

EC-Council

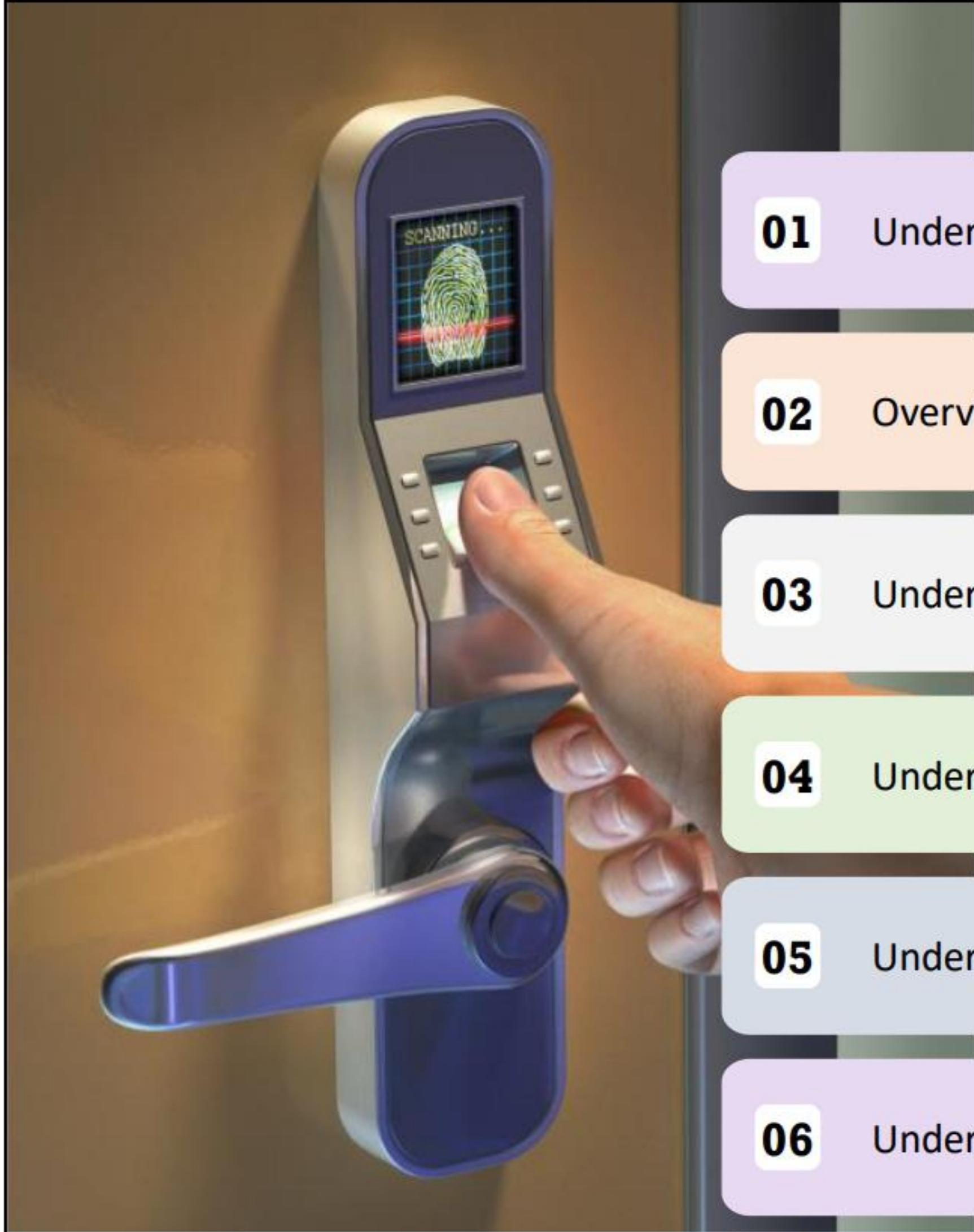
CCT

Certified Cybersecurity Technician

Module - 04

Identification, Authentication, and Authorization

This page is intentionally left blank.



Module Objectives

- 01 Understanding the Terminology, Principles, and Types of Access Control
- 02 Overview of Identity and Access Management (IAM)
- 03 Understanding the User Access Management
- 04 Understanding the Different Types of Authentication
- 05 Understanding the Different Types of Authorization
- 06 Understanding the User Accounting

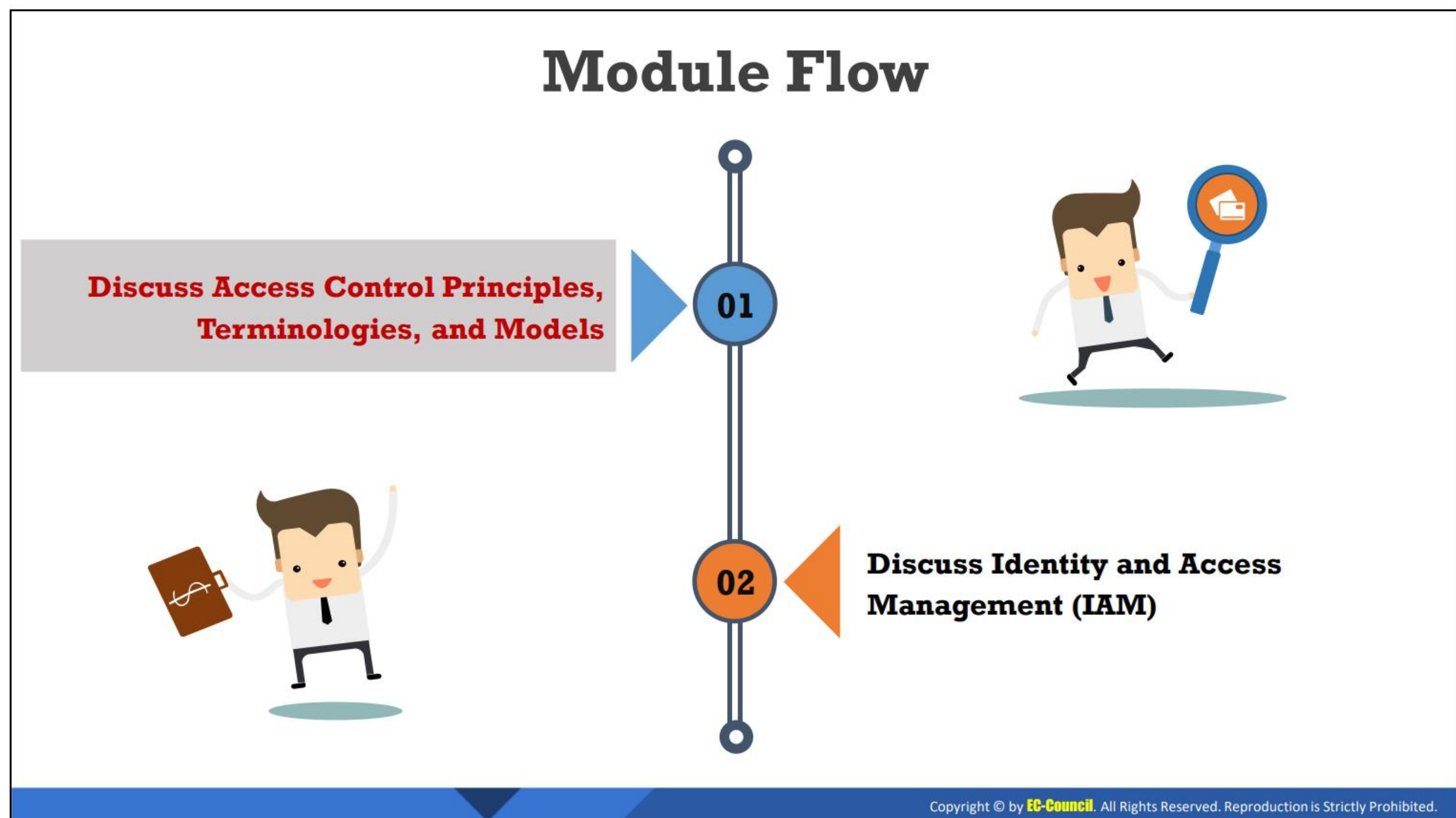
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Module Objectives

The most serious risk that organizations are facing today is unauthorized access to sensitive data. To control such data breaches, organizations require strong identification, authentication, and authorization mechanisms to effectively manage access to critical assets and sensitive data. This module provides an overview of various methods and techniques used for the identification, authentication, and authorization of users accessing critical assets and resources.

At the end of this module, you will be able to do the following:

- Understand the terminology, principles, and types of access control
- Describe identity and access management (IAM)
- Understand user access management
- Explain the different types of authentication
- Explain the different types of authorization
- Understand user accounting

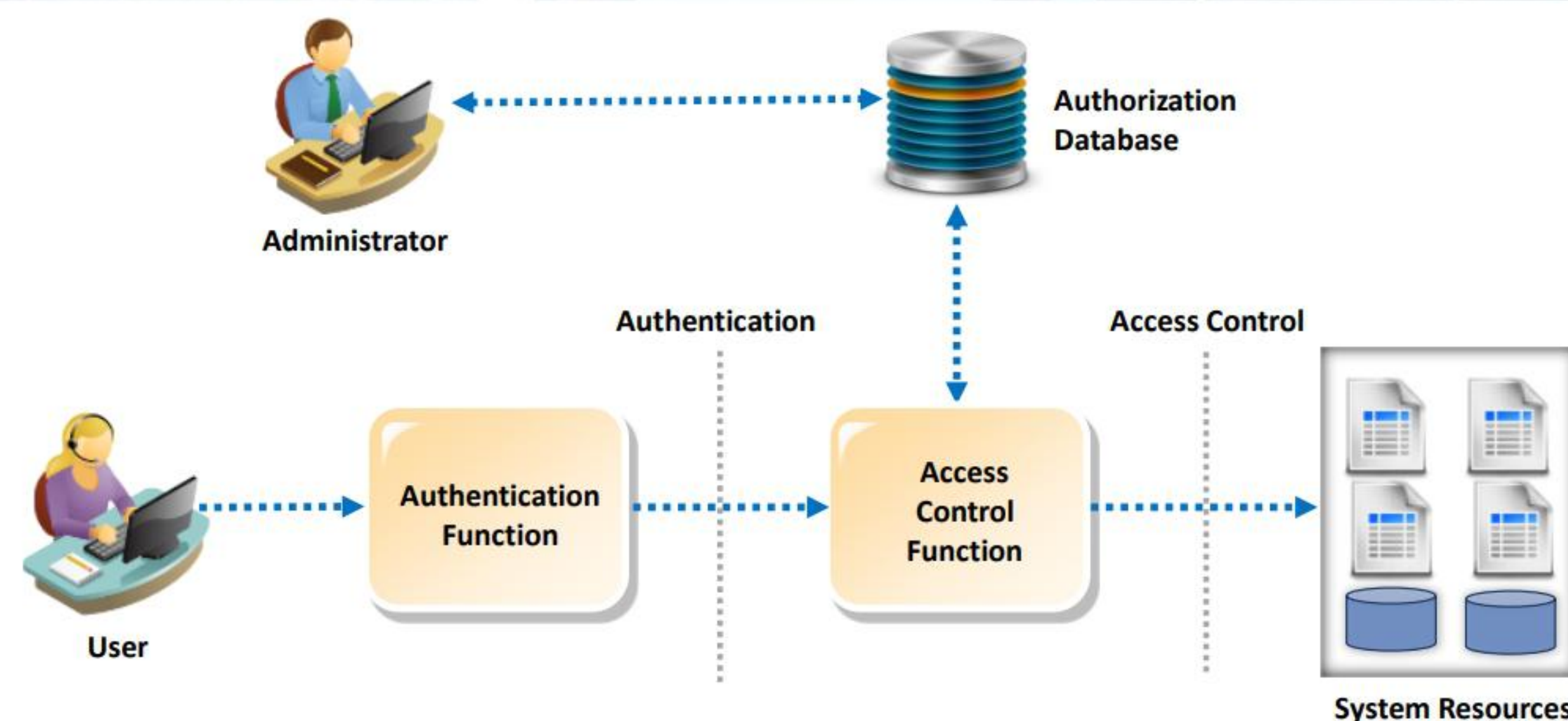


Discuss Access Control Principles, Terminologies, and Models

The objective of this section is to explain the concept of access control by introducing the principles of access control, the terminologies used, and the different models that describe how access control helps in controlling the access of users to specific resources in a network.

Access Control

- ❑ Access control is the **selective restriction** of access to an asset or a system/network resource
- ❑ It **protects the information assets** by determining who can access what
- ❑ Access control mechanism uses **user identification**, **authentication**, and **authorization** to restrict or grant access to a specific asset/resource



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Access Control

Access control is a method of limiting the access of an organization's resources for the users. A crucial aspect of implementing an access control is to maintain the integrity, confidentiality, and availability of the information.

An access control function uses identification, authentication, and mechanisms to identify, authenticate, and authorize the user requesting access to a specific resource. The access permissions determine the approvals or permissions provided to a user for accessing a system and other resources.

The general steps involved in the access control mechanism are as follows:

- **Step 1:** A user provides their credentials/identification while logging into the system.
- **Step 2:** The system validates the user with the database on the basis of the provided credentials/identification such as a password, fingerprint, etc.
- **Step 3:** Once the identification is successful, the system provides the user access to use the system.
- **Step 4:** The system then allows the user to perform only those operations or access only those resources for which the user has been authorized.

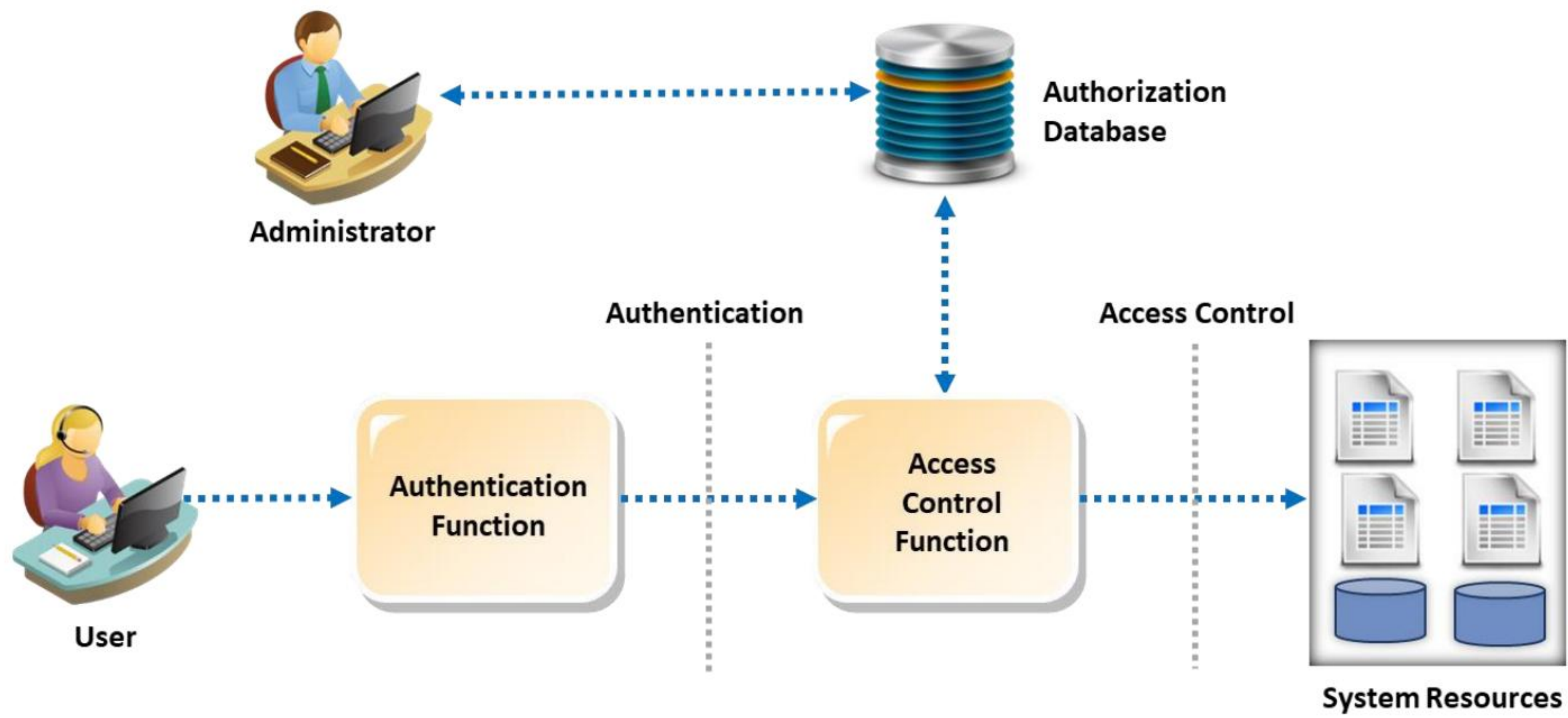
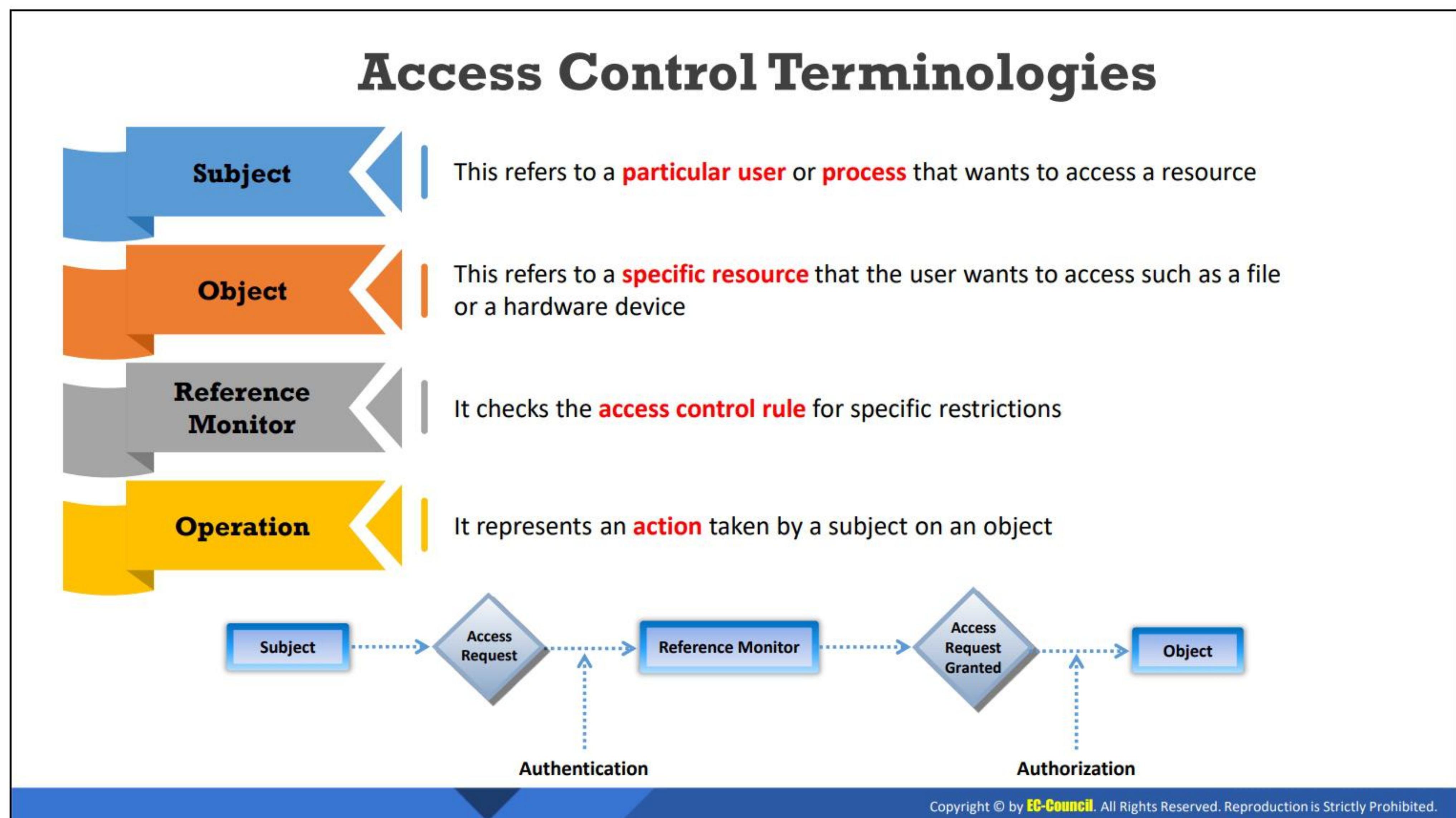


Figure 4.1: Access Control Mechanism



Access Control Terminologies

The following terminologies are used to define the access control on specific resources:

- **Subject**

A subject can be defined as a user or a process that attempts to access the objects. The subjects are those entities that perform certain actions on the system.

- **Object**

An object is an explicit resource on which an access restriction is imposed. The access controls implemented on the objects further control the actions performed by the user. Examples of an object are a file or a hardware device.

- **Reference Monitor**

A reference monitor monitors the restrictions imposed on the basis of certain access control rules. It implements a set of rules on the ability of the subject to perform certain actions on the object.

- **Operation**

An operation is an action performed by a subject on an object. A user trying to delete a file is an example of an operation. Here, the user is the subject, the action of deleting refers to the operation, and the file is the object.

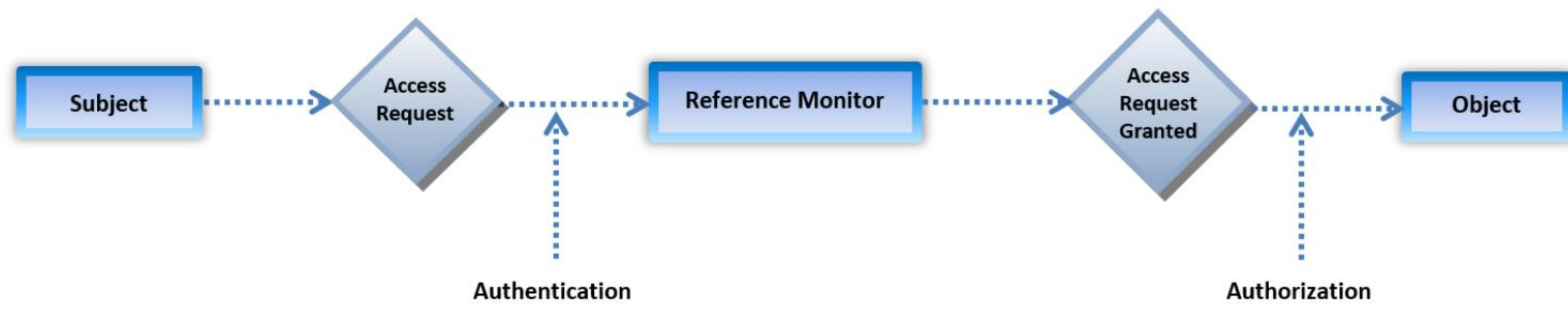


Figure 4.2: Access Control Terminologies

Access Control Principles

Separation of Duties (SoD)

- Involves a breakdown of the authorization process into various steps
- Different privileges are assigned at each step to the **individual subjects** requesting for a resource
- This ensures that no single individual has the authorization rights to perform all functions and simultaneously **denies access** of all the objects to a single individual



Need-to-know

- Under the need-to-know access control principle, access is provided only to the **information** that is required for performing a specific task



Principle of Least Privilege (POLP)

- Principle of least privilege extends the **need-to-know** principle in providing access to a system
- POLP believes in providing employees a need-to-know access, i.e., not more, not less;
- It helps an organization by protecting it from malicious behavior, achieving better **system stability**, and system security



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Access Control Principles

The principles of access control describe the access permission levels of users in detail. By enabling the access control process, the security of the processes and resources can be ensured. The process of access control should be based on the following principles:

▪ Separation of Duties (SoD)

This involves a breakdown of the authorization process into various steps. Different privileges are assigned at each step to the individual subjects requesting for a resource. This ensures that no single individual has the authorization rights to perform all functions and simultaneously denies access of all the objects to a single individual. This division ensures that a single person is not responsible for a larger process. For example, granting web server administrator rights to only configure a web server without granting administrative rights to other servers.

▪ Need-to-know

Under the need-to-know access control principle, access is provided only to the information that is required for performing a specific task.

▪ Principle of Least Privilege (POLP)

The principle of least privilege (POLP) extends the need-to-know principle in providing access to a system. In other words, POLP is based on providing employees exactly the need-to-know level of access i.e., not more and not less. It helps an organization by protecting it from malicious behavior as well as improving system stability and system security.

Least privilege provides access permissions to only those users who really need the access and resources. The permissions granted depend on the roles and responsibilities of the user requesting the access. There are two underlying principles involved in the least privilege method: low rights and low risks. On the basis of these principles, a user needs to complete a task using the limited number of resources in a limited amount of time provided to them. This approach reduces the probability of unauthorized access to the system resources.

Access Control Models



- ❑ Access control models are the **standards which provide a predefined framework** for implementing the necessary level of access control



Mandatory Access Control (MAC)

- ✓ Only the administrator/system owner has the rights to assign privileges
- ✓ It does not permit the end user **to decide who can access the information**



Discretionary Access Control (DAC)

- ✓ End user has complete access to the information they own



Role-based Access Control (RBAC)

- ✓ Permission are assigned based on user roles



Rule-based Access Control (RB-RBAC)

- ✓ Permissions are assigned to a user role dynamically based on a set of rules defined by the administrator

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Access Control Models

Access control models are the standards which provide a predefined framework for implementing the necessary level of access control. Access control models specify how a subject can access an object.

▪ **Mandatory Access Control**

The mandatory access control (MAC) determines the usage and access policies for the users. A user can access a resource only if they have the access rights to that resource. MAC is applied in the case of data that has been marked as highly confidential. The administrators impose MAC depending on the operating system and the security kernel. It does not permit the end-user to decide who can access the information.

The following are the advantages and disadvantages of MAC:

- It provides a high level of security since the network defenders determine the access controls.
- The MAC policies minimize the chances of errors.
- Depending on the MAC, an operating system marks and labels the incoming data, thereby creating an external application control policy.

Examples of MAC include Security-Enhanced Linux (SELinux) and Trusted Solaris.

▪ **Discretionary Access Control**

Discretionary access control (DAC) determines the access control taken by any possessor of an object in order to decide the access control of a subject on that object. DAC is

alternatively named as a **need-to-know** access model. The decision taken by the owner depends on the following measures:

- **File and data ownership:** Determines the access policies of the user
- **Access rights and permissions:** Involves the possessor setting the access privileges to other subjects

An owner can provide or deny access to any particular user or a group of users. The attributes of a DAC include the following:

- The owner of an object can transfer the ownership to another user.
- The access control prevents multiple unauthorized attempts to access an object.
- The DAC prevents unauthorized users from viewing details like the file size, filename, directory path, etc.
- The DAC uses access control lists in order to identify and authorize users.

Disadvantage: A DAC requires maintenance of the access control list and access permissions for the users. Examples of DAC include UNIX, Linux, and Windows access control.

▪ **Role-Based Access Control**

In a role-based access control (RBAC), the access permissions are available based on the access policies determined by the system. The access permissions are beyond the user control which implies that users cannot amend the access policies created by the system. The rules for determining the role-based access controls are as follows:

- **Role assignment:** A certain role is required to be assigned to a user which enables them to perform a transaction.
- **Role authorization:** A user needs to perform a role authorization in order to achieve a particular role.
- **Transaction authorization:** Transaction authorization allows the users to execute only those transactions for which they have been authorized.

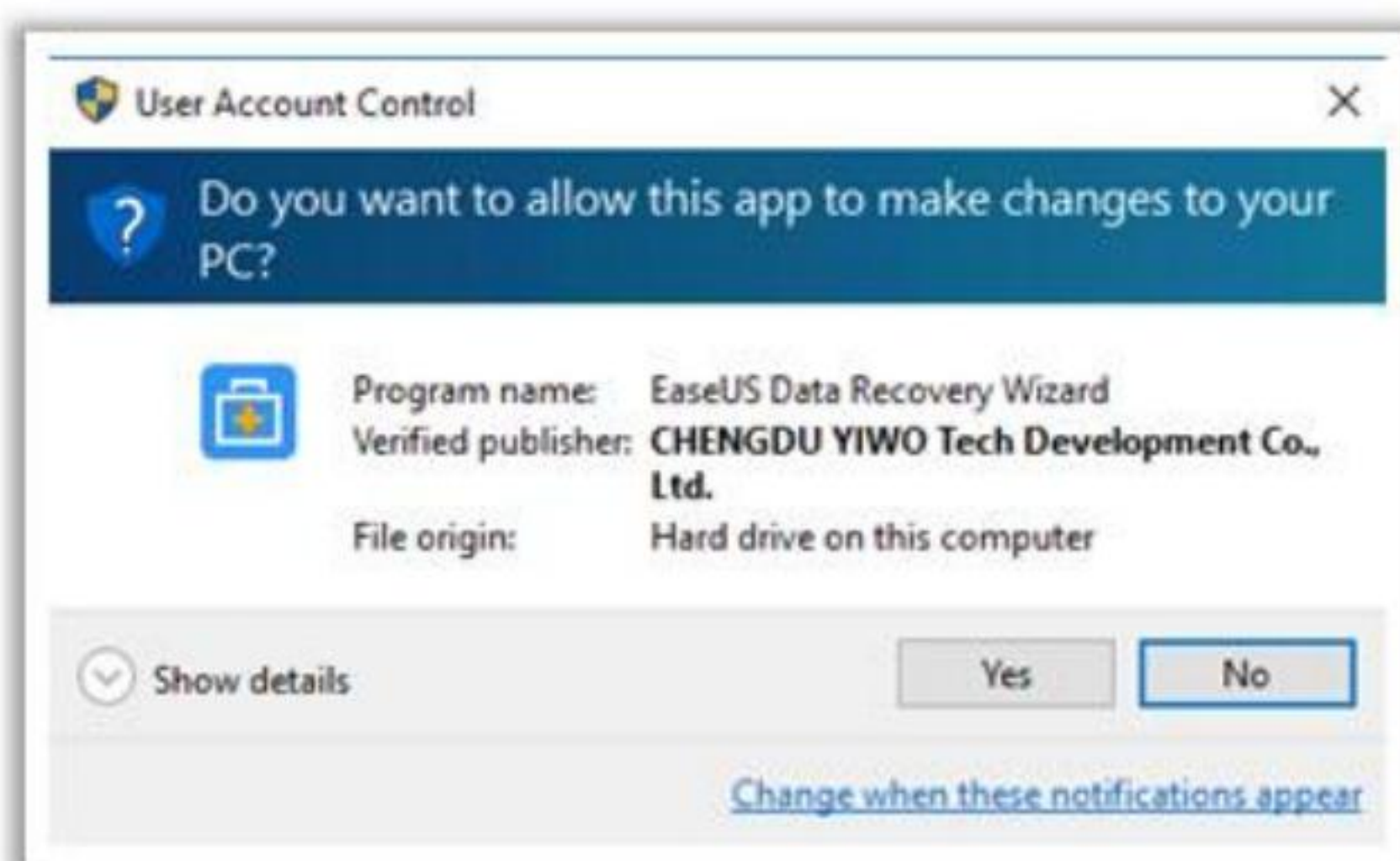
▪ **Rule-based Access Control (RB-RBAC)**

Permissions are assigned to a user role dynamically based on a set of rules defined by the administrator.

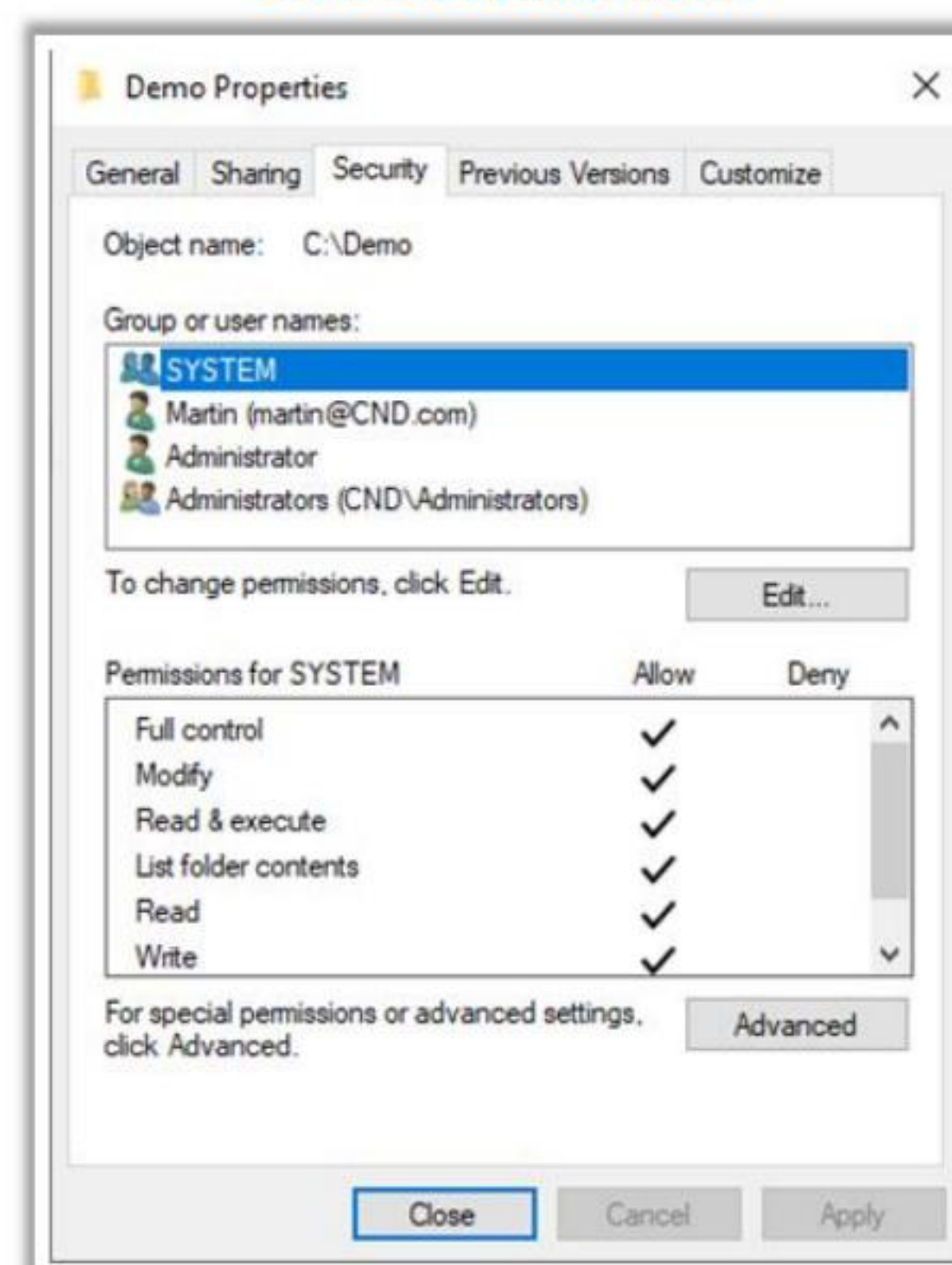
Logical Implementation of DAC, MAC, and RBAC

- Logical implementation of access control is performed using **access control lists (ACLs)**, **group policies**, **passwords**, and **account restrictions**

MAC Implementation: The User Account Control (UAC) tool of Windows OS



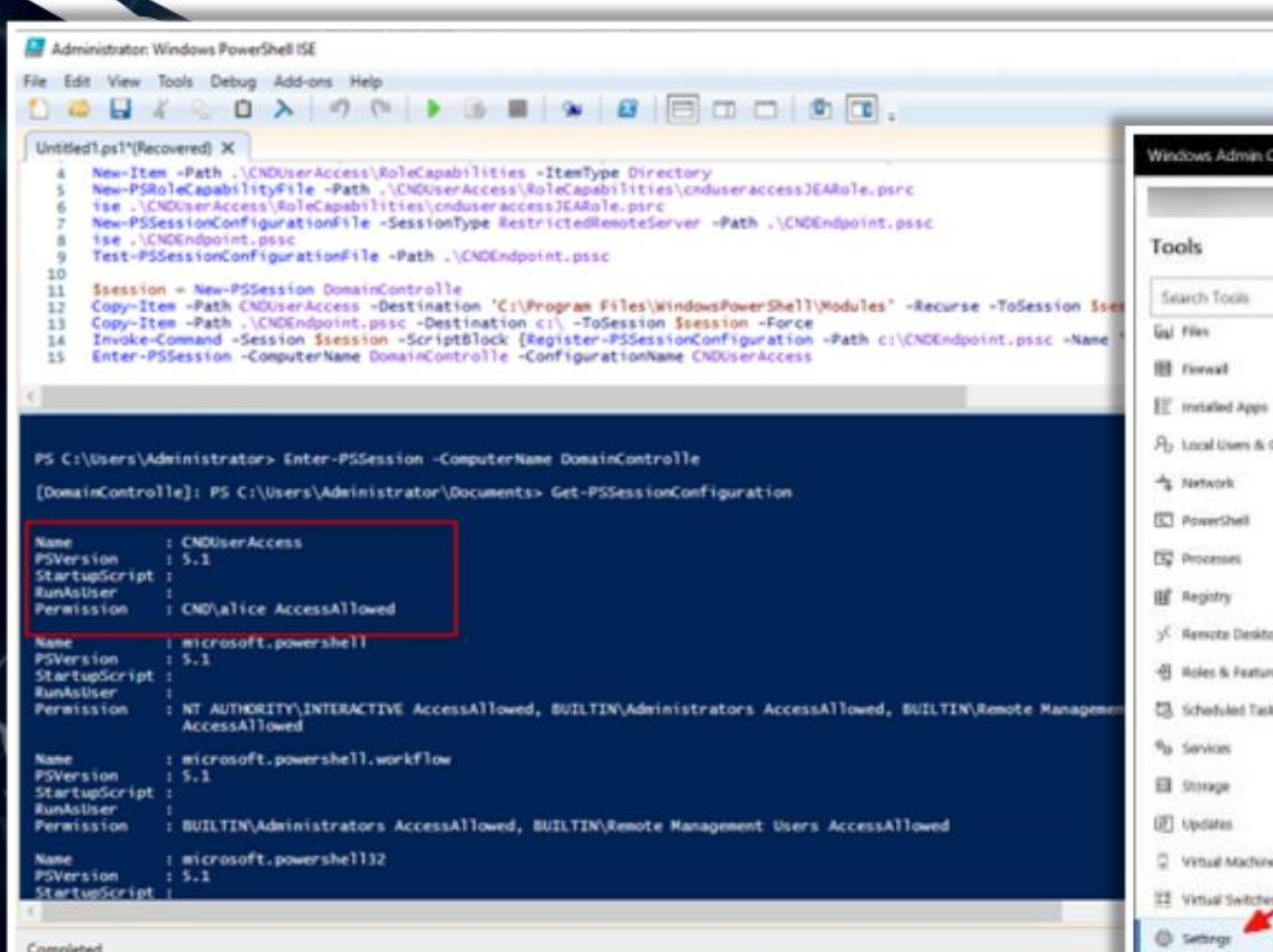
DAC Implementation: Windows File Permissions



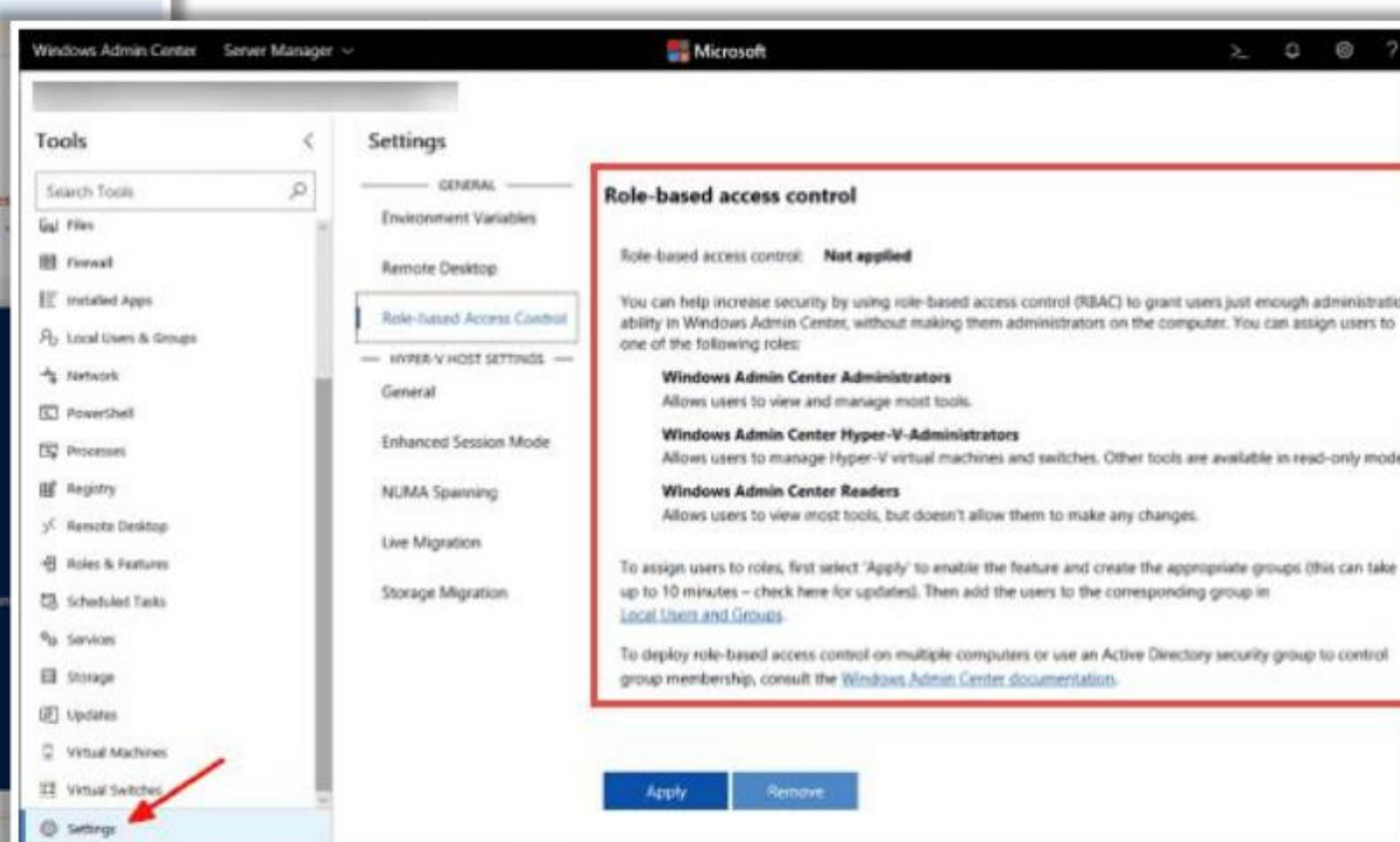
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Logical Implementation of DAC, MAC, and RBAC (Cont'd)

RBAC Implementation: Just Enough Administration (JEA)



RBAC Implementation: Windows Admin Center (WAC)



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Logical Implementation of DAC, MAC, and RBAC

In the Windows operating system (OS), the User Account Control (UAC) feature implements the MAC security model. It restricts the installation of any application software only through administrator authorizations. In other words, users without administrative privileges are restricted to install any application on the system.

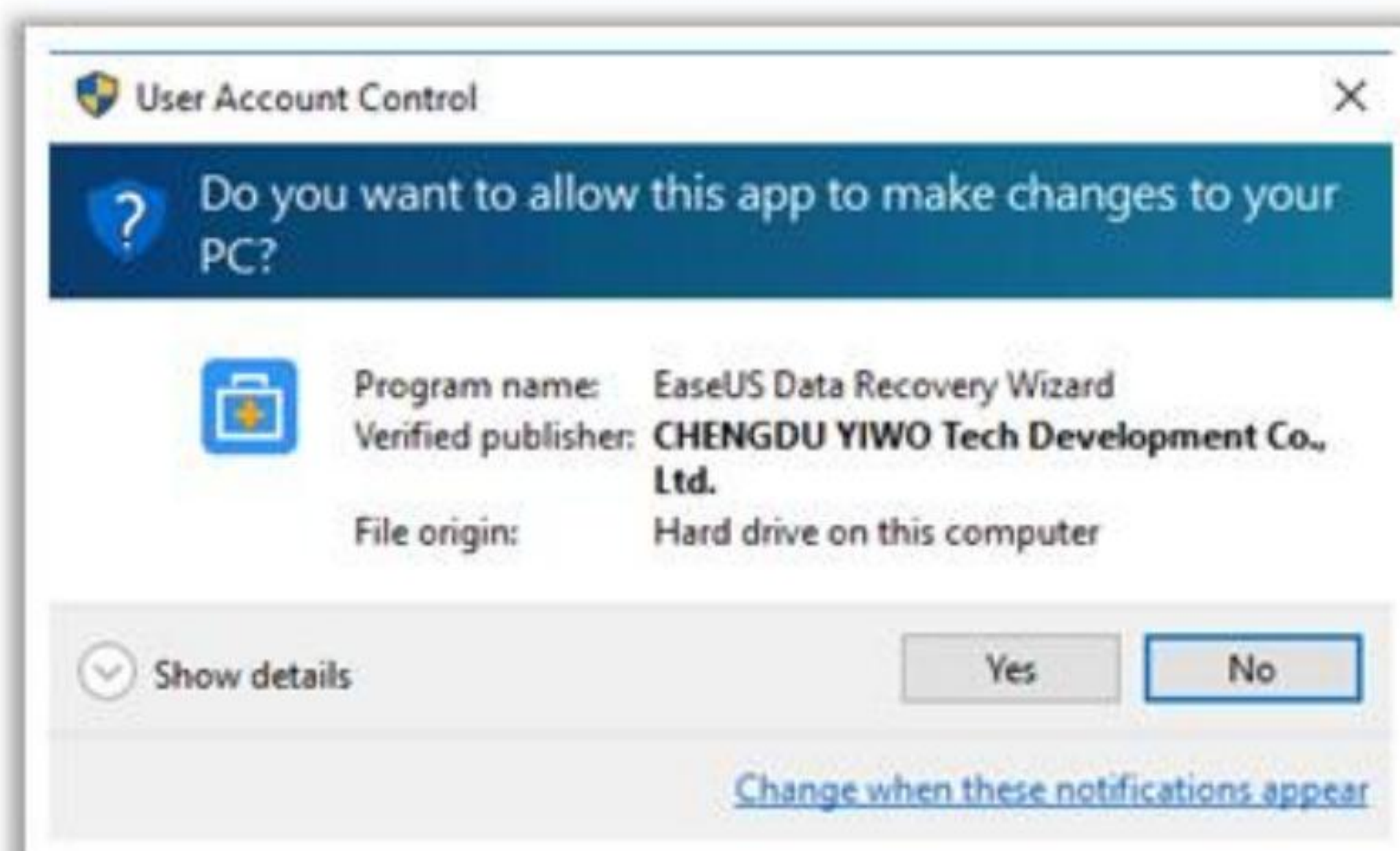


Figure 4.3: Mac Implementation: The User Account Control tool of Windows OS

Logical Implementation of DAC: Windows File Permissions

In the Windows OS, DAC is implemented for assigning file permissions to specific groups/users. Permissions to access files and folders on a system, to access files that exist on an old account of a user, or to edit system files are all controlled using DAC.

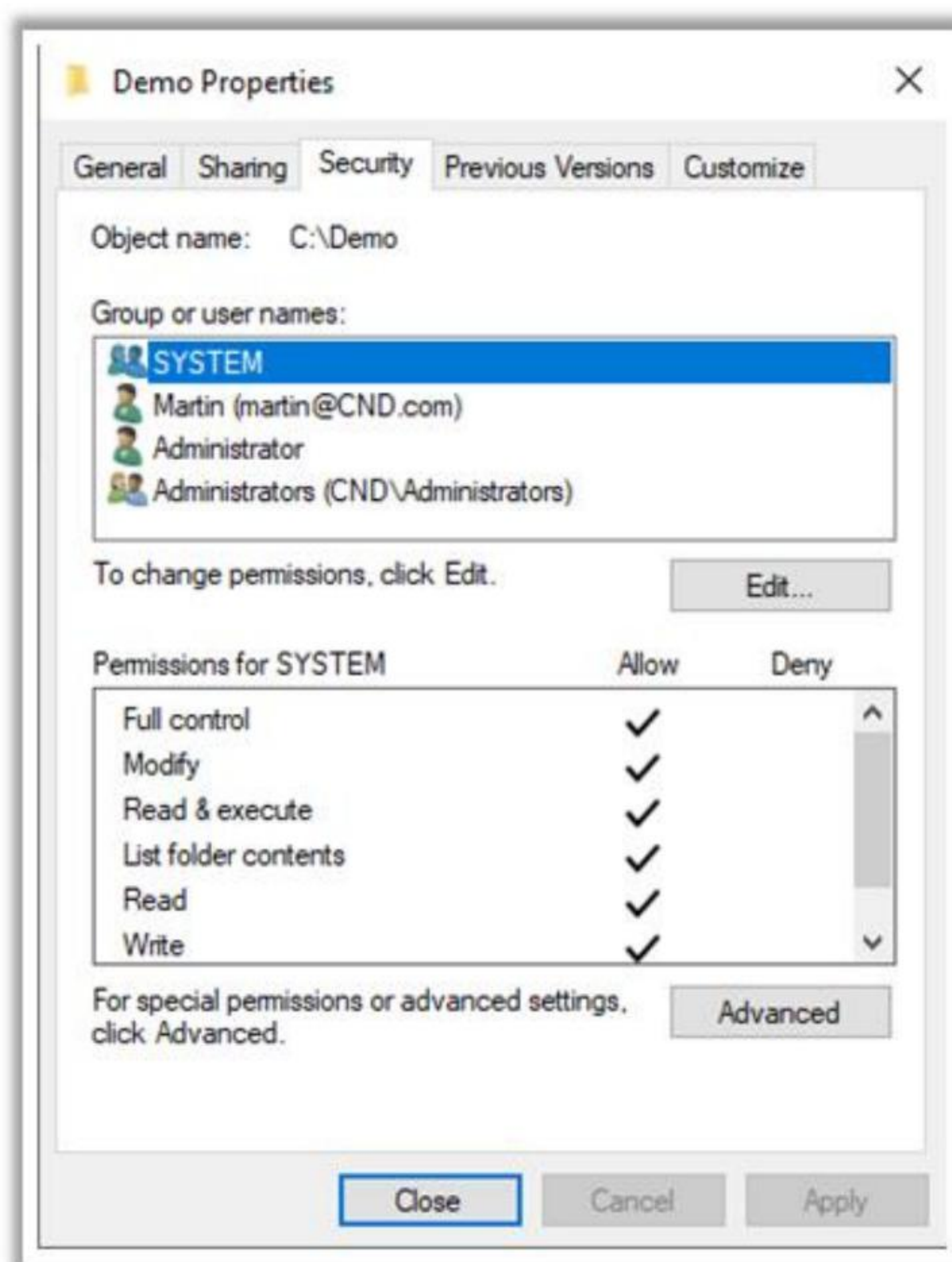
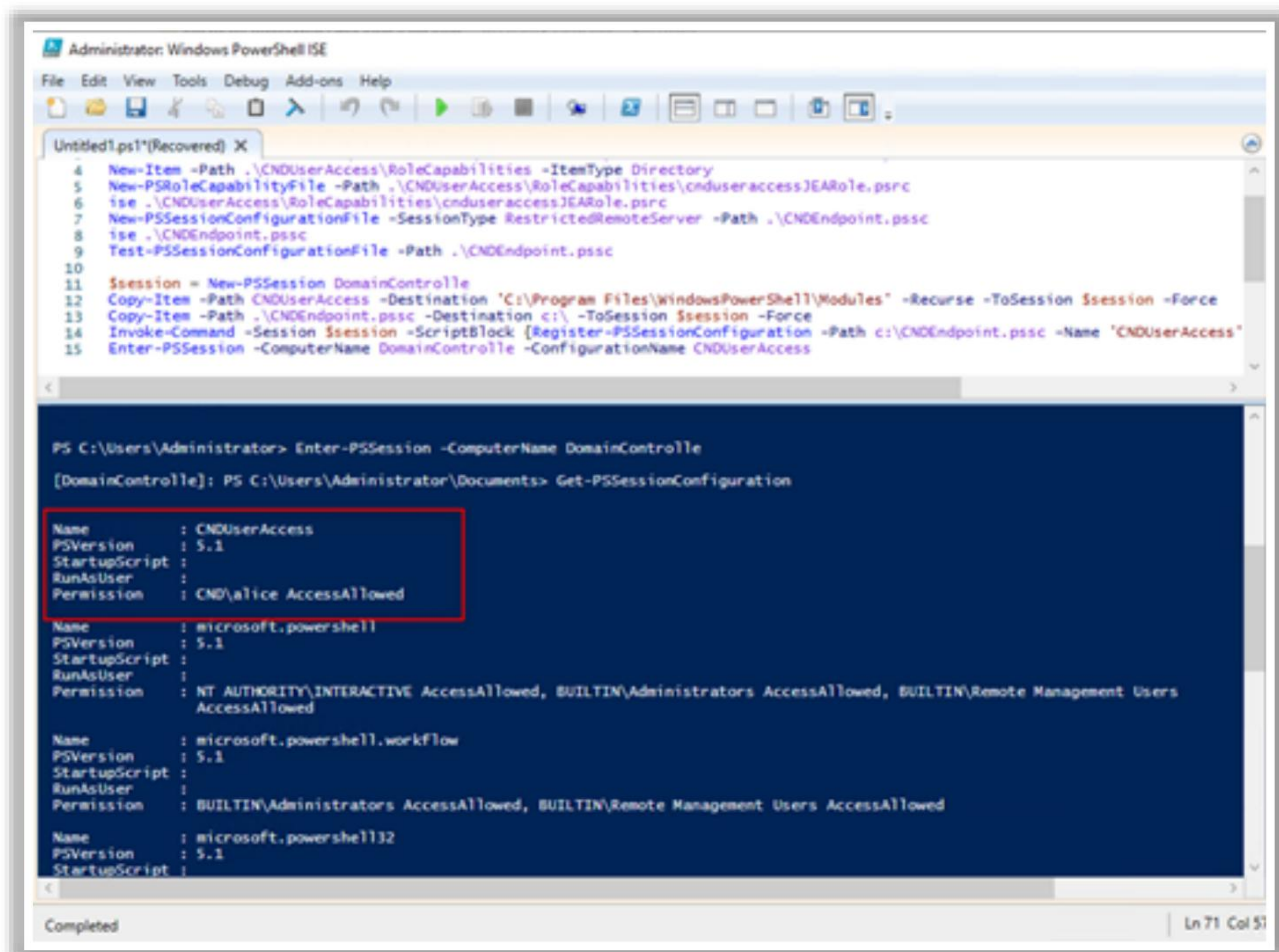


Figure 4.4: DAC Implementation: Windows File Permissions

Logical Implementation of RBAC: Just Enough Administration (JEA)

The Just Enough Administration (JEA) management framework of in the Windows OS implements RBAC to restrict the rights of IT administrators in remote PowerShell sessions. Using JEA a fine-grained access control can be implemented for non-administrators to run specific commands, scripts, and executables.



The screenshot shows the Windows PowerShell ISE interface. The top pane contains a PowerShell script for creating and configuring a role-based access control (RBAC) role named 'CNDUserAccess'. The bottom pane shows the output of the 'Enter-PSSession' and 'Get-PSSessionConfiguration' commands. A red box highlights the output for the 'CNDUserAccess' session configuration.

```
4 New-Item -Path .\CNDUserAccess\RoleCapabilities -ItemType Directory
5 New-PSRoleCapabilityFile -Path .\CNDUserAccess\RoleCapabilities\cnduseraccessJEARole.psnc
6 ise .\CNDUserAccess\RoleCapabilities\cnduseraccessJEARole.psnc
7 New-PSSessionConfigurationFile -SessionType RestrictedRemoteServer -Path .\CNDEndpoint.pssc
8 ise .\CNDEndpoint.pssc
9 Test-PSSessionConfigurationFile -Path .\CNDEndpoint.pssc
10
11 $session = New-PSSession DomainControlle
12 Copy-Item -Path CNDUserAccess -Destination 'C:\Program Files\WindowsPowerShell\Modules' -Recurse -ToSession $session -Force
13 Copy-Item -Path .\CNDEndpoint.pssc -Destination c:\ -ToSession $session -Force
14 Invoke-Command -Session $session -ScriptBlock [Register-PSSessionConfiguration -Path c:\CNDEndpoint.pssc -Name 'CNDUserAccess'
15 Enter-PSSession -ComputerName DomainControlle -ConfigurationName CNDUserAccess

PS C:\Users\Administrator> Enter-PSSession -ComputerName DomainControlle
[DomainControlle]: PS C:\Users\Administrator\Documents> Get-PSSessionConfiguration

Name       : CNDUserAccess
PSVersion  : 5.1
StartupScript :
RunAsUser   :
Permission  : CND\alice AccessAllowed

Name       : microsoft.powershell
PSVersion  : 5.1
StartupScript :
RunAsUser   :
Permission  : NT AUTHORITY\INTERACTIVE AccessAllowed, BUILTIN\Administrators AccessAllowed, BUILTIN\Remote Management Users
AccessAllowed

Name       : microsoft.powershell.workflow
PSVersion  : 5.1
StartupScript :
RunAsUser   :
Permission  : BUILTIN\Administrators AccessAllowed, BUILTIN\Remote Management Users AccessAllowed

Name       : microsoft.powershell32
PSVersion  : 5.1
StartupScript :
```

Figure 4.5: RBAC Implementation: Just Enough Administration (JEA)

Logical Implementation of RBAC: Windows Admin Center (WAC)

Windows Admin Center (WAC) is a tool which helps configure a role-based access control for managing a server. The concept of a role is based on JEA, which enables granting the required rights to non-administrators.

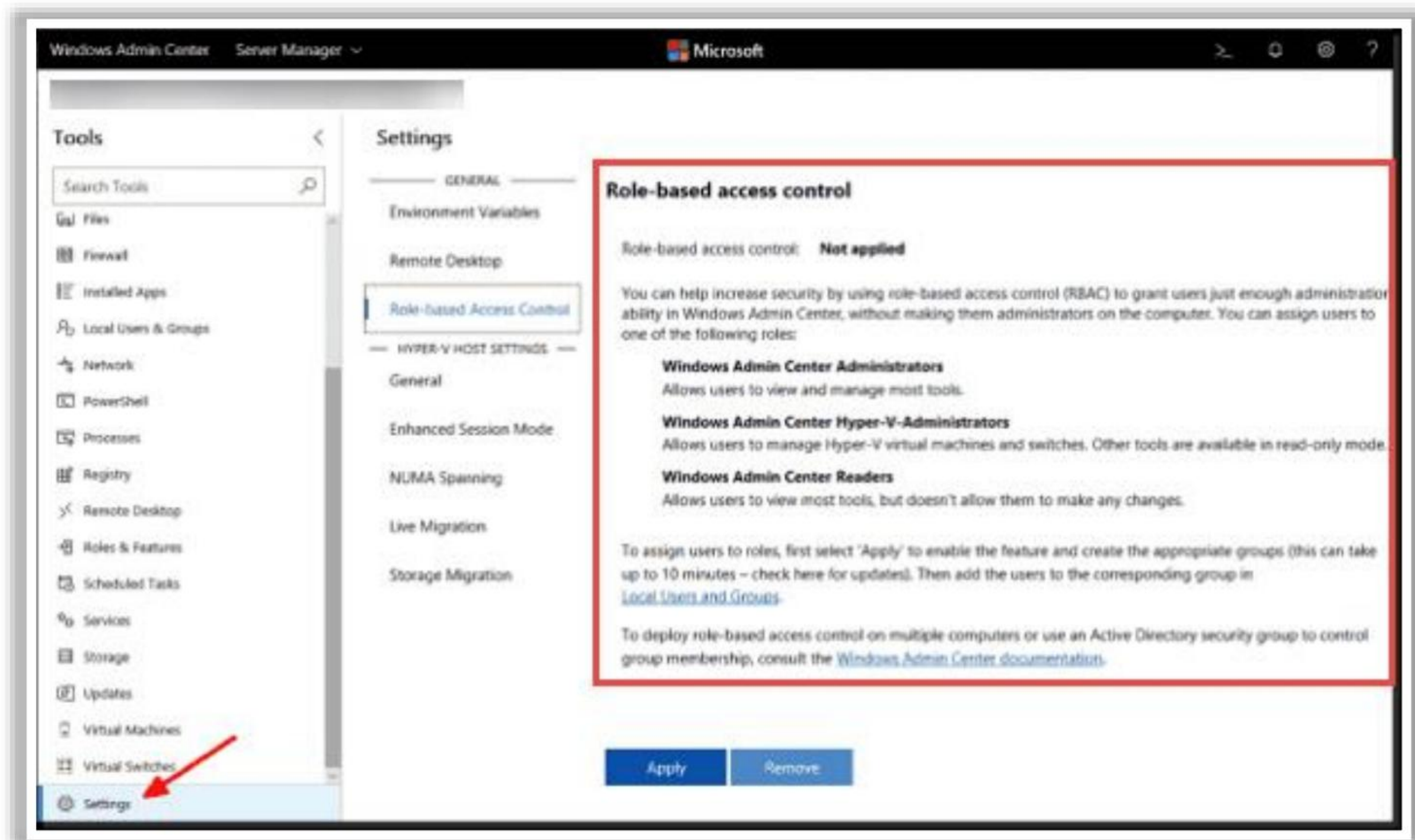
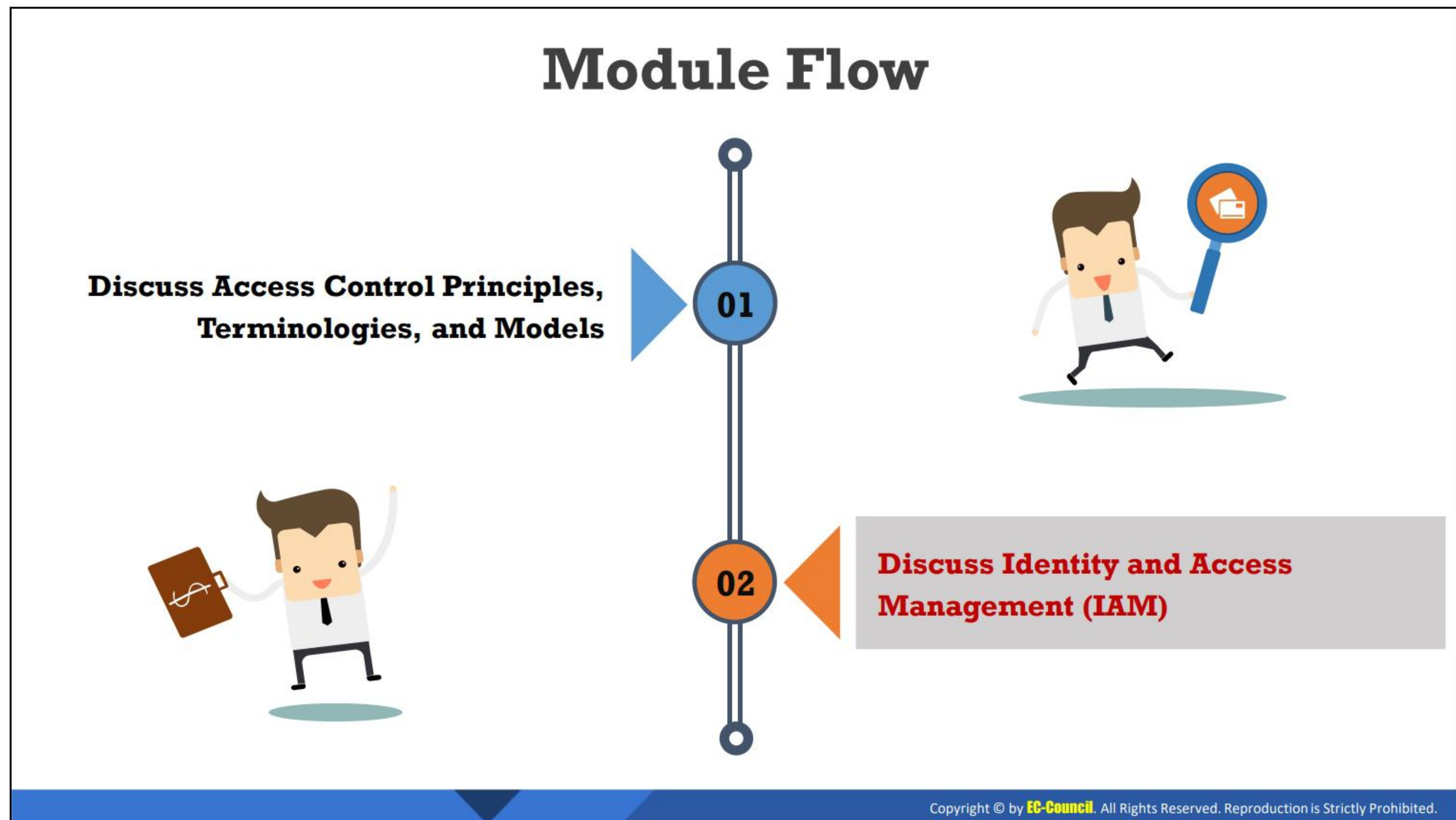


Figure 4.6: RBAC Implementation: Windows Admin Center (WAC)

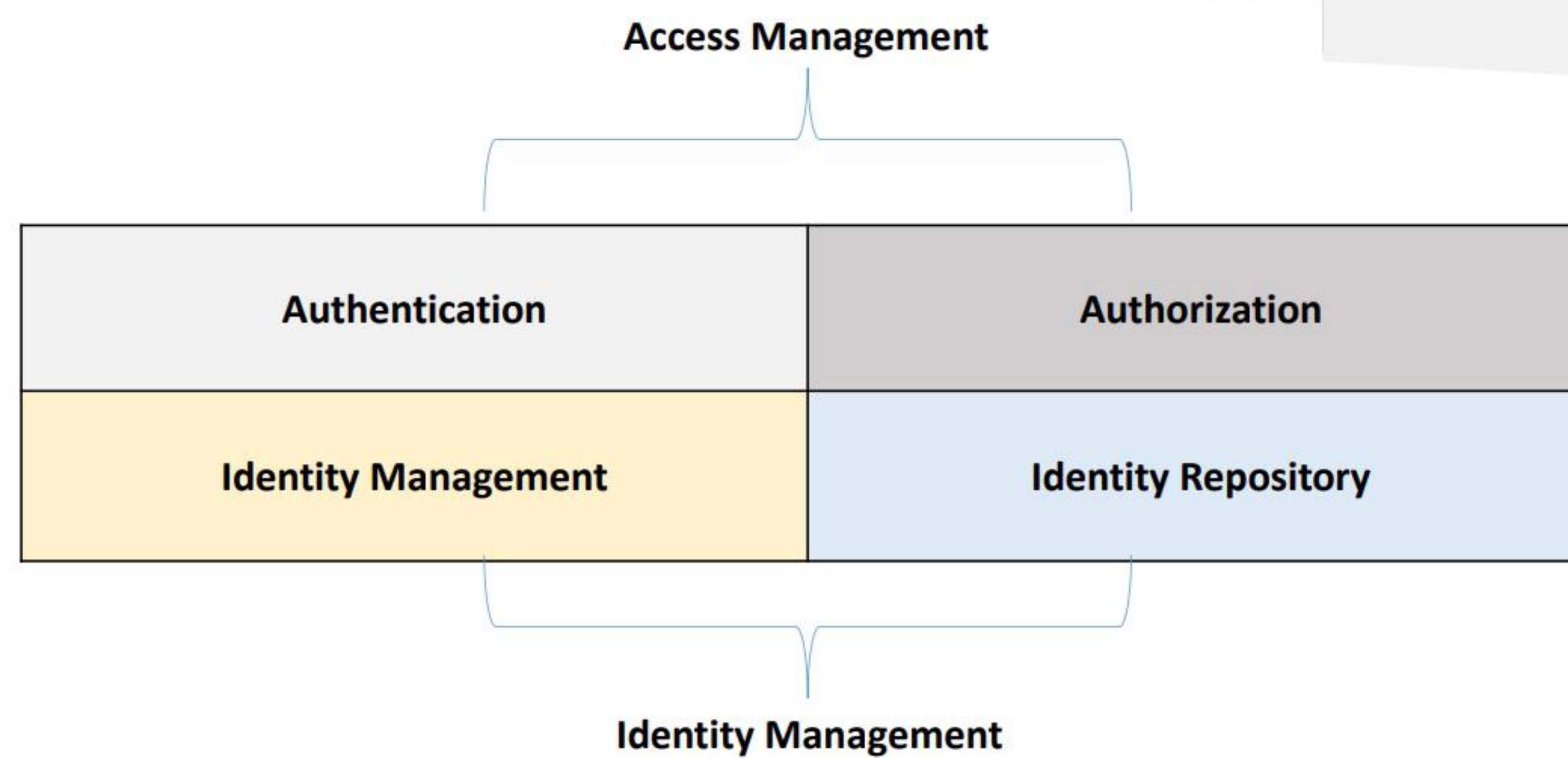


Discuss Identity and Access Management (IAM)

In an enterprise security, Identity and Access Management (IAM) plays an important role. It ensures that only authorized users have access to the network resources. The objective of this section is to explain the role of IAM and the security terminologies associated with it.

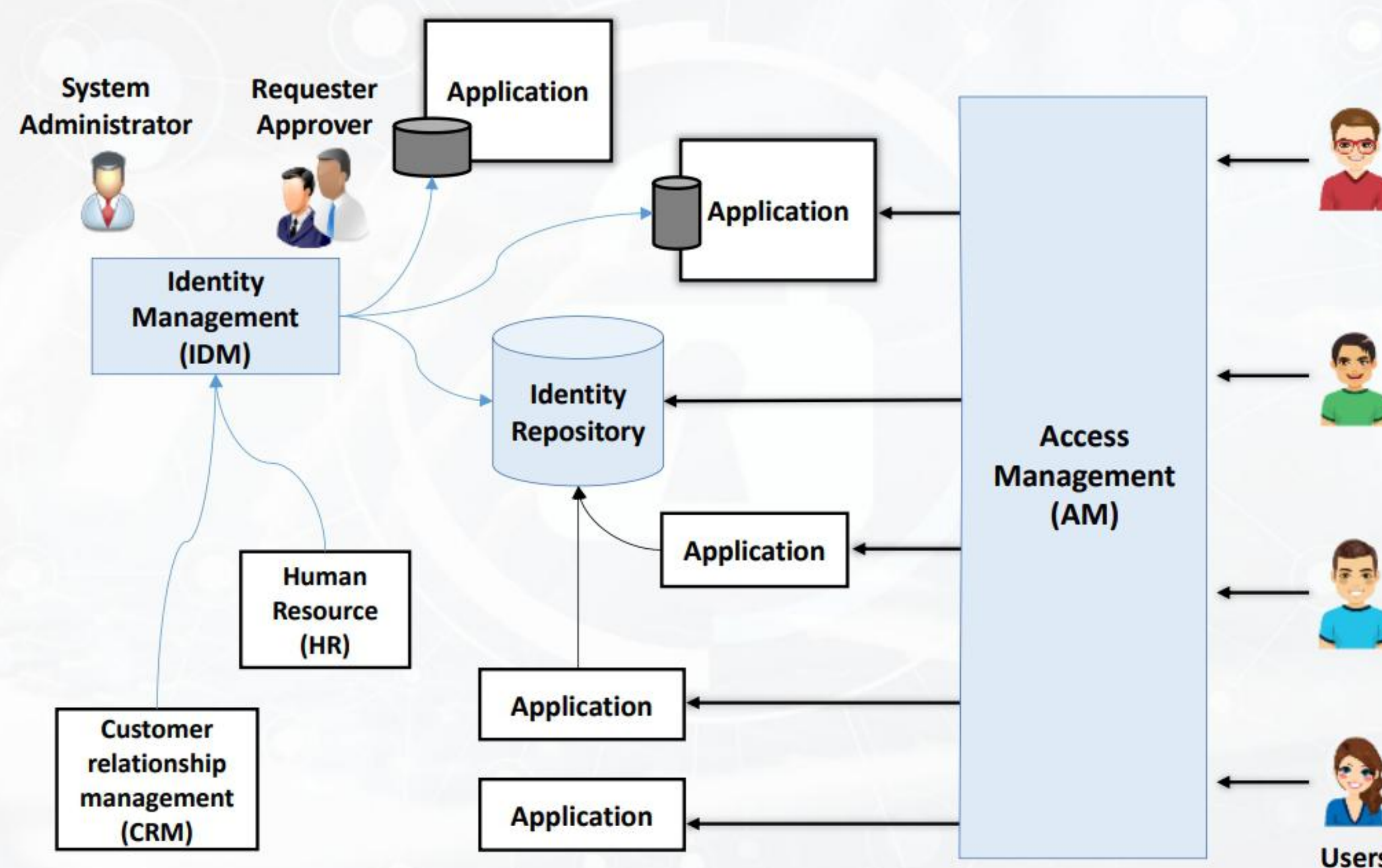
Identity and Access Management (IAM)

- ❑ IAM is responsible for providing the **right individual with right access at the right time**



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Identity and Access Management (IAM) (Cont'd)



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Identity and Access Management (IAM)

Identity and access management (IAM) is responsible for providing the right individual with the right access at the right time. It offers a role-based access control to the customers or employees of an organization for accessing critical information within the enterprise. It comprises of business processes, policies, and technologies that allow monitoring electronic or digital identities. IAM products provide the system administrators with tools and technologies for regulating user access (i.e., creating, managing, and removing access) to systems or

networks based on the roles of individual users within the enterprise. Organizations generally prefer an all-in-one authentication implementation which can be extended to identity a federation. This is because the identity federation includes IAM with a single sign-on (SSO) and a centralized active directory (AD) account for a secured management.

Organizations should ensure the correctness of data for the proper functioning of the IAM framework. An IAM framework can be divided into four areas, namely, authentication, authorization, user management, and central user repository/identity repository. All the IAM components are grouped under these four areas.

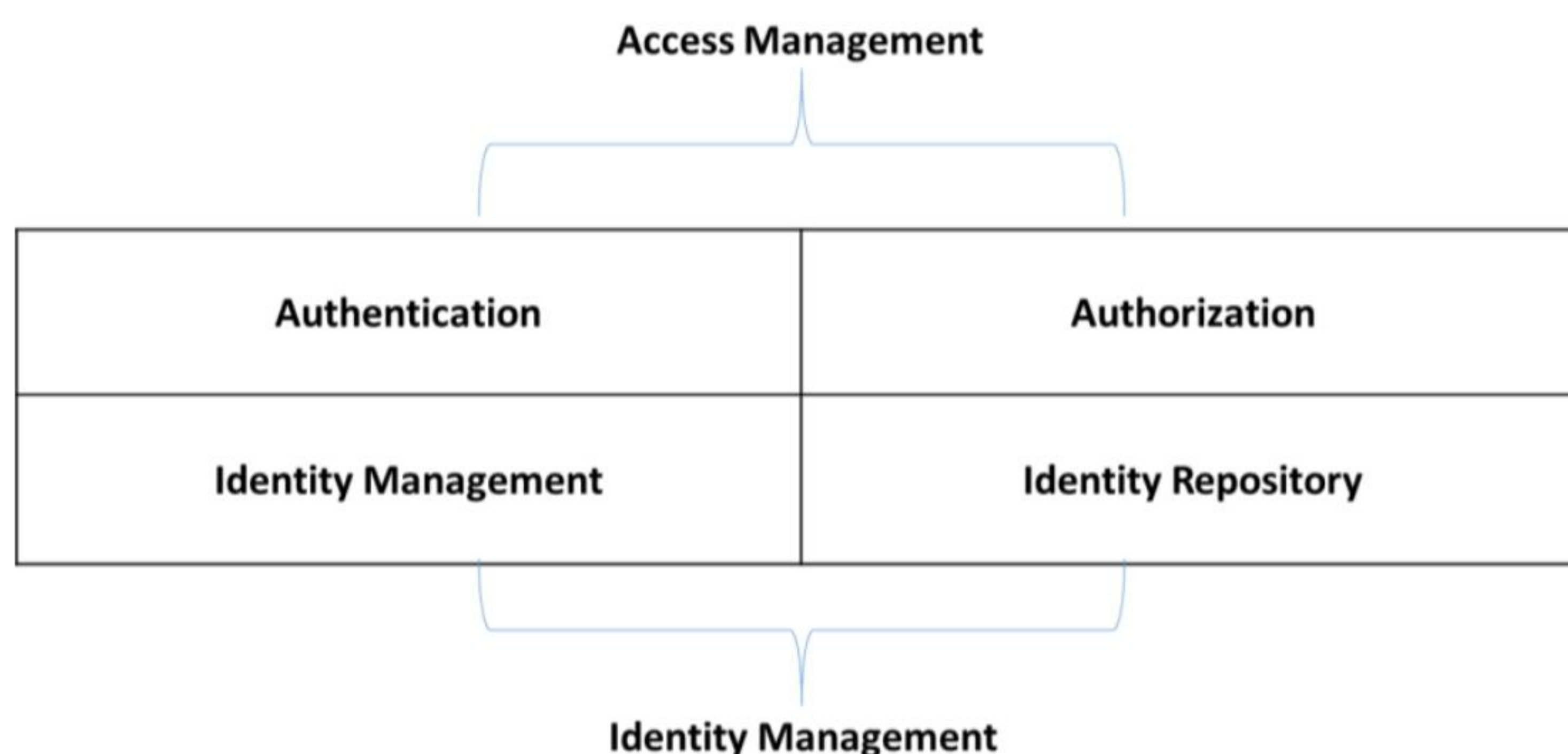


Figure 4.7: IAM Classification

Working of an IAM:

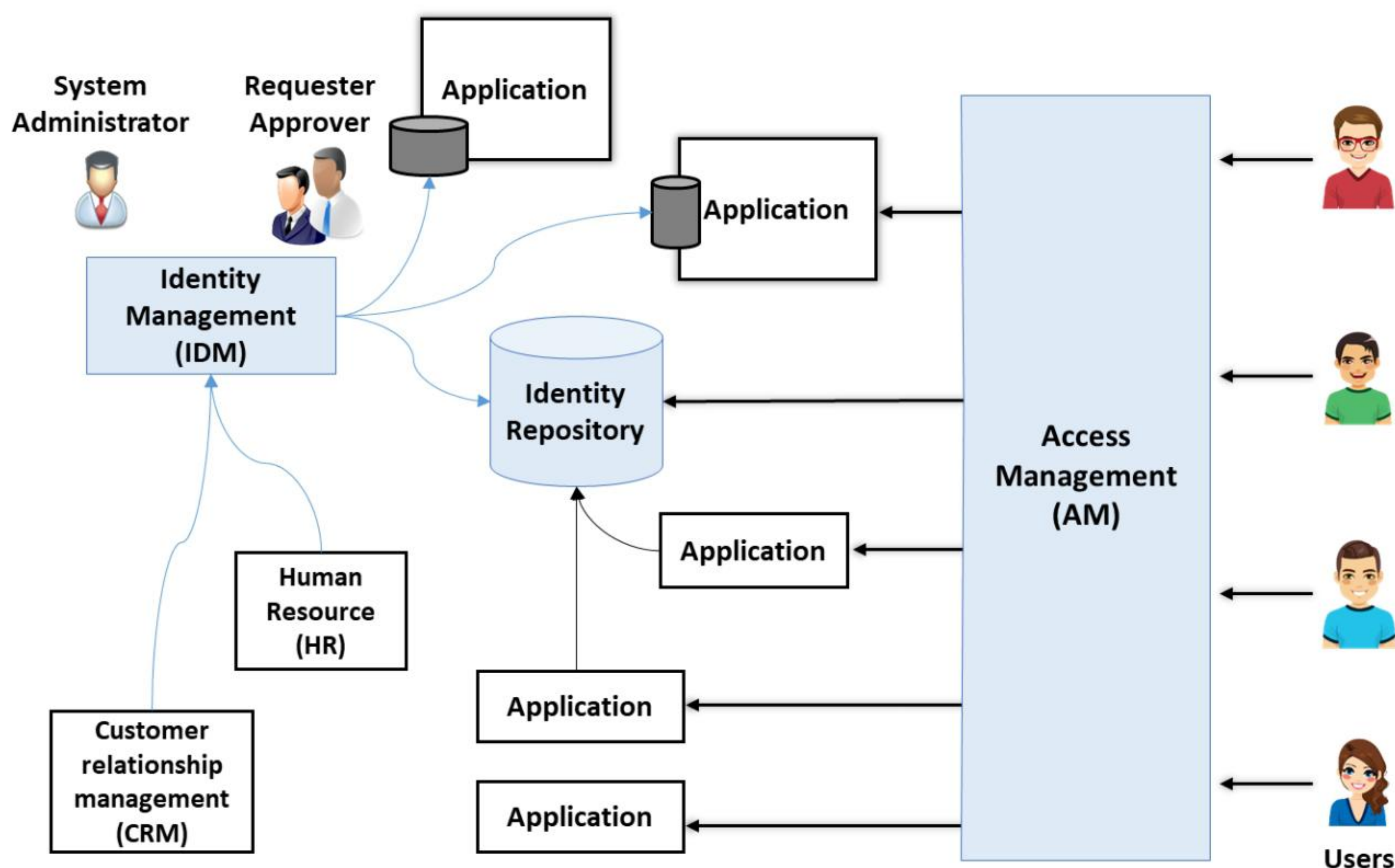
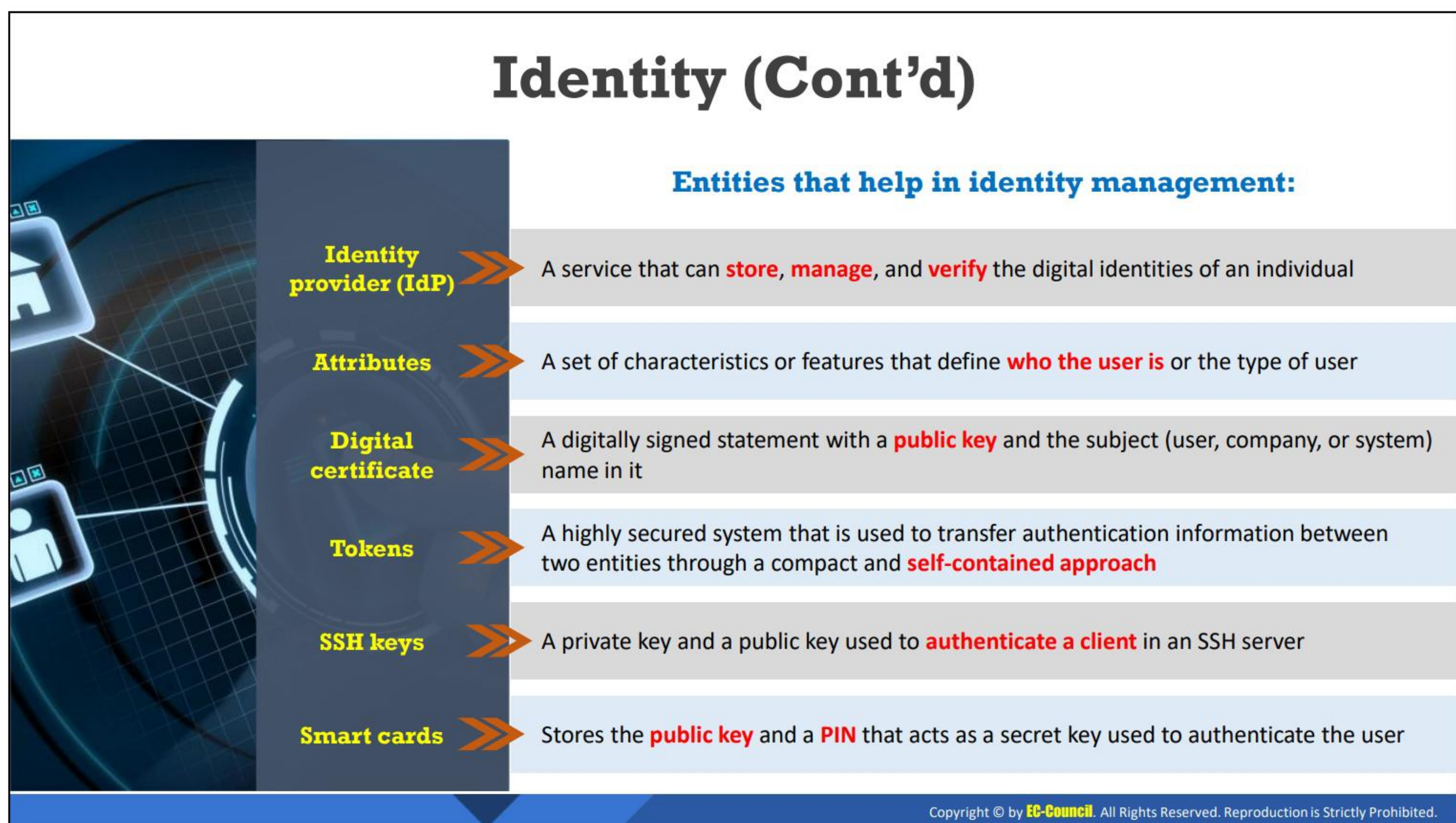
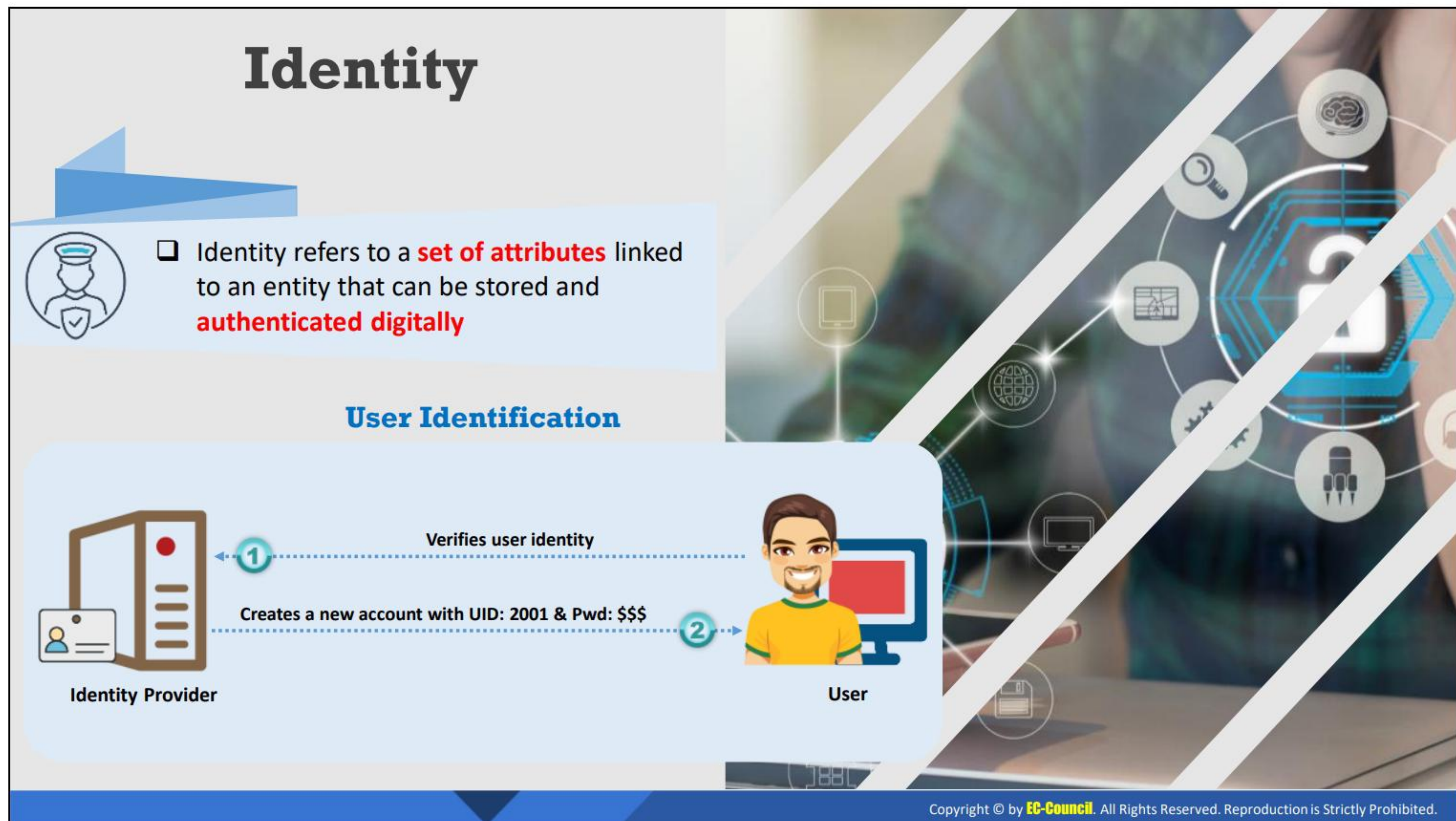


Figure 4.8: Working of IAM

The key responsibility of the identity management (IDM) framework is to manage the shared identity repository that is being accessed by the applications and the access management system.



Identity

Identity refers to a unique collection of features and characteristics that is related to a distinctively identifiable person, organization, or electronic device; identity can be stored and authenticated digitally. It is also referred to as a set of attributes linked to an entity and can be called a digital identity. User identity is related to authentication factors, namely, knowledge, possession, and innate qualities such as fingerprints or retina scans. Most of authentication, authorization, and accounting (AAA) systems depend on the identity to ensure that network

resources are secured. Digital identity helps in performing online transactions and other types of user interactions on the Internet more easily and securely.

The following are the set of entities that help in identity management.

- **Identity Provider (IdP)**

An identity provider (IdP) is a service that can store, manage, and verify digital identities of an individual. IdP can provide a single set of login credentials to the users and devices that are connected to the Internet to verify the entities through various networks, applications, and platforms. IdP also checks for the user identities through username and password combinations, as in single-sign-on (SSO) checks.

In a private network, identity management can be performed locally using services such as identity directories and application authorization. Many organizations now employ cloud-based identity services, which implement a variety of frameworks and protocols across various web-based services. These services also allow users to create one identity and reuse that identity during the authorization of different applications.

- **Attributes**

Identity attributes consist of a set of characteristics or features that define who the user is or the type of user, such as employee, customer, or client. The attributes also include the user's personal metadata such as name, DOB, fingerprints, job role, company, and contact details.

- **Digital certificate**

Public key infrastructure (PKI) is a comprehensive system that allows the use of public-key encryption and digital signature services across a wide variety of applications. PKI authentication depends on digital certificates that certification authorities (CAs) sign and provide. A digital certificate is a digitally signed statement with a public key and the subject name (user, company, or system).

A CA simultaneously generates public and private keys with the same algorithm. The private key is held only by the subject mentioned in the certificate, while the public key is made publicly available in a directory that all parties can access. The subject keeps the private key secret and uses it to decrypt the text encrypted by someone else using the corresponding public key.

The digital certificate can be stored locally on a computer, smart card, trusted platform module (TPM), or USB and can be used to authenticate various systems and applications.

- **Tokens**

A token is a highly secure system that is used to transfer authentication information between two entities through a compact and self-contained approach. In the current scenario, users need to provide authentication for every application they use. To avoid reentering credentials for every application, many online services have introduced the single-sign-on (SSO) functionality, by which the user provides authentication to an IdP

and the IdP generates and sends a cryptographic token to the user. The user can use the token for authorization with every application that supports SSO. As tokens are vulnerable to replay attacks, applications need to be designed in a secure manner to defend against such attacks.

- **SSH keys**

Secure Shell (SSH) keys are secure keys consisting of a private key and a public key. An SSH key is used to authenticate a client in an SSH server. System administrators mainly use SSH keys for automated processes and to implement SSO. It is difficult to decode SSH keys through brute forcing.

- **Smart cards**

Smart cards are secure devices or plastic cards with an embedded microchip. It can store sensitive information such as account numbers, passwords, or personal information. In smart-card authentication, the smart card stores the user's public key and a PIN that is used as a secret key to authenticate the user. As the data stored in the smart card cannot be erased, altered, or recovered, it is very difficult to create a duplicate copy, thereby providing enhanced security.

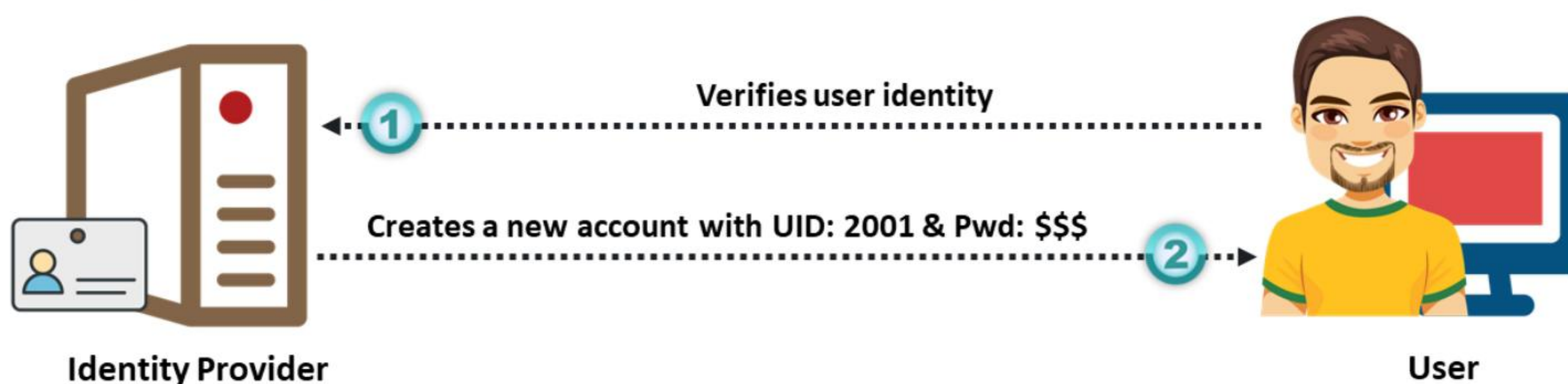
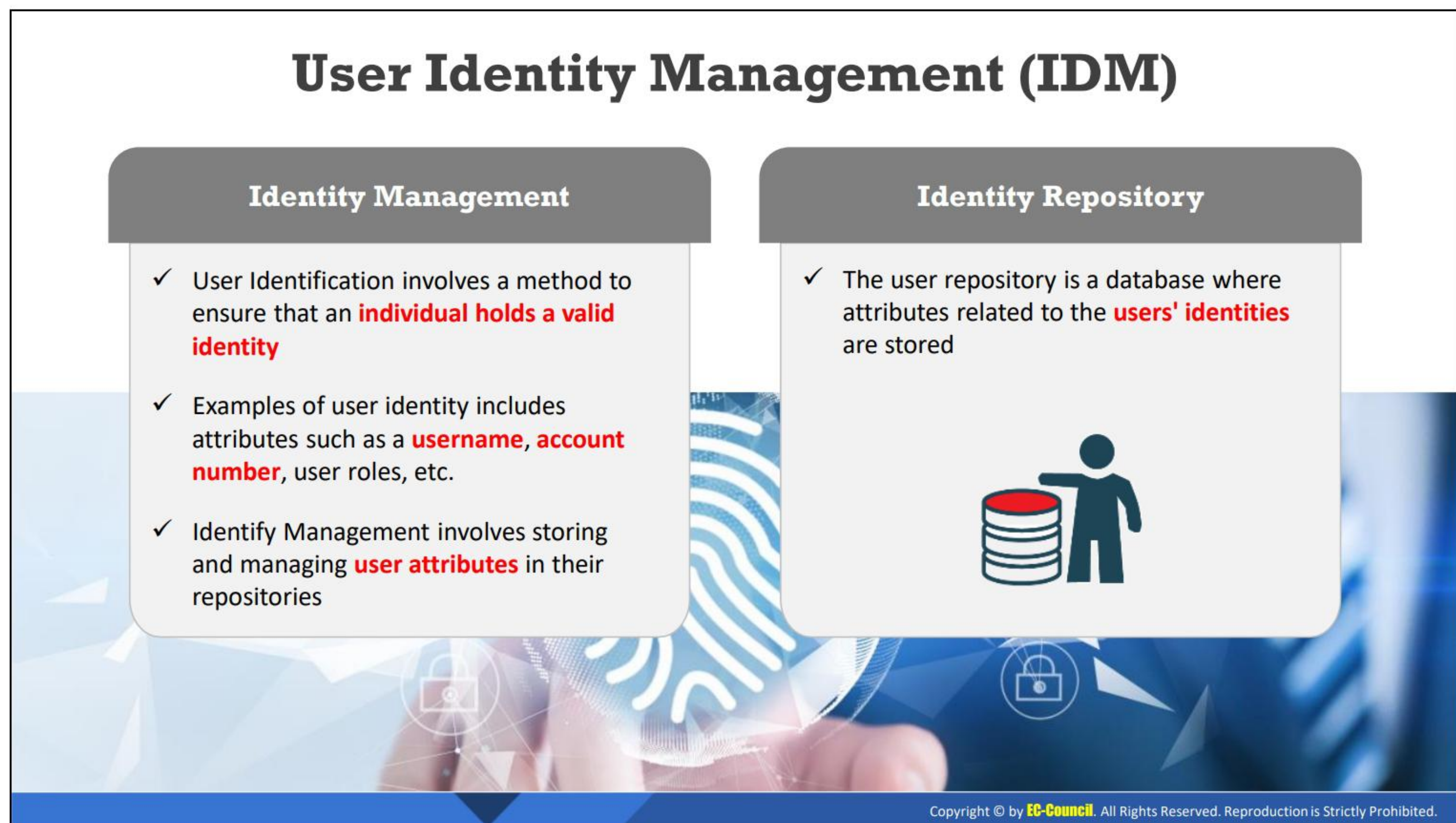


Figure 4.9: User identification




User Identity Management (IDM)

Identification deals with confirming the identity of a user, process, or device accessing the network. User identification is the most commonly used technique for authenticating the users in the network and applications. Users have a unique user ID which helps in their identification. Identify Management involves storing and managing user attributes in their repositories. Here, the user repository is a database where attributes related to the users' identities are stored.

The authentication process includes verifying a user ID and a password. Users are required to provide both the credentials in order to gain access to the network. The network administrators provide access controls such as the username, account number, etc. and permissions to various other services depending on the user IDs.

User Access Management (AM)

- ❑ User access management (AM) is the process of **identifying**, **monitoring**, and **regulating** authorized users' access to an IT system, application, or resource
- ❑ It includes all the **policies**, **processes**, **methodologies**, and **tools** required for the maintenance of AM with IT infrastructure
- ❑ Identity management **creates** and **controls** various users, roles, groups, and policies
- ❑ AM **monitors** and **ensures** that all the defined roles and policies are followed



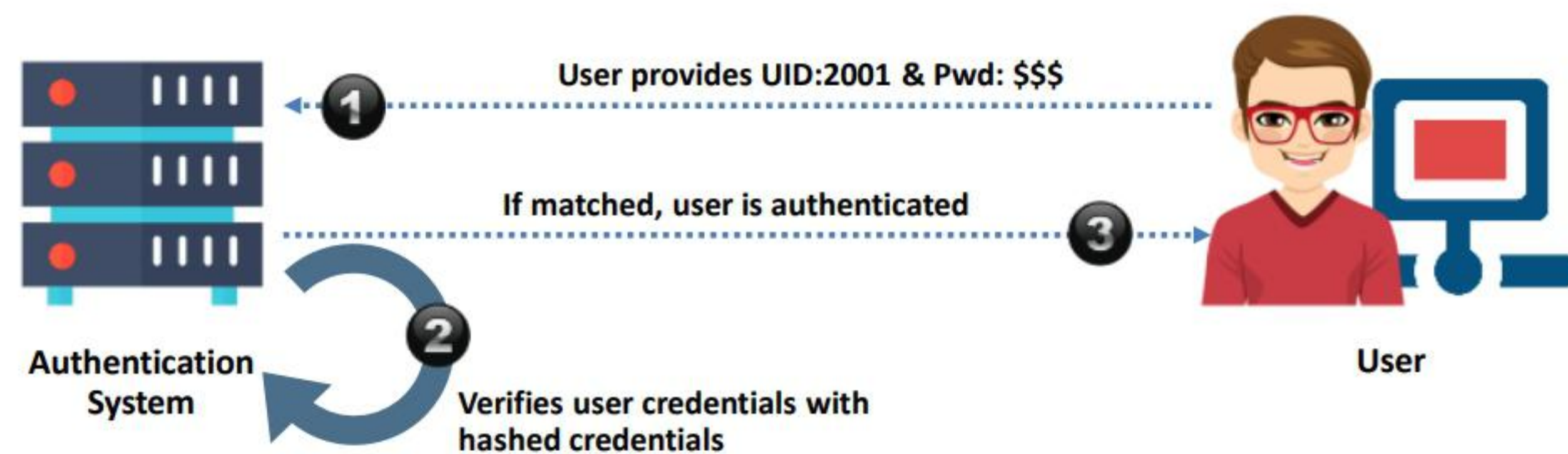
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

User Access Management (AM)

User access management (AM) is the process of identifying, monitoring, and regulating authorized users' access to an IT system, application, or resource. It includes all the policies, processes, methodologies, and tools required for the maintenance of AM with IT infrastructure. User AM is used as part of the IAM functionality. Identity management creates and controls various users, roles, groups, and policies. AM monitors and ensures that all the defined roles and policies are followed.

User Access Management (AM): User Authentication

- ❑ Authentication involves **validating the identity** of an individual with a system, application, or network



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

User Access Management (AM): User Authentication (Cont'd)



Factors

- ✓ **Something you know:** Also known as knowledge factor. Something that user knows like username or password
- ✓ **Something you have:** Also known as owner factor or hard tokens. Something that users retain with them such as a hardware token, employee ID cards, etc.
- ✓ **Something you are:** Also known as biometric factor. Something that is inherent in users, such as biometrics, voice recognition, retina scan, etc.



Attributes

- ✓ **Somewhere you are:** Also known as location-based authentication. Refers to authentication based on user's physical location
- ✓ **Something you do:** Also known as behavioral characteristics. Refers to authentication based on user's actions such as gesture or touch inputs
- ✓ **Something you exhibit:** Also known as behavioral-based authentication and authorization. Refers to authentication exhibiting some device or thing
- ✓ **Someone you know:** Relies on web of trust model. Refers to authentication through peer-level certification and reputation networks

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

User Access Management (AM): User Authentication

Authentication involves verifying the credentials provided by a user while attempting to connect to a network. Both wired and wireless networks authenticate users before allowing them to access the resources in the network. A typical user authentication scheme consists of a user ID and a password. Other forms of authentication include the authentication of a website using a digital certificate and the comparison of the product and label associated with it.

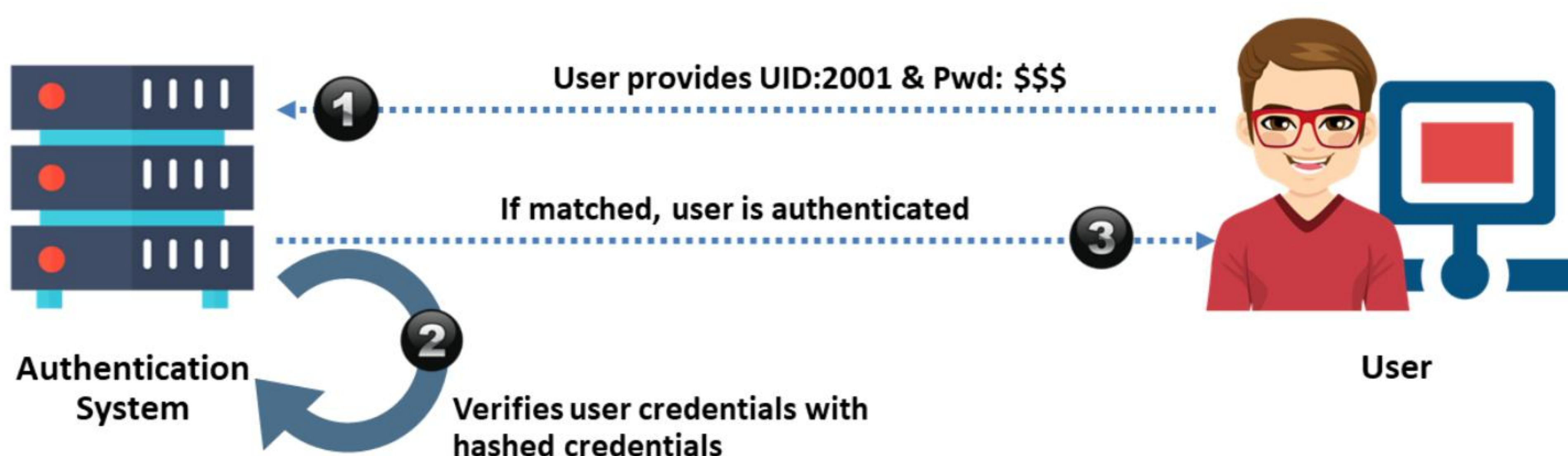


Figure 4.10: User authentication

Factors

Factors refer to the ways in which users can verify their identity during the authentication process. The factors associated with the process of authentication are as follows.

- **Something you know:** Something you know is also known as the knowledge factor. The user should know information such as username and password when attempting to log into a system or network. Other examples of this factor include personal identification numbers (PINs), swipe patterns on touchscreen devices, and challenge questions during the account reset process.
- **Something you have:** Something you have is also known as the owner factor or hard tokens. The user should hold information such as a one-time password token, USB fob, employee ID cards, and account number when attempting to log into a system or network.
- **Something you are:** Something you are is also known as the biometric factor. The user should use their biometric characteristics such as voice, retina scan, and fingerprint scan when attempting to log into a system or network.

Attributes

A user can be identified using a collection of attributes supported in authentication. Some of the attributes are as follows.

- **Somewhere you are:** Somewhere you are is also known as location-based authentication. It refers to the physical location of the user that can be used for authentication. It is also used to block access if the location does not match. This attribute includes geo-location and IP address.
- **Something you do:** Something you do is also known as behavioral characteristics. It refers to the actions that a user must perform to gain access. This attribute includes gestures or touch inputs.
- **Something you exhibit:** Something you exhibit is also known as behavioral-based authentication and authorization. It refers to a device or an object that the user must exhibit in order to authenticate themselves. For example, a legitimate user can be identified based on the normal behavioral pattern of a user on a device; if it deviates, the device will be locked out, and the user must attempt reauthentication.

- **Someone you know:** Someone you know relies on the web-of-trust model. It refers to authentication through peer-level certification and reputation networks, which depends on the reliance of humans on one another. It can be used when primary factors such as passwords and hardware tokens are unavailable.

The commonly used authentication methods are as follows:

- Password Authentication
- Smart Card Authentication
- Biometric Authentication
- Two-factor Authentication
- Single Sign-on (SSO) Authentication

Types of Authentication: Password Authentication

- ☐ Password Authentication uses a **combination** of a username and a password to authenticate the network users
- ☐ The password is checked against a **database** and the user is given access if it matches
- ☐ Password authentication can be vulnerable to **password cracking attacks** such as brute force or dictionary attacks



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Authentication: Password Authentication

In password authentication, users are required to provide usernames and the passwords to prove their identity to a system, application, or a network. These are then matched against a list of authorized users in the database/Windows AD. Once matched, the users can access the system.

The user password should follow standard password creation practices, including a mixture of alphabets, numbers, and special characters and having a length greater than 8 characters (since small passwords are easy to guess).

Password authentication is vulnerable to brute force attacks or dictionary attacks, e.g., a person trying possible combinations of characters to guess the password or capture packets using a “packet sniffer” while sending data across the network as plain text.

Types of Authentication: Two-factor Authentication

- 1** Two-factor authentication involves using two different authentication factors out of three (something you know, something your have, and something you are) to verify the **identity of an individual** in order to enhance the **security in authentication systems**
- 2** **Combinations of two-factor authentication:** password and smart card/token, password and biometrics, password and one-time password (OTP), smart card/token and biometrics, etc.
- 3** “Something you are” is the best companion of two-factor authentication as it is considered as the **hardest to forge** or **spoof**



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Authentication: Two-factor Authentication

Two-factor authentication is a process in which a system confirms user identity in two steps. The user could use a physical entity such as a security token as one of the credentials, and the other credential can include security codes. These security codes can be sent to the end user in the following ways.

- **Email:** The security code is sent via an email message to the registered email account.
- **SMS:** The security code is sent as a short message service (SMS) message to the registered mobile number.
- **Telephone/mobile phone:** The security code is sent via a voice call to the registered telephone or mobile number.
- **Push notification:** An authenticator app on a mobile or PC receives the security code.

Two-factor authentication depends on three factors:

- Something you have
- Something you know
- Something you are

The factor “Something you are” is the best companion of two-factor authentication as it is considered as the hardest to forge or spoof.

Example: A bank card – A user is required to swipe the bank card and enter a PIN while accessing the bank card. Here, the bank card is the physical entity and the PIN is the security code.

The advantage of the two-factor authentication includes decreasing the chances of identity theft and phishing. However, there are certain drawbacks of this two-step process. There are situations where the user will have to wait for the organization to issue the physical token to the user. The delay in receiving the token results in the users waiting for a long time to access their private data.

Identity evaluation depends on knowledge, possession, and inherent factors. Out of these, inherent factors are difficult to change as they depend on the characteristics of a human being.

There are many combinations available in the two-factor authentication process. The most commonly found combinations are:

- Password and smart card
- Password and biometrics
- Password and one-time password (OTP)
- Smart card and biometrics

Two-factor authentications performed without using tokens are called tokenless authentication. They can be implemented quickly across the network.

Two-factor Authentication Techniques: Tokens



Hardware Tokens

- ☐ Physical devices such as a key fob or USB dongle having an **in-built token**; used as an authentication factor for accessing any type of restricted resources
- ☐ Valid only for a short period of **approximately 30 seconds**





Software Tokens

- ☐ A software-based security token includes a **single-use login PIN** or dynamically generated token



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Two-factor Authentication Techniques

▪ Tokens

A token is a highly secure system that is used to transfer authentication information between two entities in a compact and self-contained approach. Tokens can strengthen and secure the authentication processes in applications and websites handling payment information.

The following are the three main elements of tokens.

- **Header:** It defines what type of algorithm and tokens are used.
- **Payload:** It comprises user data and metadata.
- **Signature:** It verifies the sender's identity and checks for the authenticity of the message.

In token-based authentication, on verifying the user's identity once, the user obtains a unique computer-generated encrypted code or token in exchange. This token help the user access protected pages or resources for a specific time without the need to re-enter their credentials each time. Token authentication is always used in a two-factor authentication technique in conjunction with a password or biometric authentication step as the second layer of security.

Tokens are of two types:

○ Hardware tokens

Hardware tokens are physical devices such as a key fob or USB dongle having an in-built token. It is used as an authentication factor for accessing any type of restricted

resources. A hardware token is valid only for a short period of time of approximately 30 seconds, after which the token changes. A simple hardware token has a very low storage capacity and appears like a USB flash drive. Complex hardware tokens consist of LCD displays, and they may also contain keypads for entering passwords. There are two types of hardware tokens: event-based and time-based hardware tokens.



Figure 4.11: Hardware Token

- **Software tokens**

A software token is a software-based security token that includes a single-use login PIN or dynamically generated token. As it is based on software, there is no need of any incremental hardware cost; further, it is updated automatically and can be downloaded or shared and installed in the user's mobile device or system.

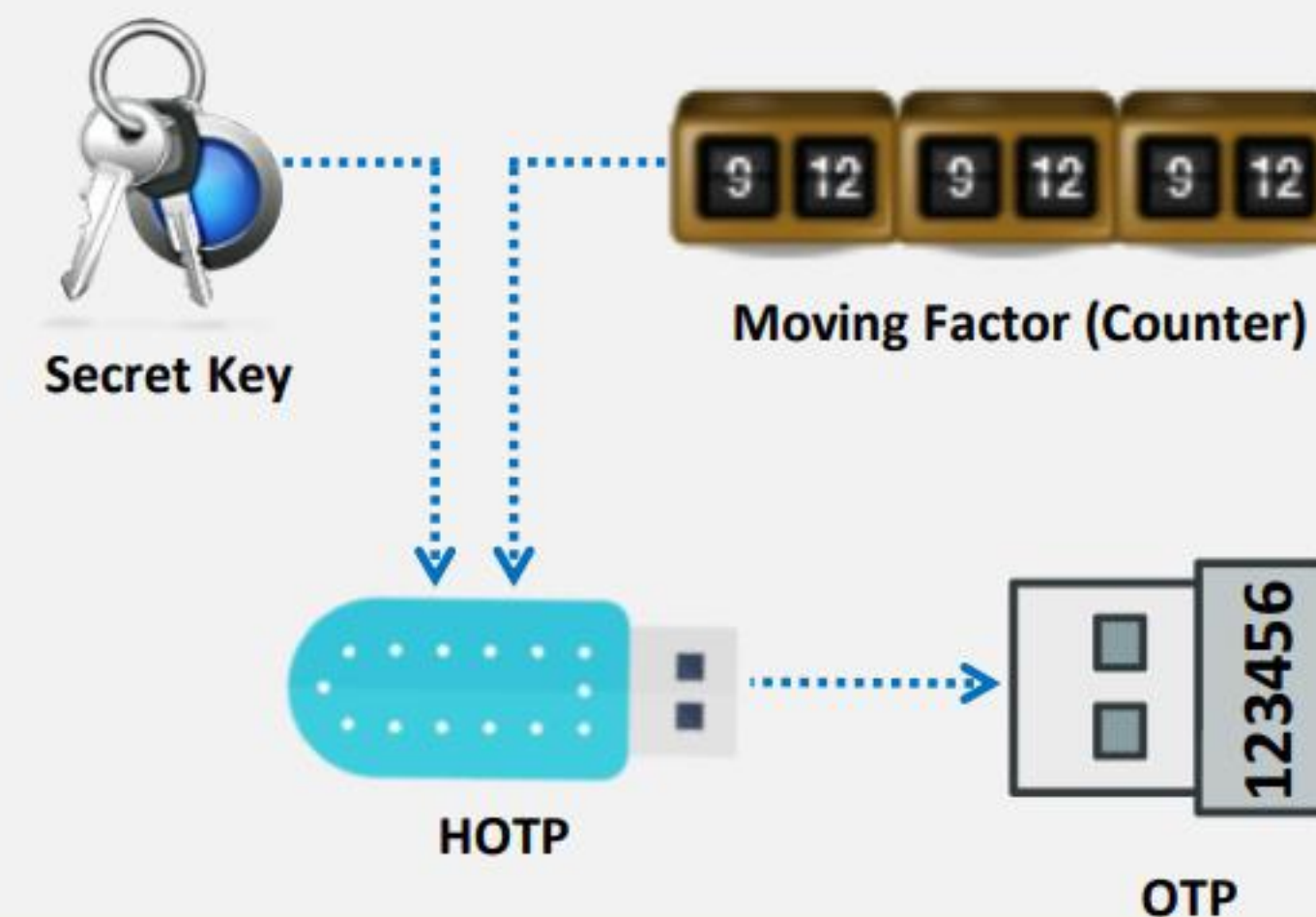
Two-factor Authentication Techniques: OATH



- ❑ Open Authentication (OATH) is a reference architecture designed by a group of companies with the aim of developing an **open strong authentication technology** that supports a wide variety of networks
- ❑ Its main goal is to provide safe and **secure online transactions** for users based on two-factor authentication
- ❑ It has introduced three algorithms, namely, **HOTP**, **OCRA**, and **TOTP**, for implementing OTP authentication

HMAC-based One-time Password (HOTP)

- ✓ An HOTP is an **event-based OTP**, where the input seed is static, and the moving factor is based on a counter
- ✓ When an HOTP is requested, the **moving factor** is incremented based on a counter

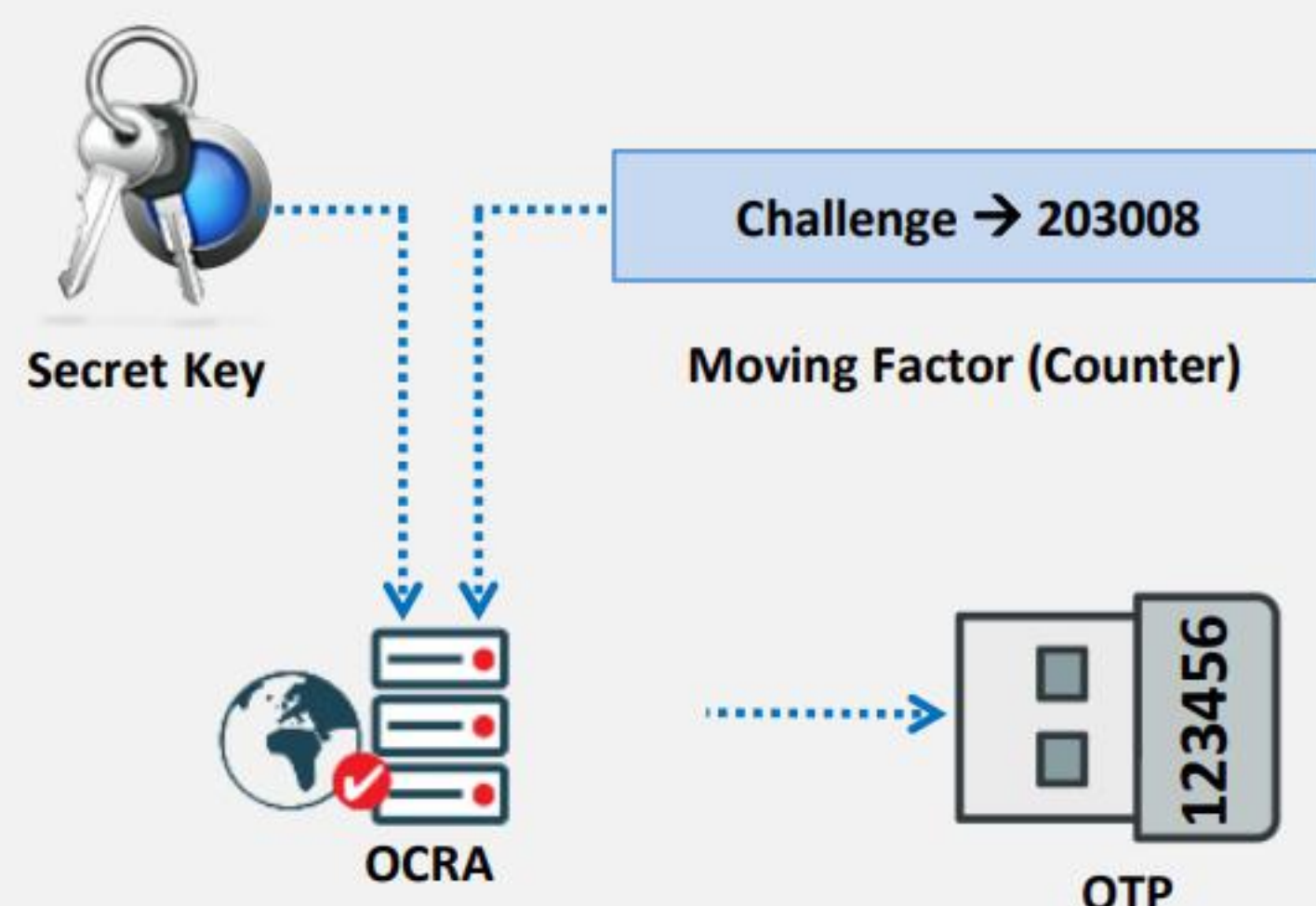


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Two-factor Authentication Techniques: OATH (Cont'd)

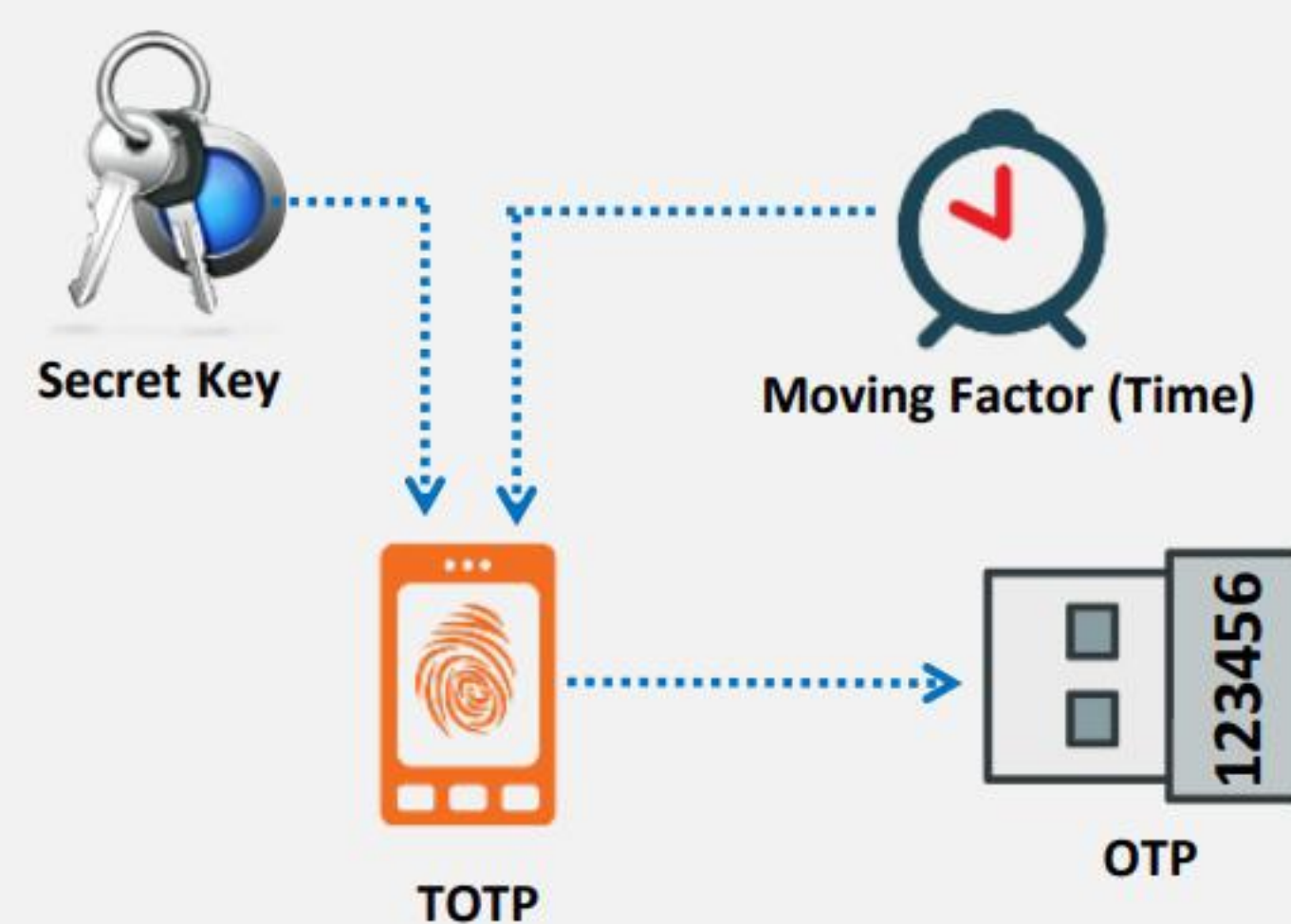
OATH Challenge-Response Algorithm (OCRA)

- ❑ OCRA is challenge-response mode of authentication **based on HOTP**
- ❑ OCRA is an extension to HOTP with a challenge mode for **verifying HOTP tokens** based on random questions



Time-based One-time Password (TOTP)

- ❑ TOTP is a time-based OTP, where the input **seed is static**, and the moving factor is time
- ❑ In TOTP, the time is incremented, and the increment is called the **timestep**



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Open Authentication (OATH)

The Open Authentication (OATH) is a reference architecture designed by a group of companies with the aim of developing an open strong authentication technology that supports a wide variety of networks. Its main goal is to provide safe and secure online transactions for users based on two-factor authentication. For implementing one-time password (OTP) authentication, OATH has introduced three algorithms, namely, hash-

based message authentication code (HMAC)–based OTP (HOTP), OATH challenge-response algorithm (OCRA), and time-based OTP (TOTP).

- **HMAC-based one-time password (HOTP)**

An HOTP is an event-based OTP that depends on two inputs: one is a secret key called seed, and the other is the moving factor. In HOTP, the seed is static, and the moving factor in the OTP code is based on a counter that is stored in the token and on the server. When the HOTP receives a user request, after validation, the moving factor is incremented based on a counter. The authentication server validates the generated OTP, which is valid till the user requests for a new OTP. When the code validation process is performed, the OTP generator and server are synchronized, thereby providing access to the user. HOTP utilizes the SHA-1 hash function in the HMAC, and the token displays the generated 160-bit value, which is then reduced to 6 or 8 decimal digits. One example of an OTP generator that follows the HOTP technique is Yubiko's Yubikey.

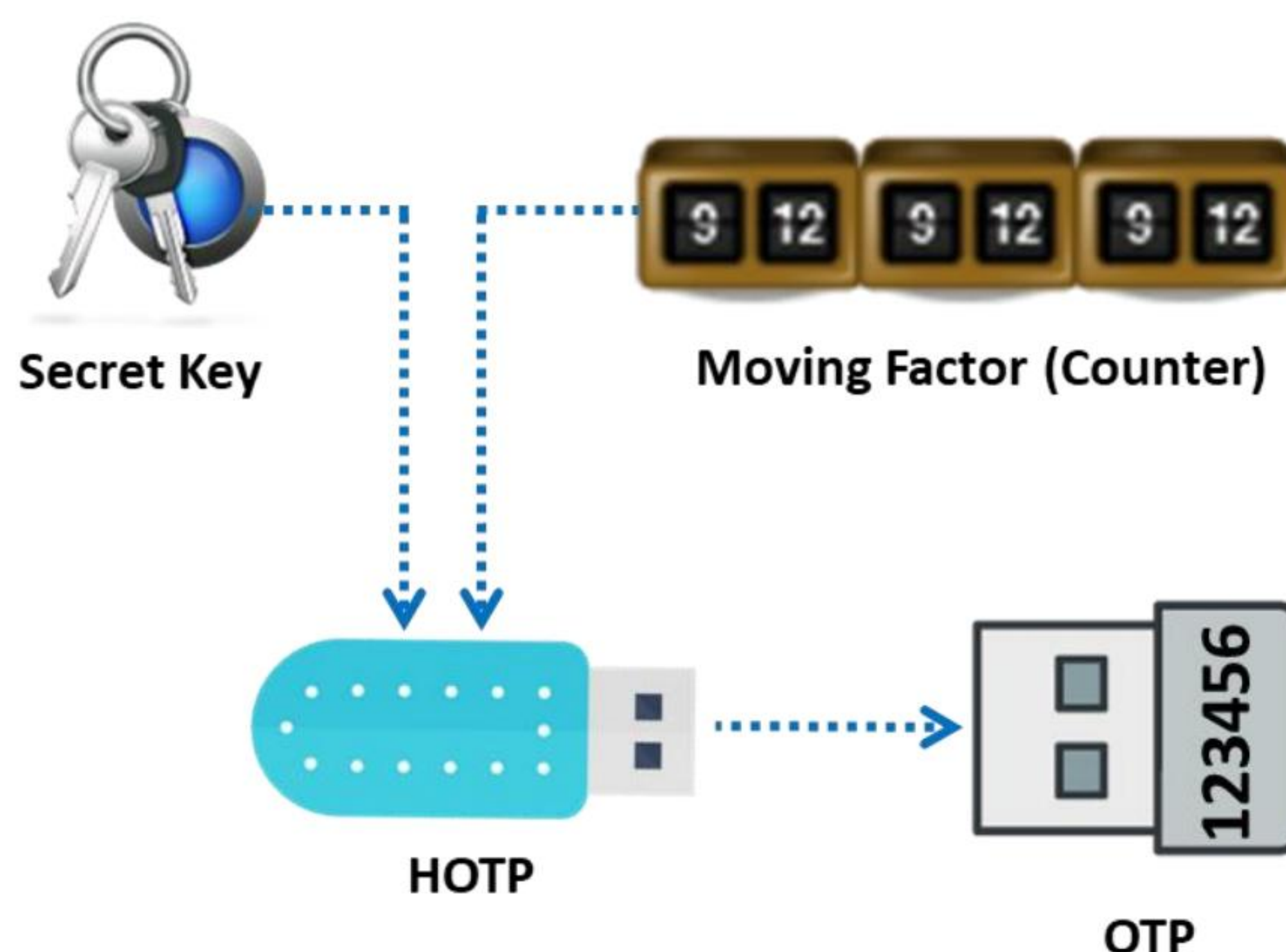


Figure 4.12: HMAC-based one-time password (HOTP)

- **OATH challenge-response algorithm (OCRA)**

The OCRA is a challenge-response mode of authentication based on HOTP. OATH has introduced OCRA as an extension to HOTP with the challenge mode for verifying HOTP tokens based on random questions. The main intention in introducing OCRA is to enhance the security of e-commerce applications.

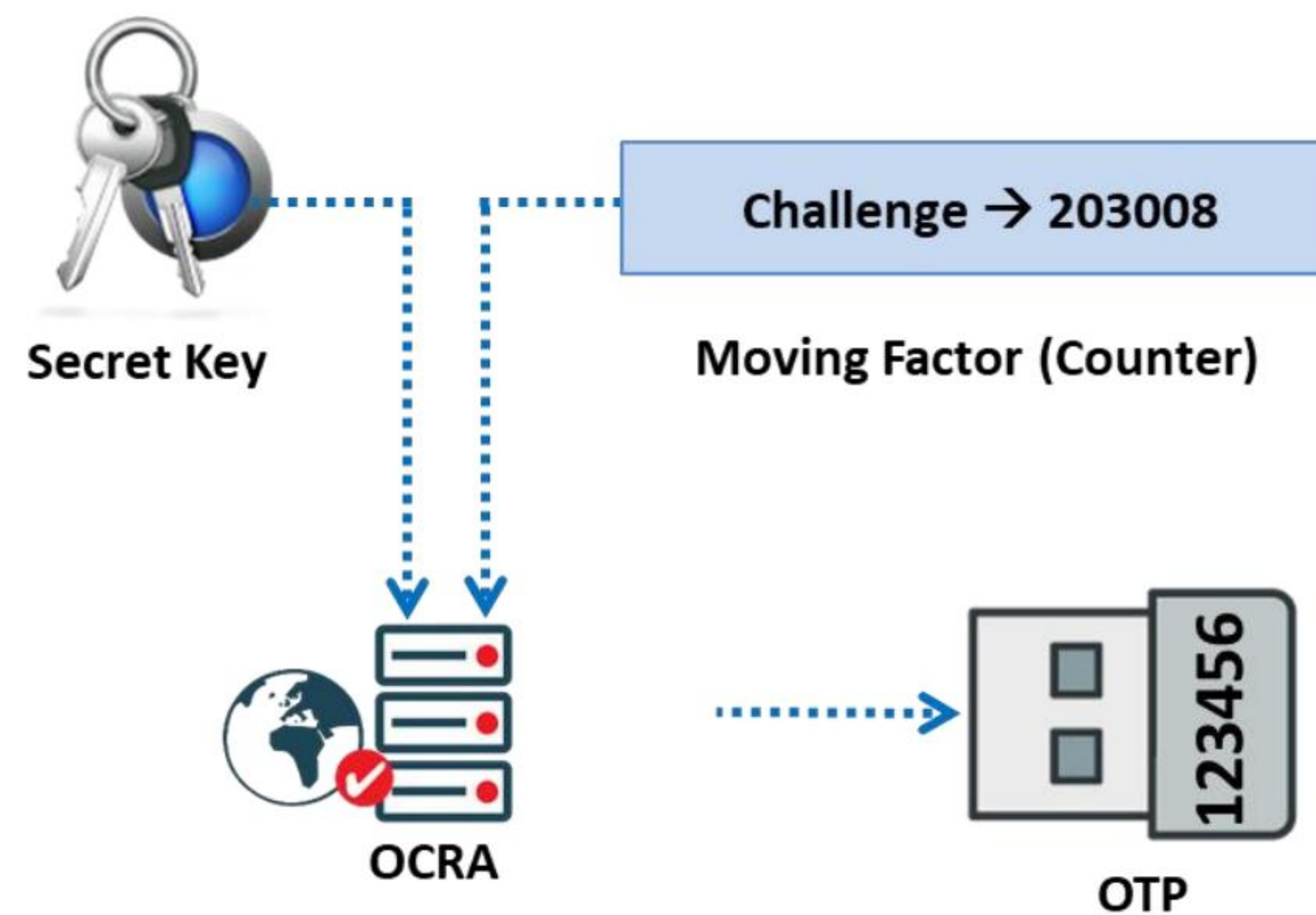


Figure 4.13: OATH challenge-response algorithm (OCRA)

- **Time-based one-time password (TOTP)**

In a TOTP, the input seed is static, as in HOTP, but the input moving factor is time, instead of a counter. In TOTP, the time is incremented by a value called the timestep. The timestep is the amount of time the password is valid for and tends to be 30 or 60 s. Once the timestep is exhausted, the password is no longer valid, and the user must request for a new one to obtain access.

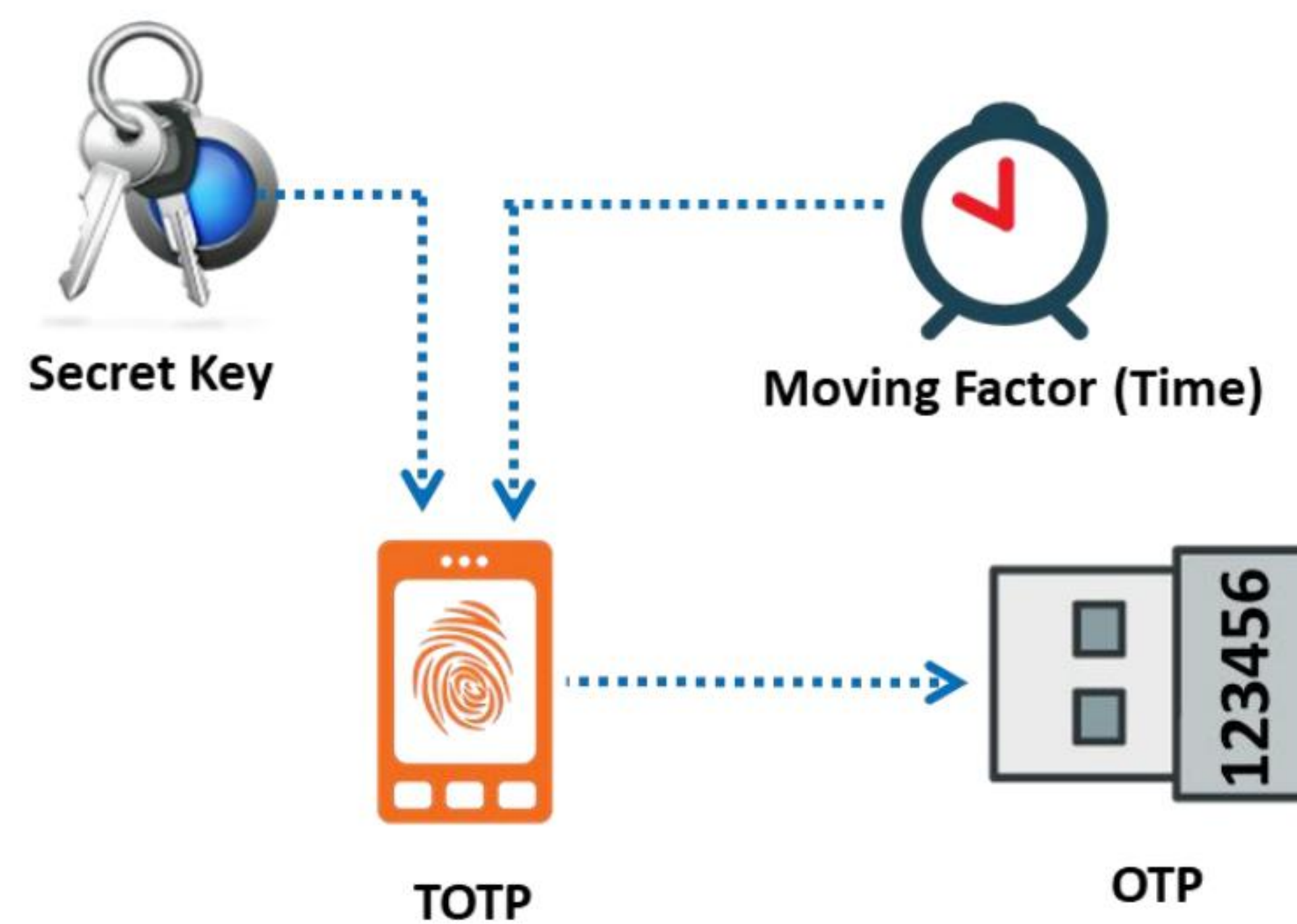


Figure 4.14: Time-based one-time password (TOTP)

Types of Authentication: Biometric Authentication

❑ Biometrics refers to the **identification of individuals** based on their physical characteristics

Biometric Identification Techniques

Fingerprint Scanning

Compares two fingerprints for verification and identification on the basis of the **patterns on the finger**

Iris Scanning

Analyzes the **colored part of the eye** suspended behind the cornea



Retinal Scanning

Analyzes the **layer of blood vessels** at the back of their eyes to identify a person

Vein Structure Recognition

Analyzes thickness and **location of veins** to identify a person

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Authentication: Biometric Authentication (Cont'd)

Biometric Identification Techniques



Face Recognition

Uses **facial features** to identify or verify a person



Voice Recognition

Uses voice patterns to identify or **verify a person**



Gait analysis

Uses **patterns of locomotion** exhibited by moving limbs while walking or running are different among different individuals

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Performance Metrics of Biometric Systems	
Efficacy Rates	Depends on technical components, devices used to capture samples, algorithms used to compare with references, and the environment in which the biometric sensor operates
False Acceptance Rate (FAR)	Percentage of identification occurrences in which an unauthorized user gains access to the resources
False Rejection Rate (FRR)	Percentage of identification occurrences in which an authorized user is denied access to the resources
Crossover Error Rate (CER)	The value of the false rejection rate and false acceptance rate when sensitivity is configured such that FRR and FAR are equal
Receiver Operating Characteristic	Visual characterization plot of the trade-off between FAR and FRR
Matching Speed	Time taken to authenticate an individual
Failure To Capture (FTC)	Ratio of the number of times the system does not capture the samples presented to it to the total number of samples presented
Failure to Enroll (FTE)	Ratio of the number of users that are not enrolled in the system to the total number of users presented to the system
Throughput	Total time taken to enroll the biometric of a user and authenticate the user

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Authentication: Biometric Authentication

Biometrics is a technology which identifies human characteristics for authenticating people. The most commonly used biometrics are fingerprint scanner, retina scanner, facial recognition, DNA, and voice recognition.

Biometric authentication involves the following steps:

- The reader scans the biometric data
- A software converts the scanned information into a digital form and compares it against the biometric data stored in the database
- If both data match, then it confirms the authenticity of the user and allows permission.

The different types of identification techniques used in biometrics are as follows:

- **Fingerprint scanning:** Compares two fingerprints for verification and identification on the basis of the patterns on the finger. The patterns depend on the ridges and minutia points that differentiate each user's fingerprints.
- **Retinal scanning:** Compares and identifies a user on the basis of the distinctive patterns of the retina blood vessels.
- **Iris scanning:** Compares and identifies the images of the iris of one or both eyes of a user. The iris pattern differs from one person to another.
- **Vein structure recognition:** Compares and identifies the patterns produced by a user's veins. Each person has a different pattern depending on the flow of blood.
- **Face recognition:** Compares and identifies a person on the basis of the facial features from an image or a video source.

- **Voice recognition:** Compares and identifies a person on the basis of the voice patterns or speech patterns.
- **Gait analysis:** The patterns of locomotion exhibited by moving limbs while walking or running are different among different individuals. In authentication based on gait analysis, these movements are captured using various methods and compared to the authentic pattern stored in the database. If the pattern matches, the user is authenticated.

Advantages of biometrics:

- It is difficult to tamper biometric details such as a password or a username. They cannot be shared or stolen using social engineering techniques. Biometric authentication requires the presence of the user which reduces the chances unauthorized access.

Disadvantages of biometrics:

- It is difficult to change the biometric factors if this information has been compromised.
- Retinal scan and vein structure scanning can create privacy issues. Both retinal scan and vein structure scan information may inadvertently disclose a medical condition.

Performance Metrics of Biometric Systems

The performance of biometric security systems, applications, or solutions can be assessed using different error rates or performance metrics. Some of the performance metrics are discussed below.

- **Efficacy rates:** The efficacy rates of biometrics depend on technical components such as the devices used to capture samples, algorithms used to compare with the references, application design, and environment in which the biometric sensor operates.
- **False acceptance rates (FAR):** FAR in authentication refers to the percentage of identification occurrences in which an unauthorized user gains access to the resources. A low FAR value indicates better performance of a system.
- **False rejection rate (FRR):** FRR in authentication refers to the percentage of identification occurrences in which an authorized user is denied access to the resources. A lower FRR value indicates better performance of a system.
- **Crossover error rate (CER):** CER is the value of false reject rate and false accept rate when sensitivity is configured such that FRR and FAR are equal. It is also known as the equal error rate and represents the overall accuracy of a biometric system.
- **Receiver operating characteristic:** It is a visual characterization plot of the trade-off between FAR and FRR.
- **Matching speed:** The matching speed is defined as the time taken to authenticate an individual.
- **Genuine accept rate (GAR):** GAR is defined as the ratio of the number of proper authentic samples to the total number of positive samples.

- **Genuine reject rate (GRR):** GRR is defined as the ratio of the number of input samples that are correctly categorized as unauthorized to the total number of unauthorized input samples.
- **Failure to capture (FTC):** FTC is the ratio of the number of times the system does not capture the samples presented to it to the total number of samples presented.
- **Failure to enroll (FTE):** FTE is the ratio of the number of users that are not enrolled in the system to the overall number of users presented to the system.
- **Throughput:** It is the overall time taken to enroll the biometric of a user and authenticate the user.



Types of Authentication: Smart Card Authentication

- Smart card is a small **computer chip device** that holds a users' personal information required to authenticate them
- Users have to insert their smart cards into the card reader machines and enter their **personal identification number** (PIN) to authenticate themselves
- Smart card authentication is a **cryptography-based authentication** and provides stronger security than password authentication

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Types of Authentication: Smart Card Authentication

Organizations use the smart card technology to ensure strong authentication. Smart cards can store password files, authentication tokens, one-time password files, biometric templates, etc. This technology is used with another authentication token, thus providing a multifactor authentication. This enables an efficient logical access security. This technology is applied in VPN authentication, email and data encryption, electronic signatures, secure wireless logon, and biometric authentication.

A smart card consists of a small computer chip that stores personal information of the user for identification. These cards are inserted into a machine for authentication and a personal identification number (PIN) is inputted for processing the authentication information on the card. Smart cards also help in storing public and the private keys.

Smart card authentication is a cryptography-based authentication and provides stronger security than password authentication. The main advantage of using a smart card is that it eliminates the risk of credentials being stolen from a computer as they are stored in the card's chip itself. However, only a limited amount of information can be stored in the card's microchip.

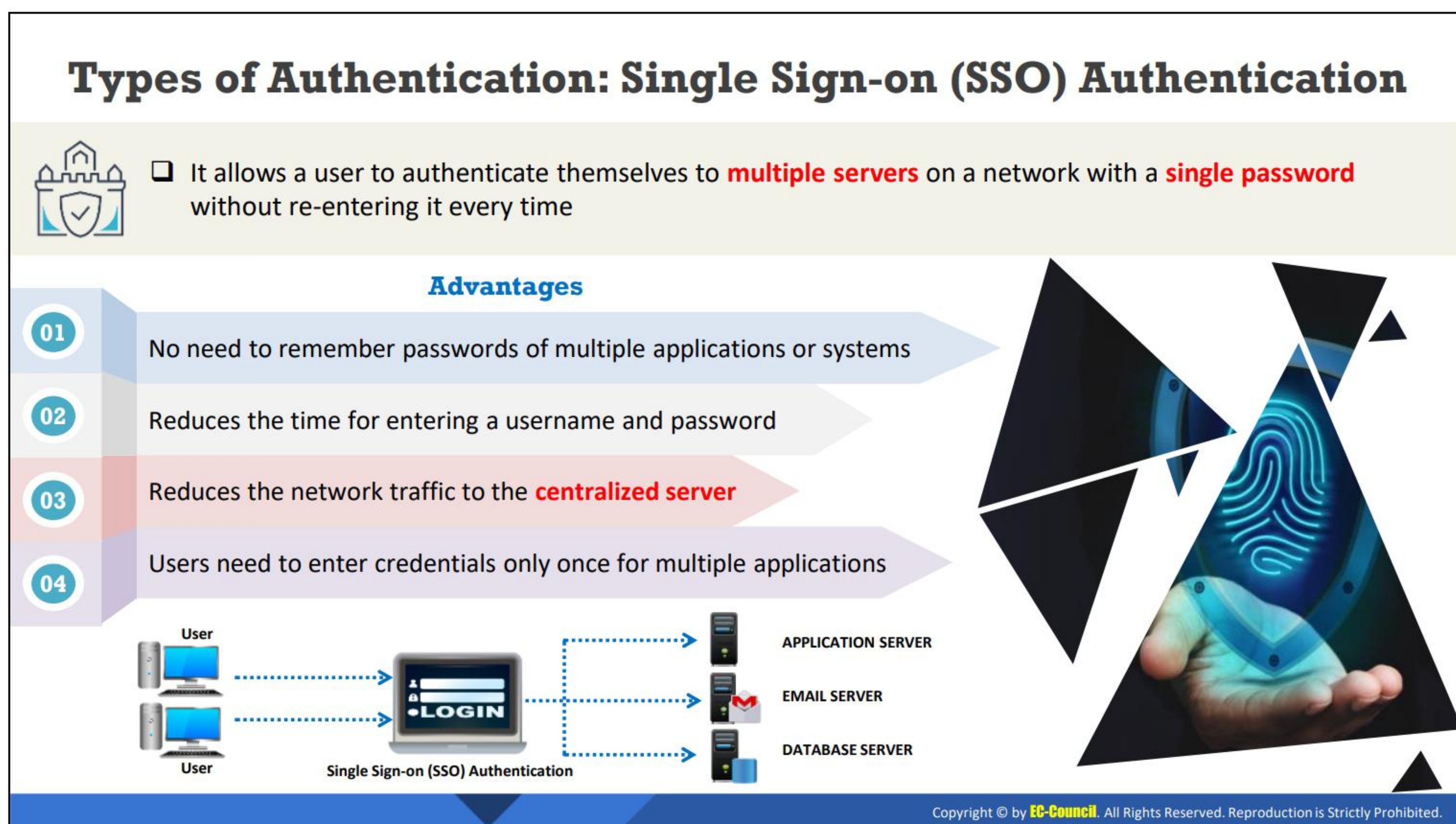
Advantages of smart cards:

- **Highly secure technology:** The smart card technology uses efficient encryption and authentication methods, thus increasing the security of the card.
- **Easy to carry:** Smart cards are easy to carry and a user simply needs to know the PIN of the card.

- **Reduced chances of deception by users:** A smart card enables users to store information such as their fingerprint and other biometric details, thereby allowing organizations to recognize their employees.

Disadvantages of smart cards:

- **Easily lost:** Since smart cards are small in size, the chances of losing them are very high.
- **Security issues:** Losing a smart card puts its owner's information and identity at great risk.
- **High cost of production:** As smart cards have microchips and other encryption technologies; their production cost is high.



Types of Authentication: Single Sign-on (SSO) Authentication

As the name suggests, it allows the users to access multiple applications using a single username and password. The SSO stores the credentials of a user in an SSO policy server. An example of SSO is Google applications. Users can access all Google applications using a single user name and password combination. Consider Google as a central service. This central service creates a cookie for all users logging in for the first time in any of the applications present in the central service. When the user attempts to access other applications of the central service, it eliminates the need for the user to enter the credentials again due to the cookie which has already been created. The system checks the credentials using the created cookie.



Figure 4.15: Single Sign-On (SSO) Authentication

Advantages of SSO:

- Reduces the chances of reauthentication, thereby increasing the productivity.
- Removes the chances of phishing.
- Provides a better management of applications owing to a centralized database.
- Assists with the account lifecycle. Provisioning and deprovisioning of accounts is simplified by the availability of a single source of truth.

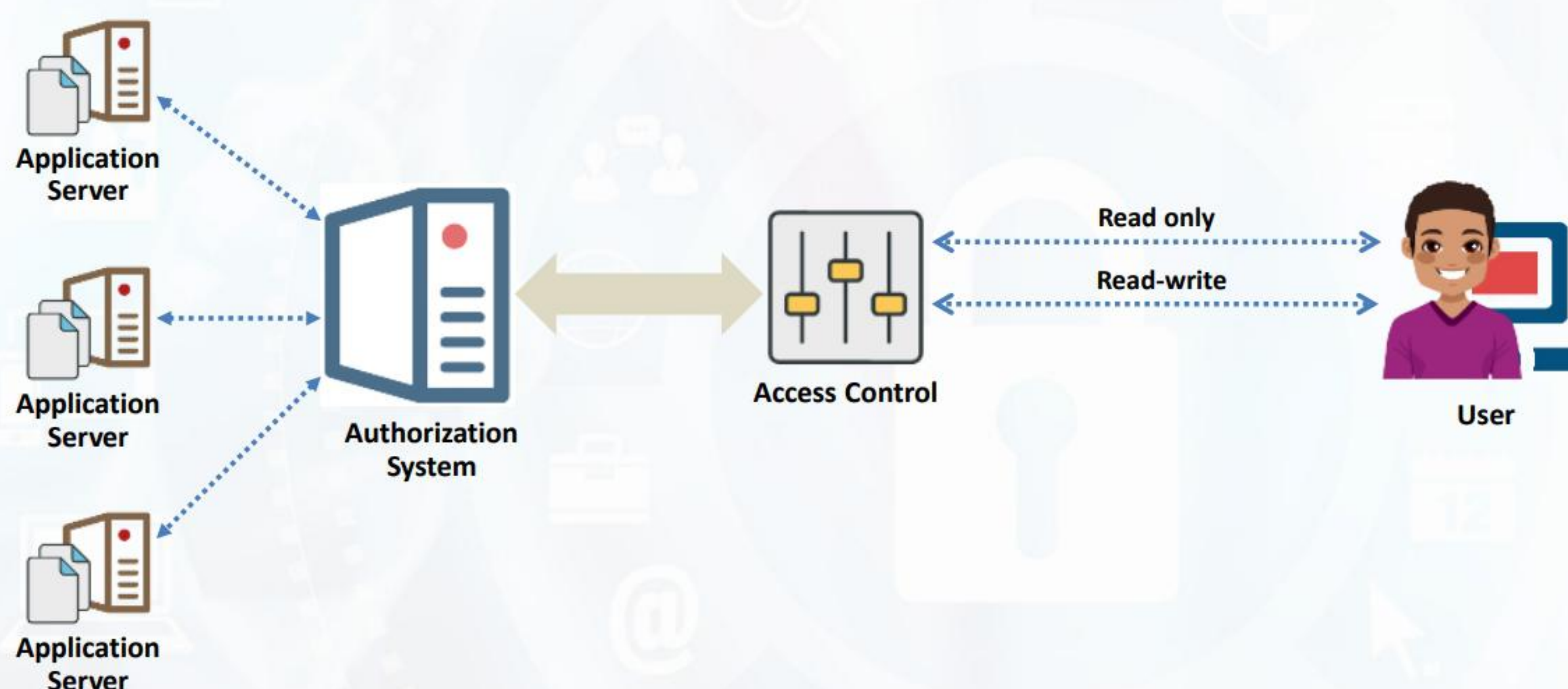
- No need to remember passwords of multiple applications or systems.
- Reduces the time for entering a username and password.

Disadvantages of SSO:

- Losing credentials has a high impact as all the applications of the central service become unavailable.
- There are many vulnerability issues related with the authentication for all the applications.
- It is an issue in multiuser computers and requires the implementation of certain security policies to ensure security.

User Access Management (AM): Authorization

- ❖ Authorization involves **controlling the access** of information for an individual (E.g.: A user can only read a file, but not write in it or delete it)



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

User Access Management (AM): Authorization (Cont'd)

Types of Authorization Systems

Centralized Authorization

- ✓ Authorization for network access is done using a **single centralized** authorization unit
- ✓ It maintains a **single database** for authorizing all the network resources or applications
- ✓ It is an **easy and inexpensive** authorization approach

Decentralized Authorization

- ✓ Each network resource maintains its **authorization unit** and performs authorization locally
- ✓ It maintains its **own database** for authorization



Implicit Authorization

- ✓ Users can access the requested resource **on behalf** of others
- ✓ The access request goes through a **primary resource** to access the requested resource

Explicit Authorization

- ✓ Unlike implicit authorization, explicit authorization requires **separate authorization** for each requested resource
- ✓ It explicitly maintains authorization for each **requested object**

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

User Access Management (AM): Authorization

Authorization refers to the process of providing permission to access the resources or perform an action on the network. s can decide the user privileges and access permissions of users on a multiuser system. The mechanism of authorization can allow the administrator to create access permissions for users as well as verify the access permissions created for each user.

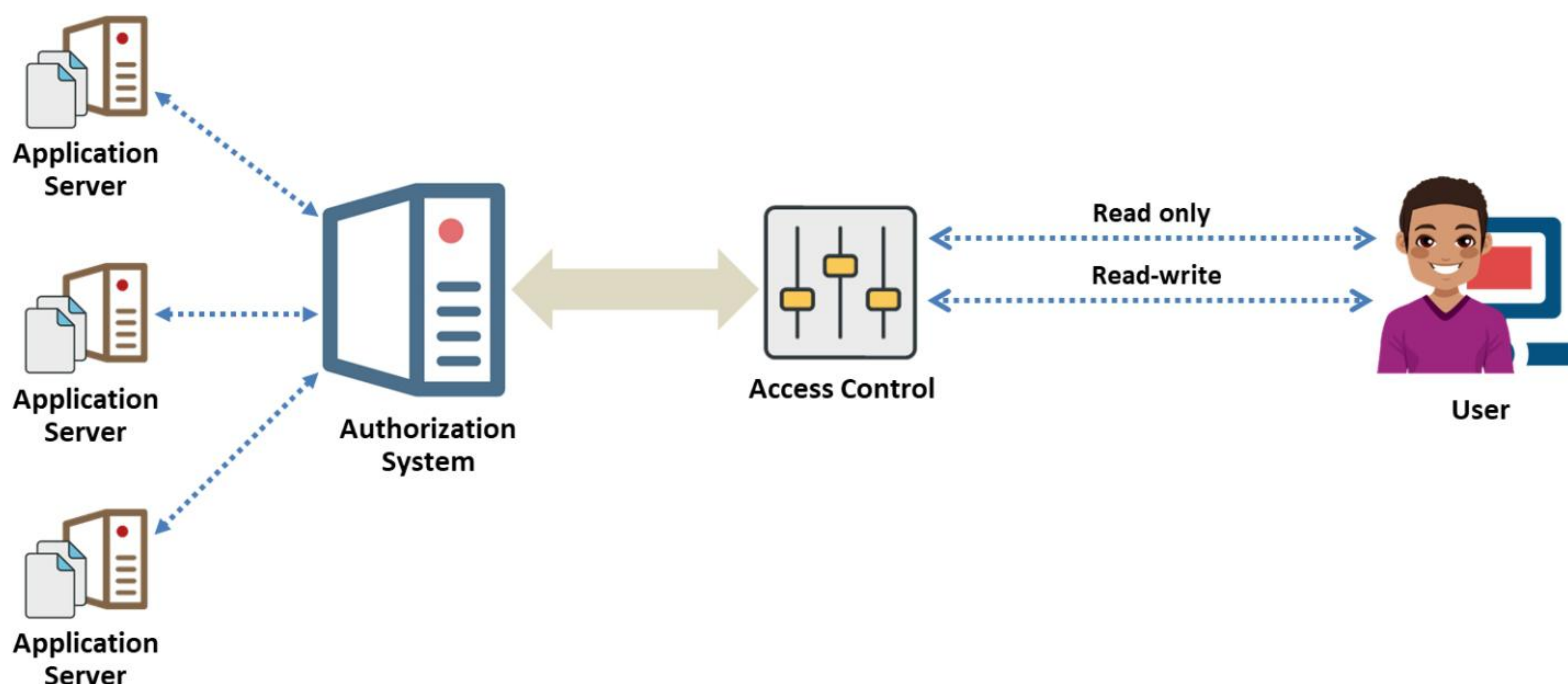


Figure 4.16: Illustration of an authorization system

Authorization can take different forms based on the needs of the organization.

- **Centralized Authorization**

The need for centralized authentication came into existence when it became difficult to implement the authorization process individually for each resource. It uses a central authorization database that allows or denies access to the users and the decision on the access depends on the policies created by the centralized units. This enables an easy authorization for users accessing different platforms. Centralized authorization units are easy to handle and have low costs. A single database provides access to all applications, thereby enabling an efficient security. A centralized database also provides an easy and inexpensive method of adding, modifying, and deleting the applications from the centralized unit.

- **Decentralized Authorization**

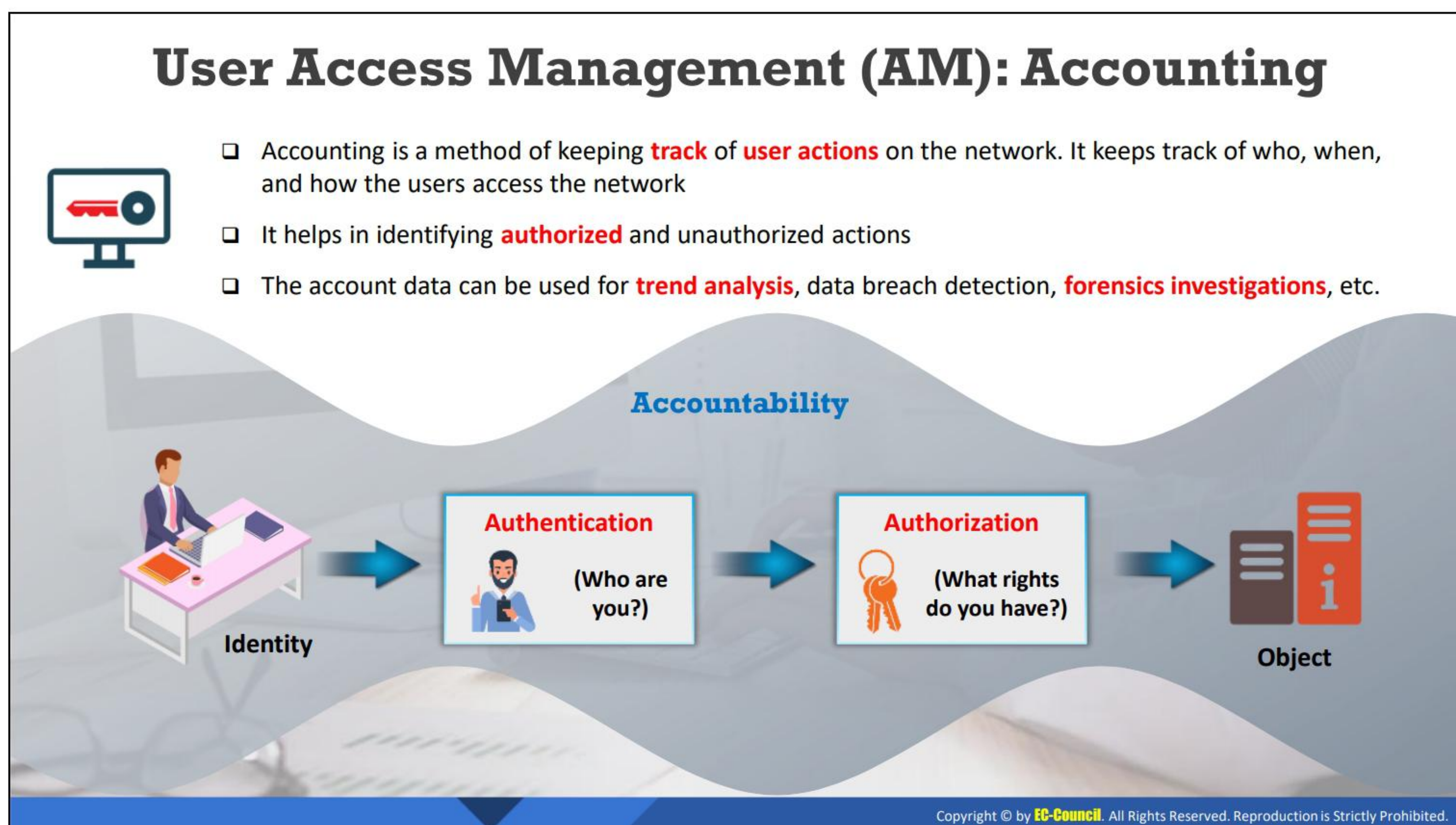
A decentralized authorization maintains a separate database for each resource. The database contains the details of all users who are permitted to access a particular resource. The decentralized authorization process enables users to provide access to other users as well. This increases the level of flexibility of the users in using the decentralized method. However, certain issues related to the decentralized authorization include cascading and cyclic authorizations.

- **Implicit Authorization**

Implicit authorization provides access to the resources indirectly. A task is possible after a user receives authorization for a primary resource through which access to the requested resource is possible. For example, a user requesting a web page has permission to access the main page as well as all pages linked to the main page. Hence, the user is gaining an indirect access to the other links and documents attached to the main page. The implicit authorization provides a level of higher granularity.

- **Explicit Authorization**

An explicit authorization maintains separate authorization details for each resource request. This technique is simpler than the implicit technique. However, it takes up a large amount of storage space for storing all authorization details.



User Access Management (AM): Accounting

User accounting involves tracking the actions performed by a user on a network. It keeps track of who, when, and how the users access the network. This includes verifying the files accessed by the user and functions such as alteration or modification of the files or data. It helps in identifying authorized and unauthorized actions. The account data can be used for trend analysis, data breach detection, forensics investigations, etc.

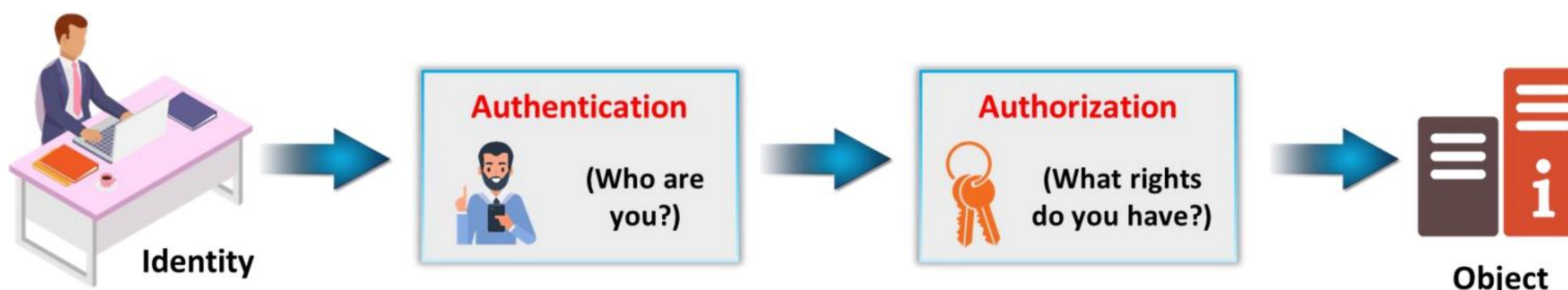


Figure 4.17: User Accounting


Account Types







-  **User Accounts**
 - Default accounts of operating systems
 - Run with the **least privileges**, with permissions such as running applications/programs and creating and manipulating files
-  **Guest Accounts**
 - Least privileged accounts without passwords, created to **share system resources**
 - Do not have any privileges to modify system files, directories, or settings
-  **Service Accounts**
 - Domain or local accounts that allow applications or **services to communicate** with the operating system and run programs
 - Has administrative privileges based on the application requirement
-  **Administrator/Root Accounts**
 - Privileged accounts that can perform various **system-level functions** such as install and uninstall applications or system software and modify system-level settings
-  **Privileged Accounts**
 - Have administrative control over one or several systems
 - Permitted to **access any resources** in the system, configure drivers, add/discard applications from service, etc.

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Account Types (Cont'd)



-  **Shared/Generic Accounts**
 - Credentials are shared **among multiple users**
 - Typically used when the network is divided and needs individual centralized units for network management
-  **Application Accounts**
 - Used by applications to **interact with databases** and **execute batch scripts**
 - Have wide access to the data stored in the organization's database
-  **Group-based Account**
 - Created to simplify the process of **allocating access rights** to individual users
 - A single user can be a participant in several groups and can have permissions from all the participating groups
-  **Third-party Accounts**
 - Used by enterprises to handle **cloud applications** or other **third-party services**
 - Set up with a cryptographic key or **password-based authentication** to use hosts through APIs or SSH

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Account Types

Organizations use different types of privileged accounts for managing systems, applications, and networks. Privileged accounts may be assigned to system or network engineers, network devices, and services. These accounts can be primary targets for attackers because they have elevated access to critical assets. Improper management or misuse of these accounts cause invite significant threats to the entire business infrastructure.

The following are common account types generally found in every organization.

- **User accounts:** User accounts are the default accounts of operating systems (OSes). User accounts permit individuals to log into the system and access resources. Initially, the system can be accessed by a single account that an administrator creates during the OS installation. These accounts run with the least privileges, with permissions such as running applications/programs and creating and manipulating files that belong to their profile.
- **Guest accounts:** Guest accounts are least privileged accounts and have no password; they are created to share system resources. These accounts do not have any privileges to modify system files, directories, or settings. Windows automatically configures guest accounts, but they can be enabled or disabled based on preferences. In Linux-based systems, an administrator is required to manually create a guest account after installing the OS. Most web services have default guest accounts that allow users to access web servers without providing credentials.
- **Service accounts:** Service accounts, referred to as domain or local accounts, allow applications or services to communicate with the OS and run programs or services. Service accounts may also have administrative privileges based on the application requirement or purpose they are intended to serve. Windows has three types of services: system, local, and network services. System services run with higher privileges compared to other accounts. These services use a local system account to start the OS and will have complete privileges on the running system. Local and network services run with the same privileges as a standard user and are allowed to access only network resources. Linux also creates service accounts while installing web servers and applications.
- **Administrator/root accounts:** These accounts are privileged accounts that can perform various system-level functions such as installing and uninstalling applications or system software; modifying system-level settings; and reading, modifying, or deleting any file on the system. It is recommended to create a small number of such accounts with elevated privileges to perform administrative activities and access the components of the file system. In general, it is difficult to remove default administrator accounts, which are created by the application or OS during its installation. The default account can have all the permissions enabled. These accounts are also known as superuser accounts. They are called administrator accounts in Windows environments and root accounts in Linux environments.
- **Privileged accounts:** Privileged accounts are granted administrative control over one or several systems. These accounts are permitted to access any resources in the system, configure or run drivers, add/discard applications from services, and make configuration changes. Typically, few accounts will have this type of elevated privileges to manage the system, network, or applications.
- **Shared/generic accounts:** In shared accounts, the login credentials are shared among multiple users. This approach is typically used when the network is divided and needs

individual centralized units for network management. Shared accounts can violate the non-repudiation mechanism; further, they can make the task of maintaining accurate audit trails challenging. If an organization's password policy requires frequent password changes, then a password change needs to be intimated to every user having access to a shared account, which is a challenging task and may lead to many security risks. Shared accounts are not considered a best security practice because there is high probability of their credentials being compromised.

- **Application accounts:** Application accounts are used by applications to interact with databases, execute batch scripts, and allow access to other applications. These accounts have wide access to the data or information stored in the organization's database. If the credentials for these accounts are integrated and saved in unencrypted files, may pose a severe threat to the organization.
- **Group-based accounts:** Group-based accounts are created to simplify the process of allocating access rights to individual users. Instead of providing rights directly, the owner of the system allocates them to individual group accounts. The rights are then reflected for all the group members. A single user can be a member of several groups; they can acquire permissions and access rights from all those groups.
- **Third-party accounts:** Third-party credentials are used by enterprises to handle cloud applications or other services provided by third-party vendors. Along with administrative sign-ins, third-party services or devices should be set up with a cryptographic key or password-based authentication to use hosts through APIs or SSH. Inefficient handling of these keys or passwords, such as their insertion in code in an unencrypted form, can cause several security breaches.



Module Summary

- ❑ This module discussed the terminology, principles, and types of access control
- ❑ It covered identity and access management (IAM)
- ❑ It also discussed user access management
- ❑ Furthermore, this module discussed the different types of authentication and authorization
- ❑ Finally, this module presented an overview of user accounting
- ❑ The next module discusses administrative network security controls in detail

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Module Summary

This module discussed the terminology, principles, and types of access control. It covered identity and access management (IAM). It also discussed user access management. Furthermore, this module discussed the different types of authentication and authorization. Finally, this module presented an overview of user accounting.

The next module discusses administrative network security controls in detail.

This page is intentionally left blank.