Lesson 15: Configuring Service Authentication

MOAC 70-411: Administering Windows Server 2012

Microsoft Official Academic Course

WILEY

Overview

- Exam Objective 5.1: Configure Service Authentication
- Configuring Service Authentication
- Managing Service Accounts

Configuring Service Authentication

Lesson 15: Configuring Service Authentication

Authentication

- ***Authentication*** is the act of confirming the identity of a user or system and is an essential part used in authorization when the user or system tries to access a server or network resource.
- Two types of authentication that Windows supports are NT LAN Manager (NTLM) and Kerberos.
- Kerberos is the default authentication protocol for domain computers.
- NTLM is the default authentication protocol for Windows NT, standalone computers that are not part of a domain, and situations in which you authenticate to a server using an IP address.

Understanding NTLM Authentication

- ***NT LAN Manager (NTLM)*** is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users.
- NTLM is an integrated single sign-on mechanism.
- NTLM uses a challenge-response mechanism for authentication in which clients are able to prove their identities without sending a password to the server.

Managing Kerberos

Kerberos:

- Is a computer network authentication protocol, which allows hosts to prove their identity over a non-secure network in a secure manner.
- Can provide mutual authentication so that both the user and server verify each other's identity.
- Protocol messages are protected against eavesdropping and replay attacks.
- Supports ticketing authentication.

Managing Kerberos

- When a user logs in to a network resource using Kerberos, the client transmits the following to the authentication server:
  - Username
  - Identity of the service the user wants to connect to (for example, a file server or a SharePoint server)
- The authentication server constructs a ticket, which contains a randomly generated session key, which is encrypted with the file server's secret key.

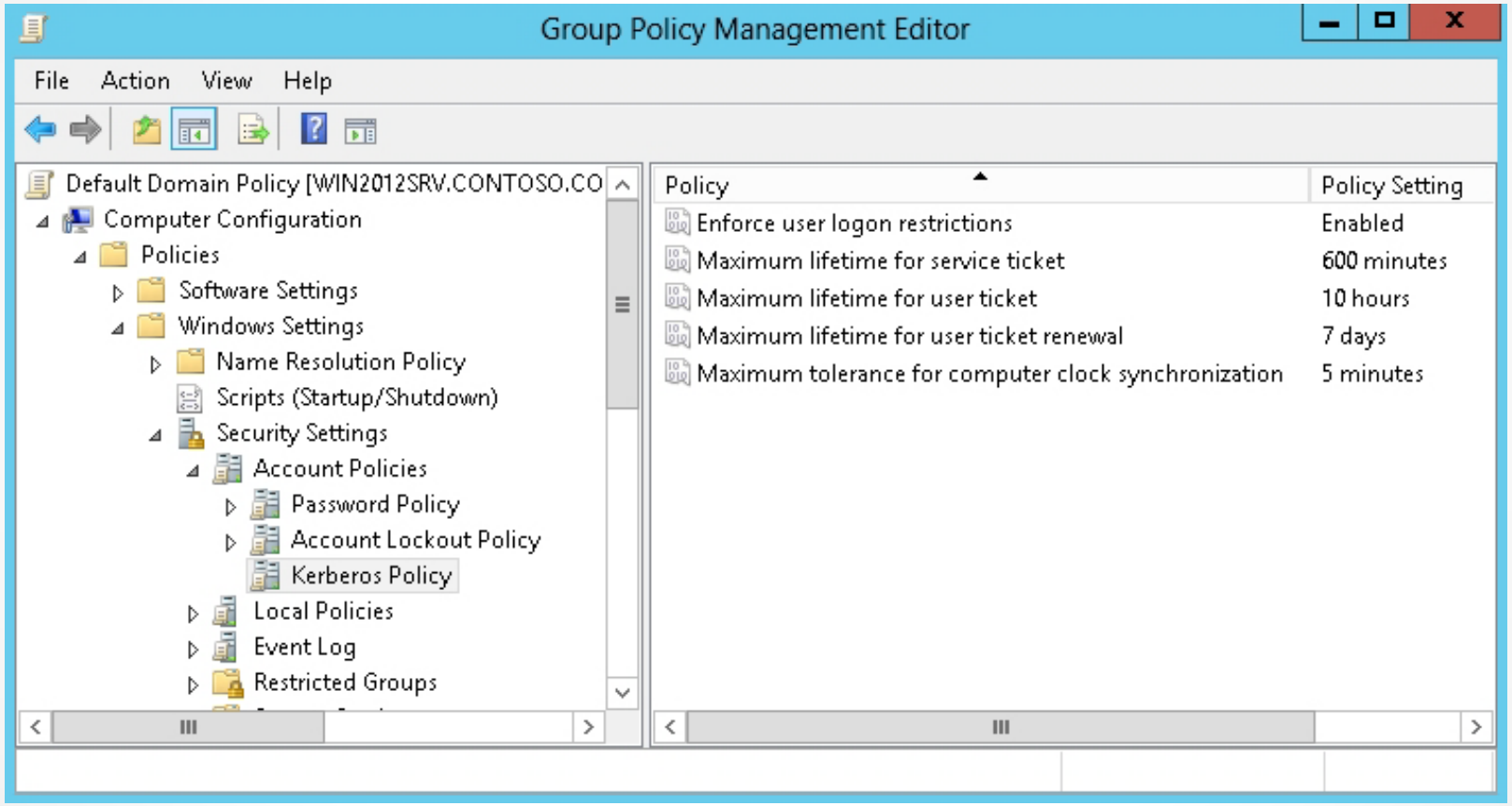
Managing Kerberos

- The ticket is then sent to the client as part of its credentials, which includes the session key encrypted with the client's key/password.
- If the user types the right password, the client can decrypt the session key, present the ticket to the file or SharePoint server, and give the user the shared secret session key to communicate between them.
  - Tickets are time stamped and typically expire after only a few hours.

Managing Kerberos

- Kerberos settings are configured with Group Policies, specifically *\Computer Configuration\Policies\Windows\Settings\Security Settings\Account Policies\Kerberos Policy*
- GPO entries:
  - Enforce user logon restrictions
  - Maximum lifetime for service ticket
  - Maximum lifetime for user ticket
  - Maximum lifetime for user ticket renewal
  - Maximum tolerance for computer clock synchronization

Managing Kerberos



Configuring Kerberos settings

Managing Service Principal Names

- A service or application that is secured by Kerberos must have an identity within the realm that the system exists on:
  - **Identity**: A user account or computer account
  - **Realm**: The domain
- A ***service principal name (SPN)*** is the name by which a client uniquely identifies an instance of a service.
- The client locates the service based on the SPN.

Managing Service Principal Names

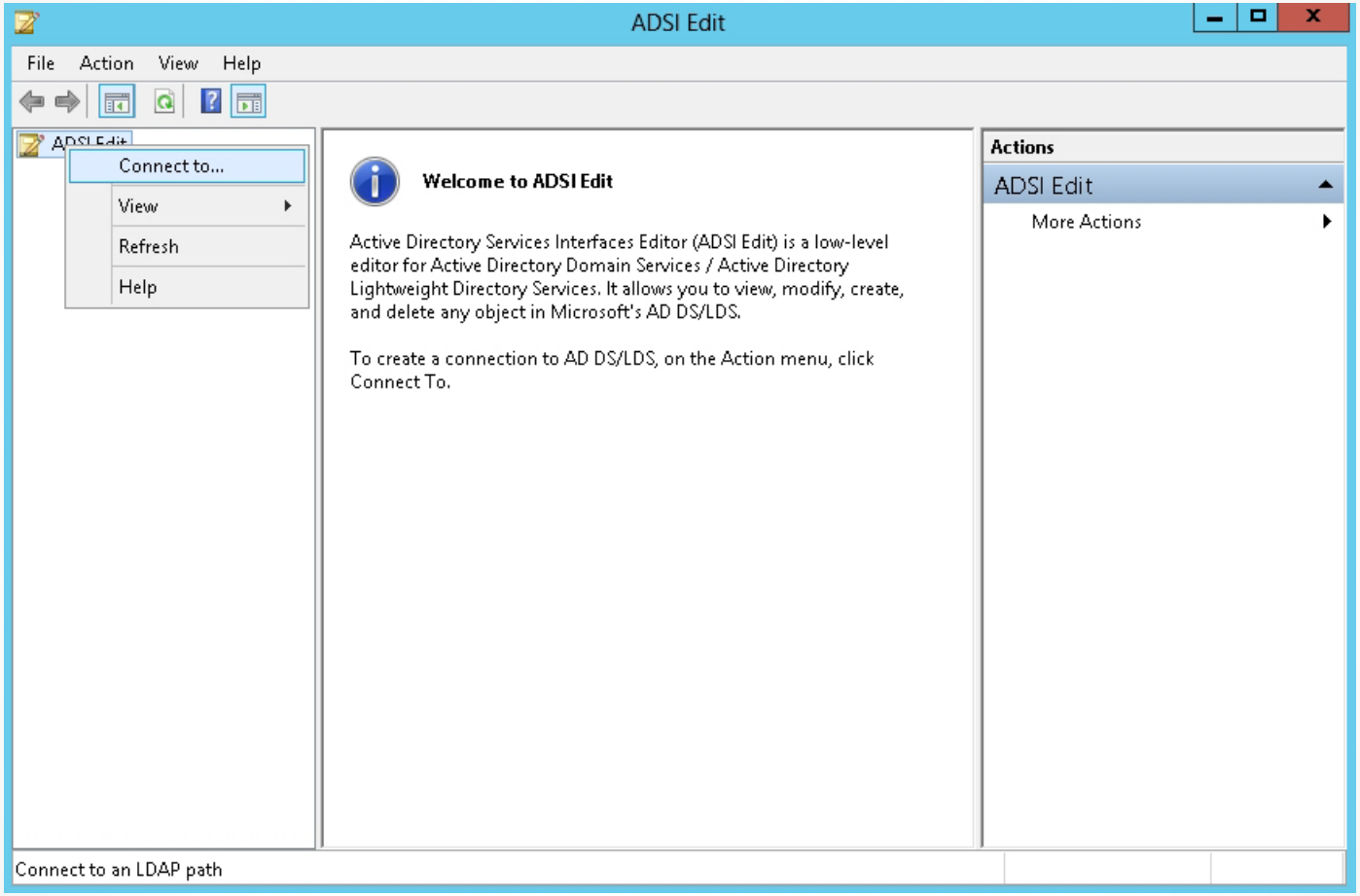
The SPN consists of three components:

- The service class, such as HTTP (which includes both the HTTP and HTTPS protocols) or SQLService
- The host name
- The port (if port 80 is not being used)

Managing Service Principal Names

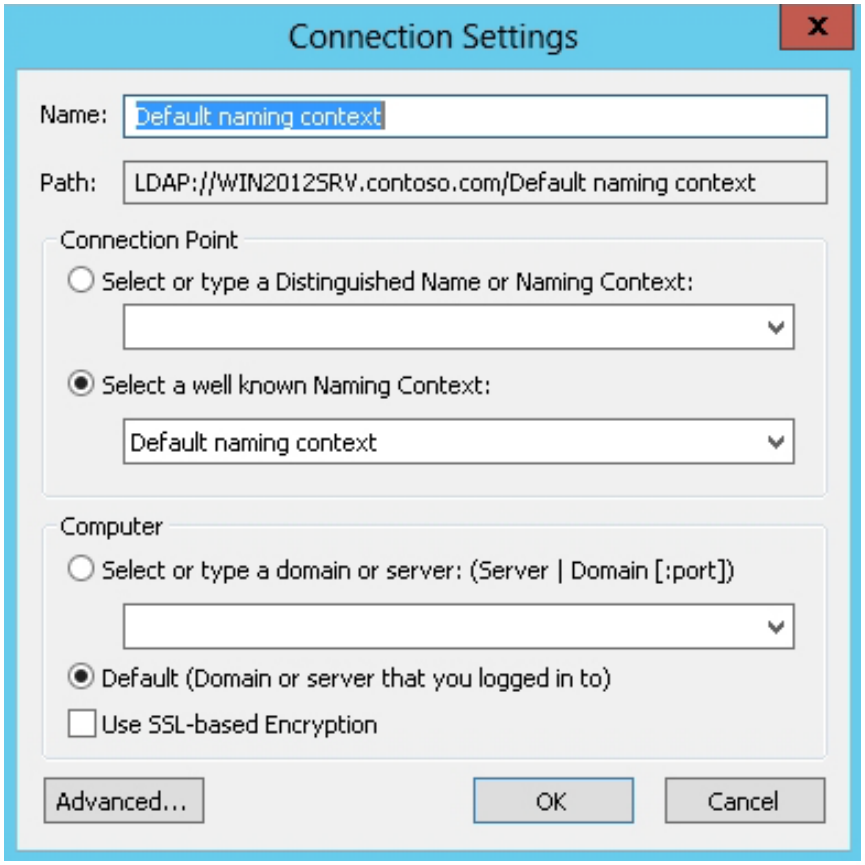
1. When a domain controller's KDC receives the service ticket request from a client, it looks up the requested SPN.
2. The KDC then creates a session key for the service and encrypts the session key with the password of the account with which the SPN is associated.
3. The KDC issues a service ticket, containing the session key, to the client.
4. The client presents the service ticket to the service.
5. The service, which knows its own password, decrypts the session key and authentication is complete.

Use the Managed Service Account with a Service



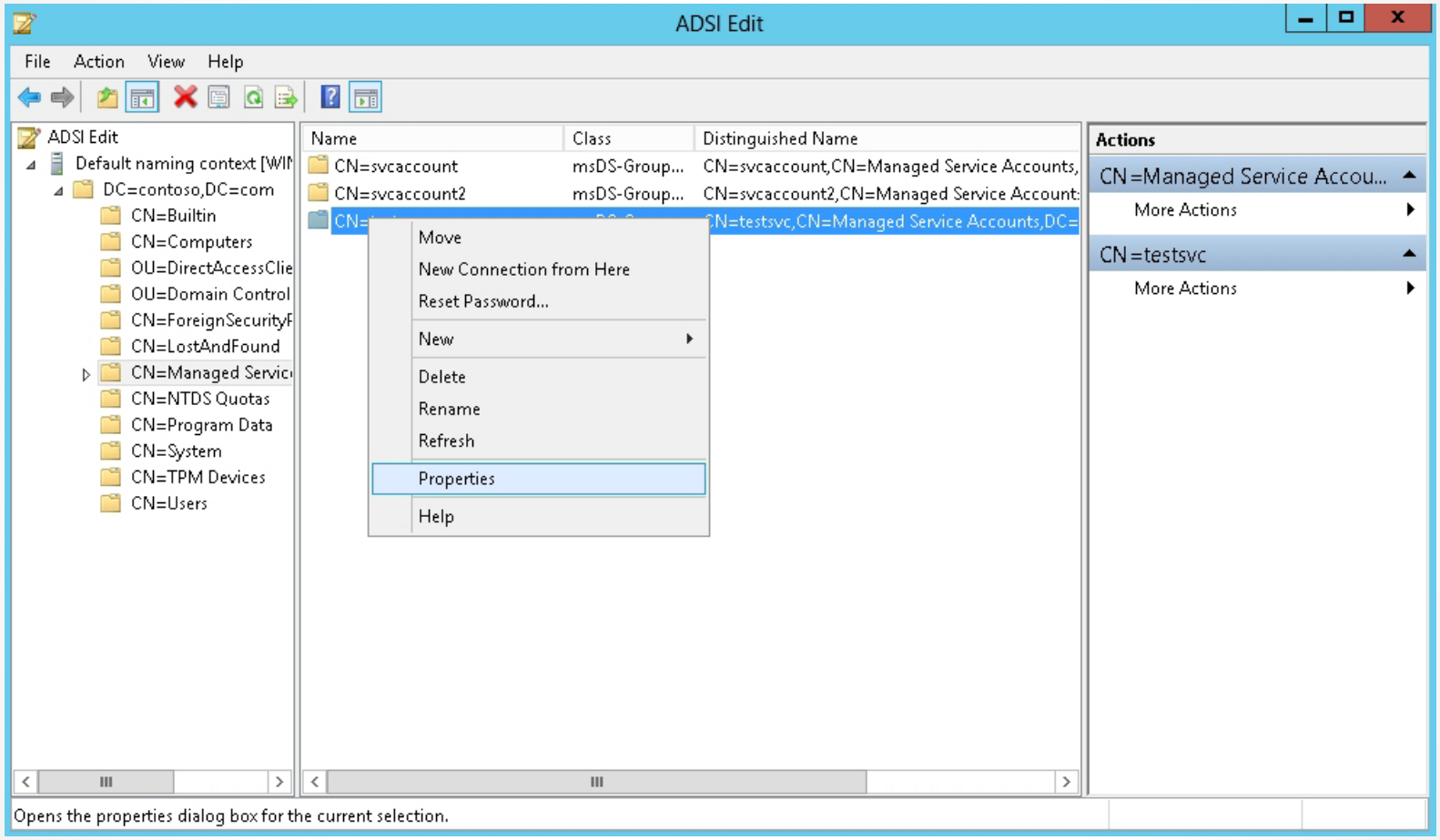
Connecting ADSI Edit to a domain controller

Use the Managed Service Account with a Service



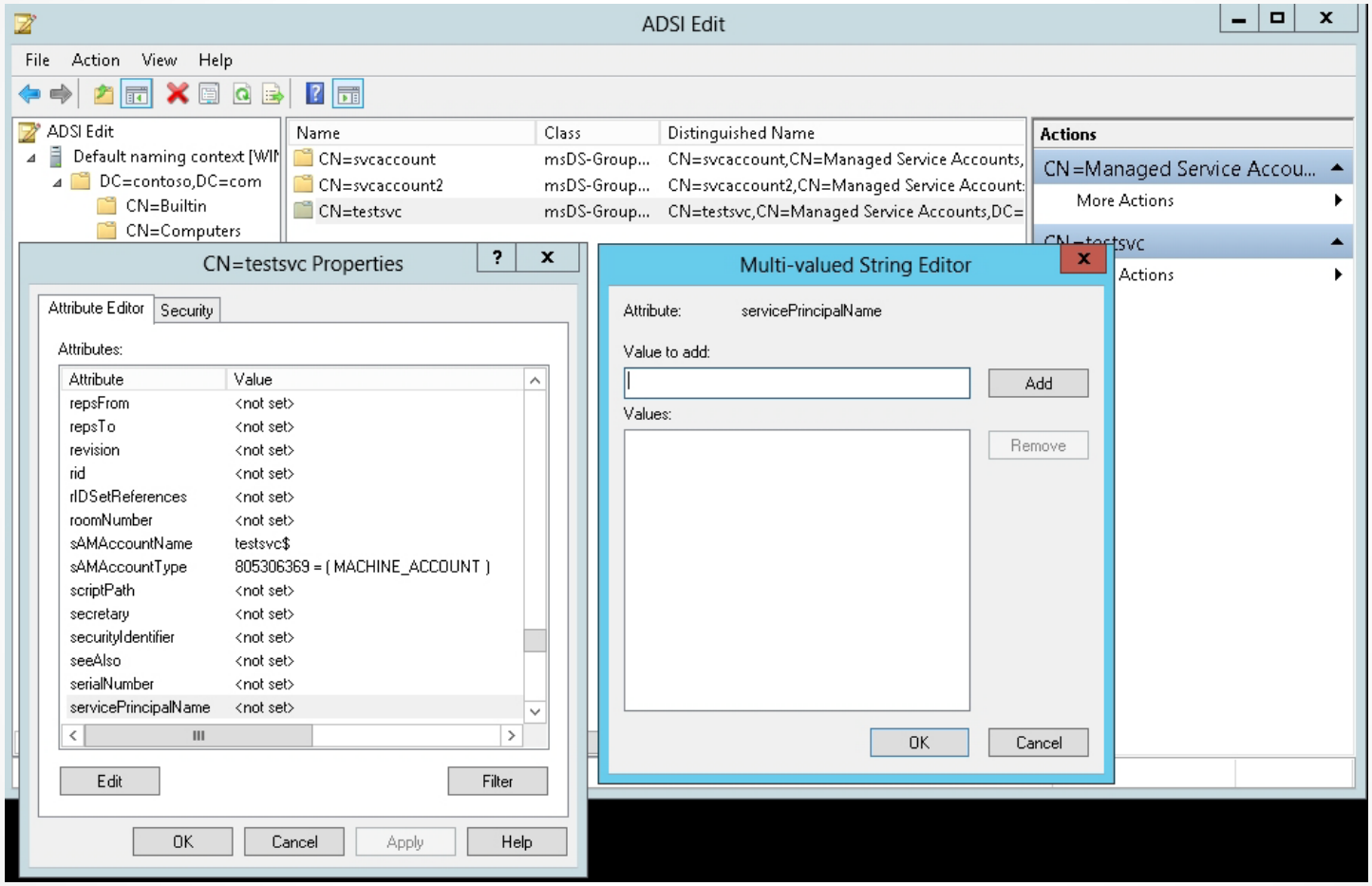
Specifying the connection settings

Use the Managed Service Account with a Service



Opening the properties of an account

Use the Managed Service Account with a Service



Managing the SPNs for an object

Using `setspn.exe` to Add SPNs to an Account

You can use `setspn.exe` to add SPNs to an account. The syntax is:

`setspn <domain\user> -s <SPN>`

whereby:

- `<domain\user>` identifies the security principal to which you want to add an SPN.
- `<SPN>` is the service principal name that you want to add.

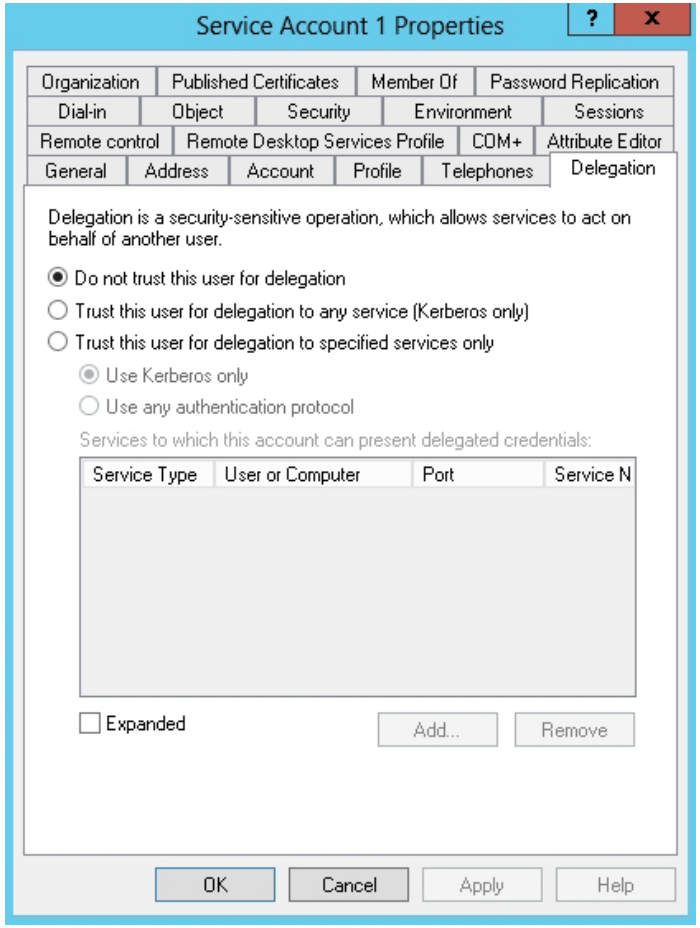
Configuring Kerberos Delegation

Kerberos delegation allows a Kerberos ticket to be created for another service on the originating user's behalf.

To configure Kerberos delegation:

1. Open *Active Directory Users and Computers*.
2. Go to the account that has an SPN.
3. Open the account's properties.
4. Click *Delegation*.

Configuring Kerberos Delegation



Configuring the Kerberos delegation

Configuring Kerberos Delegation

To allow full delegation:

- Select *Trust this user for delegation to any service (Kerberos only)*.

To allow for constrained delegation:

- Select *Trust this user for delegation to specified services only*.

You can then select to use only for Kerberos, or you can specify *Use any authentication protocol*, and then click the *Add* button, to specify which services to be delegated for a user or computer and specify the user or computer.

Managing Service Accounts

Lesson 15: Configuring Service Authentication

Service Accounts

- A ***service account***:
  - Is an account under which an operating system, process, or service runs.
  - Allows the application or service specific rights and permissions to function properly while minimizing the permissions required for the users using the application server.
- Service accounts are used to run Microsoft Exchange Microsoft SQL Server, Internet Information Services (IIS), and SharePoint.

Creating and Configuring Service Accounts

- Service accounts do not use an interactive login.
- Therefore, configure the password not to expire.
- On an account that does not expire, the password is more vulnerable because more time is available for cracking the password.

Creating and Configuring Service Accounts

Guidelines for reducing the risk of using service accounts:

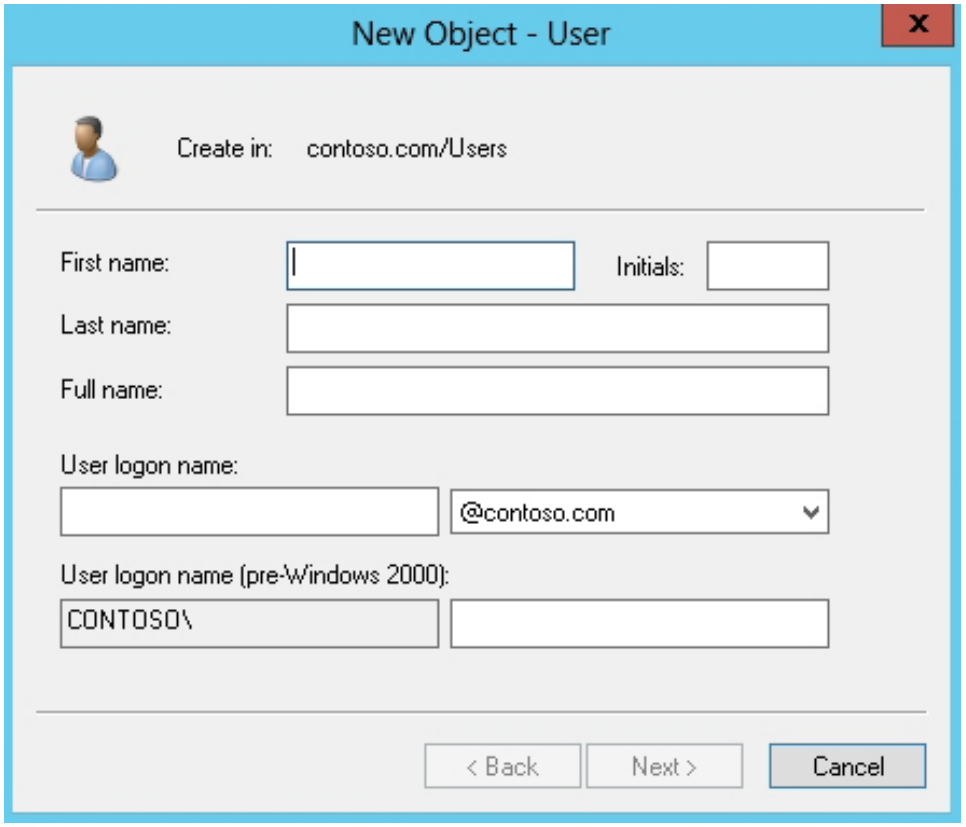
- Require a unique account to run the service on each server.
- If possible, set up the account as a local account rather than a global domain account.
- Use a strong password for the service account.
- Make sure that the password changes often. Of course, when you change the password for the account, you will have to change the password for the services or applications that use the service account simultaneously.

Creating and Configuring Service Accounts

Guidelines for reducing the risk of using service accounts (continued):

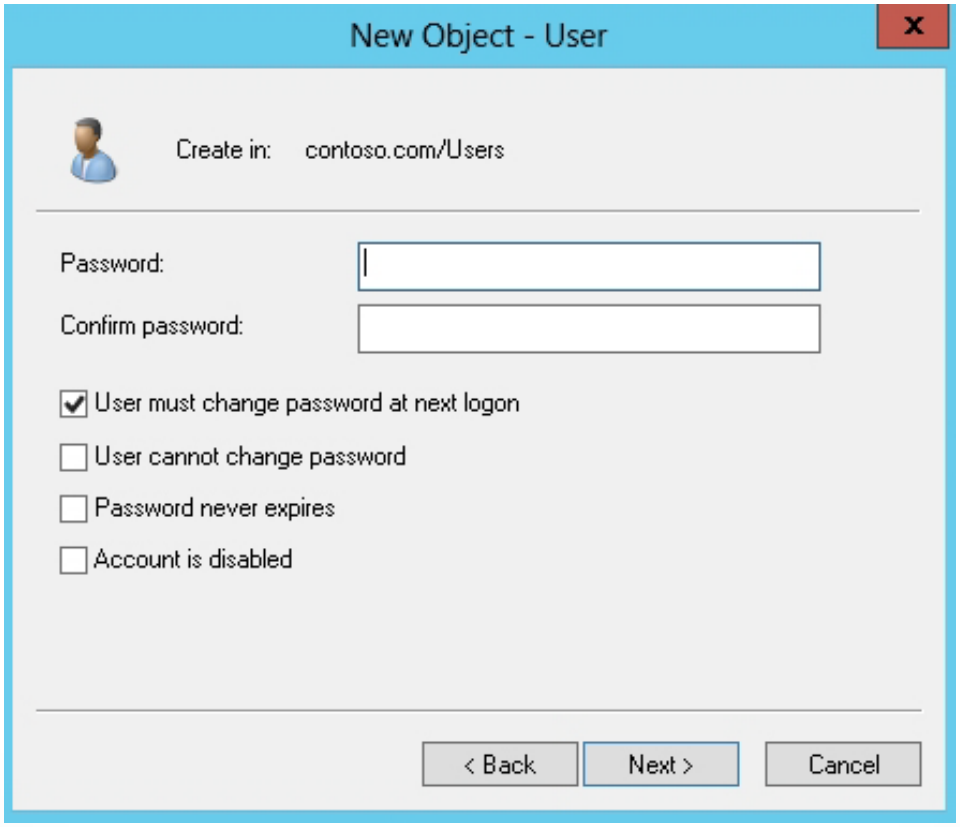
- Give the account the least amount of access (user rights, NTFS permissions, and share permissions) it needs to perform its necessary tasks.
- Do not share the password, and store the password in a safe location.

Create a Service Account



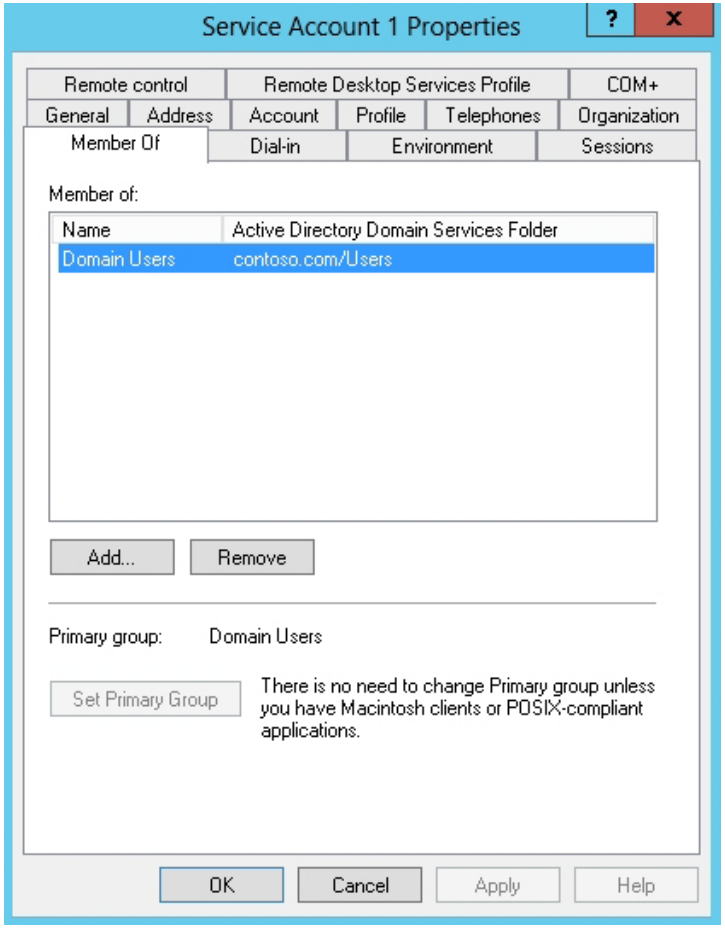
Creating a new user

Create a Service Account



Specifying the password options

Create a Service Account



Configuring service accounts

Managed Service Accounts

- Were introduced with Windows Server 2008 R2.
- Are used to improve the use of the traditional service account in Windows.
- Are an Active Directory `msDS-ManagedServiceAccount` object class that enables automatic password management and SPN management for service accounts.

Creating/Configuring Managed Service Accounts

- Rather than manually changing the account password and the password for the service or application, you use the MSA where the password will automatically change on a regular basis.
- MSAs are stored in Active Directory Directory Sevices (AD DS) as `msDS-ManagedServiceAccount` objects in Windows Server 2008 and `MSDS-GroupManagedServiceAccount` on Windows Server 2012.

Creating/Configuring Managed Service Accounts

- Similar to computer accounts, an MSA establishes a complex, cryptographically random, 240-character password.
  - That password changes when the computer changes its password.
  - By default, this occurs every 30 days.
- An MSA cannot be locked out and cannot perform interactive logons.

MSA Benefits

- Automatic password management
- Simplified SPN management

Where MSAs Are Stored

- MSAs are stored in the `CN=Managed Service Accounts, DC=<domain>, DC=<com>` container.
- This container can be used if you enable the Advanced Features option in the View menu within Active Directory Users and Computers.
- You can also see the container using the Active Directory Administrative Center.

MSA Requirements

- Windows Server 2008 R2 or Windows Server 2012 domain controller
- .NET Framework 3.5.x
- Active Directory module for Windows PowerShell

Using Windows PowerShell

- Before you can create an MSA object type, you need to create a key distribution services root key for the domain.
- To create the root key, run the following cmdlet from the Active Directory PowerShell module for Windows PowerShell:

```
Add-KDSRootKey -EffectiveTime ((Get-
Date).AddHours(-10))
```

- You specify 10 hours so that AD DS replication has a chance to replicate the changes to other domain controllers in the domain. For testing environments, you can use `add-kdsrootkey -EffectiveImmediately` instead.

Create and Associate an MSA Using PowerShell

To create and associate an MSA:

1. Create an Active Directory AD service account:

```
New-ADServiceAccount -Name <MSA_Name>
-DNSHostname <DNS name of Domain_Controller>
```

2. `Add-ADComputerServiceAccount` associates the MSA with a computer account in the AD DS domain:

```
Add-ADComputerServiceAccount -identity
<Host_Computer_Name>
-ServiceAccount <MSA_Name>
```

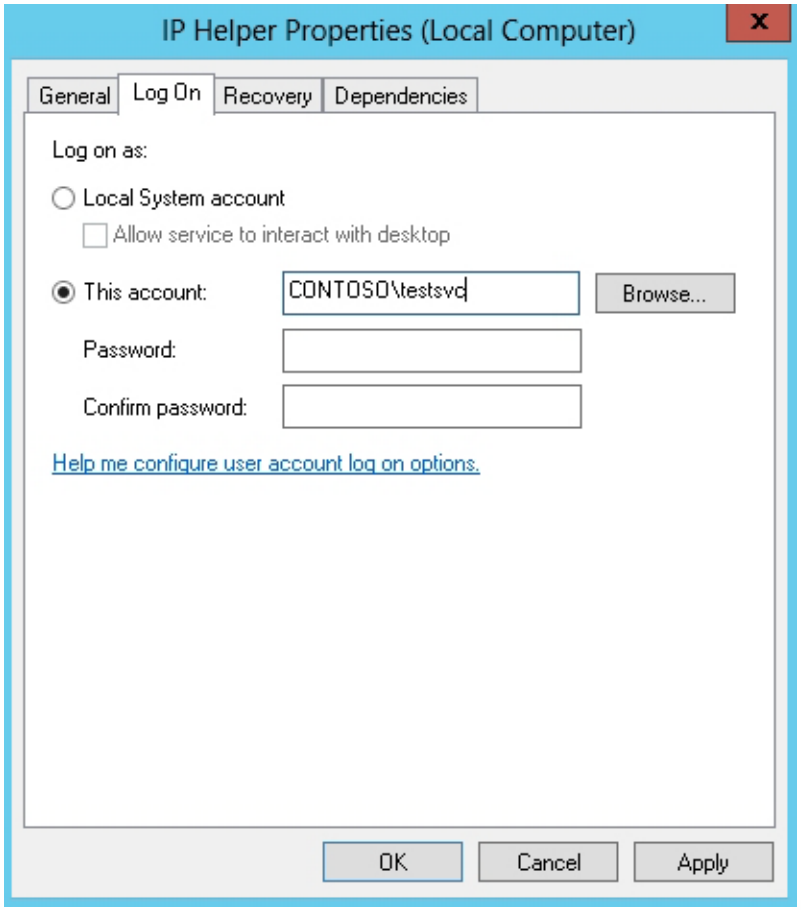
Create and Associate an MSA Using PowerShell

To create and associate an MSA (continued):

3. `Install-ADServiceAccount` installs the MSA on a host computer in the domain, and makes the MSA available for use by services on the host computer:

```
Install-ADServiceAccount -Identity
<MSA_Name>
```

Use the MSA with a Service



Using the MSA

Create and Associate an MSA Using PowerShell

If you move a service to another computer and you want to use the same managed service account on the target system, you must first use:

- The `Uninstall-ADServiceAccount` cmdlet to remove the managed service account from the current computer
- The `Install-ADServiceAccount` cmdlet on the new computer

Create and Associate an MSA Using PowerShell

- When you create the new MSA, you can specify the SPN by using the `-ServicePrincipalNames <SPN_string>`.

```
New-ADServiceAccount –Name svcaccount
–DNSHostname win2012srv.contoso.com
-ServicePrincipalNames
HTTP/portal.contoso.com,HTTP://portal
```

Create and Associate an MSA Using PowerShell

- To change the parameter for a service account, use `Set-ADServiceAccount`.
- To delete a group service account using a Windows PowerShell command, use `Remove-ADServiceAccount`.
- To display a list of the service accounts, use `Get-ADServiceAccount`.

Creating/Configuring Group Managed Service Accounts

- If you have a cluster or farm where you need to run the system or application service under the same service account, you cannot use Managed Service Accounts.
- ***Group Managed Service Accounts*** are similar to Managed Service Accounts, but they can be used on multiple servers at the same time.

Creating/Configuring Group Managed Service Accounts

To use Group Managed Service Accounts, you must:

- Have one domain controller that is running Windows Server 2012, so that it can store managed password information
- Create a KDS root key.

Create a Group MSA Using PowerShell

- Use `New-ADServiceAccount` with the `-PrincipalsAllowedtoRetrieveManagedPassword` option to define one or more comma-separated computer accounts or AD DS groups.
- You can then go to each server and use `Install-ADServiceAccount`.

Lesson Summary

- Authentication is the act of confirming the identity of a user or system and is an essential part used in authorization.
- NT LAN Manager (NTLM) is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users.
- Kerberos is a computer network authentication protocol, which allows hosts to prove their identity over a non-secure network in a secure manner. It can also provide mutual authentication between the user and server.
- Kerberos settings are configured with Group Policies, specifically \Computer Configuration\Policies\Windows\Settings\Security Settings\Account Policies\Kerberos Policy.

Lesson Summary

- You can use ADSI Edit or use the `setspn` command to add SPNs to an account.
- Kerberos delegation allows a Kerberos ticket to be created for another service on the originating user's behalf. This can be done with full delegation or with constrained delegation.
- Constrained delegation is Kerberos delegation that can be executed only against a limited set of services.
- A service account is an account under which an operating system, process, or service runs.
- Managed Service Accounts (MSAs) are an Active Directory `msDS-ManagedServiceAccount` object class that enables automatic password management and SPN management for service accounts.
- Group Managed Service Accounts are similar to Managed Service Accounts, but they can be used on multiple servers at the same time.