Active Directory: Section 1 Transcript

Introduction

Welcome to the Active Directory module.

During this module, we're going to discuss what Active Directory is, how it works, and how it's managed. Throughout this module, you'll be presented with opportunities to assess and apply what you've learned.

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

- Identify the components of the Active Directory Logical Structure,
- Identify the components of the Active Directory Physical Structure, and
- Demonstrate the use of command line tools to interact with Active Directory.

Bypass Exam Introduction

If you are already familiar with the subject matter presented in this module, you can choose to take a Bypass Exam to skip this module.

The Bypass Exam option provides a single opportunity to successfully demonstrate your competence with the material presented within the module. If you pass, you'll receive credit for completing the module, unlocking the content within, and you will be free to proceed to the next module. If you do not pass, you will need to successfully complete the module, including all exercises and the Module Exam, to receive credit.

Click the Next Section button to continue.

Active Directory: Section 2 Transcript

What is Active Directory? 1/19

So, what is Active Directory?

Active Directory, or AD, is essentially a hierarchical database designed to the X.500 standard for global directories; it is a centrally controlled database that provides a single point of access for systems administration and network management. As a directory service, AD organizes, tracks, and stores "things" in its database - those "things" are referred to as objects.

AD can be implemented in small environments or distributed across the world.

Active Directory Objects 2/19

AD does more than store usernames and passwords - it stores network resources like printers and share folder information. Other services like email can use AD as well. The bottom line is that everything in AD is considered an object regardless of whether it's a user, group policy, computer, or printer.

AD is a database that holds objects which are the most basic component of AD. An object is a distinctly named set of attributes that represents a network resource. Every object in Active Directory has a Global Unique Identifier (GUID) and a unique Lightweight Directory Access Protocol (LDAP) Distinguished Name (DN). A GUID is a 128-bit number. The GUID is guaranteed to be unique within a Forest and will not change even if the DN changes.

Next, we'll discuss the Active Directory Schema.

Active Directory Schema 3/19

The Active Directory schema defines objects that can be stored in AD. The schema is a list of definitions that determine the kinds of objects and types of information about those objects that can be stored in AD. Since the schema definitions themselves are stored as objects, they can be administered in the same manner as the rest of the objects in AD.

The schema is defined by two types of objects:

- Schema Class objects (referred to as schema class), and
- Schema Attribute objects (referred to as schema attribute).

Schema Class Objects:

- Define the possible AD objects that can be created, and
- Function as a template for creating new AD objects,
- Each schema class is a collection of schema attribute objects, and
- When you create a schema class, the schema attributes store the information that describes

the object.

Schema Attribute Objects:

- Define the schema class objects with which they are associated. Each schema attribute is defined only once and can be used in multiple schema classes.

Active Directory Logical Structure 4/19

In order for users and computers to be put into a logical management structure, domains are created. The domain is the core unit of logical structure in AD and can be used to store millions of objects vital to the network. A domain is a logical group of computers that share the same AD database and namespace. All computers in a domain share the same namespace.

For example, if your domain namespace is abccorp.com, all computers within that domain share that namespace, such as PC1.abccorp.com.

Active Directory Logical Structure 5/19

Network resources like users, computers, and printers can be added to a domain which is managed within the AD database. If the network has multiple domains, it has separate AD databases; however, resources can be shared across domains.

When discussing domains and AD, it's important to mention Domain Controllers.

Active Directory Physical Structure 6/19

Domain Controllers, or DC, run Active Directory Domain Services. As soon as the Active Directory Domain Services role is added to the server, it becomes a DC. The DC holds a copy of the AD database and replicates changes to other DCs.

The primary job of the DC is to authenticate users and determine accesses after authentication occurs. Additionally, DCs perform collision detection, provide fault tolerance, and manage all aspects of the users' domain interactions.

Since most AD networks contain multiple domain controllers and users could theoretically attach to any domain controller for authentication or information, each of the servers needs to be kept up to date.

DCs stay up to date by replicating the database between each other by using a pull method. After a change, the DC initiates a replication after waiting 15 seconds, in Windows 2003 or 5 minutes, in Windows 2000. Windows Server 2003 uses technology to only replicate changed information and compresses replication over WAN links.

AD uses Multimaster replication. Multimaster replication does not rely on a single primary domain controller, but instead treats each DC as an authority. When a change is made on any DC, it is replicated to all other DCs. Although each DC is replicated alike, all of the DCs aren't equal. There are several flexible single-master operation roles that are assigned to one domain controller at a

time.

AD uses Remote Procedure Calls (RPC) for replication and can use SMTP for changes to schema or configuration.

Microsoft recommends using as few domains as possible and suggests using Organizational Units (OUs) for structure since domains can contain multiple nested OUs.

Active Directory Logical Structure

An OU is a container which provides a domain hierarchy and structure. It is used for ease of administration and to create an AD structure in the company's geographic or organizational terms or needs. OUs can contain OUs, allowing for the creation of multi-level structures.

Active Directory Logical Structure

There are three primary reasons for creating OUs:

- Organizational Structure,
- Security Rights, and
- Delegated Administration.

Click each OU for more information.

- **Organizational Structure**: First, creating OUs allows a company to build a structure in AD which matches their firm's geographic or organizational structure. This permits ease of administration and a clean structure.
- **Security Rights**: The second reason to create an OU structure is to assign security rights to certain OUs. This, for example, would allow you to apply AD policies to one OU which are different than another. You could setup policies which install an accounting software application on computers in the Accounting OU.
- **Delegated Administration**: The third reason to create OUs is to delegate administrative responsibility. AD architects can design the structure to allow local administrators certain administrative responsibility for their OU and no other. This allows for a delegated administration not available in Windows NT networks.

Active Directory Logical Structure

The most basic design of an AD is a single forest, single domain, and no OU design. For a small organization, this might be adequate, but almost every organization can benefit from some structure.

Typically, companies design their OU trees based on geographical separation or organizational design. There is no incorrect way to design an AD environment; however, consistency is key and should be considered prior to implementation.

Active Directory Logical Structure

It's important to make the distinction between Trees and Forests. A tree is a grouping or hierarchical

arrangement of one or more domains that easily accommodates organizational changes. Trees are created by adding one or more child domains to a parent domain. In a tree, all domains share the same namespace and naming structure. By adding domains to a tree, you can retain the security configuration through the tree (domain), and allow for administration to be delegated to a single OU or a single domain.

At the top of the AD structure is a forest. A forest holds all objects, OUs, domains, and attributes in its hierarchy. A forest is a grouping or hierarchical arrangement of one or more separate and completely independent domain trees. Under a forest are one or more trees which holds domains, OUs, objects, and attributes.

Trust Relationships are important in an AD environment so forests and domains can communicate with one another and pass credentials. Within a single forest, trusts are created when a domain is created. By default, domains have an implicit two-way transitive trust. This means each domain trusts each other for security access and credentials. A user in domain A can access resources permitted in domain B while a different user in domain B can access resources permitted in domain A.

Next, we'll focus our attention to the Active Directory Physical Structure.

Active Directory Physical Structure

Sites in AD represent the physical structure, or topology, of your network. AD uses topology information, stored as site and site link objects in the directory, to build the most efficient replication topology. Active Directory Sites and Services are used to define sites and site links. A site is a set of well-connected subnets. Sites differ from domains; sites represent the physical structure of your network, while domains represent the logical structure of your organization.

Typically, sites are used for:

- Physical Location Determination, and
- Replication.

Physical Location Determination enables clients to find local resources such as printers, shares, or domain controllers. Replication optimizes replication between domain controllers by creating links. AD, by default, uses automatic site coverage; however, you can purposefully setup sites and resources.

Active Directory Physical Structure

In AD, sites map the physical structure of your network, while domains map the logical or administrative structure of your organization.

This separation of physical and logical structure provides the following benefits:

- You can design and maintain the logical and physical structures of your network independently,
- You do not have to base domain namespaces on your physical network,
- You can deploy domain controllers for multiple domains within the same site, and

- You can also deploy domain controllers for the same domain in multiple sites.

As a network gets larger, it can contain multiple domains and many domain controllers. Each domain only contains records from its own domain in its AD database to keep the database small and replication manageable. That said, the AD domain relies on a global catalog database which contains a global listing of all objects in the forest.

Active Directory Physical Structure

The Global Catalog is held on domain controllers configured as global catalog servers. The global catalog contains a subset of information, such as a user's first name and last name, and the distinguished name of the object so your client can contact the proper domain controller if you need more information.

The distinguished name is the full address of an object in the directory; every object in Active Directory has a distinguished name.

For example, a printer in the Organizational Unit Accounting in the abccorp.com domain might have this distinguished name (CN=AcctLaser1,OU=Accounting,DC=abccorp,DC=com).

Active Directory Physical Structure

All domain controllers are not created equal, as some DCs have more responsibility than others. In a forest, there are at least five Flexible Single-Master Operations (FSMO) roles that are assigned to one or more domain controllers - two are forest-wide and three are domain-wide roles.

The five FSMO roles are:

- Schema Master,
- Domain Naming Master,
- Infrastructure Master,
- Relative ID (RID) Master, and
- PDC Emulator.

The forest-wide roles are:

- Schema Master, and
- Domain Naming Master.

The three domain-wide roles are:

- Infrastructure Master,
- RID Master, and
- PDC Emulator.

Click each DC to view more information about each of the roles.

Schema Master: The schema master domain controller controls all updates and modifications to the schema. To update the schema of a forest, you must have access to the schema master. There can

be only one schema master in the whole forest.

Domain naming master: The domain naming master domain controller controls the addition or removal of domains in the forest. There can be only one domain naming master in the whole forest.

Infrastructure Master: The infrastructure is responsible for updating references from objects in its domain to objects in other domains. At any one time, there can be only one domain controller acting as the infrastructure master in each domain.

Relative ID (RID) Master: The RID master is responsible for processing RID pool requests from all domain controllers in a particular domain. At any one time, there can be only one DC acting as the RID master in the domain.

PDC Emulator: The PDC emulator is a DC that advertises itself as the primary domain controller (PDC) to workstations, member servers, and DCs that are running earlier versions of Windows. For example, if the domain contains computers that are not running Microsoft Windows XP Professional or Microsoft Windows 2000 client software, or if it contains Microsoft Windows NT backup DCs, the PDC emulator master acts as a Windows NT PDC. It is also the Domain Master Browser, and it handles password discrepancies. At any one time, there can be only one DC acting as the PDC emulator master in each domain in the forest.

LDAP

AD is an LDAP compliant directory service, which means that all access to directory objects occurs through LDAP. LDAP requires that names of directory objects be formed according to RFC 1779 and RFC 2247, which define the standard for object names in an LDAP directory service.

Objects are located within AD domains according to a hierarchical path, which includes the labels of the AD domain name and each level of container objects.

The full path to the object is defined by the distinguished name, also known as a DN. The name of the object itself, separate from the path to the object, is defined by the relative distinguished name.

The distinguished name is unambiguous and unique. By using the full path to an object, including the object name and all parent objects to the root of the domain, the distinguished name uniquely and unambiguously identifies an object within a domain hierarchy.

It contains sufficient information for an LDAP client to retrieve the object's information from the directory.

User Accounts

In AD, each user account has a user logon name, a pre-Windows 2000 user logon name, which is the security account manager account name, and a User Principle Name (UPN) suffix. The administrator enters the user logon name and selects the UPN suffix when creating the user account. Active Directory suggests a pre-Windows 2000 user logon name using the first 20 bytes of the user logon name. Administrators can change the pre-Windows 2000 logon name at any time.

SAM account names are required for compatibility with Windows NT 4.0 and Windows NT 3.x

domains, referred to as flat names. Every name must be unique within the domain.

A UPN is a "friendly" name that is shorter than the distinguished name, consists of a user name, the "at" sign (@) and usually the DNS name of the domain where the user resides, and is often the user's email address. The default UPN is <username>@<DNSdomainname>.

UPNs can be mapped to make the user name easier to remember or even to clarify OU membership.

A UPN could be:

- jdoe@abccorp.com, or
- jdoe@us.abccorp.com or other variations.

The UPN is independent of the distinguished name of the user object so a user object can be moved or renamed without affecting the user logon name.

Tools

Windows Server includes several tools that let you manage AD from the command line.

Click each server tool for more information.

LDIFDE.exe (LDAP Data Interchange Format (LDIF) Data Exchange tool)

Can be used to:

- Import, export, modify, and delete directory objects
- Export AD user and group information to other applications or services
- Populate AD with data from other directory services
- Populating passwords, which CSVDE doesn't
- This includes information that is stored in the schema

CSVDE.exe (Comma Separated Value Data Export tool)

Can be used to:

- Export and import (add) object data only
- Export directly to or import directly from an MS Excel spreadsheet
- Execute batch operations that are based on CSV

CSVDE cannot be used to modify or delete objects.

ENUMPROP.exe (Enumerate Properties tool)

Can be used to:

- Provide information about the domain, domain controller, and site
- Display the security descriptor, or list only a given set of attributes for an object
- Recognize globally unique identifiers (GUIDs) (objectGUID and schemaGUID) and security identifiers (SIDs)

This is a good command to run when you first land on a new target since it requires no authentication - you just need to be on the AD domain.

Tools

Domain Service (DS) tools are a set of command line tools that began shipping natively with Windows Server 2003.

Click each one of the tools for a brief description or for a downloadable version of the Domain Service Tools reference, see the Resources for this module.

DSADD: This tool's commands add specific types of objects to the directory.

- DSADD COMPUTER: Adds a computer to the directory
- DSADD CONTACT: Adds a contact to the directory
- DSADD GROUP: Adds a group to the directory
- DSADD OU: Adds an organizational unit to the directory
- DSADD USER: Adds a user to the directory
- DSADD QUOTA: Adds a quota specification to a directory partition

DSGET: This tool's commands display the selected properties of a specific object in the directory.

- DSGET COMPUTER: Displays properties of computers in the directory
- DSGET CONTACT: Displays properties of contacts in the directory
- DSGET SUBNET: Displays properties of subnets in the directory
- DSGET GROUP: Displays properties of groups in the directory
- DSGET OU: Displays properties of organizational units in the directory
- DSGET SERVER: Displays properties of servers in the directory
- DSGET SITE: Displays properties of sites in the directory
- DSGET USER: Displays properties of users in the directory
- DSGET QUOTA: Displays properties of quotas in the directory
- DSGET PARTITION: Displays properties of partitions in the directory

DSMOD: This tool's commands modify existing objects in the directory.

- DSMOD COMPUTER: Modifies an existing computer in the directory
- DSMOD CONTACT: Modifies an existing contact in the directory
- DSMOD GROUP: Modifies an existing group in the directory
- DSMOD OU: Modifies an existing organizational unit in the directory
- DSMOD SERVER: Modifies an existing domain controller in the directory
- DSMOD USER: Modifies an existing user in the directory
- DSMOD QUOTA: Modifies an existing quota specification in the directory
- DSMOD PARTITION: Modifies attributes of one or more existing partitions in the directory

DSQUERY: This tool's commands allow you to query the directory according to specific criteria.

- DSQUERY COMPUTER: Finds computers in the directory
- DSQUERY CONTACT: Finds contacts in the directory
- DSQUERY SUBNET: Finds subnets in the directory

- DSQUERY GROUP: Finds groups in the directory
- DSQUERY OU: Finds organizational units in the directory
- DSQUERY SITE: Finds sites in the directory
- DSQUERY SERVER: Finds domain controllers in the directory
- DSQUERY USER: Finds users in the directory
- DSQUERY QUOTA: Finds quota specifications in the directory
- DSQUERY PARTITION: Finds partitions in the directory
- DSQUERY *: Finds any object in the directory by using a generic LDAP query

DSMOVE: This tool's commands move or rename objects within the directory

DSRM: This tool's commands remove/delete objects from the directory.

Knowledge Check Introduction

It is time for a Knowledge Check. This Knowledge Check will not be scored, but may indicate areas that you need to review prior to the Module Exam.

Active Directory: Section 3 Transcript

Kerberos

Kerberos is the default authentication mechanism implemented within AD, providing strong mutual authentication for the client and the server. Encrypted messages are exchanged between the Kerberos Key Distribution Center (KDC), the client, and the requested resource.

Kerberos uses symmetric key cryptography to protect user passwords while in transit and relies on timestamps during the verification process in order to diminish the possibility of an attacker using old keys.

Kerberos Authentication Process

Let's briefly review how Kerberos works.

When a client requests access to a resource, it must first send a message, an Authentication Service Request (AS_REQ) to the KDC requesting authentication. The request is encrypted with the NTLM hash of the user's password. Within the KDC, the Authentication Service (AS) receives the request and verifies that the user is valid by checking AD to confirm that the user is a member of the domain, and then uses the hash stored in its database to decrypt the request. Assuming the user is verified, the AS responds by sending a Ticket Granting Ticket (TGT).

Kerberos Authentication Process

Next, the client generates a Ticket Granting Service Request (TGS_REQ) and sends it to the Ticket Granting Service (TGS) within the KDC.

The TGS checks its database to determine if the resource exists and assuming that it does, sends the Ticket Granting Service Response (TGS_REP) to the client.

The client then decrypts the response via the TGS session key contained in the TGT.

The client generates and sends an Authenticator message and the encrypted Service Ticket (ST) to the resource.

The client can now access and use the resource.

Active Directory: Section 4 Transcript

Summary

You have completed the Active Directory module.

During this module, we discussed components and features of ADs, the LDAP syntax, and various command line tools that are used to interact with objects located in the AD database. Then we focused on applying what was learned by providing practical exercises to demonstrate the use of command line tools to interact with AD objects.

You should now be able to:

- Identify the components of Active Directory Logical Structure,
- Identify the components of Active Directory Physical Structure, and
- Demonstrate the use of command line tools to interact with AD.

To receive credit and advance to the next module, you must achieve a passing score on the Module Exam.

Click the Next Section button to begin the Module Exam.