



WinRM-CertAuth

Für das automatische Deployment von Zertifikaten und den zu setzenden Einstellungen zur Absicherung von WinRM, ist folgendes zu tun.

GPO

Das neue Gruppenrichtlinienobjekt benötigt eine Richtlinie zur Freischaltung des lokalen eingehenden Ports 5986 und zwar nur für das Domänenprofil.

Computer Configuration (Enabled)

Policies

Windows Settings

Security Settings

Windows Firewall with Advanced Security

Global Settings

Policy	Setting
Policy version	2.31
Disable stateful FTP	Not Configured
Disable stateful PPTP	Not Configured
IPsec exempt	Not Configured
IPsec through NAT	Not Configured
Preshared key encoding	Not Configured
SA idle time	Not Configured
Strong CRL check	Not Configured

Inbound Rules

Name	Description
WinRM-HTTPS	This rule might contain some elements that cannot be interpreted by the current version of GPMC reporting module
Enabled	True
Program	Any
Action	Allow
Security	Require authentication
Authorized computers	
Authorized users	
Protocol	6
Local port	5986
Remote port	Any
ICMP settings	Any
Local scope	Any
Remote scope	Any
Profile	Domain
Network interface type	All
Service	All programs and services
Allow edge traversal	False
Group	

Connection Security Settings

Die Windows Remote Management-Einstellungen sollten für Client/Server wie folgt aussehen. *** Negotiate wird gebraucht ***

Administrative Templates

Policy definitions (ADMX files) retrieved from the central store.

Windows Components/Windows Remote Management (WinRM)/WinRM Client

Policy	Setting	Comment
Allow Basic authentication	Disabled	
Allow CredSSP authentication	Disabled	
Allow unencrypted traffic	Disabled	
Disallow Digest authentication	Enabled	
Disallow Kerberos authentication	Disabled	
Disallow Negotiate authentication	Enabled	

Windows Components/Windows Remote Management (WinRM)/WinRM Service

Policy	Setting	Comment
Allow Basic authentication	Disabled	
Allow CredSSP authentication	Disabled	
Allow unencrypted traffic	Disabled	
Disallow Kerberos authentication	Disabled	
Disallow Negotiate authentication	Enabled	
Turn On Compatibility HTTP Listener	Disabled	

Das automatische Enrollment muss folgende Einstellungen gesetzt bekommen!

Public Key Policies/Certificate Services Client - Auto-Enrollment Settings

Policy	Setting
Automatic certificate management	Enabled
Option	Setting
Enroll new certificates, renew expired certificates, process pending certificate requests and remove revoked certificates	Enabled
Update and manage certificates that use certificate templates from Active Directory	Enabled



WinRM-CertAuth

Der Scheduled-Task braucht keine besonderen Parameter zum Starten des Skripts. Es sollte jedoch signiert sein, um in keine andere (Powershell-signierte Skripte) Härtungsfälle zu laufen.

The screenshot shows the 'WinRM-HTTPS' task configuration in Windows Task Scheduler. The task is set to run every 5 minutes for an indefinite duration, starting at 1:45:46 PM on 5/19/2023. The action is to start a program, specifically the PowerShell executable, with arguments pointing to a script named 'WinRMDeploy.ps1'. The settings indicate that the task should run with the highest privileges and should not stop if the computer is idle.

Property	Value
Name	WinRM-HTTPS
Author	dwpjw
Description	
Run only when user is logged on	S4U
UserId	NT AUTHORITY\SYSTEM
Run with highest privileges	HighestAvailable
Hidden	No
Configure for	1.2
Enabled	Yes

Property	Value
Repeat task every	5 minutes
Duration	Indefinitely
Stop all running tasks at end of repetition duration	No
Activate	5/19/2023 1:45:46 PM
Synchronize across time zones	No
Enabled	Yes

Property	Value
Program/script	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Arguments	-WinRMDeploy.ps1 -WinRMCertTemplateName "SHA256-WinRM-HTTPS"

Property	Value
Stop if the computer ceases to be idle	No
Restart if the idle state resumes	No
Start the task only if the computer is on AC power	No
Stop if the computer switches to battery power	No
Allow task to be run on demand	Yes
Run task as soon as possible after a scheduled start is missed	No
Stop task if it runs longer than	3 days
If the running task does not end when requested, force it to stop	Yes
If the task is already running, then the following rule applies	IgnoreNew

Netlogon:

Für die Skript-Freigabe reicht die Gruppe der Authenticated Users aus.

The screenshot shows a Windows File Explorer window displaying the 'netlogon' share. The address bar shows the path '\\dwp\netlogon'. The file list contains one item, 'WinRMDeploy.ps1', which is a Windows PowerShell script file, 5 KB in size, and was last modified on 5/19/2023 at 1:51 PM.

Name	Date modified	Type	Size
WinRMDeploy.ps1	5/19/2023 1:51 PM	Windows PowerS...	5 KB



WinRM-CertAuth

Certificate Authority:

Das Certificate-Template benötigt lediglich die im Bild zu sehenden Settings. Im Format entweder DNS name oder Common Name.

The image displays two side-by-side screenshots of the 'SHA256-WinRM-HTTPS Properties' dialog box, showing different tabs.

Left Screenshot (Subject Name tab):

- Superseded Templates:** General, Compatibility, Request Handling, Cryptography, Key Attestation
- Subject Name:**
 - ☐ Supply in the request
 - ☐ Use subject information from existing certificates for autoenrollment renewal requests
 - ☒ Build from this Active Directory information
 - Select this option to enforce consistency among subject names and to simplify certificate administration.
 - Subject name format: **DNS name** (dropdown)
 - ☐ Include e-mail name in subject name
 - Include this information in alternate subject name:
 - ☐ E-mail name
 - ☒ DNS name
 - ☐ User principal name (UPN)
 - ☐ Service principal name (SPN)

Right Screenshot (Security tab):

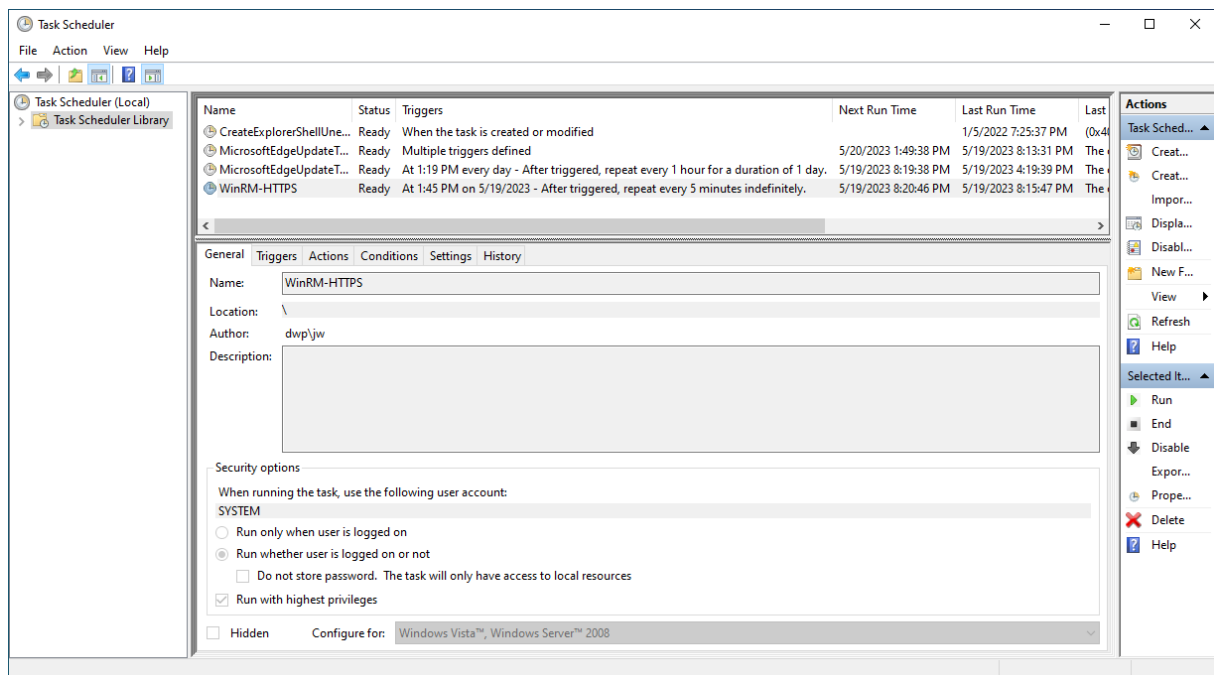
- Superseded Templates:** Subject Name, Issuance Requirements, General, Compatibility, Request Handling, Cryptography, Key Attestation
- Security:**
 - Purpose: **Signature and encryption** (dropdown)
 - ☐ Delete revoked or expired certificates (do not archive)
 - ☐ Include symmetric algorithms allowed by the subject
 - ☐ Archive subject's encryption private key
 - ☐ Use advanced Symmetric algorithm to send the key to the CA
 - ☐ Authorize additional service accounts to access the private key (Key Permissions... button)
 - ☐ Allow private key to be exported
 - ☒ Renew with the same key
 - ☐ For automatic renewal of smart card certificates, use the existing key if a new key cannot be created
 - Do the following when the subject is enrolled and when the private key associated with this certificate is used:
 - ☒ Enroll subject without requiring any user input
 - ☐ Prompt the user during enrollment
 - ☐ Prompt the user during enrollment and require user input when the private key is used

Deployment:

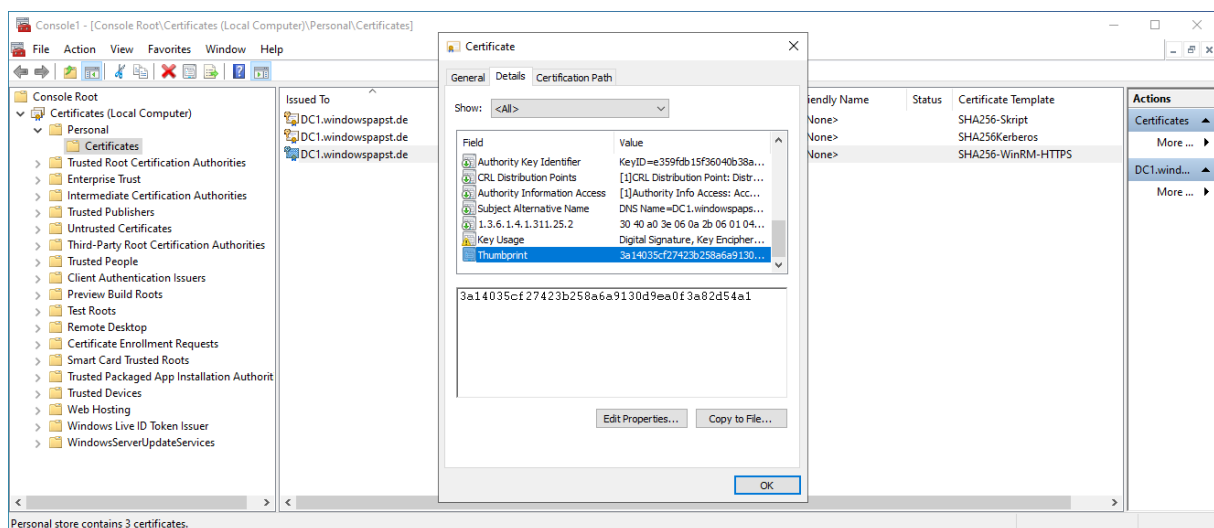
Nachdem das GPO angewendet wurde, sollte der Task auf dem Member-Server in der Library zu sehen sein.



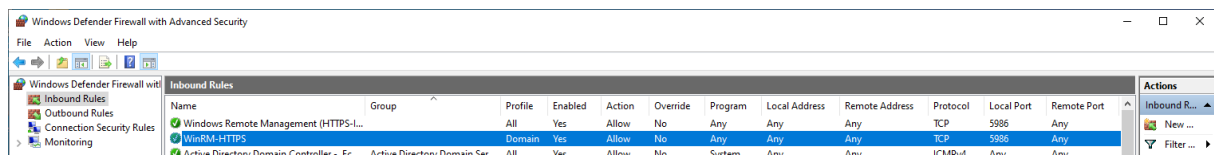
WinRM-CertAuth



Das Zertifikat sollte ebenfalls schon vorhanden sein. Wenn nicht, dann wird das gesteuerte Skript nochmals ein gpupdate ausführen, um den Prozess anzustoßen. Andernfalls per certutil -pulse; sollte aber nicht nötig sein.



Auf dem Member-Server sollte ebenfalls der eingehende Firewallport 5986 freigeschaltet sein.





WinRM-CertAuth

Zur Kontrolle fragen wir die WinRM-Listener ab. Das neue Zertifikat sollte gebunden sein. Kontrolle des Thumbprints.

```
Administrator: Windows PowerShell

PS C:\Windows\system32> winrm enumerate winrm/config/Listener
Listener
  Address = *
  Transport = HTTPS
  Port = 5986
  Hostname = DC1.windowspapst.de
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint = 3A14035CF27423B258A6A9130D9EA0F3A82D54A1
  ListeningOn = 127.0.0.1, 172.18.32.31, ::1

PS C:\Windows\system32> _
```

Die Member-Server WinRM-Config sollte alle Policy-Einstellungen aus dem GPO wiedergeben. Leider funktioniert die Abfrage nicht ohne Negotiate.

```
Administrator: Command Prompt

C:\Windows\system32>winrm get winrm/config
WSManFault
  Message = The WinRM client cannot process the request. Negotiate authentication is currently disabled in the client configuration. Change the client configuration and try the request again. If this is a request for the local configuration, use one of the enabled authentication mechanisms still enabled. To use Kerberos, specify the local computer name as the remote destination. To use Basic, specify the local computer name as the remote destination, specify Basic authentication and provide user name and password.

Error number: -2144108319 0x803380E1
The WinRM client cannot process the request. Negotiate authentication is currently disabled in the client configuration. Change the client configuration and try the request again. If this is a request for the local configuration, use one of the enabled authentication mechanisms still enabled. To use Kerberos, specify the local computer name as the remote destination. To use Basic, specify the local computer name as the remote destination, specify Basic authentication and provide user name and password.

C:\Windows\system32>_
```

Wenn Negotiate nur Client-seitig aktiv ist erscheint folgende Ausgabe:

```
Administrator: Command Prompt

C:\Windows\system32>winrm get winrm/config
WSManFault
  Message = The WinRM client cannot process the request. The WinRM client tried to use Negotiate authentication mechanism, but the destination computer (localhost:47001) returned an 'access denied' error. Change the configuration to allow Negotiate authentication mechanism to be used or specify one of the authentication mechanisms supported by the server. To use Kerberos, specify the local computer name as the remote destination. Also verify that the client computer and the destination computer are joined to a domain. To use Basic, specify the local computer name as the remote destination, specify Basic authentication and provide user name and password. Possible authentication mechanisms reported by server: Kerberos

Error number: -2147024891 0x80070005
Access is denied.

C:\Windows\system32>
```



WinRM-CertAuth

Sobald der Authentifizierungsmechanismus Negotiate Client/Server-seitig wieder zur Verfügung steht, kann auch die Konfiguration wieder abgefragt werden. Nur interessant sofern es abgeschaltet wurde.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> winrm get winrm/config
Config
  MaxEnvelopeSizeKb = 500
  MaxTimeoutms = 60000
  MaxBatchItems = 32000
  MaxProviderRequests = 4294967295
  Client
    NetworkDelaysms = 5000
    URLPrefix = wsman
    AllowUnencrypted = false [Source="GPO"]
    Auth
      Basic = false [Source="GPO"]
      Digest = false [Source="GPO"]
      Kerberos = true [Source="GPO"]
      Negotiate = true
      Certificate = true
      CredSSP = false [Source="GPO"]
    DefaultPorts
      HTTP = 5985
      HTTPS = 5986
    TrustedHosts
  Service
    RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
    MaxConcurrentOperations = 4294967295
    MaxConcurrentOperationsPerUser = 1500
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds = 120
    AllowUnencrypted = false [Source="GPO"]
    Auth
      Basic = false [Source="GPO"]
      Kerberos = true [Source="GPO"]
      Negotiate = true [Source="GPO"]
      Certificate = false
      CredSSP = false [Source="GPO"]
      CbtHardeningLevel = Relaxed
    DefaultPorts
      HTTP = 5985
      HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
    EnableCompatibilityHttpListener = false [Source="GPO"]
    EnableCompatibilityHttpsListener = false
    CertificateThumbprint
    AllowRemoteAccess = true
  Winrs
    AllowRemoteShellAccess = true
    IdleTimeout = 7200000
    MaxConcurrentUsers = 2147483647
    MaxShellRunTime = 2147483647
    MaxProcessesPerShell = 2147483647
    MaxMemoryPerShellMB = 2147483647
    MaxShellsPerUser = 2147483647
PS C:\Windows\system32> _
```



WinRM-CertAuth

Member-Server Connection-Test via SSL & Kerberos:

```
Administrator: Windows PowerShell

PS C:\Windows\system32> Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Kerberos
[DC2.windowspapst.de]: PS C:\Users\jw\Documents> Exit-PSsession
PS C:\Windows\system32> Enter-PSsession -CN DC2.windowspapst.de -UseSSL
[DC2.windowspapst.de]: PS C:\Users\jw\Documents> Exit-PSsession
PS C:\Windows\system32> Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Kerberos
[DC2.windowspapst.de]: PS C:\Users\jw\Documents> netstat | findstr "5986"
TCP 172.18.32.32:5986 DC1:60055 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60056 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60057 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60058 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60059 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60060 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60061 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60062 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60065 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60066 ESTABLISHED
TCP 172.18.32.32:5986 DC1:60067 ESTABLISHED
[DC2.windowspapst.de]: PS C:\Users\jw\Documents>
```

Der Versuch über Negotiate und Basic Auth ist gescheitert sofern deaktiviert!

```
Administrator: Windows PowerShell

PS C:\Windows\system32> Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Negotiate -Port 5986
Enter-PSsession : Connecting to remote server DC2.windowspapst.de failed with the following error message : The WinRM
client cannot process the request. Negotiate authentication is currently disabled in the client configuration. Change
the client configuration and try the request again. If this is a request for the local configuration, use one of the
enabled authentication mechanisms still enabled. To use Kerberos, specify the local computer name as the remote
destination. To use Basic, specify the local computer name as the remote destination, specify Basic authentication
and provide user name and password. For more information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:1
+ Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Negot ...
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (DC2.windowspapst.de:String) [Enter-PSsession], PSRemotingTransportExce
+ FullyQualifiedErrorId : CreateRemoteRunspaceFailed

PS C:\Windows\system32>
```

```
Administrator: Windows PowerShell

PS C:\Windows\system32> Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Basic
Enter-PSsession : The WinRM client cannot process the request. Requests must include user name and password when Basic
or Digest authentication mechanism is used. Add the user name and password or change the authentication mechanism and
try the request again.
At line:1 char:1
+ Enter-PSsession -CN DC2.windowspapst.de -UseSSL -Authentication Basic
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (DC2.windowspapst.de:String) [Enter-PSsession], PSInvalidOperationException
+ FullyQualifiedErrorId : CreateRemoteRunspaceFailed

PS C:\Windows\system32>
```

Problem!

Negotiate wird für den automatischen Task und für weitere Aufgaben benötigt. Ohne Negotiate gibt es auch Probleme mit dem Server Manager.



WinRM-CertAuth

Authentication-Methods

Es gibt 3 Möglichkeiten der Authentifizierung.

1: Basisauthentifizierung

Auf dem Zielsystem erfolgt die Anmeldung über einen lokalen Benutzer. Passwort und Daten werden unverschlüsselt (Base64 ist keine Verschlüsselung) per HTTP übertragen.

2: Domänenbenutzerauthentifizierung

Die Authentifizierung erfolgt über Kerberos. Die Übertragung der Nutzerdaten erfolgt weiterhin unverschlüsselt per HTTP.

- Der Benutzer muss Mitglied der lokalen Administratoren sein.
- Die WinRM-Config wird erwartet: Auth Kerberos = true; Auth Basis=false; AllowUnencrypted=false

3: Zertifikatbasierte Authentifizierung

Voraussetzung ist, dass das Zielsystem über ein eigenes Serverzertifikat verfügt. In diesem Fall werden sowohl die Anmeldung als auch die Daten verschlüsselt über HTTPS übertragen.

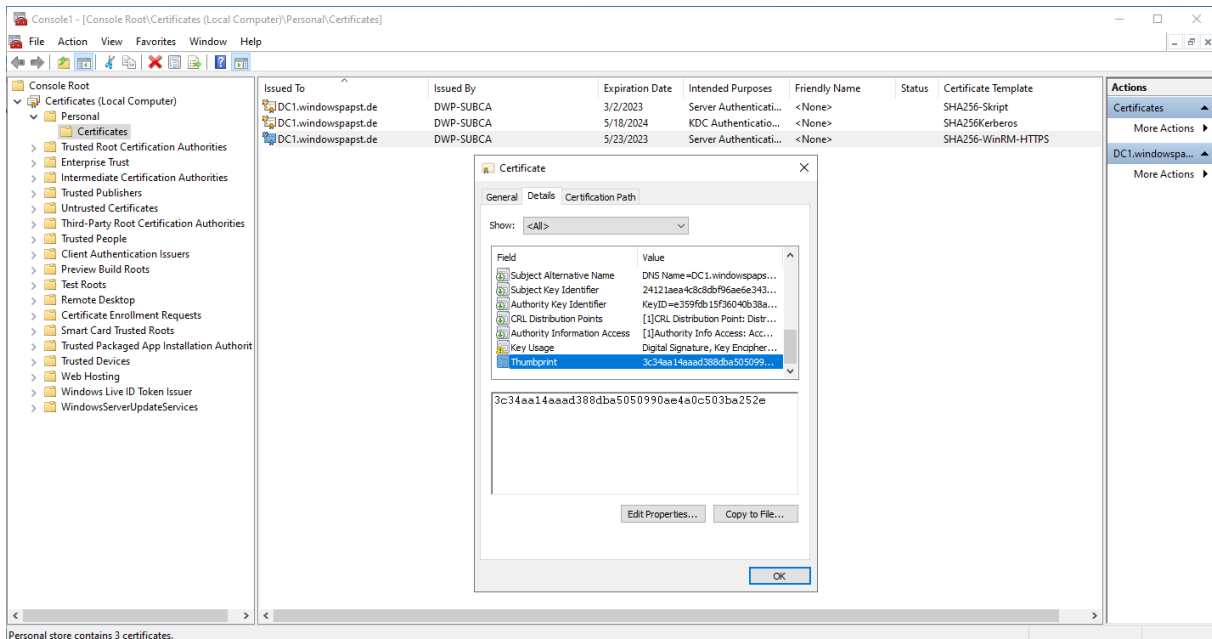
- SHA-256 Zertifikat
- WinRM HTTPS-Listener
- Funktionierende Kerberos Authentifizierung



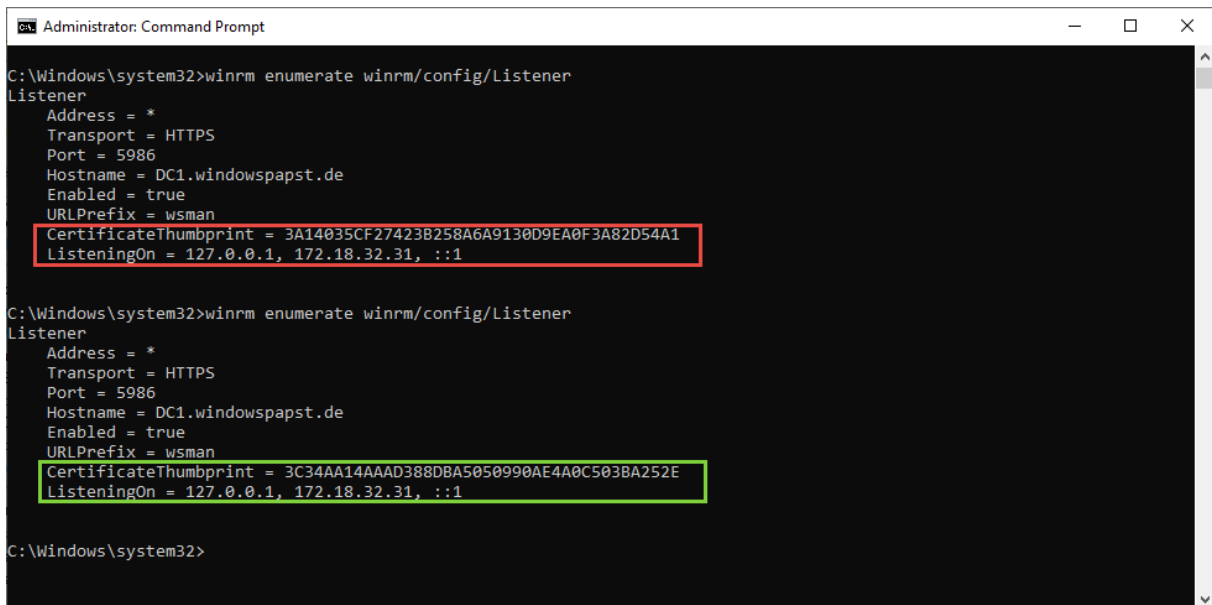
WinRM-CertAuth

Certificate Renewal Process

Das Zertifikat wurde während der Renewal Phase automatisch erneuert.



Der Deployment Task hat erkannt, dass das alte und bereits ersetzte Zertifikat nicht mehr der ursprünglichen Konfiguration entspricht und hat es somit an das aktuelle Zertifikat gebunden.





WinRM-CertAuth

Remove HTTPS-Config and Re-Create http Listener

Zuerst geben wir die Listener samt Thumbprints aus. Das dient lediglich der Überprüfung.

winrm enumerate winrm/config/listener

```
Administrator: Windows PowerShell
PS C:\Windows\system32> winrm enumerate winrm/config/listener
Listener
  Address = *
  Transport = HTTPS
  Port = 5986
  Hostname = FI.windowspapst.de
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint = ACE6B9F1D28A0FB68B1EA618A5EBC88778D17ECF
  ListeningOn = 127.0.0.1, 172.18.32.41, ::1
PS C:\Windows\system32>
```

Als nächstes liest man den/die Listener-ID aus und löscht diese anschließend.

dir wsman:\localhost\listener

Remove-Item -Path WSMan:\Localhost\listener\Listener_1305953032

```
Administrator: Windows PowerShell
PS C:\Windows\system32> dir wsman:\localhost\listener

WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Listener

Type      Keys                                     Name
----      -
Container {Transport=HTTPS, Address=*}           Listener_1305953032

PS C:\Windows\system32> Remove-Item -Path WSMan:\Localhost\listener\Listener_1305953032

Confirm
The item at WSMan:\localhost\Listener\Listener_1305953032 has children and the Recurse parameter was not specified. If you continue, all children will be removed with the item. Are you sure you want to continue?
[Y] Yes  [A] Yes to All  [N] No   [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
PS C:\Windows\system32> winrm enumerate winrm/config/listener
PS C:\Windows\system32>
```

Zum Abschluss stellt man den http Listener wieder her.

Winrm create winrm/config/listener?Address=*+Transport=HTTP

dir wsman:\localhost\listener

```
Administrator: Windows PowerShell
PS C:\Windows\system32> winrm create winrm/config/listener?Address=*+Transport=HTTP
ResourceCreated
  Address = http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
  ReferenceParameters
    ResourceURI = http://schemas.microsoft.com/wbem/wsman/1/config/listener
    SelectorSet
      Selector: Address = *, Transport = HTTP

PS C:\Windows\system32> dir wsman:\localhost\listener

WSManConfig: Microsoft.WSMan.Management\WSMan::localhost\Listener

Type      Keys                                     Name
----      -
Container {Transport=HTTP, Address=*}           Listener_1084132640

PS C:\Windows\system32>
```



WinRM-CertAuth

Manuelle Umsetzungshilfe – Cert-Binding:

```
New-SelfSignedCertificate -DnsName  
([System.Net.Dns]::GetHostByName($env:computerName)).Hostname -  
CertStoreLocation "cert:LocalMachineMy" -FriendlyName DC1  
  
$thumb = Get-ChildItem Cert:LocalMachineMy | where FriendlyName -eq DC1 | select  
Thumbprint  
  
New-WSManInstance -ResourceURI winrm/config/Listener -SelectorSet  
@{address="*";transport="https"} -ValueSet  
@{Hostname=[System.Net.Dns]::GetHostByName(($env:computerName)).Hostname;Ce  
rtificateThumbprint=$dump.thumbprint  
  
winrm e winrm/config/listener  
  
winrm set winrm/config/service/auth @{Certificate="true"}
```

Das beiliegende Powershell-Skript habe ich so gestaltet, das es für englische und deutsche Systeme geeignet ist.