Active Administrator

E-BOOK

Windows Server 2008 R2: Top Tips & Tricks

Learn best practices for failover clusters, get Active Directory management tips and find out how to master Windows Server 2008 backup basics in this free guide!

Windows Server 2008 R2: Networking in Failover Clusters 1

Managing Active Directory Password Policies..... 4

Backup Basics in Windows Server 2008 R2 9

Redmond

Redmondmag.com

I got this many AD tasks done today.

Active Administrator 6

What about You?

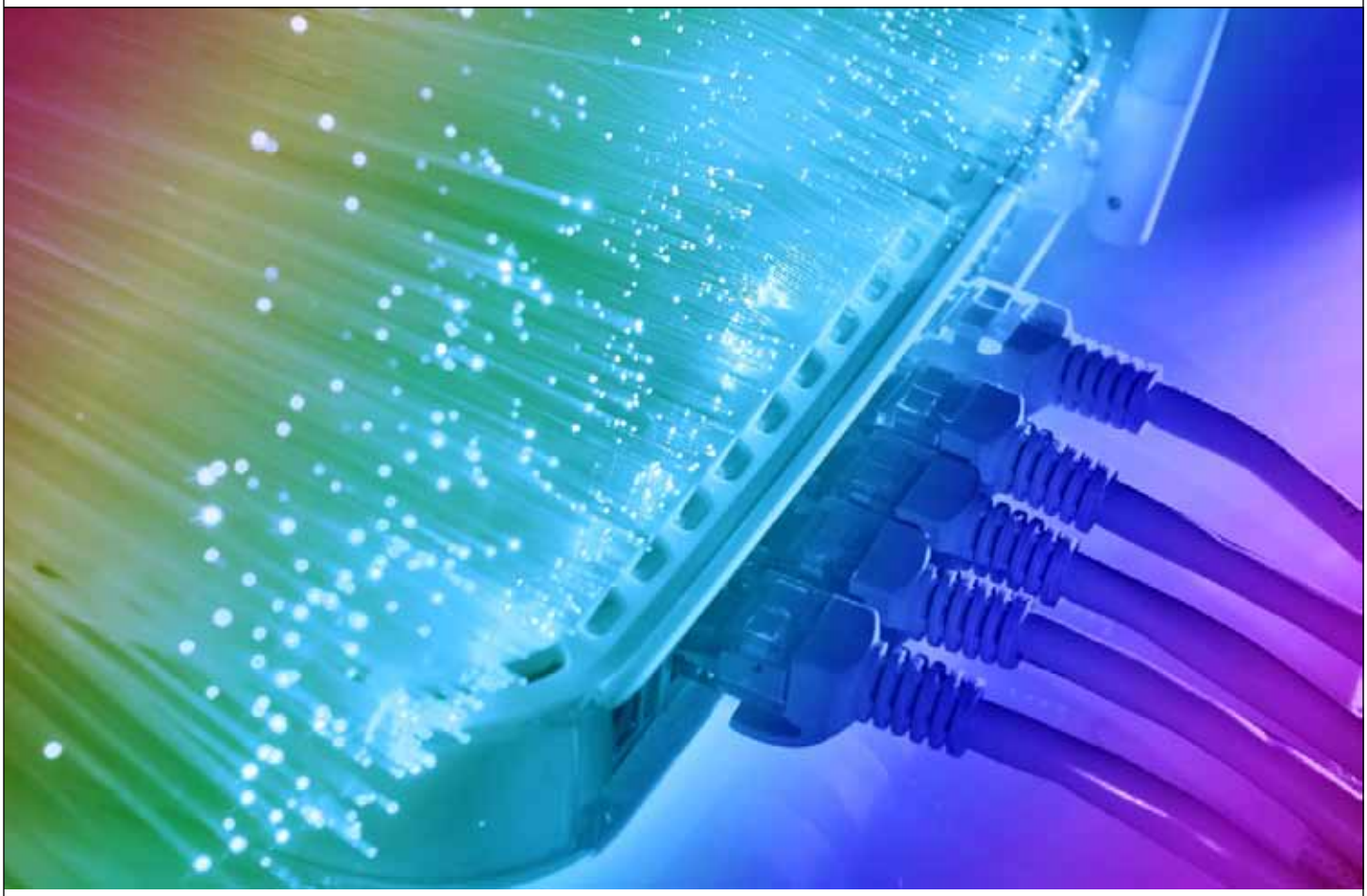
Download a FREE copy of Active Administrator at:

WWW.SCRIPTLOGIC.COM/AA6

SCRIPTLOGIC

A QUEST SOFTWARE® COMPANY

BY JOHN MARLIN



Windows Server 2008 R2: Networking in Failover Clusters

When failure isn't an option, configuring failover clusters in Windows Server can help ensure high availability.

The networking model in Windows Server 2008 and Windows Server 2008 R2 Failover Clustering provides more robust and reliable communication among all cluster nodes, which greatly improves the efficiency and dependability of failover clustering. There are also several new features, including:

- More reliable communication using TCP and UDP unicast
- Support for IPv6
- Support for locating cluster nodes on separate, routed subnets
- More fine-grained control over network failure detection

You'll need to use network hardware marked as "Certified for Windows Server 2008." Any other component of your failover cluster solution must also be similarly certified. If you use iSCSI, your network adapters need to be dedicated for either network communication or iSCSI—not both.

When designing the network infrastructure to connect your cluster nodes, it's essential to avoid single points of failure. There are many ways you can accomplish this. You can connect your cluster nodes with multiple, distinct networks. You could also connect your cluster nodes with one network built with teamed network adapters, redundant switches, redundant routers or similar hardware

that remove single points of failure. These architectural requirements differ from server clusters in Windows Server 2003, which required two distinct networks.

Cluster Communications

Windows Server 2008 Failover Clustering now uses a virtual network adapter called Microsoft Failover Cluster Virtual Adapter to communicate between nodes in the cluster. You'll also see this in Device Manager under Network Adapters (select Show hidden devices). You'll also see it when issuing the command IPCONFIG /ALL. This network adapter handles all packet routing over the proper networks for communication, joins and so on.

This adapter will have an APIPA address defined in the address block 169.254.0.0/16. In IPv6, they're assigned with the fe80::/10 prefix. In some environments, when adapters have an APIPA address, those adapters are disabled. If you disable the Cluster Virtual Adapter, you'll disable communication between the nodes.

The goal is to sustain TCP/IP connectivity between two or more systems, despite the failure of any component in the network path. So there has to be an alternate physical path. In other words, a network component failure (whether it's an NIC, router, switch or hub) shouldn't cause a communication breakdown.

Communication should continue in a timely manner. There might be a slower response, but communication will persist as long as there's an alternate physical route or link. This really comes into play when you talk about having nodes in separate sites or subnets.

Another change in Windows Server 2008 Failover Clustering is the cluster heartbeat mechanism. While it still uses port 3343, it has transitioned from a UDP broadcast health-checking mechanism to a UDP unicast communication. It's similar to a ping in that it uses a Request-Reply process, but it includes more sophisticated features such as security and sequence numbering.

The default behavior has also changed in terms of how many replies are needed before the node is considered unreachable, initiating a Regroup to obtain a new view of the cluster membership. The cluster heartbeats let all nodes know which is up and down. As a default, the settings for this are controlled by:

- SameSubnetDelay: heartbeat frequency for nodes in the same subnet
- SameSubnetThreshold: threshold of the delays for nodes in the same subnet
- CrossSubnetDelay: heartbeat frequency for nodes in different subnets
- CrossSubnetThreshold: threshold of the delays for nodes in different subnets.

These settings, and the method for changing them, are defined on the "Configure Heartbeat and DNS Settings in a Multi-Site Failover Cluster" TechNet Library page. There's a "heartbeat" sent across with a sequence number, say from Node1 to Node2. Node2 responds with the same sequence number. Node1 again sends the same sequence number to Node2, and Node2 returns it one last time.

Node1 would then determine this heartbeat sequence complete and start the process again with another sequence number. If any of the heartbeat sequences are dropped or not received, it's considered a "missed" heartbeat. By default, if any five of these sequences are missed, the node is considered down or inactive.

You can change these settings to increase the delay or threshold, but you can only work around any network problems. If there are any network latency issues, this could get around it, but it won't fix the problem. So keep in mind that making changes to the delay or threshold settings isn't considered a troubleshooting technique.

The heartbeats, by default, are going to use IPv6, as it's a faster protocol than IPv4. If IPv6 has been disabled, it will instead use IPv4. A failover cluster will not mix and match IPv6 and IPv4. It will use one or the other, but not both at the same time.

Cluster Creation

When you create a cluster in Windows Server 2008 and Windows Server 2008 R2, the cluster-networking driver detects and creates the networks based on whether a default gateway is on the adapter. If it detects a default gateway, that network is set to allow clients to connect and use it for cluster communications.

This lets cluster IP addresses and client access points (network names) use the network. It also gives it a metric value starting at 10,000. If a network doesn't have a default gateway, it will be given a metric value starting at 1,000. Then it will only be selected for Cluster Communications.

NAME	METRIC
iSCSI Network	1000
Backup Network	1100
Host Access	10000 <<– has default gateway
CSV Network	1200
Live Migration Network	1300

Each network it detects increases the metric increment by 100.

One thing about the way it works now is that there's no more concept of a "public" and "private" network. Therefore, the old "Recommended Private 'Heartbeat' Configuration on a Cluster Server" article for Windows Server 2003 clustering is invalid. Cluster communications are still going to go through all networks.

In previous versions, you defined which network you wanted to use for cluster communications. As long as that network was available, the cluster would use only that network. Windows Server 2008 and Windows Server 2008 R2 use all networks. If there's an issue with one network, it will automatically switch between networks.

There's an internal metric the Cluster Network driver uses. It doesn't use the general TCPIP metric value. You can see the metric values with the following Windows PowerShell command:

```
Get-ClusterNetwork | FT Name,
Metric
```

The metric values really come into play when talking about having a cluster with highly available virtual machines (VMs) and using Cluster Shared Volumes.

For example, say you want to run this command with these networks configured in the chart on page 2.

When using Cluster Shared Volumes, it will use the lowest metric value network for any CSV traffic or redirected mode access. When using the live migration feature of failover clustering, it will use the second-lowest metric.

In the example, CSV traffic will go over the iSCSI Network and live migrations will go over the network used for backups. When taking a backup of the VMs, the Cluster Shared Volumes will go into a redirected mode access. This is going to interfere with the ISCSI connections and could lead to disk failures. A data backup on the local drive of Node1 and a Live Migration would interfere with each other.

NAME	METRIC
iSCSI Network	1100
Backup Network	1000
Host Access	10000 <<– has default gateway
CSV Network	800
Live Migration Network	900

You need to reconfigure the networks to get everything you need. For the Live Migration network, you can change this by bringing up the properties of one of the VMs. On the Live Migration tab, change it to the LM Cluster network. For this, you only need to do it on a single VM because this is a global setting for all VMs.

For the CSV network, you can only affect this change through Windows PowerShell. To order the networks from Low to High, use the following commands:

```
Get-ClusterNetwork "CSV Cluster" |
%{$_.Metric=800}
Get-ClusterNetwork "LM Cluster" |
%{$_.Metric=900}
Get-ClusterNetwork "Backup
Network" | %{$_.Metric=1000}
Get-ClusterNetwork "ISCSI Storage
Network" | %{$_.Metric=1100}
```

Running the command to see the metrics will now show in the chart above.

The CSV cluster network is set for metric 800. Adding any new network that doesn't have a default gateway would be higher. Now with properly configured metrics, you can take backups or live migrate VMs without any conflicts on the networks.

The last thing to mention is cluster validation. You can run some network validation tests to determine connectivity issues, network configurations and so on. You can run these tests at any time without affecting production.

The cluster validation tests include:

- Cluster Configuration
- List Cluster Network Information
- Network
- List Network Binding Order
- Validate Cluster Network Configuration
- Validate IP Configuration
- Validate Multiple Subnet Properties
- Validate Network Communication

You can find the details of the Cluster Validation tests on the "Understanding Cluster Validation Tests" TechNet Library page. This will show you exactly what the tests look for and what each test does. R

John Marlin is a senior support escalation engineer in the Commercial Technical Support Group. He has been with Microsoft for more than 19 years, with the last 14 years focusing on Cluster Servers.

BY DEREK MELBER

Managing Active Directory Password Policies

So, you think you know how password policies work in Active Directory? Well, you might … or you might not. Find out how to manage Active Directory password policies in Windows Server 2008 and Windows Server 2008 R2.

Some things in life, like death and taxes, are guaranteed. There are other things in life that you think are guaranteed, or at least you think you know how they work—such as Active Directory password policies. Then, there are things that you want to work, and when they come along, you feel you know how they work before you even look at them—such as fine-grained password policies (FGPPs). I'm not going to discuss death and taxes, but I am going to clarify the misconceptions surrounding Active Directory password policies and FGPPs.

With the technology of password policies having existed for more than 10 years now, you'd think this topic would be infinitely clear. However, based on my exposure to network administrators who are still confused about how Active Directory password policies work, that's not the case.

Basic Facts

These basic facts have been the same in Active Directory domains since Windows 2000, which was released 11 years ago:

- The Default Domain Policy defines the password policies by default for every user in Active Directory and every user located in the local Security Account Manager (SAM) on every server and desktop that joins Active Directory.
- There can be only one password policy for domain users in a Windows 2000 and Windows Server 2003 Active Directory domain.
- It's not possible to configure the password policy for an organizational unit (OU) of users to be different than that of other users in the domain or in a different OU.
- The password policy settings can't be extended to include additional settings without using a third-party

tool or developing a custom password policy solution.

▶ It's not possible to configure a password policy for the root domain and have it "funnel" down to the other domains in the Active Directory tree.

I still see administrators and organizations try to explain that they have an environment different than what is possible. With that said, I'd encourage all of the admins and organizations that think they have a different configuration for passwords to "test" what they believe. Unless you have a third-party product in place or have Windows Server 2008 native mode domains, you can't have anything but what I detailed here.

Possible Settings in the Password Policy

When you edit a standard Group Policy Object (GPO) from the Group Policy Management Console (GPMC), you'll find the same options for the Account Policy. To find the password policy settings, which are under the Account Policy, open up the following path of policy folders: Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies. Once there, you'll find three policy folders: Password Policy, Account Lockout Policy and Kerberos Policy.

For each of these folders and the settings contained within them, there's a default in Windows Server 2003, Windows Server 2008 and Windows Server 2008 R2 freshly installed domains. The default settings are as shown in Table 1.

Policy Setting	Default Value
Enforce password history	24 days
Maximum password age	42 days
Minimum password age	1 day
Minimum password length	7
Password must meet complexity requirements	Enabled
Store passwords using reversible encryption	Disabled
Account lockout duration	Not defined
Account lockout threshold	0
Reset account lockout counter after	Not defined
Enforce user logon restrictions	Enabled
Maximum lifetime for service ticket	600 minutes
Maximum lifetime for user ticket	10 days
Maximum lifetime for user ticket renewal	7 hours
Maximum tolerance for computer clock synchronization	5 minutes

Table 1. *Account Policy settings default values.*

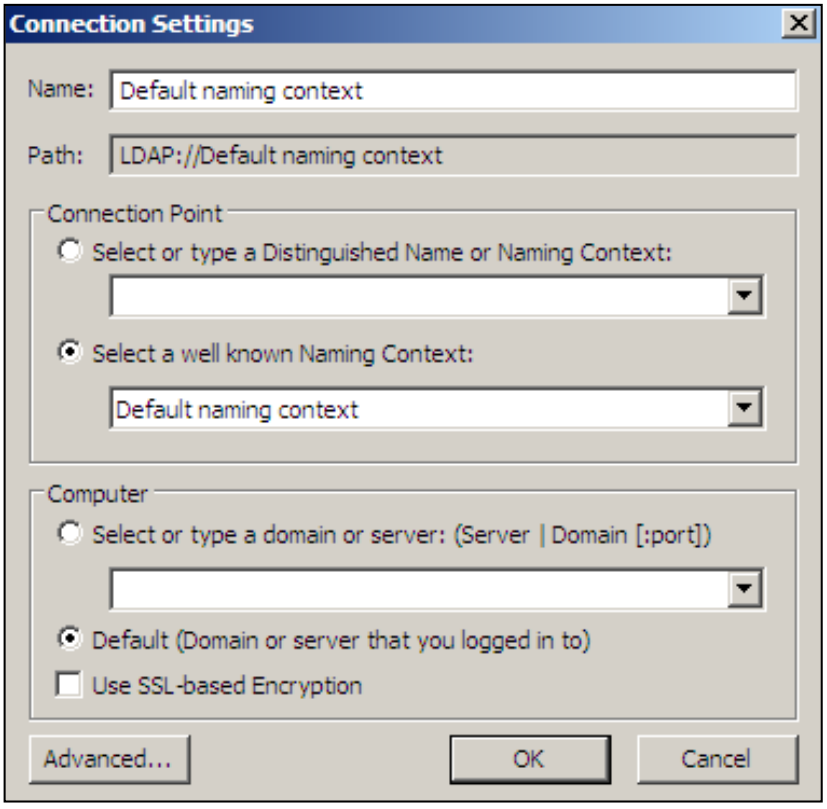


Figure 1. *ADSIEDIT connection options.*

Limitations of the Password Policy for Domain Users

To ensure you understand what I mean by domain users, let's scope out where these users reside. Domain users are those users that are created and stored in the Active Directory database. This means all users stored on your domain controllers (DCs) fall under this definition. One easy way to see whom this entails would be to open up the Active Directory Users and Computers (ADUC) and do a search on all users for that domain. Every user that shows up on that search falls into this scope.

The only way to control the password policy for domain users is to configure the aforementioned Account Policy in a GPO linked to the domain. That is the only way by default! Yes, it's true the GPO that contains the default password policy settings is the Default Domain Policy, but this is just the default. You can easily create a new GPO, configure the Account Policy settings as you wish and ensure this GPO has the highest precedence in the GPMC. The result will be that this new GPO will control the Account Policy settings for all domain users.

Default Password Policies

When you install a new Active Directory domain within Windows Server 2008 or Windows Server 2008 R2, or upgrade a Windows 2000 or Windows Server 2003 domain to have Windows Server 2008 or Windows Server 2008 R2 DCs, you can configure the domain to be at the Windows Server 2008 Domain Functional Level. At this functional level, you have more capabilities for configurations within the domain, but that doesn't mean that the default behavior changes. This is the case with the Account Policies for domain users.

When you have a basic Active Directory domain that's running at the Windows Server 2008 Domain Functional Level, the Account Policies for all domain users behave the exact same way they always have. A Windows Server 2008 or Windows Server 2008 R2 Active Directory domain, without FGPPs implemented, has the following characteristics for passwords affecting domain users (see page 7).

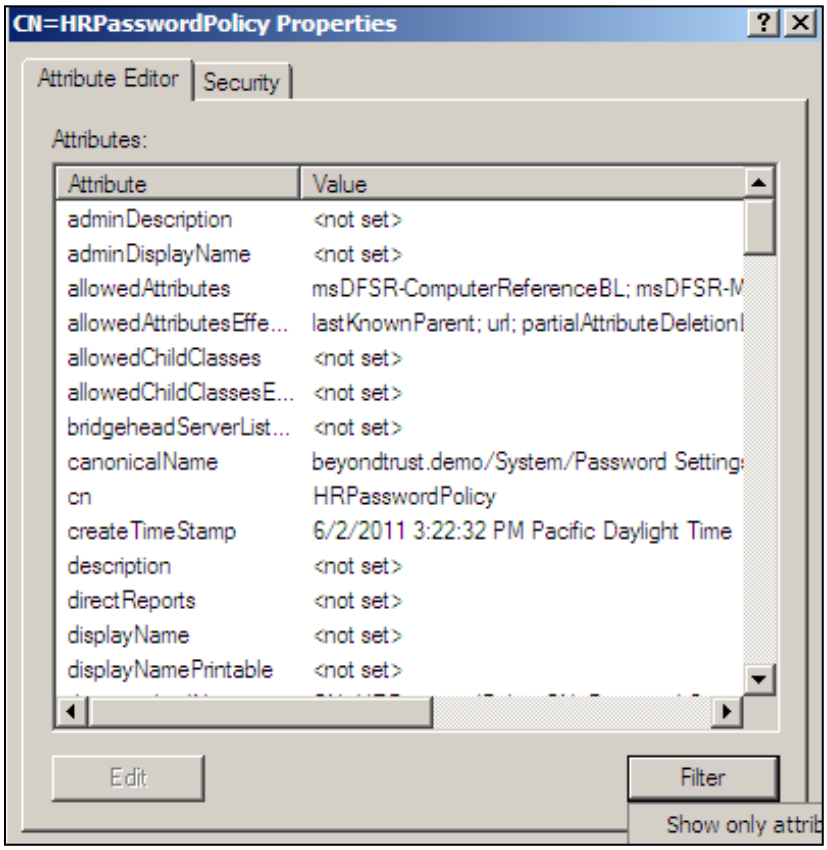


Figure 2. *FGPP/PSO filter settings to see correct attributes for setting up permissions.*

Attribute	Value	Explanation
Cn	HRPasswordPolicy	The name of the password policy object in Active Directory. Should be named after which user group it will affect.
msDS-PasswordSettingsPrecedence	10	A reference number, compared to other precedence settings for other FGPPs, which will resolve a conflict if user is member of two groups and each group has an FGPP. Smaller numbers have higher precedence.
msDS-PasswordReversibleEncryptionEnabled	False	Boolean value to define if passwords should be stored with reversible encryption.
msDS-PasswordHistoryLength	24	Number of unique passwords user must input before reusing a password.
msDS-PasswordComplexityEnabled	True	Defines if password complexity should be enabled or not.
msDS-MinimumPasswordLength	15	Minimum number of characters in each user password.
msDS-MinimumPasswordAge	-864000000000	Minimum password age (one day).
msDS-MaximumPasswordAge	-36288000000000	Maximum password age (42 days).
msDS-LockoutThreshold	30	Number of failed password attempts before user is locked out.
msDS-LockoutObservationWindow	-18000000000	Elapsed time to reset password lockout counter to maximum (in this case 30 minutes).
msDS-LockoutDuration	-18000000000	If the number of bad passwords is met in observation window time, this defines how long the account should remain locked out (30 minutes).

Table 2. *FGPP/PSO values to create a new object.*

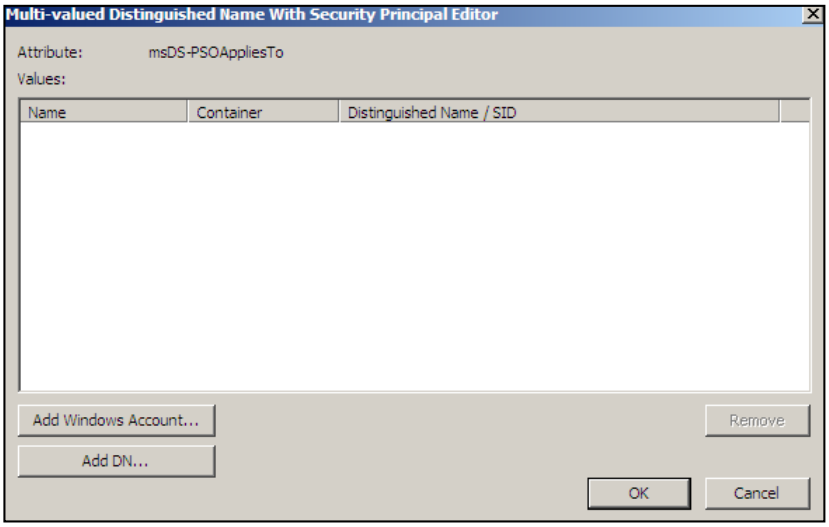


Figure 3. *Multi-valued Distinguished Name With Security Principal Editor for FGPP/PSO.*

Time Unit	Formula	Example Time	Value
m minutes	-60*(10^7) = - 600000000	30 minutes	-18000000000
h hours	-60*60* (10^7) = -36000000000	10 hours	-360000000000
d days	-24*60*60*(10^7) = -864000000000	42 days	-36288000000000

Table 3. *The "18" data type formatting for minutes, hours and days.*

- The Default Domain Policy defines the password policies by default for every user in Active Directory and every user located in the local SAM on every server and desktop that joins Active Directory.
- There can be only one password policy for domain users using Group Policy.
- It's not possible to configure the password policy in a GPO linked to an OU to affect users in the OU differently than other users in the domain or in a different OU.
- The password policy settings can't be extended to include additional settings without using a third-party tool or developing a custom password policy solution.
- It's not possible to configure a password policy for the root domain and have it "funnel" down to the other domains in the Active Directory tree.

Notice that the bullet list here is very similar to the list that was at the beginning of this article. The reason is that the Account Policy and password policy, even for Windows Server 2008 R2 domains, behave the exact same way as previous Windows 2000 and 2003 domains by default.

FGPPs

The preceding section was clear in stating that the default behavior of the Account Policies in a Windows Server 2008 and Windows Server 2008 R2 domain is exactly the same as it is in any other Active Directory domain before it. The difference comes when the Active Directory domain contains only Windows Server 2008 or Windows Server 2008 R2 DCs, and is moved to Windows Server 2008 Domain Functionality Level. When this occurs, it opens the door for FGPPs. Again, just to reiterate, without FGPPs configured, any Windows domain (including Windows Server 2008 R2 domains) acts the same as it always has.

The reason you'd want to configure FGPPs is to allow multiple password policies in the same Active Directory domain. Yes, that's correct. The same Active Directory domain can have multiple password policies. The result could be the following:

- IT employees have a minimum character limit of 20
- HR and finance employees have a minimum character limit of 15
- Standard employees have a minimum character limit of 10

In order to configure FGPPs, you won't be using Group Policy—FGPPs

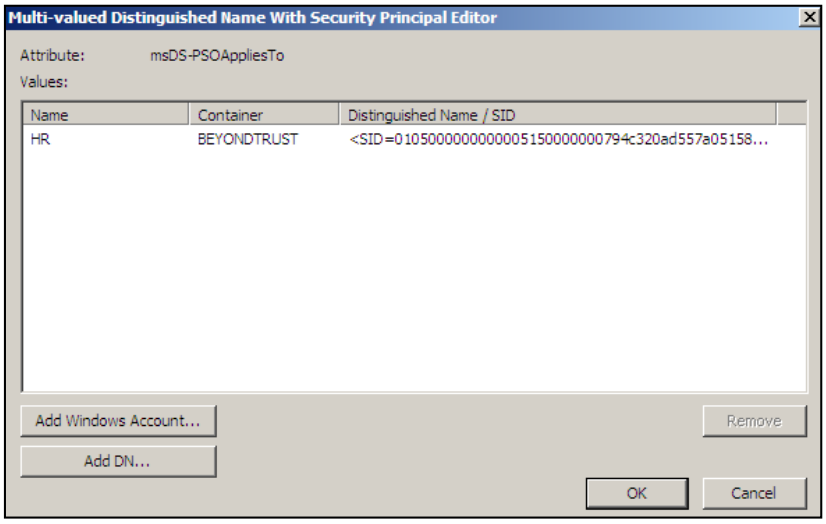


Figure 4. *HR group added to the HRPasswordPolicy FGPP/PSO.*

don't use Group Policy. Instead, the implementation of FGPPs is done by modifying the Active Directory database. The database is altered by adding one or more additional Active Directory objects, referred to as Password Settings Objects (PSOs). This might sound odd, and I must agree it is. If you decide to implement FGPPs, you'll have a mixture of Account Policy settings, via GPOs and FGPPs, in your environment.

To complete the configuration of your FGPPs, you'll need to complete the following steps:

1. Launch ADSIEDIT.MSC on your DC.
2. Select the View toolbar menu option, then click on the Connect to option.
3. In the Connection Settings dialog box click the OK button (see Figure 1, p. 5).
4. Within ADSIEDIT, expand the view of your domain down to the CN=System, so you can see the contents available under this node.
5. Right-click on the CN=Password Settings Container.
6. Select the option to Create | Object.
7. Fill out the values for each entry; Table 2 (p. 6) is a guide.

Note that the values inputs for minute/hour/day in Table 2 (p. 6) seem very odd. This is due to the fact that they're input in the "18" data type. The 18 data type follows an odd format, which can be seen in Table 3 (p. 7).

In order to link the FGPP/PSO to the correct user or group, you'll need to configure an object attribute. In order to see the correct object attribute, ensure the FGPP/PSO in ADUC or ADSIEDIT is set properly, which can be seen in Figure 2 (p. 6).

In the attribute list for your FGPP/PSO, scroll down to the msDS-PSOAppliesTo entry and double-click this attribute to see the Multi-valued Distinguished Name With Security Principal Editor dialog box, as shown in Figure 3 (p. 7).

You can enter a domain name, username or security group into the editor. Select the correct button, then add in your object to the editor. I added the HR group, as shown in Figure 4 (p. 7).

Verify that user in the HR group has the correct password policy by viewing the user account properties from within ADUC, then looking at the msDS-ResultantPSO attribute.

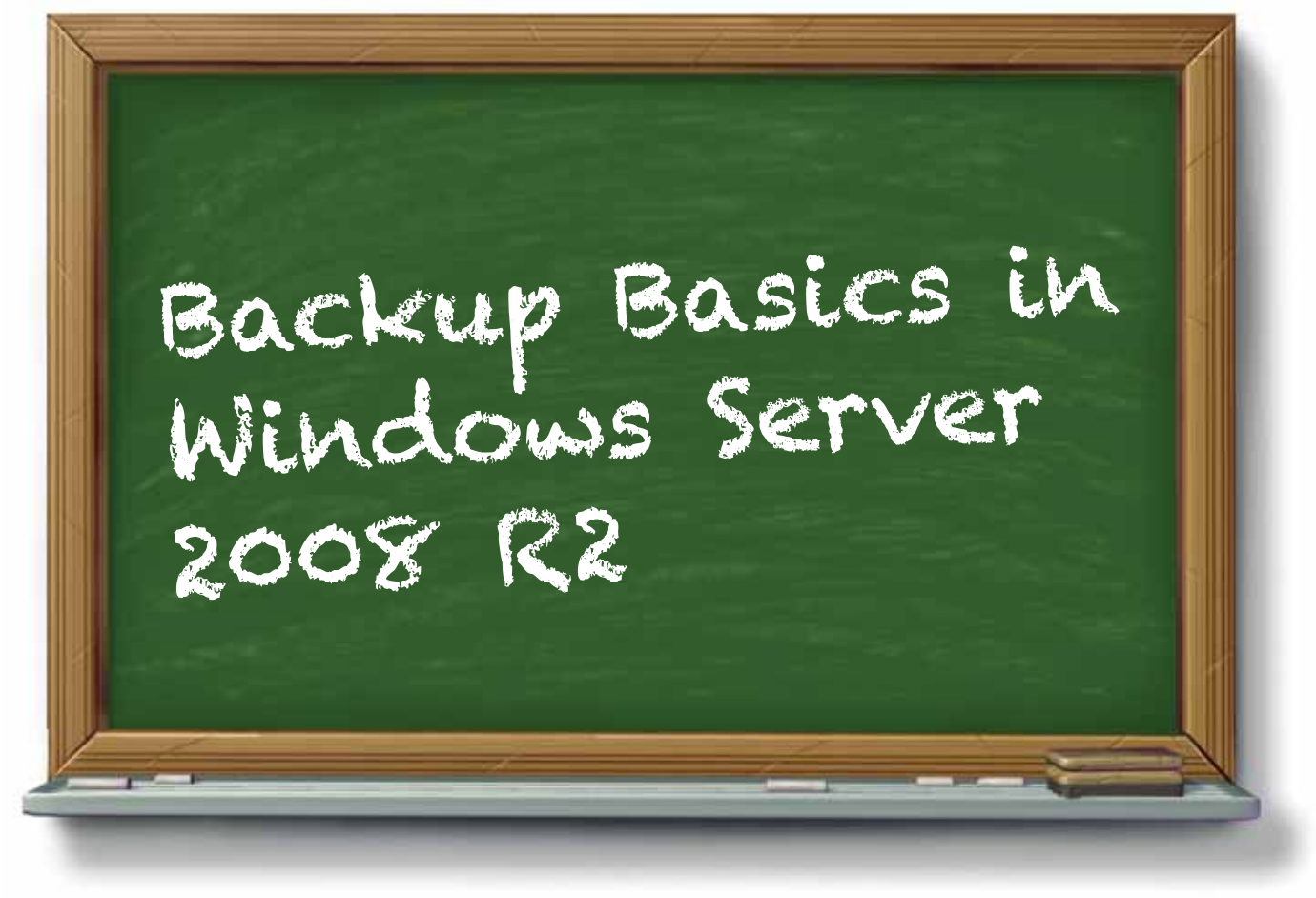
A New Path

The default password policy settings for a Windows Active Directory domain haven't changed for the past 11 years, and in a default Windows Server 2008 R2 domain they're the same to begin with. The Default Domain Policy controls all domain user password policies by default but can be altered by another GPO linked to the domain with higher precedence. Once the domain is configured to be a Windows Server 2008 Domain Functional Level domain, FGPPs can be used.

You can use ADSIEDIT.MSC to create and configure one or more FGPP objects or PSOs, which will now allow you to have multiple password policies in the same domain. The FGPPs/PSOs will be associated with a domain name, user or group—and have nothing to do with Group Policy, which you've known password policies to rely on for the past 11 years. Now you can obtain that segregation of password lengths for the different users in your single Active Directory domain. R

Derek Melber, MCSE, MVP, is an independent consultant and speaker, as well as the author of many IT books. He is president and CTO of BrainCore.Net and is author of "Windows Group Policy Resource Kit" (Microsoft Press, 2008). You can reach Melber at derekm@braincore.net.

BY JEFFERY HICKS



A free tool from Microsoft can make backing up data in Windows Server 2008 R2 efficient and almost hassle-free. Here's how to use it effectively.

Back in the day, Microsoft's free backup tool was the now venerable NTBackup. However, that utility has gone to the great recycle bin in the sky. Windows Server 2008 offers a new set of backup tools, and I want to show you how easy it is to use them with the new Windows Server 2008 R2. Be aware that the new backup feature can't manage backups created with NTBackup.

Installation

First off, we need to install the backup feature, as it's not installed by default. Use the Add Features wizard in Server Manager and add the Windows Server Backup Features (Figure 1, p. 10). I'm going to use the command-line tools sub-feature so that I can use Windows PowerShell, which I will explain in more detail later in this article.

You can also use command-line tools, including ServerManagerCMD.exe, to install the feature:

```
C:\ servermanagercmd –install backup-features
```

The next step is to identify locations for your backups. You can back up files to a network share, a local volume or a dedicated disk. You can't back up data to tape, but given the growth and widespread availability of inexpensive USB-attached storage, this isn't that much of a setback these days.

Creating a Backup Job

Windows Backup is intended to provide a one-stop setup to protect a server. You can enable a scheduled task to back up files and the system state, or to provide for a bare-metal restore. Microsoft assumes you'll have one scheduled task for this purpose. I'm assuming that you're

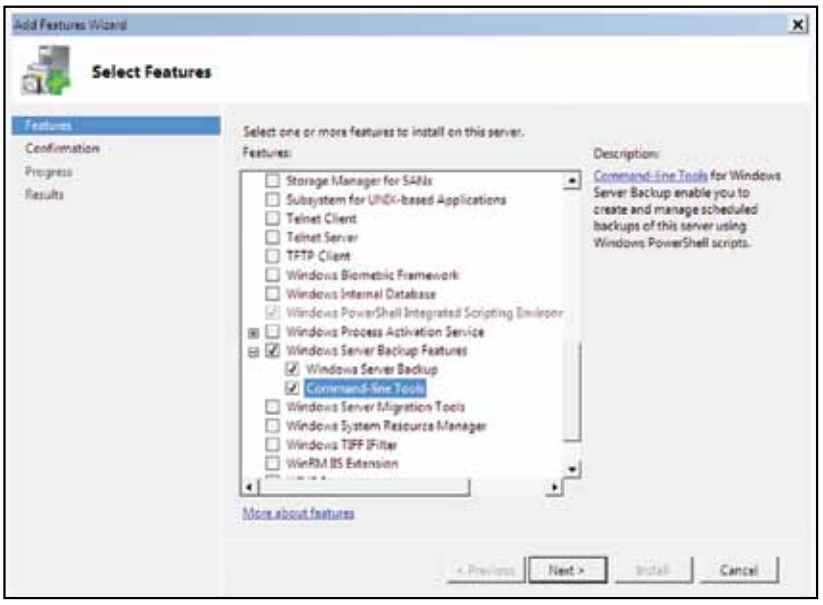


Figure 1. *The backup feature is not installed by default, so you must install it using the Add Features Wizard.*

using the Windows Backup feature because of limited budget and are after maximum protection given the utility's constraints.

After you install the Windows Backup feature, expand the Storage node in Server Manager and select Windows Server Backup. In the Actions pane, select "Backup Schedule," which will start the Backup Schedule Wizard. Then, click Next on the Getting Started screen.

During step two, specify what type of backup you want. Try doing a complete server backup. You can also create a custom backup and pick items such as selected files and system state. I'll show you how to do a quick file backup later, but for now I'm assuming you want complete server protection.

In the third step, specify when you want the backup task to run. Most of the time, a single backup should be sufficient, but you can run it more than once a day. If you're backing up critical files, this might be a good choice.

In step four, determine where to store the backup. Microsoft recommends using a dedicated hard disk. Remember, this drive will be reformatted and unavailable for anything else. You can also use a volume or a network share. Pay close attention to the warnings and limitations. You might see a warning reminding you that the disk will be reformatted. If you don't see all the disks, click the Show All Available Disks button to refresh. When you select a new disk, you'll be warned.

Once selected, you'll have a chance to confirm your backup settings. If anything is incorrect, use the Previous button to go back and correct the error. If all goes well, you should get a summary screen. The next day, you can check the Windows Server Backup node for results or errors.

You can also use Windows Backup to run a one-time backup. Select the Backup Once option in the Actions pane. You can use the same settings as your scheduled job or pick something completely different. If you select the latter, the wizard runs again and you can enter new parameters. For example, you might want to copy files to a network share. Remember, any existing backups to the same folder will be overwritten. The backup will execute immediately. If this is a separate backup task you'd like to do often, then you'll want to take advantage of a scripted solution from the command line or Windows PowerShell. I'll cover that procedure later.

Restoring Data

Windows Backup uses a time stamp as version information. Using the Recover task launches a wizard that's easy to follow. Select the appropriate

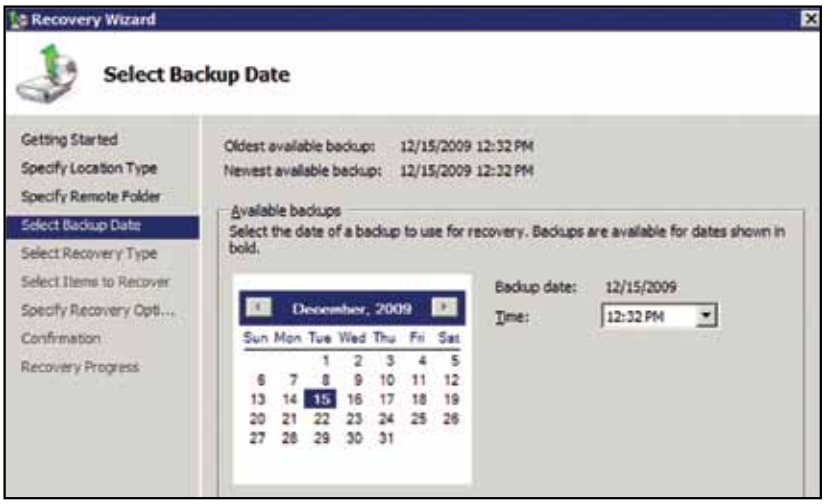


Figure 2. *Data recovery is easy with the Recovery Wizard.*

backup source. The Recovery Wizard will display a datetime control of all available backups (see Figure 2, p. 10). Select the appropriate one. Depending on the type of backup, you may only have once choice.

Moving on, select what type of data you want to recover. If you select Files and Folders, you'll be able to highlight the files you want to recover. Unfortunately, selecting files from multiple directories is next to impossible. You can easily recover everything or recover selected files from one directory. Keep that in mind when you set up the backup job.

When you recover files, you'll need to specify the target folder, which can be the original folder or an alternate location. You can also control what happens when you restore a current file if a current version exists. You can create a copy so that you have both versions; you can overwrite the existing version; or you can skip restoring if an existing version is detected. The recovery process happens immediately.

Using WBADMIN.EXE

If you installed the command backup tools, then you have a few more options. Open a command prompt and look at help for WBADMIN.EXE. You can use the tool to set up a scheduled backup, but I think the GUI is much easier. I find this tool more useful for creating one-time backup jobs. Run the following command to see syntax help:

```
C:\> wbadmin start backup /?
```

I don't have space to cover all the options, but let me demonstrate how you might use the command-line tool to periodically back up files to a network share:

```
@echo off
::Demo-Backup.bat
::demonstration script using
WBADMIN.EXE on a Windows
Server 2008 R2 Server

rem backup share UNC
set backupshare=\\mycompany-
dc01\backup

rem files and folders to include
set include=c:\scripts,c:\files

rem define date time variables for
building the folder name
set m=%date:~4,2%
set d=%date:~7,2%
set y=%date:~10,4%
set h=%time:~0,2%
set min=%time:~3,2%
set sec=%time:~6,2%

rem defining a new folder like \\
mycompany-dc01\backup\
RESEARCHDC\12152009_132532
set newfolder=%backupshare%\%c
omputername%\%m%%d%%y%_%h
%%min%%sec%
echo Creating %newfolder%

mkdir %newfolder%

rem run the backup
echo Backing up %include% to
%newfolder%
wbadmin start backup
-backuptarget:%newfolder%
-include:%include% -quiet
rem Clear variables
set backupshare=
set include=
set m=
set d=
set y=
set h=
set min=
set sec=
set newfolder=
```

I don't want to overwrite any existing backups, so I'll create a new folder that uses the computer name and a datetime stamp as part of the file name. The batch file has code to handle that task. The main function of the script is to call WBADMIN.EXE to create a backup on the specified share. Look at syntax help if you want to tweak this step. I like this script because I can set up my own scheduled task using the Task Scheduler. So, even though the backup wizard only lets me create one scheduled task, I can create as many as I want using WBADMIN.EXE. I can also use this tool to create system state backups, as well.

To see what backup jobs have executed, run this command:

```
C:\> wbadmin get versions
```

Pay attention to the version identifier; you'll need it to recover files using WBADMIN (you can also use the Recovery Wizard).

Backing up with PowerShell

The other command-line approach is to use Windows Backup PowerShell cmdlets. To access them, you'll first need to load the Windows backup snap-in:

```
PS C:\> add-pssnapin Windows.
ServerBackup
```

To see which cmdlets are included, use Get-Command:

```
PS C:\> get-command -pssnapin
windows.server backup
```

Unfortunately, creating a backup job is a multistep process. While you can type the necessary commands at the prompt interactively, I think you'll find it easier with a scripted approach. Here's a PowerShell

version of my original batch file:

```
#requires -version 2.0
#requires -pssnapin Windows.
ServerBackup

#Demo-WBBackup.ps1

$policy = New-WBPolicy
$files=new-WBFileSpec c:\
scripts,c:\files
Add-wbFileSpec -policy $policy
-filespec $files
$backdir=("\\mycompany-dc01\
backup\{0}\{1:MMd dyyyy_
hhmmss}" -f
$env:computername,(get-date))

write-host "Creating $backdir"
-foregroundcolor Green
mkdir $backdir | out-null

$backupLocation = New-WB-
BackupTarget -network $backdir

Add-WBBackupTarget -Policy $policy
-Target $backupLocation

write-host "Backing up $files to
$backdir" -fore groundcolor Green
$policy
Start-WBBackup -Policy $policy
```

The PowerShell cmdlets are based around creating and executing a policy. The policy includes the files or volumes to include or exclude, as well as where to back up the files and a few assorted options. You can also create system-state and bare-metal recovery jobs. In my demonstration, I'm simply backing up a few directories. The Start-WBBackup cmdlet carries out the backup task.

When you look at the list of Windows Backup cmdlets, you'll notice one glaring omission. There are no cmdlets for restoring data. I imagine the assumption is that you wouldn't want to automate this step, although you can with WBADMIN.EXE. Perhaps cmdlets will be added in the future. In the meantime, you can use the Recovery Wizard or WBADMIN.EXE to restore files.

Your Turn

As you try your hand at these tools, I'm sure you'll realize there's a great deal more that Windows Backup brings to the party. As with any backup software, make sure you practice the restore process in a non-production setting. You don't want to learn the process when you have to recover for real and your boss is breathing down your neck. Become familiar with the process so that when the time comes, you end up being the hero.

Jeffery Hicks (jhicks@redmondmag.com), MCSE, MCSA, MCT, is a Microsoft MVP and author, trainer and consultant. A 17-year IT veteran specializing in admin scripting and automation, Hicks is an active blogger and conference presenter. His latest book is "Windows PowerShell 2.0: TFM" (Sapien Press, 2009).