

# Powershell Scripts

- [Active Directory](#)
  - [Edit Global Proxy Settings](#)
- [Office 365](#)
  - [Safe Sender Domains](#)
- [SQL](#)
  - [SQL enable TCP](#)
- [Install Scripts](#)
  - [Rustdesk Silent Installer](#)

# Active Directory

# Edit Global Proxy Settings

```
#tombehrendt
# 08/03/2024

$UserProfiles = Get-ItemProperty "HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\ProfileList\*" |

    Where {$_.PSChildName -match "S-1-5-21-(\d+-?){4}$" } |

    Select-Object @{Name="SID"; Expression={$_.PSChildName}},
@{Name="UserHive";Expression={"$(($_.ProfileImagePath)\NTuser.dat)"}

foreach ($UserProfile in $UserProfiles) {

    # Load User ntuser.dat if it's not already loaded

    if (($ProfileWasLoaded = Test-Path "Registry::HKEY_USERS\$(($UserProfile.SID)") -eq $false)
    {

        Start-Process -FilePath "CMD.EXE" -ArgumentList "/C REG.EXE LOAD
HKU\$(($UserProfile.SID) $(($UserProfile.UserHive))" -Wait -WindowStyle Hidden

    }

    # Check if the user profile was loaded successfully before accessing the registry path

    if ($ProfileWasLoaded) {

        $key =
"Registry::HKEY_USERS\$(($UserProfile.SID)\Software\Microsoft\Windows\CurrentVersion\Internet
Settings"

        New-ItemProperty -Path $key -Name "ProxyServer" -Value "101.137.122.232:3129" -
PropertyType String -Force | Out-Null
```

```
New-ItemProperty -Path $key -Name "ProxyOverride" -Value "*.local;<local>" -
PropertyType String -Force | Out-Null

New-ItemProperty -Path $key -Name "ProxyEnable" -Value 1 -PropertyType DWord -Force |
Out-Null

$key =
"Registry::HKEY_USERS\$(($UserProfile.SID)\Software\Microsoft\Windows\CurrentVersion\Internet
Settings\UserInfo"

New-Item -Path $key -Force | Out-Null

New-ItemProperty -Path $key -Name "LoginName" -Value
"$($ENV:USERDOMAIN)\$(($ENV:USERNAME))" -PropertyType String -Force | Out-Null

# Unload NTuser.dat

if ($ProfileWasLoaded -eq $false) {

    [GC]::Collect()

    Start-Sleep 1

    Start-Process -FilePath "CMD.EXE" -ArgumentList "/C REG.EXE UNLOAD
HKU\$(($UserProfile.SID))" -Wait -WindowStyle Hidden | Out-Null

}

}

}
```

# Office 365

# Safe Sender Domains

Vorher per Powershell Verbindung zum Exchange Server aufbauen! Danach Powershell Script ausführen und "domainlist.txt" anpassen / erstellen.

```
# domainlist.txt anlegen und Domains hinterlegen!

# NAME: Add365SafeDomains.ps1

Param(
    [Parameter(Mandatory=$True,Position=1)]
    [string]$ruleName,

    [Parameter(Mandatory=$True)]
    [string]$domainListFilePath
)

#Read the contents of the text file into an array
$safeDomainList = Get-Content $domainListFilePath

#Create a new array and remove all text for each line up to and including the @ symbol, also
remove whitespace
$newSafeDomainList = @()
$newSafeDomainList += foreach ($domain in $safeDomainList)
{
    $tmpdomain = $domain -replace ".*@"
    $tmpdomain.trim()
}
```

```
#If the rule already exists update the existing allowed sender domains, else create a new
rule.
if (Get-TransportRule $ruleName -EA SilentlyContinue)
{
"Updating existing rule..."
$safeDomainList = Get-TransportRule $ruleName |select -ExpandProperty SenderDomainIs
$completeList = $safeDomainList + $newSafeDomainList
$completeList = $completeList | select -uniq | sort
set-TransportRule $ruleName -SenderDomainIs $completeList
}
else
{
"Creating new rule..."
$newSafeDomainList = $newSafeDomainList | sort
New-TransportRule $ruleName -SenderDomainIs $newSafeDomainList -SetSCL "-1"
}
```

# SQL

# SQL enable TCP

```
[System.Reflection.Assembly]::LoadWithPartialName('Microsoft.SqlServer.SqlWmiManagement')

# Instantiate a ManagedComputer object which exposes primitives to control the
# installation of SQL Server on this machine.

$wmi = New-Object 'Microsoft.SqlServer.Management.Smo.Wmi.ManagedComputer' localhost

# Enable the TCP protocol on the default instance. If the instance is named,
# replace MSSQLSERVER with the instance name in the following line.

$tcp = $wmi.ServerInstances['SQLINSTANZ'].ServerProtocols['Tcp']
$tcp.IsEnabled = $true
$tcp.Alter()

# You need to restart SQL Server for the change to persist
# -Force takes care of any dependent services, like SQL Agent.
# Note: if the instance is named, replace MSSQLSERVER with MSSQL$ followed by
# the name of the instance (e.g. MSSQL$MYINSTANCE)

Restart-Service -Name MSSQL$SQLINSTANZ -Force
```

Der Port kann hier ausgelesen werden in REGEDIT:

```
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\SQLEXPRESS\MSSQLServer\SuperSocketNetLib\Tcp
```

# Install Scripts

# Rustdesk Silent Installer

```
# Set error handling
$ErrorActionPreference = 'Stop'

# Generate random password
$rustdesk_pw = -join ((65..90) + (97..122) + (48..57) | Get-Random -Count 12 | % {[char]$_})

# Check admin rights
if (-Not
([Security.Principal.WindowsPrincipal][Security.Principal.WindowsIdentity]::GetCurrent()).IsIn
Role([Security.Principal.WindowsBuiltInRole]::Administrator)) {
    Start-Process PowerShell -Verb RunAs "-NoProfile -ExecutionPolicy Bypass -Command `cd
'$pwd'; & '$PSCommandPath'`""
    Exit
}

# Check if already installed
$installDir = "$env:ProgramFiles\RustDesk"
if (Test-Path "$installDir\rustdesk.exe") {
    Write-Output "❑ RustDesk already installed"
    # exit 0 # Uncomment to skip reinstallation
}

# Temp directory setup
if (!(Test-Path C:\Temp)) { New-Item -Path C:\Temp -ItemType Directory -Force | Out-Null }
Set-Location C:\Temp

# Download (if needed)
if (!(Test-Path .\rustdesk.exe)) {
    Write-Output "❑ Downloading RustDesk..."
    Invoke-WebRequest "https://github.com/rustdesk/rustdesk/releases/download/1.2.3-
2/rustdesk-1.2.3-2-x86_64.exe" -OutFile rustdesk.exe -UseBasicParsing
}

# Installation with timeout
```

```
Write-Output "☐☐Installing RustDesk (max 30 seconds)..."
$installJob = Start-Job -ScriptBlock {
    Start-Process $args[0] -ArgumentList "--silent-install" -Wait -NoNewWindow
}
```