Inhalt

Einleitung

Zielsetzung

Meine Serverumstellung auf Windows Server 2019 geht in die nächste Runde. Dieses Mal sind die beiden Server WS-RA1 und WS-RA2 dran. Beide laufen aktuell unter Windows Server 2016 als virtuelle Maschinen. Im folgenden Abschnitt prüfe ich, welche Services auf den Servern laufen und wie ich diese migrieren werde.

Bereitgestellte Services

Web Application Proxy (WAP) & Active Directory Federation Service (ADFS)

Die Umstellung auf den HAProxy habe ich bereits im Oktober durchgeführt und in einem anderen Artikel beschrieben. Diese spielt hier also keine Rolle mehr.

Network Policy Service (NPS)

Dazu stellt der Server WS-RA1 noch einen Network Policy Service (**NPS** – auch als Radius Server bekannt) bereit. Diesen nutzt ein WLAN-Accesspoint für WPA2-Enterprise-Anmeldungen meiner Clients. Die Funktion wird weiter benötigt und muss daher auf einen neuen Server migriert werden. Dabei halte ich mir eine Erweiterung auf eine hochverfügbare Lösung offen.

Die Migration wird mittels Wipe & Load vorgenommen, da ich aktuell keine Hochverfügbarkeitsanforderung gestellt habe. Für den Wechsel ist eine Downtime erforderlich.

VPN-Service

Die Namen der beiden Server habe ich aus dem Servicenamen RemoteAccess abgeleitet. Ich nutzte die Server als VPN-Server für die Einwahl von extern.

Die Formulierung in der Vergangenheitsform deutet es schon an: Ich nutze seit Ewigkeiten kein VPN mehr für die Arbeiten von außen. Diese Funktion bilde ich über meine Remote Desktop Services dank des RD-Gateways ab. Der Service VPN wird also nicht mehr benötigt und kann einfach entfernt werden.

Planung der Migration

Damit sind die Arbeitsschritte für die komplette Migration klar:

- Schritte im vorherigen Artikel
    - Zuerst entferne ich alle nicht mehr benötigten Services und deren Konfigurationen in der richtigen Reihenfolge.
- Schritte in diesem Artikel
    - Danach migriere ich den Service NPS auf einen neuen Windows Server 2019 mit dem Namen WS-NPS1.
    - Zuletzt entferne ich die beiden alten Server aus meiner Infrastruktur..

Für die Migration des NPS werde ich den neuen Server neben dem alten synchron aufbauen. Der eigentliche Austausch wird durch die Übergabe der alten IPv4-Konfiguration an den neuen Server vorgenommen. Denn nur über diese IPv4 findet der WLAN-AccessPoint den NPS-Server. Damit spare ich mir die Rekonfiguration des WLAN-AccessPoints und die Anpassung der Firewall-Ausnahmen. Und ich könnte auch schnell wieder auf den alten Server zurückschwenken, indem ich die IP-Änderung wieder zurücknehme. Ein Rollback-Szenario ist immer gut.

Die Vorarbeiten sind bereits abgeschlossen. Dazu zählt die Entfernung des Web Application Proxy Clusters. Diese habe ich in einem anderen Artikel beschrieben. In diesem geht es daher nur noch um die Entfernung der beiden alten Server und um die Migration meines Network Protection Services (NPS)

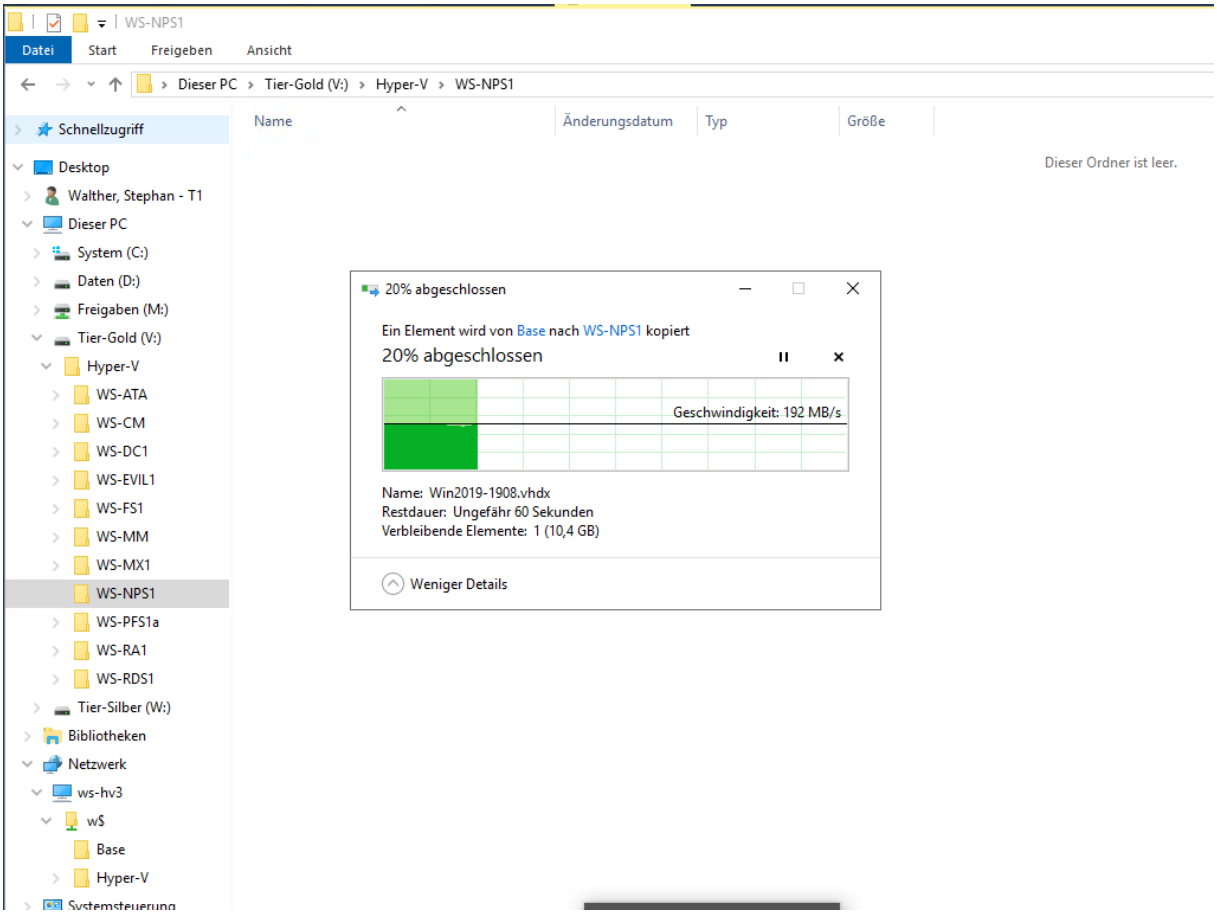
Migration NPS

Aufbau der neuen VM

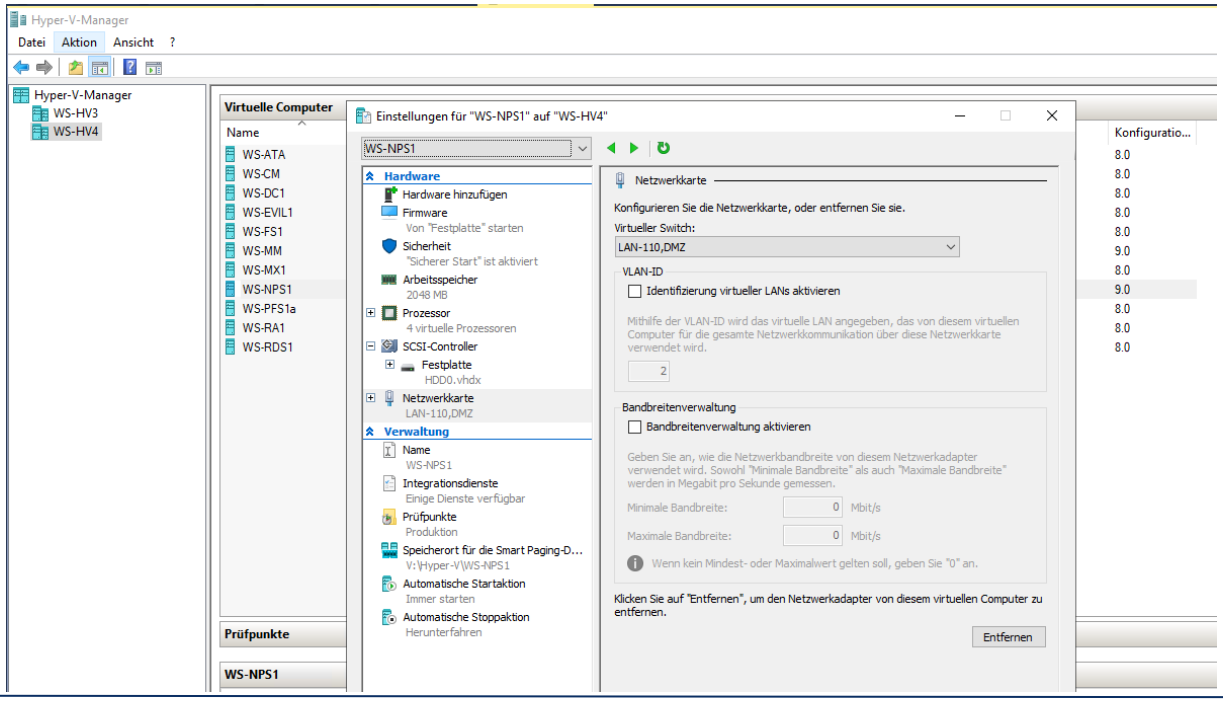
WAP, ADFS und VPN sind bereits entfernt. So bleibt nur noch die Rolle NPS. Diese möchte ich auf einen neuen Server migrieren. Den Namen leite ich aus der Funktion ab: WS-NPS1. Mit der Ziffer 1 halte ich die Option einer späteren

Skalierung um einen weiteren Server WS-NPS2 offen. Damit könnte ich einen hochverfügbaren Radius-Service konfigurieren.

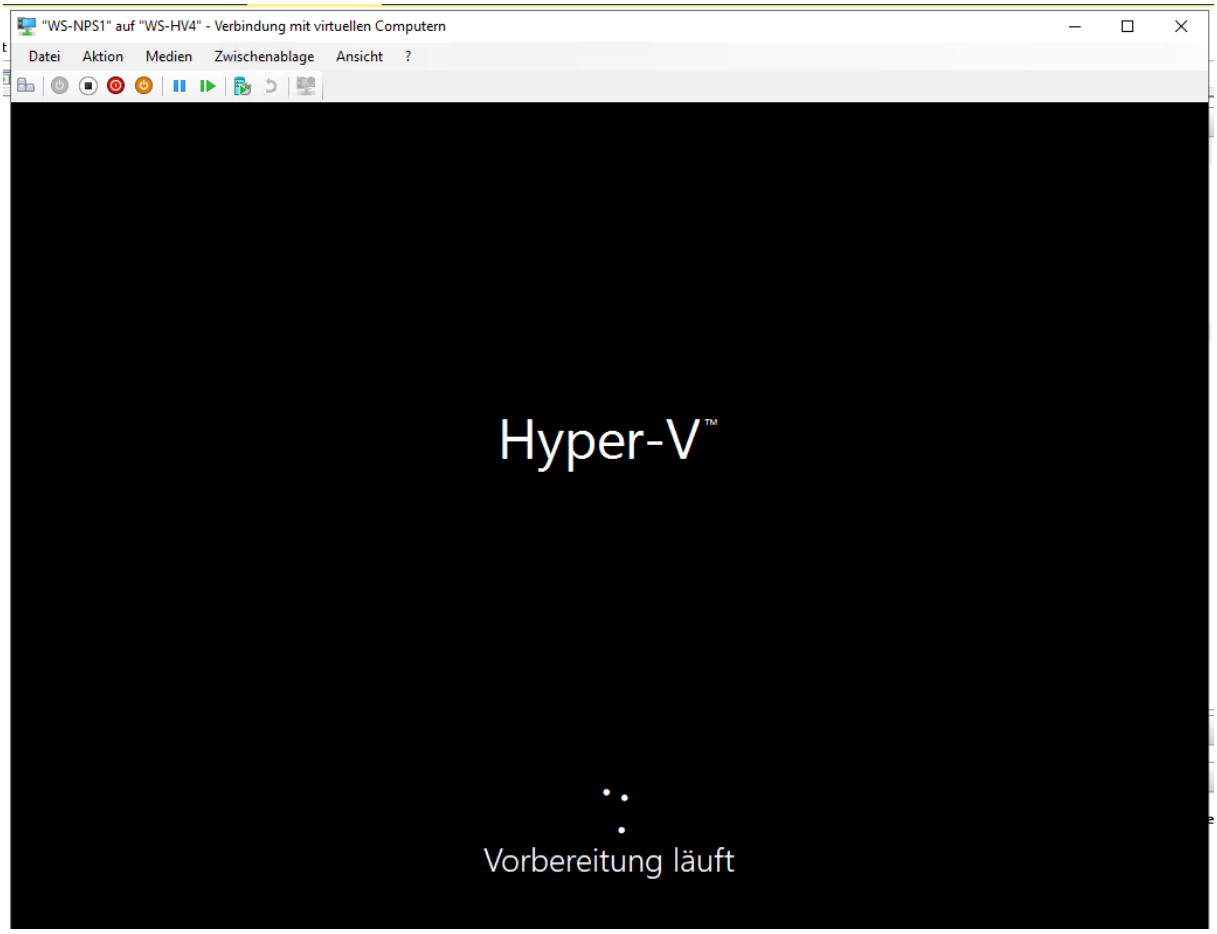
Den neuen Server baue ich als VM in meinem Hyper-V-Host auf. Dazu kopiere ich zuerst ein Basefile (eine VHDX mit einem vorbereiteten Betriebssystem) in das VM-Verzeichnis. Diese VHDX enthält Windows Server 2019 mit der grafischen Oberfläche. Leider kann laut Microsoft der NPS-Service nicht auf einem Server Core betrieben werden:



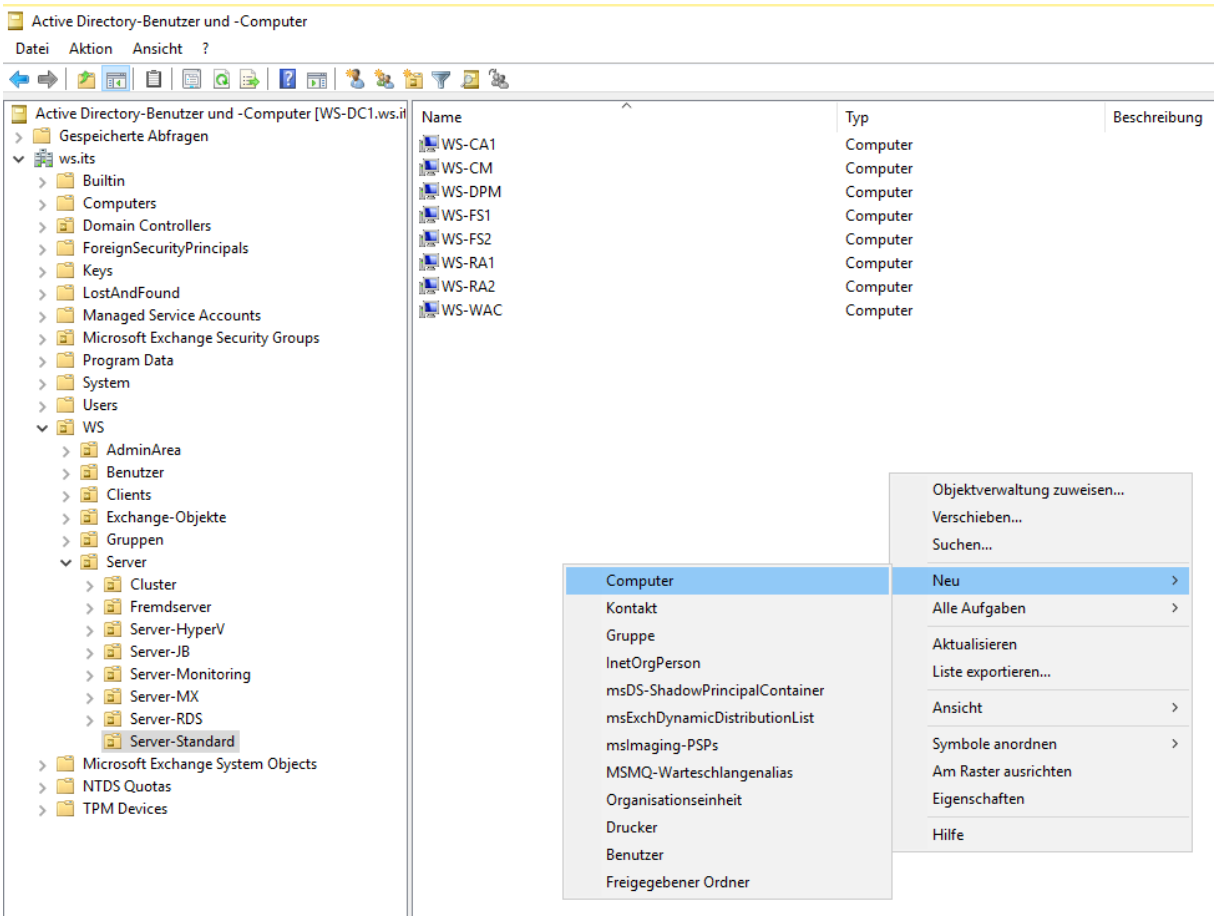
Dann erstelle ich eine neue VM mit den passenden Spezifikationen. Der NPS-Service braucht nicht viel:



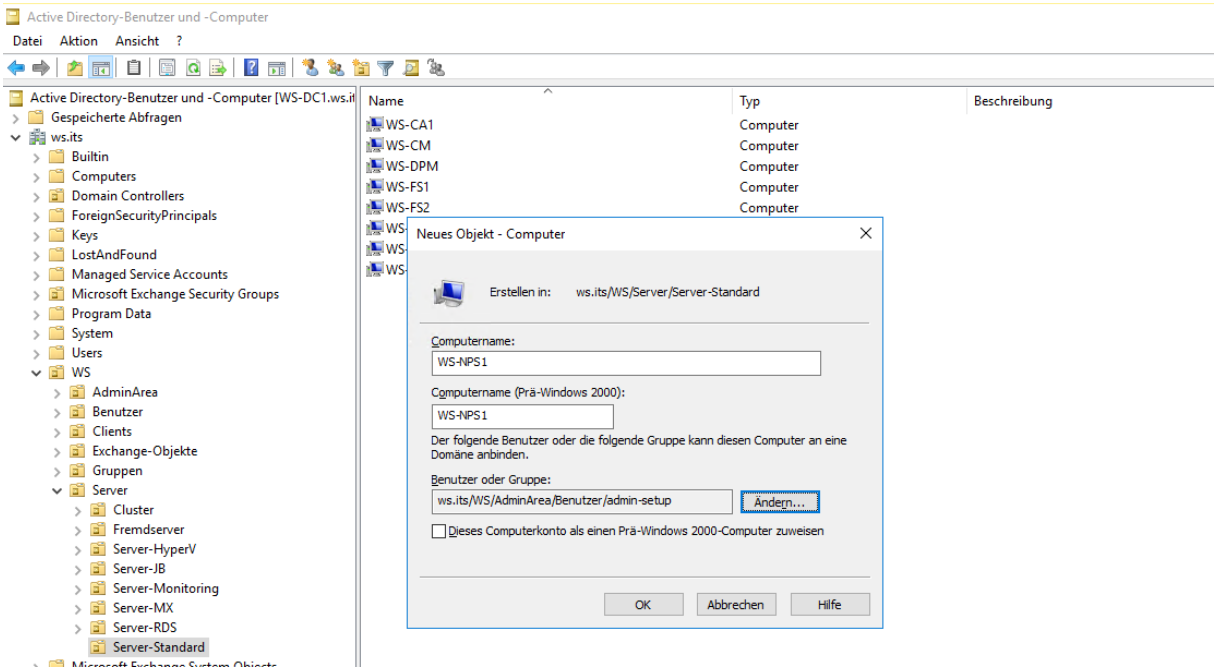
Dann bekommt die VM ihr Startsignal. Windows Server 2019 beginnt seine Erkennung und Ersteinrichtung:



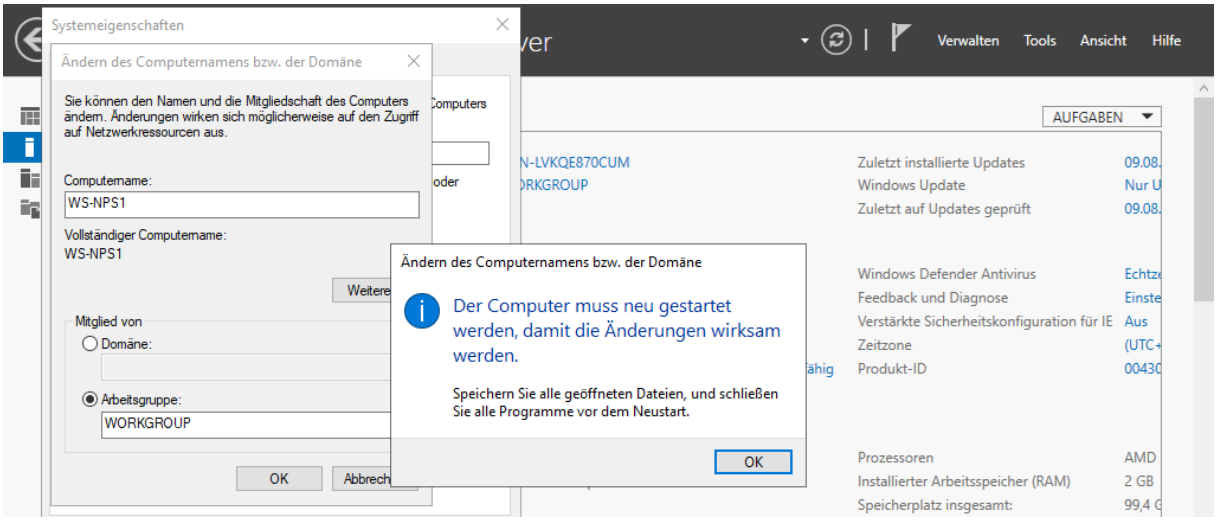
Währen dessen bereite ich ein neues Computerobjekt im Active Directory vor:



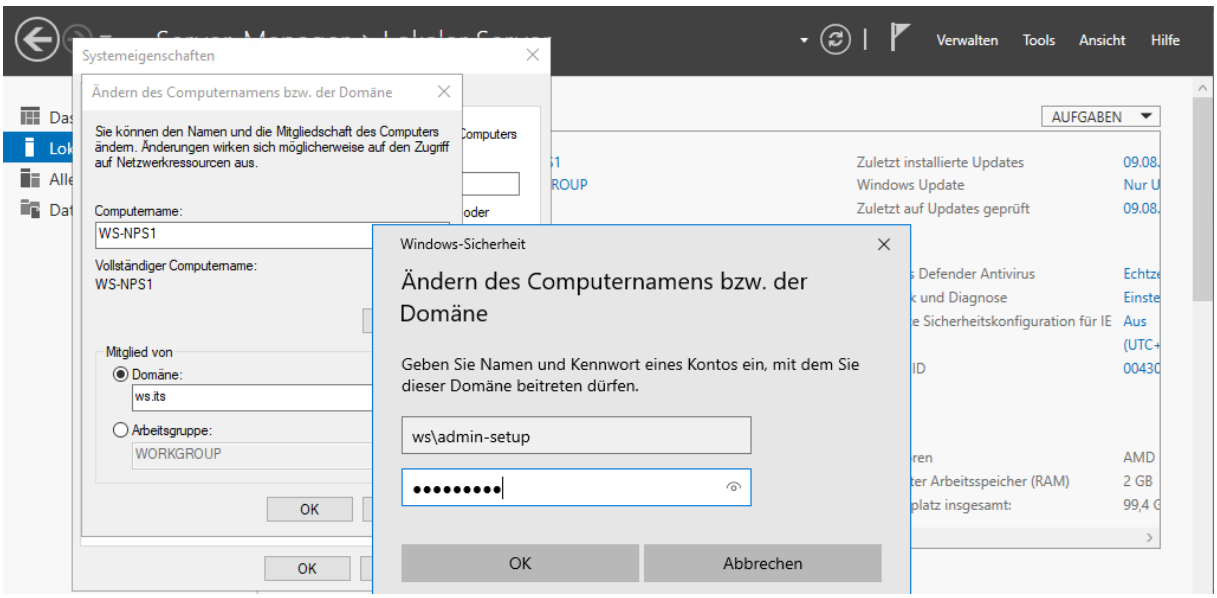
Für den Domain Join delegiere ich das Recht an einen Setup-Adminaccount:



Nach der Eingabe eines Passwortes für den lokalen Admin kann der Server konfiguriert werden. Ich benenne das Betriebssystem um:



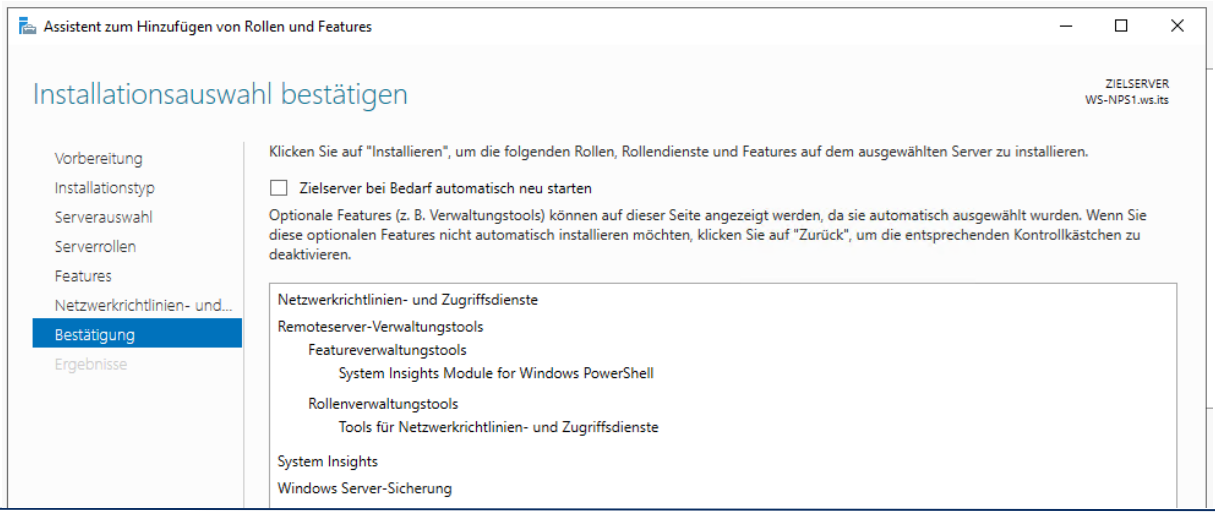
Und nach dem Neustart darf der Server der Domain beitreten:



Das sind alles Standardaufgaben. Eigentlich gehört auch die IP-Konfiguration dazu. Aber diese kommt für den Schwenk des NPS erst später.

Installation der Rollen und Features

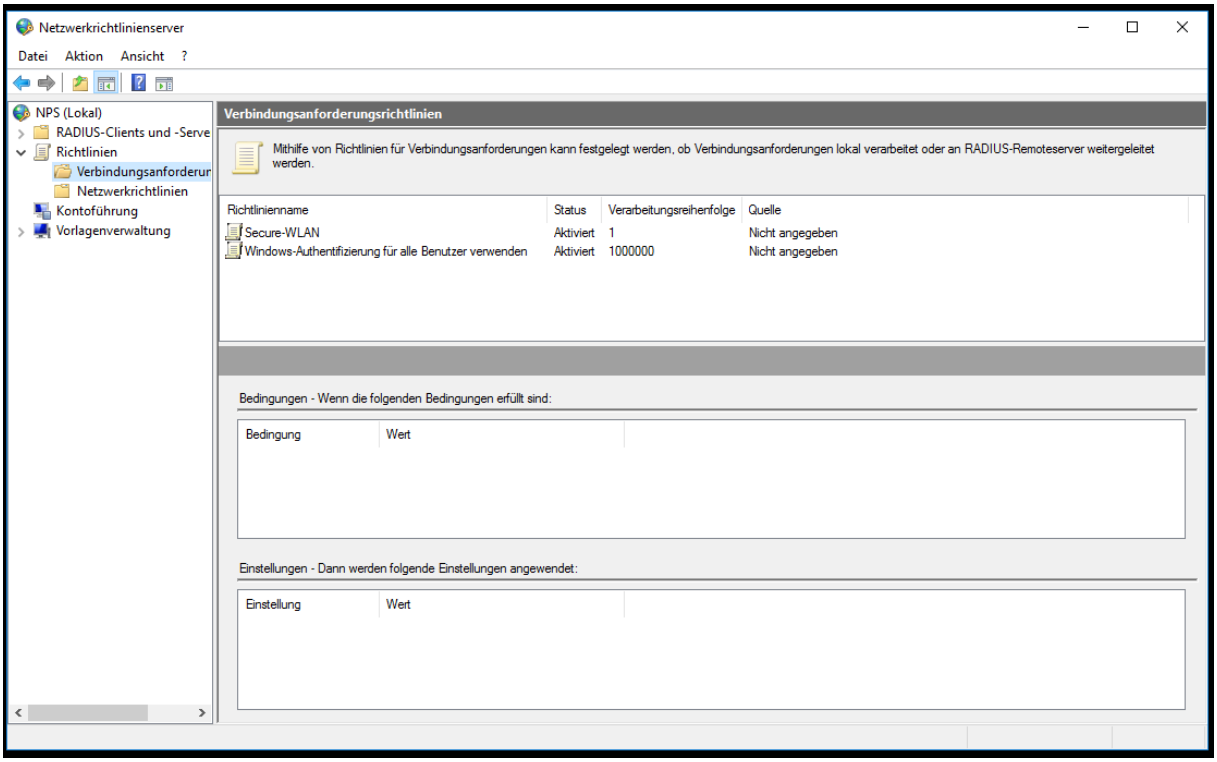
Nun installiere ich die Rollen und Features. Das ist ebenfalls eine Standardaufgabe:



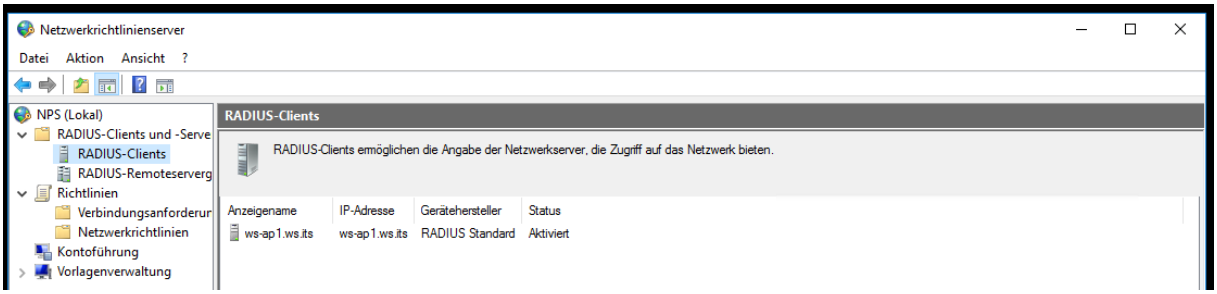
Das war kein Problem.

Migration der Rolle NPS

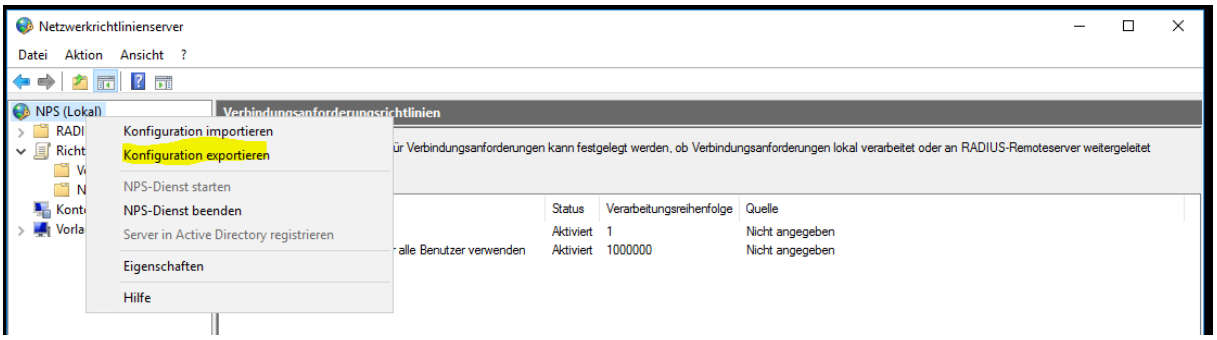
In meinem alten NPS-Server gibt es die Konfigurationen in der Konsole zu sehen. Bei mir ist es nicht viel. Das könnte ich eigentlich sogar einfach im neuen Server abtippen. Aber in großen Umgebungen ist das keine Option. Also wird das professioneller gehen dürfen. Hier ist eine Regel für mein WLAN:



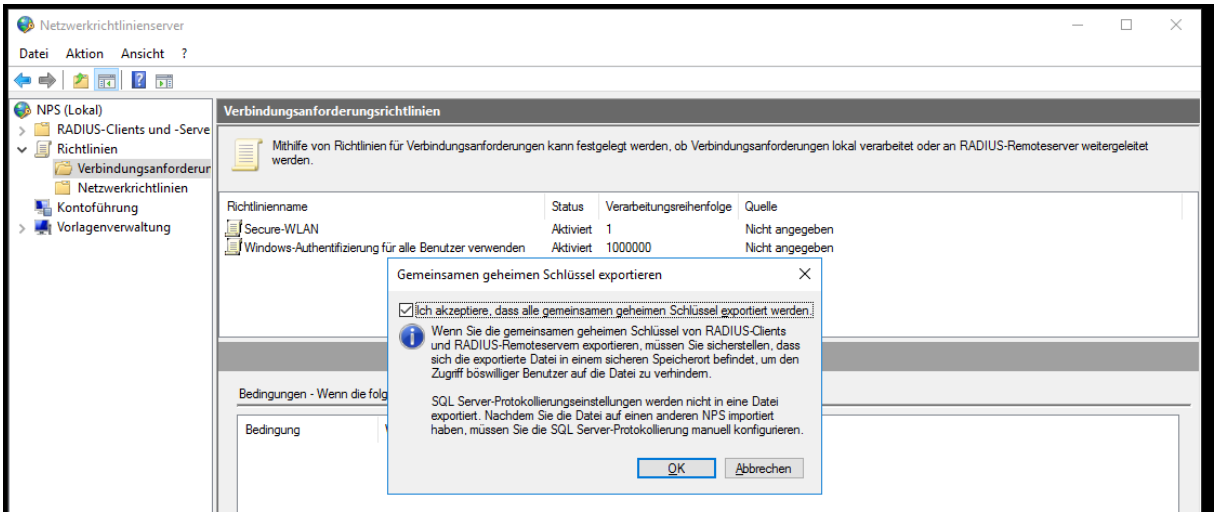
Und das ist der eine WLAN-AccessPoint:



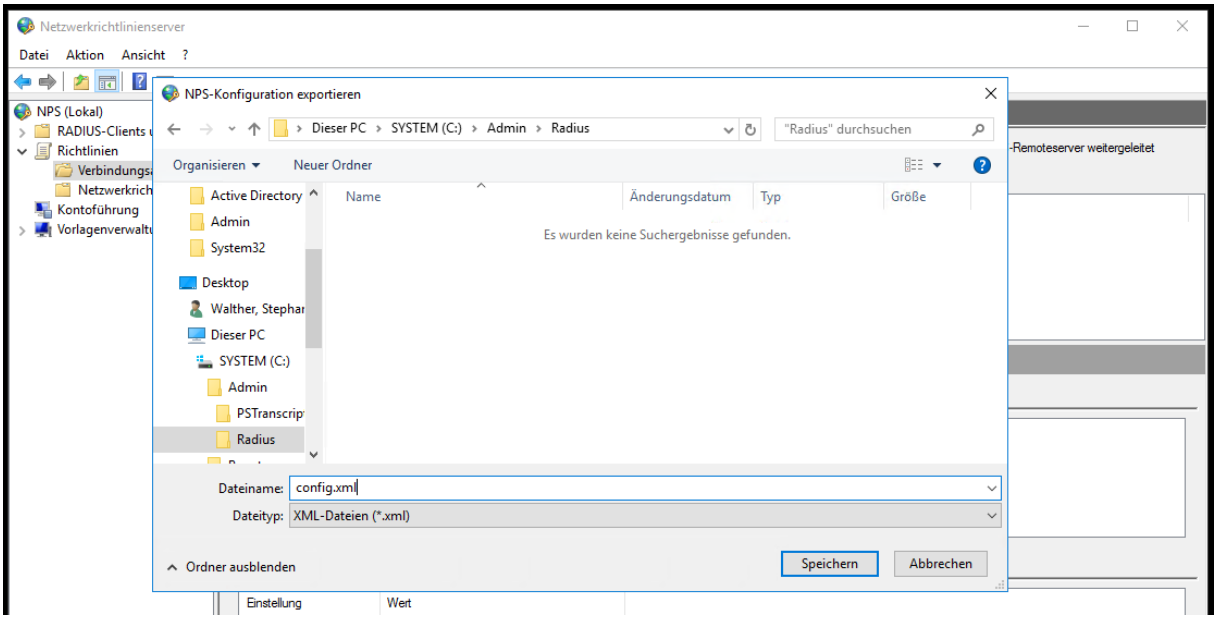
Microsoft hat eine Export-Funktion eingebaut. Diese nutze ich für den Transfer der Konfiguration. Wichtig ist aber, dass nach dem Export keine Änderungen am Altsystem mehr vorgenommen werden:



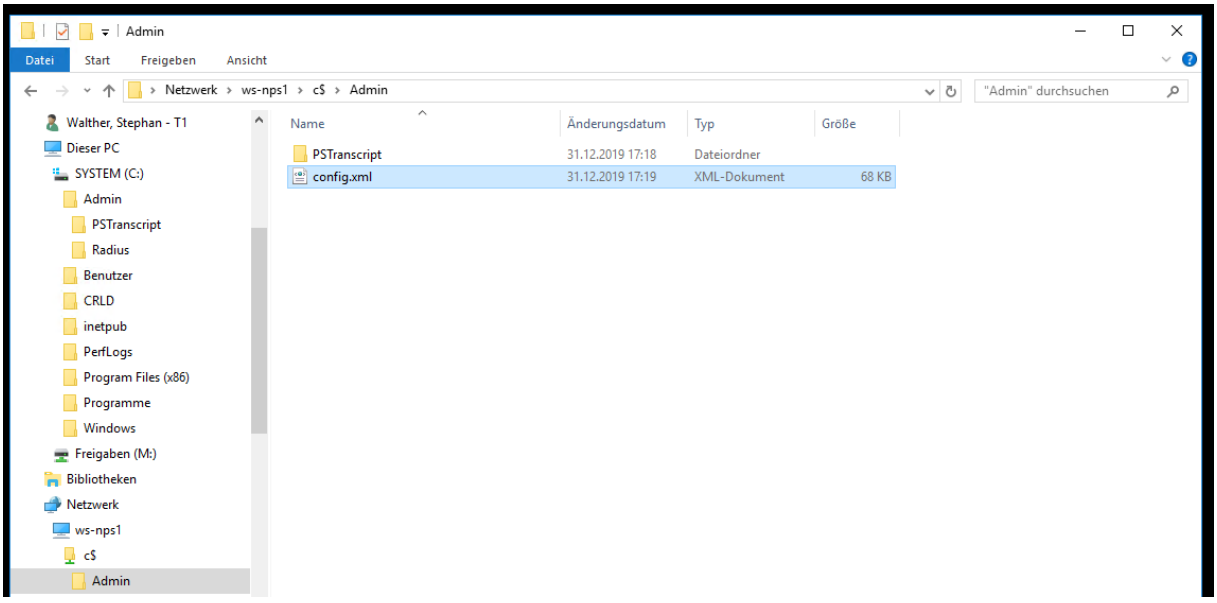
Zwischen den Radius-Clients und dem Server wird verschlüsselt kommuniziert. Der Schlüssel wird im Klartext in der Exportdatei liegen. Das ist bei mir aber kein Problem, da die Datei nur für Administratoren sichtbar sein wird, die auch den Export erstellen können:



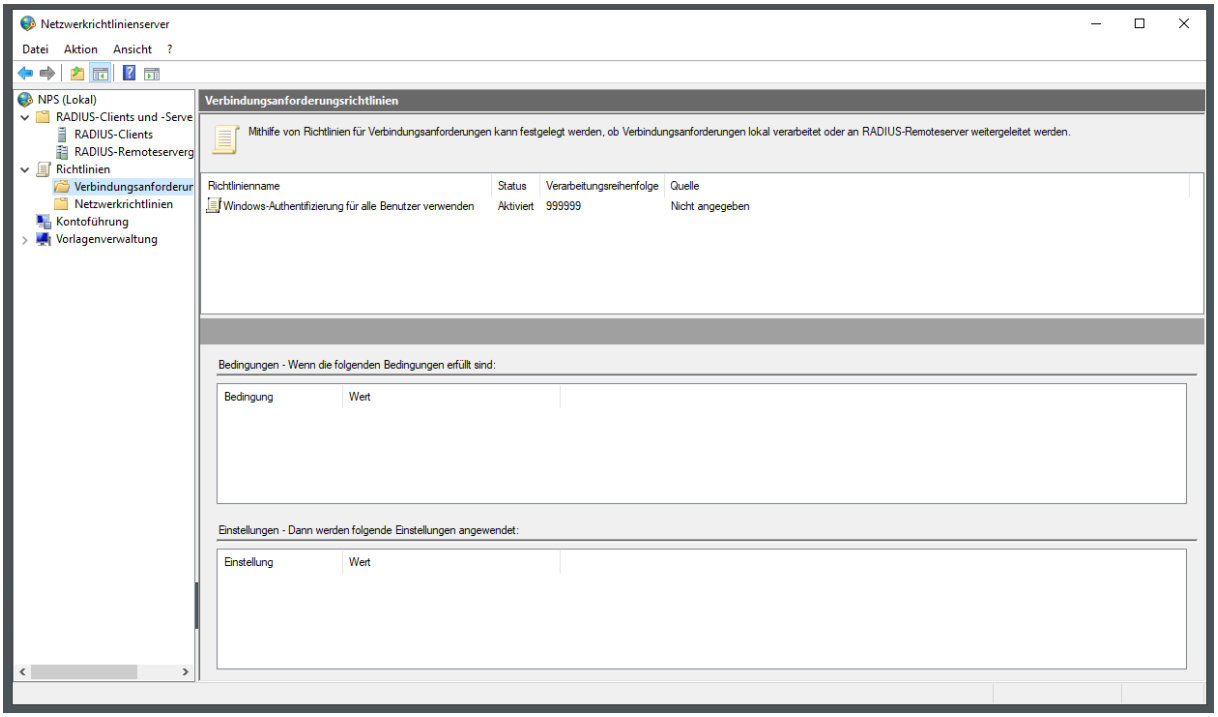
Der Assistent benötigt nur noch den Speicherpfad:

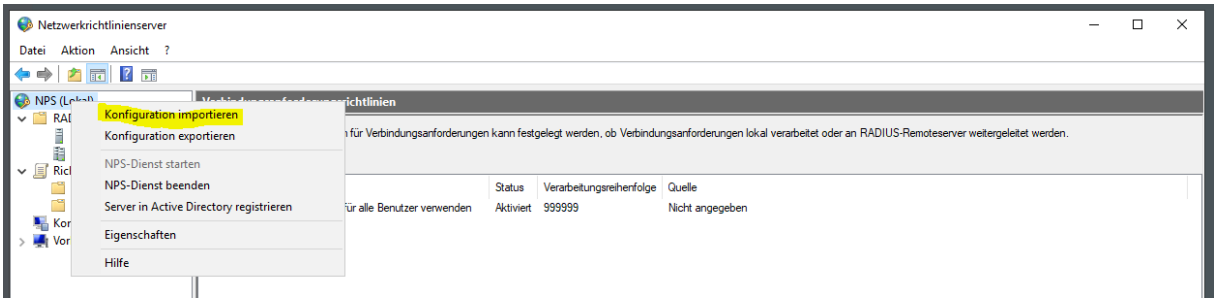


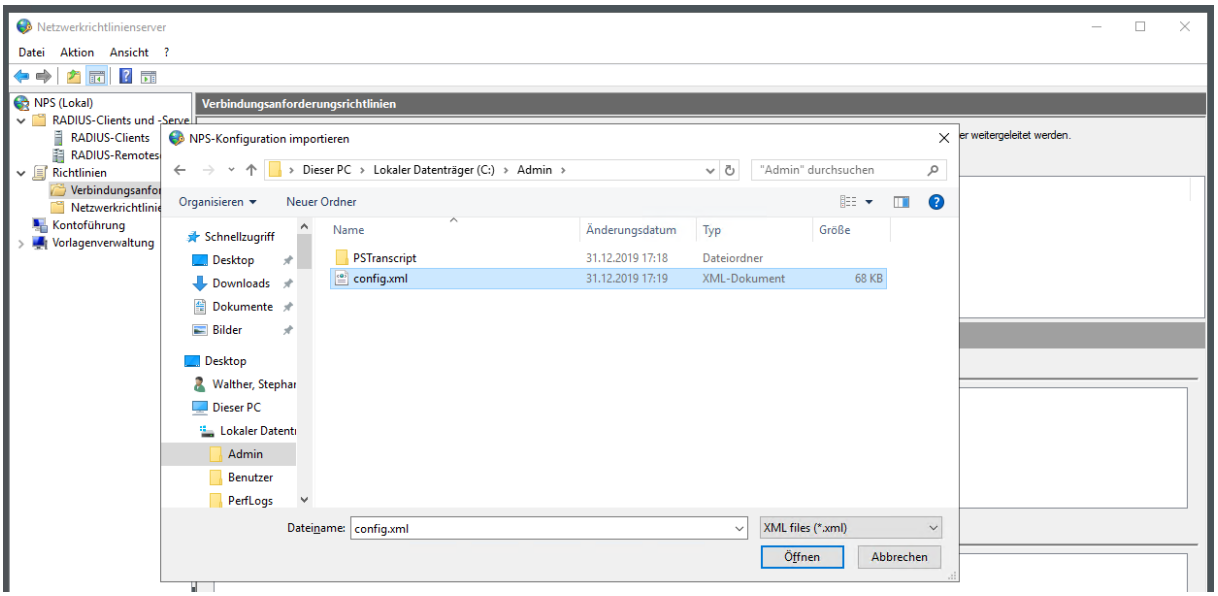
Die Daten werden als xml-Datei gespeichert:



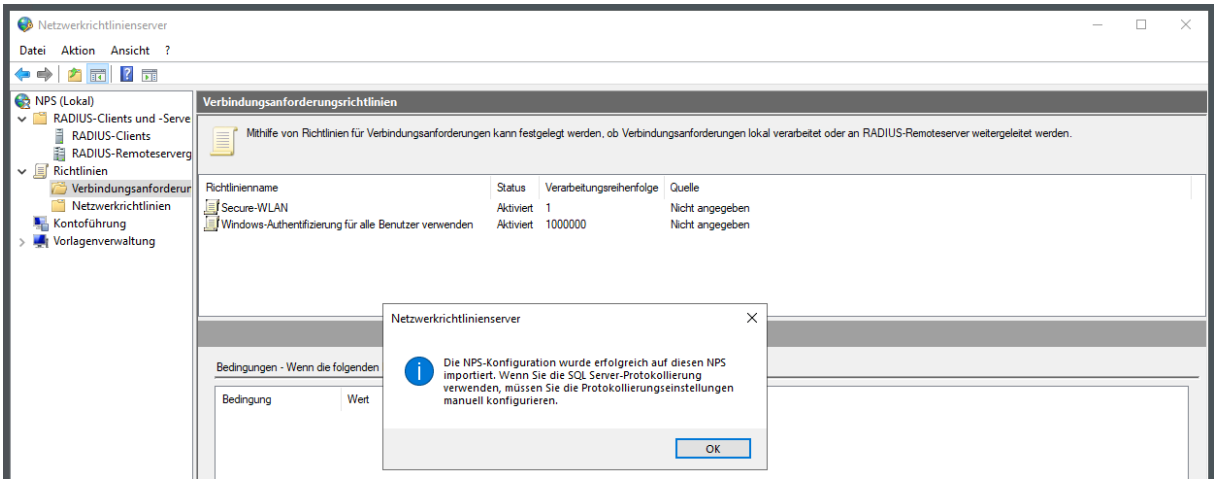
Während der alte NPS weiterläuft, importiere ich die xml-Datei im neuen Server. Hier hat sich mit Windows Server 2019 nicht wirklich etwas verändert. Daher erwarte ich auch eine entsprechende Kompatibilität der Exportdatei:



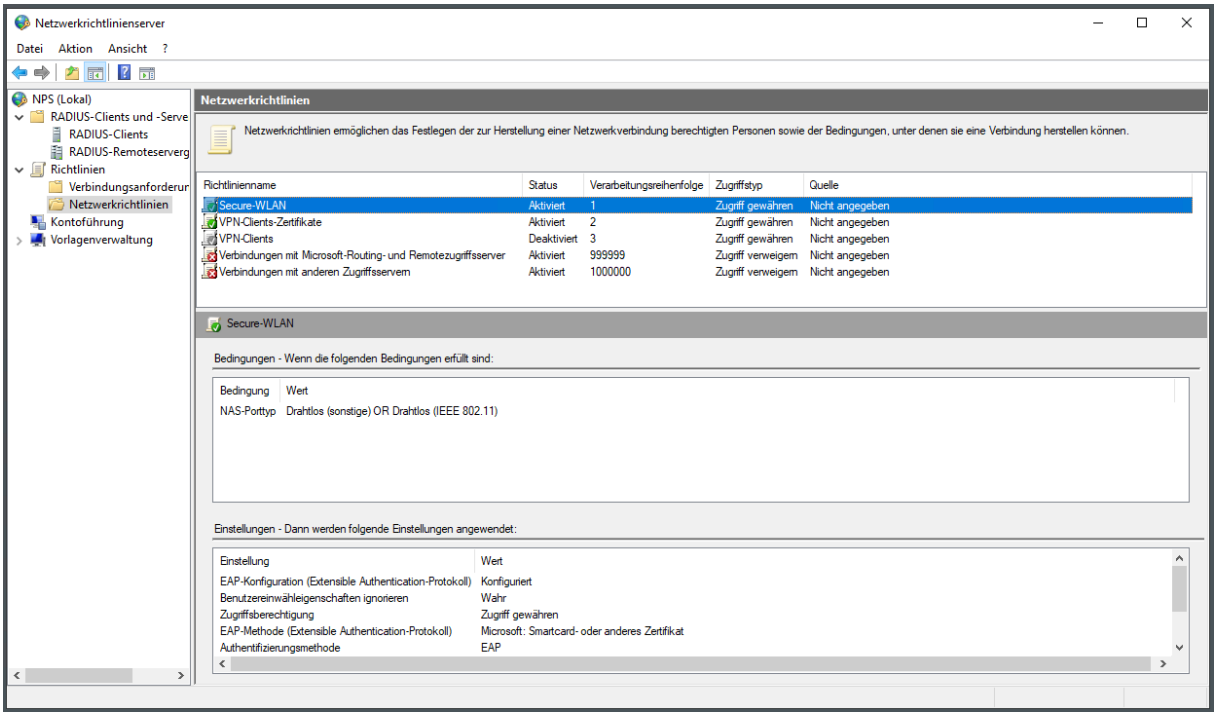




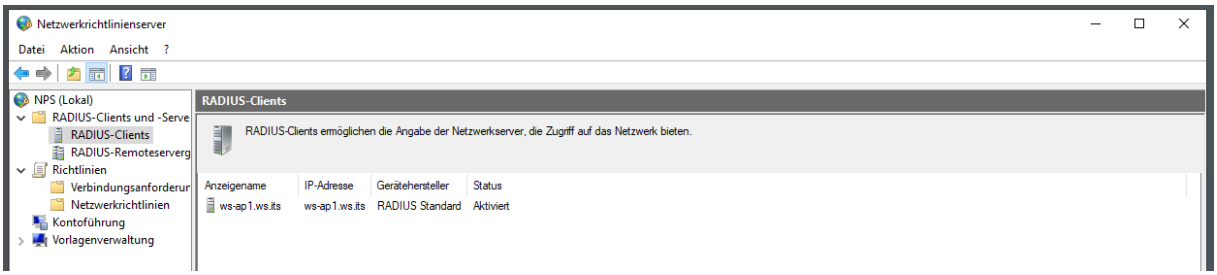
Dieser Schritt war sehr einfach. Die Bestätigung klingt vielversprechend:



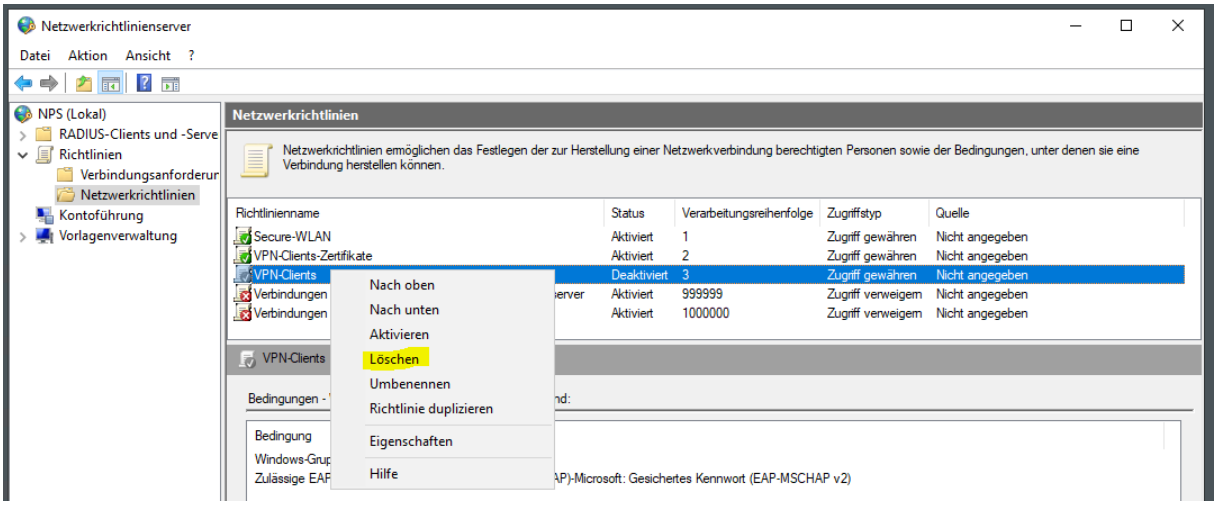
Meine Richtlinien sind alle angekommen:



Und auch der Radius-Client wird angezeigt:



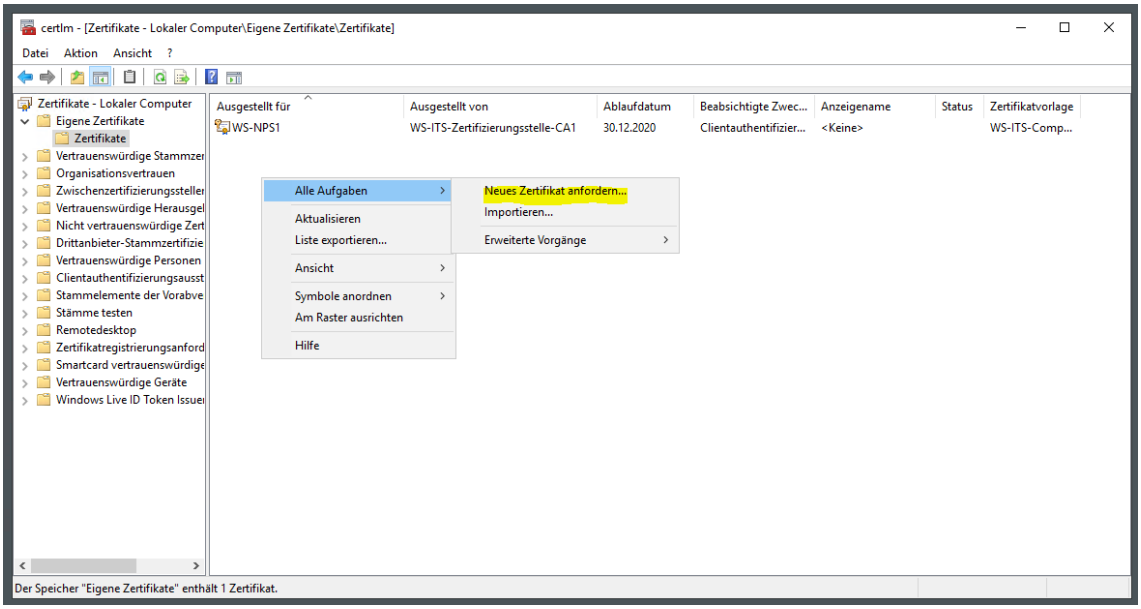
Bevor es jetzt in die Umstellungsphase geht nutze ich die Gelegenheit, um etwas aufzuräumen. In den Netzwerkrichtlinien ist noch die Richtlinie für den VPN-Service gelistet. Da ich diesen Service nicht mehr bereitstelle, kann ich die Regel löschen:



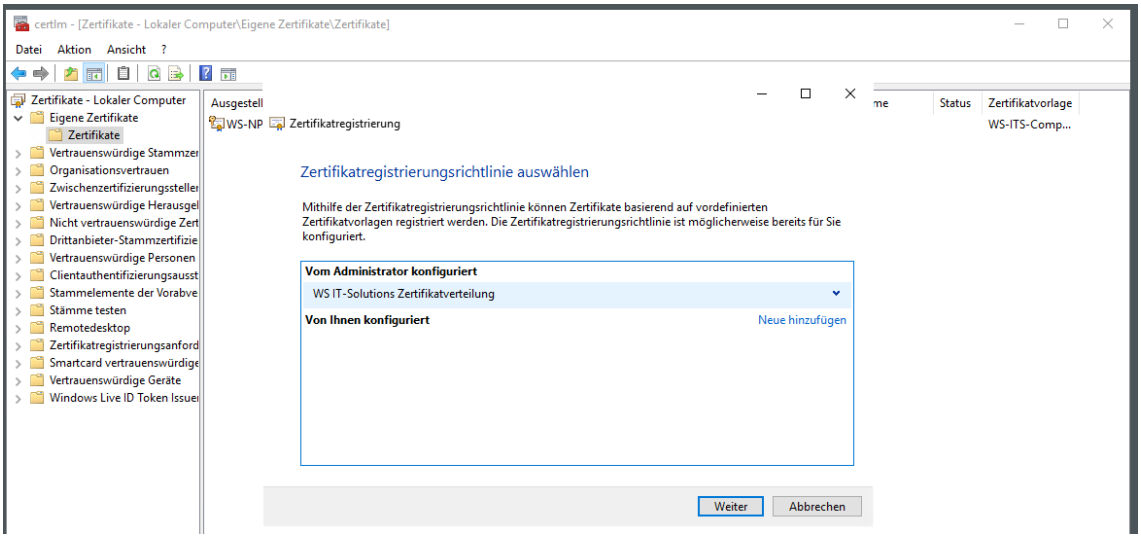
Damit habe ich die Funktionalität des NPS im Hintergrund kopiert. Für die Umstellung fehlt aber noch ein wichtiges Detail.

Konfiguration des Serverzertifikats

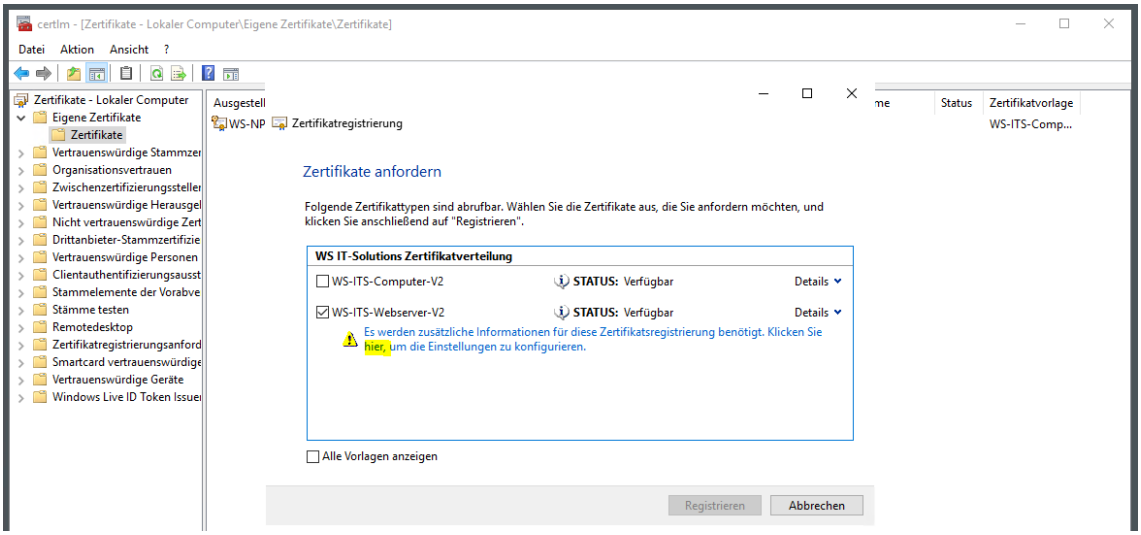
Dieses Detail ist das Zertifikat für den Radius-Server. In meiner Gruppenrichtlinie für die mobilen Clients habe ich für den internen WLAN-Zugriff die gegenseitige Authentifizierung angefordert: So muss nicht nur der Client seine Identität beweisen, sondern auch der Radius-Server. Und das kann er mit einem Sicherheitszertifikat. Durch mein PKI-Zertifikat-AutoEnrollment hat der Server WS-NPS1 bereits ein Clientauthentifizierungszertifikat. Für den NPS-Service benötige ich aber ein anderes. Dieses frage ich manuell in der Konsole certlm.msc an:



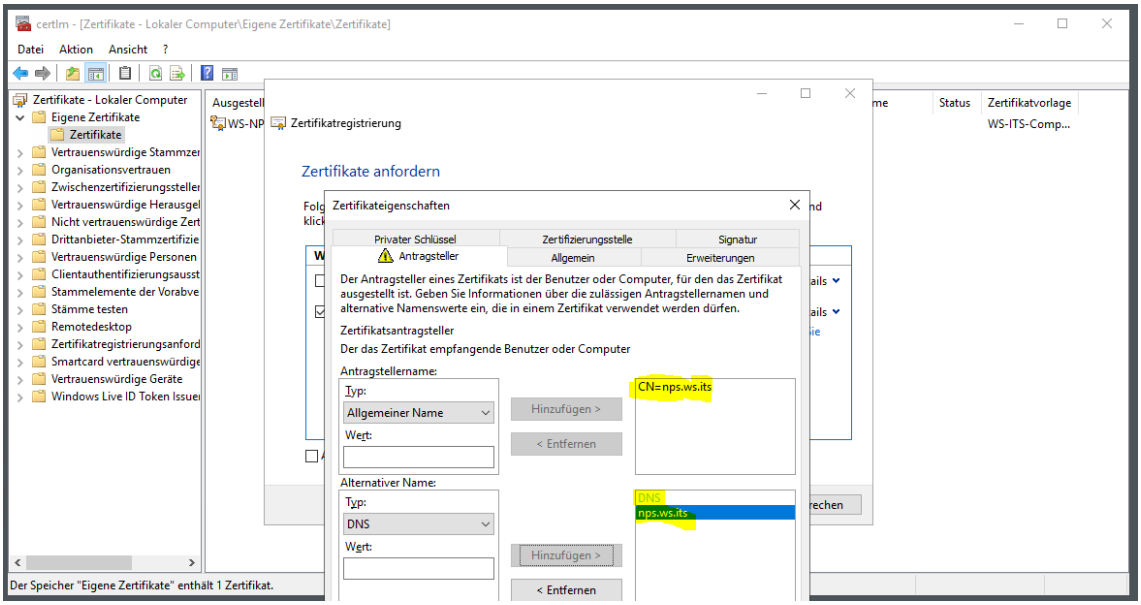
Der Assistent verbindet sich mit meiner PKI über den CEPCES-Endpunkt:



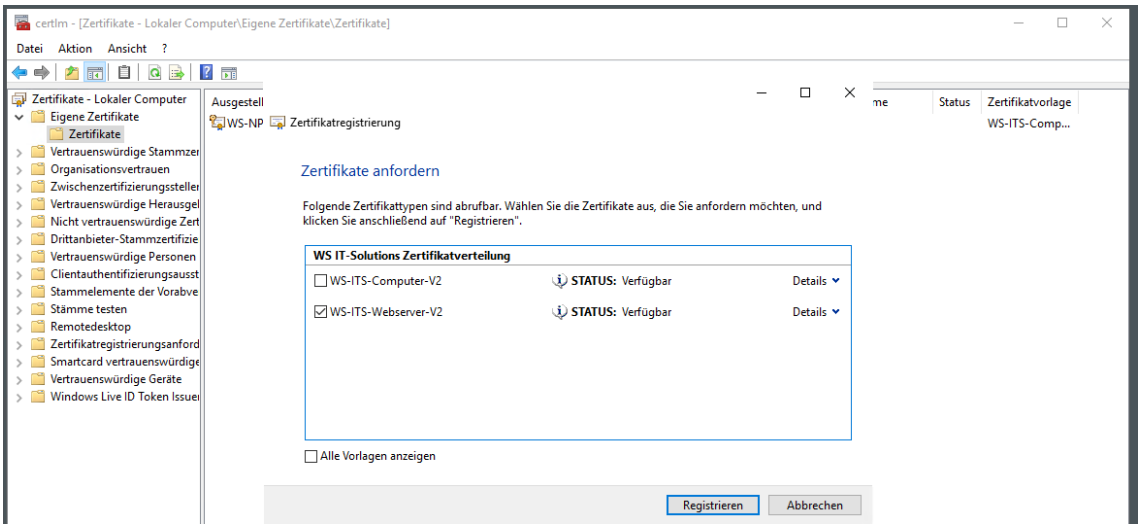
Für NPS benötige ich ein Webserver-Zertifikat. Dafür habe ich bereits eine Vorlage erstellt. Der Subject-Wert muss manuell eingegeben werden:



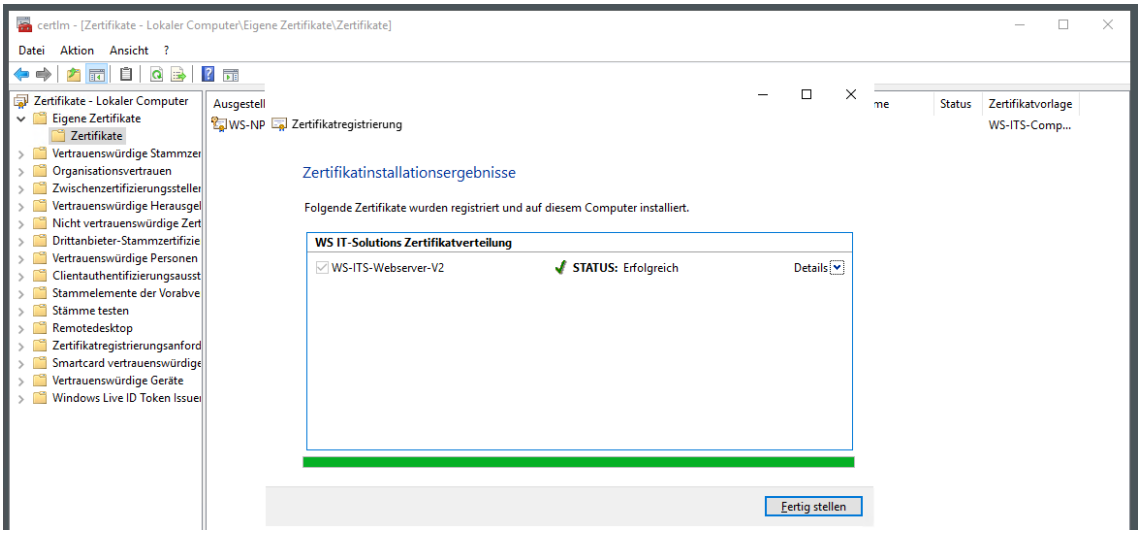
Ich benenne den CN mit einem Alias nps.ws.its:



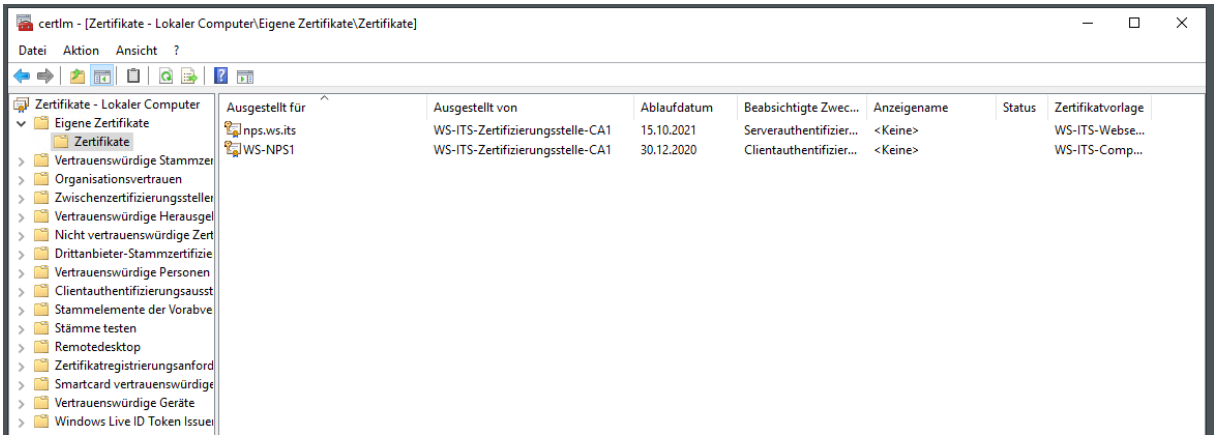
So ausgefüllt kann der Request zur PKI gesendet werden:



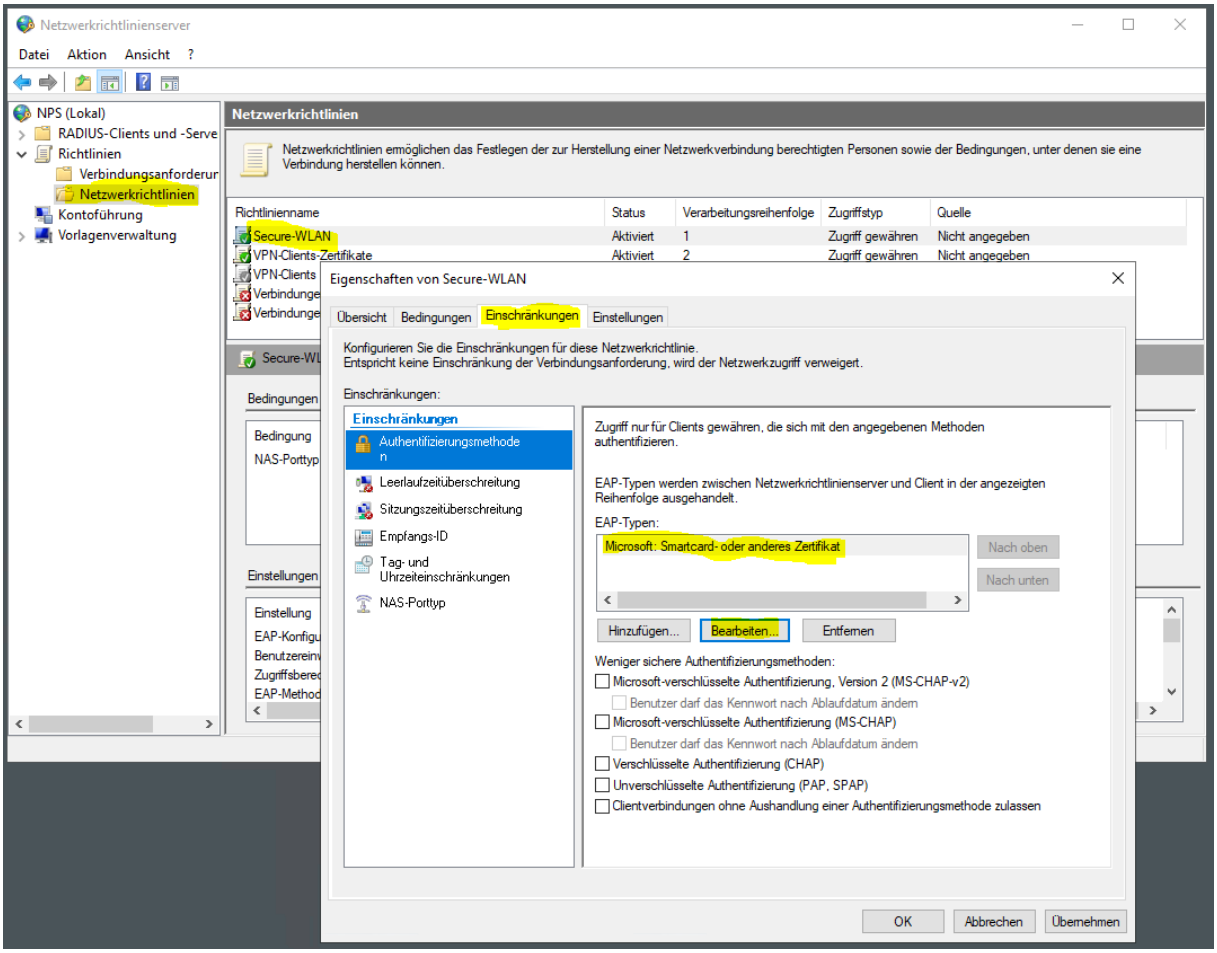
Und dort wird er auch direkt genehmigt:



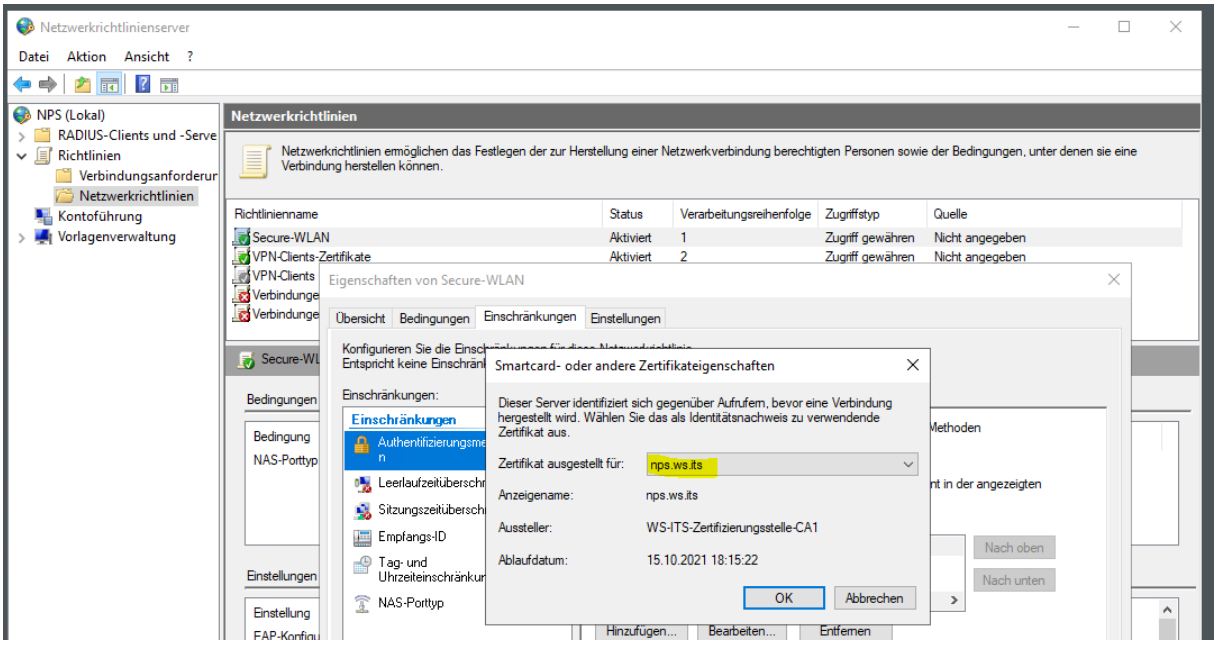
Das Zertifikat ist jetzt einsatzbereit:



Jetzt muss ich dem NPS noch mitteilen, dass er dieses neue Zertifikat verwenden soll. Dazu geht es in die NPS-Konsole in die Richtlinie für mein Secure-WLAN:



Der Server hat das Zertifikat sofort gefunden:



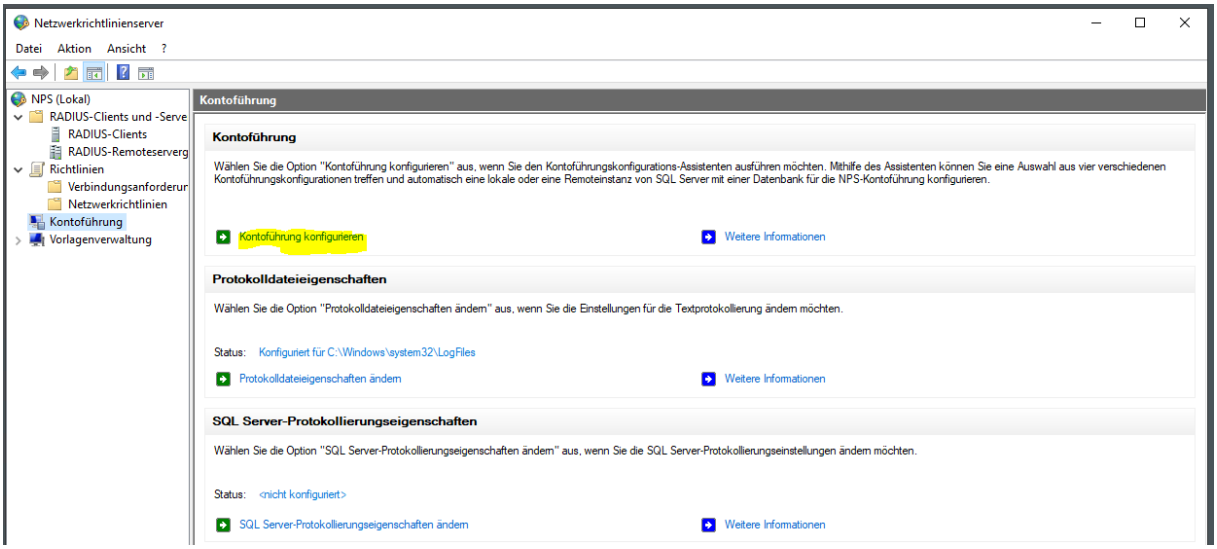
Damit steht dem Schwenk des NPS nur noch eine Kleinigkeit im Weg.

Konfiguration der Protokollierung

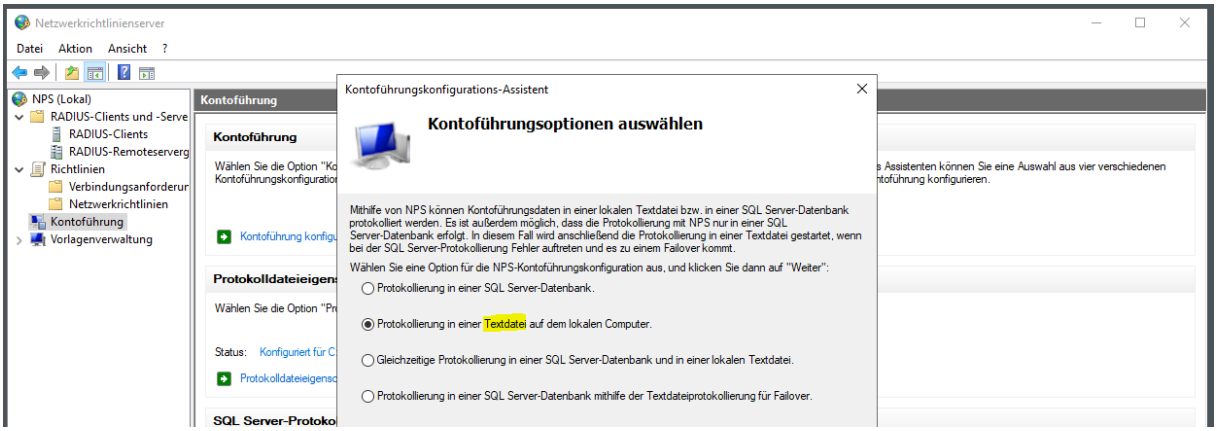
Ich bin durchaus ein Fan von Protokollen und Logfiles. Zu oft hatte ich schon Szenarien, in denen diese nicht oder nicht mehr zur Verfügung standen. So wird effizientes Troubleshooting, nachträgliche Forensik von Sicherheitsproblemen und

proaktives Monitoring ein Kraftakt. Daher überlege ich gerne, welcher Service welche Informationen auf welchem Wege zur Verfügung stellen kann.

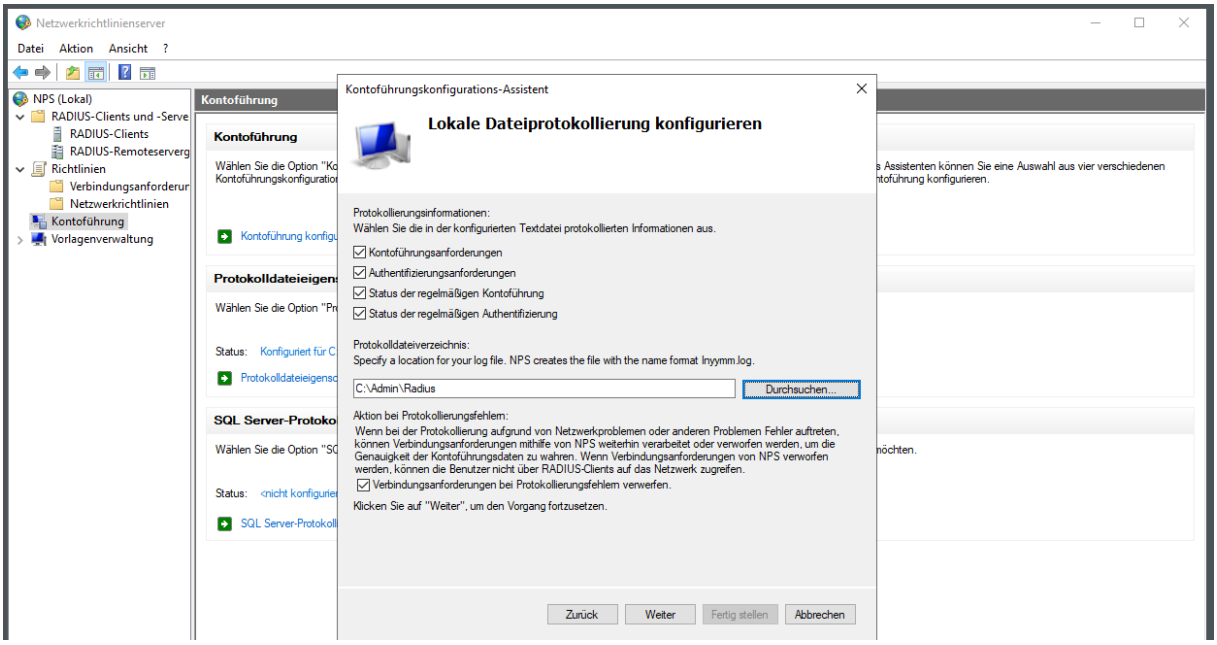
Der NPS hat ein sogenanntes Accounting. Dieses ist per Default nicht aktiv. Also werde ich das in der Konsole jetzt anpassen:



Aktuell ist der Server alleine. Daher könnte ein lokales Accounting in eine Textdatei genügen. Später wäre ein zentraler SQL-Server auf einem anderen Server die bessere Option:



Die Auswahl der möglichen Eintragstypen und die Angabe des Speicherpfades sind selbsterklärend:

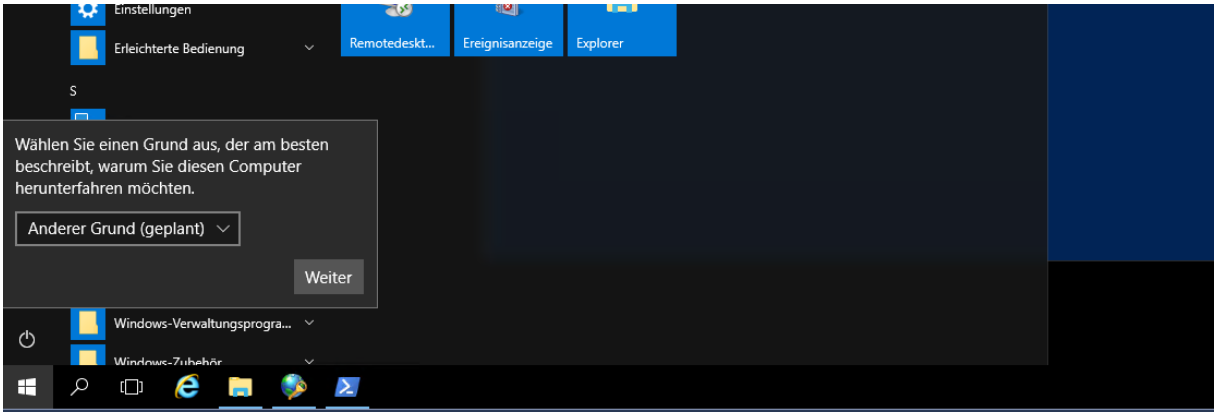


Für einen Test brauche ich aktive Verbindungen.

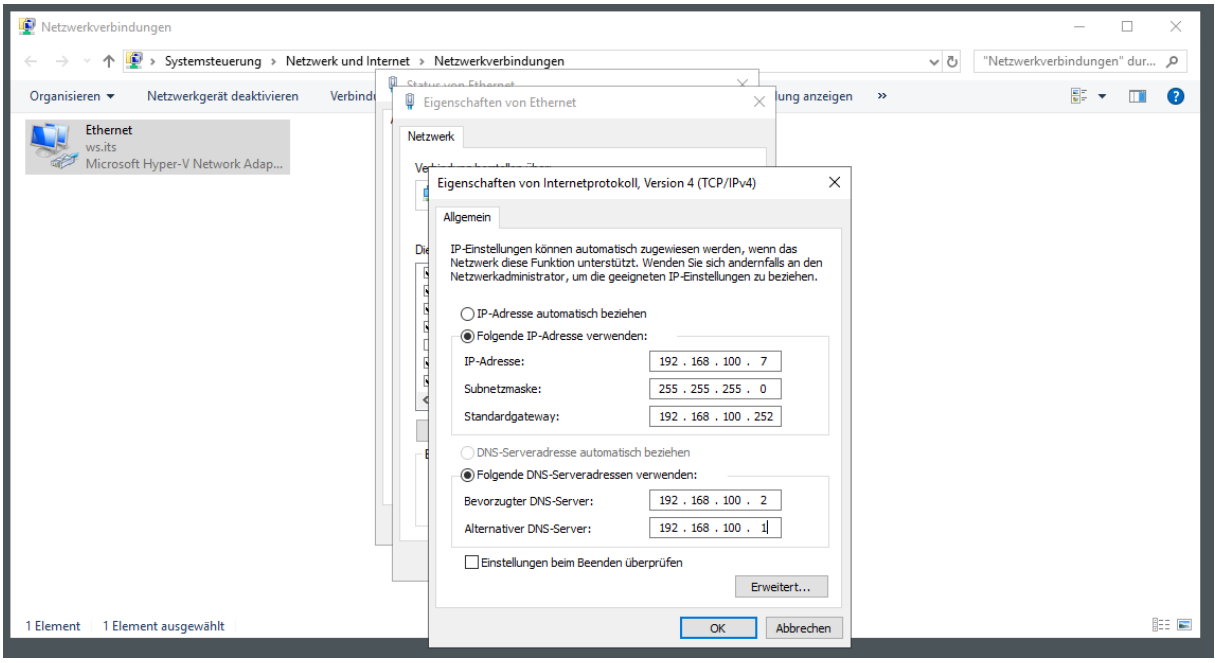
Austausch des NPS

Der NPS ist vollständig konfiguriert. Der nächste Schritt kann also der eigentliche Austausch des alten NPS gegen den neuen sein. Wie eingangs geplant wird das von mir durch den Austausch der IPv4-Konfiguration erledigt, da mein Radius-Client nur die IP-Adresse des NPS kennt. Zuerst sichte ich auf dem alten Server die aktuelle Konfiguration:

Nun könnte ich dem alten Server eine neue IP-Adresse geben. Ich habe aber alle anderen Komponenten des Servers entfernt. Also schalte ich den Server einfach aus. Ein Vorteil: sollte der neue Server nicht funktionieren, dann kann ich den neuen einfach ausschalten und den alten wieder hochfahren. So kommt der WLAN-AccessPoint wieder am funktionalen NPS auf dem alten Server raus:



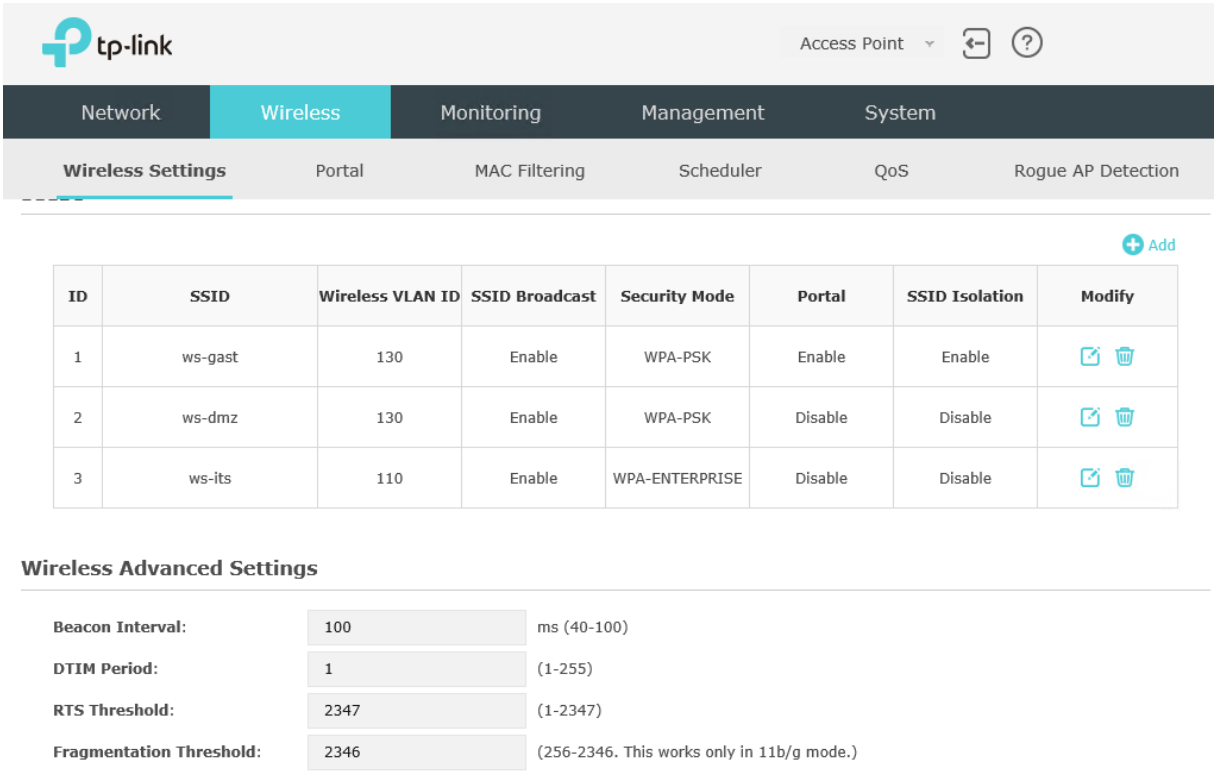
Natürlich muss ich sicherstellen, dass der alte und der neue Server nicht zeitgleich aktiv sind. Im neuen Server trage ich nun die statische IPv4-Konfiguration ein:



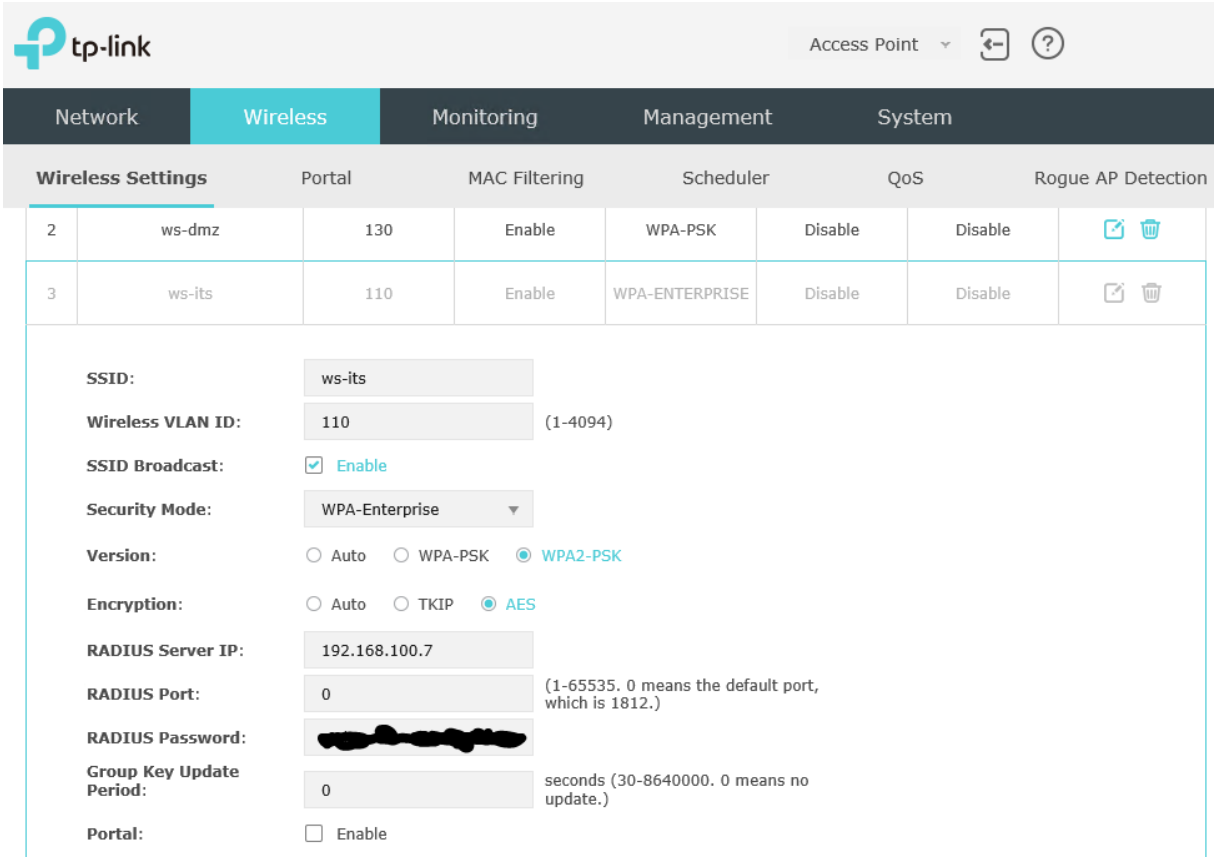
Ab jetzt sollten neue Clientverbindungen über WLAN vom WS-NPS1 authentifiziert werden.

Funktionsprüfung

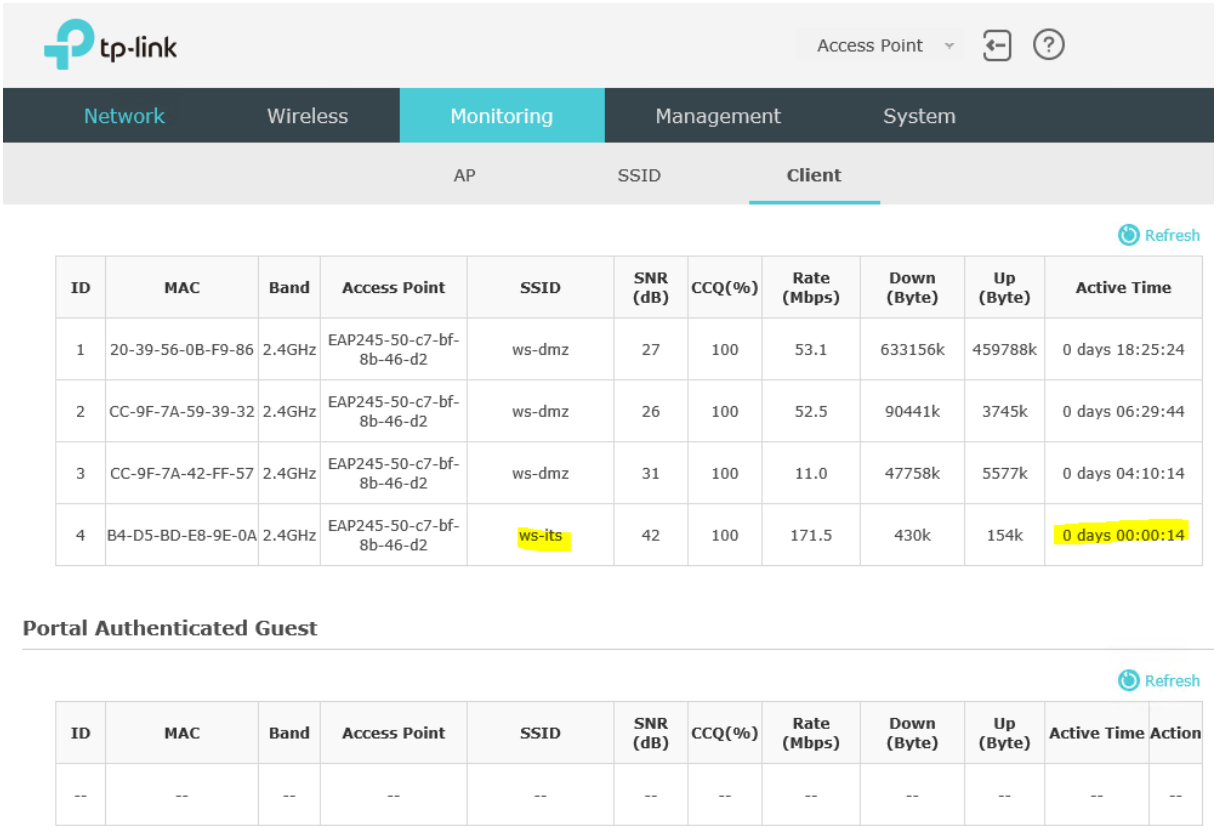
Das muss natürlich getestet werden. Damit die Zeit der Unterbrechung so gering wie möglich wird, habe ich bereits einen WLAN-Client zum Testen aufgebaut und mich von meinem Rechner aus mit dem Administrations-Portal meines WLAN-AccessPoints verbunden. Das WLAN mit der SSDI ws-ist verwendet WPA-Enterprise – also die Authentifizierung gegen den NPS-Server:



Hier steht die IPv4 des NPS-Servers. Bei nur einem AP hätte ich hier auch einfach die neue IP des NPS eintragen können. Aber mit mehreren AP wäre ohne zentralen Server der Aufwand höher:



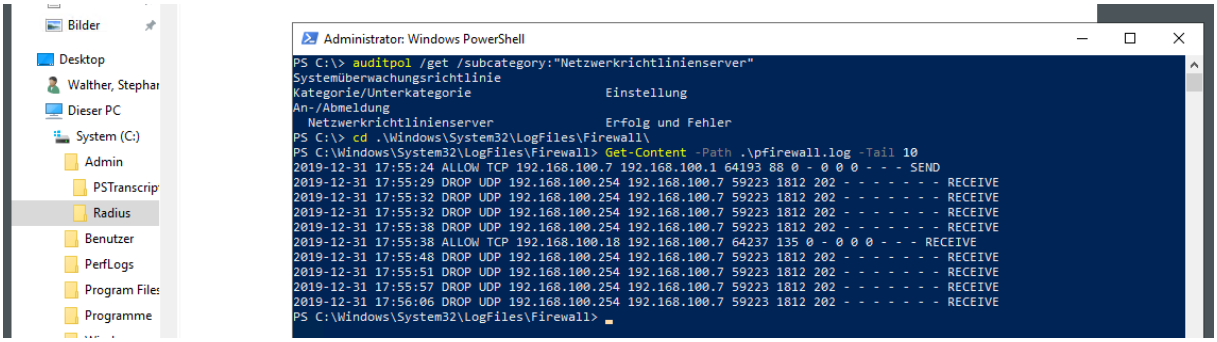
Jetzt versuche ich eine Verbindung zwischen meinem Testclient und dem WLAN. Der AccessPoint sieht die Verbindung:



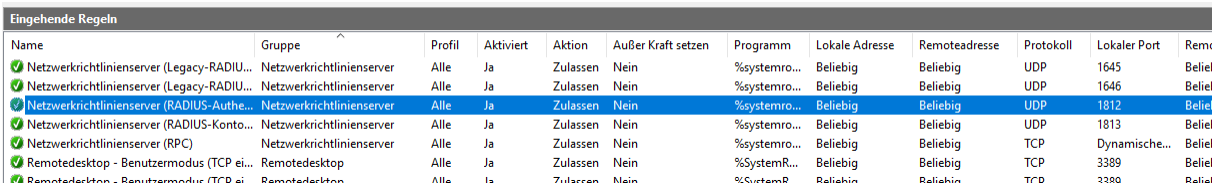
Aber der Client kann sich nicht mit dem WLAN verbinden. Woran liegt das? Ich kenne das Problem bereits aus einem Projekt bei einem Kunden. Ein PowerShell-Command gibt mir recht. Mit dem Befehl

```
Get-Content -Path c:\windows\system32\logfiles\firewall\pfirewall.log -tail 10
```

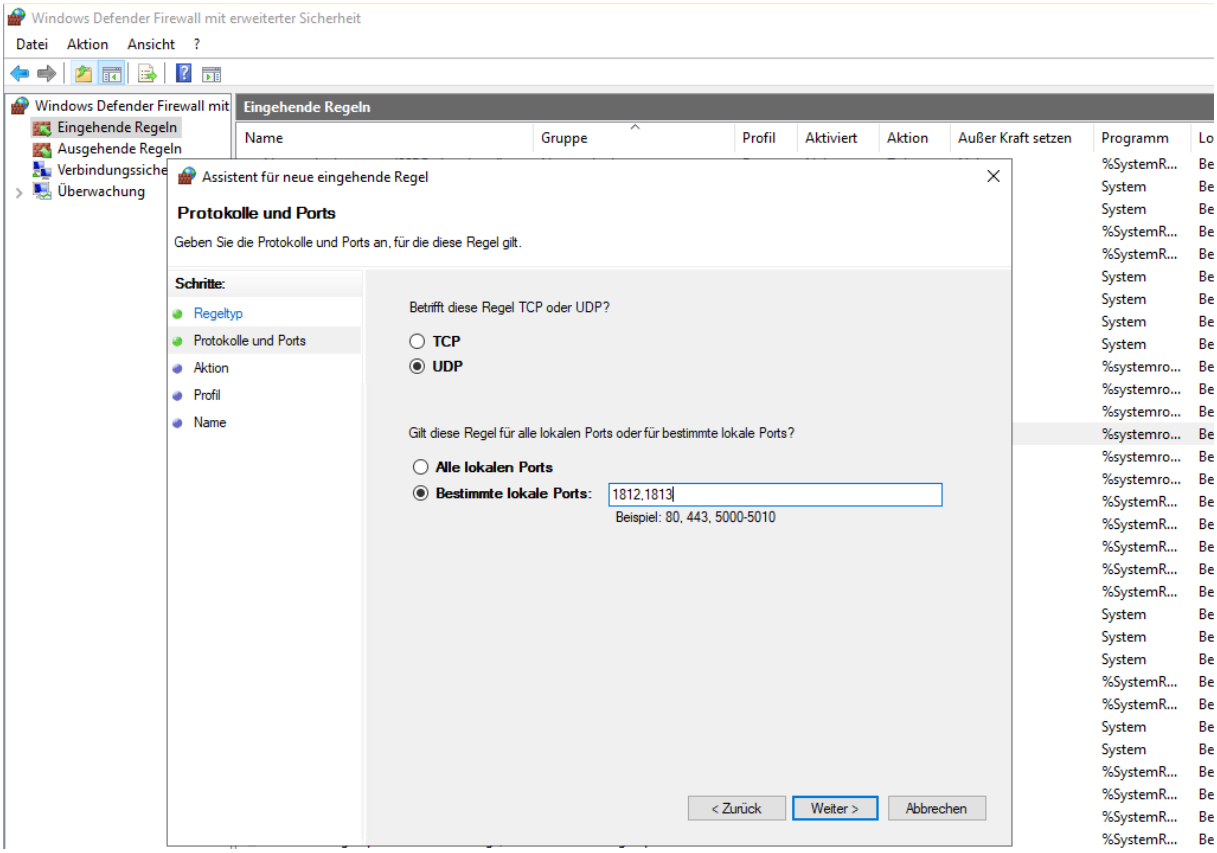
kann ich die letzten 10 Zeilen des Windows-Firewall-Logfiles ansehen. Dieses Logfile ist nicht standardmäßig aktiv. Bei mir wird es über eine Gruppenrichtlinie proaktiv eingeschaltet (wo wir ein schönes Beispiel für das erfolgreiche Troubleshooting mit zuvor aktivierten Logfiles haben). Man sieht schnell, dass es hier einige DROPs gibt. Die dazugehörigen Verbindungen auf UDP 1812 gehören zum Service NPS. Das sind die eingehenden Versuche vom WLAN-AccessPoint:

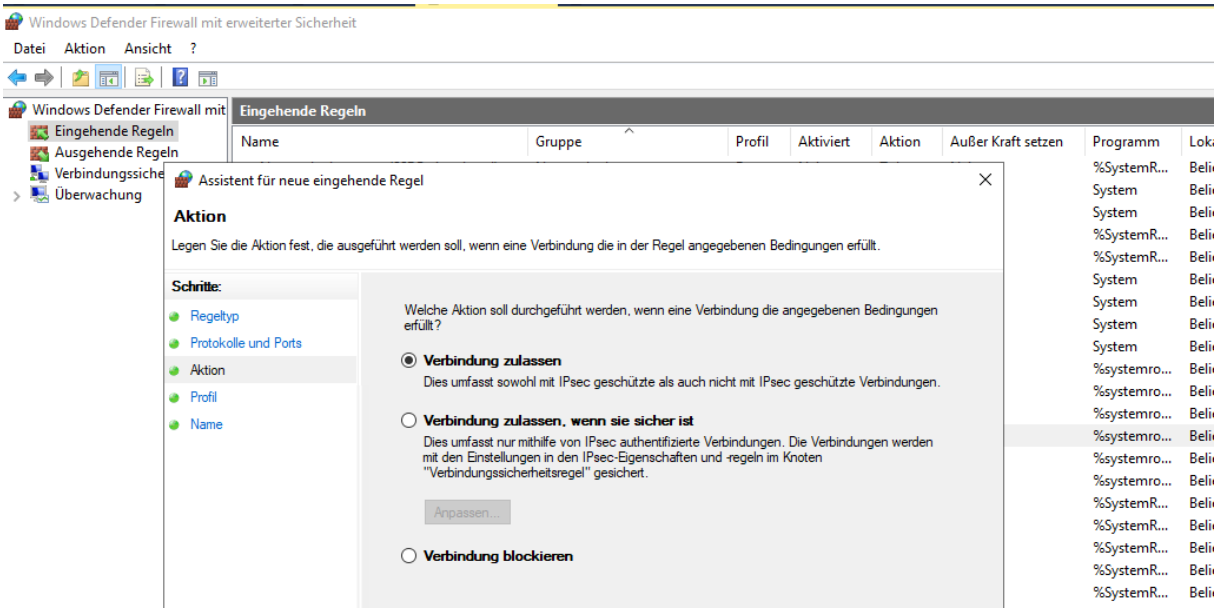


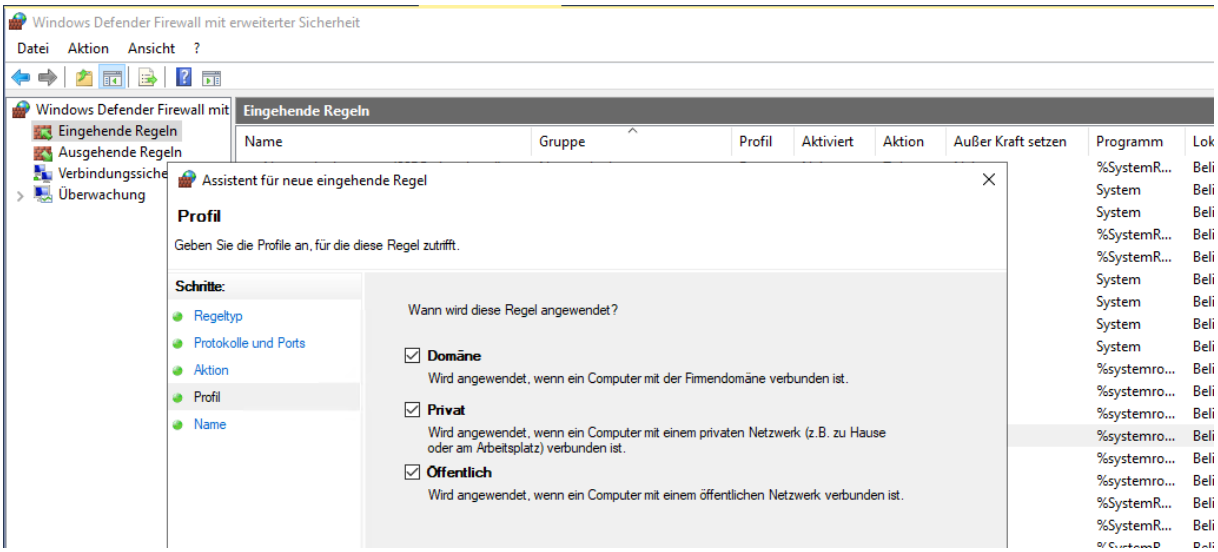
Aber warum kommen diese Verbindungen nicht durch? Natürlich ist meine Windows Firewall aktiv. Aber normalerweise wird doch bei einer Rolleninstallation die erforderliche Ausnahme automatisch erstellt? So kenne ich das von fast allen anderen Rollen. Und ein Blick in die Regeln zeigt auch eine NPS-Regel für UDP 1812, die eingehenden Traffic erlaubt:

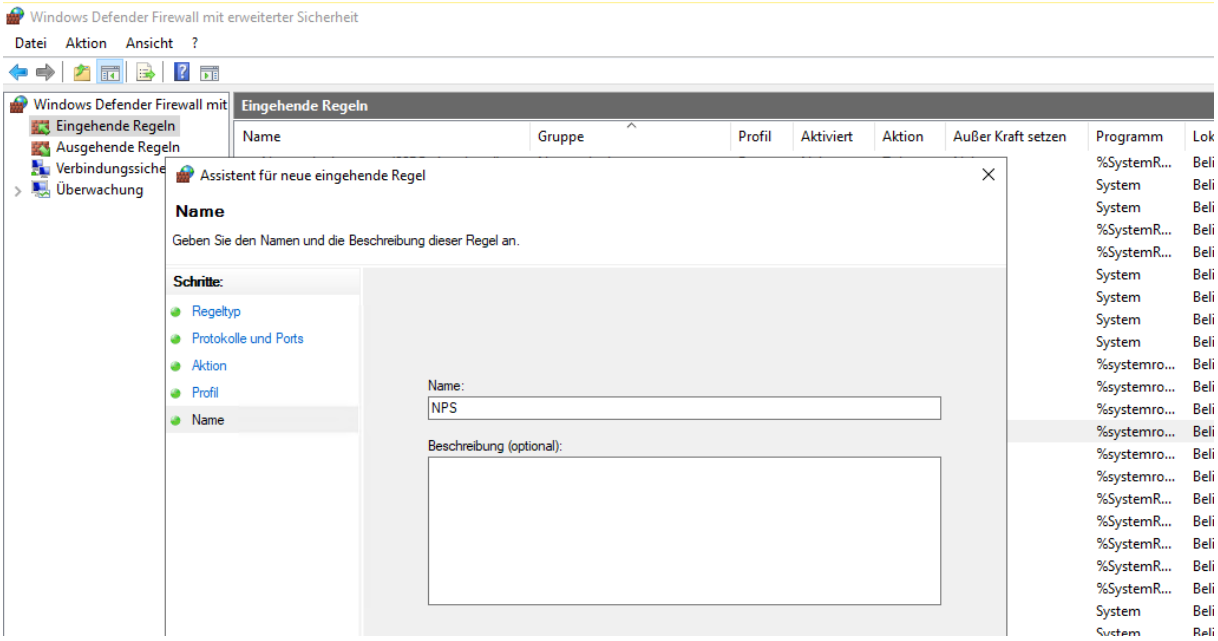


Die Lösung des Rätzels ist interessant: die vordefinierte Regel funktioniert nicht. Man muss die Regel selber noch einmal erstellen. Das Phänomen kenne ich bisher nur beim NPS des Windows Server 2019. Gefixt wurde es bis heute wohl noch nicht. Egal, ich kann die neue Regel einfach selber lokal erstellen:

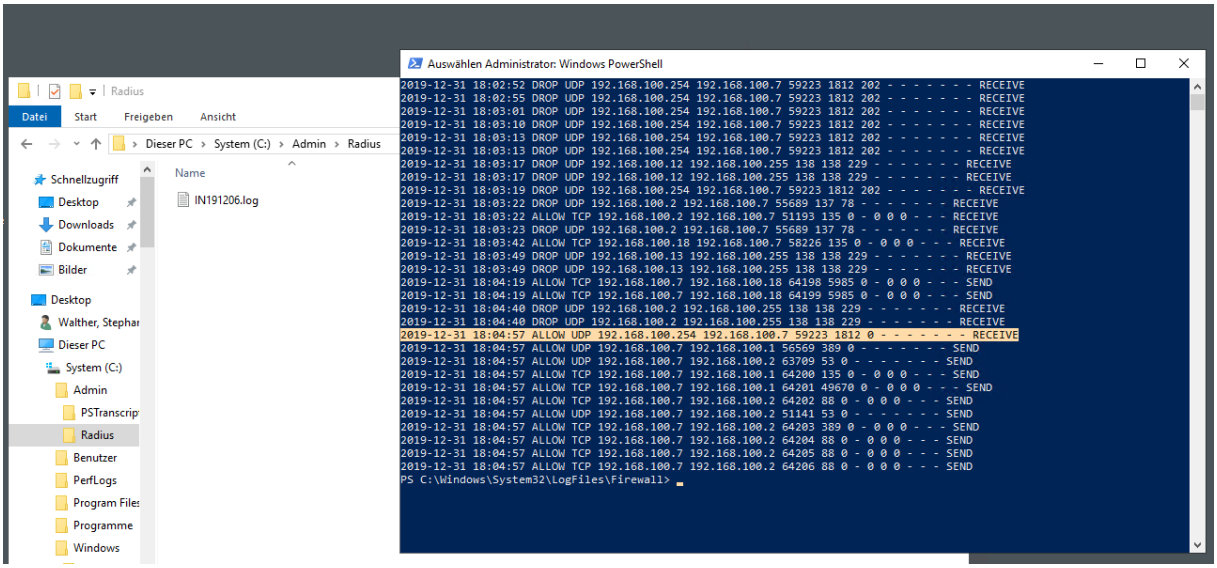



Windows Defender Firewall mit erweiterter Sicherheit
Datei Aktion Ansicht ?
Windows Defender Firewall mit
Eingehende Regeln
Ausgehende Regeln
Verbindungssiche
Überwachung
Eingehende Regeln
Name Gruppe Profil Aktiviert Aktion Außer Kraft setzen Programm
Assistent für neue eingehende Regel
Aktion
Legen Sie die Aktion fest, die ausgeführt werden soll, wenn eine Verbindung die in der Regel angegebenen Bedingungen erfüllt.
Schritte:
Regeltyp
Protokolle und Ports
Aktion
Profil
Name
Welche Aktion soll durchgeführt werden, wenn eine Verbindung die angegebenen Bedingungen erfüllt?
Verbindung zulassen
Dies umfasst sowohl mit IPsec geschützte als auch nicht mit IPsec geschützte Verbindungen.
Verbindung zulassen, wenn sie sicher ist
Dies umfasst nur mithilfe von IPsec authentifizierte Verbindungen. Die Verbindungen werden mit den Einstellungen in den IPsec-Eigenschaften und -regeln im Knoten "Verbindungssicherheitsregel" gesichert.
Anpassen...
Verbindung blockieren

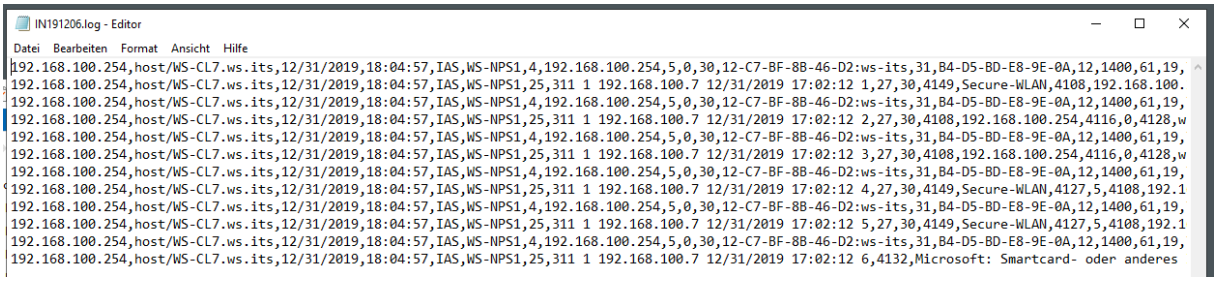

Windows Defender Firewall mit erweiterter Sicherheit
Datei Aktion Ansicht ?
Windows Defender Firewall mit
Eingehende Regeln
Ausgehende Regeln
Verbindungssiche
Überwachung
Eingehende Regeln
Name Gruppe Profil Aktiviert Aktion Außer Kraft setzen Programm
Assistent für neue eingehende Regel
Profil
Geben Sie die Profile an, für die diese Regel zutrifft.
Schritte:
Regeltyp
Protokolle und Ports
Aktion
Profil
Name
Wann wird diese Regel angewendet?
Domäne
Wird angewendet, wenn ein Computer mit der Firmendomäne verbunden ist.
Privat
Wird angewendet, wenn ein Computer mit einem privaten Netzwerk (z.B. zu Hause oder am Arbeitsplatz) verbunden ist.
Öffentlich
Wird angewendet, wenn ein Computer mit einem öffentlichen Netzwerk verbunden ist.


Windows Defender Firewall mit erweiterter Sicherheit
Datei Aktion Ansicht ?
Windows Defender Firewall mit
Eingehende Regeln
Ausgehende Regeln
Verbindungssiche
Überwachung
Eingehende Regeln
Name Gruppe Profil Aktiviert Aktion Außer Kraft setzen Programm
Assistent für neue eingehende Regel
Name
Geben Sie den Namen und die Beschreibung dieser Regel an.
Schritte:
Regeltyp
Protokolle und Ports
Aktion
Profil
Name
Name:
NPS
Beschreibung (optional):

Die Regel wird sofort aktiv. Ich starte auf meinem Client einen weiteren Verbindungsversuch und dieses mal ist er erfolgreich. Auf dem NPS-Server sehe ich nun die erlaubten Verbindungen:



Zum Funktionalitätstest gehört auch die Prüfung des Accountings. Die eingehenden Authentifizierungen werden wie gewünscht in Text-Logfiles gespeichert:



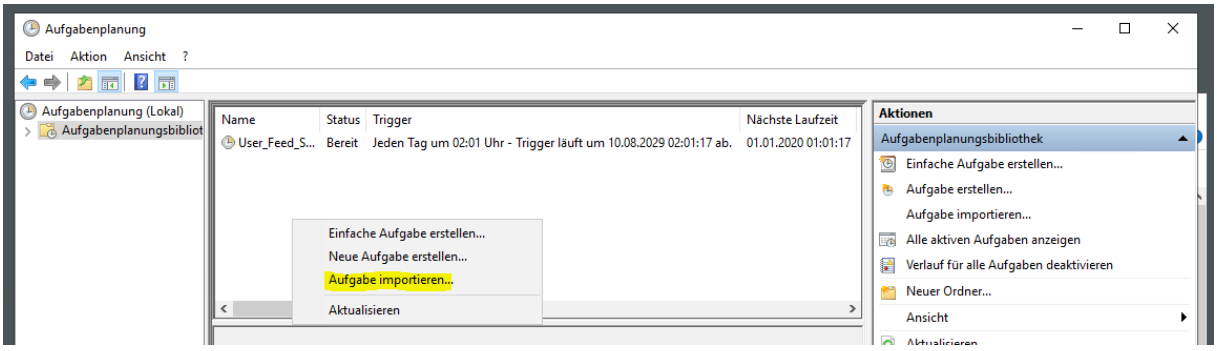
Damit ist mein NPS-Service erfolgreich auf Windows Server 2019 migriert.

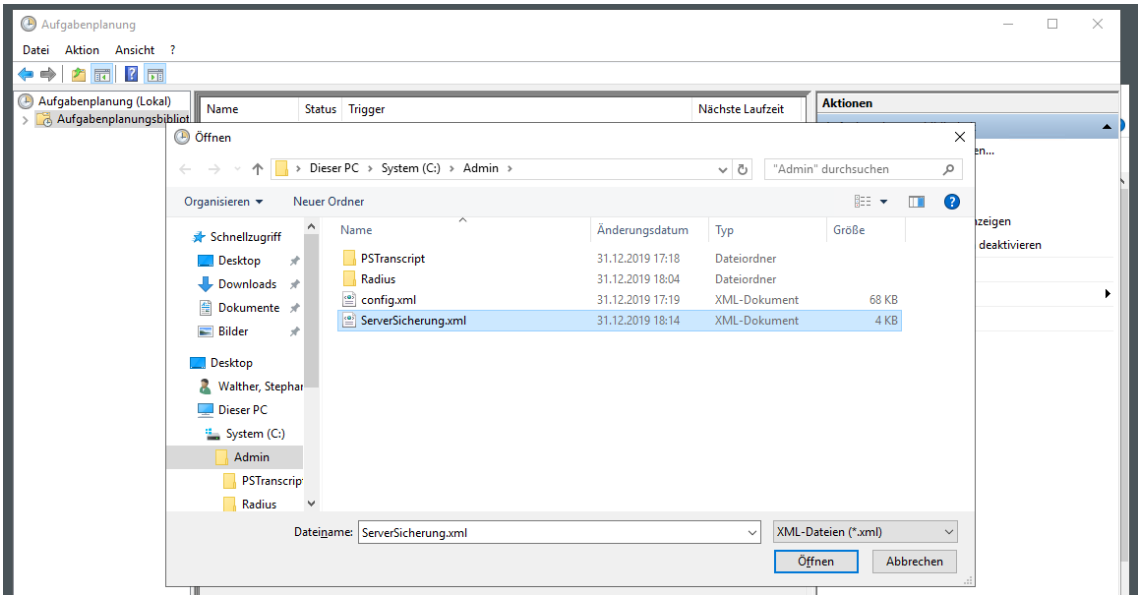
Nacharbeiten

Datensicherung

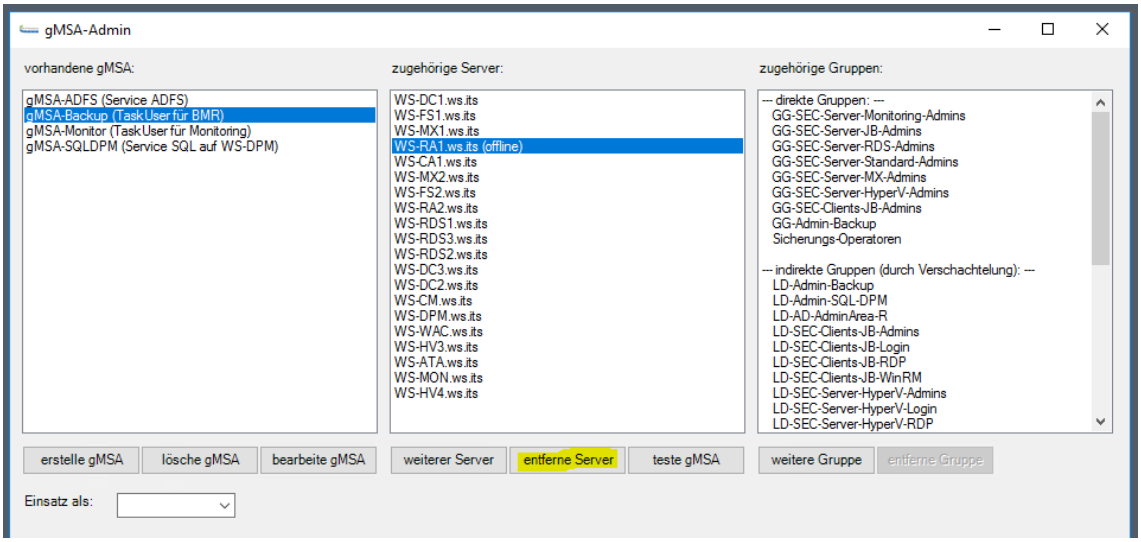
Zu den üblichen Nacharbeiten gehört natürlich die Einrichtung der Datensicherung. Der Server ist zwar mit der exportierten Konfiguration recht schnell wiederaufgebaut, aber mit einer Recovery geht es einfach schneller. Zudem sichere ich so auch die Logfiles des NPS mit.

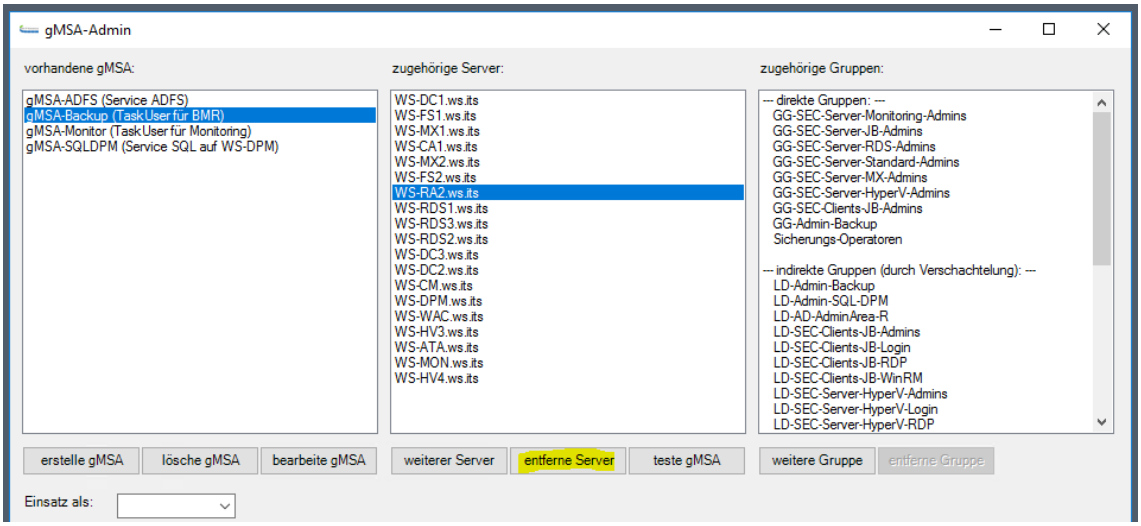
Ich konfiguriere meine zentral gesteuerte SystemState-Sicherung mit Windows Server Backup. Das Feature ist bereits installiert. Es fehlt nur noch die geplante Aufgabe. Diese importiere ich als xml-Datei in der Konsole „Aufgabenplanung“:



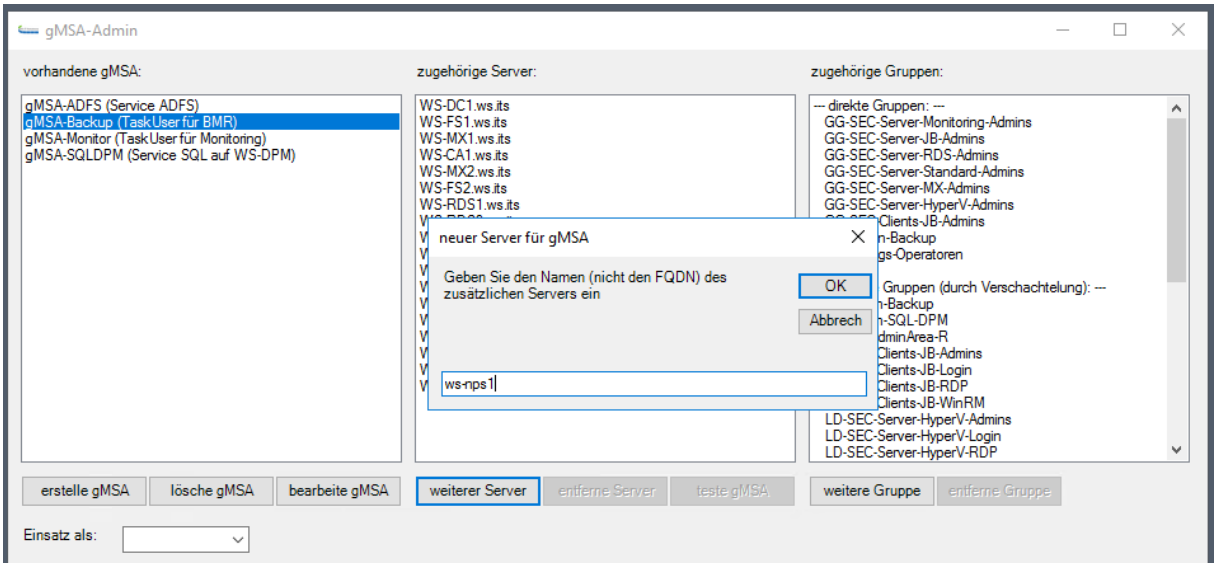


Der ausführende Account ist wieder ein Group Managed Service Account, denn ich wie bei den anderen Migrationen auch von meinem Domain Controller aus mit einer selbst programmierten PowerShell-GUI einrichte. Zuerst entferne ich hier aber noch die alten ServerAccounts WS-RA1 und WS-RA2:

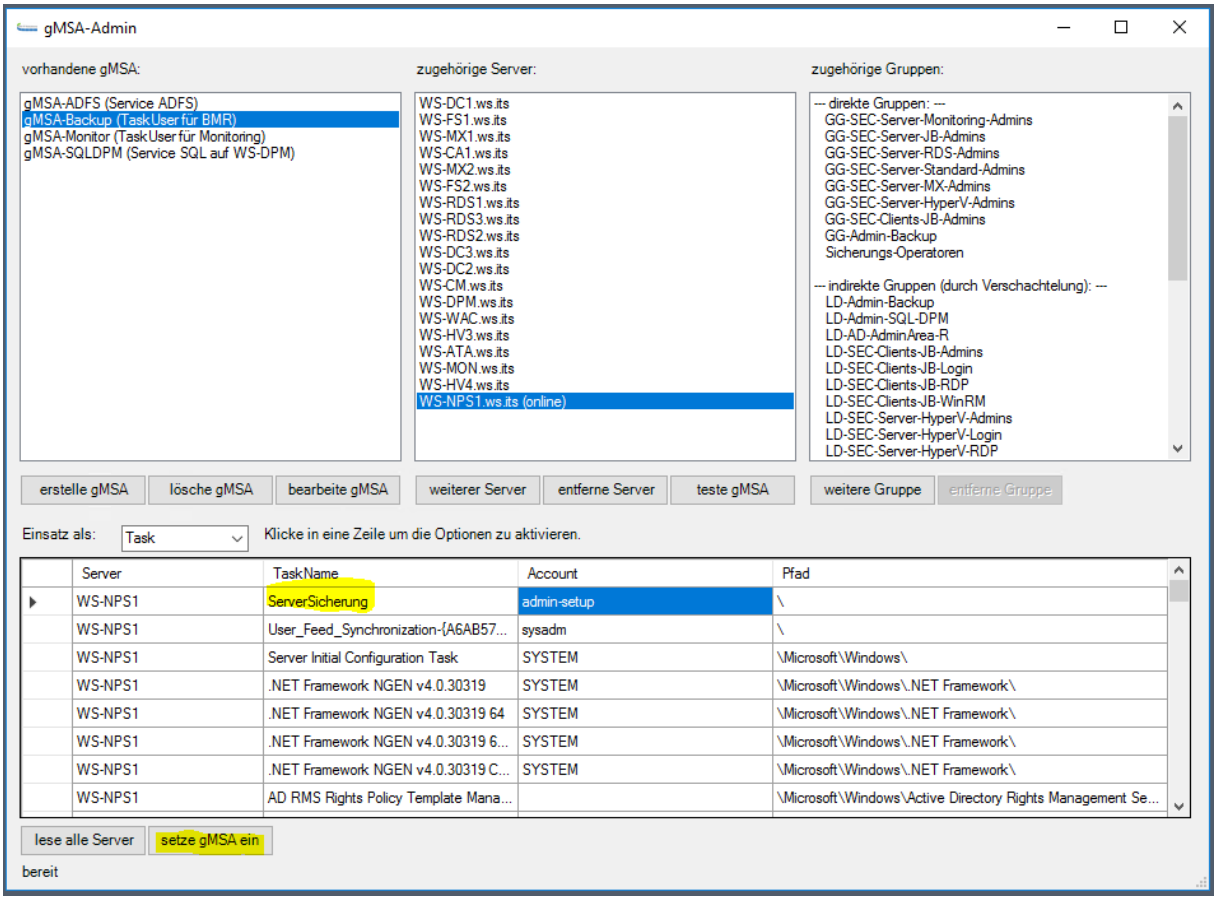


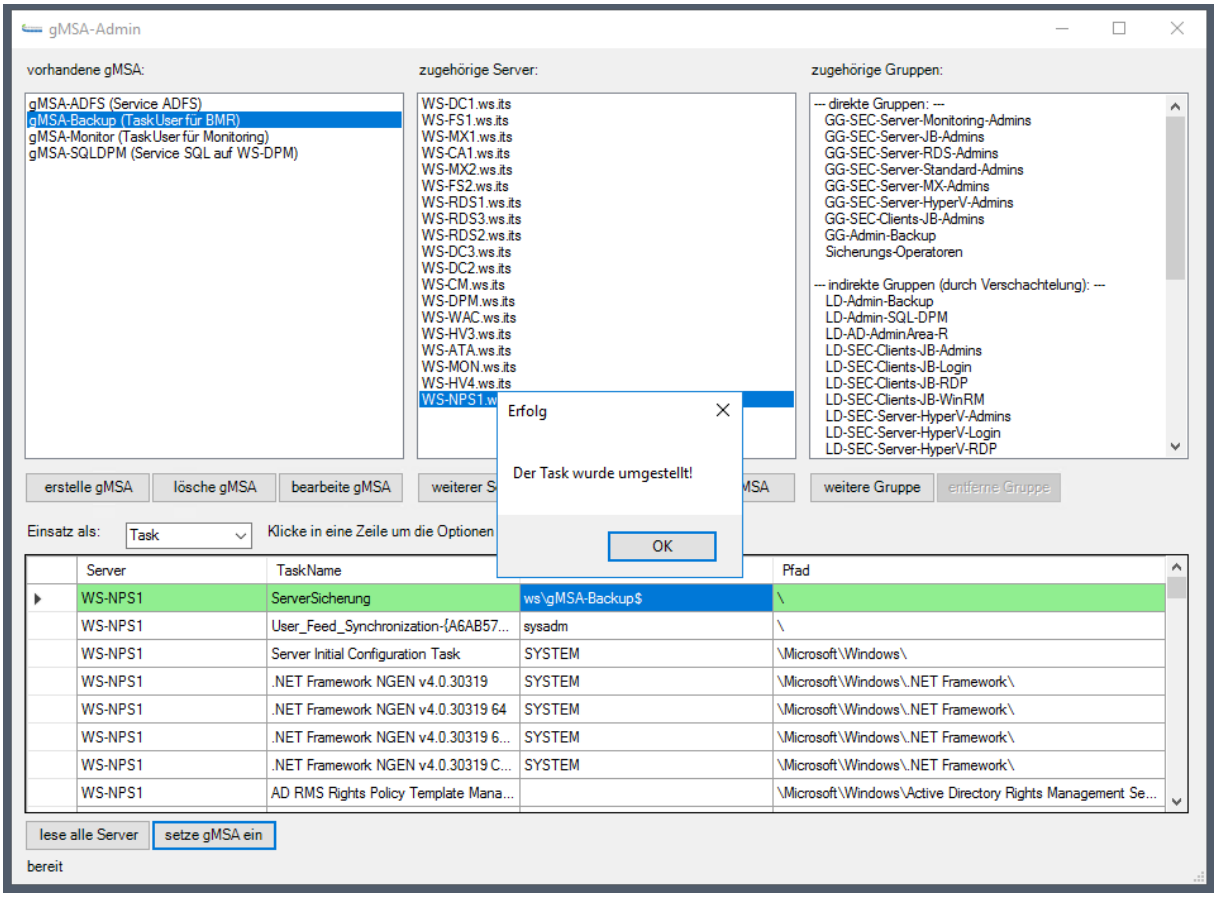


Dann erlaube ich die Passwortübertragung zum WS-NPS1:

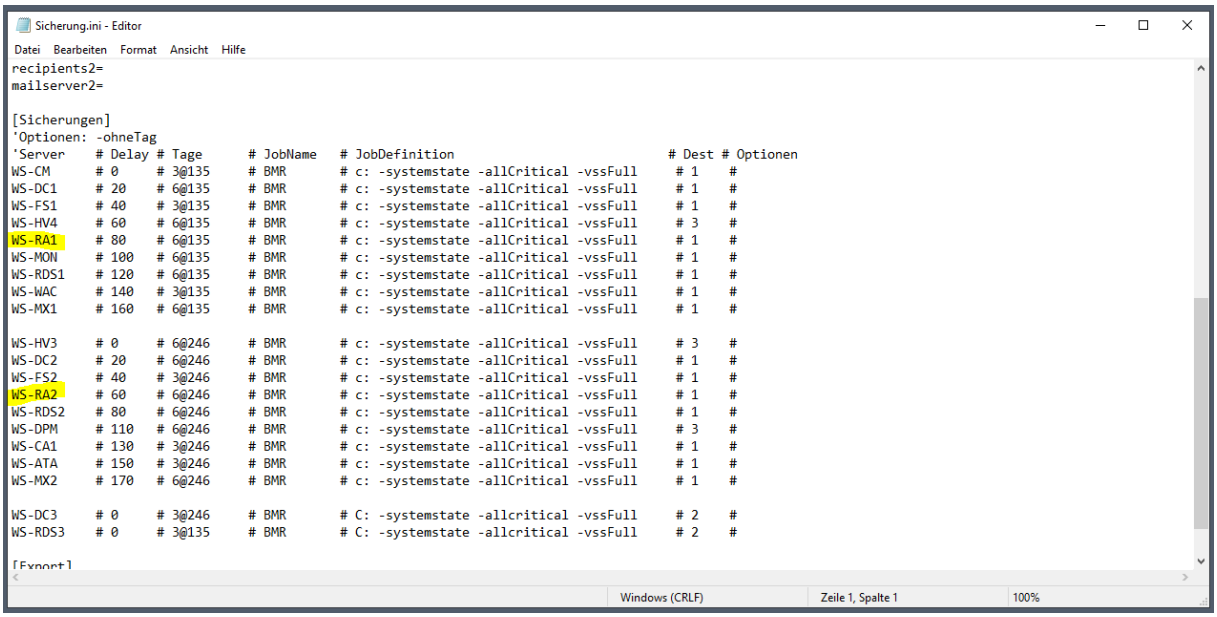


Über das PowerShell-Remoting kann ich dann auf dem Server WS-NPS1 die geplante Aufgabe remote umkonfigurieren:

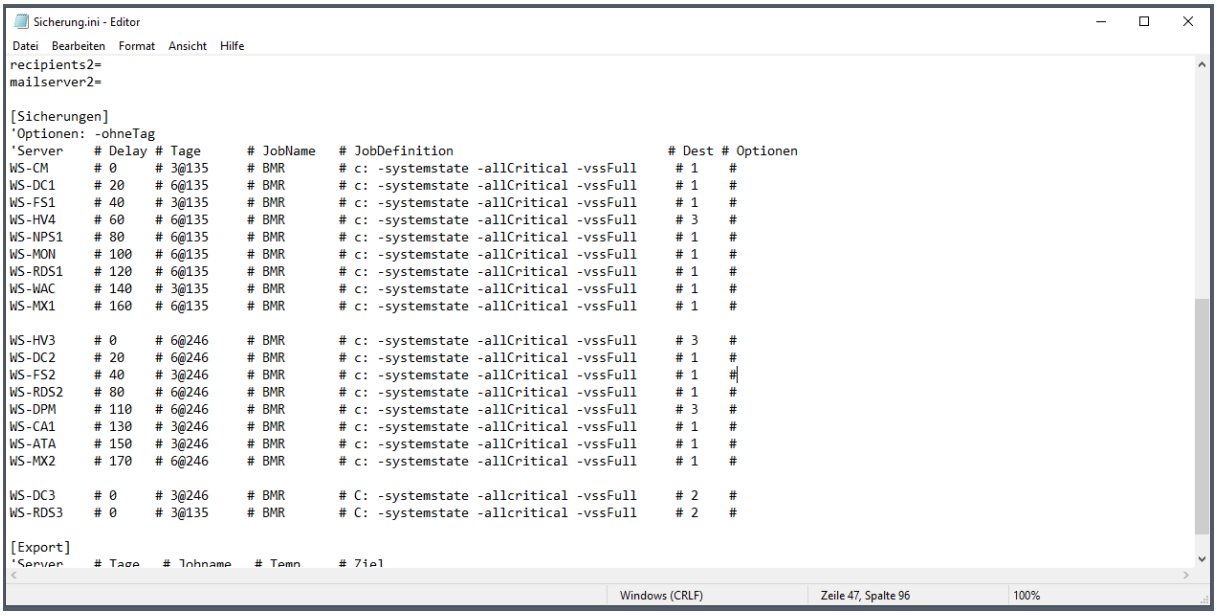




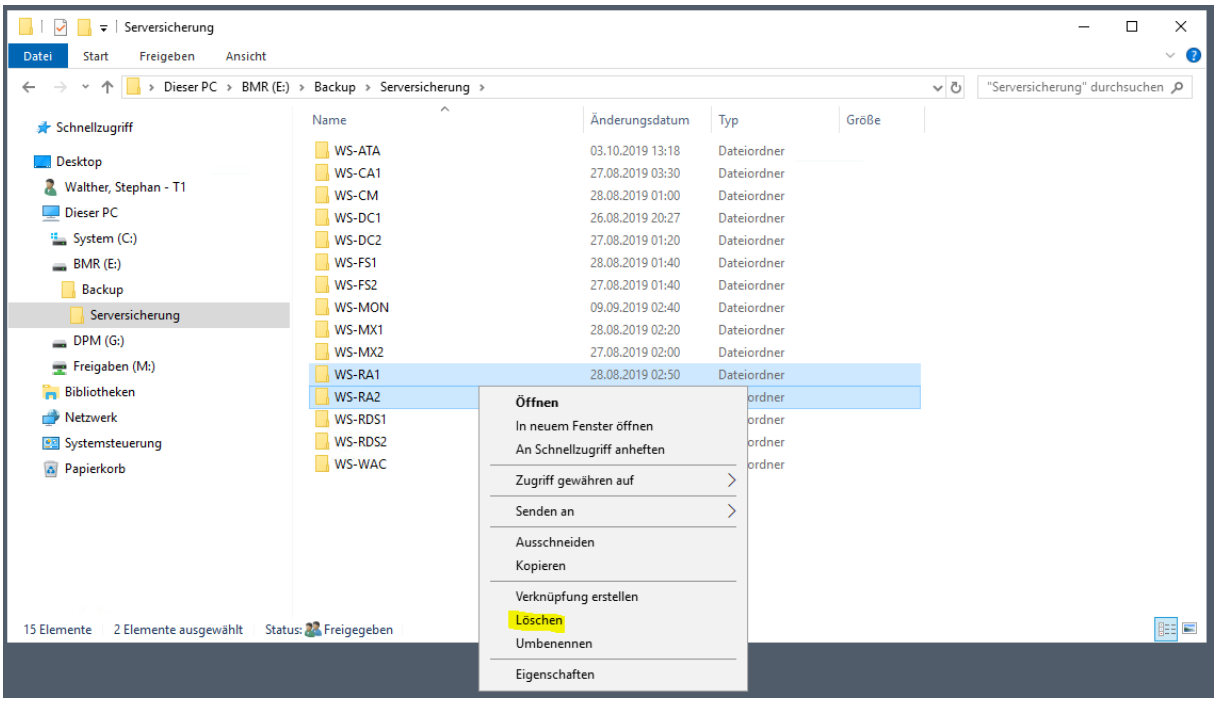
Die geplante Aufgabe startet ein zentral abgelegtes Script. Dessen Konfiguration ist eine simple ini-Datei. Hier entferne ich die Sicherungsjobs der alten Server:



Und dann konfiguriere ich die Datensicherung für den neuen WS-NPS1:



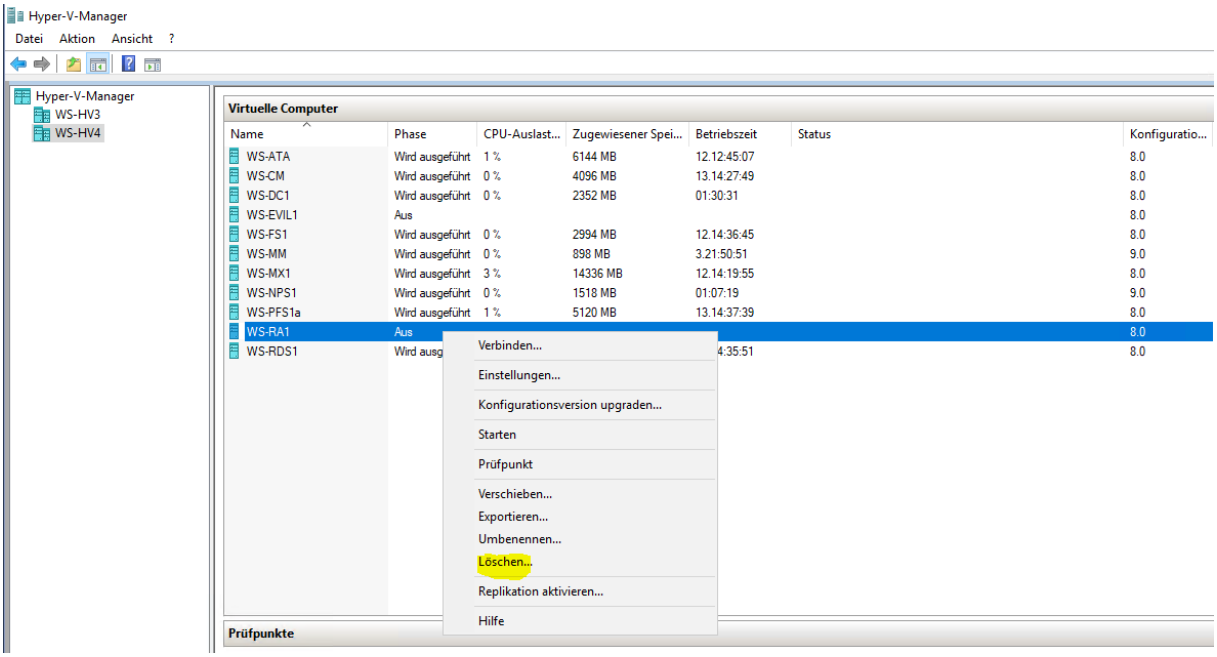
Die alten Server hatten natürlich Sicherungen erstellt. Diese benötige ich nicht mehr. Daher entferne ich sie aus dem Sicherungsverzeichnis:

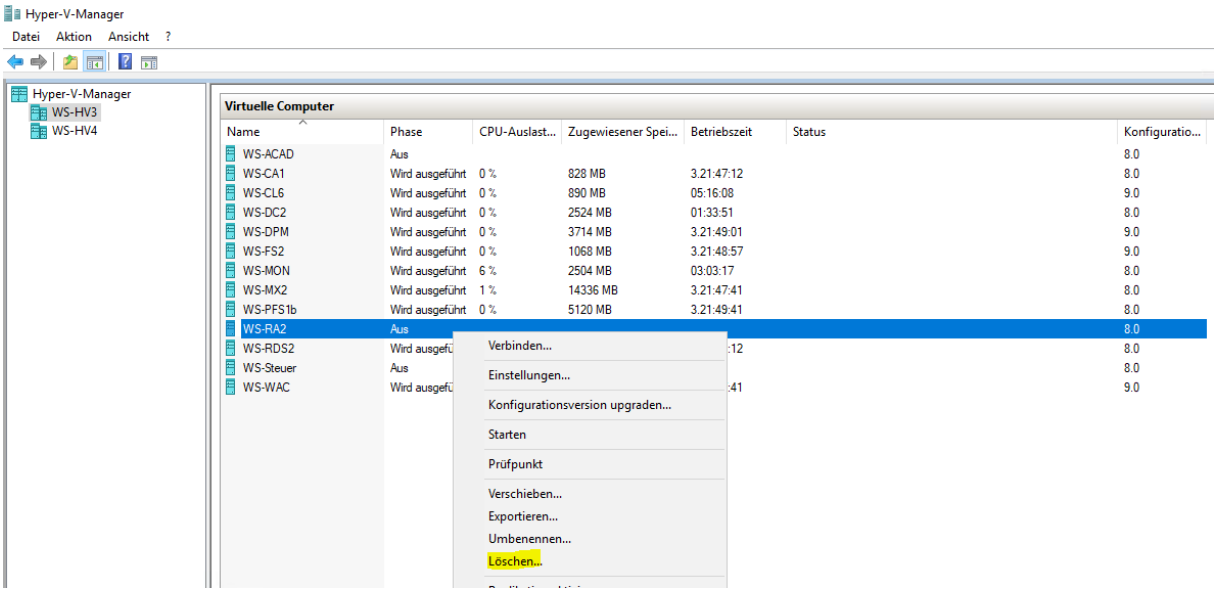


Morgen kann ich die Datensicherung des neuen Servers in der Monitor-Email sehen. Die Sicherung umfasst den gesamten Server. Eine zusätzliche Nutzdatensicherung ist nicht erforderlich.

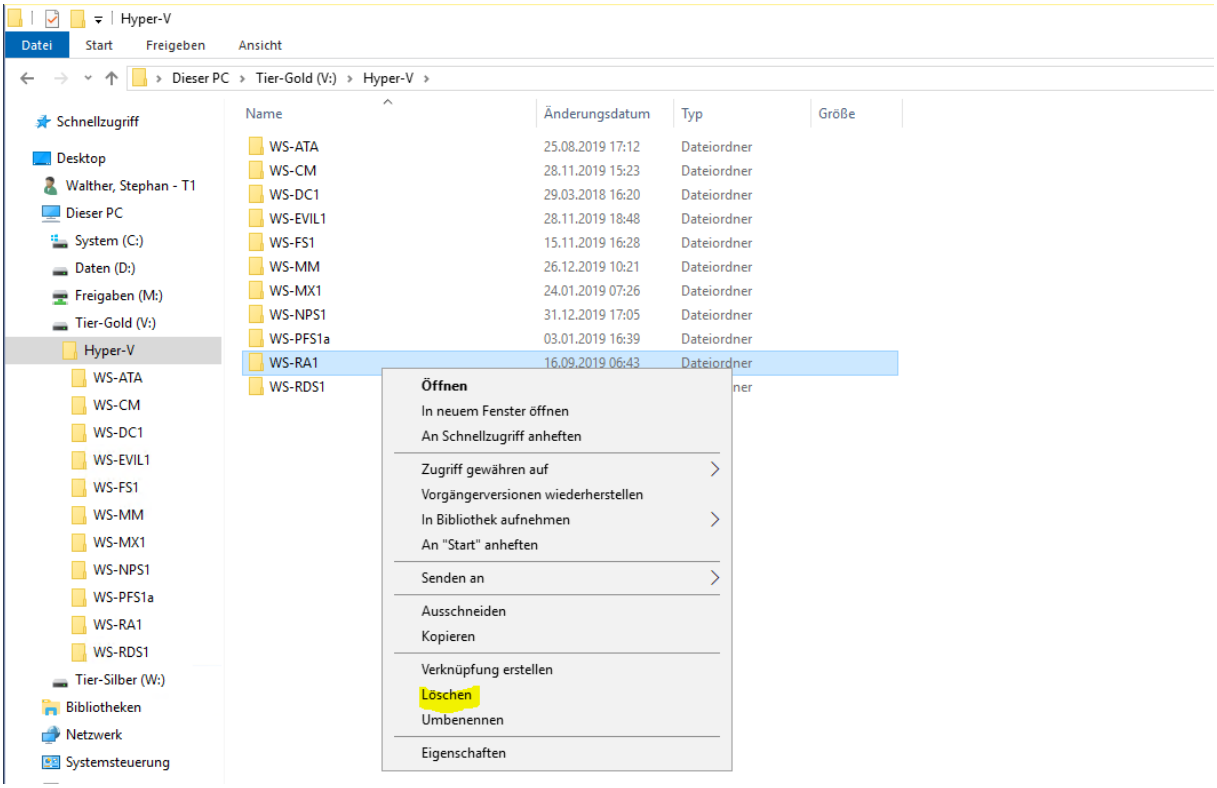
Bereinigung der VMs

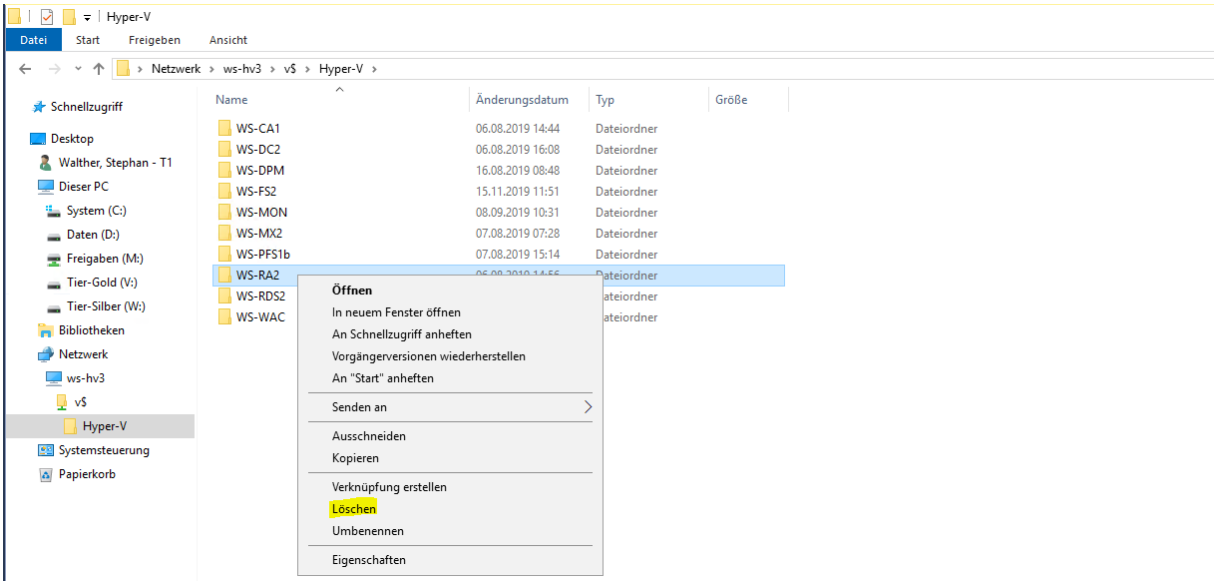
Jetzt kann ich den Speicherplatz in den Hyper-V-Servern freigeben. Dazu lösche ich die beiden alten VMs:



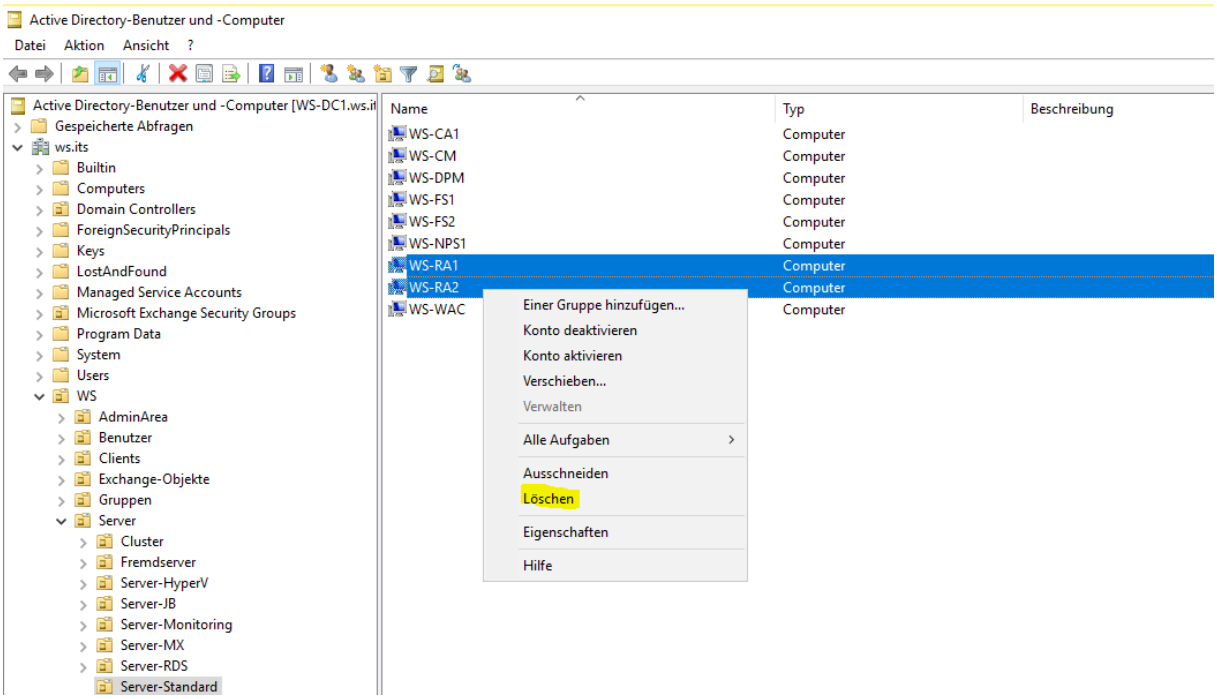


Die virtuellen Festplatten bleiben dabei erhalten. Diese lösche ich manuell mit dem Windows Explorer:





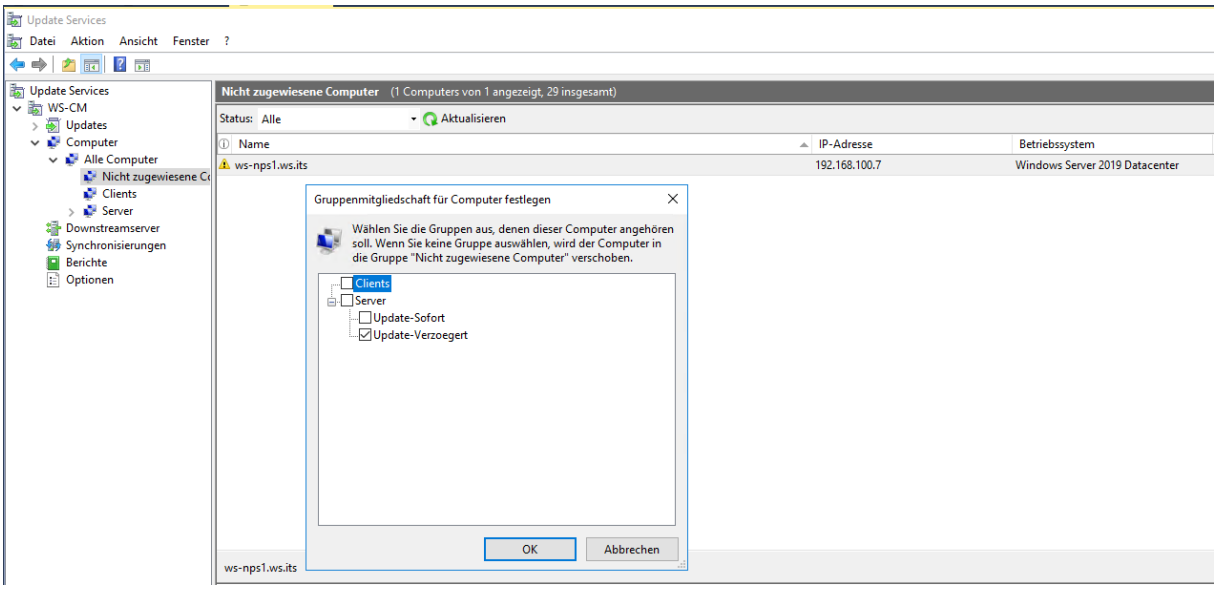
Im Active Directory finde ich die beiden verwaisten Computerkonten. Diese lösche ich:



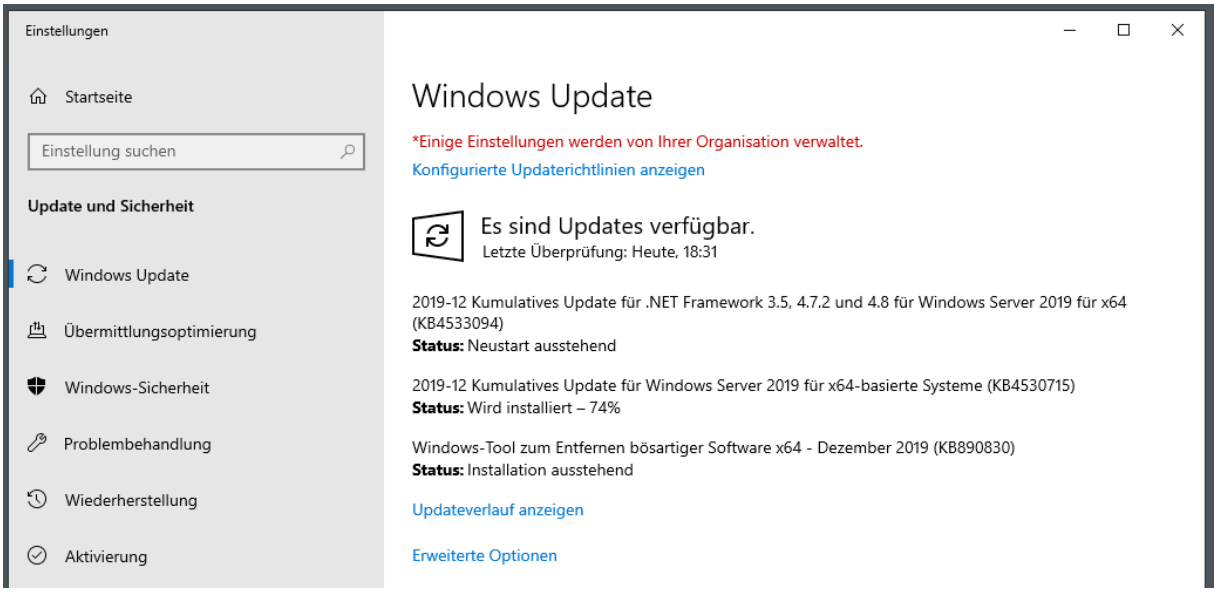
Damit sind die alten Server bereinigt.

Windows Updates

Der neue Server darf natürlich auch Updates über meinen WSUS erhalten. Da er aber noch nicht hochverfügbar ist, kommt er in die Gruppe mit der verzögerten Update-Genehmigung. Sollte mal ein Update Probleme bereiten, dann erkenne ich das an Servern der Gruppe „Update-Sofort“ und habe dann die Möglichkeit, die Verteilung aufzuhalten:



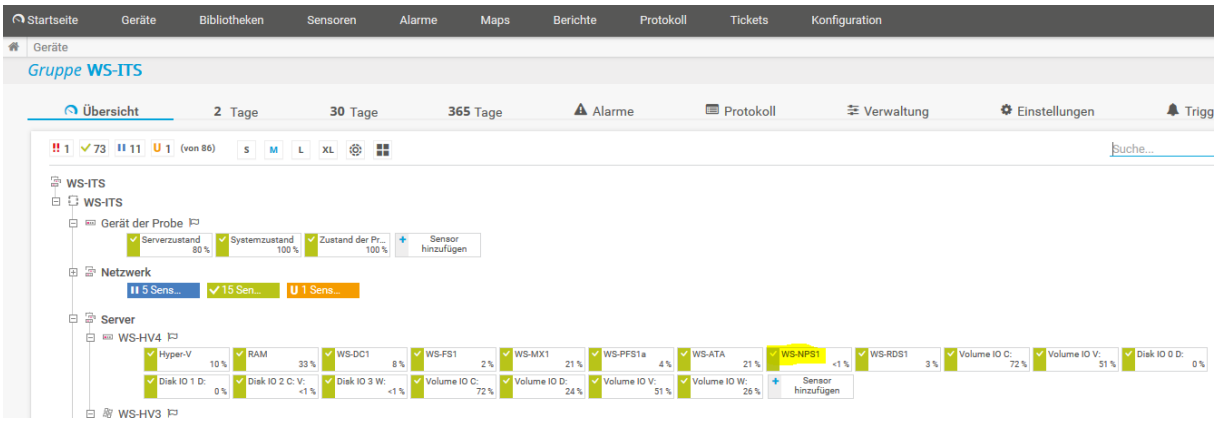
Der Server holt die Updates automatisch nach. Ich starte die Installation aber manuell, damit ich deren Ausführung überwachen kann:



Einen Neustart später ist der Server up-to-date.

Monitoring

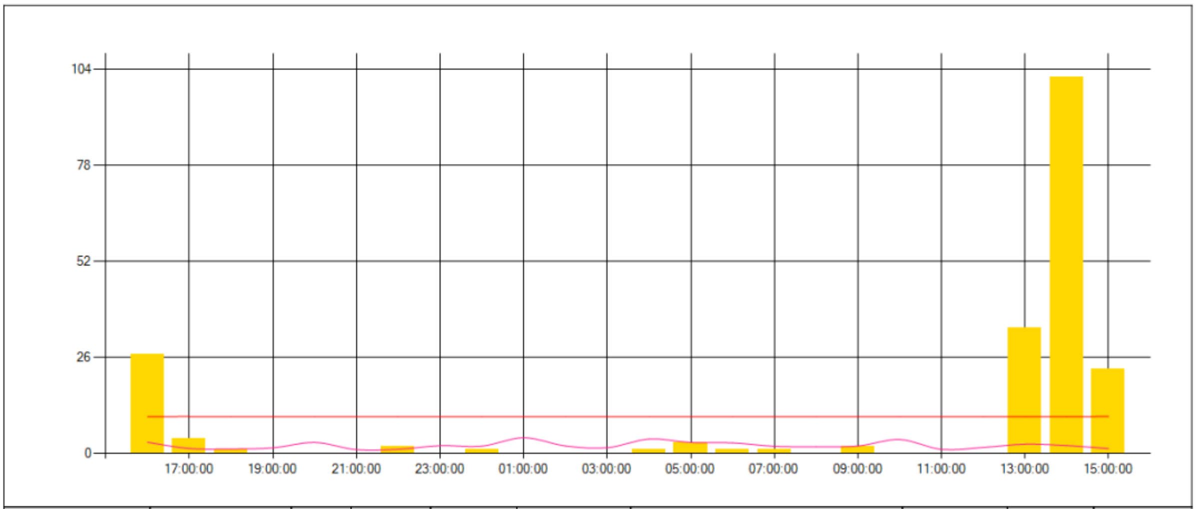
Mittlerweile hat mich mein PRTG-Monitoring via Push-Benachrichtigung informiert, dass 2 virtuelle Computer fehlen und die WAP-Dienste nicht erreichbar sind. Daher wird es Zeit, die Konfiguration anzupassen. Ich entferne die alten VMs und trage dafür die neue VM ein. Dann entferne ich noch alle Sensoren für ADFS und WAP:



Abhängigkeit zur PKI

Unmittelbar nach dem Abschluss einer Migration kann eigentlich noch keine Aussage über deren Erfolg getroffen werden. Manche Abhängigkeiten und Probleme bemerkt man erst im Anschluss. So ist es mir einen Tag später auch gegangen. Ich erhalte jeden Morgen einige Mails, mit denen ich verschiedene Funktionalitäten prüfen kann. Eine Mail davon sendet mir eine Auswertung der letzten 24 Stunden, in der verdächtige Anmeldeaktivitäten aufgezeigt werden. Mit der PowerShell hab ich dazu eine hübsche grafische Darstellung gerendert:

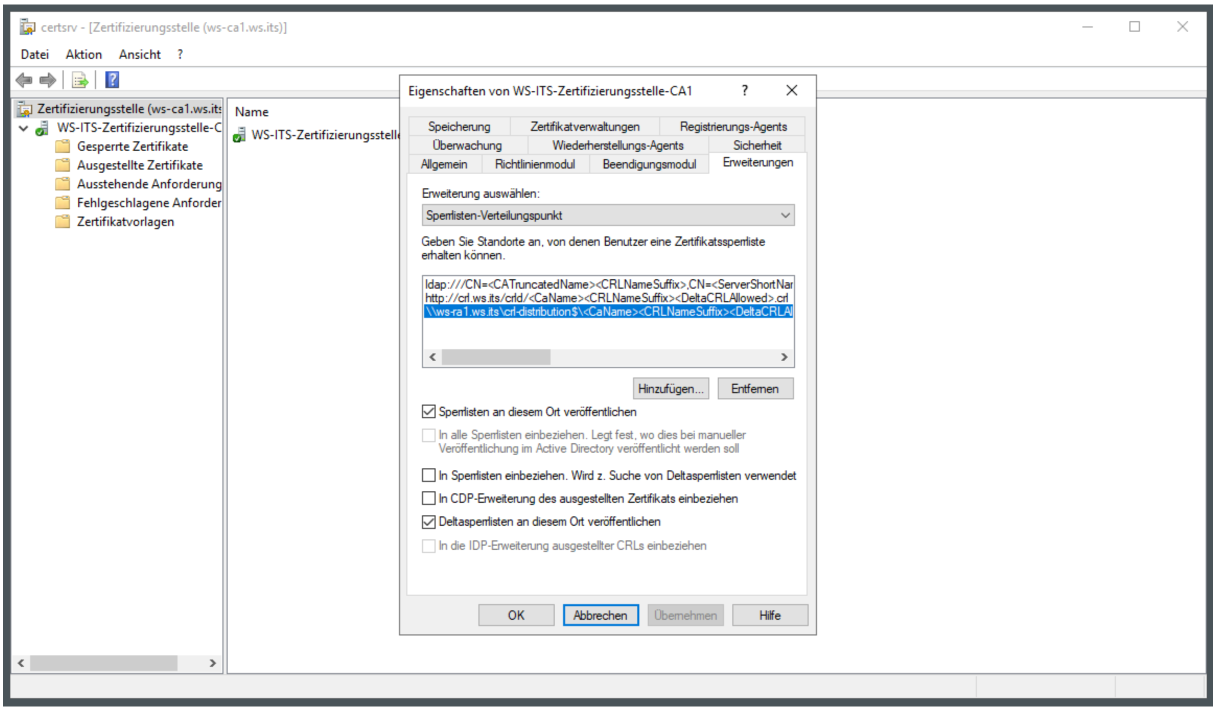
34 neue Events: 4004 - NTML-Authentication



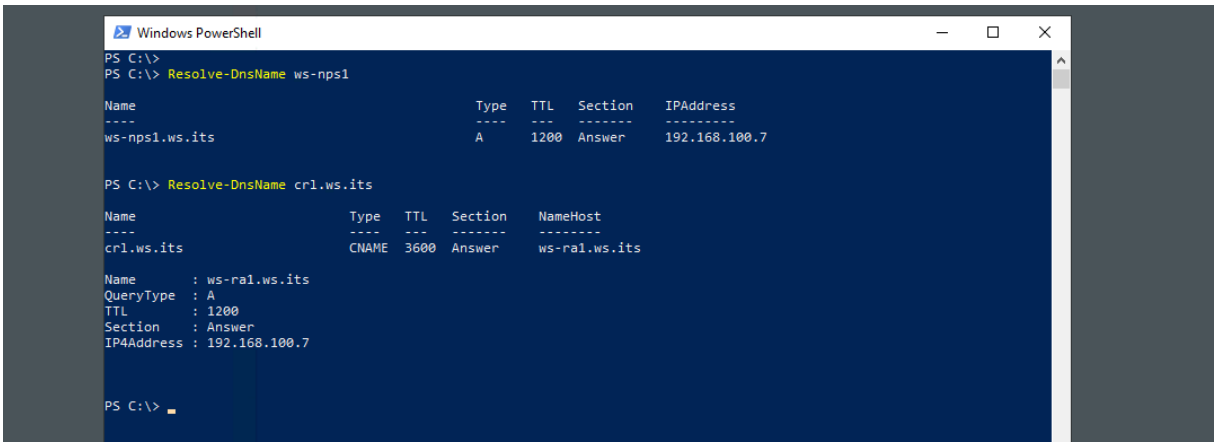
Zeitpunkt	DomainController	Quelle	Benutzer	EreignisID	EreignisSubID	Ereignis	ClientIP	ClientPort	ClientName
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1
2020-01-02 15:05:52	WS-DC2		WS-CA1$	4004		NTLM-Auth to target ' WS-NPS1 ' blocked	192.168.100.6		WS-CA1

Interessant ist dabei der Zeitpunkt. Ab 13:00 des Vortages hat es vermehrt Authentifizierungsversuche gegeben, die aber alle geblockt wurden. Diese Anfragen kamen von meiner PKI, die auf WS-CA1 läuft. Was will denn die PKI von meinem WS-NPS1? Ha, ganz einfach: ich hatte ganz vergessen, dass auf WS-RA1 auch ein Sperrlistenverteilungspunkt platziert war. Diesen konnten die Clients über http ansprechen und die Sperrliste meiner PKI herunterladen. Dazu muss der WS-CA1 aber zuvor seine Sperrlistendatei auf dem Server speichern können. Und dafür ist wiederum eine Anmeldung erforderlich.

Ein Blick in die Konfiguration der Zertifizierungsstelle zeigt den jetzt falschen Eintrag:

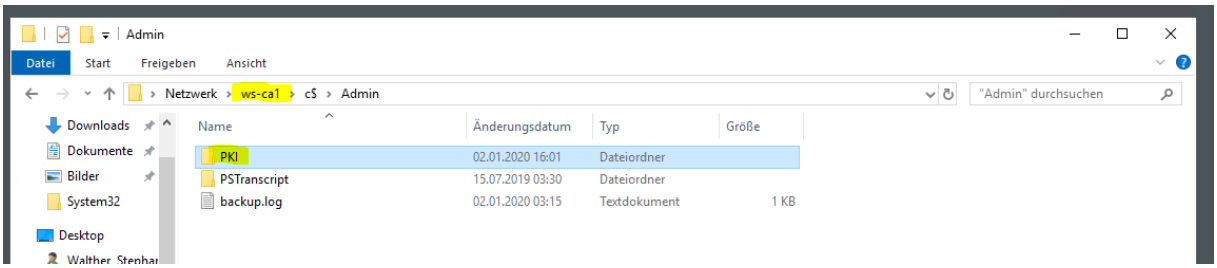


Gleichzeitig ist damit auch klar, dass ich im DNS noch die beiden alten Hostnames mit ihren IP's registriert habe. Denn sonst würde die Verbindung nicht funktionieren:

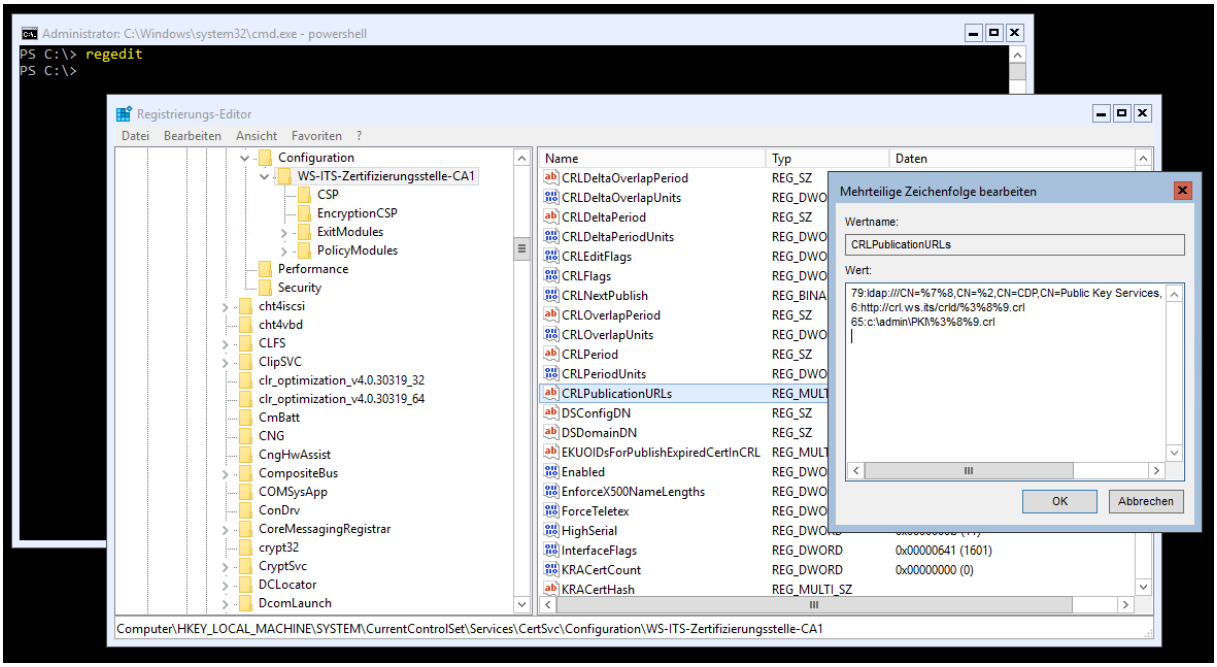


OK, diese Funktion muss angepasst werden. Der alte WS-RA1 war auch ein Webserver durch seine VPN-Funktionalität. Daher konnte er auch die CRL-Datei veröffentlichen. Der neue NPS-Server ist aber kein Webserver mehr. Abgesehen davon macht diese Kombination von PKI und NPS keinen Sinn. Daher verschiebe ich den Endpunkt auf die Zertifizierungsstelle selbst. Das kann ich ja bei der Migration der PKI wieder anpassen.

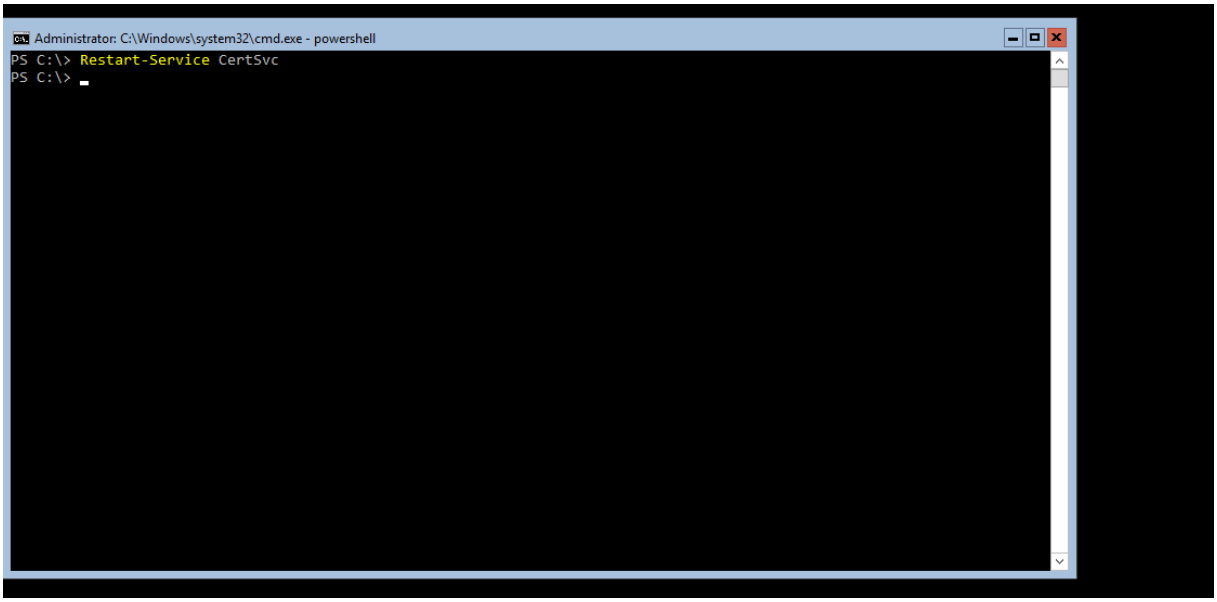
Mein Server WS-CA1 ist ein Server Core mit Windows Server 2016. Dort lege ich ein Verzeichnis für die Sperrlisten an:



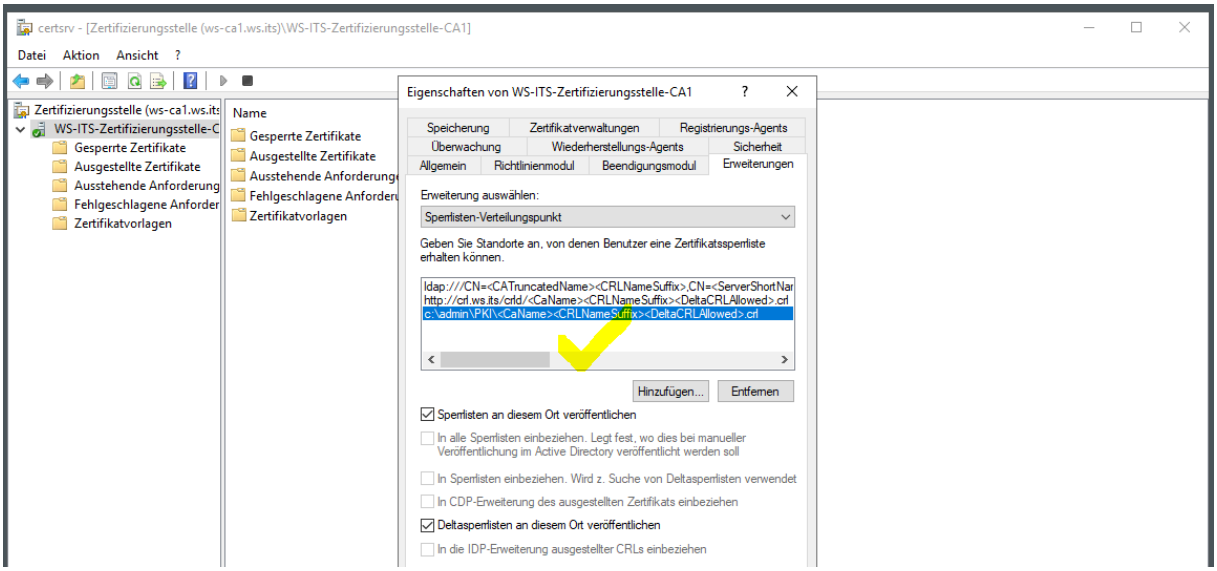
Lokal editiere ich nun den Wert in der Registry. Das geht deutlich schneller als der Umweg über den grafischen Assistenten (dort wäre ein Abändern des Pfades nicht vorgesehen):



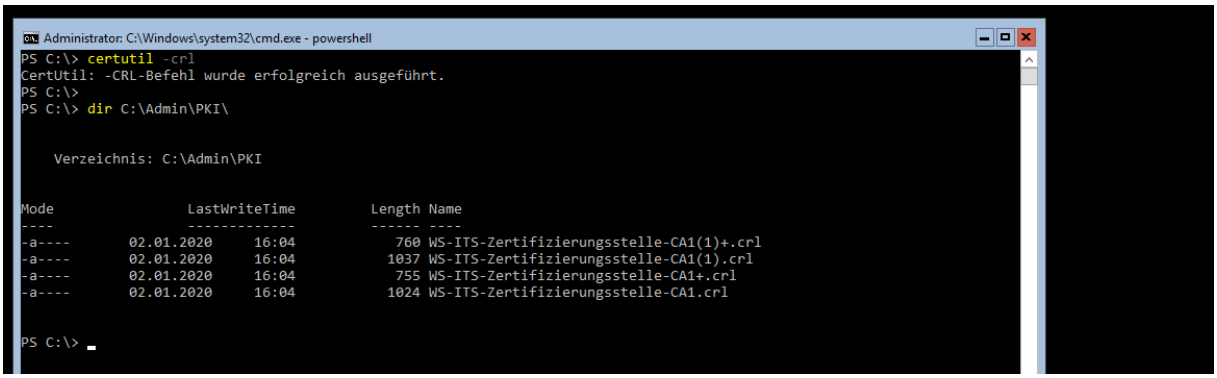
Die Zertifizierungsstelle muss als Service neustarten, damit diese Konfiguration geladen wird:



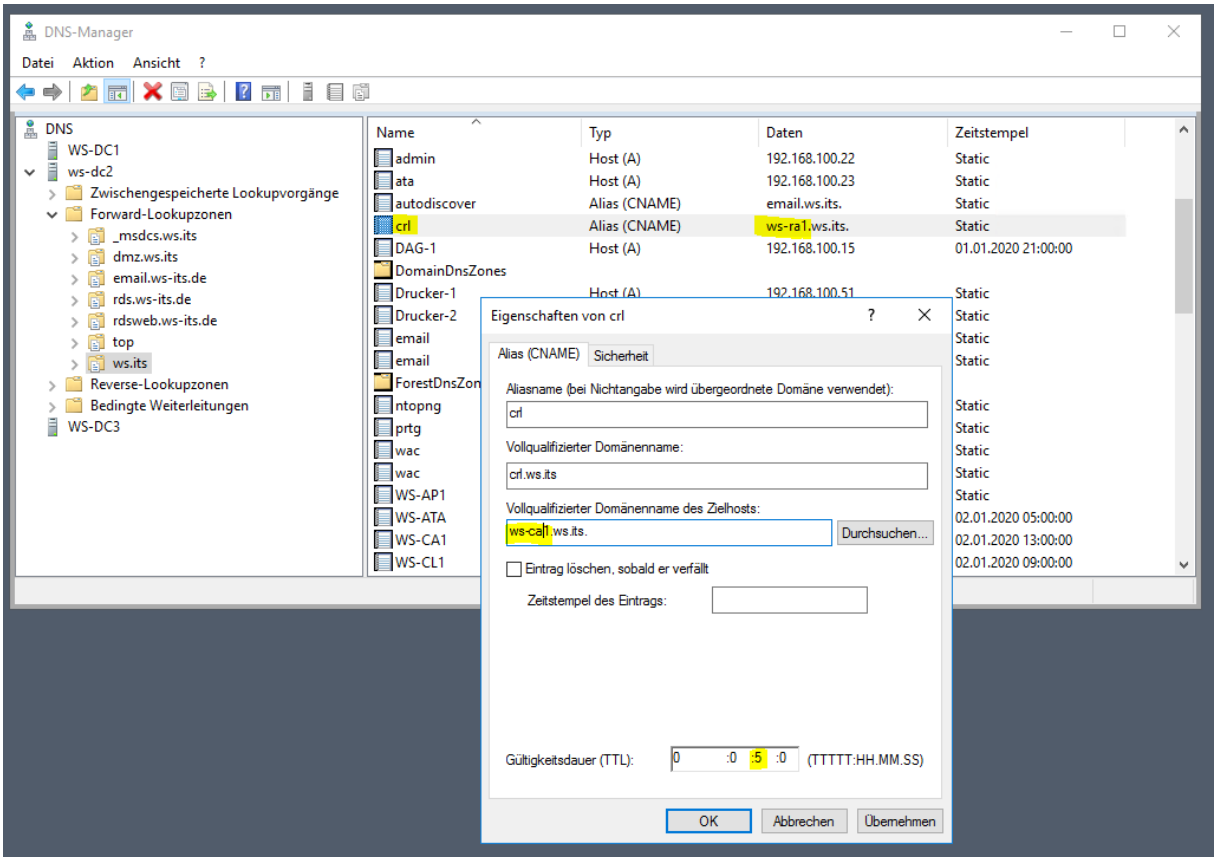
Der Wert wird korrekt in der Verwaltungskonsole angezeigt:



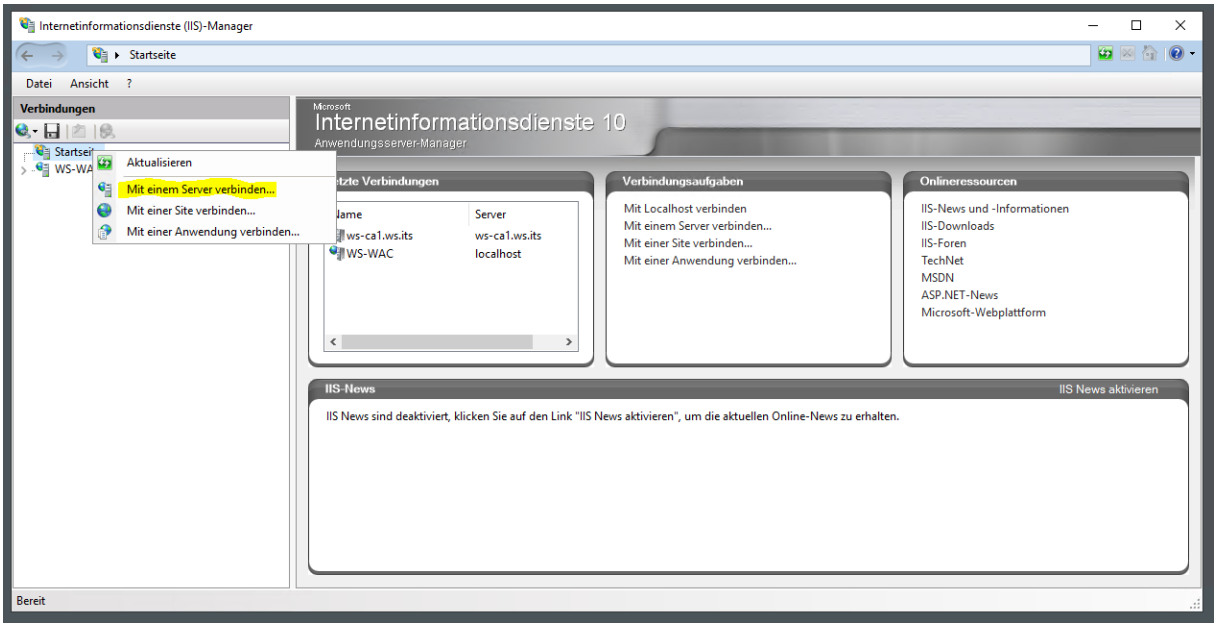
Und auch die Sperrlistendateien tauchen korrekt auf:

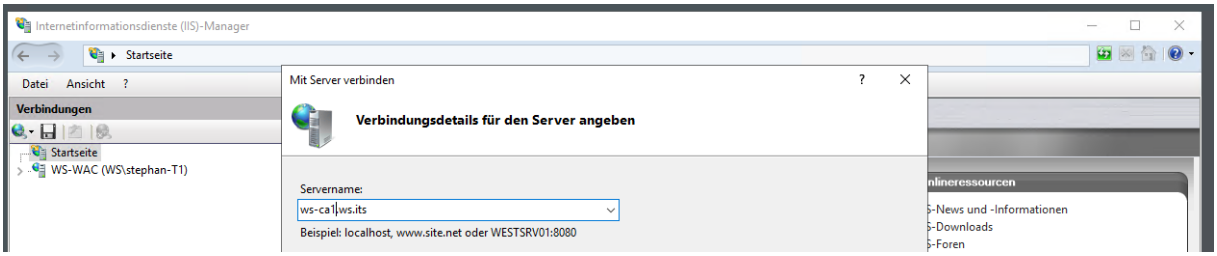


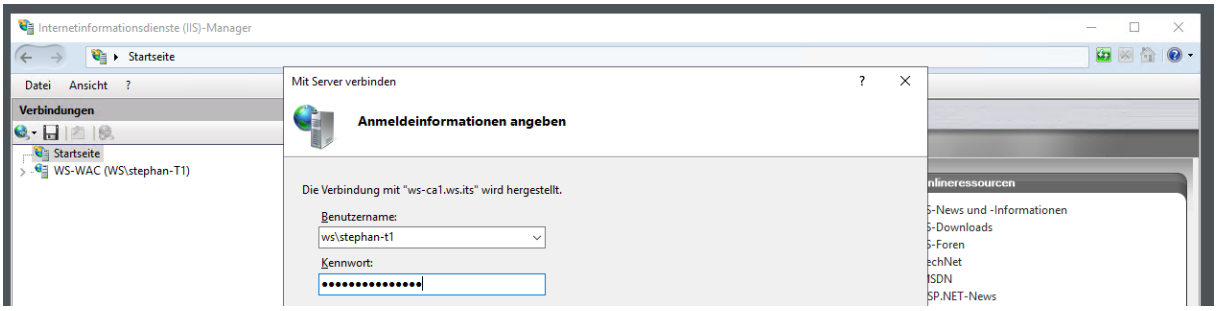
Aber wie sollen meine Clients und Server diese Dateien finden und herunterladen? Ganz einfach: Bisher war der Verteilungspunkt ein Webservice hinter einem CNAME. Dieser zeigte auf meinen alten WS-RA1. Im DNS kann ich den Record nun auf meine Windows PKI zeigen lassen:



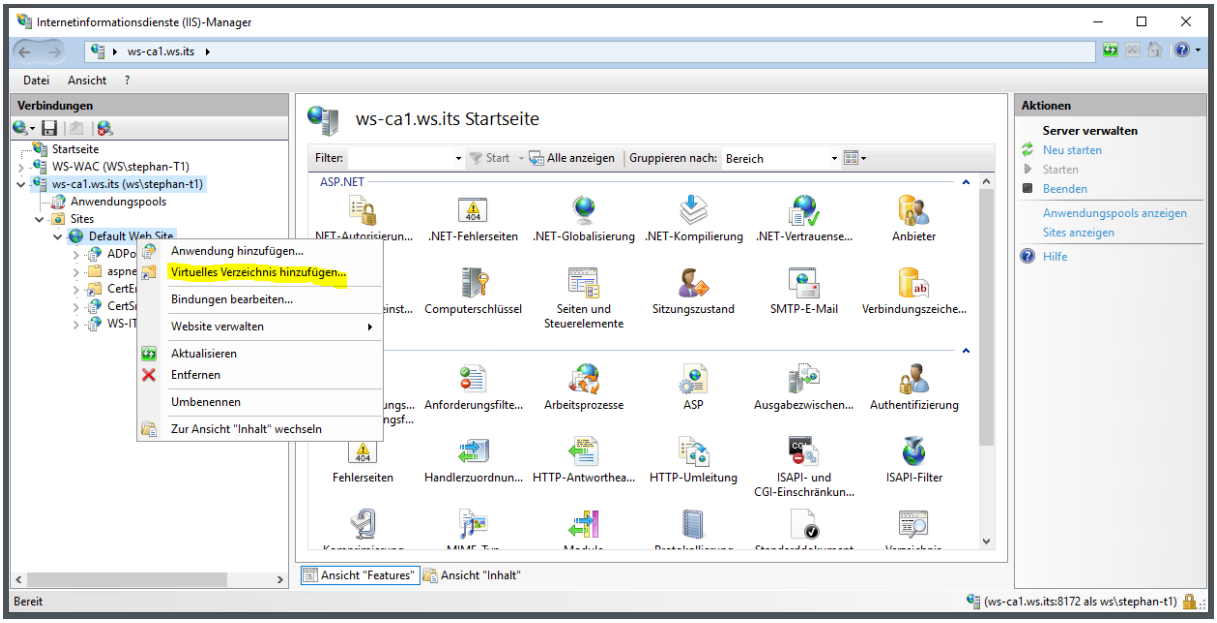
Dazu muss ich dort aber auch ein virtuelles Verzeichnis im IIS erstellen. Dieses fungiert als eine Art http-Freigabe auf den lokalen Ordner. Der Name ist bereits allen Systemen bekannt. Den muss ich also beibehalten. Der IIS ist bereits auf meiner Windows PKI installiert. Aber die Verwaltungstools fehlen, da es ja ein Server Core ist. Also verwende ich die Konsole auf meinem Admin Server und stelle aus dieser eine Verbindung zur PKI her:



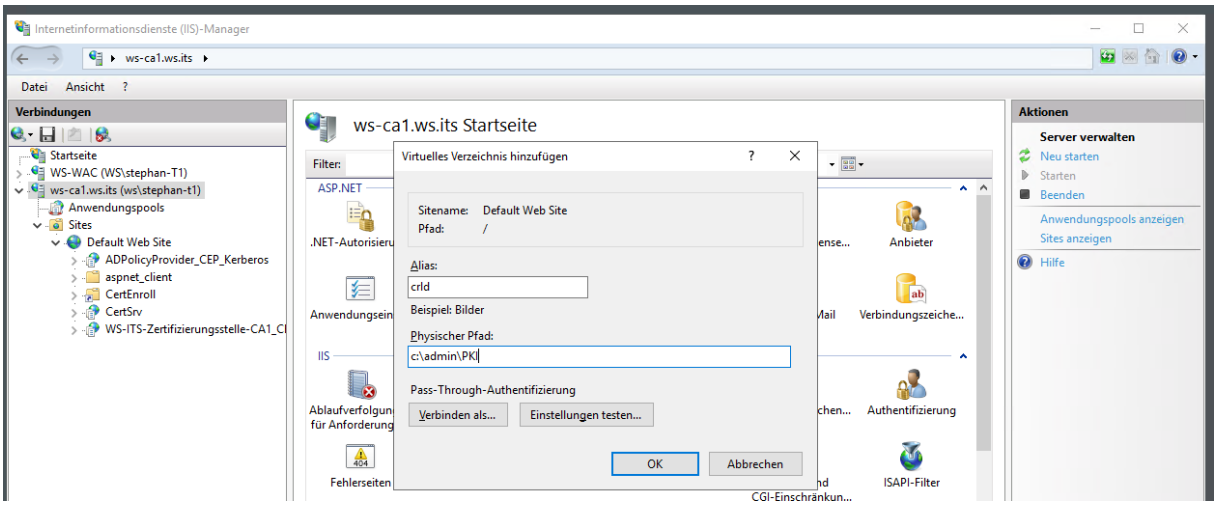




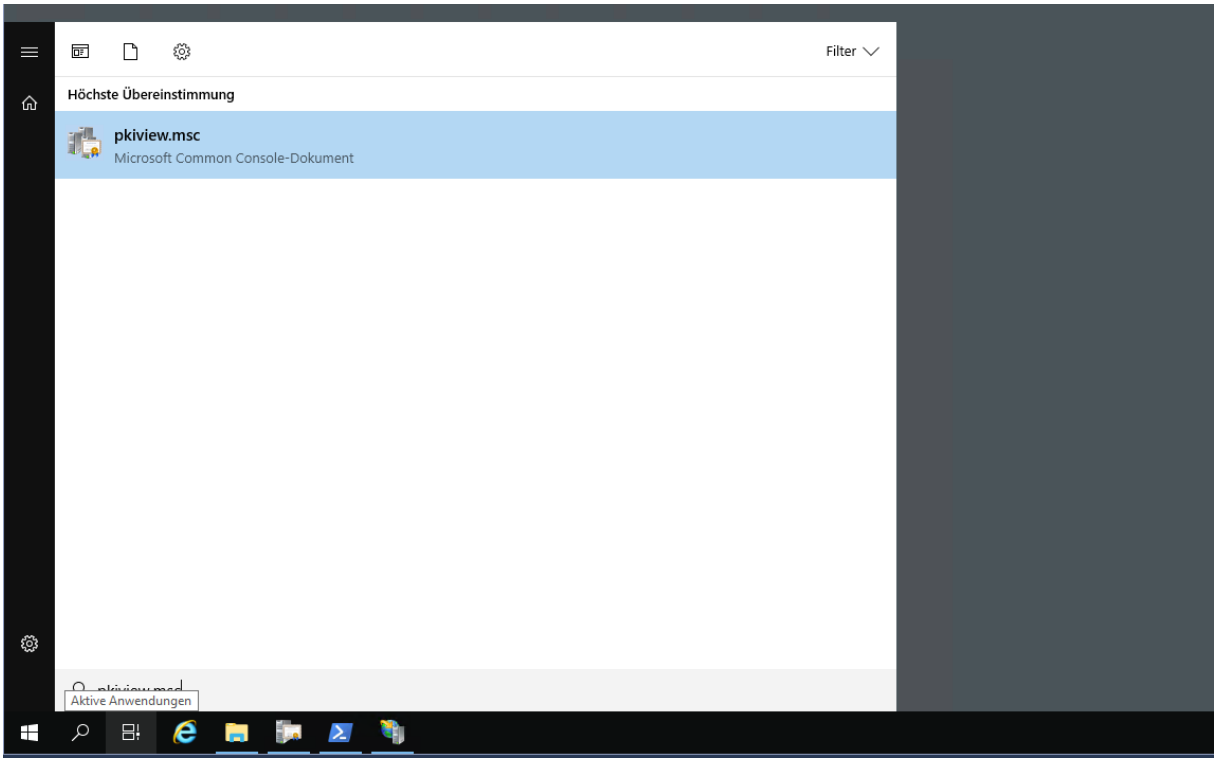
Das virtuelle Verzeichnis hänge ich direkt in die Root des IIS:



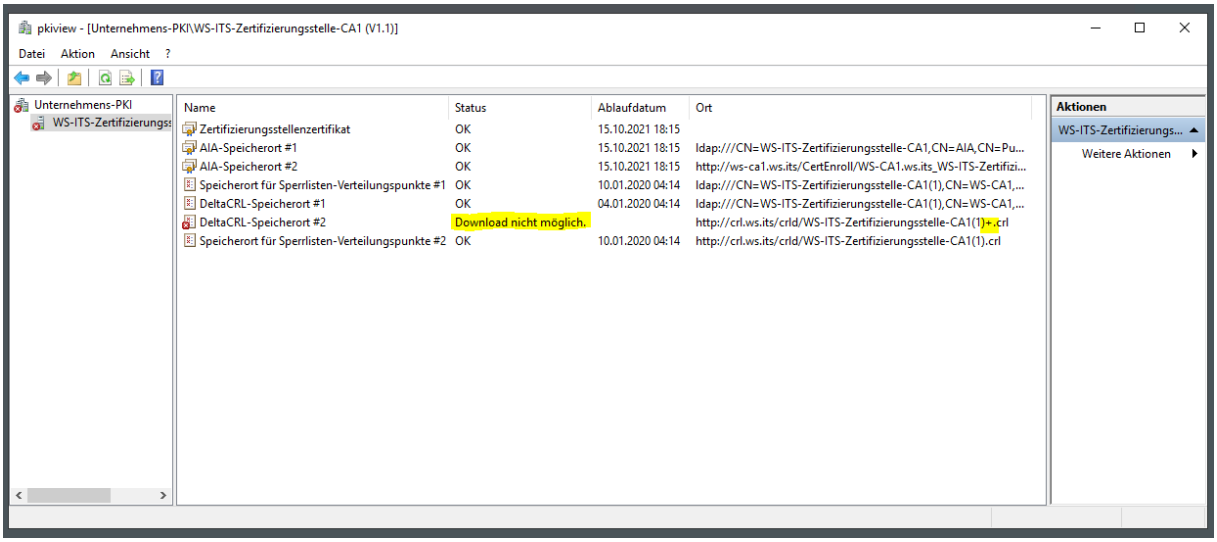
Der Name „crl“ ist bereits veröffentlicht, daher muss ich den beibehalten. Der Eintrag zeigt auf das zuvor erstellte Verzeichnis mit den CRL-Dateien:



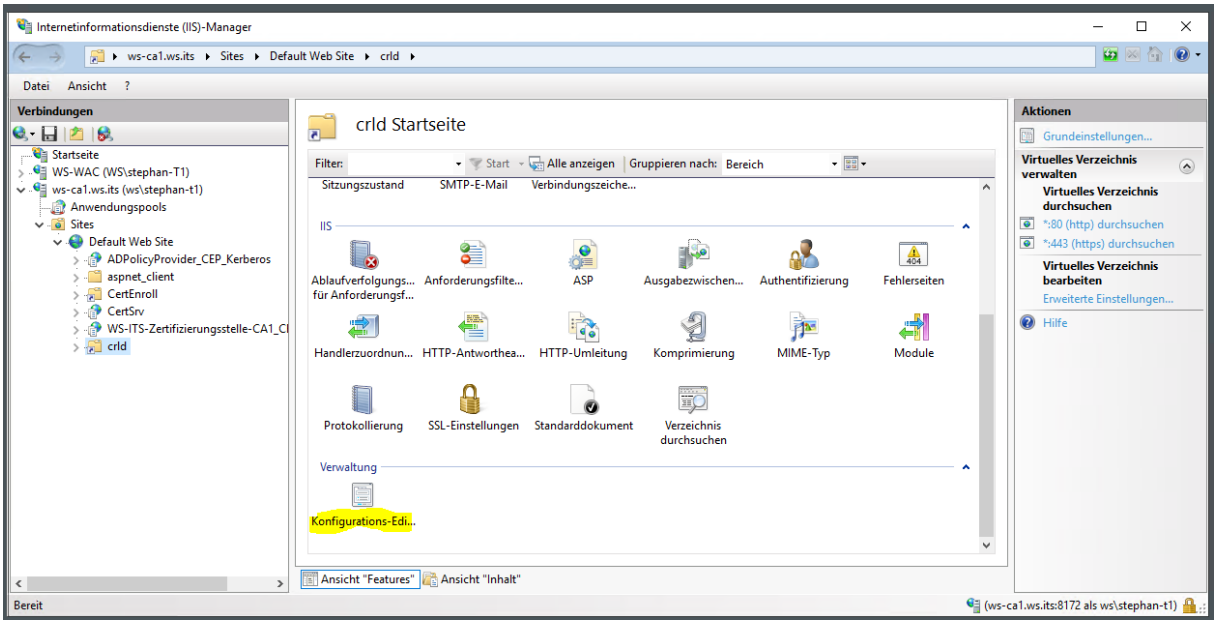
Nach einem Klick auf OK ist alles erledigt. Aber ich kontrolliere immer gerne mit pkiview.msc, ob alle Einstellungen auch aus der Perspektive des Clients erreichbar sind. Daher starte ich das Tool auf meinem Admin Server:



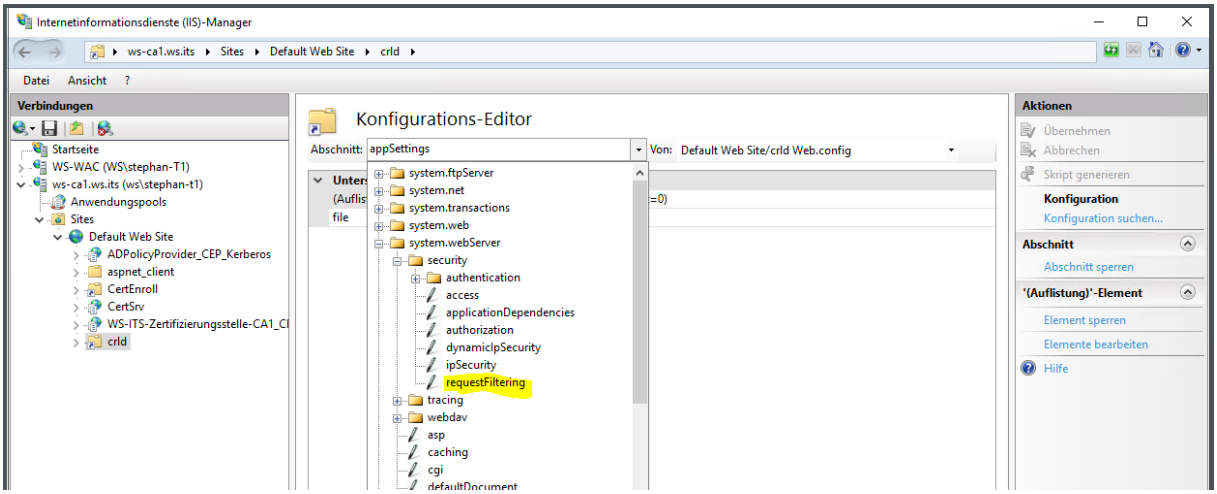
Und der Blick wird mit einem Fehler belohnt. Das ist wohl die Deltasperrliste nicht erreichbar:



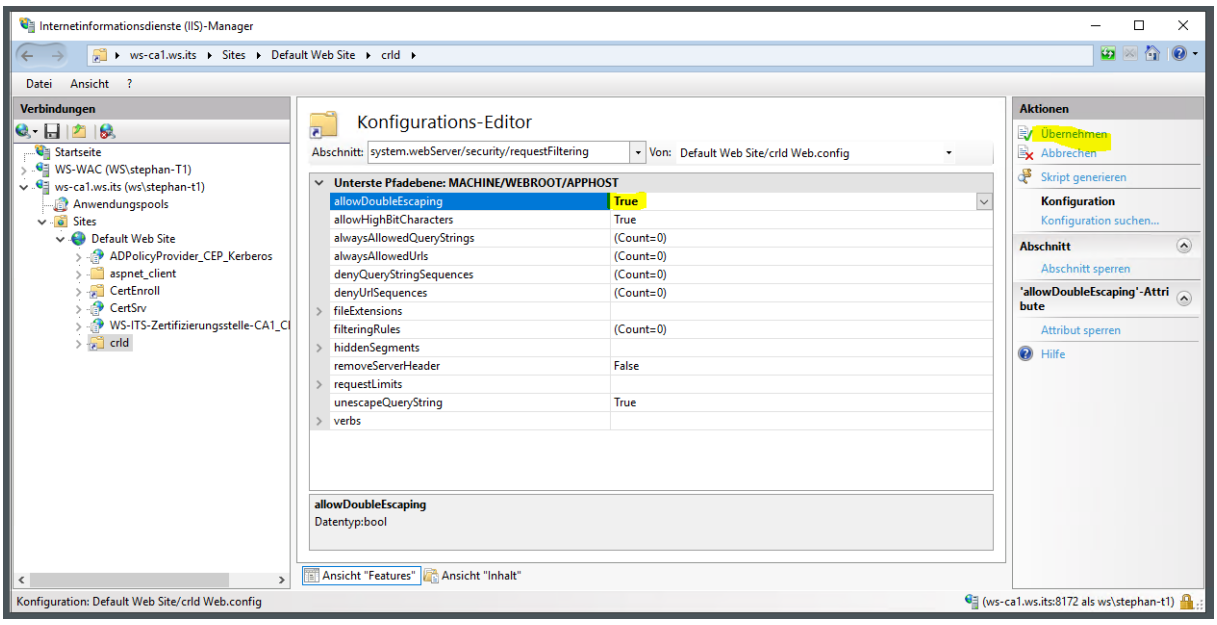
Die Ursache ist klar, wenn man schon einmal mit PKI zu tun hatte: Die Deltasperrliste wird mit einem Pluszeichen im Namen gekennzeichnet. Dieses Zeichen hat aber in einer URL eine andere Bedeutung. Daher reagiert der IIS nicht auf die Anfrage. Das Verhalten lässt sich mit einer IIS-Option anpassen. Diese kann im Konfigurationseditor vorgenommen werden:



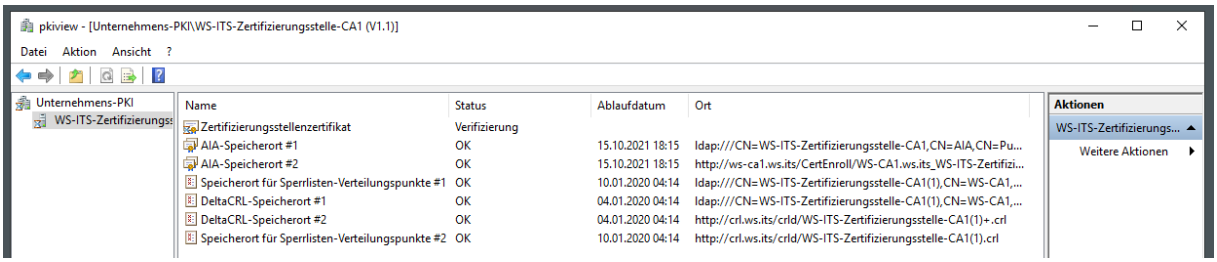
Der anzupassende Wert ist das DoubleEscaping im RequestFiltering:



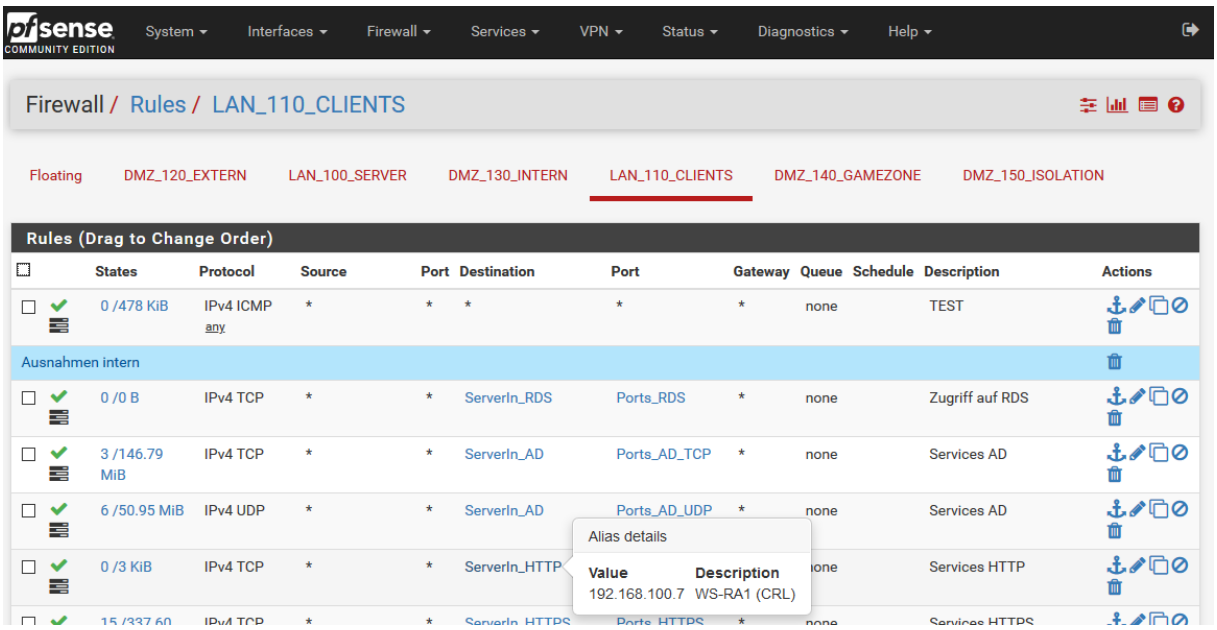
Dieses muss aktiviert werden:



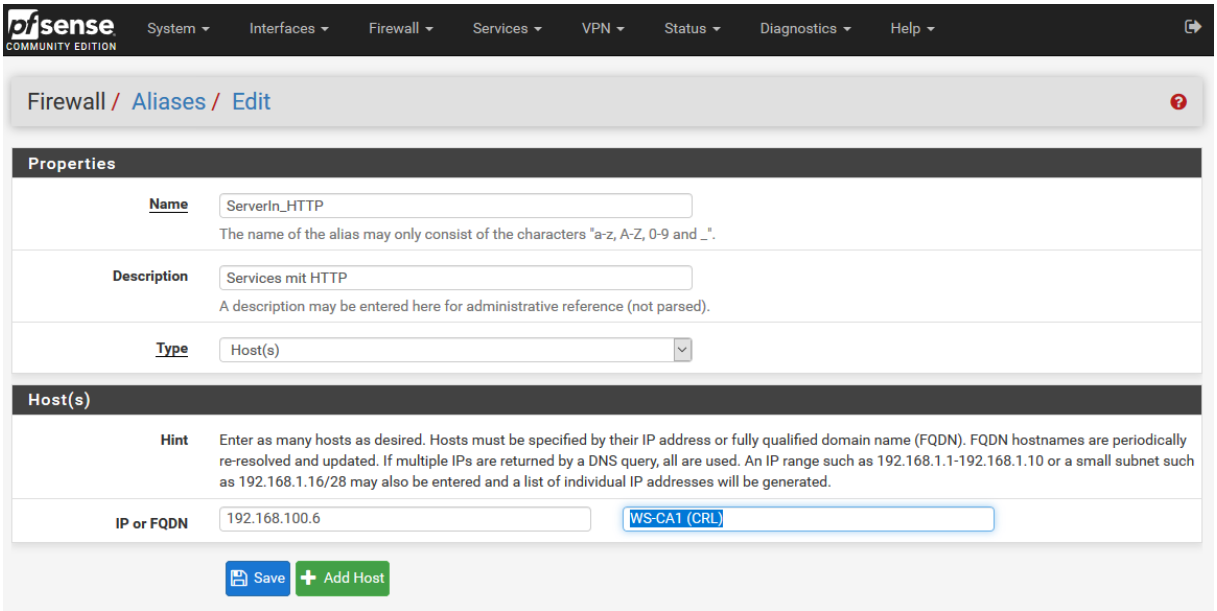
Nach einem IIS-Reset ist die Datei dann im pkiview erreichbar und alles ist grün:



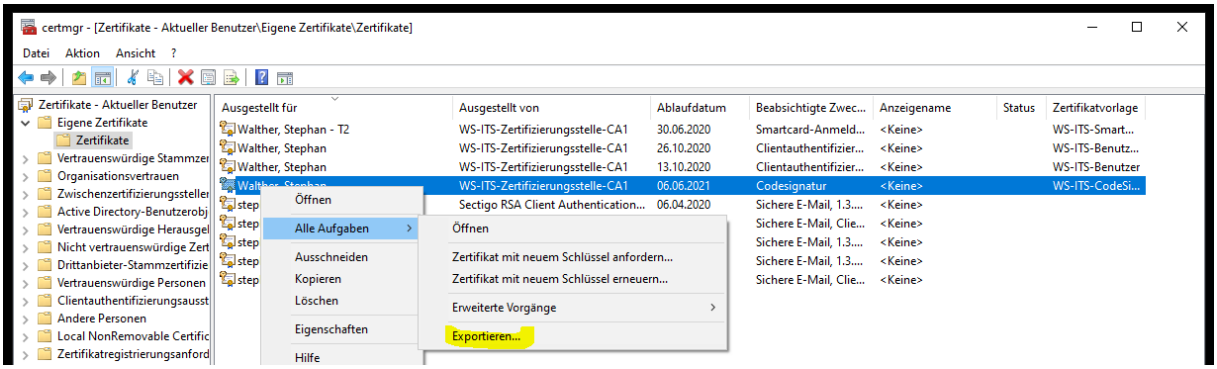
Der Admin Server kann die Datei und den dazugehörigen Server direkt ansprechen. Meine Clients stehen aber in anderen Subnets. Für den Zugriff muss der Zertifizierungsserver über TCP Port 80 (http) erreichbar sein. Dazu habe ich in meiner Firewall entsprechende Regeln definiert. Die aktuelle Regel verweist noch auf den alten WS-RA1:

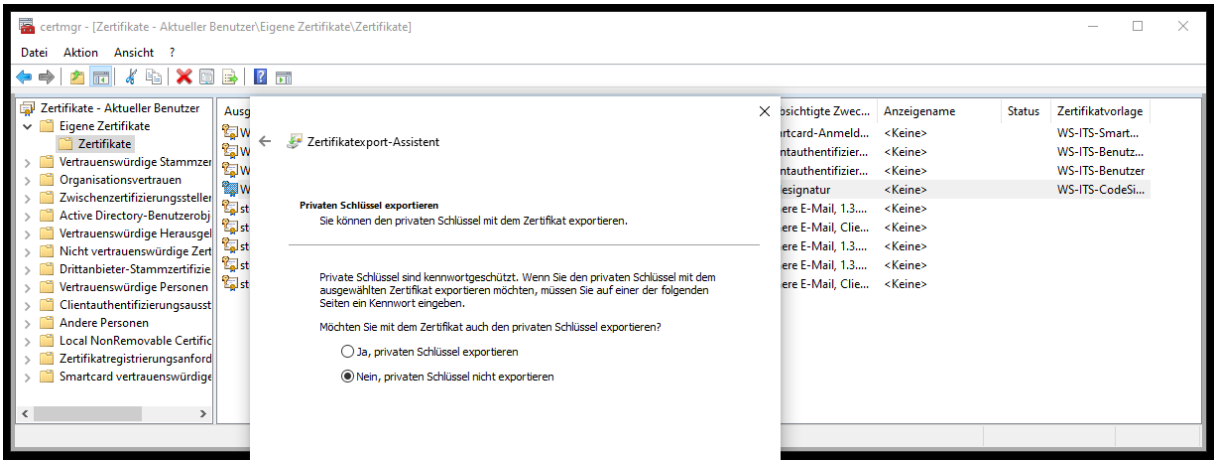


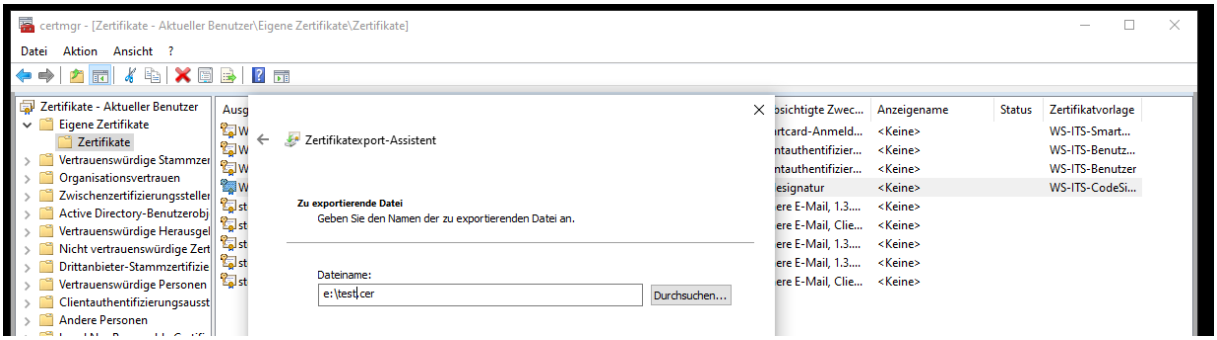
Hinter der Regel steht ein Alias. In diesem gebe ich nun die IP-Adresse meines WS-CA1 an:



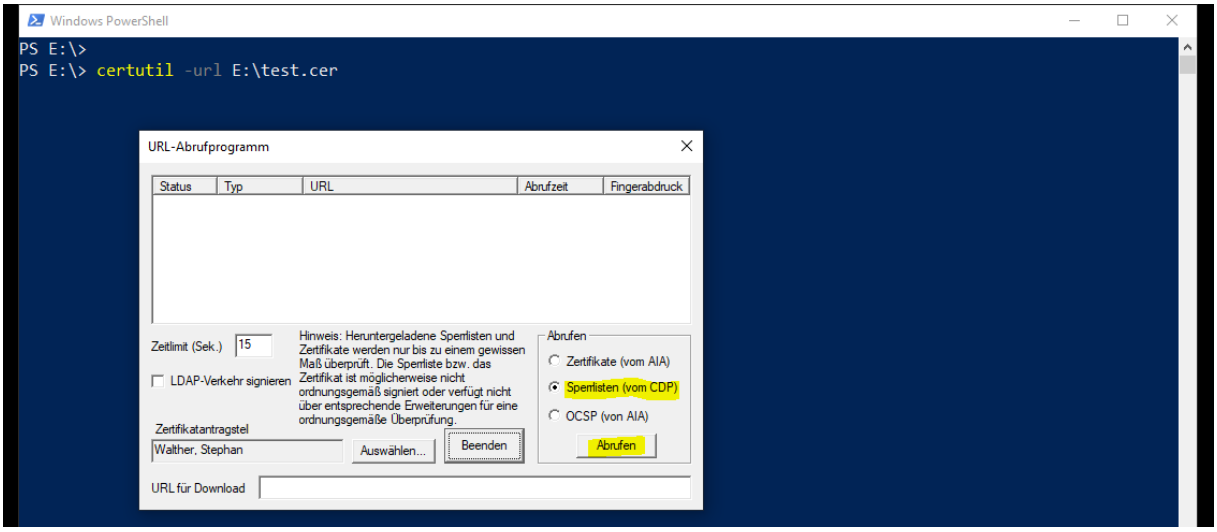
Nun wird es Zeit für einen Test von einem Client aus. Dazu nutze ich mein Notebook. Hier sind einige interne Zertifikate vorhanden. Ich exportiere eines davon in eine cer-Datei:



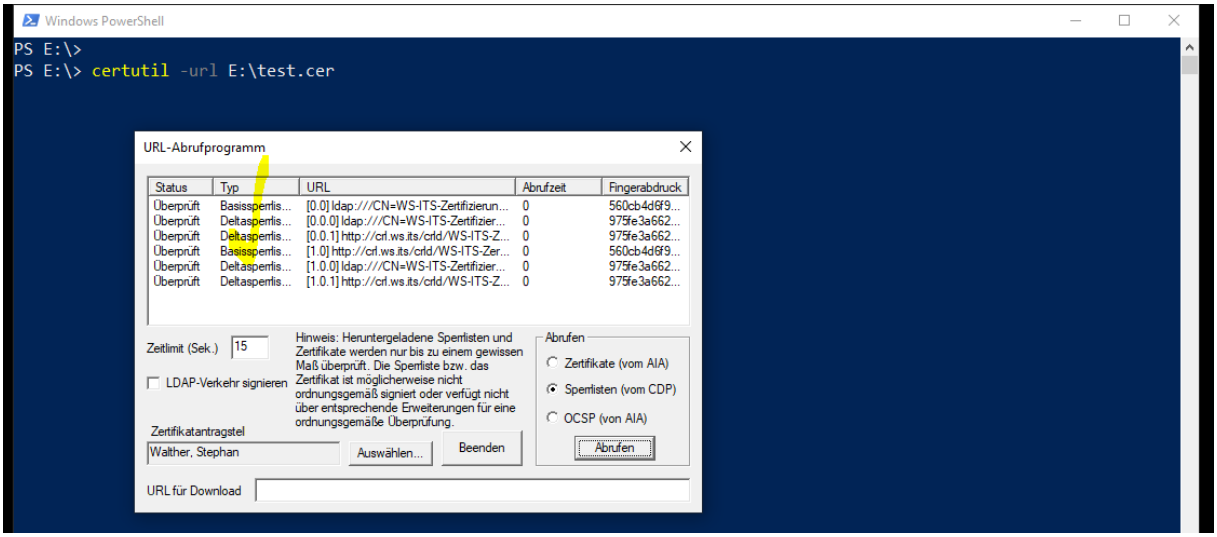




Mit certutil kann ich nun einen Sperrlisten-Test vornehmen:



Das Tool prüft, ob die Sperrlistendaten von den Servern geladen werden können:



Das sieht gut aus. Damit sollte diese Migration abgeschlossen sein.