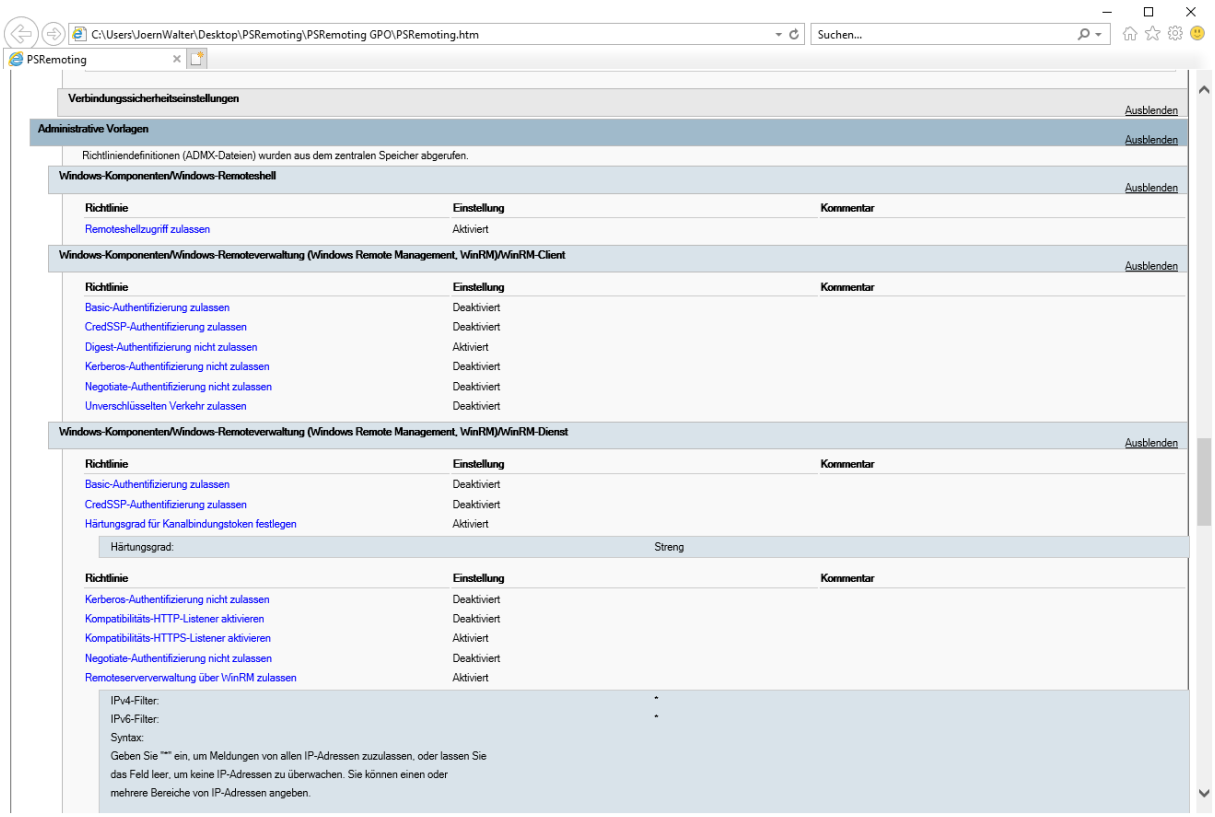
PSRemoting aktivieren

Das Ziel ist es, alle Windows Computer aus der Ferne sicher steuern zu können. Zum Einsatz kommt hier das Protokoll (Web Services for Management) WSMan. Es ist sicher und einfach anzuwenden und wir bauen bei der Authentifizierung auf die Nutzung von TLS-Zertifikaten. Bevor man loslegen kann, muss erst einmal die Remote Verwaltung WINRM (Windows Remote Management) konfiguriert werden.

Über ein Gruppenrichtlinienobjekt nehmen wir die lokale Konfiguration der Systeme vor.

- Authentifizierung
- Firewall
- Services
- Listener

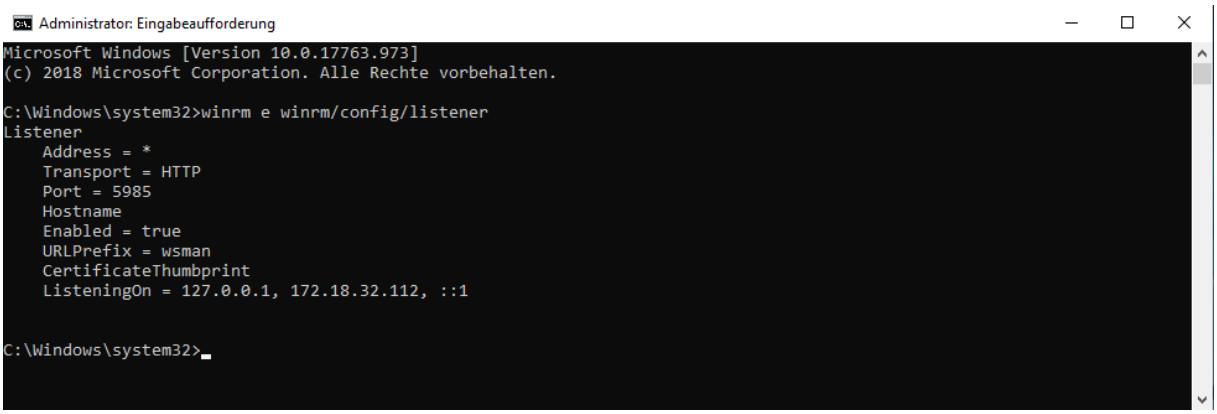


Darüber hinaus verteilen wir über die interne Zertifizierungsstelle Zertifikate an jedes System.

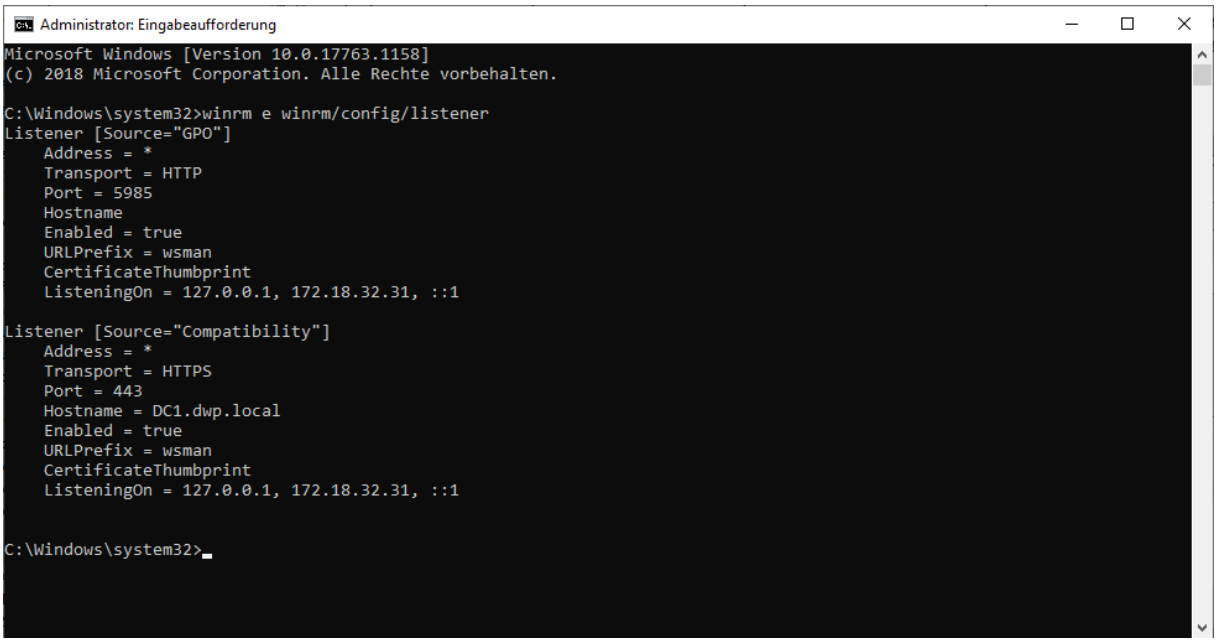
Die Umsetzung erfolgt mittels einer dafür eingerichtete Zertifikatsvorlage und eine separate Richtlinie.

PSRemoting aktivieren

Hier sehen wir die WinRM Standardkonfiguration eines Server Systems.

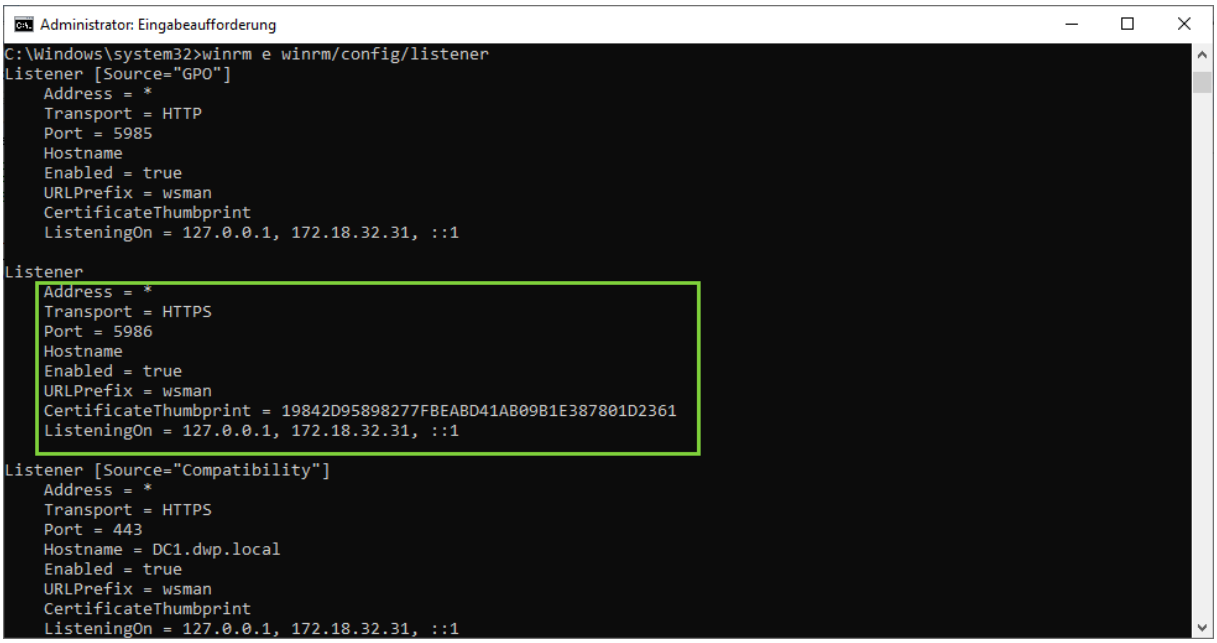


Hier sehen wir, das Einfluss auf die Konfiguration mittels einer Richtlinie genommen wurde.



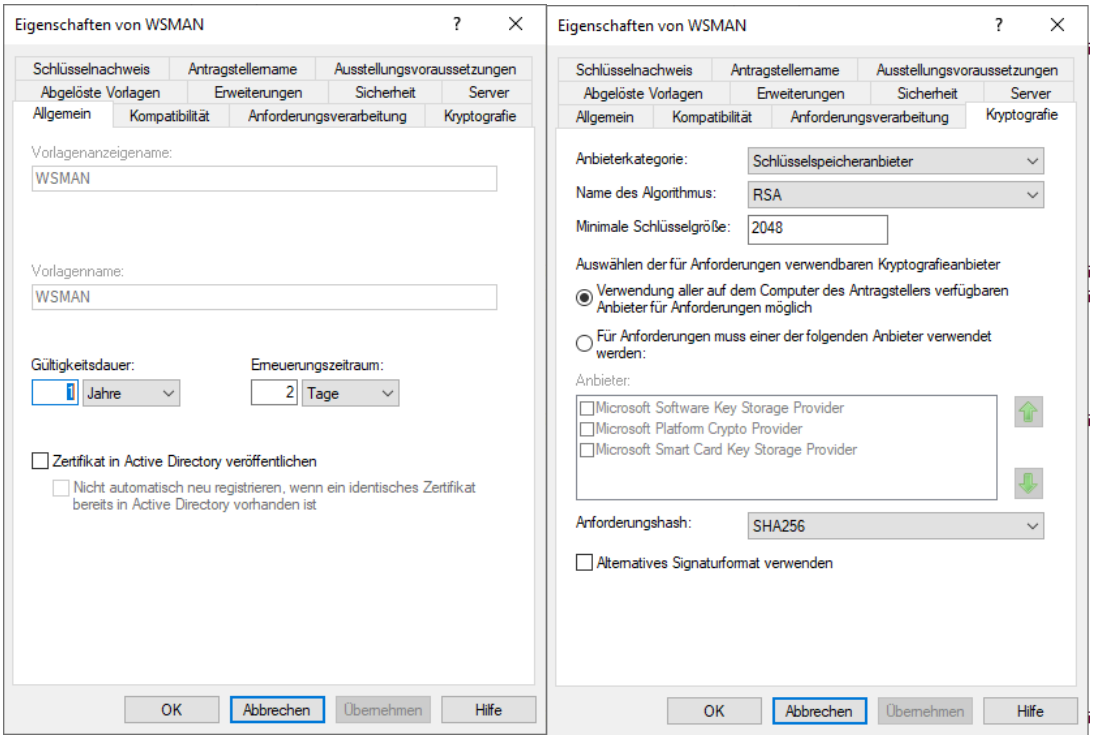
PSRemoting aktivieren

Hier sehen wir eine Konfiguration, die ein Zertifikat zur Authentifizierung einsetzt.

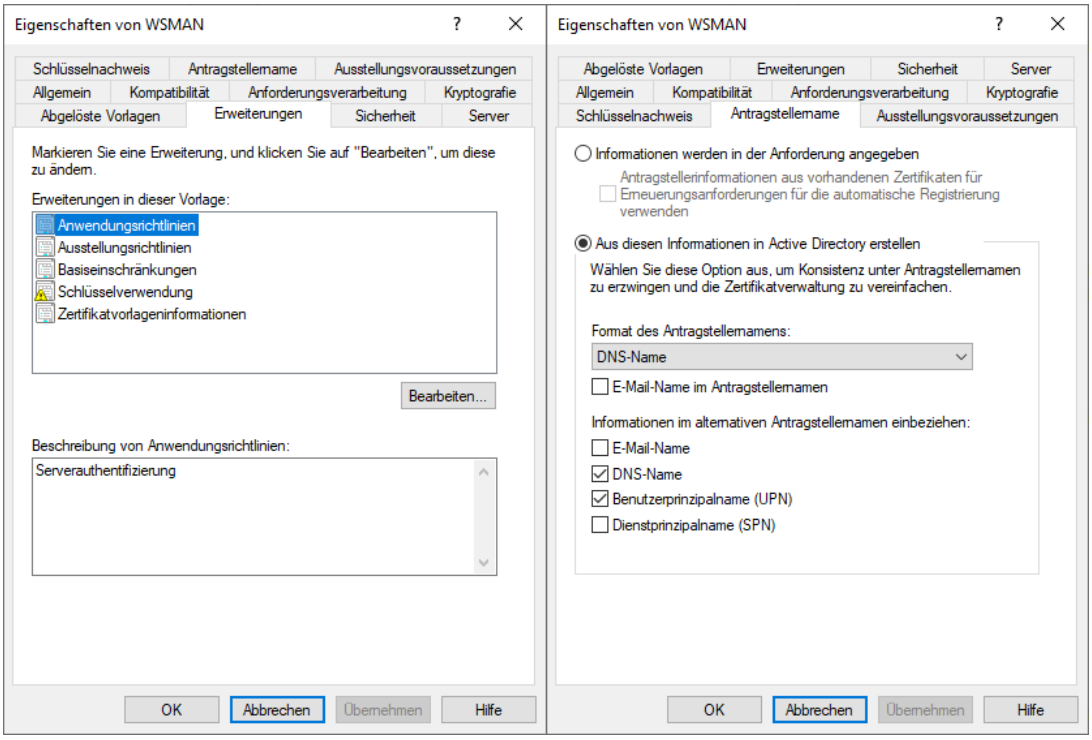


Kommen wir zur Umsetzung:

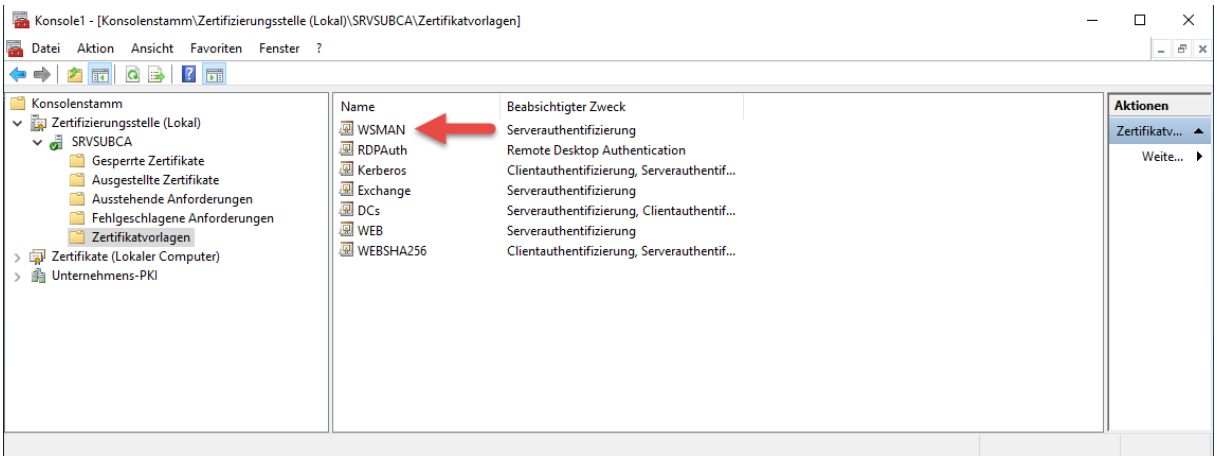
Zertifikatsvorlage auf der Zertifizierungstelle einrichten.



PSRemoting aktivieren



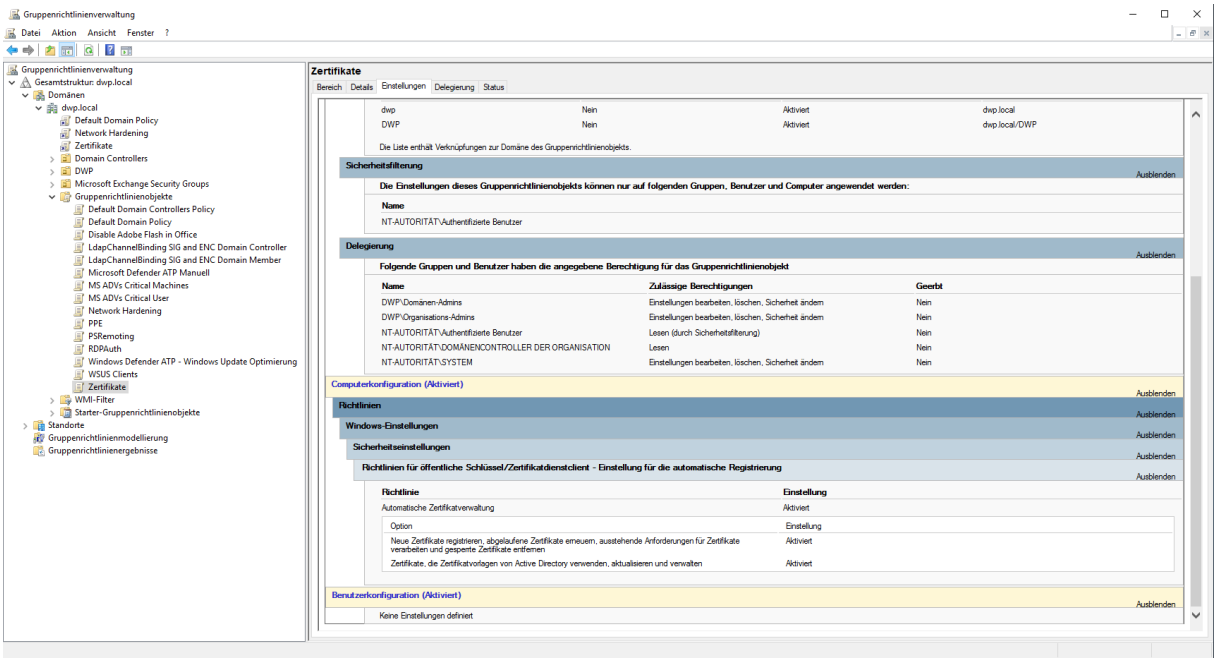
Die Vorlage wird veröffentlicht.

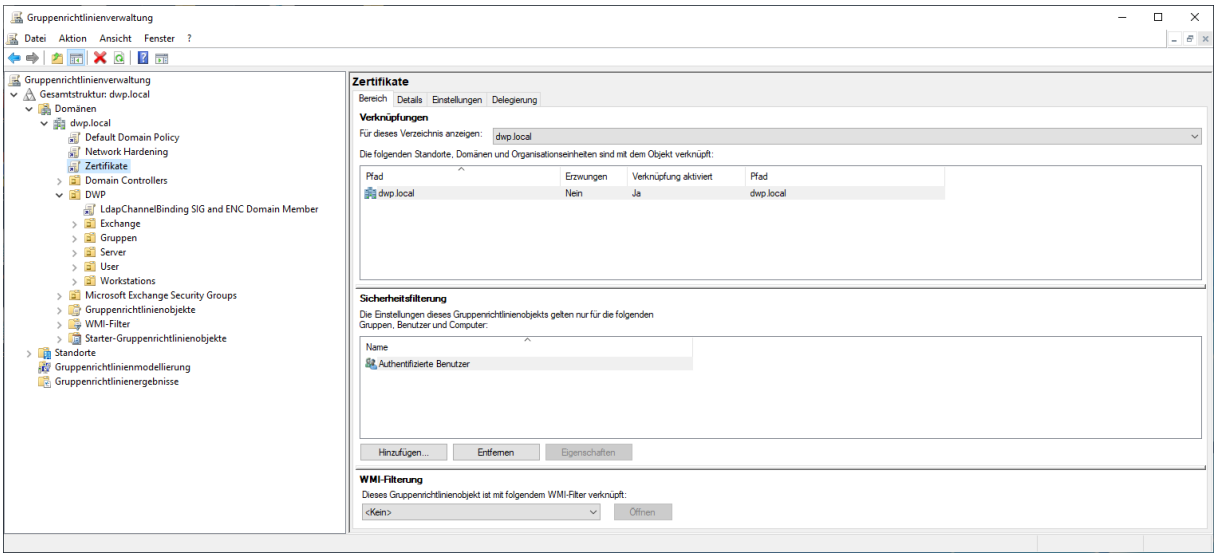


Erstellt von Jörn Walter **02.05.2020**

PSRemoting aktivieren

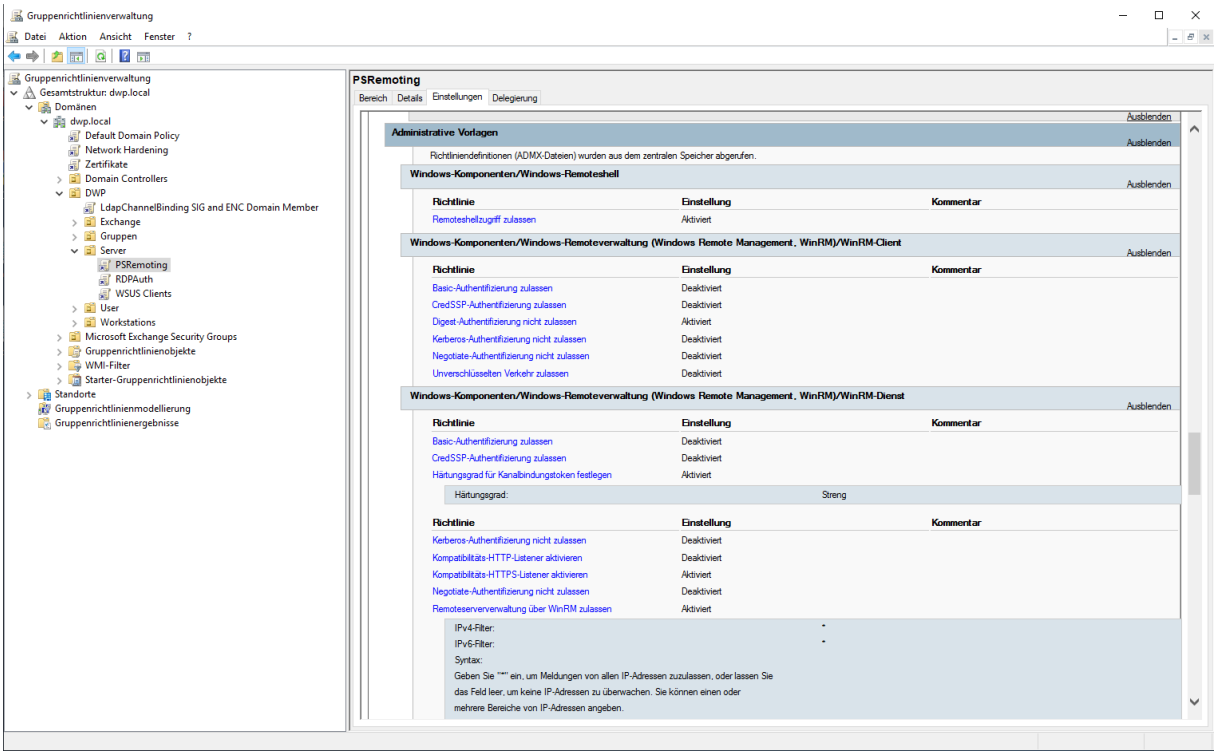
Für die automatische Verteilung (Registrierung) und Erneuerung von Zertifikaten (WSMAN) benötigen wir eine Richtlinie.





PSRemoting aktivieren

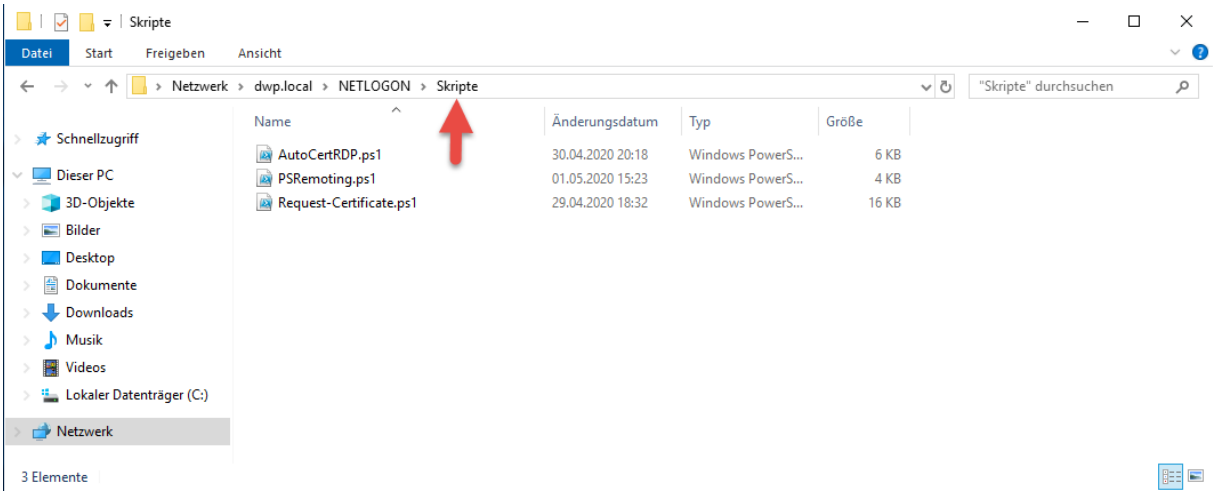
Zum Abschluss erstellen wir die Gruppenrichtlinie zur Konfiguration von WSMan, der Firewall, der Authentifizierung und der Listener als geplante Aufgabe.

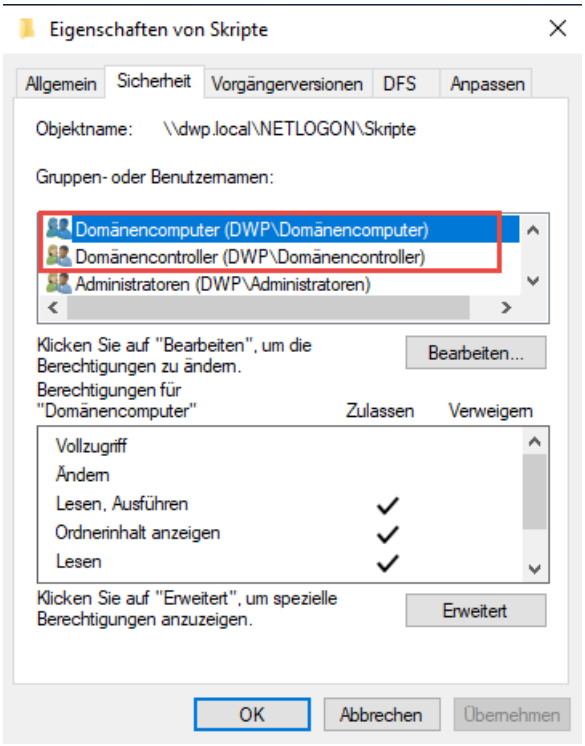


Das Skript befindet sich in meinem Lab im Ordner Skripte unter Netlogon.

```
)

$Event = Test-EventLog "PSRemoting"

If ($Event -eq $false) {

New-EventLog -LogName Application -Source "PSRemoting"

Write-Eventlog -LogName Application -Source "PSRemoting" -EntryType Information -EventId 0 -Message "Die Event-Source PSRemoting wurde angelegt."
}
# Zertifikatsspeicher festlegen
$certLocation = "LocalMachine\My"
# Funktion zum Auslesen der Erweiterung
function get_InstalledCertificateInfo($certLocation) {

    $array = gci Cert:\$certLocation | `

        select Thumbprint , `

        @{n="CN";e={($_.Subject).split("=")[1]}} , `
# Bei einem englischen OS -match Template
        @{n="IssuedfromTemplate";e={[regex]::match( ($_.extensions.Format(0) | ? { $_ -match "Vorlage" } ).split(",")[0] , '^(Vorlage\=)(\w.*)(\([0-9.]*(\))$' ).Groups[2].value}}
    return $array
}
# Varibalen setzen
$WinRMcertFound = $false
$WinRMCertTemplateName = "WSMAN"

$timeout = New-TimeSpan -Seconds 90
$endTime = (Get-Date).Add($timeout)
# Aufgabe zur Ermittlung des Fingerprints der oben genannten Zertifikatsvorlage
# Ist ein Zertifikat basierend auf der Vorlage vorhanden wird es an den HTTPS Listener gebunden
do {

    if (Get_InstalledCertificateInfo $certLocation | ? {$_.IssuedfromTemplate -eq $WinRMCertTemplateName}){
        $WinRMCorporateCertThumbprint = ((Get_InstalledCertificateInfo $certLocation | ? {$_.IssuedfromTemplate -eq $WinRMCertTemplateName}).Thumbprint).replace(" ","").toupper()
        $WinRMCorporateCertCN = (Get_InstalledCertificateInfo $certLocation | ? {$_.IssuedfromTemplate -eq $WinRMCertTemplateName}).CN
        Write-Eventlog -LogName Application -Source "PSRemoting" -EntryType Information -EventId 1 -Message "Ein Zertifikat basierend auf das Template $WinRMCertTemplateName mit dem Thu
        write-host "Ein Zertifikat basierend auf das Template $WinRMCertTemplateName mit dem Thumbprint $WinRMCorporateCertThumbprint wurde im Computerspeicher gefunden" -ForegroundColo
        $WinRMcertFound = $true

        Write-Host "Der HTTPS Listener wird neu konfiguriert" -ForegroundColor Cyan
        winrm delete winrm/config/Listener?Address=*+Transport=HTTPS
        New-Item WSMan:\localhost\Listener -Transport HTTPS -Address * -CertificateThumbprint $WinRMCorporateCertThumbprint -Force
```

PSRemoting aktivieren





Das Skript sorgt dafür, dass das Zertifikat aus der Vorlage WSMAN immer an den HTTPS Listener gebunden wird/bleibt, auch nach einer Erneuerung des Zertifikats.

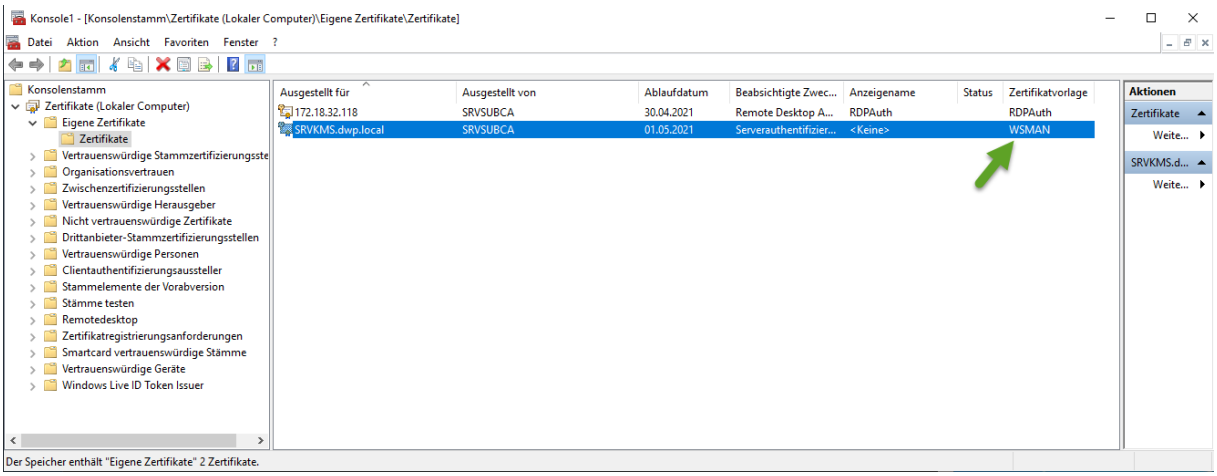
Wie lange bei euch ein Zertifikat gültig ist und wie oft ihr den Task laufen lasst, sollte jeder für sich entscheiden.

Mit dieser Methode habe ich für mich eine brauchbare Lösung gefunden.

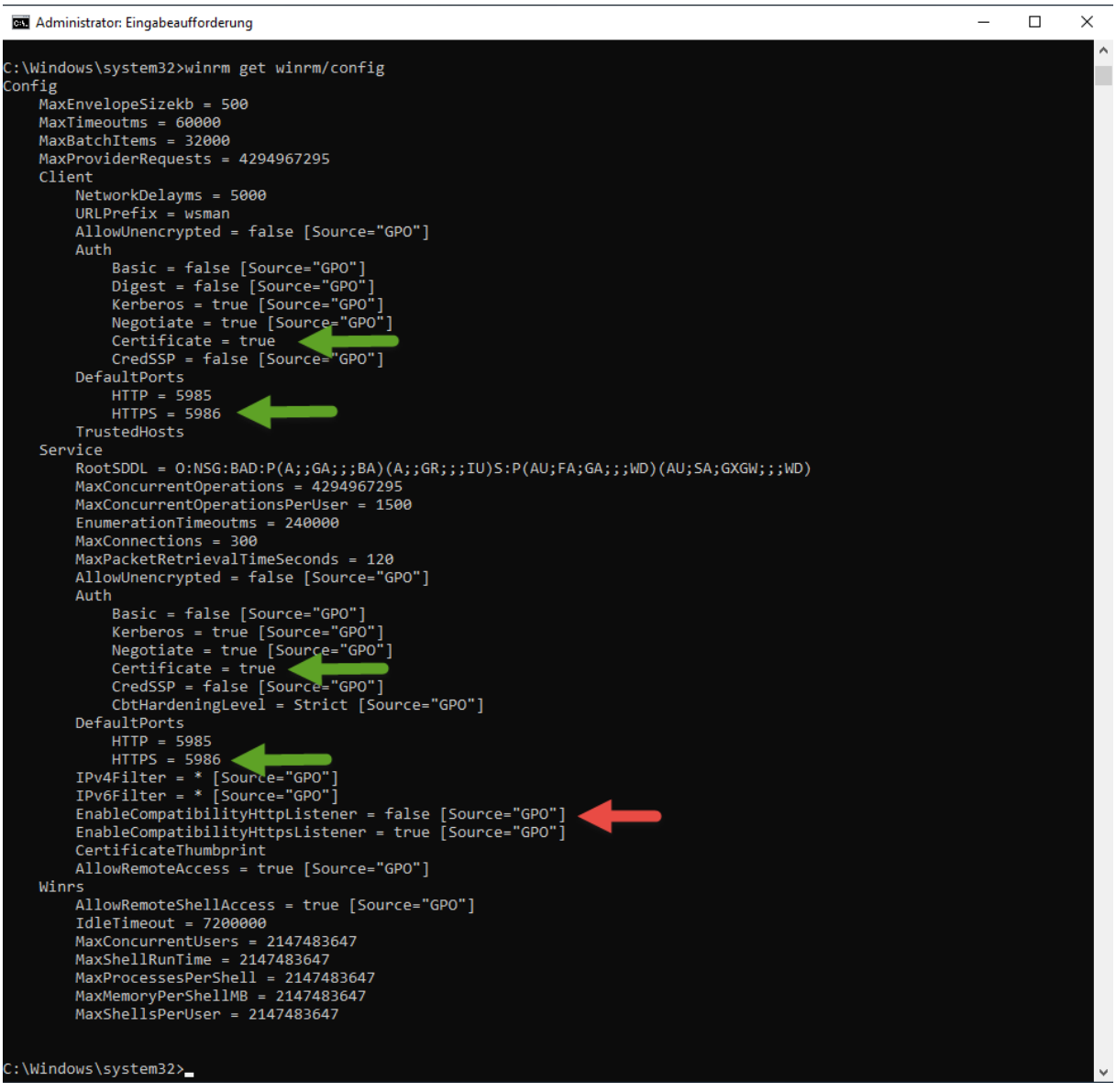
PSRemoting aktivieren

Nachdem alles umgesetzt wurde, sieht das Ergebnis wie folgt aus:

Es existiert ein Zertifikat auf Basis der Vorlage namens WSMAN.

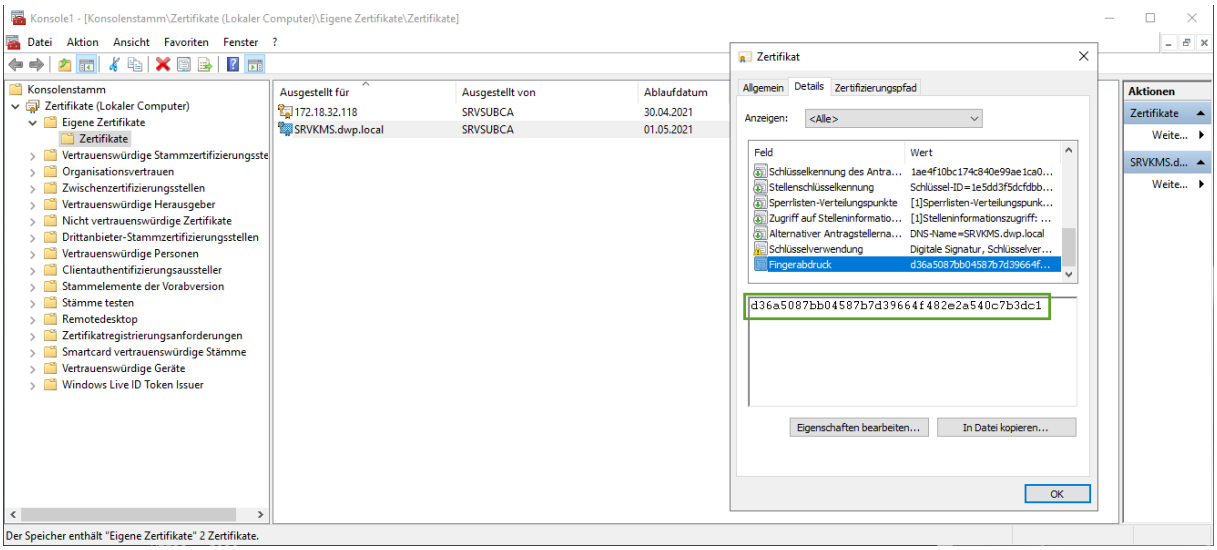


winrm get winrm/config



PSRemoting aktivieren

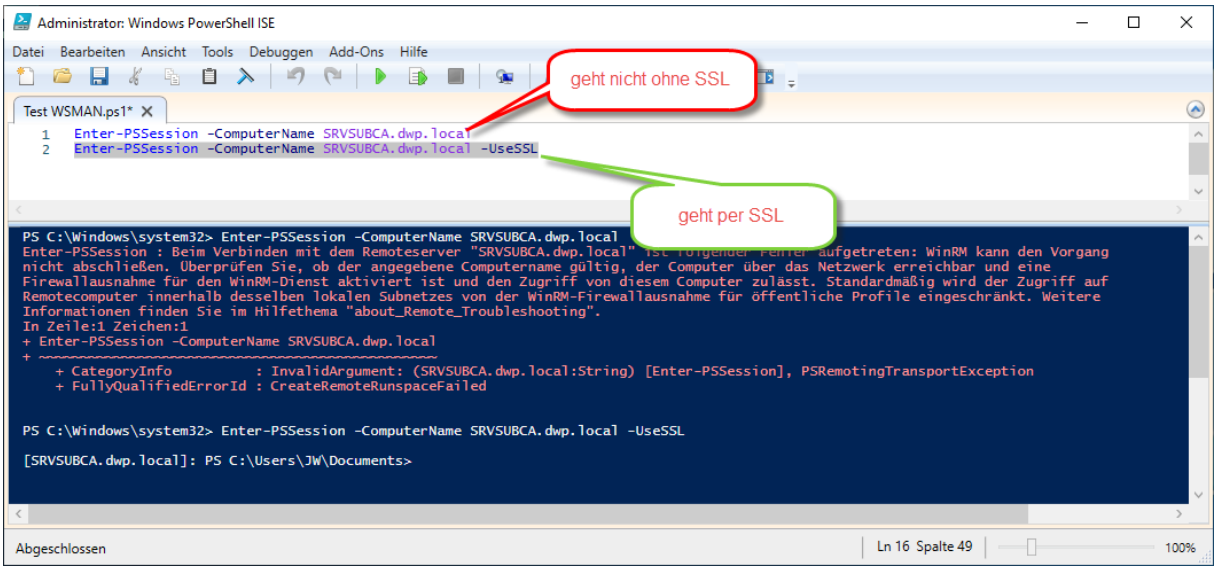
Das individuelle „WSMAN" Zertifikat,



wurde odnungsgemäß an WSMAN gebunden.

PSRemoting aktivieren

Der Test bestätigt die saubere Umsetzung.



Zum Testen:

Enter-PSSession -CN SRVSubCA.dwp.local # Fehler wegen fehlender SSL Nutzung
Enter-PSSession -CN SRVSubCA.dwp.local -UseSSL
Stop-Service spooler
Get-Service spooler
Start-Service spooler
Restart-Service spooler
Exit-PSSession

Enter-PSSession -CN SRVSubCA.dwp.local -UseSSL -Authentication Kerberos
Exit-PSSession

New-PSSession -CN SRVSubCA.dwp.local -UseSSL -Authentication Kerberos
Enter-PsSession –ID 9
Exit-PSSession
Remove-PSSession -ID 9
Get-PSSession | Disconnect-PSSession

Invoke-Command -CN SRVKMS.dwp.local,SRVSubCA.dwp.local -UseSSL -ScriptBlock {Get-Service spooler}