KIGALI INDEPENDENT UNIVERSITY

SCHOOL OF SCIENCE AND TECHNOLOGY

NETWORK AND WEB SECURITY

Dr. GASPARD

Assignment presented by:

KAMATE KATENDE TIMOTHEE

kamatekatende@gmail.com, tim58841@gmail.com

202312051

MSIS/Y2/W

School Year: 2023-2025

QUESTION 1

1)Three Main Concerns with Password Authentication

a. **Weak Passwords and Human Error**

- **Concern**: Many users create weak passwords that are easy to guess or crack, such as "123456" or "password." Additionally, users may reuse passwords across multiple platforms, increasing the risk of exposure if one system is breached.
- **Implication**: Weak or reused passwords make systems vulnerable to brute-force attacks, dictionary attacks, and credential-stuffing attacks. An attacker gaining access to one account can potentially access multiple accounts if passwords are reused.

b. **Storage and Transmission Vulnerabilities**

- **Concern**: If passwords are stored in plaintext or with weak hashing algorithms on a server, a data breach can expose sensitive user information. Similarly, passwords can be intercepted during transmission if communication is not encrypted (e.g., no HTTPS).
- **Implication**: Exposed or intercepted passwords can lead to unauthorized access, resulting in data theft, account hijacking, or further breaches. Proper hashing and secure transmission protocols are critical for protection.

c. **Susceptibility to Social Engineering and Phishing**

- **Concern**: Attackers exploit human psychology to trick users into revealing their passwords. Social engineering techniques include phishing emails, deceptive websites, or impersonating trusted individuals.

- **Implication**: Even the strongest password policies can fail if users are manipulated into sharing their credentials, rendering technical safeguards ineffective.

Social Engineering Attack on Passwords

Definition

A social engineering attack on passwords refers to methods attackers use to manipulate or deceive individuals into revealing their passwords or other sensitive information. These attacks target the human element of security rather than exploiting technical vulnerabilities.

Examples of Social Engineering Attacks

1. **Phishing**
    - Attackers send fraudulent emails, messages, or create fake websites that mimic legitimate organizations. Users are tricked into entering their credentials on these platforms.
    - Example: A user receives an email claiming to be from their bank, asking them to "verify" their account by clicking a link and logging in.
2. **Pretexting**
    - Attackers create a fabricated scenario to convince a victim to divulge sensitive information.
    - Example: An attacker pretends to be IT support and contacts a user, claiming there is an issue with their account and requesting their password to resolve it.
3. **Baiting**
    - Attackers use enticements, such as free software or USB drives, to lure victims into a trap where they reveal their credentials or unknowingly install malware.

- Example: A victim downloads a "free antivirus program" that prompts them to log in with their credentials, which are then captured.

4. **Shoulder Surfing**
    - Direct observation techniques to obtain a user's password, such as looking over their shoulder or using cameras to capture keystrokes.

Countermeasures Against Social Engineering

- **User Awareness and Training**: Educate users on recognizing phishing emails, suspicious links, and impostor tactics.
- **Multi-Factor Authentication (MFA)**: Use MFA to ensure that even if a password is compromised, the account remains secure.
- **Password Managers**: Encourage the use of password managers to generate and store strong, unique passwords.
- **Verification Mechanisms**: Implement robust verification processes to ensure that requests for sensitive information are legitimate.
- **Zero-Trust Principles**: Design systems that do not rely solely on passwords, minimizing the impact of social engineering attacks.

2. **Broad Classification of Attacks on Passwords**

Attacks on passwords are typically classified based on the method used to gain unauthorized access to credentials or authentication systems. These attacks exploit weaknesses in users, systems, or the cryptographic methods employed to protect passwords. They can be broadly categorized into **Offline Attacks**, **Online Attacks**, and **Social Engineering Attacks**.

a. Offline Attacks

Offline attacks involve obtaining a copy of the password database or hashes and then attempting to crack the passwords without interacting with the authentication system. These attacks leverage computing power to guess or derive the passwords.

Types of Offline Attacks

1. **Brute-Force Attacks**
    - **How It Works**: The attacker systematically tries all possible combinations of characters until the correct password is found.

- **Strength**: Effective against short or simple passwords but computationally expensive for long, complex passwords.
- **Mitigation**: Use long, complex passwords and strong hashing algorithms like bcrypt or Argon2, which slow down brute-force attempts.

2. **Dictionary Attacks**
   - **How It Works**: Attackers use precompiled lists of commonly used passwords (e.g., "123456," "password") and attempt them sequentially.
   - **Strength**: Faster than brute-force for common passwords; ineffective against unique or complex passwords.
   - **Mitigation**: Encourage users to create unique, non-dictionary passwords.
3. **Rainbow Table Attacks**
   - **How It Works**: Attackers use precomputed tables mapping password hashes to plaintext passwords, significantly reducing cracking time.
   - **Strength**: Effective when weak or no salt is used in hashing.
   - **Mitigation**: Always salt hashes with unique, random values to make rainbow tables impractical.
4. **Keyloggers or Malware**
   - **How It Works**: Malicious software records keystrokes or extracts saved credentials from devices, allowing attackers to bypass hashing entirely.
   - **Strength**: Effective and stealthy but relies on infecting a target's device.
   - **Mitigation**: Use updated antivirus software and secure endpoints.

b. Online Attacks

Online attacks are conducted by directly interacting with the authentication system in real-time, typically over a network. These attacks rely on sending multiple login attempts to the system.

Types of Online Attacks

1. **Credential Stuffing**
   - **How It Works**: Attackers use stolen username-password pairs from breached databases to log in to other services where users may have reused credentials.
   - **Strength**: Exploits poor password hygiene and is highly automated.
   - **Mitigation**: Use Multi-Factor Authentication (MFA) and enforce unique passwords.
2. **Password Spraying**

- **How It Works**: Attackers try a few common passwords (e.g., “Password123”) across many accounts to avoid detection by account lockout mechanisms.
- **Strength**: Avoids triggering lockouts but relies on weak passwords being in use.
- **Mitigation**: Implement account lockout thresholds and monitor login attempts.

3. **Phishing and Spoofing**
    - **How It Works**: Attackers create fake login pages or send deceptive emails/messages to trick users into providing their passwords.
    - **Strength**: Highly effective against inattentive or uninformed users.
    - **Mitigation**: Educate users, verify URLs, and deploy anti-phishing tools.
4. **Man-in-the-Middle (MITM) Attacks**
    - **How It Works**: Attackers intercept network traffic to capture login credentials during transmission.
    - **Strength**: Exploits weak encryption or unencrypted communication.
    - **Mitigation**: Always use HTTPS, TLS, and encrypted communication channels.

e. Social Engineering Attacks

Social engineering attacks exploit human behavior and psychology rather than technical vulnerabilities to gain passwords.

Types of Social Engineering Attacks

1. **Phishing**
    - **How It Works**: Deceptive messages trick users into entering credentials on fake websites or sending them directly to attackers.
    - **Example**: An email claiming to be from IT support requesting urgent password reset.
2. **Pretexting**
    - **How It Works**: Attackers impersonate trusted individuals or organizations to manipulate victims into revealing passwords.

    - **Example**: A fake customer service agent requesting credentials for account verification.
3. **Baiting**
    - **How It Works**: Luring victims into compromising situations, such as using infected USB drives or downloading malicious software, to steal credentials.
    - **Example**: Offering “free movie downloads” that require users to log in with sensitive credentials.
4. **Shoulder Surfing**
    - **How It Works**: Observing users entering passwords in public spaces.
    - **Example**: Watching someone type their password on a laptop in a café.

Mitigation Strategies Across All Attack Types

1. **Strong Password Policies**: Require long, complex passwords with a mix of characters, numbers, and symbols.
2. **Multi-Factor Authentication (MFA)**: Use secondary authentication methods, such as biometrics or one-time passwords.
3. **Password Hashing and Salting**: Store passwords securely with modern hashing algorithms and unique salts.
4. **User Training**: Educate users on recognizing phishing attempts and secure password practices.
5. **Secure Communication Protocols**: Ensure all network communication is encrypted.
6. **Rate Limiting and Lockout Policies**: Prevent automated attacks by limiting login attempts.
7. **Monitoring and Alerts**: Detect and respond to suspicious login patterns.

3. Access Control Matrices and Access Control Lists

Access Control Matrices (ACMs) are a fundamental model for defining access rights in a system. They represent the permissions of subjects (users, processes) over objects (files, resources) in a tabular format, with:

- **Rows**: Representing subjects.
- **Columns**: Representing objects.

- **Entries**: Representing the specific rights (e.g., read, write, execute) a subject has over an object.

While the ACM provides a conceptual framework for managing permissions, it is rarely implemented directly due to its inefficiency in large systems. Instead, it is represented in practical implementations such as **Access Control Lists (ACLs)**.

How Access Control Lists Represent ACMs

An **Access Control List (ACL)** is a row-centric or object-centric representation of an ACM. Instead of storing the entire matrix, each object maintains a list of subjects and their associated permissions. This approach simplifies management and improves efficiency by focusing on the access rights of a single object.

Structure of an ACL

Each ACL is associated with an object and contains entries (known as Access Control Entries or ACEs). Each ACE specifies:

- A subject (e.g., user, group, or role).
- The access rights granted to that subject (e.g., read, write, delete).

Example of Representation

Given an ACM:

	File A	File B	File C
Alice	Read	Write	Read
Bob	Write	-	Read
Charlie	-	Execute	-

The equivalent representation using ACLs:

1. **File A**

    - Alice: Read
    - Bob: Write
2. **File B**
    - Alice: Write
    - Charlie: Execute
3. **File C**
    - Alice: Read
    - Bob: Read

In this format, each file maintains its own ACL, listing all subjects with access rights specific to that file.

Advantages of Using ACLs

1. **Efficiency**: ACLs optimize storage by focusing only on objects and the subjects that interact with them, avoiding the storage of redundant "no-access" entries.
2. **Modularity**: Permissions for each object can be modified independently, providing localized control.
3. **Widely Supported**: ACLs are supported by most modern operating systems and file systems (e.g., NTFS, Linux ext4).

Comparison of ACLs and ACMs

Aspect	Access Control Matrix	Access Control List
Storage Model	Tabular representation of all permissions	Object-specific lists of permissions
Efficiency	Inefficient for large systems	Efficient by focusing on relevant permissions
Flexibility	Hard to scale for dynamic systems	Scales well with changes to objects or subjects
Real-World Use	Rarely implemented directly	Commonly implemented in OS and file systems

Environments Where Access Control Lists (ACLs) Are Widely Used

Access Control Lists (ACLs) are extensively utilized in environments requiring fine-grained, object-level access control. They are particularly valuable in systems where the access rights of subjects (users, roles, or processes) must be enforced at a granular level. Common environments include:

1. Operating Systems

- **Use Case**: File and directory permissions. ACLs control which users or groups can read, write, or execute files.
- **Examples**:
    - **Windows NTFS**: Provides detailed ACLs for files and folders.
    - **Linux (ext4, XFS)**: Uses setfacl to assign permissions beyond standard chmod.

2. Networking and Firewalls

- **Use Case**: Regulating access to network resources by defining which IP addresses, protocols, or ports are allowed or denied.
- **Examples**:
    - **Routers and Switches**: Cisco ACLs manage traffic flow within and between networks.
    - **Firewalls**: Determine whether specific traffic is permitted or blocked based on ACL rules.

3. Cloud Computing Platforms

- **Use Case**: Controlling access to storage buckets, databases, or APIs.
- **Examples**:
    - **Amazon S3 Buckets**: ACLs define who can read, write, or manage objects.
    - **Azure and Google Cloud**: Use ACLs for granular access control on cloud resources.

4. Database Management Systems

- **Use Case**: Restricting access to tables, rows, or columns based on user roles.
- **Examples**:
    - PostgreSQL and MySQL allow ACL-style role-based permissions.
    - Data access platforms like Snowflake use ACLs for row-level security.

5. Application Development

- **Use Case**: Fine-grained authorization for accessing APIs, web applications, or specific services.
- **Examples**:
    - Web frameworks (e.g., Django) implement ACLs to enforce access policies.
    - Microservices use ACLs to control API access based on tokens or roles.

6. IoT and Embedded Systems

- **Use Case**: Regulating device interactions or communication in a network.
- **Examples**:
    - Smart home hubs use ACLs to determine which devices can interact.
    - Industrial IoT systems secure sensors and actuators with ACL-based rules.

4. We analyze the scenarios based on the Access Control List (ACL) policies and the group permissions provided.

Scenario Analysis for File 1

Given ACL:

- File 1:

    - **Group 1**: Read (R)
    - **Group 2**: Read and Write (RW)

Alice's Group Membership:

- **Group 1**: R
- **Group 2**: RW

Case 1: First Relevant Entry Policy

- **Definition**: The system evaluates permissions by checking the first matching entry in the ACL for a user. Once a match is found, no further entries are considered.
- **Analysis**:
    - Alice is in **Group 1**, where the first entry grants **Read (R)** permissions.
    - The evaluation stops here, so Alice will **not be allowed** to write to File 1.

Case 2: Any Permission in List Policy

- **Definition**: The system evaluates all applicable entries in the ACL for a user and grants access if any of the entries include the requested permission.
- **Analysis**:
    - Alice is in both **Group 1** (R) and **Group 2** (RW).
    - The **RW** permission from Group 2 allows her to write.
    - Alice will **be allowed** to write to File 1 under this policy.

Scenario Analysis for File 2

Given ACL:

- File 2:
    - **Group 3**: Read, Write, Execute (RWE)

Alice's Group Membership:

- Alice is not in **Group 3**.
- Hence, Alice does **not** have access to File 2.

Removing Group 3 Using "Access None"

- The **"Access None"** rule explicitly denies access to specific users or groups, overriding all other permissions.
- To remove the need for Group 3, list individual permissions for users instead.

Revised ACL for File 2:

- File 2:
    - Bob: Read, Write, Execute (RWE)
    - Cynthia: Read, Write, Execute (RWE)
    - Access None: Alice, David, Eve

In this revised ACL:

- Permissions are explicitly granted to Bob and Cynthia.
- Alice, David, and Eve are explicitly denied access using the **Access None** rule.

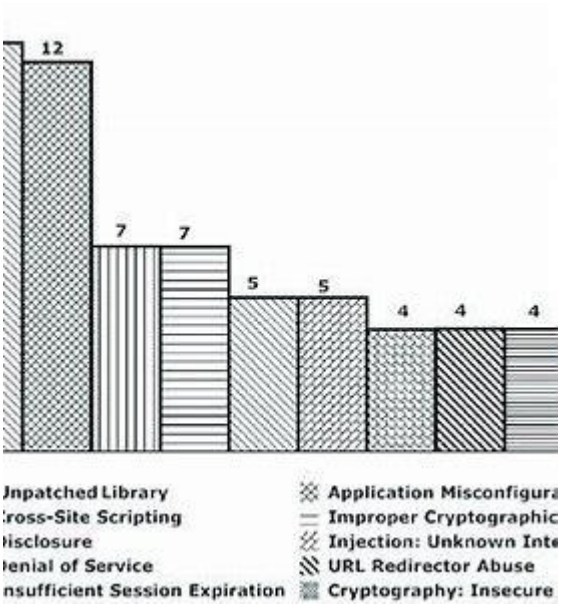
File 1 (First Relevant Entry Policy): Alice cannot write because the first relevant entry (Group 1) grants only Read permission.

File 1 (Any Permission in List Policy): Alice can write because the RW permission from Group 2 is considered.

File 2 (Remove Group 3): Replace Group 3 with explicit permissions for Bob and Cynthia, while denying access to others using "Access None."

QUESTION 2

In cybersecurity, threats are potential events or actions that can compromise the confidentiality, integrity, or availability of a cyber system. Below are different types of threats, categorized based on their nature and impact:



1. Threat Categories by Source

1. **External Threats**
    - **Hacking**: Unauthorized access to systems or networks.
    - **Malware**: Viruses, worms, trojans, ransomware, etc., aimed at damaging or stealing data.
    - **Phishing**: Social engineering attacks to steal sensitive information.
    - **Denial of Service (DoS)**: Overloading systems to make them unavailable.
2. **Internal Threats**
    - **Malicious Insider**: Employees or partners intentionally harming the organization.
    - **Accidental Insider**: Mistakes by authorized users, like misconfiguring systems or leaking data.
3. **Third-Party Threats**
    - **Supply Chain Attacks**: Exploits introduced through third-party vendors or software.

- **Dependency Vulnerabilities**: Flaws in external libraries or APIs.

2. Threat Categories by Intent

1. **Intentional Threats**
    - Cybercrime: Financially motivated attacks like fraud or ransomware.
    - Cyberterrorism: Attacks aimed at causing fear or disruption.
    - Espionage: Stealing sensitive information for competitive or political advantage.
2. **Unintentional Threats**
    - Human Errors: Misconfigurations, weak passwords, or accidental disclosures.
    - Environmental Disruptions: Power failures, hardware malfunctions, or natural disasters.

3. Threat Categories by Target

1. **Data Breaches**: Theft or exposure of sensitive data.
2. **System Compromises**: Exploits of vulnerabilities to gain unauthorized control.
3. **Network Threats**: Man-in-the-middle attacks, sniffing, or interception of network traffic.

In cybersecurity, threats are potential events or actions that can compromise the confidentiality, integrity, or availability of a cyber system. Below are different types of threats, categorized based on their nature and impact:

1. Threat Categories by Source

1. **External Threats**
    - **Hacking**: Unauthorized access to systems or networks.
    - **Malware**: Viruses, worms, trojans, ransomware, etc., aimed at damaging or stealing data.
    - **Phishing**: Social engineering attacks to steal sensitive information.
    - **Denial of Service (DoS)**: Overloading systems to make them unavailable.
2. **Internal Threats**

- **Malicious Insider**: Employees or partners intentionally harming the organization.
- **Accidental Insider**: Mistakes by authorized users, like misconfiguring systems or leaking data.

3. **Third-Party Threats**
   - **Supply Chain Attacks**: Exploits introduced through third-party vendors or software.
   - **Dependency Vulnerabilities**: Flaws in external libraries or APIs.

2. Threat Categories by Intent

1. **Intentional Threats**
   - Cybercrime: Financially motivated attacks like fraud or ransomware.
   - Cyberterrorism: Attacks aimed at causing fear or disruption.
   - Espionage: Stealing sensitive information for competitive or political advantage.
2. **Unintentional Threats**
   - Human Errors: Misconfigurations, weak passwords, or accidental disclosures.
   - Environmental Disruptions: Power failures, hardware malfunctions, or natural disasters.

3. Threat Categories by Target

1. **Data Breaches**: Theft or exposure of sensitive data.
2. **System Compromises**: Exploits of vulnerabilities to gain unauthorized control.
3. **Network Threats**: Man-in-the-middle attacks, sniffing, or interception of network traffic.

Dangerous Threats vs. Tolerable Threats

More Dangerous Threats

1. **Advanced Persistent Threats (APTs)**:
   - Sophisticated, long-term attacks targeting critical infrastructure or sensitive systems.
   - Examples: Nation-state attacks, espionage.

    - **Impact**: High risk due to the potential for espionage, economic damage, or large-scale harm.
2. **Ransomware**:
    - Encrypts data, demanding payment for decryption keys.
    - **Impact**: Can paralyze an organization and cause financial loss.
3. **Zero-Day Exploits**:
    - Exploits unknown vulnerabilities before a fix is available.
    - **Impact**: Extremely difficult to detect and mitigate.
4. **Supply Chain Attacks**:
    - Target dependencies or third-party software to introduce vulnerabilities.
    - **Impact**: Widespread because of the ripple effect on multiple systems.

Tolerable Threats

These are threats that pose lower risks or have manageable impacts:

1. **Spam and Adware**:
    - Annoying but usually not harmful.
    - **Mitigation**: Filters and removal tools.
2. **Low-Level Phishing Attempts**:
    - Easily recognizable by well-trained users.
    - **Mitigation**: Employee training and email filtering.
3. **Non-Critical System Failures**:
    - Temporary disruptions in less essential systems.
    - **Mitigation**: Backup systems and redundancy.
4. **Minor Human Errors**:
    - Simple mistakes like sending an email to the wrong recipient.
    - **Mitigation**: Proper training and role-based access controls.