

Windows Stuff

- [Windows 10 / Windows 11](#)
 - [Reset Windows DNS Server](#)
- [Windows Deployment Services](#)
 - [Admin Benutzer erstellen mit verschlüsseltem Kennwort](#)
- [Windows Server](#)
 - [Massenprüfpunkte entfernen](#)

Windows 10 / Windows 11

Reset Windows DNS Server

Unter folgendem Pfad findet Ihr

```
Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces
```

den Key "ProfileNameServer" dort ist dann fälscherlicherweise eine IP Adresse hinterlegt. Dort am besten einfach den Wert rauslöschen, den Key selber aber bestehen lassen ! Mit einem

```
ipconfig /renew
```

sollte er dann wieder die richtige IP Adresse nutzen die vom DHCP zugeteilt wurde.

Windows Deployment Services

Admin Benutzer erstellen mit verschlüsseltem Kennwort

@ Tom Behrendt | tom@tombehrendt.de

Info: Um sicherzustellen, dass das verschlüsselte Passwort auf einem anderen Computer entschlüsselt und verwendet werden kann, müssen wir einen kryptografischen Schlüssel (einen "Key") verwenden, um das Passwort zu verschlüsseln und zu entschlüsseln. Dieser Schlüssel muss auf beiden Computern vorhanden sein.

1. Starte "2_create-key.ps1", um einen private Key und das verschlüsselte Kennwort für den User zu erstellen
2. Bearbeite die "3_setpasswd.bat" -> passe ggf. die Pfade vom Key und dem verschlüsselten Kennwort (EncryptedKeys.txt) an, falls du diese woanders abgespeichert hast. Passe bitte auch den Benutzernamen an!
3. Kopiere nun das Keyfile, die EncryptedKeys.txt sowie die 3_setpasswd.bat auf einen Client, wo ein User erstellt werden soll
4. Starte nun die "3_setpasswd.bat" als Admin!
Es wird unter C:\ ein Log erstellt.

Sollte kein Output gewünscht sein (drücken sie eine beliebige Taste zum Fortfahren...) entferne in der vorletzten Zeile der "3_setpasswd.bat" das Wort Pause (empfehlenswert bei z.B. Intune)

Der User wurde nun erstellt, das Kennwort läuft nie ab und er ist lokaler Administrator

2_create_key.ps1

```
# created by Tom Behrendt | tom@tombehrendt.de

# ASCII-Art anzeigen
$asciiArt = @"
```

```

ad88888888888ba
dP'      `8b,
8 ,aaa,      "Y888a      ,aaaa,      ,aaa,      ,aa,
8 8' `8      "88baadP""""YbaaadP""""YbdP""Yb
8 8 8      """"      """"      ""      8b
8 8, ,8      ,aaaaaaaaaaaaaaaaaaaaaaaaadddd88P
8 `""""      ,d8""
Yb,      ,ad8"      Tom Behrendt | tom@tombehrendt.de
"Y88888888888P"

```

"@

Write-Output \$asciiArt

Generiere einen zufälligen Schlüssel

\$Key = New-Object Byte[] 32

[Security.Cryptography.RNGCryptographyServiceProvider]::Create().GetBytes(\$Key)

Konvertiere den Schlüssel zu Base64 für die Speicherung

\$KeyBase64 = [Convert]::ToBase64String(\$Key)

Speichere den Base64-kodierten Schlüssel in einer Datei

\$KeyBase64 | Out-File -FilePath "secret.key"

Write-Output "Schlüssel wurde erfolgreich gespeichert."

\$passMatch = \$false

while (-not \$passMatch) {

Passwort eingeben und verschlüsseln

\$Password1 = Read-Host "Geben Sie das Passwort ein" -AsSecureString

\$Password2 = Read-Host "Geben Sie das Passwort erneut ein" -AsSecureString

Überprüfen, ob beide Passwörter übereinstimmen

\$Password1Plain =

[System.Runtime.InteropServices.Marshal]::PtrToStringAuto([System.Runtime.InteropServices.Marshal]::SecureStringToBSTR(\$Password1))

\$Password2Plain =

[System.Runtime.InteropServices.Marshal]::PtrToStringAuto([System.Runtime.InteropServices.Marshal]::SecureStringToBSTR(\$Password2))

```

hal)::SecureStringToBSTR($Password2))

if ($Password1Plain -eq $Password2Plain) {
    Write-Output "Passwörter stimmen überein. Verschlüsselung wird durchgeführt."
    $passMatch = $true

    # Passwort verschlüsseln
    $EncryptedPassword = ConvertFrom-SecureString -SecureString $Password1 -Key $Key
    $EncryptedPassword | Out-File -FilePath "EncryptedPassword.txt"
    Write-Output "Verschlüsseltes Passwort wurde erfolgreich gespeichert."
} else {
    Write-Output "Passwörter stimmen nicht überein. Bitte versuchen Sie es erneut."
    Start-Sleep -Seconds 3
}
}

# Wartet auf Benutzereingabe bevor das Fenster geschlossen wird
Read-Host -Prompt "Drücken Sie die Eingabetaste, um das Fenster zu schließen"

```

3_setpwd.bat

```

@echo off
setlocal

REM Speichern des aktuellen Verzeichnisses
set "currentDir=%~dp0"

REM Hostname abrufen
for /f "tokens=*" %%i in ('hostname') do set "hostname=%%i"

REM Überprüfen, ob das Skript mit administrativen Rechten ausgeführt wird
net session >nul 2>&1
if not %errorlevel% equ 0 (
echo #####
echo !
    echo Das Skript muss als Administrator ausgeführt werden!
echo !
echo #####
    pause
    exit /b
)

```

```

)

REM Name des neuen Benutzers
set "username=myAdmin"

REM Pfade zu den Schlüssel- und Passwort-Dateien (absolute Pfade verwenden)
set "keyFile=%currentDir%secret.key"
set "encryptedPasswordFile=%currentDir%EncryptedPassword.txt"

REM Überprüfen, ob die Dateien existieren
if not exist "%keyFile%" (
    echo #####
    echo !
    echo Fehler: Die Datei "%keyFile%" wurde nicht gefunden!
    echo !
    echo #####
    pause
    exit /b
)

if not exist "%encryptedPasswordFile%" (
    echo #####
    echo !
    echo Fehler: Die Datei "%encryptedPasswordFile%" wurde nicht gefunden!
    echo !
    echo #####
    pause
    exit /b
)

REM Log-Datei
set "logfile=C:\createUser-%hostname%.log"

REM Temporäre PowerShell-Datei erstellen
set "psfile=%temp%\create_user.ps1"

REM PowerShell-Skript zum Entschlüsseln des Passworts und Erstellen des Benutzers
>"%psfile%" echo Set-Location "%currentDir%"
>>"%psfile%" echo $KeyBase64 = Get-Content "%keyFile%"
>>"%psfile%" echo $Key = [Convert]::FromBase64String($KeyBase64)

```

```
>>"%psfile%" echo $EncryptedPassword = Get-Content "%encryptedPasswordFile%"
>>"%psfile%" echo $SecurePassword = ConvertTo-SecureString -String $EncryptedPassword -Key
$Key
>>"%psfile%" echo New-LocalUser -Name "%username%" -Password $SecurePassword -
PasswordNeverExpires
>>"%psfile%" echo Add-LocalGroupMember -Group "Administratoren" -Member "%username%"
```

REM PowerShell-Skript ausführen und Ausgabe umleiten

```
powershell -ExecutionPolicy Bypass -File "%psfile%" > "%logfile%" 2>&1
type "%logfile%"
```

REM Temporäre PowerShell-Datei löschen

```
del "%psfile%"
```

```
echo.
```

```
echo #####
```

```
echo Benutzer %username% wurde erfolgreich erstellt, das Kennwort gesetzt und zur
Administratorengruppe hinzugefügt.
```

```
echo Die Ausführungsergebnisse wurden in der Datei %logfile% gespeichert.
```

```
echo.
```

```
echo #####
```

```
pause
```

```
endlocal
```

Windows Server

Massenprüfpunkte entfernen

Für die Entfernung können Sie die Powershell nutzen. In dem Fall, dass Sie die Daten der Checkpoints (auch Snapshots genannt) einsehen wollen, überprüfen Sie das Vorhandensein mit

```
get-vm | Get-VMSnapshot
```

Zur Entfernung einfach den Befehl:

```
get-vm | Get-VMSnapshot | Remove-VMSnapshot
```

nutzen. Die Zusammenführung kann im laufendem Betrieb erfolgen, ein Shutdown der virtuellen Maschine ist nicht notwendig.