Azure Cloud & AI Domain Blog

AC&AI domain is the largest technology domain within the Microsoft Consulting Services Organization. We aim to deliver world-class solutions with our team of expert Consultants, Project Managers and Architects across Data & AI, Apps, Security and Azure Infrastructure

Most Common Mistakes in Active Directory and Domain Services – Part 3

Omer Eldan   Active Directory   January 27, 2019   6 Minutes

This blog post is the third (and last) part in the 'Most Common Mistakes in Active Directory In Domain Services" series.
In the previous parts, we covered some major mistake like configuring multiple password policies using GPO and keeping FFL/DFL at a lower version.
The 3'rd part of the series is no exception. we'll go on and review three additional mistakes and summarize this series.

Series:

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

Mistake #7: Installing Additional Server Roles and Applications on a Domain Controller

When I review a customer's Active Directory environment, I often find additional Windows Server roles (other than the default ADDS and DNS roles) installed on one or more of the Domain Controllers.

This can be any role – from RDS Licensing, through Certificate Authority and up to DHCP Server. Beside Windows Server roles, I also find special applications and features running on the Domain Controllers, like KMS

(Key Management Service) host for volume activation, or Azure AD Connect for integrating on-premises directories with Azure AD.

There is a wide variety of roles and applications which administrators install on the Domain Controllers, but there is one thing common to all of them: **Domain Controllers are NOT the place for them**.

By default, any Domain Controller in a domain provides the same functionality and features as the others, what makes the Active Directory Domain Services not be affected if one Domain Controller becomes unavailable. Even in a case where the Domain Controller holding the FSMO roles becomes unavailable, the Domain Services will continue to work as expected for most scenarios (at least in the short-term).

When you install additional roles and applications on your Domain Controllers, two problems are raised:

1. Domain Controllers with additional roles and features become unique and different compares to other Domain Controllers. If any of these Domain Controllers will be turned off or get damaged, its roles and features might be affected and become unavailable. This, in fact, creates a dependency between ADDS and other roles and affect the redundancy of the Active Directory Domain Services.
2. Upgrading your Active Directory environment becomes a much more complicated task. A DHCP Server or a Certificate Authority roles installed on your Domain Controllers will enforce you to deal with them first, and only then move forward and upgrade the Active Directory itself. This complexity might also affect other tasks like restoring a Domain Controller or even put a Domain Controller into maintenance.

This is why putting additional roles and applications on your Domain Controllers is not recommended for most cases.
You can use the following PowerShell script to easily get a report with your Domain Controllers installed roles. Pay attention that this script is working only for Windows Server 2012 and above. For Windows Server 2008, you can use WMI Query.

```
#Get Installed Roles on each Domain Controller
$DCsInForest = (Get-ADForest).Domains | % {Get-ADDomainController -Filter * -Server $_}
$DCsRolesArray = @()
foreach ($DC in $DCsInForest) {
    $DCRoles=""
    $Roles = Get-WindowsFeature -ComputerName $DC.HostName | Where-Object {$_.Installed -like "Tru
    foreach ($Role in $Roles) {
        $DCRoles += $Role.DisplayName +","
    }
    try {$DCRoles = $DCRoles.Substring(0,$DCRoles.Length-1)}
    catch {$DCRoles = "Server roles cannot be obtain"}
    $DCObject = New-Object -TypeName PSObject
    Add-Member -InputObject $DCObject -MemberType 'NoteProperty' -Name 'DCName' -Value $DC.HostNam
    Add-Member -InputObject $DCObject -MemberType 'NoteProperty' -Name 'Roles' -Value $DCRoles
    $DCsRolesArray += $DCObject
}
$DCsRolesArray | Out-GridView
```

view raw

Create-DomainControllersRolesReport.ps1

hosted with ❤ by GitHub

Bottom Line: Domain Controllers are designed to provide directory services for your users – allowing access to domain resources and respond to security authentication requests.
Mixing Active Directory Domain Services with other roles and applications creates a dependency between the two, affect Domain Controller performance and make the administrative tasks a much more complicated.

Do It Right: Use Domain Controllers for Active Directory Domain Services only, and install additional roles (let it be KMS or a DHCP server) on different servers.

Mistake #8: Deploying Domain Controllers as a Windows Server With Desktop Experience

When you install Windows Server, you can choose between two installation options:

- **Windows Server with Desktop Experience** – This is the standard user interface, including desktop, start menu, etc.
- **Windows Server** – This is the Server Core, which leaving the standard user interface in favor of command line.

Although Windows Server Core has some major advantages compares to Desktop Experience, most administrators are still choosing to go with the full user interface, even for the most convenient and supported server roles like Active Directory Domain Services, Active Directory Certificate Services, and DHCP Server.

Windows Core is not a new option, and it has been here since Windows Server 2008R2. It works great for the supported Windows roles and has some great advantages compares to the Windows Server with Desktop Experience. Here are the most significant ones:

- **Reduce potential attack surface and lower the chance for user mistakes** – Windows Server Core reduces the potential attack surface by eliminating binaries and features which does not require for the supporting roles (Active Directory Domain Services in our case).
  For example, the Explorer shell is not installed, which of curse reduces the risks and exploits that can be manipulated and used to attack the server.
  Other than that, when customers are using Windows Server with Desktop Experience for Active Directory Domain Services, they are also usually performing administrative tasks directly on their Domain Controllers

using Remote Desktop.
This is a very bad habit as it may have a significant impact over the Domain Controllers performance and functionality. It might also cause a Domain Controller to become unavailable by accidentally turn it off or running a heavy PowerShell script which drains the server's memory.

- **Improve administrative skills while still be able to use the GUI tools –** by choosing Windows Server Core, you'll probably get the chance to use some PowerShell cmdlets and improve your PowerShell and scripting skills.
  Some customers think that this is the only way to manage and administer the server and its role, but that's not true.
  Alongside the Command Line options, you'll find some useful remote management tools, including Windows Admin Center, Server Manager, and Remote Server Administration Tools (RSAT).
  In our case, the RSAT includes all the Active Directory Administrative tool like the Active Directory Users and Computers (dsa.msc) and the ADSI Editor (adsiedit.msc).
  It also important to be familiar with the 'Server Core App Compatibility Feature on Demand' (FOD), which can be used to increase Windows Server Core 2019 compatibility with other applications and to provide administrative tools for troubleshooting scenarios.
  My recommendation is to deploy an administrative server for managing all domain services roles, including Active Directory Domain Services, DNS, DHCP, Active Directory Certificate Services, Volume Activation, and others.
- Other advantages like reducing disk space and memory usage are also here, but they, by themselves, are not the reason for using Windows Server Core.

You should be aware that unlike Windows Server 2o12R2, you cannot convert Windows Server 2016/2019 between Server Core and Server with Desktop Experience after installation.

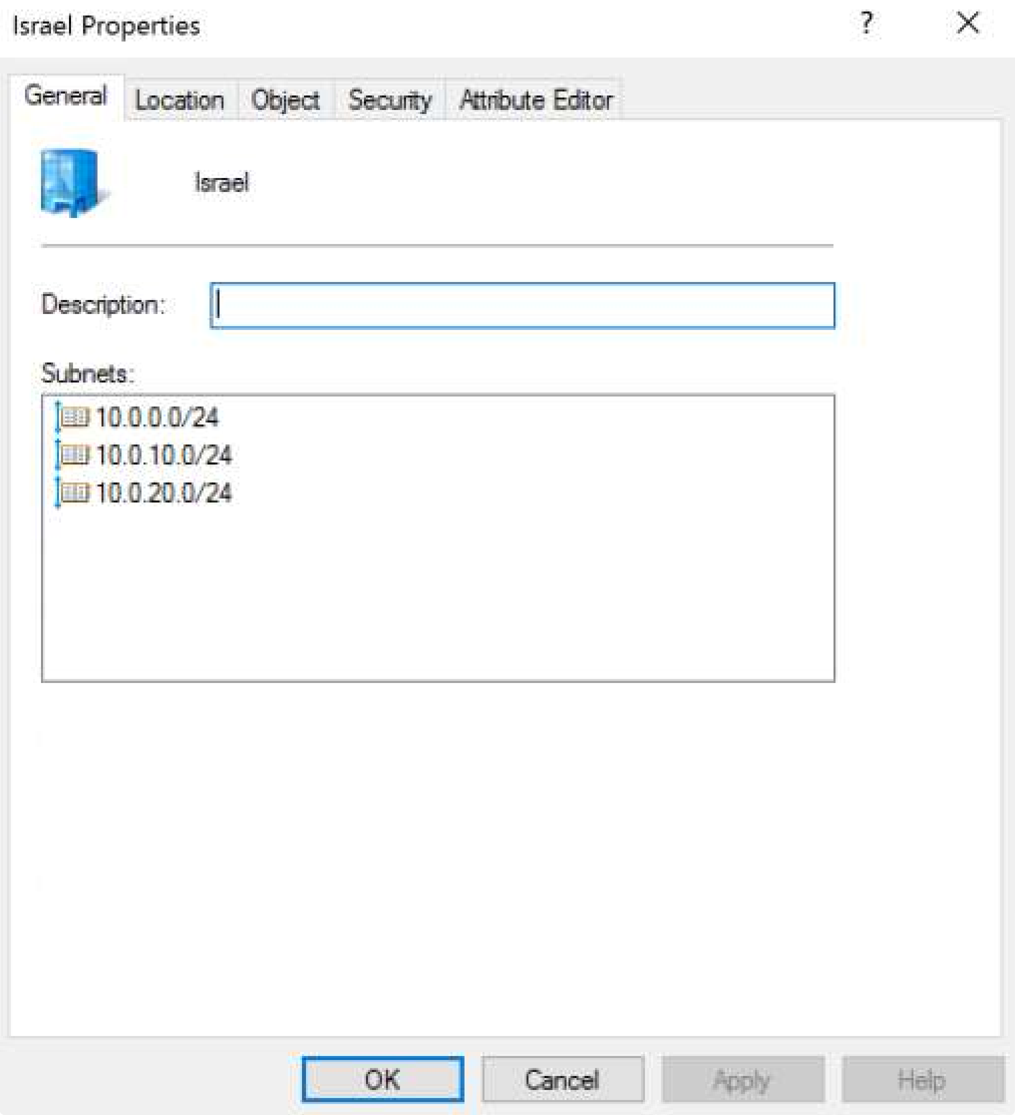
Bottom Line: Windows Server Core is not a compromise. For the supported Windows Server roles, it is the official recommendation by Microsoft. Using Windows Server with Full Desktop Experience increases the chances that your Domain Controllers will get messy and will be used for administration tasks rather than providing domain services.

Do It Right: Install your Domain Controllers as a Windows Server Core, and use remote management tools to administer your domain resources and configuration. Consider deploying one of your Domain Controller as a Windows Server with Full Desktop Experience for forest recovery scenarios.

Mistake #9: Use Subnets Without Mapping them to Active Directory sites

Active Directory uses sites for many purposes. One of them is to inform clients about Domain Controllers available within the **closest** site as the client.

For doing that, each site is associated with the relevant subnets, which correspond to the range of IP addresses in the site. You can use Active Directory Sites and Services to manage and associate your subnets.



Israel

Description:

Subnets:

10.0.0.0/24
10.0.10.0/24
10.0.20.0/24

When a Windows domain client is looking for the nearest Domain Controller (what's known as the DC Locator process), the Active Directory (or more precisely, the NetLogon in one of the Domain Controllers) is looking for the IP address of the client in its subnets-to-sites association data.
If the client's IP address is found in one of the subnets, the Domain Controller returns the relevant site information to the client, and the client use this information to contact a Domain Controller within its site.

When the client's IP address cannot be found, the client may connect to any Domain Controller, including ones that are physically far away from him.
This can result in communication over slow WAN links, which will have a direct impact on the client login process.

If you suspect that you have missing subnets in your Active Directory environment, you can look for event ID 5807 (Source: NETLOGON) within your Domain Controllers.
The event is created when there are connections from clients whose IP addresses don't map to any of the existing AD sites.
Those clients, along with their names and IP address, are listed by default in C:Windowsdebugnetlogon.log.

You can use the following PowerShell script to create a report of all clients which are not mapped to any AD sites, based on the Netlogon.log files from all of the Domain Controllers within the domain.

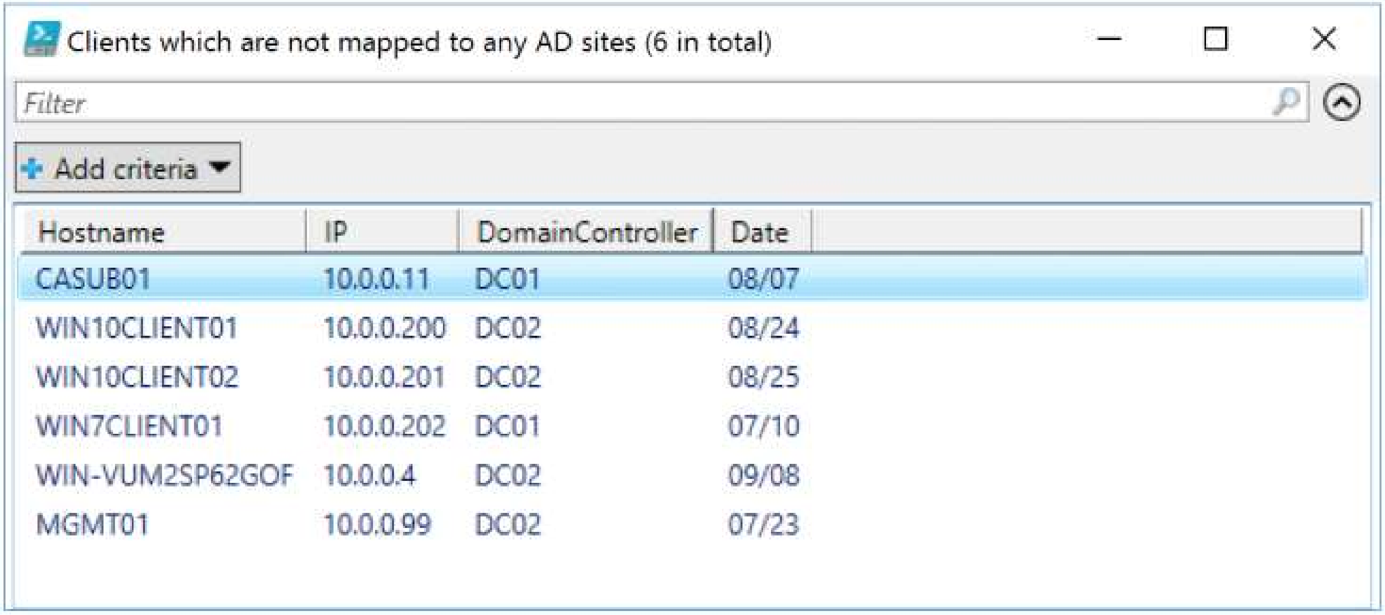
```powershell
#Get Domain Controllers for current domain
$DCs = Get-ADGroupMember "Domain Controllers"
#Initiate the clients array
$Clients = @()
Foreach ($DC in $DCs) {
    #Define the netlogon.log path
    $NetLogonFilePath = "\\" + $DC.Name + "\C$\Windows\debug\netlogon.log"
    #Reading the content of the netlogon.log file
    try {$NetLogonFile = Get-Content -Path $NetLogonFilePath -ErrorAction Stop}
    catch {"Error reading $NetLogonFilePath"}
    foreach ($Line in $NetLogonFile) {
        #Splitting the line to isolate each variable
        $ClientData = $Line.split(' ')
        #Creating the client object
        $ClientObject = New-Object -TypeName PSObject
        Add-Member -InputObject $ClientObject -MemberType NoteProperty -Name 'Hostname' -Value $Cl
        Add-Member -InputObject $ClientObject -MemberType NoteProperty -Name 'IP' -Value $ClientDa
        Add-Member -InputObject $ClientObject -MemberType NoteProperty -Name 'DomainController' -V
        Add-Member -InputObject $ClientObject -MemberType NoteProperty -Name 'Date' -Value $Client
        $Clients += $ClientObject
     }
}
$UniqueClients = $Clients | Sort-Object -Property IP -Unique
$UniqueClients | Out-GridView -Title "Clients which are not mapped to any AD sites ($($UniqueClien
```

view raw

Create-ClientsWithNoAssociatedSiteReport.ps1

hosted with ❤ by GitHub

The script output should look similar to this:


Clients which are not mapped to any AD sites (6 in total)

Hostname	IP	DomainController	Date
CASUB01	10.0.0.11	DC01	08/07
WIN10CLIENT01	10.0.0.200	DC02	08/24
WIN10CLIENT02	10.0.0.201	DC02	08/25
WIN7CLIENT01	10.0.0.202	DC01	07/10
WIN-VUM2SP62GOF	10.0.0.4	DC02	09/08
MGMT01	10.0.0.99	DC02	07/23

Bottom Line: The association of subnets to Active Directory sites has a significant impact on the client machines performance. Missing this association may lead to poor performance and unexpected login times.

Do It Right: Work together with your IT network team to make sure any new scope is covered and has a corresponded subnet that associated to an Active Directory site.

So… this was the last part of the 'Most Common Mistakes in Active Directory and Domain Services' series. Hope you enjoyed reading these blog posts and learned a thing or two.

Author

Omer Eldan

4 thoughts on "Most Common Mistakes in Active Directory and Domain Services – Part 3"

Jerry Peres
July 22, 2019 at 7:31 am

Thanks for sharing great stuff about active directory and domain services. It's really helpful for any person who wants to buy a new domain.

Log in to Reply

Omer Eldan
July 22, 2019 at 1:49 pm

Hi Jerry,

Thank you for the feedback!
I'm very glad you find it useful.

Log in to Reply

Ajay Parasher
August 13, 2019 at 8:10 am

Thanks , Omer . Really helpful for KMS and VMAT !

Log in to Reply

Omer Eldan
August 13, 2019 at 8:48 am

Thank you Ajay for your feedback! Happy to hear you found 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