AAA (Authentication, authorization, Accountability

AAA

- One primary goal of deploying cybersecurity is creating the ability of distinguishing between authorized entities and unauthorized entities.
- Cybersecurity goals/services:
  - Authentication(the process of confirming the identity of a user or entity attempting to access a system).
  - Authorization(the process of granting or denying access to specific resources).
  - Accountability, each user in the organization or the system must have specific responsibilities.

Authentication

- Authentication: the process of verifying the identity of a system entity.
  - An entity could be any subject in a system such as a person, a process, or a program.
- Authentication represents fundamental security building block
  - Basis of authorization and accountability.
- Authentication involves two steps:
  - Enrollment, a user or entity initially registers with the system by providing their identity information and creating authentication credentials.
  - Verification, the user or entity can then authenticate themselves by presenting the previously established credentials to the system.

Authentication Factors

- Four means of authenticating entity's identity
- Based on something you
  - know, e.g. password, PIN
  - have, e.g. key, token, smartcard
  - are (static/physiological biometrics), e.g. fingerprint, retina, iris
  - are (dynamic/behavioral biometrics), e.g. voice, signature, gait
- Can use alone or combined(Multi-factor Authentication)
- All can provide user authentication
- All have issues (weakness, complexity, usability)

Continuous Authentication

- Continuous Authentication (CA): is the mechanism of verifying the user identity from login through the end of the user session.
- Unlike static authentication which authenticates users just once at login.
- CA evaluates user behavior patterns on an ongoing basis, by taking in the consideration changing risk factors such as location, device posture, and typing style.

CA Over SA

- **Enhanced Security**: Continuous authentication provides ongoing monitoring of user activity, allowing for real-time detection of suspicious behavior or unauthorized access. This proactive approach can help prevent security breaches by identifying and responding to threats more quickly than static authentication methods.
- **Reduced Risk of Unauthorized Access:** Static authentication methods, such as passwords or biometric scans performed only at login, are susceptible to attacks such as credential theft or spoofing. Continuous authentication continuously verifies the user's identity throughout their session, reducing the risk of unauthorized access even if credentials are compromised.
- **Adaptive Access Controls**: Continuous authentication enables adaptive access controls that adjust dynamically based on the user's behavior and risk profile. For example, if a user's behavior suddenly deviates from their normal patterns, the system can trigger additional authentication measures or restrict access until the user's identity is verified.

Session Hijacking

- Session: is a time-limited communication path between two or more computing devices, usually client and server.
- Session is the period of activity between a user logging-in and logging-out of a system.
- Cookie: is small blocks of data created by a web server while a client is browsing a website, and placed on the client's web browser.

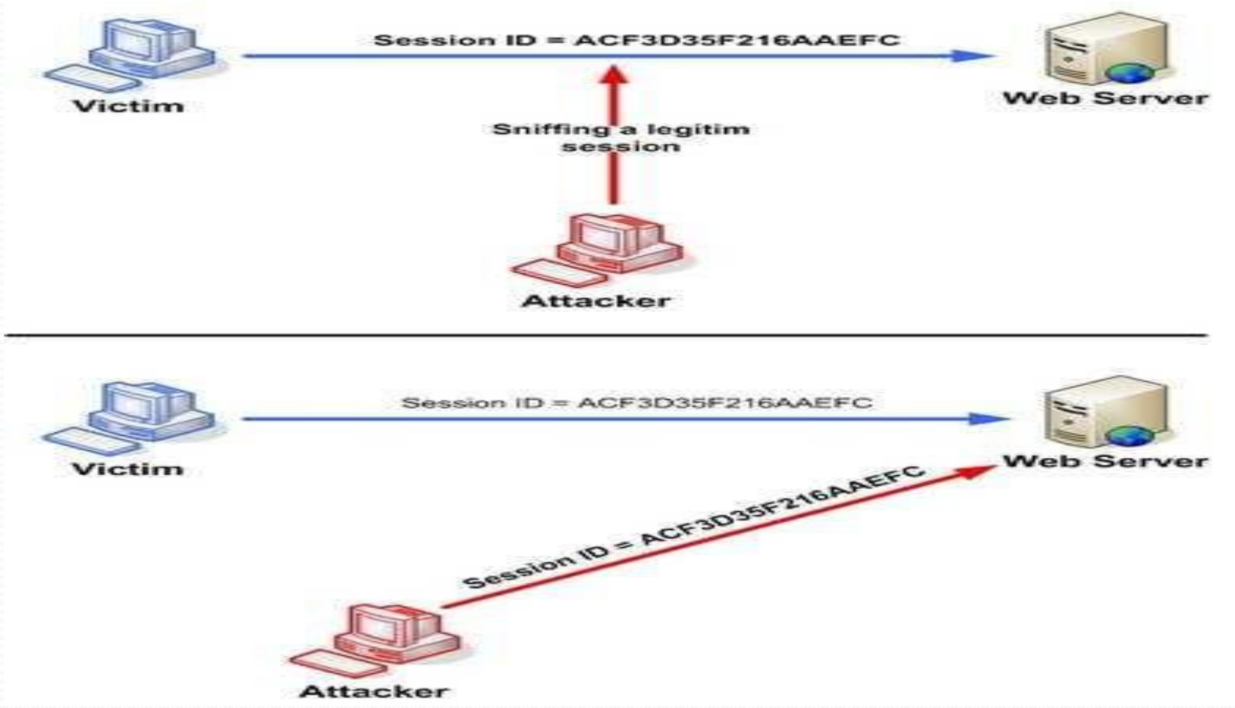
Session Hijacking

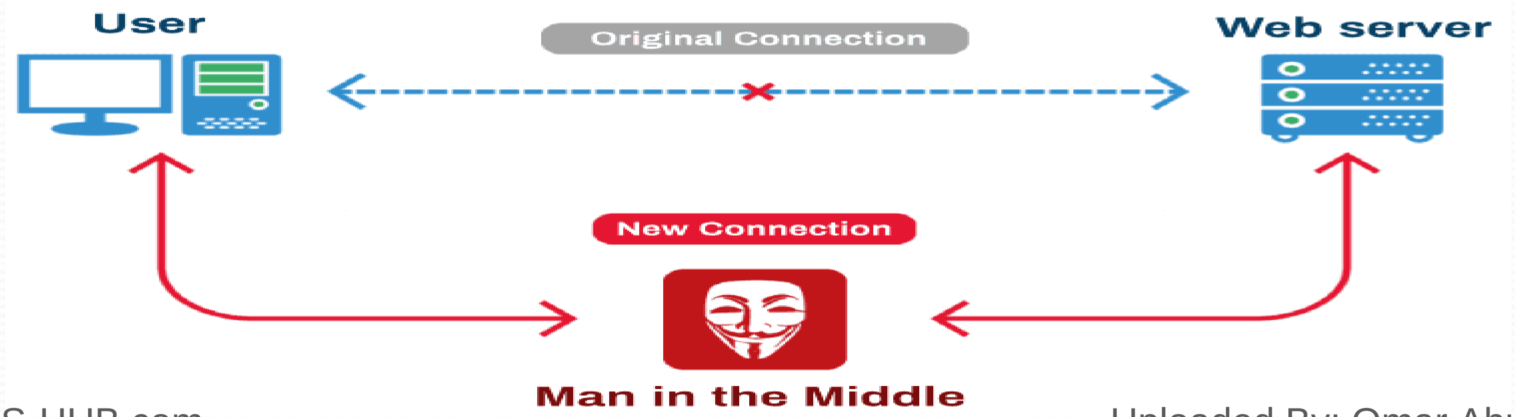
- Session Hijacking: is the mechanism of compromising a valid web session to gain unauthorized access to a web server, by stealing session cookies used to authenticate a client to a web server.
- *Suppose Alice is logged into her online banking account. Her session identifier or session cookie is transmitted over an unencrypted connection. Meanwhile, Mallory, an attacker, intercepts this communication and captures Alice's session identifier.*
- A session cookie could be stolen by using different techniques, such as:
  - XSS(Cross Cite Scripting)
  - Packet Sniffing
  - MITM(Man in the Middle).
  - Malware

Scenario(Session hijacking)

- Techniques used by attackers in session hijacking:
  - **Packet Sniffing**, Mallory uses packet sniffing tools to intercept and analyze network traffic. When Alice communicates with the server, Mallory captures packets containing the session identifier, allowing her to steal the session.
  - **XSS(Cross Cite Scripting)** Mallory injects malicious scripts into a vulnerable web page visited by Alice. When Alice visits the compromised page, her browser executes the script, which steals her session identifier and sends it to Mallory.
  - **MITM(Man in the Middle)** Mallory positions herself between Alice and the server, intercepting all communication between them. She captures the session identifier sent by Alice and forwards it to the server, allowing her to impersonate Alice and hijack her session.
  - **Malware**, Mallory may have infected Alice's computer with a key-logger through a phishing email or malicious software download. As Alice enters her login credentials, the key-logger captures her session identifier along with her username and password.

Session Hijacking


Victim
Session ID = ACF3D35F216AAEFC
Web Server
Sniffing a legitim session
Attacker
Victim
Session ID = ACF3D35F216AAEFC
Web Server
Session ID = ACF3D35F216AAEFC
Attacker



Session Hijacking

- Countermeasures:
  - Encrypting traffic between client and server, Encrypting session identifiers and other authentication data helps prevent interception and tampering by attackers attempting to hijack the session.
  - Continuous Authentication, Continuous authentication systems can dynamically adjust access controls based on the user's behavior and risk profile. For example, if the system detects suspicious activity or a high-risk scenario, it can prompt the user to re-authenticate using additional factors or temporarily restrict access until the user's identity is verified.
  - Multi-Factor Authentication, By requiring the user to continuously authenticate using multiple factors, the system adds layers of security, making it more difficult for attackers to hijack the session.

Authorization

- Authorization: is the process of granting someone permissions/privileges to access certain levels of resources.
- This permission can be granted by a person or an automated system.
- Authorization is usually done with the goal of preventing unauthorized access to resources.

Authorization

- Authorization policies define what an individual identity or group may access in a system.
- Access controls represent the methods/measures that are used to enforce the deployment of authorization policies.
- Improper access control could lead to:
  - Unauthorized access
  - Information disclosure
  - Privilege escalation, this can happen using different scenarios such as Vulnerable web application, which leads to Exploiting file inclusion vulnerability, then executing arbitrary code.

Access Control Mechanisms

- Discretionary Access Control (DAC)
  - Access policies of resources are deployed based on rules specified by the subject/user (user can change the access policies).
- Mandatory access control (MAC)
  - Access policies of resources are deployed based on rules specified by central authority, such as a system manager, admin, or OS (user can **NOT** change the access policies).
- Role-Based Access Control (RBAC)
  - Access policies of resources are deployed based on the role assigned to a user or group in a system.
- Attribute-Based Access Control (ABAC)
  - Access policies of resources are deployed based on assigned attributes of the subject, attributes of the object, and attributes of the environment.

Privilege Escalation Attack

- Privilege Escalation, is the act of exploiting a vulnerability, or design flaw in a system to gain unauthorized elevated access to resources.
- The attacker keeps probing the compromised system to gain more/higher privileges.
- This attack happens after the initial unauthorized access (obtaining certain level of access privileges).
- Two types:
  - Horizontal Privilege Escalation
  - Vertical Privilege Escalation

Privilege Escalation Attack

- Horizontal Privilege Escalation:

Represents the mechanism where an attacker is moving laterally across resources of similar privileges (same level of privileges).

- Example:

Suppose an attacker has gained unauthorized access to a student account on Ritaj system. He is trying to steal files and the files he is got from this student account is not enough. He will start probing to gain access to other students accounts.

Privilege Escalation Attack

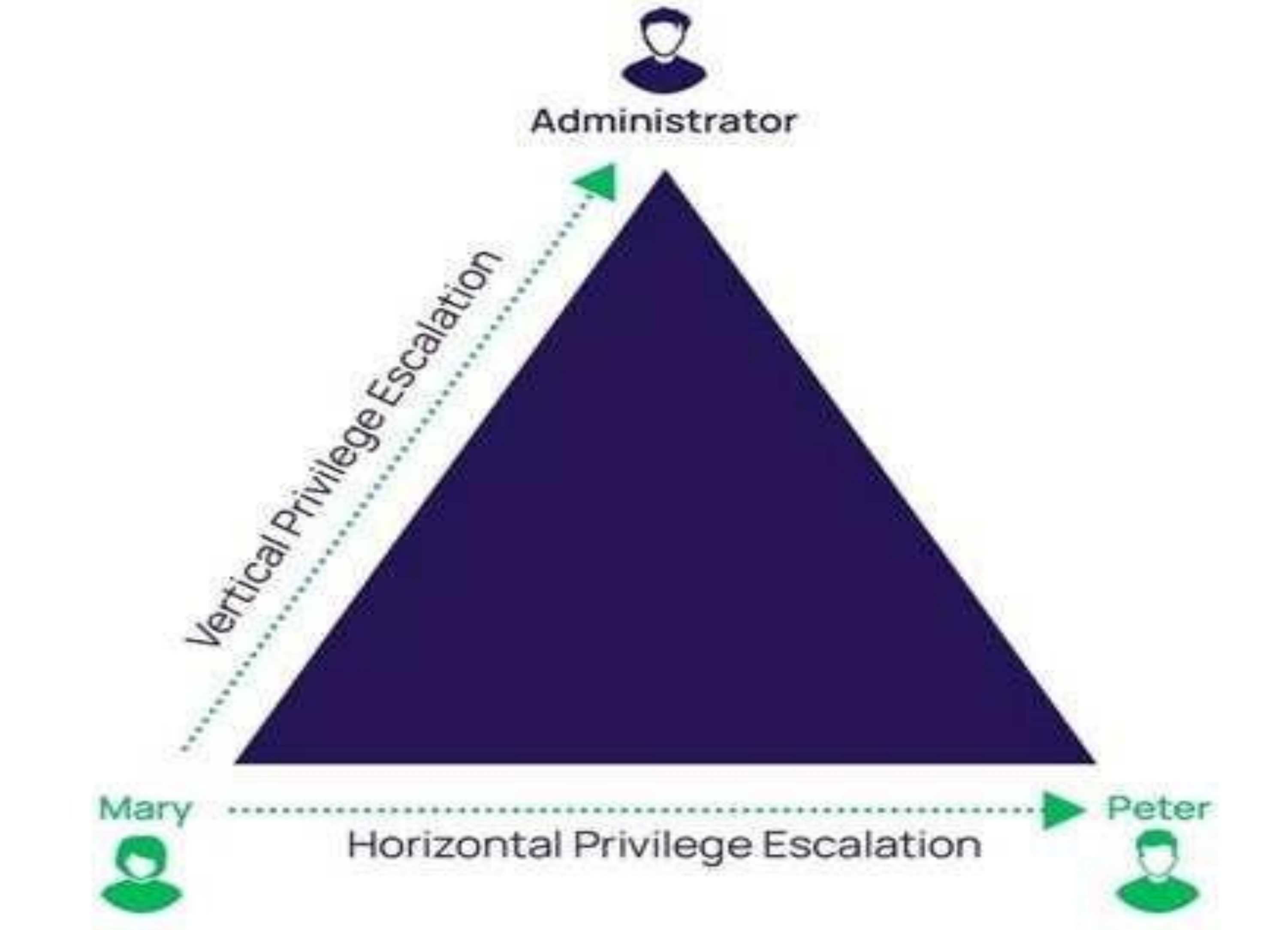
- Vertical Privilege Escalation:

The mechanism where an attacker is expanding/elevating a compromised user privileges to higher privileges, such as admin/root privileges (higher level of privileges).

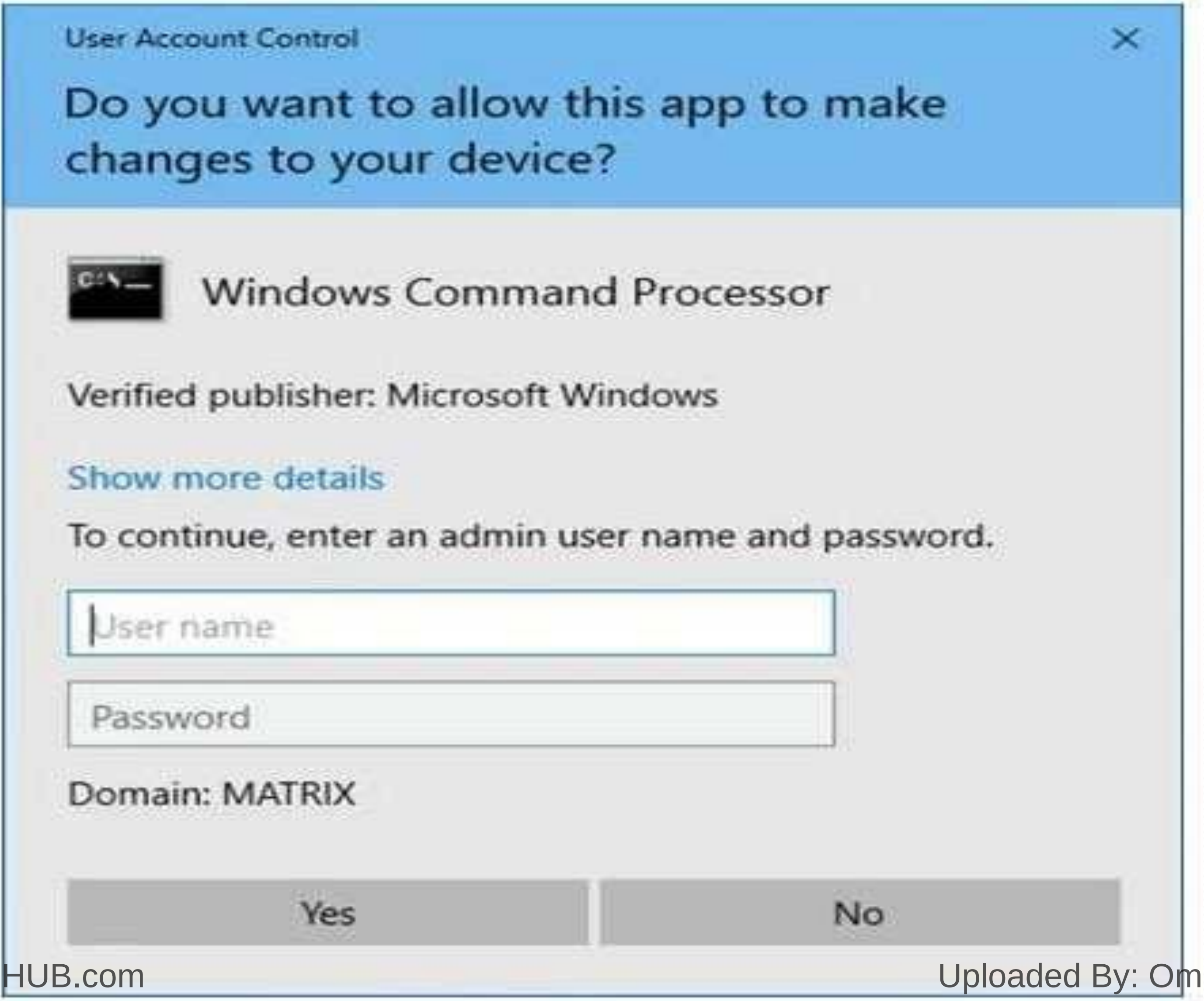
- Example:

Suppose an attacker has gained unauthorized access to a student account on Ritaj system. Then, the attacker starts probing the compromised student account to find a vulnerability that expands his permissions to admin privileges.

Privilege Escalation Attack


Administrator
Vertical Privilege Escalation
Mary
Peter
Horizontal Privilege Escalation

Privilege Escalation Attack



User Account Control

Do you want to allow this app to make changes to your device?

Windows Command Processor

Verified publisher: Microsoft Windows

Show more details

To continue, enter an admin user name and password.

User name

Password

Domain: MATRIX

Privilege Escalation Attack

- Countermeasures:
  - Deploying strong authentication mechanisms.
  - Deploying vulnerability scanners for web applications.
  - Deploying input validation for websites submissions.

Principle of Least Privilege

- Principle of Least Privilege (POLP): is the mechanism of granting the minimum required access of resources that a user needs to perform a specific task.
- POLP supports restrictive access rights, in order to mitigate system exposure to cyber-attacks, by minimizing the connection between users and systems.
- Example: using workstations in the labs.

Accountability

- Accountability: is the mechanism of making sure that an action of an entity in a system is traceable (i.e. knowing ***who*** did ***what*** action and ***when***).
  - Methods:
  - ✓ Non-repudiation techniques, such as signing a request of an action by a digital signature.
  - ✓ Auditing techniques, such as reviewing log files.

Accountability

- Auditing capabilities ensure users are accountable for their actions:
  - System-level events
  - Application-level events
  - User-level events
- Reviewing log information
- Protecting log information

Identification

- Identification + AAA = IAAA
- Identification: the process of establishing an identity of a system entity.
  - An entity could be a person, or non-person entity (NPE), such as process or device.
  - Example: user name, ID number.
  - Identification is a requirement for authentication.
  - Identification is one-to-many matching process.
  - Authentication is one-to-one matching process.

Identification

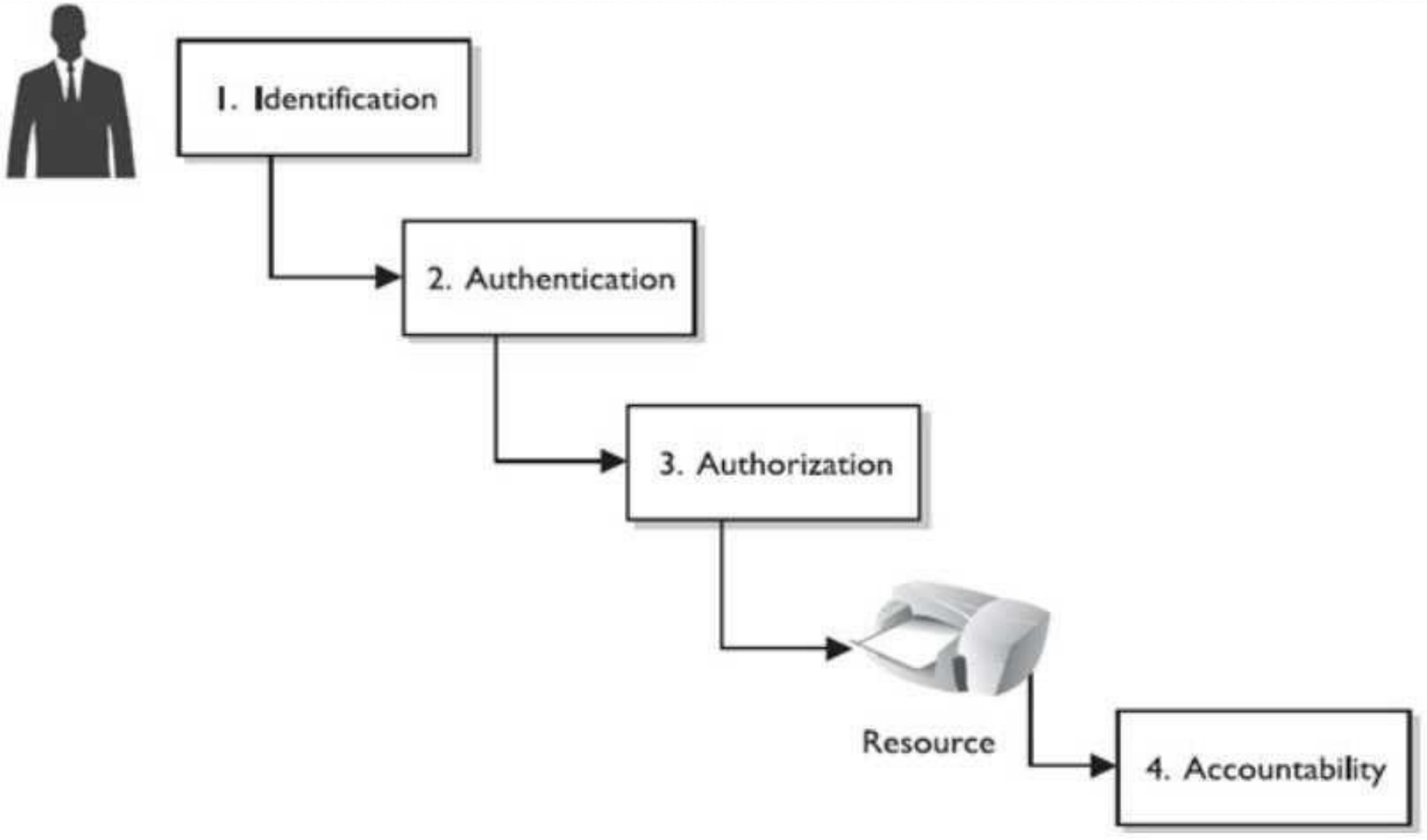
- Why do we need identification?
  - Verification (authentication), whereas Identification is the first step in the authentication process, where the entity provides its identity (e.g., username, device ID, digital certificate) to the system or network. The system then verifies the identity provided by the entity through various authentication mechanisms, such as passwords, biometrics, tokens, or digital certificates.
  - Access control decisions, based on the authentication and authorization for identified entity the access control can be achieved.
  - Auditing (accountability), whereas each identified entity's interactions with the network or system are recorded, including login/logout events, resource access, and system activities.

Identification

The following criteria should be taken in the consideration when creating a new identity for a user in a particular system:

- Each identity should be unique.
- A standard naming scheme should be followed.
- The identity should be non-descriptive of the user’s position or tasks.

IAAA


1. Identification
2. Authentication
3. Authorization
Resource
4. Accountability

IAM

- IAM: Identity and Access Management
- Also known as Identity Management (IdM)
- IAM: is a set of policies, processes, and technologies that manages the identities of entities, and authenticate these identities for accessing data and other resources.

IAM

- IAM can be define also as a framework of polices, technologies, and process that used to manage digital identities and control access to digital resources within an organization.
- IAM systems enable organizations to ensure that the right individuals have access to the right resources at the right time, while also enforcing security policies and compliance requirements.

IAM components and principles

- **Identity Lifecycle Management,** managing the lifecycle of digital identities, including employees, contractors, partners, and customers. This involves processes such as provisioning (creating accounts), de-provisioning (disabling or deleting accounts), and managing changes to user attributes (e.g., role changes, name changes).
- **Authentication,** which refers to the process of verifying the identity of users attempting to access using various authentication methods such as password, biometric, and token.
- Authorization, controlling which resource the user allowed to access and what action can be performed.
- Auditing and Reporting, IAM system provide logging, auditing, and reporting capabilities to track user activity.
- Examples for such frameworks, NIST SP800-63, and ISO/IEC27002

IAM Advantages

- IAM is an automated system, which:
  - ✓ Reduces human errors
  - ✓ Reduces cost, time, and efforts, since it will reduce administrative overhead.
  - ✓ Improves productivity and performance, since it support Single sign -on(SSO), which allow the user to access multiple applications with single set of credentials.
  - ✓ Enhances security, since it will ensure that users have appropriate levels of access based on their roles and responsibilities, and by reducing the risk of unauthorized access and data breaches.

Positive Security Implications of SSO:

- **Reduced Password Fatigue**: since the user required only to remember and mange one set of credential, which will lead to reduce weak passwords and passwords reuse.
- **Centralized Authentication**, SSO centralizes authentication processes, making it easier for organizations to enforce stronger authentication methods, such as multi-factor authentication (MFA), across multiple applications and systems.
- Improve Experience, SSO simplifies the user login experience by eliminating the need to enter credentials repeatedly, which can reduce the likelihood of users resorting to insecure practices (like writing down passwords).