10
Securing Active Directory

From a business perspective, Active Directory needs to be an available, confidential attribute store with absolute integrity. The security measures in this chapter detail how to achieve a higher level of confidentiality and integrity.

The following recipes will be covered in this chapter:

- Applying fine-grained password and account lockout policies
- Backing up and restoring GPOs
- Backing up and restoring the Active Directory database
- Working with Active Directory snapshots
- Managing the DSRM passwords on domain controllers
- Implementing LAPS
- Managing deleted objects
- Working with group Managed Service Accounts
- Configuring the advanced security audit policy
- Resetting the KRBTGT secret
- Using SCW to secure domain controllers
- Leveraging the Protected Users group
- Putting authentication policies and authentication policy silos to good use
- Configuring Extranet Smart Lockout

Applying fine-grained password and account lockout policies

Active Directory comes with a built-in password policy. Admins can configure stricter password policies and account lockout policies. This way, privileged accounts can be configured with more secure password and account lock-out settings. This recipe shows how.

Getting ready

To apply fine-grained password and account lockout policies, you should sign into a domain controller or a member server and/or device with the **Remote Server Administration Tools** (**RSAT**) for Active Directory Domain Services installed. Ideally, the domain controller or member server runs Windows Server 2012, or a newer version of Windows Server.

Sign in with an account that is a member of the Domain Admins group, or with an account that is delegated to fine-grained password and account lockout policies in the domain.

Fine-grained password and account lockout policies require the Windows Server 2008 **Domain Functional Level** (**DFL**), or a higher version of the DFL.

How to do it...

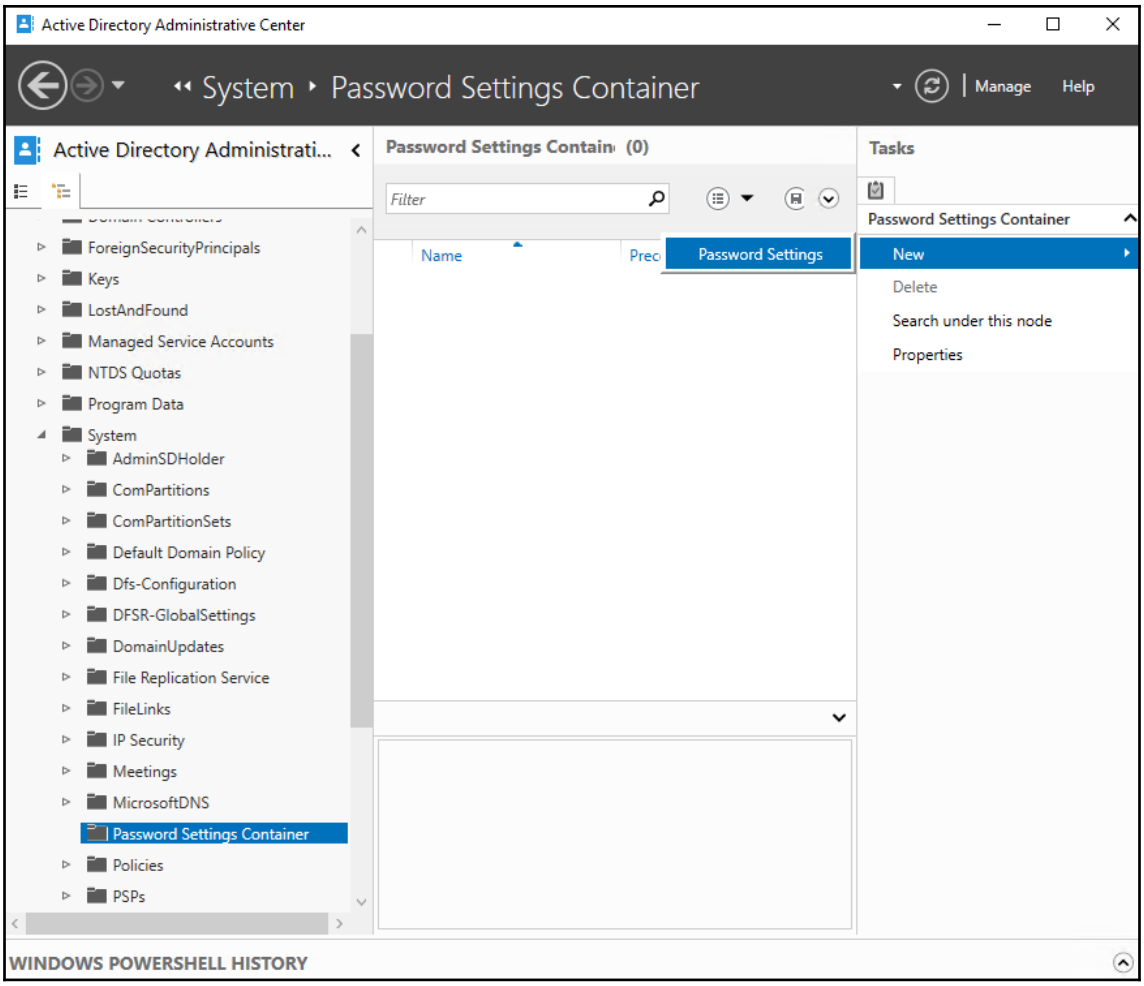
There are two ways to manage fine-grained password and account lockout policies:

- Using the Active Directory Administrative Center
- Using the Active Directory module for Windows PowerShell

Using the Active Directory Administrative Center

Follow these steps to create and apply a fine-grained password and account lock-out policy in the Active Directory Administrative Center:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. In the left navigation window, switch to **Tree view**.
3. Navigate to the **Password Settings Container** under the **System** container:



4. Right-click the **Password Settings Container** and select **New** and then **Password Settings** from the menu.
   The **Create Password Settings** screen appears.
5. In the **Create Password Settings** screen, in the **Password Settings** section, fill in values for the **Name** and **Precedence** fields.
6. Enable every setting that you want to apply to meet the requirements by selecting them.

There is no inheritance from other fine-grained password and account lockout policies. Only one policy applies to any user account. Define all the settings that you want to apply.

7. In the **Directly applies to** section, press the **Add…** button.
8. In the **Select Users or Groups** window, type the name of the user account(s) or groups you want to fine-grained password policy to directly apply to, or click the **Advanced** button to search for the user account(s) and/or groups.
9. Click **Check Names**.
10. Click **OK** to have the fine-grained password policy directly apply to the user(s) and/or group(s).
11. Click **OK** to create the fine-grained password and account lock-out policy.

Using the Active Directory Module for Windows PowerShell

To create a fine-grained password and account lock-out policy, use the following lines of PowerShell on a system with the Active Directory Module for Windows PowerShell installed:

```
Import-Module ActiveDirectory

New-ADFineGrainedPasswordPolicy PolicyName -ComplexityEnabled $true
-LockoutDuration "00:30:00" -LockoutObservationWindow "00:30:00"
-LockoutThreshold "5" -MaxPasswordAge "42.00:00:00" -MinPasswordAge
"7.00:00:00" -MinPasswordLength "15" -PasswordHistoryCount "21" -Precedence
"1" -ReversibleEncryptionEnabled $false -ProtectedFromAccidentalDeletion
$true
```

To apply a fine-grained password and account lock-out policy, use the following lines of PowerShell on a system with the Active Directory Module for Windows PowerShell installed:

```
Import-Module ActiveDirectory

Add-ADFineGrainedPasswordPolicySubject PolicyName -Subjects GroupName
```

How it works...

Active Directory comes with a built-in password policy. This is the password policy that is automatically set at the Active Directory domain level. This default policy does not enable account lock-out.

The password policy applies when the password is changed and when it is set by an admin. Account lockout policies observe bad password attempts. When a bad password is typed, it is added to the bad password count. When this count reaches the limit within the time specified as the observation period, the account is locked for the duration of the time-out period. Accounts can be locked indefinitely. In this case, accounts need to be unlocked by a person using their admin account or otherwise privileged account.

Admins can configure stricter password policies and also account lockout policies on the domain level, but they can also configure these policies granularly as follows:

- Per user account
- Per group

Only one password policy can apply. When multiple fine-grained password policies are applied, the policy applied to a user account directly applies. When a user account has memberships in multiple groups, the password policy with the lowest value for precedence is applied. All other password policies are ignored. The precedence value can be interpreted as a priority. Specifying unique precedence values for password policies is key to having the right policy applied.

In scenarios where lock-out is configured identically over all password policies, the maximum lifetime for the password makes for an excellent precedence value.

There's more...

When you're unsure which fine-grained password and account lock-out policy applies, look at the user account's **msDS-ResultantPSO** attribute. It exposes the reference to the password policy that is applied.

The `Get-ADUserResultantPasswordPolicy` PowerShell cmdlet can also be used for this purpose.

Backing up and restoring GPOs

The Group Policy Management Console does not offer to rollback changes in **Group Policy Objects** (**GPOs**). However, when in the process of modifying a GPO, a step is added to create a backup of the GPOs, inadvertent settings can be rolled back by restoring a previous backup.

This recipe shows what that step would look like and how to restore a GPO.

Getting ready

You should sign into a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed. Ideally, the domain controller or member server runs Windows Server 2012, or a newer version of Windows Server.

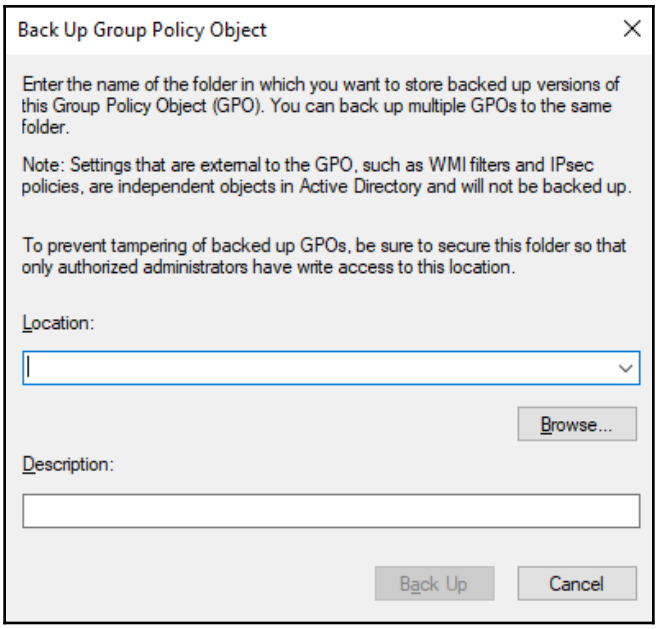
Sign in with an account that is a member of the Domain Admins group.

In contrast to the delegation of creating, linking, managing, editing, and reading GPOs, backing up and restoring GPOs cannot be delegated.

How to do it...

Follow these steps to back up GPOs:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left navigation pane, expand the forest, then the **Domains** node and then the domain for which you want to backup the GPOs.
3. Select the **Group Policy Objects** node and right-click it.
4. From the menu, select **Back Up All…**.
   The **Back Up Group Policy Object** window appears:

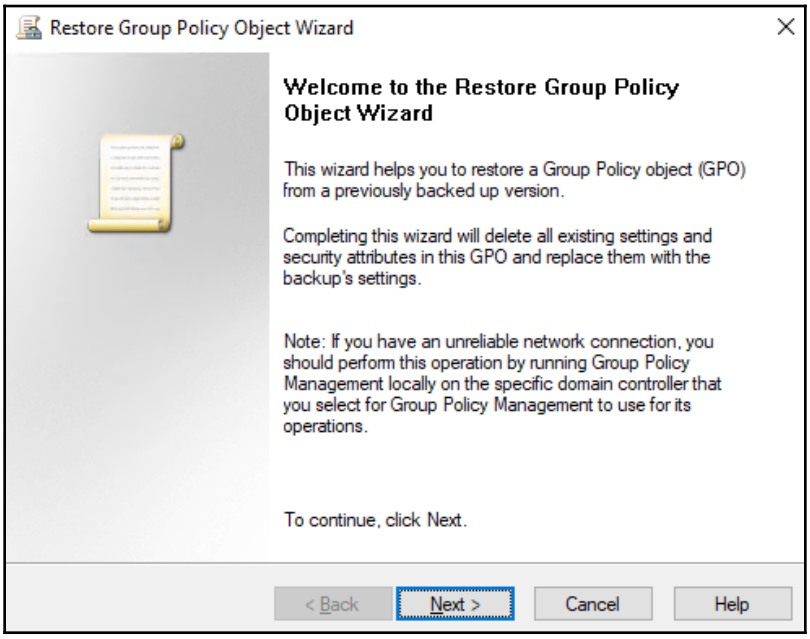


5. In the **Back Up Group Policy Object** window, for **Location:** provide a location to store the backups. Use the **Browse...** button to locate and/or create a folder for Group Policy backups.
6. Provide a description in the **Description:** field.
7. Click the **Back Up** button.
8. In the **Backup** window, review the statistics of the backup and click **OK** when done.

Follow these steps to restore inadvertent settings for a GPO:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left navigation pane, expand the forest, then the **Domains** node, and then the domain for which you want to restore a GPO.
3. Expand the **Group Policy Objects** node and locate the GPO you want to restore from a previous backup.
4. Select the GPO to restore.

5. Right-click the GPO and select **Restore from Backup** from the menu. The **Restore Group Policy Wizard** window appears:

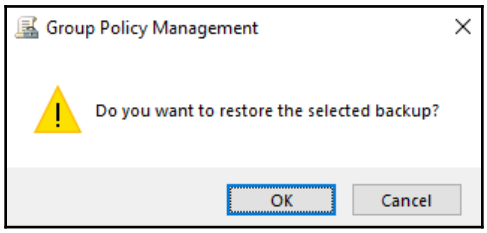


6. On the **Welcome to the Restore Group Policy Object Wizard** screen, Click **Next >**.
7. On the **Backup Location** screen, type the location of previous backups, or use the **Browse...** button to look it up. Click **Next >**.
8. On the **Source GPO** screen, select the GPO which you want to restore. Click **Next >**.
9. On the **Completing the Restore Group Policy Object Wizard screen,** click **Finish** to start restoration.
10. In the **Restore** window, review the outcome, and click **OK** when done.

Follow these steps to restore an inadvertently deleted GPO:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left navigation pane, expand the forest, then the **Domains** node and then the domain for which you want to restore a GPO.
3. Select the **Group Policy Objects** node and right-click it.
4. From the menu, select **Manage Backups…**.

5. Click the **Browse** button to navigate to the folder that contains the previous backups.
6. Select the GPO(s) that you want to restore from the backup.
7. Click **Restore** to start restoration.
   The **Group Policy Management** popup window appears:



8. In the **Group Policy Management** popup window, click **OK**.
9. In the **Restore** window, review the outcome, and click **OK** when done.

How it works...

When you create backups of GPOs, the settings for these objects are stored in a backup file. Then, when inadvertent changes are made to these objects, they can be restored from the backup file.

Depending on the purpose of the GPO backup, the location to store the backups can be on the domain controller or in a remote location. Storing on a domain controller may be a good option for a fast restore test, duplicating group policies between Active Directory forests, or versioning. However, for true backups, always use a remote location.

Alternatively, you can implement the **Advanced Group Management Tool** (**AGPM**).

There's more...

Backups for GPOs can only be restored in the same Active Directory forest. To recreate GPOs from one Active Directory forest to another, use the **Export** and **Import** functionality in the Group Policy Management Console.

Backing up and restoring Active Directory

To avoid the situation where Active Directory, the backbone of every Microsoft-oriented networking infrastructure, is irreversibly lost, Active Directory restores should be performed. Restores should be performed regularly in an isolated environment to make sure backups can be restored and procedures are up to date and known to admins.

This recipe shows how to create backups of Active Directory using Windows Backup.

Getting ready

To make a backup of a domain controller, sign into a domain controller with a user account that is a member of the Domain Admins group or of the Backup Operators group.

To restore a domain controller, you need to know the **Directory Services Restore Mode** (**DSRM**) password for the domain controller.

First off, the Windows Backup feature needs to be installed. Use the following PowerShell one-liner in an elevated PowerShell window to do so:

```
Install-WindowsFeature Windows-Server-Backup
```

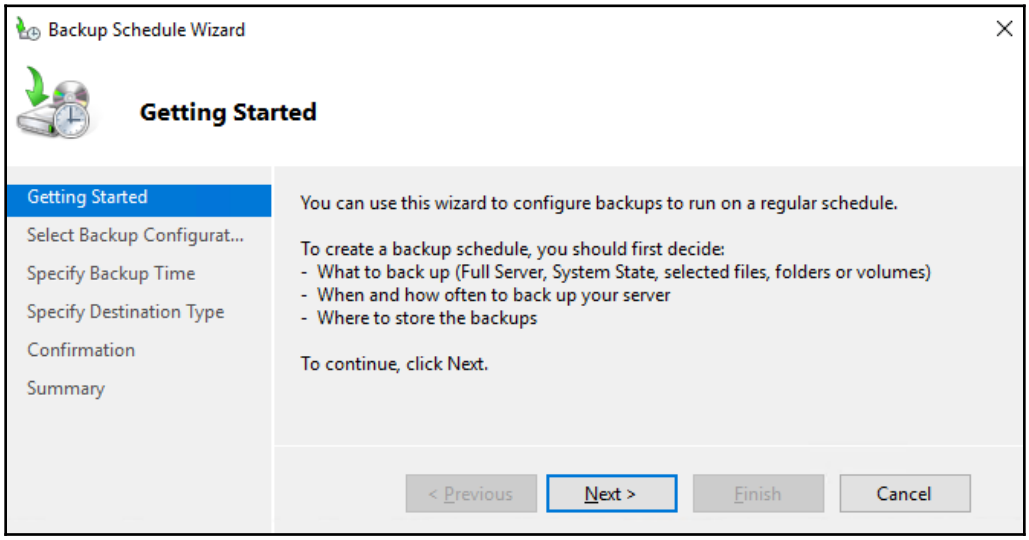
To avoid any dependencies on Active Directory-integrated network and file access, make sure you back up to a dedicated (USB) hard drive for physical domain controllers or to a separate LUN in the virtualization fabric for virtual domain controllers. When working with USB hard disks, purchase at least two devices for off-site storage options and replacement upon failure scenarios.

How to do it...

Follow these steps to create backups of a domain controller:

1. Log on to the domain controller.
2. Plug in the dedicated hard drive or LUN to which you want to back up. Install drivers, if necessary.
3. Start **Server Manager** (`servermanager.exe`) if it doesn't start automatically by default.
4. From the **Tools** menu in the top grey bar, choose **Windows Server Backup**.
5. In the left navigation pane, select **Local Backup**.

6. In the action pane on the right side of **Windows Server Backup**, click **Backup Schedule…**.
   The **Backup Schedule Wizard** window appears:



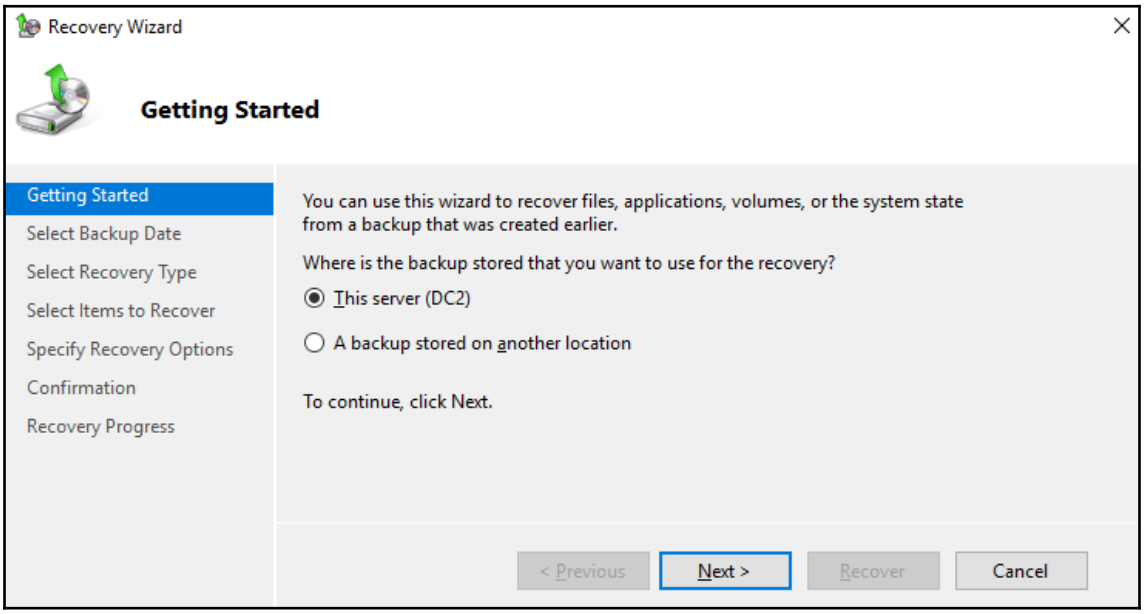
7. In the **Getting Started** screen, click **Next >**.
8. In the **Select Backup Configuration** screen, select **Full Server (recommended)**. Alternatively, select **Custom**, but in this case, always select the **System State** on the **Select Items for Backup** screen as part of the backup configuration if you want to be able to restore the domain controller functionality from the backups. Click **Next >**.
9. In the **Specify Backup Time** screen, select **Once a day**. Choose a time that is outside the typical opening hours or working day(s) in your organization. If you have other processes running out-of-hours, be sure not to collide with them. Click **Next >**.
10. In the **Specify Destination Type** screen, select **Back up to a hard disk that is dedicated for backups (recommended)**. Click **Next >**.
11. In the **Select Destination Disk** screen, select the removable disk to backup to and click **Next >**.
12. In the **Confirmation** screen, click **Finish**.

Windows Server Backup can also be used on the Command Prompt (`cmd.exe`). To create an instant System State Backup to a hard disk attached as `F:\`, use the following command on an elevated Command Prompt (`cmd.exe`):

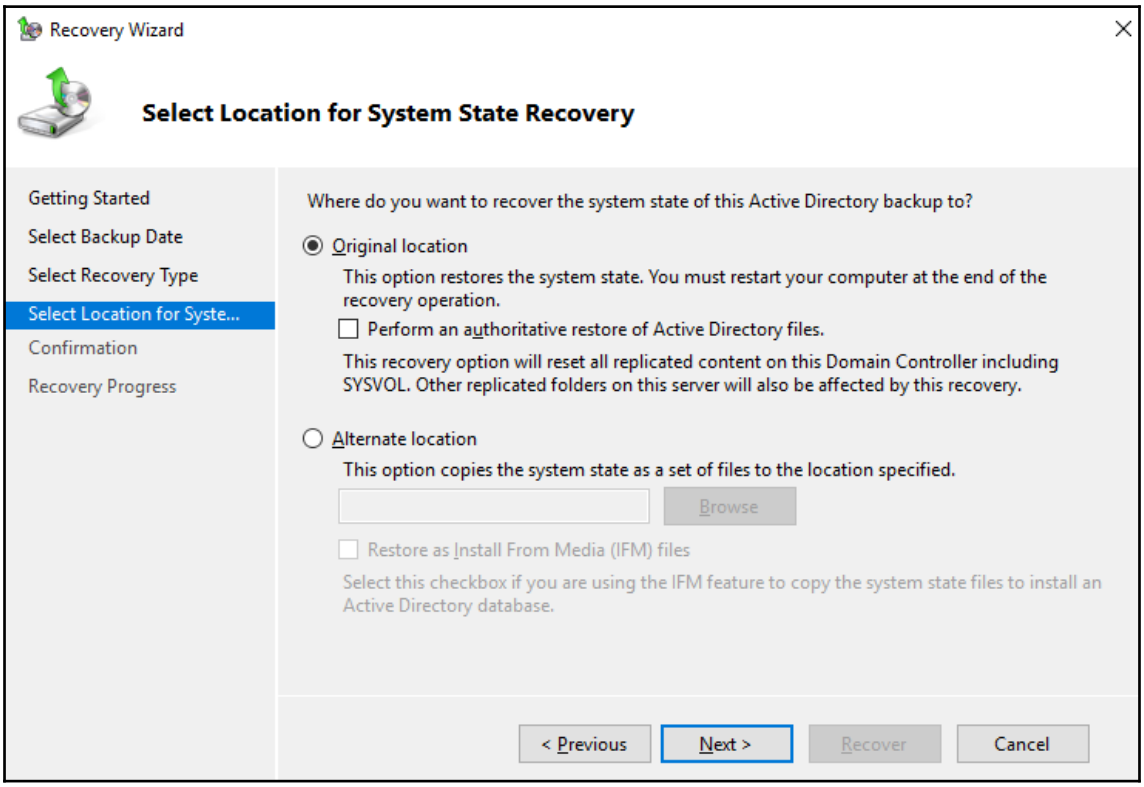
```
wbadmin.exe start systemstatebackup --backuptarget:F:
```

Follow these steps to restore a backup of a domain controller:

1. Start the domain controller in DSRM.
2. Log on to the domain controller with the username **Administrator** and the DSRM password as the password.
3. Plug in the dedicated hard drive or LUN where you want to restore from.
4. Start **Server Manager** (`servermanager.exe`) if it doesn't start automatically by default.
5. From the **Tools** menu in the top grey bar, choose **Windows Server Backup**.
6. In the left navigation pane, select **Local Backup**.
7. In the action pane on the right side of **Windows Server Backup**, click **Recover...**. The **Recovery Wizard** appears:



8. In the **Getting Started** screen, select the **This server (DC2)** option and click **Next >**.
9. In the **Select Backup Date** screen, select the backup to restore. Click **Next >**.
10. In the **Select Recovery Type** screen, select **System State** and click **Next >**.
    You reach the **Select Location for System State Recovery** screen:



11. In the **Select Location for System State Recovery** screen, select **Original location** and click **Next >.**
    Optionally, select the **Perform an authoritative restore of Active Directory files.** option.
12. In the **Confirmation** screen, click **Finish**.
13. After restoration, restart the domain controller normally.

Command Prompt (`cmd.exe`) is also available in this scenario; use the following command to get the backups on the hard disk connected as `F:\`:

```
wbadmin.exe get versions

wbadmin.exe start systemstaterecovery --version:04/26/2019-21:00
```

Add the `--AuthSysvol` parameter if you want to restore the domain controller authoritatively.

How it works...

By creating a backup of the system state, all the information to restore a domain controller is copied off the system and onto removable media. This way, when a domain controller becomes non-functional, the backup can be used to restore the functionality to a new Windows Server or to boot up from the backup media to restore the entire domain controller.

Windows Server Backup uses Volume Shadow Copies with the Active Directory VSS Writer to make a backup of the Active Directory files while they are in use. This way, there is no need to stop the Active Directory Domain Services service to make a consistent backup. In most third-party applications, this functionality is called **Application-consistent backups**.

Domain controllers can be restored authoritatively or non-authoritatively. When restored authoritatively, the restored domain controller will take the role of authoritative replication partner for Active Directory and SYSVOL replication; all domain controllers will assume that its version of the database and SYSVOL are the truth. When restoring non-authoritatively, the domain controller will report itself as a new Active Directory replication partner and replicate from other domain controllers, ignoring any changes it might have made before being restored.

The DSRM password for the domain controller is stored on the system and provides the ability to logon with a local administrator account when the Active Directory Domain Services service is not running. When the service runs on a domain controller, this password cannot be used. Document the password properly.

As modern malware attacks on environments feature invalidating backups, make sure to store backups for domain controller off the network, and ideally off-site.

Working with Active Directory snapshots

This recipe shows how to work with Active Directory snapshots as an alternative to having to restore entire backups for a domain controller to restore a couple of objects.

Getting ready

To work with snapshots for a domain controller, sign into a domain controller with a user account that is a member of the Domain Admins group or of the Backup Operators group.

How to do it...

1. To make a snapshot, type the following command in an elevated Command Prompt (`cmd.exe`) window:

   ```
   ntdsutil.exe "activate instance ntds" "snapshot" "create" q q
   ```

2. To view all snapshots, type the following command in an elevated Command Prompt (`cmd.exe`) window to get a numbered list of all available snapshots:

   ```
   ntdsutil.exe "activate instance ntds" "snapshot" "list all" q q
   ```

3. To mount a snapshot, type the following command in an elevated Command Prompt (`cmd.exe`) window, using the number of the snapshot you want to mount from the previous command:

   ```
   ntdsutil.exe "activate instance ntds" "snapshot" "mount GUID" q q
   ```

   The preceding command will output the folder where the database is mounted.

4. Run the following command to expose it as an LDAP Store:

   ```
   dsamain.exe -dpbath "Location from previous command" -ldapport
   PortNumber
   ```

Keep this command running for as long as you want the LDAP Server running.

To look up information, use the cmdlets in the Active Directory Module for Windows PowerShell. Specify the `-Server` parameter and type the hostname of the server, and the port number separated by a semi-colon, as follows:

```
Get-ADComputer -Identity * -Server Localhost:PortNumber
```

How it works...

Snapshots for Active Directory may be useful in scenarios where the organization has a need to compare information from a certain point in time (the time the snapshot was taken) with the information from another point in time (for instance: today).

Creating an Active Directory snapshot requires the Volume Shadow Copy functionality and a functional Active Directory VSS Writer. These features are available by default on Windows Server. However, if anything is not working, check them first.

There's more...

When you want to transfer information between a snapshot and the Active Directory, we can use tools like `ldifde.exe` and `csvde.exe`. There is no native tooling available to perform these kinds of actions.

Managing the DSRM passwords on domain controllers

This recipe shows how to manage the password to sign in to domain controllers when the Active Directory Domain Services service is not running.

Getting ready

To make a backup of a domain controller, sign into a domain controller with a user account that is a member of the Domain Admins group, the Backup Operators group, or the Server Operators group.

For the scenario where the DSRM Administrator password is automatically synchronized with an account in Active Directory, create a disabled user account with a strong password. Document the password in a password vault. Additionally, make sure all domain controllers run Windows Server 2008, or newer versions of Windows Server.

How to do it...

1. To manually reset the DSRM Administrator password on a domain controller, type the following command in an elevated Command Prompt (`cmd.exe`) window when the domain controller is running fine:

```
ntdsutil.exe
> set dsrm password
> reset password on server null
```

2. Type or paste the password to use as the DSRM Administrator password:

```
> quit
> quit
```

3. To synchronize the DSRM Administrator password on a domain controller, type the following command in an elevated Command Prompt (`cmd.exe`) window, when the domain controller is running and replicating fine:

```
ntdsutil.exe "set dsrm password" "sync from domain account
DSRMDCXUser" quit quit
```

As there is no interaction needed with the preceding command and there are no passwords involved, this command line is suitable to be placed on one line and, therefore, suitable to be rolled out using Group Policy preferences.

How it works...

When the Active Directory Domain Service service is not running, or the domain controller is non-functional, you'll need a way to log on to the domain controller. As the Active Directory database is not available, a special password is maintained for the domain controller-local built-in Administrator account.

When a Windows Server-based member server is promoted to a domain controller, this password is configured as the DSRM Administrator password.

The password can be managed in two ways:

- The manual reset scenario
- The domain account password sync scenario

In the first scenario, the password is set manually per domain controller and then to be documented in a password vault.

In the second scenario, the domain controller is instructed to synchronize the password from a specific Active Directory account. This password is then set manually and then to be documented.

A recommended practice is to have different passwords for each domain controller and reset the passwords yearly.

Implementing LAPS

Microsoft's free **Local Administrator Password Solution** (**LAPS**) allows admins to periodically change the password for the local administrator password on domain-joined devices. This recipe shows how to implement and use it.

Getting ready

First, download LAPS from `http://aka.ms/LAPS`. Download the `*.msi` file that corresponds to the client operating system architecture(s) used in the organization. Most likely, this will be x64.

Make sure that all domain controllers in the environment run Windows Server 2003 with Service Pack 1 or a newer version of Windows Server.

If your organization places devices in the default Computers container, move the computer objects that you want to be part of LAPS from this container to an **Organizational Unit** (**OU**) dedicated to devices.

How to do it...

There are two sides to LAPS; implementing it and managing it.

Implementing LAPS

Implementing LAPS requires four steps:

Extending the schema

Follow these steps to extend the Active Directory schema with the LAPS extensions:

1. Sign into a domain controller or Windows Server-based management server that has .NET Framework 4.0 installed (or a newer version of the .NET Framework) with an account that is a member of the **Schema Administrators** group.
2. Double-click the MSI installer to install LAPS on the Windows Server.
3. Follow the instructions to install LAPS.
4. Open an elevated PowerShell window and type the following two lines of PowerShell to import the LAPS PowerShell module, and then to extend the Active Directory schema:

```
Import-Module AdmPwd.PS

Update-AdmPwdADSchema
```

Setting permissions

Next, we need to set permissions in Active Directory to enable devices to write to the new **mS-MCS-AdmPwd** and **mS-MCS-AdmPwdExpirationTime** attributes. Follow these steps:

1. Sign into the domain controller or Windows Server-based management server that has LAPS installed with an account that is a member of the **Domain Admins** group, or is delegated **Full Control** over the OU containing the devices in scope for LAPS (and its child OUs).

Open an elevated PowerShell window and type the following two lines of PowerShell to import the LAPS PowerShell module, and then to set the permissions on the OU with devices:

```
Import-Module AdmPwd.PS

Set-AdmPwdComputerSelfPermission -OrgUnit "OU ShortName"
```

Do not run the preceding PowerShell command on the entire directory, as it would include domain controllers, too. We don't want domain controllers to reset the built-in Administrator account every 30 days…

Creating the GPO to install the LAPS Client-side Extensions

As outlined in *Installing Applications* recipe, from `Chapter 9`, *Getting the Most Out of Group Policy*, follow these steps to install the LAPS **Client-side Extensions** (**CSEs**):

1. Log into a system with the **Group Policy Management** Console (`gpmc.msc`) installed with an account that is either a member of the Domain Admins group, or the current owner of an existing GPO, or delegated the **Edit Settings** or **Edit settings, delete and modify security permission** on an existing GPO.
2. Open the **Group Policy Management** Console (`gpmc.msc`).
3. In the left pane, navigate to the **Group Policy objects** node.
4. Locate the Group Policy Object that you want to use and select it, or right-click the **Group Policy Objects** node and select **New** from the menu.
5. Right-click the Group Policy object and select **Edit…** from the menu.
   The Group Policy Management Editor window appears.
6. In the main pane of the Group Policy Management Editor window, expand the **Computer Configuration** node, then **Policies** and the **Software Settings** node.
7. Right-click the **Software Installation** node and select **New** from the menu, and then **Package…**.
8. In the **Open** screen, browse to the network share that has the LAPS MSI package. Select the application and click **Open**.
9. In the **Deploy Software** popup screen, select **Assigned**.
10. Click **OK** to save the settings. The package will be listed with its version, its deployment state and source path.
11. In the left navigation window, expand the **Administrative Templates** node and then the **LAPS** node.
12. Double-click the **Enable local admin password management** setting and enable it.
13. Click **OK**.
14. Double-click the **Do not allow password expiration time longer than required by policy** setting and enable it.
15. Click **OK**.
16. Close the **Group Policy Management Editor** window.

Linking the GPO to OUs with devices

As outlined in *Linking a GPO to an OU* recipe from `Chapter 9`, *Getting the Most Out of Group Policy*, follow these steps to link the GPO to OUs with devices in scope for LAPS:

1. Log into a system with the **Group Policy Management** Console feature installed with an account that is either a member of the Domain Admins group, or the current owner of the Group Policy Object, and have the **Link GPOs** permission on the OU(s), Site(s), and/or Domain(s) where the Group Policy Object is to be linked, or is delegated the **Edit Settings,** or **Edit settings, delete and modify security permission** on the GPO, and have the **Link GPOs** permission on the Organizational Unit(s) where the Group Policy Object is to be linked.
2. Open the **Group Policy Management** Console (`gpmc.msc`).
3. In the left navigation pane, navigate to the OU where you want to link the LAPS GPO.
4. Right-click the OU and select **Link an existing GPO…** from the menu.
5. In the **Select GPO** window, select the LAPS GPO.
6. Click **OK** to link the GPO.

Repeat steps 4-6 to link the LAPS GPO to all OUs that require the LAPS GPO. Take **Block Inheritance** into account for OUs by linking the LAPS GPO specifically to include all devices in its scope.

Managing passwords

After LAPS is implemented, the passwords in LAPS' Active Directory attributes can be viewed and managed. The LAPS UI is the preferred tool to manage passwords:

Viewing an administrator password

Follow these steps to view an administrator password:

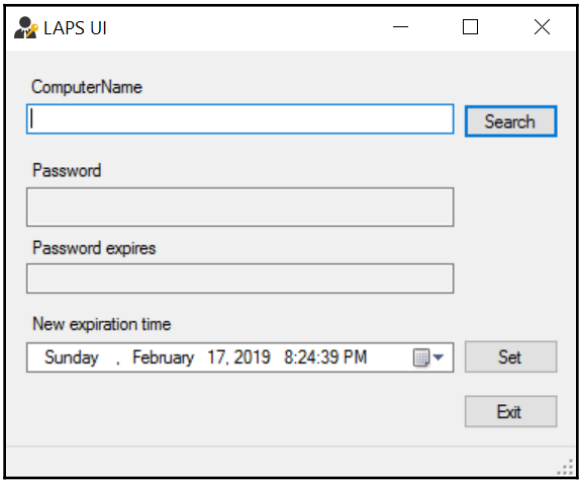
1. Sign into the domain controller or Windows Server-based management server that has LAPS installed with an account that is a member of the **Enterprise Admins** group or the **Domain Admins** group.
2. Start the LAPS UI from the Start Menu of the Start Screen.
3. In the LAPS UI Window, enter a device name in the **ComputerName** field or use the **Search** button to search for a device.

4. The password is shown in the **Password** field, together with the moment the **Password expires**.
5. Press **Exit** to close the LAPS UI.

Resetting an Administrator password

Follow these steps to reset an administrator password:

1. Sign into the domain controller or Windows Server-based management server that has LAPS installed with an account that is a member of the **Enterprise Admins** group or the **Domain Admins** group.
2. Start the LAPS UI from the Start Menu of the Start Screen:



3. In the **LAPS UI** window, enter a device name in the **ComputerName** field or use the **Search** button to search for a device.
4. Press the **Set** button to immediately expire the password and have the LAPS CSE on the device communicate with Active Directory to set a new password and reset the **Password expires** timeframe.
5. Press **Exit** to close the **LAPS UI**.

How it works...

The Local Administrator Password Solution has three components:

- The LAPS GPO instructs domain-joined devices in scope with settings.
- The LAPS Client-side Extensions set and exchange clear-text passwords with Active Directory, based on the GPO.
- The LAPS attributes for computer objects in Active Directory store passwords and expiration timeframes for LAPS. Devices have SELF permissions to write to these attributes.

Passwords for local Administrator accounts are stored in clear-text in the LAPS attributes. The **Filtered Attribute Set** (**FAS**) protects these attributes from view. By default, only members of the Enterprise Admins and Domain Admins group have access to these attributes. The FAS relies on domain controllers running Windows Server 2003 Service Pack 1, or newer versions of Windows Server, to work reliably.

See also

Refer to the *Installing applications* recipe, in `Chapter 9`, *Getting the Most Out of Group Policy*.

Refer to the *Linking a GPO to an OU* recipe, in `Chapter 9`, *Getting the Most Out of Group Policy*.

Managing deleted objects

This recipe shows how to manage deleted objects in an Active Directory environment with the Active Directory Recycle Bin enabled.

Getting ready

To manage deleted objects, sign into a domain controller, a Windows Server-based management server with the RSAT for Active Directory Domain Services installed, or a Windows installation with the Remote Server Administration Tools installed with an account that is a member of the **Domain Admins** group.

How to do it...

There are two ways to manage deleted objects:

- Using the Active Directory Administrative Center
- Using Windows PowerShell

Using the Active Directory Administrative Center

To manage deleted objects using the Active Directory Administrative Center, perform these steps:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. In the left navigation pane, switch to **Tree view**.
3. Navigate to the **Deleted Objects** container.
4. Perform one of these actions:
    - When the object to restore is an Organizational Unit, expand the **Deleted Objects** container and select the OU to restore. Right-click it and select **Restore** from the menu.
    - When the object to restore is a user object, computer object, or group, select it in the main pane. Right-click it and select **Restore** from the menu.

Using Windows PowerShell

To view the deleted objects for a domain, use the following lines of PowerShell on a system with the Active Directory Module for Windows PowerShell installed:

```
Import-Module ActiveDirectory

Get-ADObject -ldapFilter:"(msDS-LastKnownRDN=*)" -IncludeDeletedObjects
```

To restore a deleted object, use the following lines of PowerShell on a system with the Active Directory Windows PowerShell Module installed:

```
Import-Module ActiveDirectory

Get-ADObject -Filter {displayName -eq "DisplayNameOfTheObject"} -
IncludeDeletedObjects | Restore-ADObject
```

How it works...

When the Active Directory Recycle Bin is not enabled, objects that are deleted are tombstoned. This allows domain controllers to replicate the deletion. When the Active Directory Recycle Bin is enabled, deleted objects are not tombstoned immediately, but recycled first.

In this state, deleted objects are shown in the **Deleted Objects** container. Objects can be restored from this container to their original location or to a different container, including all their group memberships and other attributes.

There's more...

In the Active Directory Administrative Center, multiple objects can be selected in the main view by selecting them, or by using the *Shift* and *Ctrl* keys. Then, the selection can be used to restore multiple objects at once.

See also

For more information on the Active Directory Recycle Bin, look at *Enabling the Active Directory Recycle Bin* recipe from `Chapter 1`, *Optimizing Forests, Domains, and Trusts*.

Working with group Managed Service Accounts

This recipe shows how to work with **group Managed Service Accounts** (**gMSAs**).

Getting ready

To create gMSAs, the Active Directory domain needs to have at least one domain controller running Windows Server 2012 or a newer version of Windows Server.

gMSAs can only be used to run services on domain-joined hosts running Windows Server 2012, or newer versions of Windows Server, or Windows 8, or newer versions of Windows. For the automatic password and **Service Principal Name** (**SPN**) management, the domain needs to run at least the Windows Server 2008 R2 **Domain Functional Level** (**DFL**).

As gMSAs depend on the Key Distribution Service on domain controllers, prepare the forest by running the following lines of PowerShell on a system with the Active Directory module for Windows PowerShell:

```
Import-Module ActiveDirectory

Add-KdsRootKey -EffectiveImmediately
```

How to do it...

1. To create a gMSA, use the following lines of PowerShell on a system with the Active Directory Module for Windows PowerShell installed:

```
Import-Module ActiveDirectory

New-ADServiceAccount MSAName -DNSHostName
DomainController.domain.tld
-PrincipalsAllowedToRetrieveManagedPassword
"CN=AppServer1,CN=Computers,DC=LucernPub,DC=com"
```

2. To install the gMSA on an application server so that it can be assigned to run a service, application, or application pool, use the following line of PowerShell:

```
Install-ADServiceAccount -Identity MSAName
```

How it works...

Service accounts are notoriously hard for admins to get right. User objects are reused for this purpose and they are typically over-privileged, not secured enough, and admins rarely change the passwords for these accounts out of fear of breaking functionality.

Managed Service Accounts (**MSAs**) were introduced in Windows Server 2008 R2 to solve this problem. In Windows Server 2012, MSAs were superseded by gMSAs. Since then, when you create this type of object as an admin, you create a gMSA, by default.

The main difference between an MSA and a gMSA is that a gMSA can be used as a service account on more than one server, where an MSA is limited to one server.

gMSAs are `msDS-GroupManagedServiceAccount` objects. They are not based on user objects, but on computer objects. Just as with computer objects, they are prohibited from logging on interactively to systems, and automatically change their passwords every 30 days, by default. This makes them much more secure than service accounts based on user objects.

gMSAs use a password that is stored in the `msDS-ManagedPassword` attribute of the object. Only domain-joined servers that are listed in the `msDS-GroupMSAMembership` attribute are provided access to the attribute by the Key Distribution Service on domain controllers.

Although the line of PowerShell to create a gMSA specifies the Root Key to be effective immediately with the `-EffectiveImmediately` switch, you will actually have to wait 10 hours for it to become active. This ensures that there is ample time to replicate the information to other domain controllers.

There's more...

The interval that gMSAs use to change their passwords can be controlled using the `msDS-ManagedPasswordInterval` attribute. Even if password changes for computer objects has been turned off, gMSAs will continue to change their passwords. If the interval can be higher in your environment because strict regulations don't apply, set the attribute to a higher value (in days) when creating gMSAs.

Configuring the advanced security audit policy

This recipe shows how to configure the advanced security audit policy.

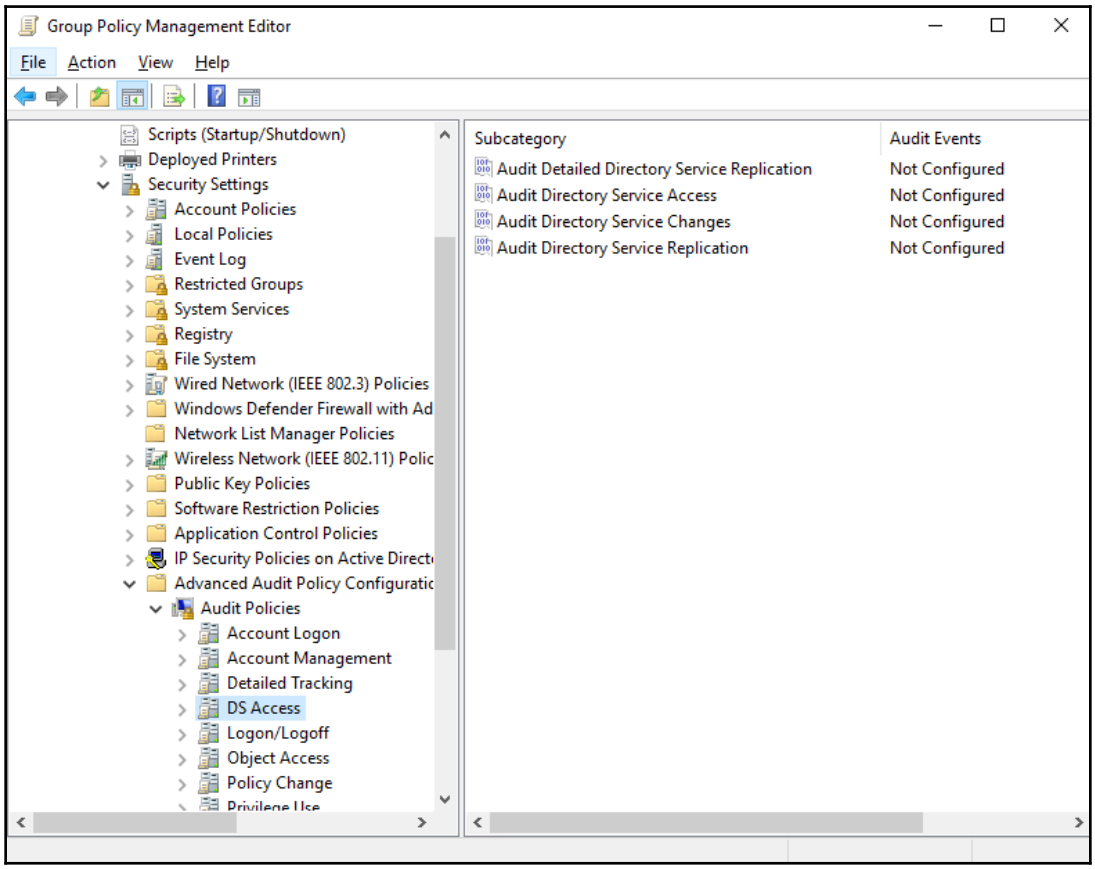
Getting ready

To configure the advanced security audit policy, sign into a domain controller with a user account that is a member of the Domain Admins group.

How to do it...

Follow these steps to configure the advanced security audit policy:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left pane, navigate to the **Domain Controllers** node for the domain in which you want to configure the advanced security audit policy.
3. Right-click the **Default Domain Controllers Policy** and select **Edit…** from the menu.
   The Group Policy Management Editor window opens.
4. In the left navigation window, expand the **Computer Configuration**, **Policies**, **Windows Settings**, **Security Settings**, **Advanced Audit Policy Configuration**, **Audit Policies**, and then **DS Access**:



5. Double-click the **Audit Directory Service Changes** setting.
   The **Audit Directory Service Changes Properties** window opens.
6. Check the **Configure the following auditing events:**. Then, check to audit **Success** and/or **Failure** audit events.
7. Click **OK** to save the settings and close the **Audit Directory Service Changes Properties** screen.
8. Close the **Group Policy Management Editor** window.

How it works...

Microsoft introduced the advanced security audit policy in Active Directory in Windows Server 2008 R2. This feature offers more granular auditing options in 10 categories:

- Account Logon
- Account Management
- Detailed Tracking
- DS Access
- Logon/Logoff
- Object Access
- Policy Change
- Privilege Use
- System
- Global Object Access Auditing

For each of these categories, several auditing options are available. When these are enabled, additional entries are added to Event Viewer with the source **Microsoft Windows security auditing**.

A recommended practice is to copy auditing events from event viewer logs on the domain controller to a centralized **Security Incident and Event Management** (**SIEM**) solution.

Resetting the KRBTGT secret

This recipe shows how to reset the password of the KRBTGT account.

Getting ready

To reset the password for the KRBTGT account, sign into a domain controller with a user account that is a member of the Domain Admins group.

How to do it...

Perform the following lines of PowerShell:

```
Import-Module ActiveDirectory

Set-ADAccountPassword -Identity (Get-ADUser krbtgt).DistinguishedName
-Reset -NewPassword (ConvertTo-SecureString "Rand0mCompl3xP@ssw0rd!"
-AsPlainText -Force)
```

How it works...

Each Active Directory domain in a multi-domain environment has its own KRBTGT account used by all fully-writable domain controllers. Each read-only domain controller has its own **KRBTGT_*** account.

The password hash for the KRBTGT account is used as the secret to encrypt all Kerberos tickets.

The password for KRBTGT is set during the creation of an Active Directory domain. Microsoft only automatically reset the secret on the KRBTGT account for Active Directory domains when the Domain Functional Level was upgraded to Windows Server 2008.

A malicious person would not just be able to read all Kerberos authentication traffic. When a malicious person wants to attain a foothold in an Active Directory, the most common way to do so is by forging tickets, as in a *golden ticket attack*. As ticket control in Active Directory is client-side, malicious people may (re)use forged Kerberos tickets for as long as 10 years.

The only way to lock out malicious persons using forged Kerberos tickets is to reset the password for KRBTGT with different values. However, the process for signing tickets is designed to handle these password changes without locking out legitimate use. Any ticket that has been signed before the password change will use the fallback method provided for the TGT lifetime. This lifetime, by default, is 7 days.

Replication convergence may take time throughout a large Active Directory environment. Therefore, the password for KRBTGT needs to be reset twice with an interval in between.

Microsoft's recommendation is to reset the password for KRBTGT twice per year.

There's more...

Microsoft offers a script on the TechNet Gallery to automate the processes of generating complex passwords, changing the password, and checking for proper replication:
`https://gallery.technet.microsoft.com/Reset-the-krbtgt-account-581a9e51`

Using SCW to secure domain controllers

This recipe shows how to secure domain controllers running older versions of Windows Server, using the Windows Server **Security Configuration Wizard** (**SCW**) and Group Policy.

Getting ready

To secure domain controllers using SCW, sign into a domain controller with a user account that is a member of the Domain Admins group.

The Security Configuration Wizard was removed from Windows Server 2016 and is not present in Windows Server versions since this version. Features are secured by default. This recipe applies to full installations of the following Windows Server versions:

- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2.

How to do it

Securing domain controllers using the Windows Server **Security Configuration Wizard** (**SCW**) and Group Policy consists of two steps:

1. Secure a representative domain controller using SCW.
2. Roll out the security settings to all domain controllers using Group Policy.

Secure a representative domain controller using SCW

Follow these steps to secure one of your domain controllers using SCW:

Follow these steps on a domain controller in a test environment to test the settings in the context of routine administration processes before rolling the settings out to all production domain controllers.

1. Open **Server Manager** (`servermanager.exe`).
2. From the **Tools** menu in the top grey pane, select **Security Configuration Wizard**.
3. On the **Welcome** screen, click **Next**.
4. On the **Configuration Action** screen, select **Create a new security policy** option.
5. Click **Next**.
6. On the **Select Server** screen, select the local server. Click **Next >**.
7. On the **Role-based Service Configuration** screen, click **Next >**.
8. On the **Select Server Roles** screen, click **Next >**.
9. On the **Select Server Features** screen, click **Next >**.
10. On the **Select Administration and Other Options** screen, click **Next >**.
11. On the **Select Additional Services** screen, click **Next >**.
12. On the **Handling Unspecified Services** screen, select the **Disable the service** option and click **Next >**.
13. On the **Confirm Service Changes** screen, click **Next >**.
14. On the **Network Security** screen, click **Next >**.
15. On the **Network Security Rules** screen, click **Next >**.
16. On the **Registry Settings** screen, click **Next >**.
17. On the **Require SMB Security Signatures** screen, check the properties of the Windows Server. These properties determine the SMB signing settings.
18. Click **Next >**.

19. On the **Require LDAP Signing** screen, select the **Windows 2000 Service Pack 3 or later** option.
20. Click **Next >**.
21. On the **Outbound Authentication Methods** screen, check the **Domain Accounts** option.
22. Click **Next >**.
23. On the **Outbound Authentication using Domain Accounts** screen, select both options to require NTLM version 2.
24. Click **Next >**.
25. On the **Inbound Authentication Methods** screen, deselect all options unless your environment contains devices running Windows XP.
26. Click **Next >**.
27. On the **Registry Settings Summary** screen, click **Next >**.
28. On the **Auditing Policy** screen, click **Next >**.
29. On the **System Auditing Policy** screen, select the **Audit successful and unsuccessful activities**.
30. Click **Next >**.
31. On the **Audit Policy Summary** screen, click **Next >**.
32. On the **Save Security Policy** screen, click **Save**.
33. Save the file with its `.xml` extension.
34. On the **Apply Security Policy** screen, click **Next >**.
35. On the **Application Complete** screen, click **Finish**.

Roll-out the security settings to all domain controllers using Group Policy

Run the following command line on an elevated Command Prompt (`cmd.exe`) to convert the settings file into a Group Policy Object:

```
scwcmd.exe transform /p:"C:\Windows\security\msscw\Policies\test.xml"
/g:"Domain Controller Security Settings"
```

The preceding command creates the Group Policy with the name **Domain Controller Security Settings**.

Next, link the new Group Policy object to the **Domain Controllers** Organizational Unit (OU), using the following steps:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left navigation pane, expand the **Forest** node.
3. Expand the **Domains** node, and then navigate to the domain where you want to link the GPO.
4. Expand the domain name.
5. Navigate to the **Domain Controllers** OU.
6. Right-click the OU and select **Link an existing GPO…** from the menu.
7. In the **Select GPO** window, select the GPO you want to link from the list of available **Group Policy objects:**.
8. Click **OK** to link the GPO.

How it works...

The Windows Server Security Configuration Wizard guides admins through the following settings:

- Permitted Server Roles and Server Features
- Permitted remote access
- Permitted services
- SMB and LDAP settings
- Auditing settings

This way, the wizard allows for straightforward management of these settings.

While the settings can be applied on a per-domain controller basis, a Group Policy can be applied with the settings, after the file is converted to a GPO using `scwcmd.exe`. After that, the GPO can be linked to the **Domain Controllers** OU to apply the settings to all domain controllers.

Leveraging the Protected Users group

This recipe shows how the Protected Users group can be used to protect privileged and sensitive accounts.

Getting ready

To use the Protected Users group, make sure the domain runs the Windows Server 2012 R2 DFL, or a newer version of the level. Also, be aware that the protections offered by the Protected Users group only apply when accounts that are members of the group are used on devices running Windows 8.1 or newer, Server 2012 R2 or newer.

To manage the Protected Users group, sign into a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed. Sign in with an account that is a member of the Domain Admins group, the Account Operators group or with an account that is delegated to manage groups in the domain or in the scope of the OU.

How to do it...

There are three ways to manage group memberships in Active Directory:

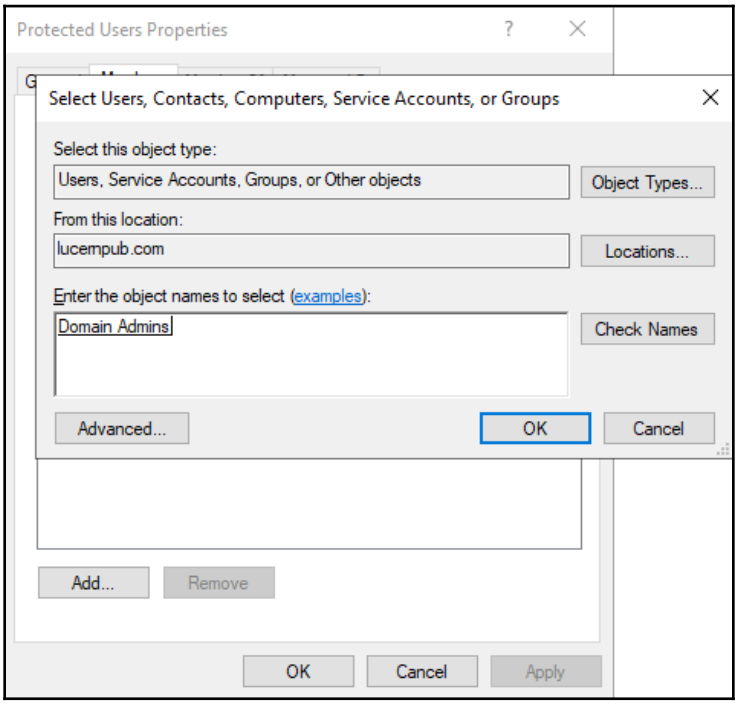
- Using Active Directory Users and Computers
- Using the Active Directory Administrative Center
- Using Windows PowerShell

Using Active Directory Users and Computers

Follow these steps to add user accounts to the **Protected Users** group using Active Directory Users and Computers:

1. Open **Active Directory Users and Computers** (`dsa.msc`).
2. From the **Action** menu, select **Find…**. In the **Name** field, type the **Protected Users**, and press *Enter*. From the list of **Search results:** select the group.
3. Right-click the group and select **Properties** from the menu.
   The **Protected Users Properties** window will now appear.
4. Navigate to the **Members** tab.
5. Click **Add…** to add users, contacts, computers, service accounts or groups to the group.

6. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** window, type the name of the user account(s) you want to add to the group, or click the **Advanced** button to search for the user account(s).
7. Click **Check Names**:



8. Click **OK** to add the user, contact, computer, service account, or group to the **Protected Users** group.
9. Click **OK** to close the **Protected Users Properties** window and save the changes.

Using the Active Directory Administrative Center

Follow these steps to add user accounts to the **Protected Users** group using the Active Directory Administrative Center:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. From the main pane menu, under **Global Search**, type the name of the group, and press *Enter*.
3. From the list of **Global Search** results, select the group.

4. Right-click the group and select **Properties** from the list.
5. In the left navigation pane, click **Members**.
6. Click **Add…** to add users, contacts, computers, service accounts, or groups to the group.
7. In the **Select Users, Contacts, Computers, Service Accounts, or Groups** window, type the name of the user account(s) you want to add to the group, or click the **Advanced** button to search for the user account(s).
8. Click **Check Names**.
9. Click **OK** to add the user to the group.
10. Click **OK** to close the **Group Properties** window and save the changes.

Using Windows PowerShell

Use the following lines of PowerShell to add a user to the **Protected Users** group in Active Directory on a system with the Active Directory Module for Windows PowerShell installed:

```
Import-Module ActiveDirectory

Add-ADGroupMember -Identity "CN=Protected
Users,CN=Users,DC=lucernpub,DC=com" -Members "User"
```

How it works...

The **Protected Users** group is a new feature in Active Directory in Windows Server 2016. Accounts that are members of the group lose the ability to do the following:

- Use cached logons.
- Use outdated authentication protocols, such as NTLM, Digest Authentication, and CredSSP.
- Use weak encryption algorithms, such as DES and RC4, for Kerberos pre-authentication.
- Be delegated as part of both **Kerberos Constrained Delegation** (**KCD**) and Kerberos unconstrained delegation.
- Use and renew their Kerberos **Ticket Granting Ticket** (**TGT**) for longer than 240 minutes, compared to the default 10-hour validity and 7-day renewal periods.

The preceding protections are non-configurable.

The Protected Users group is ideal for privileged and sensitive user accounts, such as members of the Domain Users group. Don't add service accounts, MSAs, gMSAs, or computer objects as members to the Protected Users group, as it may break their functionality.

Putting authentication policies and authentication policy silos to good use

This recipe shows how to use Authentication Policies and Authentication Policy Silos.

Getting ready

To use Authentication Policies and Authentication Policy Silos, make sure the domain runs the Windows Server 2012 R2 DFL, or a newer version of the level. Also, be aware that the protections offered by the Authentication Policies and Authentication Policy Silos only apply when accounts that are members of the group are used on devices running Windows 8.1 or newer versions of Windows or Windows Server 2012 R2 or newer versions of Windows Server.

To manage Authentication Policies and Authentication Policy Silos, sign into a domain controller or a member server and/or device with the RSAT for Active Directory Domain Services installed. Sign in with an account that is a member of the **Domain Admins** group.

How to do it...

Putting Authentication Policies and Authentication Policy Silos to good use consists of five steps:

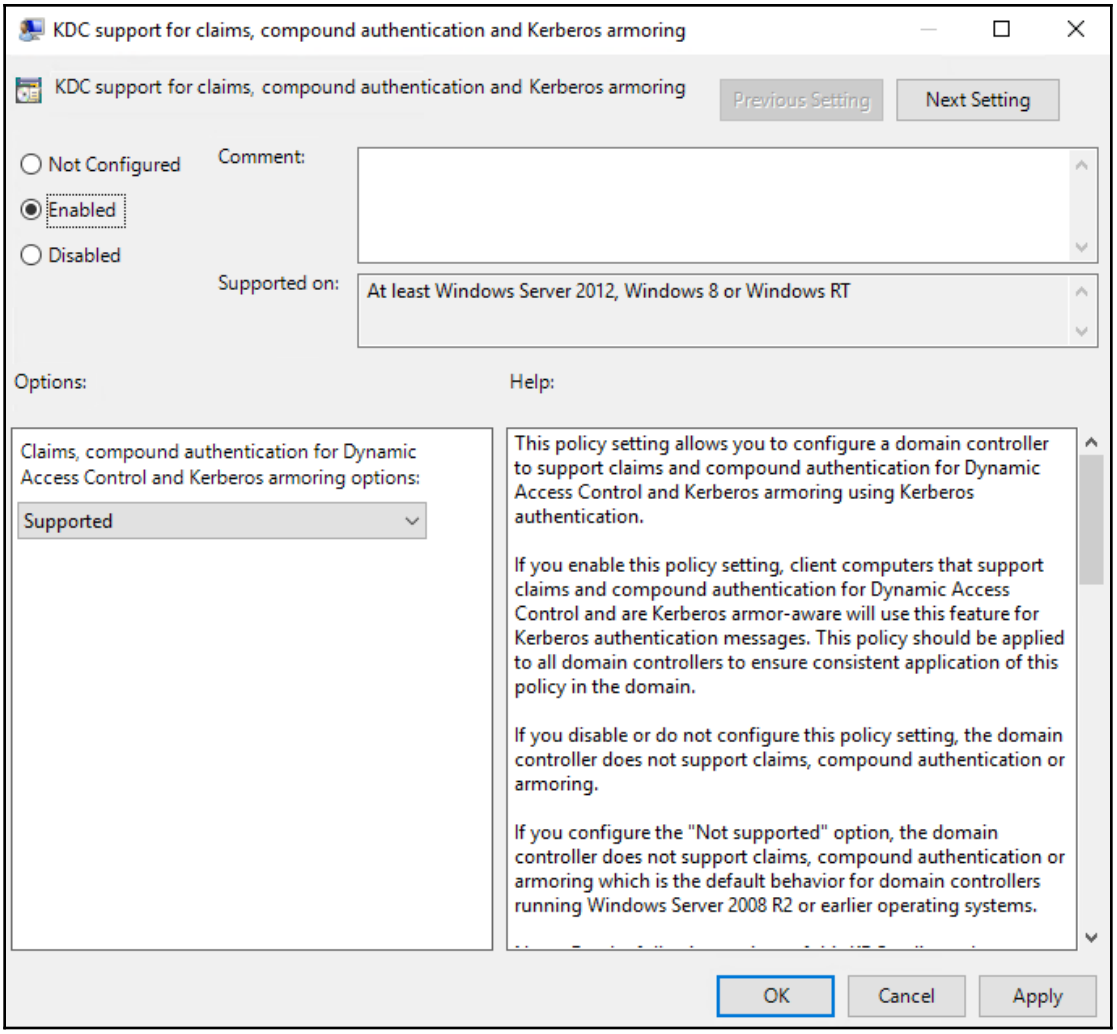
- Enable domain controller support for claims.
- Enable compound claims on devices in scope for an authentication policy.
- Create an Authentication Policy.
- Create an Authentication Policy Silo.
- Assign the Authentication Policy Silo.

Enable domain controller support for claims

Follow these steps to enable domain controller support for claims:

1. Open the **Group Policy Management** Console (`gpmc.msc`).
2. In the left navigation pane, expand the **Forest** node.
3. Expand the **Domains** node, and then navigate to the domain where you want to enable compound claims on devices.
4. Expand the domain name.
5. Right-click the **Group Policy Objects** node and select **New** from the menu.
6. In the **New GPO** popup window, enter the name of the Group Policy Object. Make sure you don't select a Starter GPO.
7. Click **OK** to create the GPO.
8. Expand the **Group Policy Objects** node.
9. Locate the Group Policy Object that you want to manage.
10. Select the Group Policy Object.
11. In the left navigation pane, right-click the GPO and select **Edit** from the menu. The Group Policy Management Editor (`gpedit.msc`) appears.
12. In the Group Policy Management Editor window, expand **Computer Configuration**, then **Policies, Windows Administrative Settings**, and **System**.
13. Select **KDC**.
14. In the main pane, right-click the **KDC support for claims, compound authentication, and Kerberos armoring** setting and select **Properties** from the menu.

15. Click **Enabled**, as shown in the following screenshot:

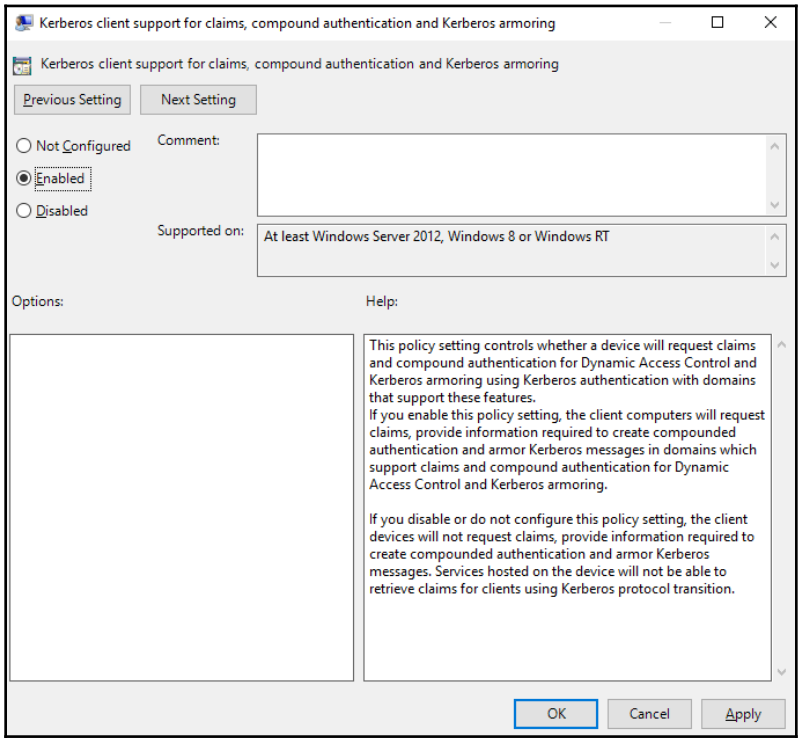


16. Click **OK** to close the Group Policy Management Editor window.
17. Link this Group Policy to the **Domain Controllers** Organizational Unit (OU) by right-clicking the OU in the left pane and selecting **Link an existing OU** from the menu.
18. In the **Select GPO** window, select the GPO you want to link from the list of available **Group Policy objects:**.
19. Click **OK** to link the GPO.

Enable compound claims on devices in scope for an authentication policy

Follow these steps to enable compound claims on devices in scope for an authentication policy:

1. Create another Group Policy by repeating previous steps 5-7, and start editing it by repeating steps 8-11 from the previous list of steps.
2. In the Group Policy Management Editor window, expand **Computer Configuration**, then **Policies, Windows Administrative Settings**, and **System**.
3. Select **Kerberos**.
4. In the main pane, right-click the **Kerberos client support for claims, compound authentication, and Kerberos armoring** setting and select **Properties** from the menu.
5. Click **Enabled:**



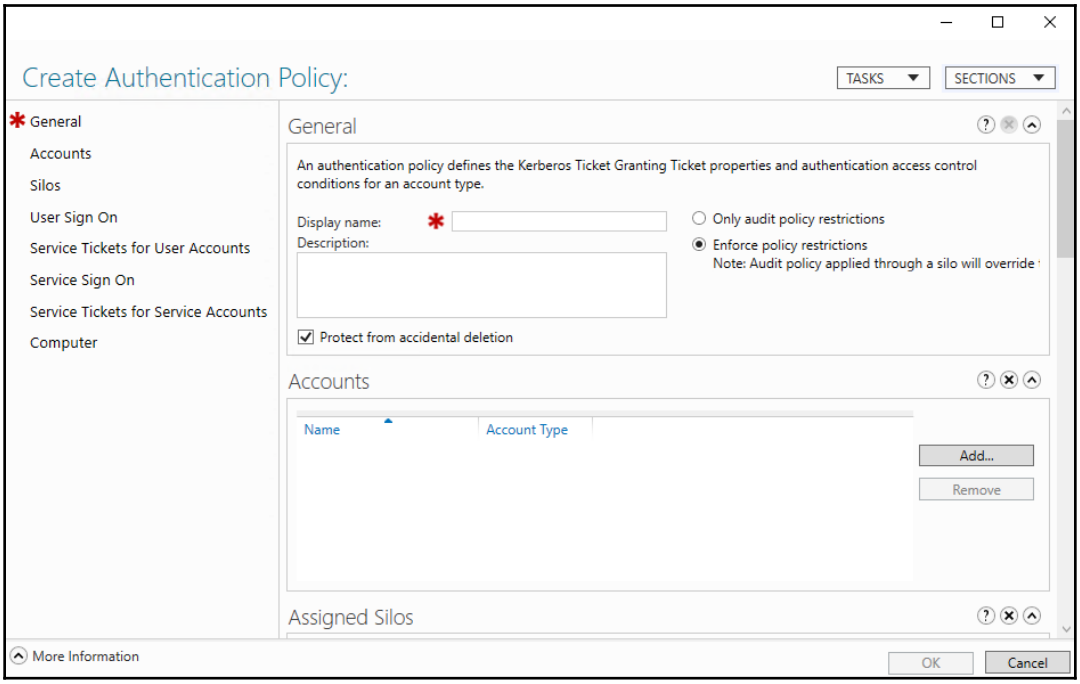
6. Click **OK** to close the Group Policy Management Editor window.
7. Link the second Group Policy to the Organizational Unit(s) with devices in scope, or to the domain by repeating steps 17-19 from the previous list of steps.

To restrict administrators from using certain devices, do not apply the preceding Group Policy object to the Organizational Unit(s) containing these devices.

Create an Authentication Policy

Follow these steps to create an Authentication Policy:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. In the left navigation window, switch to **Tree view**.
3. In the left navigation pane, click **Authentication**.
4. In the main pane select the **Authentication Policies** node.
5. In the **Tasks** pane to the right, click **New** under **Authentication Policies**.
6. Select **Authentication Policy** from the menu.
   The **Create Authentication Policy** screen appears:

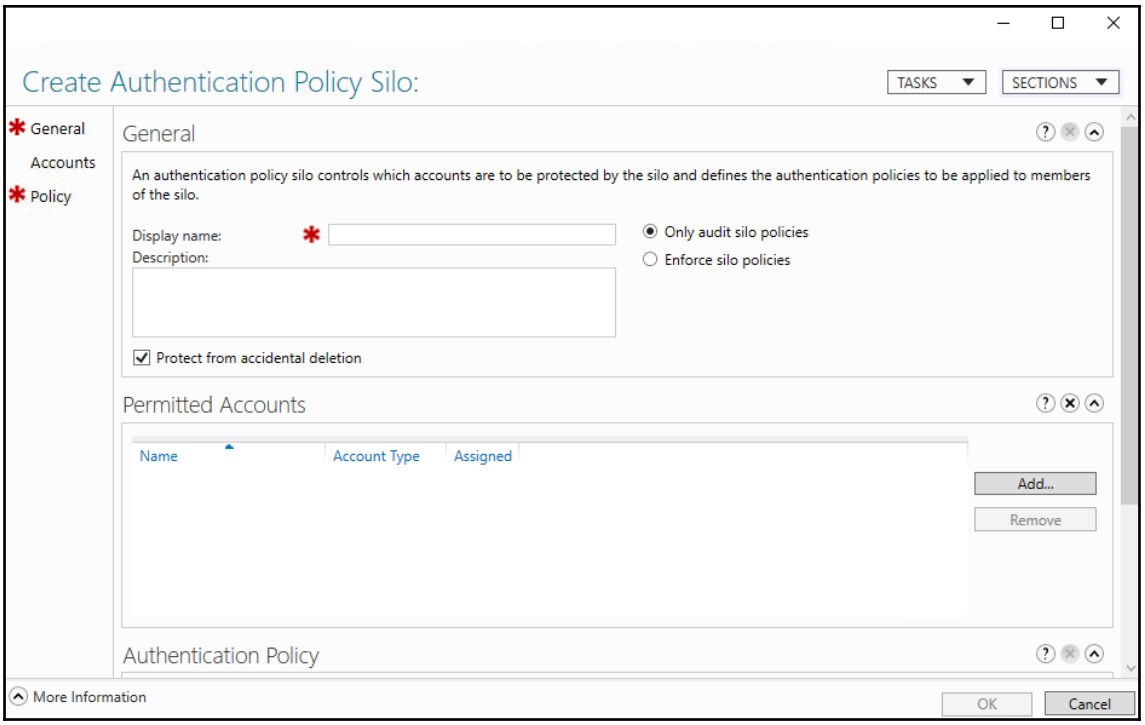


7. Provide a name for the authentication policy in the **Display name:** field.
8. Optionally, you can also provide a **Description:**.
9. In the left navigation pane, click **User Sign On**.
10. Select the settings you want to configure, such as the **Specify a Ticket Granting Ticket lifetime for user accounts.** option. Then, select a value between 45 and 2147483647 (2^{31}-1)for the **Ticket-Granting-Ticket Lifetime (minutes):** to limit the lifetime for the TGT for objects in scope for this Authentication Policy.
11. Click **OK** to close the **Create Authentication Policy** window and save its settings.

Create an Authentication Policy Silo

Follow these steps to create an Authentication Policy Silo:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. In the left navigation pane, click **Authentication**.
3. In the main pane select the **Authentication Policy Silos** node.
4. In the **Tasks** pane to the right, click **New** under **Authentication Policy Silos**. The **Create Authentication Policy Silo** window appears:



5. Provide a name for the authentication policy silo in the **Display name:** field.
6. Optionally, you can also provide a **Description:**.
7. As the behavior for this authentication policy silo, select **Enforce silo policies**.
8. In the left navigation pane, select **Accounts**.
9. In the list of **Permitted Accounts:**, add the accounts for which you want the policy silo to apply. Use the **Add…** button to add accounts.
10. In the left navigation pane, select **Policy**.
11. Select the **Use a single policy for all principals that belong to this authentication policy silo** option.
12. In the **authentication policy that applies to all accounts in this silo:** field, select the Authentication Policy you created in the previous steps from the drop-down list.
13. Click **OK** to close the **Create Authentication Policy Silo** window and save its settings.

Assign the Authentication Policy Silo

Follow these steps to assign the Authentication Policy Silo:

1. Open the **Active Directory Administrative Center** (`dsac.exe`).
2. In the left navigation pane, click **Authentication**.
3. Under the **Authentication Policy Silos** node, select the Authentication Policy Silo you created earlier.
4. Right-click it and select **Properties** from the menu.
5. In the left pane click **Permitted Accounts**.
6. Double-click the first item in the **Permitted Accounts:** list to open its properties. The properties window for the account will now open.
7. In the left pane, click **Silo**.
8. In the **Authentication Policy Silo** section, select the **Assign Authentication Policy Silo** option. Use the drop-down list for **Authentication Policy Silo:** to select the Authentication Policy silo created earlier.
9. Click **OK** to save the Authentication Policy Silo DN in the object's **msDS-AssignedAuthNPolicySilo** attribute.
10. Repeat steps 5-8 for all other accounts in the list of **Permitted Accounts:**.
11. Click **OK** to close the **Create Authentication Policy Silo** window.

How it works...

Using Authentication Policies and Authentication Policy Silos is a perfect way to set the scene for Microsoft's *Privileged Access Workstation* strategy to prevent people from signing in with their privileged account to devices other than their secure ones. This way, lateral movement toward admin (cached) credentials is hugely limited, benefiting the overall security posture of the organization.

Authentication Policies define policies, but do not assign these policies to accounts. Authentication Policy Silos assign policies to accounts. An Authentication Policy can be assigned through many Authentication Policy Silos, as the policies need to be the same, but for different audiences of accounts.

When a person tries to log on with an account that is governed by an Authentication Policy on a device that does not support claims, compound authentication, and Kerberos armoring, signing into the device will be prohibited with the following error:

```
Your account is configured to prevent you from using this PC. Please try
another PC.
```

Configuring Extranet Smart Lock-out

This recipe shows how to configure Extranet Smart Lock-out on an **Active Directory Federation Services** (**AD FS**) farm running Windows Server 2016, or a newer version of Windows Server.

Getting ready

When using AD FS on Windows Server, make sure at least the June 2018 Cumulative update for Windows Server 2016 (KB4284880 (`https://support.microsoft.com/en-us/help/4284880/windows-10-update-kb4284880`), OS build 14393.2312), is installed on all AD FS servers in the AD FS farm.

Sign in with an account that is an AD FS Administrator. By default, members of the **Domain Admins** group have the required permissions. Sign into the primary AD FS server when the AD FS farm is using the **Windows Internal Database** (**WID**) as its replication model, or any AD FS server when the AD FS farm leverages SQL Server as its configuration database.

How to do it...

To enable Extranet Smart Account Lock-out for an AD FS farm running SQL Server, run the following lines of PowerShell to update the AD FS Artifact Store:

These three lines of PowerShell do not need to be run on AD FS farms using Windows Internal Database.

```
$cred = Get-Credential

Import-Module ADFS

Update-AdfsArtifactDatabasePermission -Credential $cred
```

To enable Extranet Smart Account Lock-out for an AD FS farm, run the following lines of PowerShell:

```
Import-Module ADFS

Set-AdfsProperties -ExtranetLockoutThreshold 10
```

```
Set-AdfsProperties -ExtranetObservationWindow (New-Timespan -minutes 5)

Set-AdfsProperties -EnableExtranetLockout $true

Set-AdfsProperties -ExtranetLockoutMode AdfsSmartLockoutEnforce

Restart-Service adfssrv
```

How it works...

When adding AD FS to an environment running Active Directory, the last thing you want is for AD FS to have a negative impact on the overall information security of the environment.

AD FS adds Extranet Lockout to the Active Directory (fine-grained) password and account lock-out policies to prevent malicious persons from locking out accounts with incorrect password attempts from the internet. However, when the AD FS Extranet Lockout threshold is reached, the account cannot be used for the period of the AD FS Extranet Lockout duration to authentication to AD FS-integrated resources (while authenticating to other resources will work without a hitch).

To prevent this latter scenario, enable Extranet Smart Lockout. With this feature enabled, IP addresses for successful authentications by users are logged as familiar IPs in the **AccountActivity** table for the account. For this IP address, the regular AD FS threshold still applies and each legitimate user may still lock himself or herself out, like always.

The difference occurs, when authentications start to fail for the account from unfamiliar IP addresses. The failed authentication count for that IP address is incremented, and when the lock-out threshold is reached, authentication attempts from that specific unfamiliar IP address are locked out. However, legitimate users do not experience any lock-outs from their familiar IP addresses.

Microsoft recommends using more strict lock-out settings for AD FS (Smart Account Lock-out) than for Active Directory (fine-grained) password and account lock-out policies to make sure AD FS authentication attempts to do not lock out accounts in Active Directory itself.