Windows | offene Ports mit PowerShell suchen

Inhaltsverzeichnis

Was wollen wir ?

Offene Ports unter Windows mit Boardmitteln identifizieren.

Warum wollen wir das?

Zu Programm-Funktions- und Sicherheitszwecken.

Und wie geht das genau?

Um offenen Ports in Windows zu suchen, benötigt man nicht immer komplexe oder teure Zusatztools.

Mit der PowerShell und auch mit der Eingabeaufforderung lassen sich mit wenigen, kurzen Befehlen die notwendigen Informationen auslesen.

Mit der PowerShell ist es möglich geöffnete Ports und Verbindungen zu testen, genauso wie mit der Befehlszeile.

Man kann analysieren, wo Ports geöffnet sind und von welchen Programmen oder Diensten.

Für diese Abfragen sind keine zusätzlichen Module oder Anwendungen notwendig.

Das wichtigste, interne Modul für die Abfrage und das Setzen von IP-Informationen ist das Modul **NetTCPIP**.

Die Cmdlets aus diesem Modul zeigt die PowerShell mit dem folgenden Befehl an:

Code

```
Get-Command -modul NetTCPIP
```

1 Offene Ports lassen sich in sekundenschnelle identifizieren

Ein Beispiel dafür ist das Testen von Endpunkten, die auf einem Rechner im Netzwerk offen sind.

Kommt zum Beispiel ein Client für Multicast DNS (mDNS) zum Einsatz, ist auf dem Rechner der UDP-Port 5353 offen.

Dieser Port kann speziell getestet werden, genauso wie andere Ports. Das CMDlet dazu ist:

Code

```
Get-NetUDPEndpoint -LocalPort 5353
```

Mit diesem Cmdlet ist sofort klar, auf welchen IP-Adressen eines Computers der Port 5353 offen ist.

Sollen auch noch verifiziert werden, welche Anwendungen diesen Port nutzen,

kann der Befehl mit typischen PowerShell-Parametern auch erweitert werden:

Code

```
Get-NetUDPEndpoint -LocalPort 5353 | Select-Object LocalAddress,LocalPort,OwningProcess,@{ N
```

Mit PowerShell-Cmdlets und Skripts können Netzwerkprobleme eingegrenzt und behoben werden.

Es lohnt sich die Möglichkeiten der verschiedenen Netzwerk-Cmdlets zu kennen, um Fehlerquellen zu erkennen und Probleme zu lösen.

Mit dem gleichen Cmdlet ist es auch möglich alle offenen UDP-Ports auf allen IP-Adressen eines Rechners anzuzeigen:

Code

```
Get-NetUDPEndpoint
```

Auch das Abfragen der einzelnen IP-Adressen ist möglich, zum Beispiel mit:

Code

```
Get-NetUDPEndpoint -LocalAddress 127.0.0.1
```

2 Informationen zu Netzwerkadaptern anzeigen

Für das Abfragen von Ports sollten natürlich zunächst die lokalen IP-Adressen und die Netzwerkadapter bekannt sein.
Dazu stehen in der PowerShell die beiden Cmdlets "**Get-NetIPAddress**" und "**Get-NetIPConfiguration**" zur Verfügung,
um Informationen zur Netzwerkkonfiguration eines Computers anzuzeigen.

Mit dem Parameter „**-Verbose**“ zeigt die PowerShell noch mehr Informationen an.

Mit den beiden Cmdlets sind alle Informationen, die über die lokale Netzwerkkonfiguration eines Rechners zur Verfügung stehen abrufbar.

Um noch mehr Daten zu den verbauten Netzwerkadaptern anzuzeigen, wird noch das Cmdlet "**Get-NetAdapter**" verwendet.

Alle diese Cmdlets ermöglichen auch das Weiterleiten der Informationen zu Format-List oder Format-Table, zum Beispiel:

Code

```
Get-NetIPAddress | Format-Table
```

3 Verbindungen zu IP-Adressen und Ports testen

Mit dem Cmdlet "**Test-NetConnection**" lassen sich Verbindungen zwischen IP-Adressen und auch zwischen den verwendeten Ports testen.

Das Tool kommt vor allem dann zum Einsatz, wenn eine Netzwerkverbindung zwischen zwei IP-Adressen getestet werden soll.

Das funktioniert mit IP-Adressen und auch mit Namen, wenn die Namensauflösung im Netzwerk richtig konfiguriert ist:

Code

```
Test-NetConnection 192.168.1.1
```

Mit diesem Cmdlet ist es auch möglich eine Verbindung zu einem bestimmten Port

auf dem entfernen Computer zu öffnen, um zu testen, ob dieser geöffnet ist.

Dazu kommt der Parameter „**-Port**“ zum Einsatz, zum Beispiel:

Code

```
Test-NetConnection 192.168.1.1 -Port 443
```

Ausführlichere Informationen zeigt der folgende Befehl an:

Code

```
Test-NetConnection -InformationLevel "Detailed"
```

Soll die Route zu einem Computer diagnostiziert werden, ist folgender Befehl nützlich:

Code

```
Test-NetConnection -ComputerName www.contoso.com -DiagnoseRouting -InformationLevel Detailed
```

Disclaimer:

Alle Anleitungen/Tutorials sind nach bestem Wissen und Gewissen verfasst, gehen immer von den definierten Software/Firmware-Versionen aus und sind auf das englische GUI ausgelegt.

Es gibt keine Garantie auf Erfolg. Im Falle eines Misserfolges hilft aber die Community hier sicherlich weiter.

Keiner der Autoren oder der Betreiber des Forums ist für die aus der Nutzung resultierenden Probleme/Herausforderungen verantwortlich.

Jegliche hier beschriebenen Schritte erfolgen ausnahmslos in eigener Verantwortung des Durchführenden.

Eltern haften für ihre Kinder.

Auswählen:

Gültige Software-Version Keine Firmware-Relevanz!