The goal is to minimize loss of accuracy while preventing inference.

5.10 Database Encryption

Databases are valuable information resources protected by multiple layers (firewalls, authentication, OS access control, DB access control) including database encryption. Encryption can be applied to the entire database (inflexible/inefficient), individual fields (simple but inflexible), or records/columns (often considered best). Attribute indexes are needed for data retrieval.

5.11 Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (NIST SP-800-145). Resources won't be a permanent part of the client's IT infrastructure.

5.12 Essential Characteristics of Cloud

Key characteristics include on-demand self-service (consumers provision capabilities unilaterally), resource pooling (resources shared and assigned dynamically), and measured service (control and optimize resource use by leveraging a metering capability).

5.13 Cloud Security Risks

Cloud computing introduces security risks, including:

- **Abuse and Nefarious Use:** Attackers can use cloud resources for malicious activities (spamming, DoS).
- **Insecure Interfaces and APIs:** Cloud providers expose APIs for customers; their security is critical.
- **Malicious Insiders:** Insiders at the cloud provider or client organization pose a risk.
- **Data Loss or Leakage:** Data stored in the cloud could be lost or leaked.

5.14 Cloud Security Countermeasures

Countermeasures for cloud security risks include:

- Abuse: Stricter/restrict registration, enhanced credit card monitoring, comprehensive traffic monitoring.
- Insecure Interfaces: Analyzing API security, ensuring strong authentication and access control, understanding dependency chains.
- Malicious Insiders: Implementing best security practices, monitoring for unauthorized activity, promoting strong authentication and access control.
- Data Loss: Implementing strong API access control, encrypting/protecting integrity of data in transit, analyzing data protection at design and runtime.

6 Malicious Software and DoS

6.1 Malicious Software (Malware) Definition

Malware is a program inserted into a system, usually covertly, with the intent of compromising confidentiality, integrity, or availability, or otherwise disrupting the victim. It exploits system vulnerabilities. Malware can be program fragments requiring a host (viruses, logic bombs, backdoors) or independent self-contained programs (worms, flooders, DDoS zombies, bots).

6.2 Viruses

A virus is a piece of software that infects programs by modifying them to include a copy of the virus code. It executes secretly when the host program is run. Viruses are often specific to OS/hardware details. A typical virus has phases: dormant, propagation (copies itself), triggering (activated), and execution (performs its function). Virus structure includes an infection mechanism, trigger, and payload. Countermeasures aim to block initial infection (difficult) or propagation (using access controls).

6.3 Virus Classification

Viruses can be classified by target (boot sector, file infector, macro virus, multipartite) or by concealment (encrypted virus, stealth virus, polymorphic virus - recreates with different signature, metamorphic virus - recreates with different signature and behavior). The Michelangelo Virus is an example of a memory resident boot sector infector.

6.4 Macro and Scripting Viruses

These became common in the mid-1990s because they are platform independent, infect documents, and spread easily, exploiting the macro capability of applications like Microsoft Office. The Melissa virus is an example, exploiting an MS Word macro to spread via email address lists. More recent software releases include protection, and many anti-virus programs recognize them.

6.5 Virus Countermeasures

Countermeasures include policy (prohibiting infected media, checking all media), detection (scanning, behavior monitoring), identification (finding the specific virus), and removal. If a virus cannot be removed, the infected program must be discarded and replaced.

6.6 Anti-virus Evolution

Anti-virus technology has evolved with viruses. Early signature scanners matched bit patterns. Later generations added heuristics (integrity checks), identified malicious actions, and combined techniques.

6.7 Generic Decryption

Generic decryption runs executable files through a scanner with a CPU emulator and virus scanner. The emulator lets the virus decrypt itself in a controlled environment, allowing the scanner to check for known signatures periodically.

6.8 Digital Immune System

A system (like the IBM/Symantec project mentioned) where a monitoring program infers a virus, sends a copy to an admin machine, which sends it encrypted to a central analysis machine. The central analysis safely executes the virus, analyzes it, provides a "prescription" (removal instructions), which is sent back to admin machines and clients, and forwarded to other organizations, with subscribers receiving regular updates.

6.9 Behavior-Blocking Software

This software integrates with the OS and looks for malicious behaviors, such as attempts to open/view/delete/modify files, format drives, modify executables/settings, or script emails to send executive contents.

6.10 Worms

A worm is a replicating program that propagates over a network using mechanisms like email, remote execution, or remote login. Like viruses, worms have dormant, propagation, triggering, and execution phases. The propagation phase actively searches for other systems, connects, and copies itself. Examples of propagation mechanisms include scanning (looking for open services like with Code Red), email (mass mailers like Love Bug), file sharing, and exploiting remote execution/login vulnerabilities.

6.11 Mobile Code

Mobile code refers to scripts, macros, or other portable instructions (JavaScript, ActiveX, VBScript) transmitted from a remote to a local system. It can act as an agent for viruses, worms, and Trojan horses. Mobile phone worms propagate via Bluetooth connections.

6.12 Social Engineering and Trojan Horses

Social engineering techniques are often used in conjunction with malware. Spam (less effective now due to improved filters) is a delivery method. A Trojan horse looks like a useful tool but contains hidden malicious code. Phishing attacks often use social engineering and potentially malicious links or attachments. Recognizing red flags in emails (sender name/domain, spelling/grammar, generic greetings, requests for personal info) helps identify phishing.

6.13 Payloads

Malware payloads can cause data destruction or theft, data encryption (ransomware), or even real-world physical damage (Stuxnet targeted industrial control software). A logic bomb is malicious code triggered by a specific event or time. A backdoor is often inserted by programmers to bypass normal authentication.

6.14 Rootkit

A rootkit modifies system tables or calls (e.g., in the OS kernel) to hide its presence or redirect system calls, allowing the attacker to maintain persistent, covert access.

6.15 Malware Countermeasures Summary

Malware countermeasures include prevention, detection, identification, and removal. Key requirements for countermeasures are generality (work against various malware), timeliness (respond quickly), resiliency (withstand attacks), minimal DoS costs, transparency (not disrupt legitimate use), and global/local coverage.

7 Denial of Service (DoS)

Denial of Service (DoS) is an attempt to prevent legitimate users from accessing system resources. It is a classic attack category.

7.1 DoS Attack Goal and Impact

The goal is to exhaust system resources like bandwidth, CPU, memory, disk space, network connections, or application processes. The impact can be disruption, financial loss, and reputational damage. Attacks can either crash the system or flood it with requests.

7.2 Distributed DoS (DDoS)

While a single source DoS attack has limited volume, a Distributed DoS (DDoS) attack uses multiple systems to generate much higher traffic volumes. These often involve compromised PCs or workstations (zombies) controlled as part of a botnet. Examples include Tribe Flood Network (TFN/TFN2K) performing various types of floods (ICMP, SYN, UDP). DDoS attacks involve a control hierarchy where an attacker commands handler zombies, which then command other handlers or agents.

7.3 SYN Flood Attack

A SYN flood is a DoS attack type that exploits the TCP three-way handshake by sending a large number of SYN requests without completing the handshake, exhausting the server's connection resources.

7.4 Random Source Attack

An attacker sends random packets with spoofed source addresses to a target. Devices may reply to the spoofed address, contributing to the DoS and making it difficult to identify the actual source after the incident.

7.5 Application-based Bandwidth Attacks

These attacks force the victim system to execute resource-consuming operations, like complex searches or database queries, exhausting application resources.

7.6 Slowloris Attack

Slowloris is an application-layer DoS attack that exploits the slow transmission of HTTP headers. It exhausts the server's available connections and capacity. It uses legitimate HTTP traffic and may not be recognized by existing signature-based IDS/IPS. Spidering (bots recursively following web links) can also contribute to application-based resource exhaustion.

7.7 Reflection Attacks (DoS)

In a reflection attack, the attacker sends packets to a known service on an intermediary system, but with the source IP address spoofed to be the target's address. The intermediary then sends its response to the target, amplifying the attack traffic directed at the target.

8 Intrusion Detection

8.1 Classes of Intruders

Intruders can be categorized, including criminals motivated by financial reward through activities like identity theft, credential theft, corporate espionage, data theft, and data ransoming. These are often young hackers operating in underground forums.

8.2 Intrusion Techniques

The objective of intrusion techniques is typically to gain unauthorized access or increase privileges. Initial attacks often exploit system or software vulnerabilities (e.g., buffer overflows) to execute code, install backdoors, or gain protected information through password guessing, acquisition, or social engineering.

8.3 Intrusion Detection Systems (IDS)

IDS are systems designed to detect unauthorized access or malicious activity. Types include:

- **Host-based IDS:** Monitor activity on a single host system.
- **Network-based IDS (NIDS):** Monitor network traffic.
- **Distributed or hybrid IDS:** Combine information from multiple sensors (host and network based) in a central analyzer to improve detection and response.

8.4 IDS Principles

IDS operate on the assumption that intruder behavior differs from legitimate user behavior. There is expected overlap between the two. IDS observe deviations from past history but face problems of false positives (valid users identified as intruders) and false negatives (intruders not identified). A compromise is often needed between loose (catching more, more false positives) and tight (catching less, more false negatives) interpretations.

8.5 IDS Requirements

Effective IDS should: run continually with minimal human supervision, be fault tolerant, resist subversion (monitor themselves), impose minimal system overhead, be configurable according to security policies, adapt to system/user changes, scale to monitor large numbers of systems, provide graceful degradation if a component fails, and allow dynamic reconfiguration.

8.6 Detection Techniques

IDS use different techniques to detect suspicious activity:

- **Anomaly (behavior) detection:** Collects data on legitimate user behavior over time and analyzes current behavior for deviations. Threshold detection is a simple form, checking excessive event occurrences, but can be crude.
- **Signature/heuristic detection:** Uses a set of known malicious data patterns or attack rules (signatures) and compares them with current behavior. Also known as misuse detection. It can only identify known attacks for which it has signatures, similar to anti-virus, and requires frequent updates. Rule-based penetration identification analyzes attack scripts to identify known penetrations or weaknesses.

8.7 Signature Detection IDS Rules Example

Examples of rules used in signature detection include checking for users logged into multiple sessions, copying password files, reading/writing other users' files, accessing files after hours, or directly opening disk devices. SNORT rules are an example of a language for defining signatures.

8.8 Host-based IDS

Host-based IDS are specialized software monitoring system activity to detect suspicious behavior. Their primary purpose is to detect intrusions, log events, and send alerts for both external and internal intrusions. Approaches include anomaly detection (comparing to normal behavior) and signature analysis (checking for known patterns/rules). Common data sources for HIDS include system call traces, audit (log file) records, file integrity checksums, and registry access.

8.9 Distributed Host-based IDS

A distributed HIDS can involve host agents, LAN agents (analyzing LAN traffic), and a central manager. Host agents retain security data in a standard format, analyze for failed access or changes, and send suspicious activity to the central manager.

8.10 Distributed Hybrid Intrusion Detection

This approach combines host-based, NIDS, and distributed host-based elements. Solutions might involve distributed adaptive IDS through peer-to-peer cooperation. Challenges include tools not recognizing new threats and difficulty dealing with rapidly spreading attacks.

8.11 Logging of Alerts

Logging alerts is crucial for all IDS types. A NIDS sensor typically logs information such as timestamp, session ID, event/alert type, rating, protocols, IP addresses, ports, bytes transmitted, decoded payload data, and state information.

8.12 Intrusion Detection Exchange Format

An Intrusion Detection Exchange Format is a proposed standard to facilitate the development of distributed IDS. It defines elements like data sources, sensors, analyzers, administrators, managers, and operators.

8.13 Honeypots

Honeypots are decoy systems filled with fabricated information and instrumented with monitors and loggers. They lure potential attackers away from critical systems, collect information about attacker activity, and encourage the attacker to stay long enough for administrators to respond. They divert and hold attackers to collect info without exposing production systems. Initially single systems, they now often emulate entire networks.

8.14 Honeypot Classification

- **Low interaction honeypot:** A software package that emulates particular IT services enough for initial interaction but doesn't run full versions. Provides a less realistic target, suitable as a component of a distributed IDS.
- **High interaction honeypot:** Complex systems running real operating systems and applications. Provide a more realistic target but are riskier.

8.15 SNORT

SNORT is an intrusion detection system that performs real-time packet capture and rule analysis. It can operate passively or inline. Its components include a decoder, detector, logger, and alerter.

8.16 SNORT Rules

SNORT uses a simple, flexible rule definition language with a fixed header (action, protocol, source/dest IP/port, direction) and options. Example rules detect specific attack patterns like a TCP SYN-FIN scan.

9 Firewalls and Intrusion Prevention Systems

10 Buffer Overflow

11 Software Security

12 Operating System Security

13 Trusted Computing and Multilevel Security

13.1 Computer Security Models

Two fundamental facts in computer security highlight the difficulty in building truly secure systems: all complex software systems eventually reveal flaws, and it is extraordinarily difficult to build hardware/software not vulnerable to attacks.

13.2 Confidentiality Policy (Bell-LaPadula Model)

The primary goal of a confidentiality policy is to prevent the unauthorized disclosure of information. This deals primarily with information flow, with integrity being an incidental concern.

13.3 Multilevel Security (MLS)

MLS systems involve multiple levels of security and data classification. A key principle is that a subject at a high security level may not convey information to a subject at a non-comparable level. This is enforced through specific properties:

- **No read up (ss-property):** A subject can only read an object of less than or equal security level.
- **No write down (*-property):** A subject can only write into an object of greater than or equal security level.

13.4 BLP Formal Description and Properties

The BLP model is based on the current state of the system, which includes the current access set (b), the access matrix (M), a level function (f) assigning security levels to subjects and objects, and a hierarchy (H) for objects. The three core BLP properties based on the current state (c) are:

- **ss-property:** For a read access (S_i, O_j, read), the security level of the subject must dominate the security level of the object: $f_c(S_i) \geq f_o(O_j)$.
- ***-property:** For an append access $(S_i, O_j, \text{append})$, $f_c(S_i) \leq f_o(O_j)$. For a write access (S_i, O_j, write), $f_c(S_i) = f_o(O_j)$.
- **ds-property:** For any access (S_i, O_j, A_x), the access mode A_x must be permitted in the access matrix: $A_x \in M[S_i, O_j]$.

BLP provides formal theorems allowing for the theoretical possibility of proving a system is secure according to the model.

13.5 BLP Operations

The model defines operations that change the system state while maintaining security properties. These include:

- get access: add $(subj, obj, access - mode)$ to b to initiate access.
- release access: remove $(subj, obj, access - mode)$ from b.
- change object level.
- change current level (for a subject).
- give access permission: Add an access mode to M for a subject to grant access.
- rescind access permission: reverse of give access permission.
- create an object.
- delete a group of objects.

13.6 BLP Example and Categories

An example involving a student (Carla) and a teacher (Dirk) in a role-based system illustrates BLP. Dirk creates an exam (f4) at a high level. He cannot directly give Carla (student, lower level) read access due to the ss-property (no read up). An administrator with higher privileges must downgrade the object's security level to allow Carla access. Carla writing answers to a file (f5) can be an example of writing up if her answer file is at a lower level than the teacher's object she is writing into, which is permitted by the *-property.

The BLP model can be expanded by adding categories to each security classification, based on the "need to know" principle. Objects can be placed in multiple categories (e.g., NUC, EUR, US). Categories form a lattice structure based on the "subset of" operation.

13.7 BLP Dominate Relationship

The dominate (dom) relationship in BLP captures the combination of security classification (level) and category set. A pair (A, C) dominates (A', C') if and only if $A' \leq A$ and $C' \subseteq C$, where A is the security classification level and C is the set of categories. Examples illustrate this relationship between security levels and category sets.

13.8 Reading and Writing Information Rules

Based on BLP, information flows up, not down. "Reads up" are disallowed, while "reads down" are allowed. The *-property is sometimes called the "no writes down" rule. The BLP model combines mandatory access control (relationship of security levels) and discretionary access control (required permissions).

13.9 Trusted Systems and Trusted Computing

A trusted system is one believed to enforce a given set of attributes to a stated degree of assurance. Trustworthiness is the assurance that a system deserves to be trusted, guaranteed through means like formal analysis or code review. A trusted computer system uses sufficient hardware and software assurance measures to allow simultaneous processing of sensitive or classified information.

Reference Monitors are a core concept in trusted systems, enforcing security rules on every access (complete mediation), protected from unauthorized access (isolation), and whose correctness can be proven (verifiability).

13.10 Key Trusted Computing Concepts

For a fully trusted system, six key technology concepts are required:

1. Endorsement key
2. Secure input and output
3. Memory curtaining / protected execution
4. Sealed storage
5. Remote attestation
6. Trusted Third Party (TTP)

13.11 Trusted Third Party (TTP)

A TTP is an entity that facilitates interactions between two parties who trust the third party. The TTP reviews critical transaction communications to help secure interactions, addressing issues like the ease of creating fraudulent digital content.

13.12 Trusted Platform Module (TPM)

The TPM, a concept from the Trusted Computing Group, is a hardware module central to hardware/software approaches to trusted computing.

13.13 TPM Functions

Key functions of a TPM include:

- A hardware random number generator.
- Secure generation of cryptographic keys for limited uses.
- **Remote attestation:** Creates a nearly unforgeable hash summary of hardware and software configuration to verify they haven't changed.
- **Binding:** Encrypts data using a unique RSA key (TPM bind key), descended from a storage key, so only that TPM can decrypt it. This process (wrapping or binding) protects keys from disclosure. Each TPM has a master wrapping key, the storage root key, stored within the TPM.
- **Sealed storage:** Specifies the necessary TPM state for data to be decrypted (unsealed).

13.14 Secure Boot

Secure boot is achieved using a hardware root of trust boot mechanism. This encrypts boot or configuration files, providing confidentiality, integrity, and authentication for secure applications. An example architecture uses RSA-4096 asymmetric authentication with SHA-3/384, employing primary (PPK) and secondary (SPK) public key pairs.

13.15 Trusted Systems and Evaluation Criteria

Work on security models for enhancing trust began in the early 1970s. This led to the Trusted Computer System Evaluation Criteria (TCSEC), known as the Orange Book, in the early 1980s. Further international work resulted in the Common Criteria in the late 1990s. The NSA's Computer Security Center also had a Commercial Product Evaluation Program for evaluating commercial products, required for Defense use.

13.16 Common Criteria (CC)

Common Criteria are ISO standards for security requirements and evaluation criteria. They aim to provide greater confidence in IT product security through formal actions during development, evaluation, and operation. Evaluated products are listed for use.

13.17 CC Requirements

CC requirements include a common set of potential security requirements for evaluation. The Target of Evaluation (TOE) is the product/system being evaluated. Requirements are categorized into:

- Functional requirements: Define desired security behavior.
- Assurance requirements: Ensure security measures are effective and correct.

13.18 Assurance

Assurance is the degree of confidence that security controls operate correctly and protect the system as intended. It applies to product security requirements, security policy, product design, implementation, and operation, utilizing various analysis, checking, and testing approaches.

13.19 Common Criteria Assurance Levels (EAL)

CC defines Evaluation Assurance Levels (EALs) from 1 to 7:

- **EAL 1:** functionally independently tested.
- **EAL 2:** structurally tested (includes design review and vulnerability analysis).
- **EAL 3:** methodically tested and checked (design testing).
- **EAL 4:** methodically designed, tested, and reviewed (high to low level vulnerability analysis).
- **EAL 5:** semiformally designed and tested. **EAL 6:** semiformally verified design and tested.
- **EAL 7:** formally verified design and tested (formal analysis and showing correspondence).

13.20 Evaluation Parties and Phases

CC evaluations involve:

- Parties: Sponsor (customer/vendor), Developer (provides evidence), Evaluator (confirms requirements), Certifier (agency monitoring evaluation).
- Phases: Preparation (initial contact), Conduct of evaluation (structured process), Conclusion (final evaluation).

Government agencies like NIST and NSA jointly operate evaluation and validation schemes (e.g., US CCEVS).

13.21 Cryptographic Module Validation Program (CMVP)

NIST issued FIPS 140 to coordinate requirements for cryptography modules (hardware and software). The CMVP, a joint effort by NIST and the Canadian Centre for Cyber Security, validates cryptographic modules to FIPS 140-3.

14 Security Management and Risk Assessment

14.1 Overview

Determining security requirements involves asking: what assets need protection, how are they threatened, and what countermeasures are effective? IT security management addresses these questions by determining security objectives, risk profile, performing risk assessment, and selecting, implementing, and monitoring controls.

14.2 IT Security Management

IT security management is a process for achieving and maintaining appropriate levels of confidentiality, integrity, availability, accountability, authenticity, and reliability. Its functions include defining objectives/strategies/policies, determining requirements, identifying/analyzing threats and risks, specifying safeguards, monitoring safeguards, developing awareness programs, and detecting/reacting to incidents.

14.3 ISO 27000 Security Standards

The ISO 27000 family of standards provides guidance on information security management.

- ISO 27000: Defines vocabulary and definitions.
- ISO 27001: Defines the Information Security Management System (ISMS) specification and requirements for certification.
- ISO 27002: Specifies a code of practice with a comprehensive set of information security control objectives and best-practice controls.
- ISO 27003: Implementation guidance.
- ISO 27004: Information security management measurement.
- ISO 27005: Information security risk management.
- ISO 13335: Provides guidance on IT security management concepts and operations.

14.4 IT Security Management Process (Plan-Do-Check-Act)

The IT security management process often follows the Deming Cycle:

- Plan: Establish policy, define objectives and processes.
- Do: Implement and operate policy, controls, processes.
- Check: Assess, measure, and report results (often based on audits).
- Act: Take corrective and preventative actions.

14.5 Organizational Context and Security Policy

The process begins by examining the organization's IT security objectives, strategies, and policies. These must be maintained and updated regularly through security reviews to reflect changing technical and risk environments. In large organizations, IT security officers may manage the process within their areas.

14.6 Security Risk Assessment

Risk assessment is a critical process component. Ideally, every asset versus risk would be examined, but this is often not feasible. Organizations choose from several alternatives based on resources and risk profile:

- Baseline
- Informal
- Formal (Detailed)
- Combined

14.7 Baseline Approach

This approach uses "industry best practice" recommendations. It is easy and cheap but gives no special consideration to the organization's specific context, potentially resulting in too much or too little security. Safeguards are implemented against the most common threats based on available checklist documents. This approach alone is suitable only for small organizations.

14.8 Informal Approach

This involves conducting a pragmatic risk analysis using the knowledge and expertise of an analyst. It is fairly quick and cheap and addresses some organization-specific issues. However, risks may be assessed incorrectly, and the results can be skewed by the analyst's views, varying over time. It is suitable for small to medium-sized organizations.

14.9 Detailed Risk Analysis (Formal Approach)

This is the most comprehensive alternative, using a formal structured process with multiple stages. It identifies the likelihood and consequences of risks, providing confidence that controls are appropriate. This approach is costly and slow, requiring expert analysts. It may be a legal requirement in some cases and is suitable for large organizations where IT systems are critical to business objectives.

14.10 Combined Approach

This approach combines elements of the others, starting with an initial baseline, using informal analysis to identify critical risks, and then performing a formal assessment on those critical systems. This process is iterated and extended over time.

14.11 Threat Sources

Threats can be natural ("acts of god") or man-made (accidental or deliberate). When considering human attackers, factors like motivation, capability, resources, probability of attack, and deterrence should be considered, along with any previous attack history on the organization.

14.12 Threat Identification

Threat identification depends on the risk assessor's experience and uses various sources like insurance statistics for natural threats, lists from standards, security surveys, and government information, tailored to the organization's environment and system vulnerabilities.

14.13 Determine Likelihood and Consequence

Risk assessment involves determining the likelihood of a threat exploiting a vulnerability and the consequence or impact of that event. Likelihood can be rated on a scale (e.g., Unlikely, Possible, Likely, Almost Certain). Consequence is rated based on the severity of impact (e.g., Insignificant).

14.14 Risk Level Determination

The likelihood and consequence ratings are combined (often using a matrix) to determine the risk level (e.g., Extreme, High, Medium, Low). Each level has a description outlining the required management attention and response.

14.15 Document in Risk Register

Identified risks are documented in a risk register, which typically includes columns for the asset, threat/vulnerability, existing controls, likelihood, consequence, level of risk, and a prioritized list of risks.

14.16 Risk Treatment Alternatives

Once risks are identified and evaluated, treatment alternatives are considered:

- **Risk acceptance:** Accept the risk, possibly due to excessive treatment costs.
- **Risk avoidance:** Do not proceed with the activity causing the risk (potentially involving loss of convenience). **Risk transfer:** Shift the risk, e.g., by buying insurance or outsourcing.
- **Reduce consequence:** Modify asset use to reduce impact (e.g., offsite backup).
- **Reduce likelihood:** Implement suitable controls to decrease the chance of occurrence.

14.17 Case Study: Silver Star Mines

A fictional case study of a global mining company illustrates the risk assessment process. The company, with a large and increasingly networked IT infrastructure, decided on a combined risk assessment approach. The mining industry is considered less risky on the spectrum, and management accepts moderate or low risk levels. The study lists key assets (SCADA nodes, data integrity, financial, procurement, maintenance, mail systems) and threats/vulnerabilities (unauthorized modification, corruption, theft, attacks/errors). A sample risk register entry shows how threats to assets are evaluated based on existing controls, likelihood, and consequence.

14.18 Summary

The sources emphasize the detailed need to perform risk assessment as a core part of the IT security management process, highlighting relevant security standards, presenting risk assessment alternatives, and detailing the process steps from context definition and asset identification to risk analysis and evaluation.

15 Security Controls and Plans

15.1 FISMA and NIST

The Federal Information Security Management Act (FISMA) is a US law requiring federal organizations and contractors to implement information security protections commensurate with risk. FISMA requires the National Institute of Standards and Technology (NIST) to develop standards and guidelines to help implement the act and improve federal information system security.

15.2 NIST Publications

NIST security publications are available online, providing guidance on various aspects of security.

15.3 Directives and NIST

Entities like the Office of Management and Budget (OMB) issue directives (Memos, Circulars) that may mandate the use of NIST guidance. Executive Orders (EOs) and Presidential Directives (PDs) from the Executive Office of the President may also direct NIST to provide guidance or develop standards, such as Homeland Security Presidential Directives (HSPDs).

15.4 Joint Task Force Transformation Initiative

This initiative is a partnership including NIST, the Department of Defense, the Intelligence Community (including ODNI and 16 US Intelligence Agencies), and the Committee on National Security Systems, aimed at transforming security guidance.

15.5 Standards/Guidelines for FISMA and RM

Key Federal Information Processing Standards (FIPS) and Special Publications (SPs) support FISMA and risk management:

- FIPS 199: Standards for Security Categorization of Federal Information and Information Systems.
- FIPS 200: Minimum Security Requirements.
- SP 800-18: Guide for System Security Plan development.
- SP 800-30: Guide for Conducting Risk Assessments.
- SP 800-34: Guide for Contingency Plan development.
- SP 800-37: Guide for Applying the Risk Management Framework.
- SP 800-39: Managing Information Security Risk.
- SP 800-53/53A: Security controls catalog/assessment procedures.
- SP 800-60: Mapping Information Types to Security Categories.
- SP 800-128: Security-focused Configuration Management.
- SP 800-137: Information Security Continuous Monitoring.
- Many others for operational and technical implementations.

15.6 NIST SP 800-37 (Risk Management Framework - RMF)

SP 800-37 provides a guide for applying the RMF to federal information systems, representing a holistic risk management process integrated into the System Development Life Cycle (SDLC). It outlines tasks for each of the six steps in the RMF at the system level.

15.7 RMF Step 1: Categorize

This step is supported by FIPS 199 and SP 800-60. FIPS 199 defines security categorization based on the security objectives of confidentiality, integrity, and availability, outlining three impact levels:

- Low: loss would have a limited adverse impact.
- Moderate: loss would have a serious adverse impact.
- High: loss would have a catastrophic adverse impact.

SP 800-60 provides guidance and catalogs information types with provisional categorizations.

15.8 NIST SP 800-18 (Guide for Developing Security Plans)

SP 800-18 provides guidance for developing a System Security Plan (SSP), including its structure and content, and offers a template. It supports all RMF steps but begins during Step 1, used to record information about the system, including its boundary, roles, responsibilities, and control implementation details.

15.9 RMF Step 2: Select

This step involves selecting security controls. FIPS 200 defines minimum security requirements across 17 security-related areas (families) representing a broad, balanced security program, including management, operational, and technical controls. It specifies implementing a minimum baseline of controls defined in NIST SP 800-53, which should be appropriately tailored.

15.10 NIST SP 800-53 (Security and Privacy Controls Catalog)

SP 800-53 provides a comprehensive catalog of security controls. It supports Step 2 (Select) of the RMF and defines three security baselines corresponding to the Low, Moderate, and High impact levels.

15.11 Security Controls Definition

Security controls are the safeguards or countermeasures prescribed for an information system to protect its confidentiality, integrity, and availability, as well as its information.

15.12 Types of Controls (SP 800-53)

SP 800-53 defines three types of controls:

- Common controls: Inherited from enterprise-wide systems.
- System specific controls: Unique to a particular system.
- Hybrid controls: Combination of common and system specific.

15.13 SP 800-53 Baselines

Baselines are defined in SP 800-53 (Appendix D, Table D-2) and are determined by the information/system categorization (L, M, H), organizational risk assessment/tolerance, and system-level risk assessment. Baselines are a starting point and should be tailored to fit the mission and system environment through parameters, scoping/compensating controls, and supplementing.

15.14 Compensating Control

A compensating control is a security control employed in lieu of recommended controls in the baselines that provides equivalent or comparable protection. An example is requiring encryption for all information stored in the cloud as a compensating control for cloud-based systems.

15.15 Availability of Controls Not in Baselines

SP 800-53 provides a comprehensive set of controls, but not every system needs to implement every control, consistent with risk management. Controls and enhancements not selected in a baseline are available as compensating or supplemental controls to strengthen protection based on system risk, organizational risk tolerance, and overlay requirements for specific communities.

15.16 RMF Step 3: Implement

Step 3 involves implementing the selected controls. Guidance is available in SP 800-37R1 tasks and numerous other publications from csrc.nist.gov. Automated tools can implement specific controls. Control implementation should be planned during the SDLC development phase ("bake it in").

15.17 Security Technical Implementation Guides (STIGs)

STIGs contain requirements flagged as applicable for a product based on a selected baseline. They automate the application of many controls for covered products.

15.18 Control Correlation Identifier (CCI)

The CCI provides a standard identifier and description for the singular, actionable, measurable statements that comprise an IA control or best practice. CCI bridges the gap between high-level policy and low-level technical implementations.

15.19 Security Control Traceability Matrix (SCTM)

An SCTM helps identify gaps in security controls, tracks implementation over time, ensures compliance, and provides a comprehensive view of a system's security posture.

16 Physical and Infrastructure Security

16.1 Importance

The significance of physical security is often underestimated. Breaches in physical security can occur without technical background. Preparation for accidents and natural disasters is also necessary as they are inevitable.

16.2 Physical Security Definition

Physical security, also called infrastructure security, protects the information systems containing data and the people who use, operate, and maintain them. It must prevent any physical access or intrusion that could compromise logical security.

16.3 Premises Security Definition

Premises security, also known as corporate or facilities security, protects people and property within an entire area, facility, or building(s). It is usually required by laws, regulations, and fiduciary obligations. It provides perimeter security, access control, fire detection/suppression, environmental protection, surveillance systems, alarms, and guards.

16.4 Protective Layers

Physical security employs layers of protection:

- Middle Protective Layers: Involve the structure (door/window/ceiling penetration controls, ventilation ducts, elevator penthouses) and environment (positive controls within the perimeter).
- Inner Protective Layers: Focus on access controls like keypads (vulnerable to keypad wear, shoulder surfing; countermeasures include randomizing numbers), mechanical and electronic access control.

16.5 Preventing Tailgating/Piggybacking

Tailgating or piggybacking occurs when an unauthorized person follows an authorized person through an access point after the door is unlocked, often using social engineering. Turnstiles and secure revolving doors are designed to prevent this by enforcing a one-person authentication rule.

16.6 Technical Threats (Physical)

Technical threats include issues with electrical power (utility problems like under-voltage, over-voltage, noise), fire and smoke (requiring alarms, preventative measures, mitigation), and water (managing lines, equipment location, cutoff sensors). Other threats involve limiting dust entry and pest control.

16.7 Mitigation Measures for Technical Threats

Mitigation for electrical power issues for critical equipment includes using Uninterruptible Power Supplies (UPS) and emergency power generators. Electromagnetic interference (EMI) can be mitigated with filters and shielding.

16.8 Mitigation Measures for Human-Caused Threats

Mitigation for human-caused threats involves physical access control for IT equipment, wiring, power, communications, and media. A spectrum of approaches can be used, from restricting building access to locked areas, secured power switches, and tracking devices. Intruder sensors and alarms are also necessary.

16.9 Recovery from Physical Security Breaches

Recovery requires redundancy to address data loss, ideally off-site and updated frequently (batch encrypted remote backup or remote hot-site with live data). Recovery from physical equipment damage depends on the nature of the damage and cleanup, potentially requiring disaster recovery specialists.

16.10 Disaster Recovery: Backup Facilities

Different types of backup facilities support disaster recovery:

- Hot sites: Ready to run immediately, high cost.
- Cold sites: Building facilities, power, and communications available, but no computing resources installed.
- Site sharing: Firms share facilities, but computing incompatibility can be an issue.

Backup tapes or resources are needed at the remote site.

16.11 Threat Assessment Process (Physical)

A structured threat assessment process for physical security involves:

1. Set up a steering committee.
2. Obtain information and assistance.
3. Identify all possible threats.
4. Determine the likelihood of each threat.
5. Approximate the direct costs.
6. Consider cascading costs.
7. Prioritize the threats.
8. Complete the threat assessment report.

16.12 Physical/Logical Security Integration

Organizations use many detection and prevention devices, but effectiveness is enhanced with central control. There is a desire to integrate physical and logical security, especially access control. Standards like FIPS 201-1 "Personal Identity Verification (PIV) of Federal Employees and Contractors" are relevant here.

16.13 Personal Identity Verification (PIV)

PIV for federal employees and contractors involves verifying identity. It includes assurance levels:

- Some confidence: Uses smart cards/PIN.
- High confidence: Includes the use of biometrics in addition to smart cards/PIN.

17 Human Factors

17.1 Key Topics

Key human factors in security include security awareness, training, and education; organizational security policy; personnel security; and email/Internet use policies.

17.2 Security Awareness, Training, and Education

This is a prominent topic in various standards and provides benefits such as improving employee behavior, increasing accountability, mitigating liability for employee actions, and complying with regulations and contractual obligations.

17.3 Awareness

Awareness seeks to inform employees and focus their attention on security issues, covering threats, vulnerabilities, impacts, and individual responsibility. It must be tailored to the organization's needs and delivered through various means like events, promotional materials, briefings, and policy documents. An employee security policy document is essential.

17.4 Training

Training teaches employees what they should do and how to do it to perform IS tasks securely. This encompasses general users learning good computer security practices.

17.5 Organizational Security Policy

A written security policy document is needed to define acceptable behavior, expected practices, and responsibilities. It clarifies what is protected and why, articulates security procedures/controls, states protection responsibility, and provides a basis for resolving conflicts. The policy must reflect executive security decisions related to protecting information, complying with laws, and meeting organizational goals.

17.6 Policy Document Content

A security policy document should address: the reason for the policy, who developed and approved it, whose authority sustains it, the laws/regulations it's based on, who will enforce it and how, whom it affects, what information assets must be protected, what users are required to do, how security breaches should be reported, and its effective/expiration dates.

17.7 Security Policy Topics

Policy topics can broadly cover principles, organizational reporting structure, physical security, hiring/management/firing, data protection, communications security, hardware, software, operating systems, technical support, privacy, access, accountability, authentication, availability, maintenance, violations reporting, business continuity, and supporting information. Resources like ISO 17799 can provide a framework.

17.8 Security in Hiring Process

The objective during hiring is to ensure employees, contractors, and third-party users understand their security responsibilities, are suitable for their roles, and to reduce the risk of theft, fraud, or misuse of facilities. Appropriate background checks, screening, and employment agreements are needed.

17.9 Background Checks and Screening

Issues include inflated resumes and reticence of former employers to provide references due to legal fears. Employers need to make significant effort in background checks and screening, obtaining detailed history, reasonably checking accuracy, and using experienced interviewers. For sensitive positions, intensive investigation may be warranted.

17.10 Employment Agreements

Employees should agree to and sign employment contract terms and conditions, which should include information on their and the organization's security responsibilities, confidentiality/non-disclosure agreements, and agreement to abide by the organization's security policy.

17.11 During Employment

Current employee security objectives include ensuring employees, contractors, and third-party users are aware of their responsibilities.

17.12 Policy Document Responsibility

A security policy needs broad support, especially from top management. It should be developed by a team including site security administrators, IT staff, user group representatives, security incident response team, responsible management, and legal counsel.

17.13 Incident Handling: Essential Control

Procedures for responding to security incidents are essential, recognizing that incidents will likely occur. These procedures codify actions to avoid panic, such as deciding whether to disconnect from the internet during a mass email worm outbreak. A responsible individual, identified in the policy, should make such decisions.

17.14 Types of Security Incidents

Security incidents include any action threatening classic security services like confidentiality, integrity, or availability. Examples include unauthorized access (viewing info, bypassing controls, using another's access, denying access) and unauthorized modification of information (corrupting, changing without authorization, unauthorized processing).

17.15 Detecting Incidents

Incidents can be detected through reports from users or administrative staff (who should be trained and encouraged to report) or by automated tools like system integrity verification tools, log analysis tools, network/host intrusion detection systems, and intrusion prevention systems. These tools should be updated to reflect new attacks or vulnerabilities. Administrators must also monitor vulnerability reports.

17.16 Incident Response Plan/Playbook

Incident response plans and playbooks, like those referenced from CISA, guide actions during a security incident.

17.17 Responders

Effective incident response involves various stakeholders:

- Leadership: Oversees response, allocates funding, makes high-impact decisions.
- Incident Handlers: Verify incidents, collect/analyze data, prioritize activities, act to limit damage, find root causes, restore operations.
- Technology professionals: Cybersecurity, privacy, system, network, cloud architects, engineers, administrators, developers involved in response/recovery.
- Legal: Review plans, policies, procedures for compliance.
- Public affairs/media relations: Inform media/public if needed.
- Human resources: Involved if employee is suspected of intentionally causing an incident.
- Physical security/facilities management: Involved if incident is physical or requires facility access.

18 Security Auditing

18.1 Terminology

A security audit is an independent review and examination of a system's records and activities to determine control adequacy, ensure policy/procedure compliance, detect breaches, and recommend countermeasure changes. Its objectives include establishing accountability for system activities.

18.2 Security Auditing Functions

Functions include auditing (review/analysis), detection (monitoring), reporting (alerts/summaries), and forensics (evidence collection).

18.3 Security Audit Process

The audit process involves:

- Data generation: Identifies auditing level, enumerates auditable events.
- Event selection: Includes/excludes events from the auditable set.
- Event storage: Creates and maintains the secure audit trail.
- Automatic response: Reactions taken upon detecting possible security violations.
- Audit analysis: Automated mechanisms to analyze audit data for violations.
- Audit review: Tools available to authorized users to assist in manual review.

18.4 Event Definition: Requirement

It is crucial to define what constitutes an auditable event. Common Criteria suggests events such as: object introduction/deletion, distribution/revocation of access rights, changes to subject/object security attributes, policy checks performed by security software, use of access rights to bypass checks, use of identification/authentication functions, security-related actions by operators/users, and data import/export from removable media.

18.5 Other Audit Requirements

Auditing can be done in real-time or batch mode. Data is often sent to a central host for analysis/storage. Logs can record date/time/location/user of access attempts (valid and invalid), attempts to change privileges, and can send violation messages to personnel.

18.6 Audit Trail Storage Alternatives

Storing audit trails involves trade-offs:

- Read/write file on host: Easy, low resource, fast access but vulnerable to intruders.
- Write-once device (CD/DVD-ROM): More secure but less convenient, delayed access.
- Write-only device (printer): Provides a paper-trail but impractical for analysis.

Integrity and confidentiality of the audit trail must be protected using encryption, digital signatures, or access controls.

18.7 Implementing Logging

The foundation of security auditing is the initial capture of audit data. Software must include "hooks" or capture points that trigger data collection and storage for preselected events. This is dependent on the operating system and application.

18.8 Windows Event Log

In Windows, each event is an entity describing an occurrence, containing details like a numeric ID and a set of attributes.

18.9 Syslog

Syslog is a standard for message logging, involving components like a generator (originates message), relayer (forwards message), and collector (gathers messages).

18.10 Logging at Application Level

Privileged applications often have security issues that may not be visible in system-level audit data (e.g., vulnerabilities due to inadequate input data checking). Therefore, capturing detailed behavior at the application level is necessary.

18.11 Interposable Libraries

Interposable libraries can intercept calls to shared library functions and perform audit-related functions.

18.12 Dynamic Binary Rewriting

This technique can also be used in auditing.

18.13 Audit Trail Analysis

Analysis programs and procedures vary widely. NIST SP 800-92 provides guidelines. Understanding the context of log entries, including related information from other logs or configurations, is crucial, as is the possibility of unreliable entries. Audit file formats can mix plain text and codes, requiring manual or automatic deciphering. Regularly reviewing entries helps establish a baseline understanding of normal activity.

18.14 Analysis Approaches

Approaches to audit data analysis include:

- **Basic alerting:** Indicates an interesting event type has occurred.
- **Baselining (anomaly detection):** Defines normal vs unusual events/patterns. Anomaly detection identifies deviations. Thresholding checks the frequency of events (e.g., number of refused connections). Can result in false positives/negatives.
- **Windowing:** Analyzing events within a set of parameters like time.
- **Correlation:** Seeking relationships among events.

18.15 Integrated SIEM Products

Integrated Security Information and Event Management (SIEM) products consolidate and analyze log and event data from various sources.

19 Legal and Ethical Aspects

19.1 Topics

Legal and ethical aspects in computer security cover cybercrime/computer crime, intellectual property issues, privacy, and ethical considerations.

19.2 Cybercrime / Computer Crime

Cybercrime is criminal activity where computers or networks are a tool, a target, or a place of activity. It can be categorized based on the computer's role. More comprehensive categorizations exist in documents like the Cybercrime Convention.

19.3 Intellectual Property

Intellectual property refers to creations of the mind, such as inventions, literary and artistic works, designs, and symbols, names and images used in commerce.

19.4 Copyright

Copyright protects the tangible or fixed expression of an idea, but not the idea itself. It is automatically assigned upon creation but may need registration in some countries. Copyright exists when the work is original and put into a concrete form, applying to various works including software.

19.5 Patents

A patent protects an invention, granting rights to prevent others from making, using, or selling the invention. Types include utility, design, and plant patents. The RSA public-key cryptosystem is an example of a patented technology.

19.6 Trademarks

A trademark is a word, name, symbol, or device used in trade with goods to indicate their source and distinguish them from others. Trademark rights allow preventing others from using a confusingly similar mark but not from making or selling the same goods under a clearly different mark.

19.7 DMCA (Digital Millennium Copyright Act)

The DMCA is a US law implementing WIPO treaties. It criminalizes the production and dissemination of technology or services used to bypass measures controlling access to copyrighted works or protecting copyright. It prohibits attempts to bypass these measures, with both criminal and civil penalties.

19.8 DMCA Exemptions

Certain actions are exempted from DMCA provisions, including fair use, reverse engineering, encryption research, security testing, and personal privacy. However, there is considerable concern that the DMCA inhibits legitimate security and cryptography research.

19.9 Digital Rights Management (DRM)

DRM refers to systems and procedures ensuring digital rights holders are clearly identified and receive stipulated payment for their works. DRM may impose further restrictions on use. There is no single standard or architecture. The goal is often to provide mechanisms for the complete content management lifecycle and persistent content protection for various digital content types, platforms, and media.

19.10 Privacy

Privacy overlaps with computer security. There has been a dramatic increase in the scale of information collected and stored, motivated by law enforcement, national security, and economic incentives. Individuals are increasingly aware of the access and use of their personal information, leading to concerns about the extent of privacy compromise and a range of responses.

19.11 EU Privacy Law

The European Union Data Protection Directive, adopted in 1998, aims to ensure member states protect fundamental privacy rights when processing personal information and prevent restrictions on the free flow of personal information within the EU. It is organized around principles of notice, consent, consistency, access, security, onward transfer, and enforcement.

19.12 US Privacy Law

US privacy law includes the Privacy Act of 1974, which permits individuals to determine records kept about them, forbid their use for other purposes, obtain access to their records, ensures agencies properly handle personal information, and creates a private right of action. There are also a range of other privacy laws.

19.13 Organizational Response to Privacy

Organizations should develop and implement a data protection and privacy policy, communicate it to all involved in processing personal information, and ensure compliance through appropriate management structure and control. This is often best achieved by appointing a data protection officer to provide guidance. Responsibility for handling personal information and awareness of data protection principles should align with relevant legislation. Appropriate technical and organizational measures must be implemented to protect personal information.

19.14 Ethical Issues

Ethics are not precise laws and many areas present ethical ambiguity. Many professional societies have ethical codes of conduct, which can serve as a positive stimulus, educate, provide support, deter and discipline, and enhance the profession's public image.

19.15 Codes of Conduct Themes

Codes from organizations like ACM, IEEE, and AITP place emphasis on responsibility to other people. Common themes include respecting the dignity and worth of others, personal integrity and honesty, responsibility for work, confidentiality of information, public safety/health/welfare, participation in professional societies to improve standards, and the notion that public knowledge and access to technology is equivalent to social power.