

Get-AllDisabledandInactiveComputers

How to Find Disabled Computers in Active Directory Using PowerShell Report?

For those who prefer command-line interfaces and automation, PowerShell offers robust capabilities to manage Active Directory, and also using it you can easily find disabled computers in your AD environment.

Here's how:

Launch a new PowerShell module and type:

```
Get-ADComputer -Filter "Enabled -eq 'false'"  
Select Name, Enabled <# Add Other attributes you wish to see #>
```

For a more in-depth result, use the following script:

```
# Import the Active Directory module  
Import-Module ActiveDirectory  
  
# Define the thresholds for inactive computers (e.g., 90 days)  
$inactiveThreshold = (Get-Date).AddDays(-90)  
  
# Fetch all computer objects from Active Directory  
$computers = Get-ADComputer -Filter * -Property Name, DistinguishedName, Enabled,  
LastLogonDate
```

```
# Initialize arrays to hold categorized computer objects
$normalComputers = @()
$inactiveComputers = @()
$disabledComputers = @()

# Categorize the computer objects
foreach ($computer in $computers) {
    if (-not $computer.Enabled) {
        $disabledComputers += $computer
    } elseif ($computer.LastLogonDate -lt $inactiveThreshold) {
        $inactiveComputers += $computer
    } else {
        $normalComputers += $computer
    }
}

# Function to create a custom object for export
function Set-CustomObject {
    param (
        [array]$Computers,
        [string]$Category
    )
    $result = @()
    foreach ($computer in $Computers) {
        $obj = [PSCustomObject]@{
            Name = $computer.Name
            DistinguishedName = $computer.DistinguishedName
            Category = $Category
        }
        $result += $obj
    }
    return $result
}

# Combine all results
$results = @()
$results += Set-CustomObject -Computers $normalComputers -Category "Normal"
$results += Set-CustomObject -Computers $inactiveComputers -Category "Inactive"
$results += Set-CustomObject -Computers $disabledComputers -Category "Disabled"
```

```
# Export to CSV
$csvPath = "C:\Temp\ADComputerCategories.csv"
$allResults | Export-Csv -Path $csvPath -NoTypeInfoation

# Function to display the results in a colored table
function Show-Results {
param (
[array]$Computers,
[string]$Category,
[string]$Color
)

Write-Host "$Category Computers:" -ForegroundColor $Color
$Computers | Format-Table Name, DistinguishedName, @{Name="Category"; Expression =
{$Category}} -AutoSize
Write-Host ""
}

# Display the results
Show-Results -Computers $normalComputers -Category "Normal" -Color "Green"
Show-Results -Computers $inactiveComputers -Category "Inactive" -Color "Yellow"
Show-Results -Computers $disabledComputers -Category "Disabled" -Color "Red"

Write-Host "Results exported to $csvPath"
```

```
computer-check.ps1 X
12 $normalComputers = @()
13 $inactiveComputers = @()
14 $disabledComputers = @()
15
16 # Categorize the computer objects
17 foreach ($computer in $computers) {
18     if (-not $computer.Enabled) {
19         $disabledComputers += $computer
20     } elseif ($computer.LastLogonDate -lt $inactiveThreshold) {
21         $inactiveComputers += $computer
22     } else {
23         $normalComputers += $computer
24     }
25 }
26
```

```
PS C:\Users\Administrator> C:\Users\Administrator\Documents\computer-check.ps1
Normal Computers:
Name DistinguishedName Category
----
CU14 CN=CU14,OU=Domain Controllers,DC=cu14mail,DC=local Normal

Inactive Computers:
Name DistinguishedName Category
----
Test-Computer CN=Test-Computer,CN=Computers,DC=cu14mail,DC=local Inactive
EnabledComputerTest CN=EnabledComputerTest,CN=Computers,DC=cu14mail,DC=local Inactive

Disabled Computers:
Name DistinguishedName Category
----
DisableComputerTest CN=DisableComputerTest,CN=Computers,DC=cu14mail,DC=local Disabled

Results exported to C:\Users\Administrator\Desktop\ADComputerCategories.csv
```

Legacy Techniques to Find Stale Computer Objects in AD

If you operate an older version of Windows Server, then the command line will suffice. It is similar to the one used to [find locked out accounts in Active Directory](#) setup.

Open CMD on your workstation and type

```
> dsquery computer -disabled
```

Or

```
> search-adaccount -accountinactive -computersonly
```

Inactive Computers

Another method is to use the Visual Basic script.

```
' This code finds disabled computer accounts in an AD domain.

' ----- SCRIPT CONFIGURATION -----
strDomainDN = "" ' To find disabled computers in Active Directory replace with your actual domain
' ----- END CONFIGURATION -----

strBase = "<LDAP://" & strDomainDN & ">";
strFilter = "(&(objectclass=computer)(userAccountControl:1.2.840.113556.1.4.803:=2))" ' Filter for disabled accounts
strAttrs = "name;userAccountControl" ' Retrieve name and account control attribute

Const ADS_UF_ACCOUNTDISABLE As Integer = &H2 ' Flag for disabled account

Set objConn = CreateObject("ADODB.Connection")
objConn.Provider = "AdsDSOobject"
objConn.Open "Active Directory Provider"
Set objRS = objConn.Execute(strBase & strFilter & strAttrs & strScope)

objRS.MoveFirst
While Not objRS.EOF
If (objRS.Fields("userAccountControl").Value And ADS_UF_ACCOUNTDISABLE) = ADS_UF_ACCOUNTDISABLE
Wscript.Echo objRS.Fields(0).Value & " (Disabled)"
Else
Wscript.Echo objRS.Fields(0).Value & " (Enabled)"
End If
objRS.MoveNext
Wend

' Clean up
Set objRS = Nothing
Set objConn = Nothing
```

Revision #2

Created 2025-03-13 16:27:57 UTC by Ryan

Updated 2025-03-13 16:30:56 UTC by Ryan