Unit 2: Server Credentials Management
User Identity and Authentication Systems

User identity and authentication systems are foundational elements of secure network and server management. They ensure that only authorized individuals gain access to resources by verifying their identity through credentials.

User Identity

a) User identity uniquely identifies a person or entity within a computer system or network.
It acts like a digital passport that distinguishes one user from another in an information system.
A user identity helps systems determine who is requesting access to resources, applications, or services.
It is an essential component of authentication, authorization, and accountability in cybersecurity environments.
Without proper user identity management, systems become vulnerable to unauthorized access and security breaches.
b) Typically represented by user accounts, which include usernames and associated credential information.
These credentials may include passwords, PINs, biometric data, or security tokens used for verification.
User accounts also store permissions and access rights that define what a user is allowed to do within the system.
Organizations often manage user accounts through centralized identity management systems to improve security and efficiency.
Proper maintenance of user accounts helps prevent unauthorized access, data leakage, and misuse of system resources.

Authentication Systems

a) a) Mechanisms to verify the claim of identity by the user.
These mechanisms ensure that a person attempting to access a system is genuinely who they claim to be. Common verification methods include passwords, biometric scans, smart cards, PINs, and multi-factor authentication techniques. Authentication mechanisms help protect systems, data, and networks from unauthorized access and cyber threats.
b) Common methods include:
   - *Passwords:* Most basic form of authentication.
   - *Multi-factor Authentication:* Combines something the user knows (password) with something the user has (token) or something the user is (biometrics).
   - *Token-based Authentication:* Using security tokens or certificates.
   - *Kerberos:* A network authentication protocol used primarily in Windows domains to provide secure identity verification using tickets.

Types of Computer Credentials

Credentials are the digital keys that users, services, or devices present to prove their identity. Different types serve different purposes:

1. **Local User Credentials**
   Local user credentials are used for authenticating accounts created and stored on a specific computer or device. These accounts are managed locally rather than through a central network server. An example is a user account created on a standalone Windows PC, such as *localuser*.
   Local credentials are useful when a system operates independently without connection to a network domain.

2. **Domain Credentials**
   Domain credentials are used for authenticating users within a centrally managed network environment. These accounts are controlled by a domain server, allowing users to access multiple network resources using one account. An example is an Active Directory account such as *jdoe@example.com*. Domain credentials improve centralized management, security, and access control across an organization's network.
   - Allows single sign-on (SSO) across multiple domain-joined computers.

3. **Service/Application Credentials**
   Service or application credentials are used by software services and applications to access systems or resources automatically without direct human involvement. These credentials may include service accounts, API keys, tokens, or client certificates with specific permissions. For example, a database service account may be used to run background operations securely. Proper management of these credentials is important to prevent unauthorized access and reduce security risks.

4. **Network Authentication Credentials**
   Network authentication credentials are used to verify users or devices when accessing network services and resources. They help ensure secure communication and controlled access across networks. Examples include Windows network logon credentials using Kerberos tickets and VPN credentials used to create secure remote connections. These credentials play a major role in protecting network systems from unauthorized access.

Active Directory Configuration

Active Directory is a directory service developed by Microsoft for Windows domain networks. It provides centralized management of users, computers, applications, and network resources within an organization. Active Directory allows administrators to control authentication, authorization, security policies, and access permissions from a single system. It plays a critical role in improving network security, resource management, and administrative efficiency.

Domain Creation

Domains represent a logical group of network objects, such as users, computers, printers, and other resources, that share a common directory database. Within a domain, administrators can centrally manage security policies, user accounts, and resource permissions. Domains help organizations organize network resources efficiently while providing secure authentication and access control across the network.

Steps to create a domain:

Install the Active Directory Domain Services (AD DS) role on a Windows Server.

1. Run the Active Directory Domain Services Configuration Wizard to promote the server to a domain controller.
2. Define the domain name (e.g., example.local).

Domain Controller Setup

A domain controller (DC) is a server that responds to authentication requests and verifies users within a network domain. It manages security policies, user logins, and access permissions for resources connected to the domain. Domain controllers store directory information and ensure that only authorized users and devices can access network services and data.

Main functions of Domain Controller:

i. Authenticate and authorize users.
ii. Store the domain directory database.
iii. Manage Group Policy distribution.

Primary domain controller (PDC) and backup domain controllers (BDCs) can exist to ensure reliability.

User and Group Creation

a) **Users** represent individual identities within a domain. Each user account contains unique login credentials and permissions that allow a person to access network resources and services securely. User accounts help administrators manage authentication, security policies, and access rights within the domain.

b) **Groups** are collections of users created to simplify the management of permissions and access rights. Instead of assigning permissions to each user individually, administrators can assign permissions to a group, making administration more efficient and organized.

Examples of groups include:

a. **Security groups** are used to assign permissions and control access to resources such as files, folders, printers, and applications.

b. **Distribution groups** are mainly used for email distribution lists, allowing messages to be sent to multiple users at once.

Creating users and groups can be performed through tools such as:

c. Active Directory Users and Computers (ADUC) MMC snap-in, which provides a graphical interface for managing domain objects.

d. PowerShell cmdlets such as *New-ADUser* and *New-ADGroup*, which allow administrators to automate user and group creation through command-line scripting.

Example PowerShell Commands:

```
# Create a new Active Directory user
New-ADUser -Name "John Doe" `
           -GivenName "John" `
           -Surname "Doe" `
           -SamAccountName "jdoe" `
           -UserPrincipalName "jdoe@example.com" `
           -Path "OU=Users,DC=example,DC=com" `
           -AccountPassword (ConvertTo-SecureString "P@ssw0rd123" -AsPlainText -
Force) `
           -Enabled $true
```

The command above creates a new domain user account named John Doe and enables the account with a password.

```
# Create a new security group
New-ADGroup -Name "ITAdmins" `
            -GroupScope Global `
            -GroupCategory Security `
            -Path "OU=Groups,DC=example,DC=com"
```

This command creates a global security group named *ITAdmins* inside the specified organizational unit.

```
# Add a user to a group
Add-ADGroupMember -Identity "ITAdmins" -Members "jdoe"
```

This command adds the user *jdoe* to the *ITAdmins* security group.

Organizational Units (OUs) Structure (Listing Format with Roman Numerals)

a) Organizational Units (OUs) in Active Directory are used to organize users, groups, computers, and other resources into logical sections within a domain. They help structure and simplify management of a network by grouping objects based on departments or roles.

Example OU Structure:
i. Domain: *example.local*
ii. Sales OU
iii. Users: John, Mary
iv. Computers: Sales-PC1, Sales-PC2
v. IT OU
vi. Users: Admin1, Tech2
vii. Computers: IT-PC1, Server01

b) OUs also allow delegation of administration and application of Group Policy Objects (GPOs) to specific parts of the organization.

Examples:
i. Delegate management of the Sales OU to a sales manager who can reset passwords only for Sales users.
ii. Apply a strict password policy GPO to the IT OU requiring stronger authentication rules.
iii. Restrict software installation on Sales computers while allowing full administrative access in the IT OU.

Group Policy Objects (GPOs)
GPOs in Active Directory define configurations and security policies applied to users and computers within a domain.They allow centralized and automated management of system settings across multiple devices.
GPOs can control password policies, software installation, network drive mapping, and security options.
GPOs can be linked to:
i. Sites
ii. Domains
iii. Organizational Units (OUs)
Example: A password complexity policy enforced through a GPO linked to the Users OU ensures all users follow the same security rules.

Active Directory Testing and Validation
• Testing ensures that authentication and security in Active Directory work correctly and securely.

Login Authentication Testing
i. Verify users can log into domain-joined computers using domain credentials.
ii. Confirm that domain controllers properly respond to login requests.

iii. Example: A test account is used to log into different machines to confirm Single Sign-On (SSO) works correctly.

Permission Verification
i. Confirm users have correct access to files, folders, and applications.
ii. Ensure security policies and permissions are correctly applied.
iii. Example tools include AccessChk (Sysinternals) and the Windows Security tab to check effective permissions.

Domain Access Validation

Network Connectivity and Domain Communication Testing Validate network connectivity and ensure client machines can communicate properly with domain controllers in Active Directory. This confirms that authentication and directory services are reachable across the network.

Test domain join status and verify domain trust relationships, especially in environments with multiple domains. This ensures that computers are correctly joined to the domain and trust relationships are functioning for secure authentication across domains.

Use the following commands for testing and troubleshooting:

i. `nltest /dsgetdc:domainname` → Used to locate and verify a domain controller for a specific domain.
Example:
```
nltest /dsgetdc:example.local
```
→ Displays the domain controller assigned to the *example.local* domain.

ii. `ping` → Used to test basic network connectivity between a client and a server.
Example:
```
ping dc01.example.local
```
→ Confirms whether the domain controller is reachable over the network.

iii. `nslookup` → Used to check DNS resolution and confirm that domain names map to correct IP addresses.
Example:
```
nslookup example.local
```
→ Returns the IP address of the domain or domain controller, confirming DNS is working properly.

Troubleshooting Authentication Errors

Common issues:

i. Incorrect passwords or expired credentials.
ii. Time synchronization problems causing Kerberos ticket failures.
iii. Account lockouts due to repeated failed logins.
iv. Replication issues between domain controllers.

Troubleshooting steps:

i Check Event Viewer logs on client and domain controllers

a) Open Event Viewer
b) Win + R then type

```
eventvwr.msc
```

Important logs to check
a)Directory Service on domain controllers
b)DNS Server logs on domain controllers
c)System logs
d)Security logs

PowerShell examples

```
Get-EventLog -LogName System -Newest 50
Get-EventLog -LogName "Directory Service" -Newest 50
```

Filter errors only

```
Get-WinEvent -LogName System | Where-Object {$_.LevelDisplayName -eq "Error"} |
Select-Object -First 20
```

ii Use dcdiag and repadmin to assess domain controller health

Run full diagnostic

```
dcdiag /v
```

Test a specific domain controller

```
dcdiag /s:DC01 /v
```

DNS focused test

```
dcdiag /test:dns /v
```

Check replication summary

```
repadmin /replsummary
```

Show detailed replication status

```
repadmin /showrepl
```

Force replication

```
repadmin /syncall /AeD
```

iii Ensure network connectivity and DNS configuration are correct

Test connectivity

```
ping domaincontroller.domain.local
```

Check DNS resolution

```
nslookup domain.local
nslookup DC01.domain.local
```

Check full network configuration

```
ipconfig /all
```

Flush and re-register DNS

```
ipconfig /flushdns
ipconfig /registerdns
```

Check domain controller discovery

```
nltest /dsgetdc:domain.local
```

iv Verify Active Directory replication status

Show replication status for all DCs

```
repadmin /showrepl *
```

Find replication failures

```
repadmin /replsummary | findstr /i "fail"
```

Recalculate replication topology

```
repadmin /kcc
```

Check secure channel trust

```
nltest /sc_verify:domain.local
```

Diagram Description: Active Directory Structure

