

AD Health Check With Email

```
<#PSScriptInfo

.VERSION 1.0

.GUID 30c7c087-1268-4d21-8bf7-ee25c37459b0

.AUTHOR Vikas Sukhija

.COMPANYNAME TechWizard.cloud

.COPYRIGHT

.TAGS

.LICENSEURI

.PROJECTURI https://techwizard.cloud/2021/05/04/active-directory-health-check-v2/

.ICONURI

.EXTERNALMODULEDEPENDENCIES

.REQUIREDSCRIPTS

.EXTERNALSCRIPTDEPENDENCIES

.RELEASENOTES https://techwizard.cloud/2021/05/04/active-directory-health-check-v2/

.PRIVATEDATA

#>

<#
```

.DESCRIPTION

Date: 12/25/2014

AD Health Status

Satus: Ping,Netlogon,NTDS,DNS,DCdiag Test(Replication,sysvol,Services)

Update: Added Advertising

Update: 5/3/2021 version2 with parameters to make it more generic

```
#>

#####Paramters#####
param (
    [string]$Smtphost = "FULL EMAIL SERVER",
    [string]$SMTPPort = "25",
    [string]$from = "EMAIL ADDRESS",
    [string]$To = "EMAIL ADDRESS",
    [string]$Attachment = "C:\Program Files\WindowsPowerShell\Scripts\ADReport.htm",
    [string]$Subject = "Domain Report",
    [String[]]$EmailReport = "EMAIL ADDRESS",
    $timeout = "120"
)

#####Define Variables#####
$EmailReport = $EmailReport -split ','
$report = ".\ADReport.htm"

if((test-path $report) -like $false)
{
    new-item $report -type file
}

#####Get ALL DC Servers#####
$getForest = [system.directoryservices.activedirectory.Forest]::GetCurrentForest()

$DCServers = $getForest.domains | ForEach-Object {$_.DomainControllers} | ForEach-Object
{$_.Name}

#####HTMl Report Content#####
Clear-Content $report
Add-Content $report "<html>"
Add-Content $report "<head>"
Add-Content $report "<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>"
Add-Content $report '<title>AD Status Report</title>'
add-content $report '<STYLE TYPE="text/css">'
```

```

add-content $report "<!--"
add-content $report "td {"
add-content $report "font-family: Tahoma;"
add-content $report "font-size: 11px;"
add-content $report "border-top: 1px solid #999999;"
add-content $report "border-right: 1px solid #999999;"
add-content $report "border-bottom: 1px solid #999999;"
add-content $report "border-left: 1px solid #999999;"
add-content $report "padding-top: 0px;"
add-content $report "padding-right: 0px;"
add-content $report "padding-bottom: 0px;"
add-content $report "padding-left: 0px;"
add-content $report "}"
add-content $report "body {"
add-content $report "margin-left: 5px;"
add-content $report "margin-top: 5px;"
add-content $report "margin-right: 0px;"
add-content $report "margin-bottom: 10px;"
add-content $report ""
add-content $report "table {"
add-content $report "border: thin solid #000000;"
add-content $report "}"
add-content $report "-->"
add-content $report "</style>"
Add-Content $report "</head>"
Add-Content $report "<body>"
add-content $report "<table width='100%'"
add-content $report "<tr bgcolor='Lavender'"
add-content $report "<td colspan='7' height='25' align='center'"
add-content $report "<font face='tahoma' color='#003399' size='4'><strong>Active Directory  
Health Check</strong></font>"
add-content $report "</td>"
add-content $report "</tr>"
add-content $report "</table>"

add-content $report "<table width='100%'"
Add-Content $report "<tr bgcolor='IndianRed'"
Add-Content $report "<td width='5%' align='center'><B>Identity</B></td>"
Add-Content $report "<td width='10%' align='center'><B>PingStatus</B></td>"
Add-Content $report "<td width='10%' align='center'><B>NetlogonService</B></td>"

```

```

Add-Content $report "<td width='10%' align='center'><B>NTDSService</B></td>"
Add-Content $report "<td width='10%' align='center'><B>DNSServiceStatus</B></td>"
Add-Content $report "<td width='10%' align='center'><B>NetlogonsTest</B></td>"
Add-Content $report "<td width='10%' align='center'><B>ReplicationTest</B></td>"
Add-Content $report "<td width='10%' align='center'><B>ServicesTest</B></td>"
Add-Content $report "<td width='10%' align='center'><B>AdvertisingTest</B></td>"
Add-Content $report "<td width='10%' align='center'><B>FSMOCheckTest</B></td>"

Add-Content $report "</tr>"

#####Ping Test#####
foreach ($DC in $DCServers){
$Identity = $DC

    Add-Content $report "<tr>"
if ( Test-Connection -ComputerName $DC -Count 1 -ErrorAction SilentlyContinue ) {
Write-Host $DC `t $DC `t Ping Success -ForegroundColor Green

    Add-Content $report "<td bgcolor= 'GainsBoro' align=center> <B> $Identity</B></td>"
    Add-Content $report "<td bgcolor= 'Aquamarine' align=center>
<B>Success</B></td>"

    #####Netlogon Service Status#####
    $serviceStatus = start-job -scriptblock {get-service -ComputerName $($args[0]) -Name
"Netlogon" -ErrorAction SilentlyContinue} -ArgumentList $DC
    wait-job $serviceStatus -timeout $timeout
    if($serviceStatus.state -like "Running")
    {
        Write-Host $DC `t Netlogon Service TimeOut -ForegroundColor Yellow
        Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>NetlogonTimeout</B></td>"
        stop-job $serviceStatus
    }
    else
    {
        $serviceStatus1 = Receive-job $serviceStatus
        if ($serviceStatus1.status -eq "Running") {
Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Green

        $svcName = $serviceStatus1.name
        $svcState = $serviceStatus1.status

```

```

        Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>$svcState</B></td>"
    }
    else
    {
        Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Red

        $svcName = $serviceStatus1.name
        $svcState = $serviceStatus1.status
        Add-Content $report "<td bgcolor= 'Red' align=center><B>$svcState</B></td>"
    }
}

#####
#####NTDS Service Status#####

$serviceStatus = start-job -scriptblock {get-service -ComputerName $($args[0]) -Name
"NTDS" -ErrorAction SilentlyContinue} -ArgumentList $DC
    wait-job $serviceStatus -timeout $timeout
    if($serviceStatus.state -like "Running")
    {
        Write-Host $DC `t NTDS Service TimeOut -ForegroundColor Yellow
        Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>NTDSTimeout</B></td>"
        stop-job $serviceStatus
    }
    else
    {
        $serviceStatus1 = Receive-job $serviceStatus
        if ($serviceStatus1.status -eq "Running") {
            Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Green

            $svcName = $serviceStatus1.name
            $svcState = $serviceStatus1.status
            Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>$svcState</B></td>"
        }
        else
        {
            Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Red

            $svcName = $serviceStatus1.name

```

```

$svcState = $serviceStatus1.status
Add-Content $report "<td bgcolor= 'Red' align=center><B>$svcState</B></td>"
    }
}

#####
#####DNS Service Status#####

$serviceStatus = start-job -scriptblock {get-service -ComputerName $($args[0]) -Name
"DNS" -ErrorAction SilentlyContinue} -ArgumentList $DC
    wait-job $serviceStatus -timeout $timeout
    if($serviceStatus.state -like "Running")
    {
        Write-Host $DC `t DNS Server Service TimeOut -ForegroundColor Yellow
        Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>DNSTimeout</B></td>"
        stop-job $serviceStatus
    }
    else
    {
        $serviceStatus1 = Receive-job $serviceStatus
        if ($serviceStatus1.status -eq "Running") {
            Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Green
            $svcName = $serviceStatus1.name
            $svcState = $serviceStatus1.status
            Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>$svcState</B></td>"
        }
        else
        {
            Write-Host $DC `t $serviceStatus1.name `t $serviceStatus1.status -ForegroundColor
Red
            $svcName = $serviceStatus1.name
            $svcState = $serviceStatus1.status
            Add-Content $report "<td bgcolor= 'Red' align=center><B>$svcState</B></td>"
        }
    }

#####
#####Netlogons status#####
add-type -AssemblyName microsoft.visualbasic
$cmp = "microsoft.visualbasic.strings" -as [type]

```

```

        $sysvol = start-job -scriptblock {dcdiag /test:netlogons /s:$($args[0])} -
ArgumentList $DC
        wait-job $sysvol -timeout $timeout
        if($sysvol.state -like "Running")
        {
            Write-Host $DC `t Netlogons Test TimeOut -ForegroundColor Yellow
            Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>NetlogonsTimeout</B></td>"
            stop-job $sysvol
        }
        else
        {
            $sysvol1 = Receive-job $sysvol
            if($cmp::instr($sysvol1, "passed test NetLogons"))
            {
                Write-Host $DC `t Netlogons Test passed -ForegroundColor Green
                Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>NetlogonsPassed</B></td>"
            }
            else
            {
                Write-Host $DC `t Netlogons Test Failed -ForegroundColor Red
                Add-Content $report "<td bgcolor= 'Red'
align=center><B>NetlogonsFail</B></td>"
            }
        }
        }
        #####
        #####Replications status#####
        add-type -AssemblyName microsoft.visualbasic
        $cmp = "microsoft.visualbasic.strings" -as [type]
        $sysvol = start-job -scriptblock {dcdiag /test:Replications /s:$($args[0])} -
ArgumentList $DC
        wait-job $sysvol -timeout $timeout
        if($sysvol.state -like "Running")
        {
            Write-Host $DC `t Replications Test TimeOut -ForegroundColor Yellow
            Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>ReplicationsTimeout</B></td>"
            stop-job $sysvol
        }

```

```

else
{
$sysvol1 = Receive-job $sysvol
if($cmp::instr($sysvol1, "passed test Replications"))
{
Write-Host $DC `t Replications Test passed -ForegroundColor Green
Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>ReplicationsPassed</B></td>"
}
else
{
Write-Host $DC `t Replications Test Failed -ForegroundColor Red
Add-Content $report "<td bgcolor= 'Red'
align=center><B>ReplicationsFail</B></td>"
}
}

#####
#####Services status#####
add-type -AssemblyName microsoft.visualbasic
$cmp = "microsoft.visualbasic.strings" -as [type]
$sysvol = start-job -scriptblock {dcdiag /test:Services /s:${args[0]}} -
ArgumentList $DC
wait-job $sysvol -timeout $timeout
if($sysvol.state -like "Running")
{
Write-Host $DC `t Services Test TimeOut -ForegroundColor Yellow
Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>ServicesTimeout</B></td>"
stop-job $sysvol
}
else
{
$sysvol1 = Receive-job $sysvol
if($cmp::instr($sysvol1, "passed test Services"))
{
Write-Host $DC `t Services Test passed -ForegroundColor Green
Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>ServicesPassed</B></td>"
}
else

```

```

    {
        Write-Host $DC `t Services Test Failed -ForegroundColor Red
        Add-Content $report "<td bgcolor= 'Red'
align=center><B>ServicesFail</B></td>"
    }
}

#####
#####Advertising status#####
add-type -AssemblyName microsoft.visualbasic
$cmp = "microsoft.visualbasic.strings" -as [type]
$sysvol = start-job -scriptblock {dcdiag /test:Advertising /s:${$args[0]}} -
ArgumentList $DC
wait-job $sysvol -timeout $timeout
if($sysvol.state -like "Running")
{
    Write-Host $DC `t Advertising Test TimeOut -ForegroundColor Yellow
    Add-Content $report "<td bgcolor= 'Yellow'
align=center><B>AdvertisingTimeout</B></td>"
    stop-job $sysvol
}
else
{
    $sysvol1 = Receive-job $sysvol
    if($cmp::instr($sysvol1, "passed test Advertising"))
    {
        Write-Host $DC `t Advertising Test passed -ForegroundColor Green
        Add-Content $report "<td bgcolor= 'Aquamarine'
align=center><B>AdvertisingPassed</B></td>"
    }
    else
    {
        Write-Host $DC `t Advertising Test Failed -ForegroundColor Red
        Add-Content $report "<td bgcolor= 'Red'
align=center><B>AdvertisingFail</B></td>"
    }
}

#####
#####FSM0Check status#####
add-type -AssemblyName microsoft.visualbasic
$cmp = "microsoft.visualbasic.strings" -as [type]

```

[illegible]

```

        Add-Content $report "<td bgcolor= 'Red' align=center>  <B>Ping Fail</B></td>"
        Add-Content $report "<td bgcolor= 'Red' align=center>  <B>Ping Fail</B></td>"
    }

}

Add-Content $report "</tr>"
#####Close HTML Tables#####
Add-content $report  "</table>"
Add-Content $report "</body>"
Add-Content $report "</html>"

#####
#####Send Email#####

if(($Smtphost) -and ($EmailReport) -and ($from)){
[string]$body = Get-Content $report
Send-MailMessage -SmtpServer $Smtphost -port $SMTPPort -UseSsl ` -Attachments $Attachment -
From $from -To $EmailReport -Subject "Active Directory Health Monitor for $($Env:COMPUTERNAME)
at $(Get-Date -Format g)" -Body $body -BodyAsHtml
}

#####EnD#####
#####

```

Revision #1

Created 2023-11-10 04:41:58 UTC by Ryan

Updated 2025-03-12 15:12:20 UTC by Ryan