Most Common Mistakes in Active Directory and Domain Services – Part 1

Omer Eldan Active Directory, Group Policy, PowerShell December 31, 2018

4 Minutes

As a Premier Field Engineer (PFE) at Microsoft, I encounter new challenges on a daily basis. Every customer has its own uniqueness, and each environment is different from the other.

And yet, there are several things I repeatedly encounter over and over again. Common mistakes that IT administrators make because lack of knowledge or changes in products they are not aware of.

This blog post is the first part of a series which will cover several of those mistakes. So… Let's get started!

Series:

- **Part 1**
- Part 2
- Part 3

Mistake #1: Configuring Multiple Password Policies for Domain Users Using Group Policy

When reviewing Group Policy settings, I often find Group Policies Objects (GPOs) that contain 'Password Policy' settings.

For example, when looking into a "Servers Policy" GPO, I can see that it has Password Policy settings defined, including Maximum password age, Minimum password length and so on.

When I ask the customer about it, he tells me that this policy was built to set a different password policy for some admins accounts or any other group of users.

As you already know (or might have guessed), this is NOT the correct way to configure different Password Policies in your environment. Here's why:

- Password Policy settings in GPO affect computers, not users.
- When you change your Domain User password, the password change takes place on the Domain Controllers.
- Therefore, the Password Policy that takes effect is the one applied on your Domain Controllers, usually by the 'Default Domain Policy" GPO.
- More accurate, the Domain Controller that holds the PDC Emulator FSMO role is the one responsible for applying the Password Policy for the domain level.
- In terms of Group Policy, there can be only one password policy for domain users.

Bottom Line: Configure a GPO with password policy and link it to an Organizational Unit (OU) won't change the password policy for users within this OU.

Do It Right: Use FGPP.

Mistake #2: Removing "Authenticated Users" from the Group Policy Object Security Filtering

In June 2016, Microsoft released a security update that changes the security context with which user group policies are retrieved.

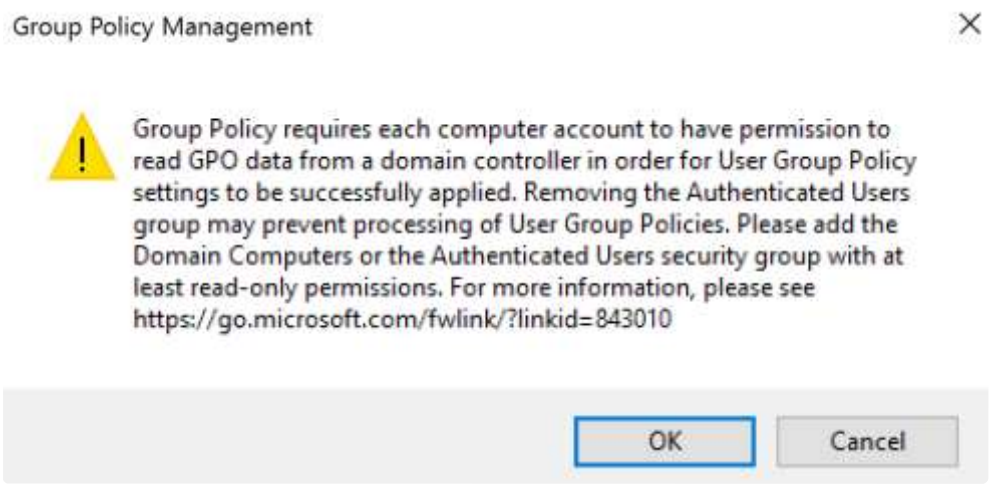
Before that update, user group policies were retrieved by using the user's security context. After installing the update, user group policies are retrieved by using the **computer's** security context.

Therefore, you should always make sure that any Group Policy in your environment could be retrieved by the relevant computer accounts.

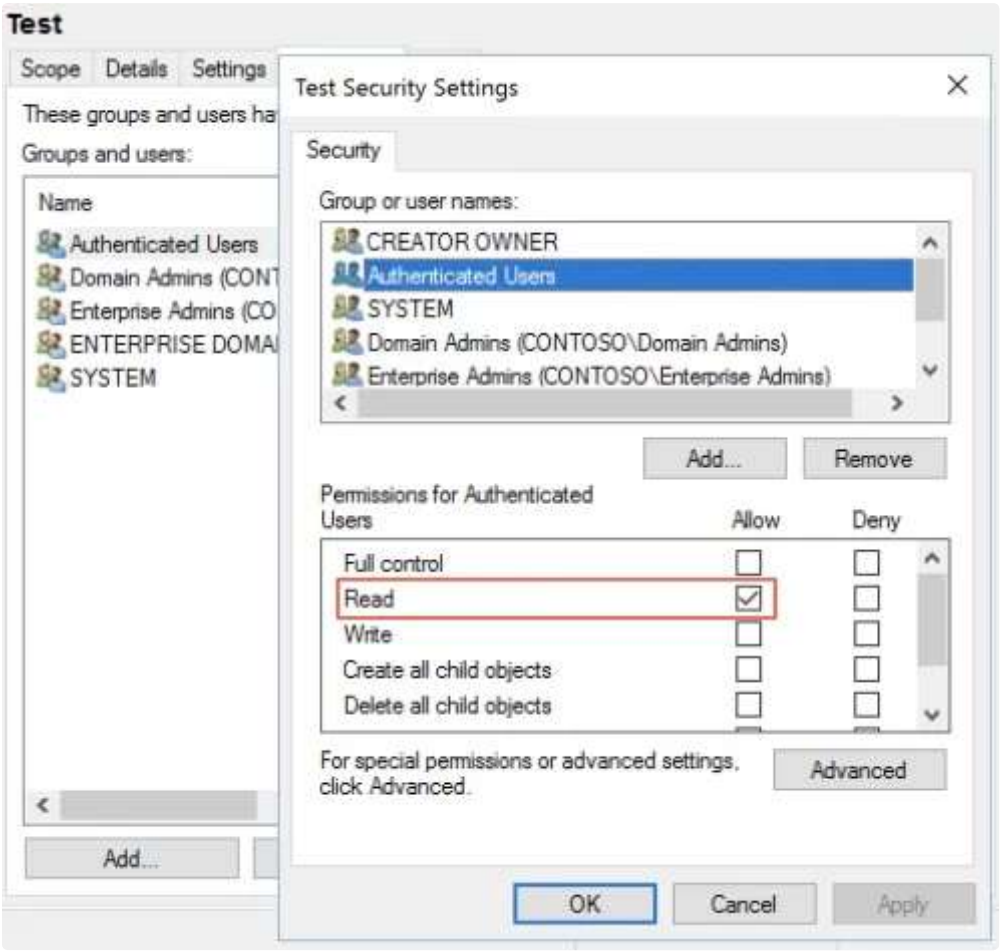
Because a lot of people are not aware of this change, I usually find Group Policies with missing permissions that are not being applied at all.

When changing Group Policy Security Filtering scope from "Authenticated Users" to any other group, the "Authenticated Users" (which contains computers account as well) are removed from the Group Policy delegation tab. As a result, computer accounts don't have the necessary "Read" permissions in order to access and retrieve group policies.

In recent versions of Group Policy Management, a warning message appears when removing the default "Authenticated Users" from the "Security Filtering" tab:



That is why you must validate that any Group Policy has the "Authenticated Users" or "Domain Computers" groups with "Read" permissions. Make sure that that you specify "Read" permission only, without selecting the "Apply group policy" permissions (otherwise any user or computer will apply this Group Policy).



The following PowerShell function can help you identify GPOs with missing permissions (missing both 'Authenticated Users' and 'Domain Computers' groups):

```
1  #Find Group Policies with Missing Permissions
```

```powershell
Function Get-GPMissingPermissionsGPOs
{
   $MissingPermissionsGPOArray = New-Object System.Collections.ArrayList
   $GPOs = Get-GPO -all
   foreach ($GPO in $GPOs) {
        If ($GPO.User.Enabled) {
            $GPOPermissionForAuthUsers = Get-GPPermission -Guid $GPO.Id -All | select -ExpandPrope
            $GPOPermissionForDomainComputers = Get-GPPermission -Guid $GPO.Id -All | select -Expan
            If (!$GPOPermissionForAuthUsers -and !$GPOPermissionForDomainComputers) {
                $MissingPermissionsGPOArray.Add($GPO)| Out-Null
            }
        }
    }
    If ($MissingPermissionsGPOArray.Count -ne 0) {
        Write-Warning  "The following Group Policy Objects do not grant any permissions to the 'Au
        foreach ($GPOWithMissingPermissions in $MissingPermissionsGPOArray) {
            Write-Host "'$($GPOWithMissingPermissions.DisplayName)'"
        }
    }
    Else {
        Write-Host "All Group Policy Objects grant required permissions. No issues were found." -F
    }
}
```

view raw

Get-GPMissingPermissionsGPOs.ps1

hosted with ❤ by GitHub

Bottom Line: Group Policies with missing permissions for computers account (“Authenticated Users”, “Domain Computers” or any other group that includes the relevant computers) will NOT be applied.

Do It Right: When changing Group Policy Security Filtering, make sure you add the “Authenticated Users” group in the delegation tab and provide it with “Read” permission only.

Mistake #3: Creating a DNS Conditional Forwarder as a Non-Active Directory Integrated Zones

When creating a DNS conditional forwarder using the DNS management console (GUI), it’s created, by default, as a non-Active Directory integrated zone, meaning that it’s saved locally in the server’s registry.

Creating a non-Active Directory integrated zone raises a few problems:

- Non-Active Directory zones do NOT replicate between the Active Directory Integrated DNS servers, therefore these zones might become out of sync when configured over two or more DNS servers.
- Non-Active Directory zones can be easily forgotten and abandoned when replacing Domain Controllers as part of an upgrade or restore procedures.
- In many cases, Non-Active Directory zones for conditional forwarder are defined on a single server, which causes inconsistent behavior between servers in terms of DNS resolving.

You can easily change this and create the zone as an Active Directory integrated zone by selecting the option "Store this conditional forwarder in Active Directory".

Using PowerShell, you can specify the parameter 'ReplicationScope' with either 'Forest' or 'Domain' scope to store the conditional forwarder zone in Active Directory:

```
1  Add-DnsServerConditionalForwarderZone -Name "contoso.com" -ReplicationScope "Forest"
```

view raw

Add-DnsServerConditionalForwarderZone.ps1

hosted with ❤ by GitHub

Bottom Line: Avoid using non-Active Directory integrated zones unless you have a really good reason.

Do It Right: When creating conditional forwarder using either PowerShell or the GUI, make sure to create it as an Active Directory-integrated forwarder.

Continue reading part 2 of the series.

Author

Omer Eldan

Tagged: DNS, PowerShell

2 thoughts on "Most Common Mistakes in Active Directory and Domain Services – Part 1"

jerryperes

June 10, 2019 at 8:59 am

Thanks for sharing helpful info about active directory and domain services. This post will help out many people who wants to buy domain services like me.

Log in to Reply

Omer Eldan
June 10, 2019 at 3:33 pm

Hi Jerry,
Thank you for the feedback! I'm glad you find it useful!

Log in to Reply

You must log in to post a comment.

Search This Blog

Search ...

Tags

Deep Technical

Microsoft Delivery Approach

Industry Solutions

Categories

Active Directory

AVD

Azure

Azure Active Directory

Azure MFA

Azure Monitor

Azure Sentinel

BI and Analytics

Certification

Defender

DevOps

Failover Clustering

Group Policy

Hyper-V

Identity

Infrastructure as Code

Intune

KMS

KQL

Log Analytics

Logic Apps

Microservices

Microsoft 365 Defender

Microsoft Authenticator Application

Microsoft Defender for Cloud Apps

Microsoft Defender for Endpoint

Microsoft Defender for Identity

Microsoft Endpoint Manager

Microsoft Intune

Microsoft Sentinel

Office 365

OMS

Performance

PowerShell

Security

SIEM

SOAR

Soft Skills

System Center

Uncategorized

Windows

WSUS

Follow Blog via Email

Enter your email address to follow this blog and receive notifications of new posts by email.

Email Address

Follow

About

Contact Us

Disclaimer