



Anmeldeinformationsverwaltung Powershell

Mit dem Modul Credential Manager haben wir Zugriff auf die Anmeldeinformationsverwaltung. Zum installieren des Moduls setzen wir den Befehl

Install-Module -Name "CredentialManager" ab.

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt3.ps1* X
1 Install-Module -Name "CredentialManager"
2 Get-Command -Module "CredentialManager"
3

PS C:\WINDOWS\system32> Install-Module -Name "CredentialManager"

This module is not from a trusted repository. Do you want to install this module?

Ja Ja, alle Nein Nein, keine Anhalten
```

Mit **Get-Command -Module "CredentialManager"** bekommen wir weitere CmdLets angezeigt.

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt3.ps1* X
1 Install-Module -Name "CredentialManager"
2 Get-Command -Module "CredentialManager"
3

PS C:\WINDOWS\system32> Install-Module -Name "CredentialManager"
PS C:\WINDOWS\system32> Get-Command -Module "CredentialManager"

CommandType Name Version Source
-----
Cmdlet Get-StoredCredential 2.0 CredentialManager
Cmdlet Get-StrongPassword 2.0 CredentialManager
Cmdlet New-StoredCredential 2.0 CredentialManager
Cmdlet Remove-StoredCredential 2.0 CredentialManager

PS C:\WINDOWS\system32>
```



Anmeldeinformationsverwaltung Powershell

Dieser Block speichert Anmeldeinformationen lokal ab.

```
$Target = "DC01"
```

```
$UserName = "NDS"
```

```
$Secure = Read-host -AsSecureString
```

```
New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure  
-Persist LocalMachine
```

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt3.ps1* X
1 Install-Module -Name "CredentialManager"
2 Get-Command -Module "CredentialManager"
3
4 $Target = "DC01"
5 $UserName = "Administrator"
6 $Secure = Read-host -AsSecureString
7 New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine
8
9

PS C:\WINDOWS\system32> $Target = "DC01"
$UserName = "Administrator"
$Secure = Read-host -AsSecureString
New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine

Windows PowerShell ISE - Eingabe

OK Abbrechen

Skript/Auswahl wird ausgeführt. Drücken Sie "Strg+Unterbrechen", um den Vorgang zu beenden, und "", um den Debugger zu öffnen. | Ln 4 Spalte 1 | 100%
```

Die Daten stehen ab sofort zur weiteren Nutzung zur Verfügung.

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt3.ps1* X
1 Install-Module -Name "CredentialManager"
2 Get-Command -Module "CredentialManager"
3
4 $Target = "DC01"
5 $UserName = "Administrator"
6 $Secure = Read-host -AsSecureString
7 New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine
8
9

PS C:\WINDOWS\system32> $Target = "DC01"
$UserName = "Administrator"
$Secure = Read-host -AsSecureString
New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine

Flags           : 0
Type            : Generic
TargetName      : DC01
Comment        : Updated by: Joern on: 28.02.2018
LastWritten     : 28.02.2018 20:30:00 Uhr
PasswordSize    : 18
Password       : P0ssw0rd!
Persist         : LocalMachine
AttributeCount  : 0
Attributes      : 0
TargetAlias     : 
UserName       : Administrator

PS C:\WINDOWS\system32>

Abgeschlossen | Ln 4 Spalte 1 | 100%
```



Anmeldeinformationsverwaltung Powershell

Wir öffnen die Anmeldeinformationsverwaltung und sehen dort den neuen Eintrag.

Startseite der Systemsteuerung

Eigene Anmeldeinformationen verwalten

Sie können gespeicherte Anmeldeinformationen für Websites, verbundene Anwendungen und Netzwerke anzeigen und löschen.

[Anmeldedaten sichern](#) [Anmeldedaten wiederherstellen](#)

Windows-Anmeldeinformationen [Windows-Anmeldeinformationen hinzufügen](#)

Es sind keine Windows-Anmeldeinformationen vorhanden.

Zertifikatbasierte Anmeldeinformationen [Zertifikatbasierte Anmeldeinformationen hinzufügen](#)

Es sind keine Zertifikate vorhanden.

Generische Anmeldeinformationen [Generische Anmeldeinformationen hinzufügen](#)

DC01	Geändert: Heute
Microsoft_PaidConnectivity_WiFi_AccessToken	Geändert: 04.11.2017
MicrosoftAccount:user=joernwalter@outlook.de	Geändert: Heute
MicrosoftOffice16_Data:live:cid=31b4e2385bc4734b	Geändert: Heute
OneDrive Cached Credential	Geändert: Heute
virtualapp/didlogical	Geändert: 14.02.2018
WindowsLive:(token):name=joernwalter@outlook.de;s...	Geändert: 03.03.2017
XboxLive	Geändert: Heute

Andere Elemente

SSO_POP_User	Geändert: Heute
SSO_POP_Device	Geändert: Heute

Siehe auch [Benutzerkonten](#)

Weitere Details zum Eintrag DC01.

Startseite der Systemsteuerung

Eigene Anmeldeinformationen verwalten

Sie können gespeicherte Anmeldeinformationen für Websites, verbundene Anwendungen und Netzwerke anzeigen und löschen.

[Anmeldedaten sichern](#) [Anmeldedaten wiederherstellen](#)

Windows-Anmeldeinformationen [Windows-Anmeldeinformationen hinzufügen](#)

Es sind keine Windows-Anmeldeinformationen vorhanden.

Zertifikatbasierte Anmeldeinformationen [Zertifikatbasierte Anmeldeinformationen hinzufügen](#)

Es sind keine Zertifikate vorhanden.

Generische Anmeldeinformationen [Generische Anmeldeinformationen hinzufügen](#)

DC01	Geändert: Heute
Internet- oder Netzwerkadresse: DC01	
Benutzername: Administrator	
Kennwort:	
Dauerhaftigkeit: Lokaler Computer	
Bearbeiten Entfernen	
Microsoft_PaidConnectivity_WiFi_AccessToken	Geändert: 04.11.2017
MicrosoftAccount:user=joernwalter@outlook.de	Geändert: Heute
MicrosoftOffice16_Data:live:cid=31b4e2385bc4734b	Geändert: Heute
OneDrive Cached Credential	Geändert: Heute
virtualapp/didlogical	Geändert: 14.02.2018
WindowsLive:(token):name=joernwalter@outlook.de;s...	Geändert: 03.03.2017
XboxLive	Geändert: Heute

Siehe auch [Benutzerkonten](#)



Anmeldeinformationsverwaltung Powershell

Mit `Get-StoredCredential -Target "DC01" -AsCredentialObject` bekommen wir einen Einblick in die hinterlegten Anmeldeinformationen.

```
Administrator: Windows PowerShell ISE (x86)
Unbenannt3.ps1 *
1 Install-Module -Name "CredentialManager"
2 Get-Command -Module "CredentialManager"
3
4 $Target = "DC01"
5 $UserName = "Administrator"
6 $Secure = Read-Host -AsSecureString
7 New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine
8
9 Get-StoredCredential -Target "DC01" -AsCredentialObject
10

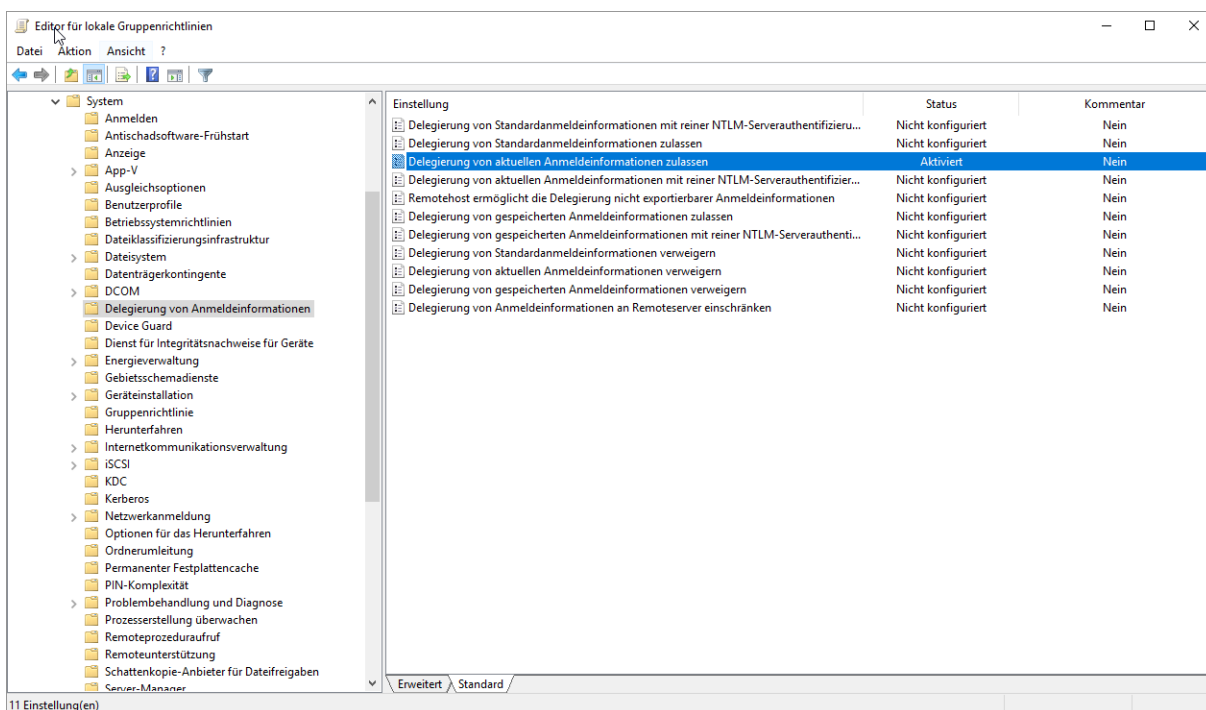
PS C:\WINDOWS\system32> Get-StoredCredential -Target "DC01" -AsCredentialObject

Flags           : 0
Type            : Generic
TargetName      : DC01
Comment        : Updated by: Joern on: 28.02.2018
LastWritten     : 28.02.2018 20:22:51 Uhr
PasswordSize    : 18
Password       : PQsw0rd!
Persist        : LocalMachine
AttributeCount  : 0
Attributes      : 0
TargetAlias     :
UserName       : Administrator

PS C:\WINDOWS\system32>
```

Abgeschlossen | Ln 9 Spalte 1 | 100%

Zur Nutzung der hinterlegten Anmeldeinformationen (PS-Session) muss die Delegierung aktiviert werden.





Anmeldeinformationsverwaltung Powershell

Zum Löschen eines Eintrags in der Anmeldeinformationsverwaltung setzen wir diesen Befehl ab **Remove-StoredCredential -Target "DC01"**

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Unbenannt3.ps1* Anmeldeinformationsverwaltung Powershell.ps1 X
7 New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine
8
9 Get-StoredCredential -Target "DC01" -AsCredentialObject
10 Enter-PSSession -ComputerName "DC01" -Authentication Credssp -Credential (Get-StoredCredential -Target "DC01")
11 Remove-StoredCredential -Target "DC01"
12
13 winrm get winrm/config
14 winrm set winrm/config/client/auth '@{CredSSP="true"}'
15 winrm enumerate winrm/config/listener
16 winrm set winrm/config/client @{TrustedHosts="DC01"}
17 winrs -r:ServerName cmd.exe

PS C:\WINDOWS\system32> Remove-StoredCredential -Target "DC01"
PS C:\WINDOWS\system32>

Abgeschlossen | Ln 11 Spalte 1 | 100%
```

```
# Install
Install-Module -Name "CredentialManager"
Get-Command -Module "CredentialManager"

# Generische Anmeldeinformation hinterlegen
$Target = "webmailer"
$UserName = "mail@joernwalter.de"
$Secure = Read-Host -AsSecureString
New-StoredCredential -Target $Target -UserName $UserName -SecurePassword $Secure -Persist LocalMachine

# Anmeldeinformation abfragen
Get-StoredCredential -Target "DC01" -AsCredentialObject
Enter-PSSession -ComputerName "DC01" -Authentication Credssp -Credential (Get-StoredCredential -Target "DC01")

# Anmeldeinformation löschen
Remove-StoredCredential -Target "DC01"

# windows Anmeldung hinzufügen
cmdkey /list
cmdkey /list:targetname
cmdkey /add:webmail /user:mail@joernwalter.de /pass:MeinPassword

# Generische Anmeldung hinzufügen
cmdkey /generic:targetname /user:username /pass:password

# Generische Anmeldung löschen
cmdkey /delete:targetname

# An Sharepoint anmelden
Connect-PnPOnline -Url https://sharepoint.meinService.de -Credentials webmailer

# WinRM Konfiguration
winrm get winrm/config
winrm set winrm/config/client/auth '@{CredSSP="true"}'
winrm enumerate winrm/config/listener
winrm set winrm/config/client @{TrustedHosts="DC01"}
winrs -r:ServerName cmd.exe
```



Anmeldeinformationsverwaltung Powershell

Eine Abfrage aller im Speicher hinterlegten Ressourcen erreichen wir mit diesem Befehl.

```
[Windows.Security.Credentials.PasswordVault,Windows.Security.Credentials,ContentType=WindowsRuntime]
```

```
(new-object Windows.Security.Credentials.PasswordVault).RetrieveAll() | % {  
$_.RetrievePassword(); $_ }
```

The screenshot shows the Windows PowerShell ISE interface. The command prompt is at the top, and the command has been executed. The output is a table of password vault entries.

IsPublic	IsSerial	Name	BaseType
True	False	PasswordVault	System.Runtime.InteropServices.WindowsRuntime.RuntimeClass

UserName	:	Administrator	
Resource	:	Administrator	
Password	:	Administrator	
Properties	:	Administrator	
UserName	:	Administrator	
Resource	:	Administrator	
Password	:	Administrator	
Properties	:	Administrator	
UserName	:	Administrator	
Resource	:	Administrator	
Password	:	Administrator	
Properties	:	Administrator	
UserName	:	Administrator	
Resource	:	Administrator	
Password	:	Administrator	
Properties	:	Administrator	
UserName	:	Administrator	
Resource	:	Administrator	
Password	:	Administrator	
Properties	:	Administrator	



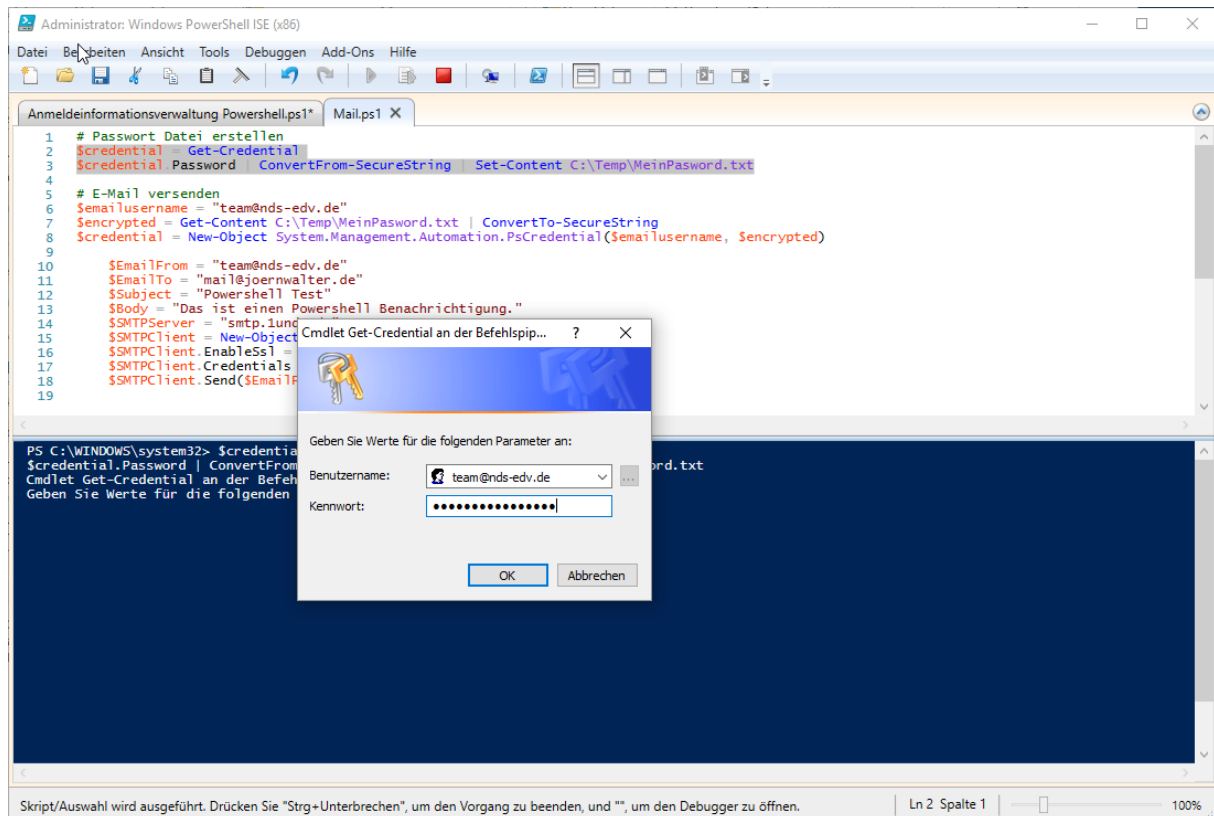
Anmeldeinformationsverwaltung Powershell

Über die Powershell eine E-Mail versenden. Die Authentifizierung passiert über eine angelegte Passwortdatei.

```
# Passwort Datei erstellen
$credential = Get-Credential
$credential.Password | ConvertFrom-SecureString | Set-Content
C:\Temp\MeinPasword.txt

# E-Mail versenden
$emailusername = "team@nds-edv.de"
$encrypted = Get-Content C:\Temp\MeinPasword.txt | ConvertTo-SecureString
$credential = New-Object System.Management.Automation.PsCredential($emailusername,
$encrypted)

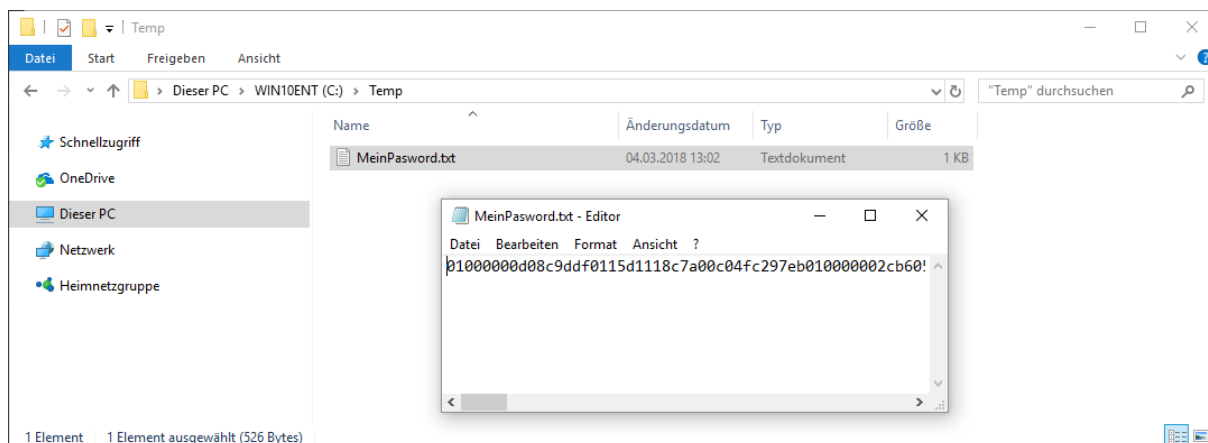
$emailFrom = "team@nds-edv.de"
$emailTo = "mail@joernwalter.de"
$subject = "Powershell Test"
$body = "Das ist einen Powershell Benachrichtigung."
$smtpServer = "smtp.lund1.de"
$smtpClient = New-Object Net.Mail.SmtpClient($smtpServer, 587)
$smtpClient.EnableSsl = $true
$smtpClient.Credentials = $credential;
$smtpClient.Send($emailFrom, $emailTo, $subject, $body)
```



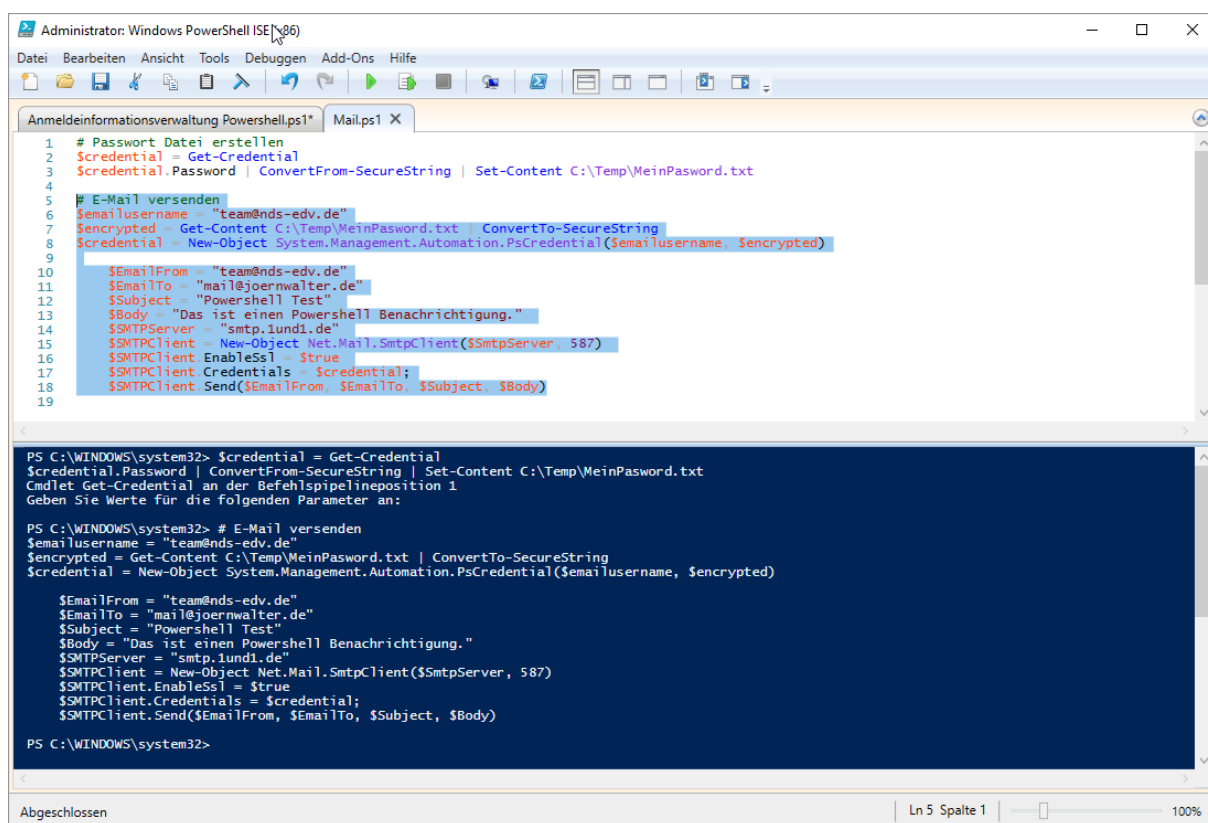


Anmeldeinformationsverwaltung Powershell

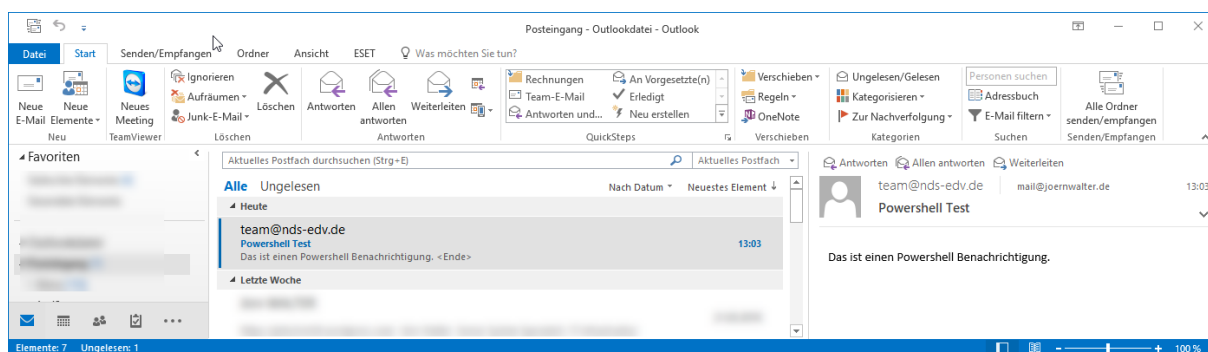
Einsicht in den Secure String.



Skript zum E-Mail Versand:



Voila!





Anmeldeinformationsverwaltung Powershell

Und nun unter Zuhilfenahme der gespeicherten Anmeldeinformationen.

```
# Generische Anmeldeinformation hinterlegen
$Target = "webmailer"
$Username = "team@nds-edv.de"
$Secure = Read-Host -AsSecureString
New-StoredCredential -Target $Target -Username $Username -SecurePassword $Secure -
Persist LocalMachine
```

```
# E-Mail versenden
$emailusername = "team@nds-edv.de"
$credential = (Get-StoredCredential -Target "webmailer")
$emailfrom = "team@nds-edv.de"
$emailto = "mail@joernwalter.de"
$subject = "Powershell Test"
$body = "Das ist eine Powershell Benachrichtigung."
$smtpserver = "smtp.lund1.de"
$smtpclient = New-Object Net.Mail.SmtpClient($smtpserver, 587)
$smtpclient.EnableSsl = $true
$smtpclient.Credentials = $credential;
$smtpclient.Send($emailfrom, $emailto, $subject, $body)
```

```
Administrator: Windows PowerShell ISE (x86)
Datei Bearbeiten Ansicht Tools Debuggen Add-Ons Hilfe

Mail.ps1* X
1 # E-Mail versenden
2 $emailusername = "team@nds-edv.de"
3 $credential = (Get-StoredCredential -Target "webmailer")
4 $emailfrom = "team@nds-edv.de"
5 $emailto = "mail@joernwalter.de"
6 $subject = "Powershell Test"
7 $body = "Das ist eine Powershell Benachrichtigung."
8 $smtpserver = "smtp.lund1.de"
9 $smtpclient = New-Object Net.Mail.SmtpClient($smtpserver, 587)
10 $smtpclient.EnableSsl = $true
11 $smtpclient.Credentials = $credential;
12 $smtpclient.Send($emailfrom, $emailto, $subject, $body)
13

PS C:\WINDOWS\system32> # E-Mail versenden
$emailusername = "team@nds-edv.de"
$credential = (Get-StoredCredential -Target "webmailer")
$emailfrom = "team@nds-edv.de"
$emailto = "mail@joernwalter.de"
$subject = "Powershell Test"
$body = "Das ist eine Powershell Benachrichtigung."
$smtpserver = "smtp.lund1.de"
$smtpclient = New-Object Net.Mail.SmtpClient($smtpserver, 587)
$smtpclient.EnableSsl = $true
$smtpclient.Credentials = $credential;
$smtpclient.Send($emailfrom, $emailto, $subject, $body)

PS C:\WINDOWS\system32>
```