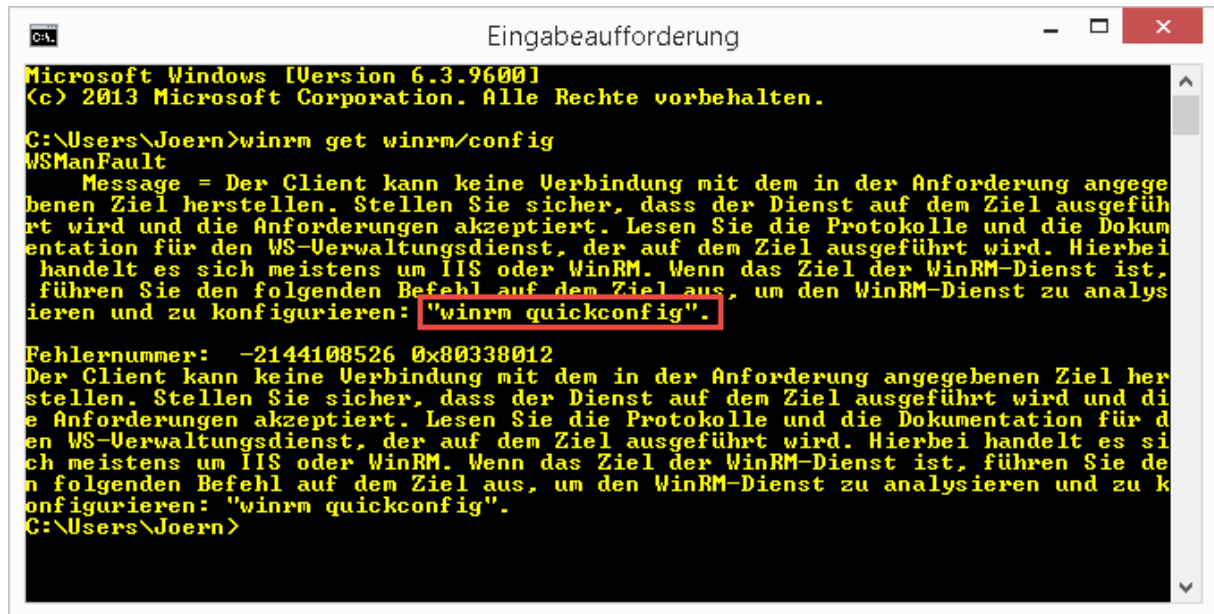


WINRM WINRS

Für eine funktionierende Remote Verwaltung muss WINRM (Windows Remote Management) erst einmal konfiguriert werden.

Mit dem Befehl **winrm get winrm/config** prüfen wir die aktuelle Konfiguration. WINRM ist nicht konfiguriert.



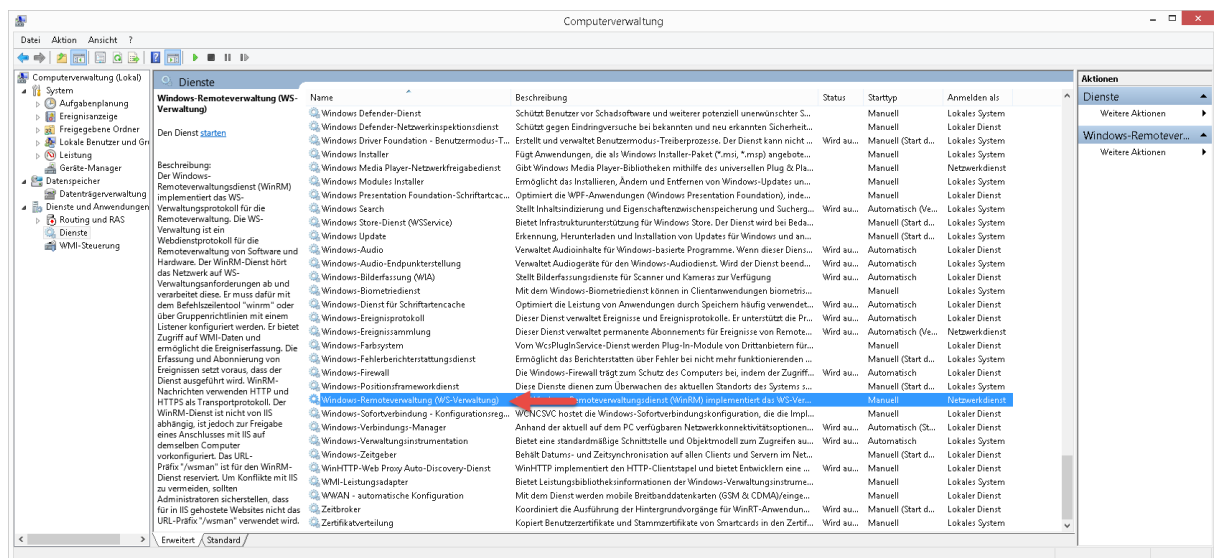
Zur Konfiguration können wir in der CMD entweder den Befehl **winrm quickconfig** ausführen oder in der Powershell das Äquivalent **Enable-PSRemoting**.

Die Voraussetzung für eine Remoteverwaltung sind immer die nötigen Berechtigungen die vorhanden sein müssen. In einer Domäne klar geregelt und am heimischen PC muss der Account auf beiden PC vorhanden sein!

Was geschieht bei der Aktivierung von WINRM auf einem Client?

- Der WINRM Dienst wird gestartet und auf automatisch gesetzt
- Ein Listener zum Abhören von Anforderungen wird erstellt; für eine beliebige IP
- Aktivieren eingehender Firewall-Regel für WS-Verwaltungsdatenverkehr

Wir sehen das der Dienst (Windows Remoteverwaltung) vor der Aktivierung/Einrichtung noch auf „Manuell“ steht.



WINRM WINRS

Ich starte die Powershell mit administrativen Rechten und führe folgenden Befehl aus:

Enable-PSRemoting

```
Administrator: Windows PowerShell

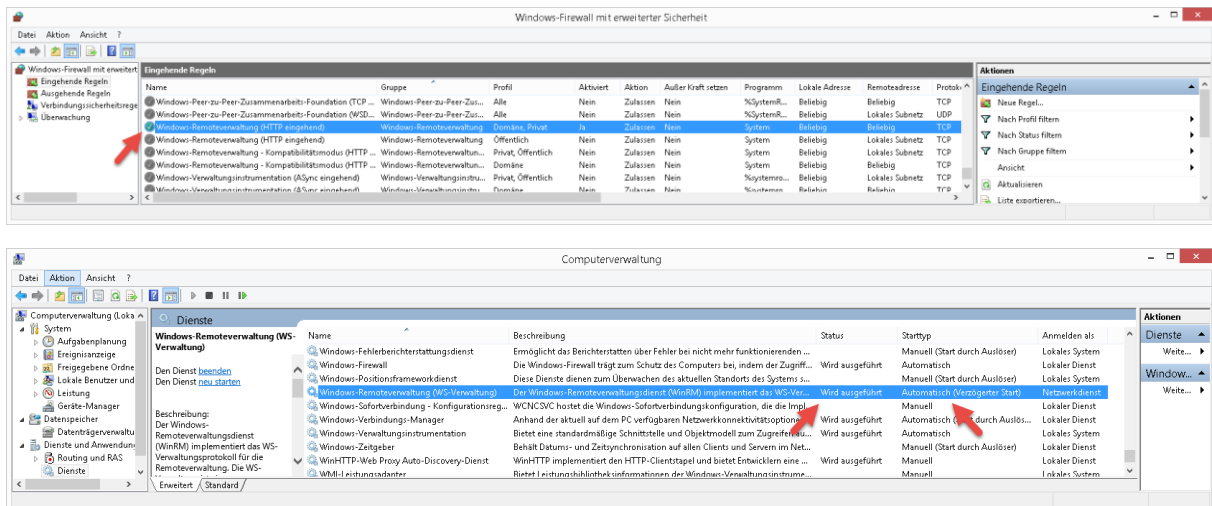
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Windows\system32> Enable-PSRemoting
WinRM wurde aktualisiert, um Anforderungen zu empfangen.
Der WinRM-Diensttyp wurde erfolgreich geändert.
Der WinRM-Dienst wurde gestartet.

WinRM wurde für die Remoteverwaltung aktualisiert.
Die WinRM-Firewallausnahme ist aktiviert.
"LocalAccountTokenFilterPolicy" wurde so konfiguriert, dass lokalen Benutzern remote Administratorrechte gewährt werden.

PS C:\Windows\system32>
```

Die Firewall wurde konfiguriert und der Dienst ist gestartet.



WINRM nutzt folgende Ports:

- 5985 http Port für WINRM
- 5986 https Port für WINRM

Einen Listener abfragen:

- winrm enumerate winrm/config/listener

WINRM WINRS

Wenn wir die Konfiguration erneut abfragen, dann sieht das Ergebnis wie gewünscht aus.



```
C:\Windows\system32>winrm get winrm/config
Config
  MaxEnvelopeSizekb = 500
  MaxTimeoutms = 60000
  MaxBatchItems = 32000
  MaxProviderRequests = 4294967295
  Client
    NetworkDelays = 5000
    URLPrefix = wsman
    AllowUnencrypted = false
    Auth
      Basic = true
      Digest = true
      Kerberos = true
      Negotiate = true
      Certificate = true
      CredSSP = false
    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;G
XGW;;;WD>
    MaxConcurrentOperations = 4294967295
    MaxConcurrentOperationsPerUser = 1500
    EnumerationTimeoutms = 240000
    MaxConnections = 300
    MaxPacketRetrievalTimeSeconds = 120
    AllowUnencrypted = false
    Auth
      Basic = false
      Kerberos = true
      Negotiate = true
      Certificate = false
      CredSSP = false
      CbtHardeningLevel = Relaxed
    DefaultPorts
      HTTP = 5985
      HTTPS = 5986
    IPv4Filter = *
    IPv6Filter = *
    EnableCompatibilityHttpListener = false
    EnableCompatibilityHttpsListener = false
    CertificateThumbprint
    AllowRemoteAccess = true
  Winrs
    AllowRemoteShellAccess = true
    IdleTimeout = 7200000
    MaxConcurrentUsers = 10
    MaxShellRunTime = 2147483647
    MaxProcessesPerShell = 25
    MaxMemoryPerShellMB = 1024
    MaxShellsPerUser = 30

C:\Windows\system32>
```

WINRM WINRS

Die Firewall kann über netsh auch manuell angepasst werden:

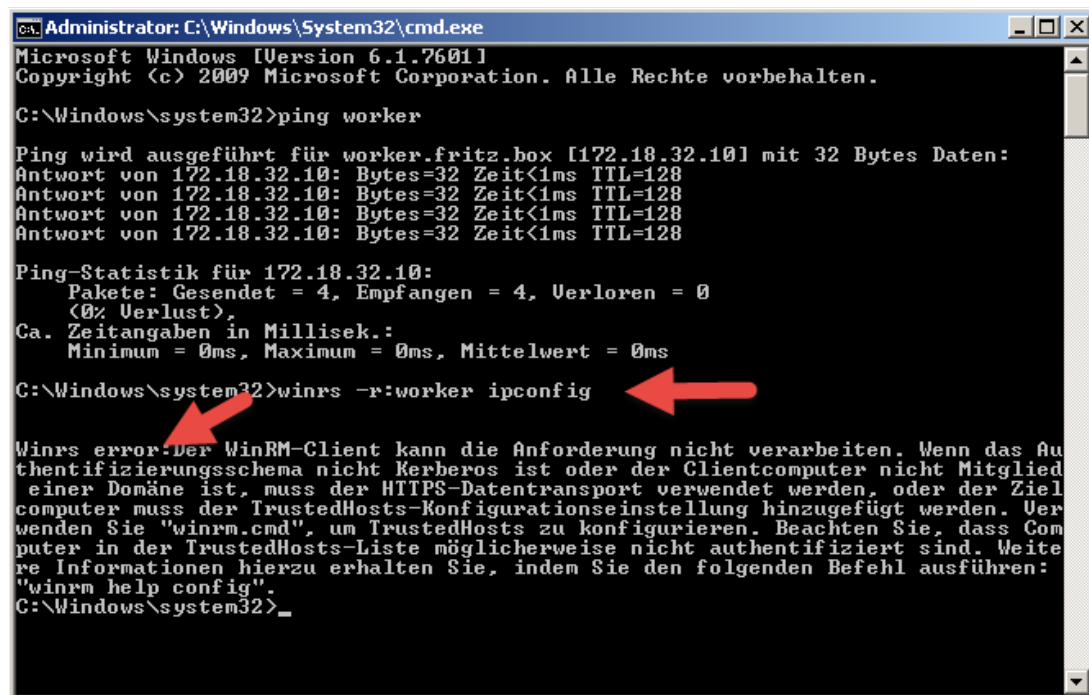
```
netsh advfirewall firewall add rule name="WinRM-HTTP" dir=in localport=5985  
protocol=TCP action=allow
```

```
netsh advfirewall firewall add rule name="WinRM-HTTPS" dir=in localport=5986  
protocol=TCP action=allow
```

Wer HTTPS einsetzt muss noch das Zertifikat binden:

```
winrm create winrm/config/Listener?Address=*+Transport=HTTPS @{Hostname="a-ssb-  
ads2.ndsedv.de";CertificateThumbprint="79358b0c62dd4a18cf8cb30a700068d28bc2ad8  
9"}
```

Okay, wenn ich jetzt versuche über den ClientX auf den Client Namens Worker Remote die IP Adresse abzufragen, erhalte ich folgende Fehlermeldung:



```
C:\Windows\system32>ping worker  
Ping wird ausgeführt für worker.fritz.box [172.18.32.10] mit 32 Bytes Daten:  
Antwort von 172.18.32.10: Bytes=32 Zeit<1ms TTL=128  
Antwort von 172.18.32.10: Bytes=32 Zeit<1ms TTL=128  
Antwort von 172.18.32.10: Bytes=32 Zeit<1ms TTL=128  
Antwort von 172.18.32.10: Bytes=32 Zeit<1ms TTL=128  
Ping-Statistik für 172.18.32.10:  
Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0  
        (0% Verlust),  
Ca. Zeitangaben in Millisek.:  
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms  
C:\Windows\system32>winrs -r:worker ipconfig  
Winrs error: Der WinRM-Client kann die Anforderung nicht verarbeiten. Wenn das Au-  
thentifizierungsschema nicht Kerberos ist oder der Clientcomputer nicht Mitglied  
einer Domäne ist, muss der HTTPS-Datentransport verwendet werden, oder der Ziel-  
computer muss der TrustedHosts-Konfigurationseinstellung hinzugefügt werden. Ver-  
wenden Sie "winrm.cmd", um TrustedHosts zu konfigurieren. Beachten Sie, dass Com-  
puter in der TrustedHosts-Liste möglicherweise nicht authentifiziert sind. Weite-  
re Informationen hierzu erhalten Sie, indem Sie den folgenden Befehl ausführen:  
"winrm help config".  
C:\Windows\system32>
```

Das Problem liegt entweder daran, dass WINRM auf ClientX nicht eingerichtet wurde oder an der Vertrauensstellung zwischen den beiden Maschinen.

Dann prüfen wir ob auf ClientX WINRM eingerichtet ist. Dazu führe ich wieder den Befehl **winrm get winrm/config** aus.

WINRM WINRS

```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\system32>hostname
ClientX
C:\Windows\system32>winrm get winrm/config
WSManFault
Message = Der Client kann keine Verbindung mit dem in der Anforderung angegebenen Ziel herstellen. Stellen Sie sicher, dass der Dienst auf dem Ziel ausgeführt wird und die Anforderungen akzeptiert. Lesen Sie die Protokolle und die Dokumentation für den WS-Verwaltungsdienst, der auf dem Ziel ausgeführt wird. Hierbei handelt es sich meistens um IIS oder WinRM. Wenn das Ziel der WinRM-Dienst ist, führen Sie den folgenden Befehl auf dem Ziel aus, um den WinRM-Dienst zu analysieren und zu konfigurieren: "winrm quickconfig".
Fehlernummer: -2144108526 0x80338012
Der Client kann keine Verbindung mit dem in der Anforderung angegebenen Ziel herstellen. Stellen Sie sicher, dass der Dienst auf dem Ziel ausgeführt wird und die Anforderungen akzeptiert. Lesen Sie die Protokolle und die Dokumentation für den WS-Verwaltungsdienst, der auf dem Ziel ausgeführt wird. Hierbei handelt es sich meistens um IIS oder WinRM. Wenn das Ziel der WinRM-Dienst ist, führen Sie den folgenden Befehl auf dem Ziel aus, um den WinRM-Dienst zu analysieren und zu konfigurieren: "winrm quickconfig".
C:\Windows\system32>
```

Auch hier müssen wir erst einmal WINRM aktivieren. Dieses Mal aber über die CMD.

Nach dem Ausführen des Befehls **winrm quickconfig** ist das Ergebnis jedoch ernüchternd. Es endet mit einer Fehlermeldung.

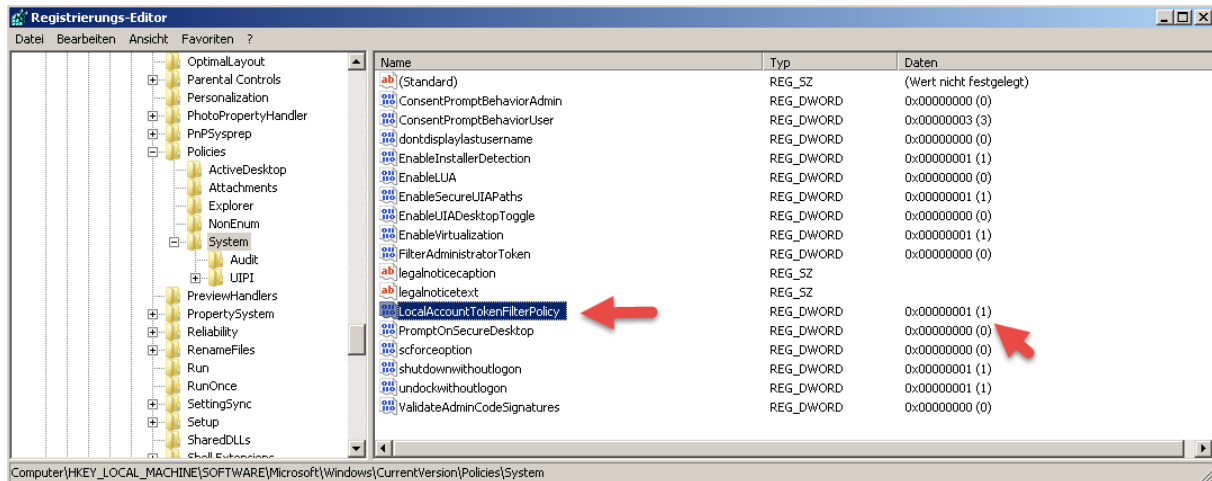
```
Administrator: C:\Windows\System32\cmd.exe
C:\Windows\system32>winrm quickconfig
WinRM ist nicht zum Empfangen von Anforderungen auf diesem Computer konfiguriert
Folgende Änderungen müssen durchgeführt werden:
Legen Sie den WinRM-Diensttyp auf einen verzögerten automatischen Start fest.
Starten Sie den WinRM-Dienst.
Konfigurieren Sie "LocalAccountTokenFilterPolicy" so, dass lokalen Benutzern remote Administratorrechte gewährt werden.
Diese Änderungen durchführen [y/n]? y
WinRM wurde aktualisiert, um Anforderungen zu empfangen.
Der WinRM-Diensttyp wurde erfolgreich geändert.
Der WinRM-Dienst wurde gestartet.
"LocalAccountTokenFilterPolicy" wurde so konfiguriert, dass lokalen Benutzern remote Administratorrechte gewährt werden.
WSManFault
Message = Zugriff verweigert
Fehlernummer: -2147024891 0x80070005
Zugriff verweigert
C:\Windows\system32>
```

WINRM WINRS

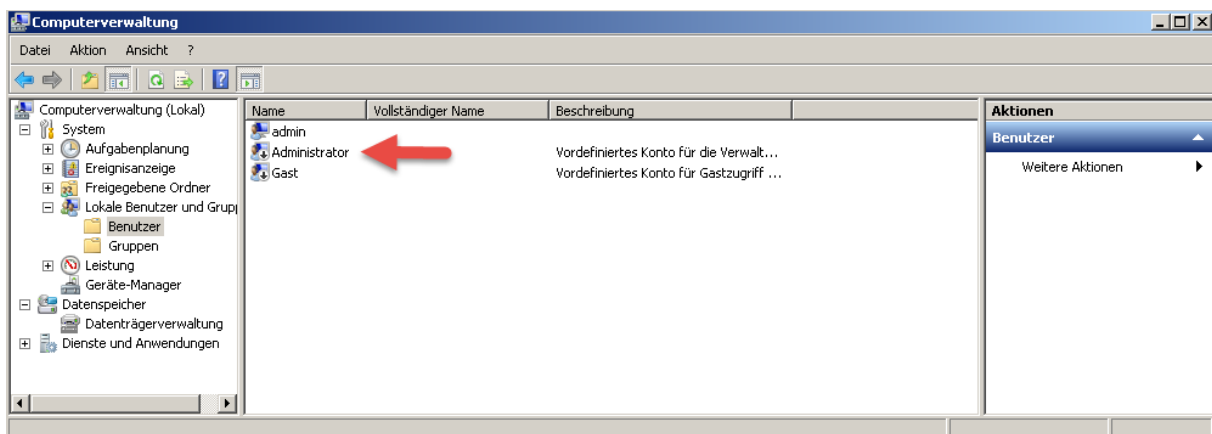
Die Lösung des Problems könnte dieser fehlende oder falsche Eintrag in der Registry sein.

Mit **REG ADD** fügen wir den Eintrag hinzu:

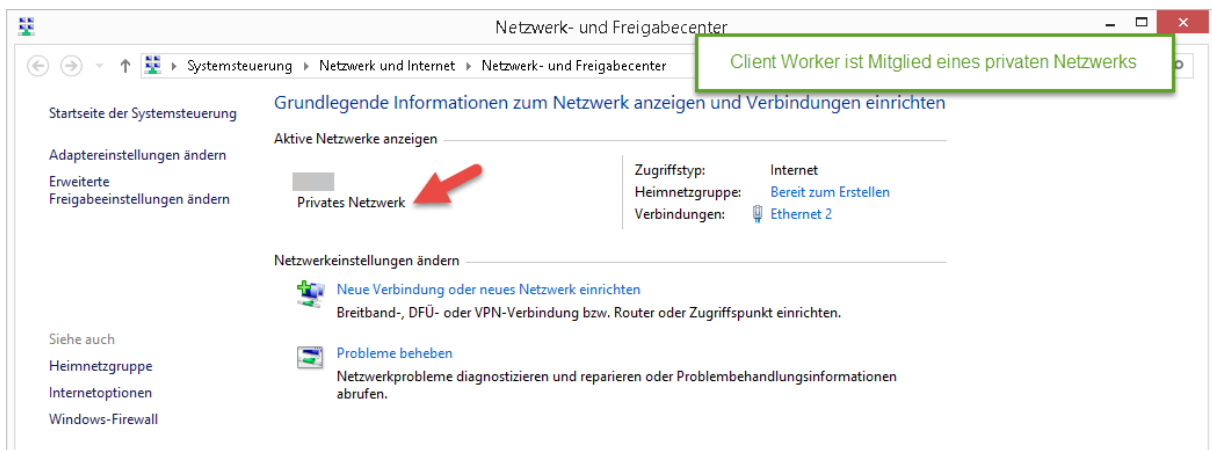
```
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v  
LocalAccountTokenFilterPolicy /t REG_DWORD /d 1 /f
```



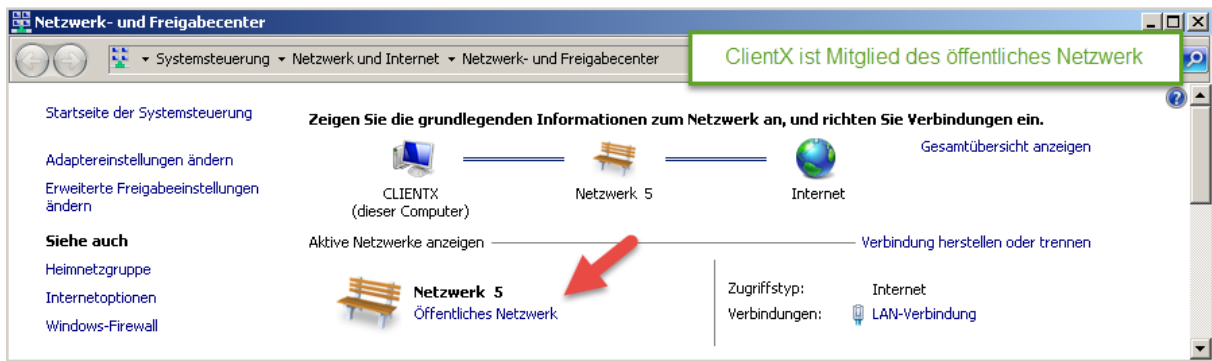
Wenn der Registry-Eintrag vorhanden ist und den Wert 1 trägt dann müssen wir den lokalen Administrator überprüfen. Wenn dieser deaktiviert und/oder kein Passwort hinterlegt ist, dann sollten wir das jetzt nachholen und später wieder deaktivieren.



Wenn auch das nicht klappt, dann müssen wir die Netzwerk-Mitgliedschaft überprüfen.

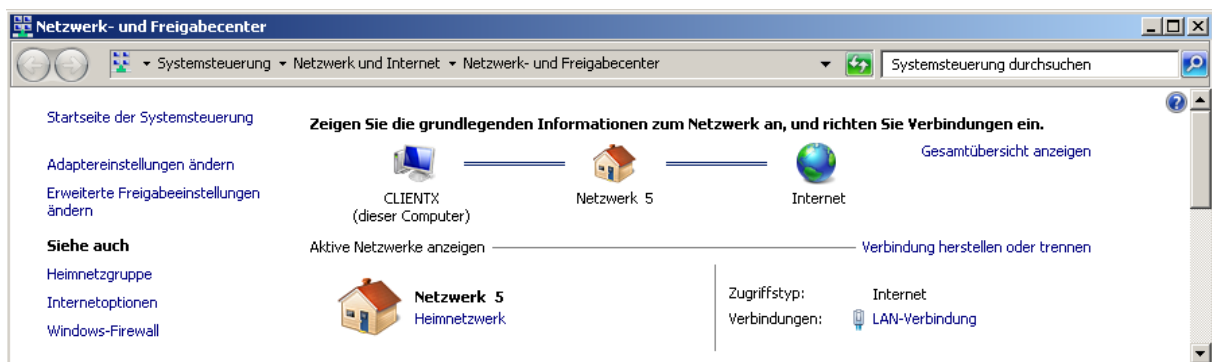


WINRM WINRS

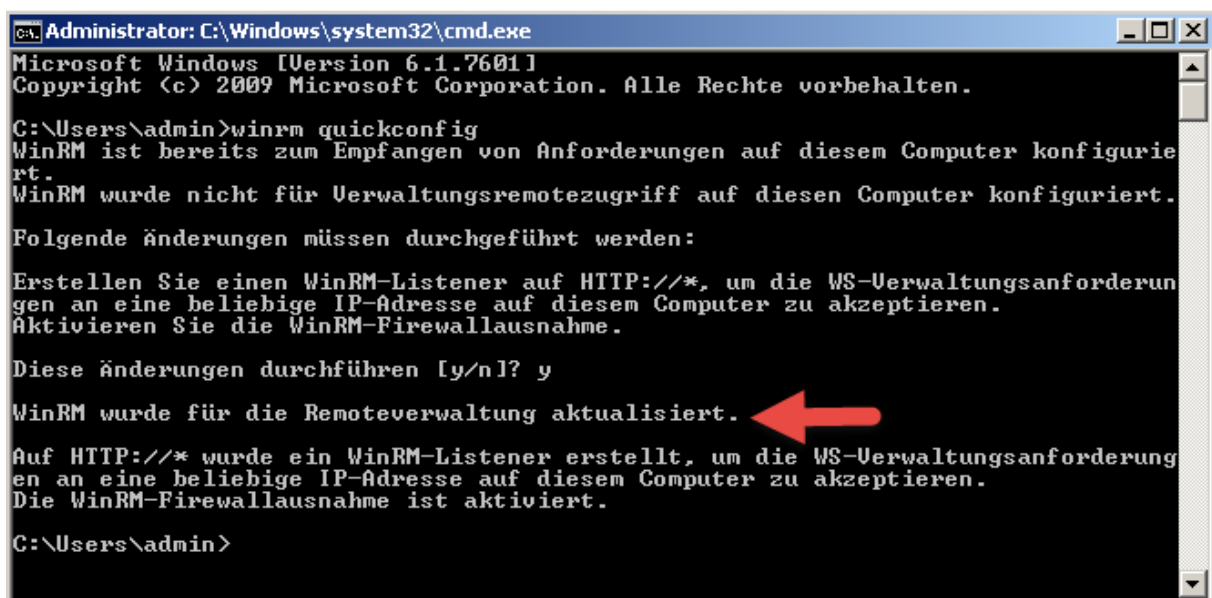


ClientX sollte Mitglied des gleichen Netzwerkprofils sein wie der Client Worker.
Über einen Klick auf Öffentliches Netzwerk können wir die Mitgliedschaft ändern.

Für das öffentliche Profil ist WINRM nämlich nicht konfiguriert.



ClientX neu starten und nochmals WINRM QUICKCONFIG ausführen. Super es hat geklappt. Auf ClientX haben wir WINRM erfolgreich eingerichtet.

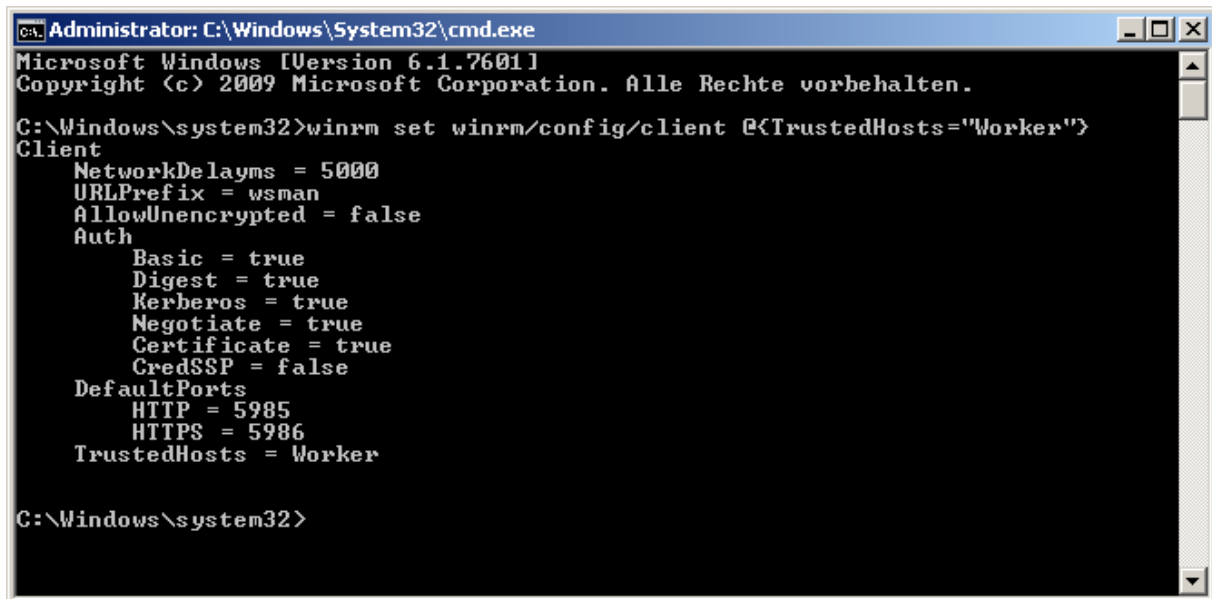


Es fehlt aber immer noch die Vertrauensstellung zwischen den beiden Maschinen.

Dazu führen wir auf ClientX nun den Befehl aus:

```
winrm set winrm/config/client @{TrustedHosts="Worker"}
```

WINRM WINRS



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>winrm set winrm/config/client @{TrustedHosts="Worker"}
Client
  NetworkDelayms = 5000
  URLPrefix = wsman
  AllowUnencrypted = false
  Auth
    Basic = true
    Digest = true
    Kerberos = true
    Negotiate = true
    Certificate = true
    CredSSP = false
  DefaultPorts
    HTTP = 5985
    HTTPS = 5986
  TrustedHosts = Worker

C:\Windows\system32>
```

Und umgekehrt genauso.



```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. Alle Rechte vorbehalten.

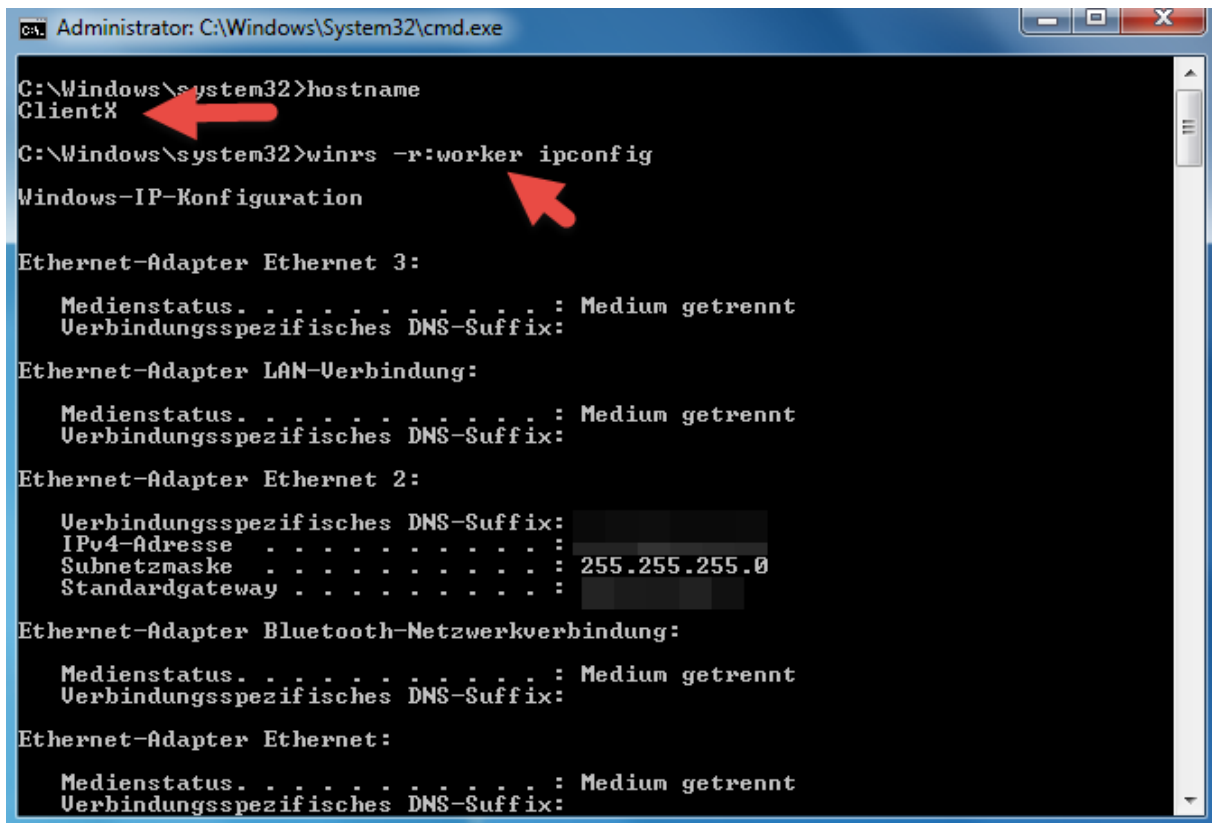
C:\Windows\system32>hostname
Worker

C:\Windows\system32>winrm set winrm/config/client @{TrustedHosts="ClientX"}
Client
  NetworkDelayms = 5000
  URLPrefix = wsman
  AllowUnencrypted = false
  Auth
    Basic = true
    Digest = true
    Kerberos = true
    Negotiate = true
    Certificate = true
    CredSSP = false
  DefaultPorts
    HTTP = 5985
    HTTPS = 5986
  TrustedHosts = ClientX

C:\Windows\system32>
```

WINRM WINRS

Jetzt frage ich noch einmal per Remote die IP Adresse vom Client Worker ab.



```
Administrator: C:\Windows\System32\cmd.exe

C:\Windows\system32>hostname
ClientX

C:\Windows\system32>winrs -r:worker ipconfig
Windows-IP-Konfiguration

Ethernet-Adapter Ethernet 3:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

Ethernet-Adapter LAN-Verbindung:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

Ethernet-Adapter Ethernet 2:

    Verbindungsspezifisches DNS-Suffix:
    IPv4-Adresse . . . . . :
    Subnetzmaske . . . . . : 255.255.255.0
    Standardgateway . . . . . :

Ethernet-Adapter Bluetooth-Netzwerkverbindung:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

Ethernet-Adapter Ethernet:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:
```

Alles gut!

Optional:

```
winrm create winrm/config/listener?Address=*+Transport=HTTP
winrm create winrm/config/listener?Address=*+Transport=HTTPS
winrm set winrm/config/client/auth @{Basic="true"}
```