Powershell für Bash-Kenner

Remote auf andere Rechner

Wie benutze ich Powershell um schnell auf ein anderes System zu kommen?

via SSH

ssh user@remotehost

mit persönlichem USER

Windows nimmt nicht einfach Username/Passwort, wäre auch zu einfach.
Ein Credential-Object muss erstellt werden das PW wird dabei interaktiv abgefragt. Das darf dann auch recycled werden

```
$mycred = Get-Credential -UserName "$env:USERDOMAIN\$env:USERNAME" -Message "Bitte PW eingeben"
```

wenn jemand gerne selber tippt:
`$mycred = Get-Credential -UserName "DOMAIN\USER" -Message "Bitte PW eingeben"`

Die beiden variablen $env:USERDOMAIN und $env:USERNAME werden durch Windows gesetzt und sind in jeder Powershell-Sitzung vordefiniert.

Der Login selbst erfolgt über:

`Enter-PSSession -ComputerName SERVERNAME -Credential $mycred`

mit technischem USER

```
$User = "Domain01\Techuser"
$PWord = ConvertTo-SecureString  -AsPlainText -Force -String "P@sSwOrd"
$Credential = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList
$User, $PWord
```

Der Login selbst erfolgt über:

`Enter-PSSession -ComputerName SERVERNAME -Credential $Credential`

In Dateien suchen

grep

```
alias sls='select-string'
sls "something" *.txt
```

tail

```
alias gc='get-content'

# Beispiel: Letzte 15 Zeilen
gc datei.txt -Last 15

# Beispiel: "tail -100f"
gc datei.txt -Last 100 -Wait
```

head

```
alias gc='get-content'

# Beispiel: Die ersten 15 Zeilen
gc datei.txt -First 15
```

Dateien Suchen

find

```
alias gci='Get-Childitem'

# Beispiel: Liste von Dateien erstellen
gci c:\PATH -File -Recurse

# Beispiel: Fehlerunterdrücken bei Berechtigungsfehlermeldungen (Powershell als admin starten ist an der Stelle meist von Vorteil)
gci c:\PATH -File -Recurse -ea SilentlyContinue
```

locate

Gibt es (bisher) nicht

Laufwerke anzeigen

df

```
get-psdrive
```

Mit freundlicher Genehmigung von Claas Biering

Revision #3
Created 2021-10-05 11:36:37 UTC by magenbrot
Updated 2021-10-05 11:46:50 UTC by magenbrot