



Windows Server 2012 R2

Active Directory Domain Services

Active Directory Domain Services 2012 R2 -
Grundkonfiguration - Teil 1

Active Directory Domain Services 2012 R2 - Grundkonfiguration - Teil 1

Inhalt

Windows Server Vorbereitung	2
Netzwerkarte(n) konfigurieren für eine saubere Namensauflösung	3
Reverse-Lookupzone im DNS-Server einrichten	4
DNSSEC einrichten (Optional zu Testzwecken)	7
Vertrauensanker (Trustanchors) im Netzwerk verteilen.....	14
Mehr Sicherheit mit SocketPoolSize und CacheLockingPercent.....	17
Gruppenrichtlinie für DNSSEC aktivieren	18
Alterung der DNS Serverzonen einstellen.....	20
Active Directory Objekte vor versehentlichen löschen schützen.....	23
Zeitserver für den PDC Emulator konfigurieren.....	24
Active Directory – Standort und Dienste.....	25

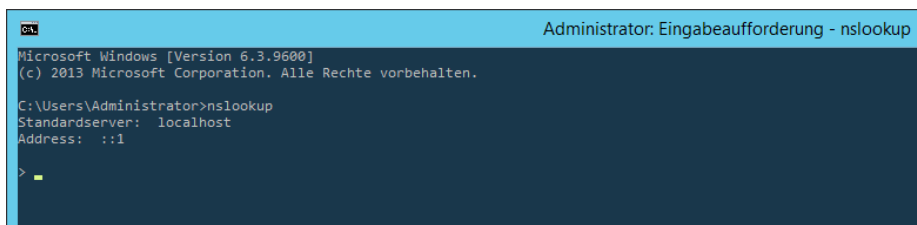
Windows Server Vorbereitung

Um dieses HowTo schreiben zu können, wurde das Ganze in einer Virtuellen Umgebung mit Oracle VM VirtualBox nachgestellt.

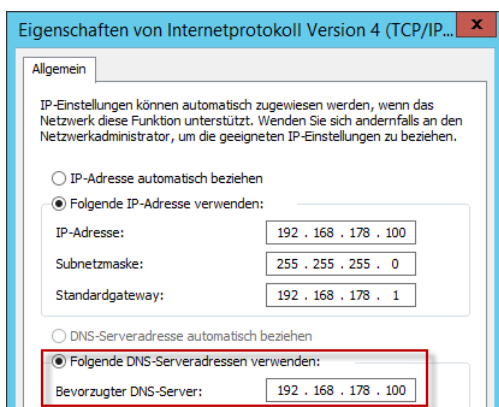
- ↳ Microsoft Windows Server 2012 R2 (Deutsch)
- ↳ Active Directory Domänendienste - Server Rolle
- ↳ DNS-Server Rolle

Netzwerkarte(n) konfigurieren für eine saubere Namensauflösung

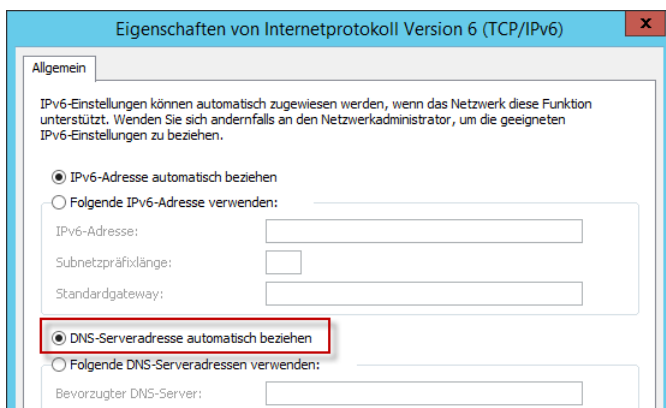
Nach dem der erste Domänencontroller im Netzwerk installiert wurde, beginnen wir den Server sauber zu konfigurieren.



Wenn man nun zum ersten Mal eine CMD Konsole öffnet und den Befehl **nslookup** absetzt, sieht ihr folgendes Ergebnis, das hat den Grund weil die Installation der Active Directory Server Rolle, die Loopback Adresse als Bevorzugten DNS-Server setzt, diese müssen wir anpassen.

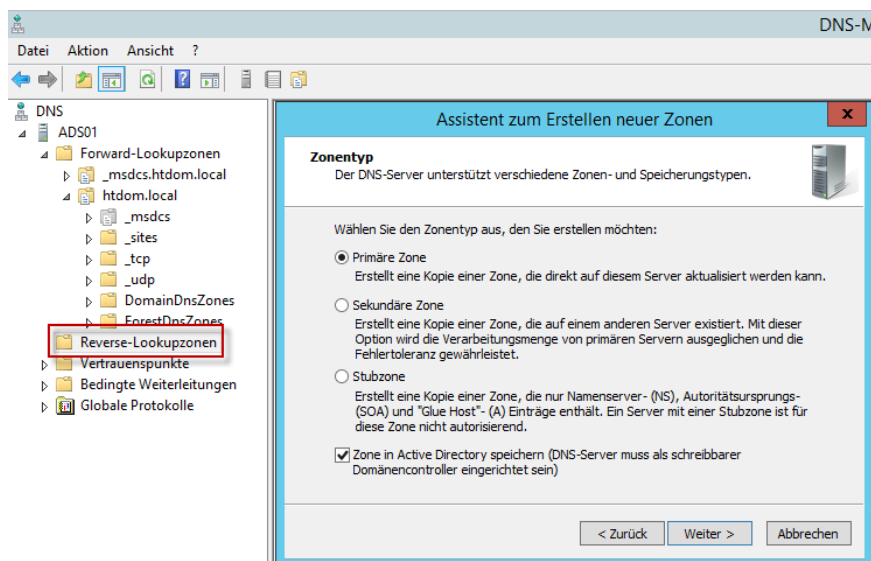


Hier setzen wir die Original Server IP-Adresse ein.

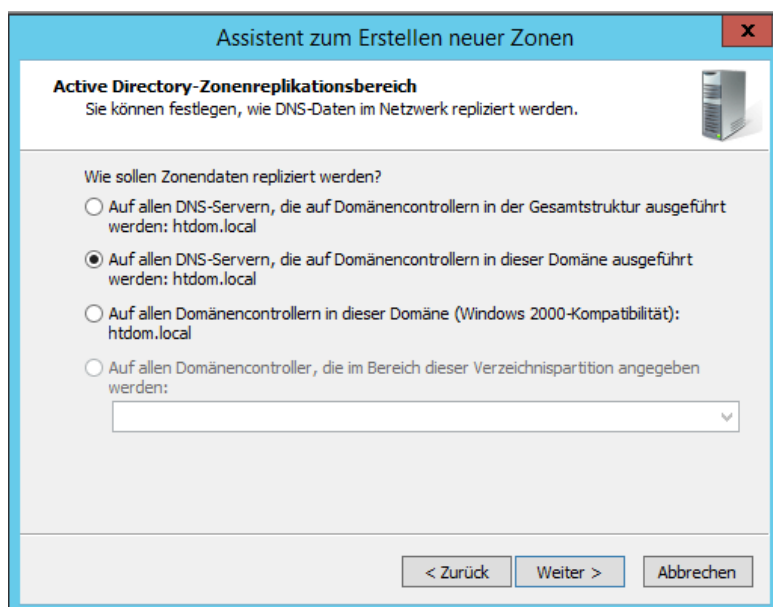


Im den Einstellungen der IPv6, setzen wir die Einstellung auf automatisch beziehen.

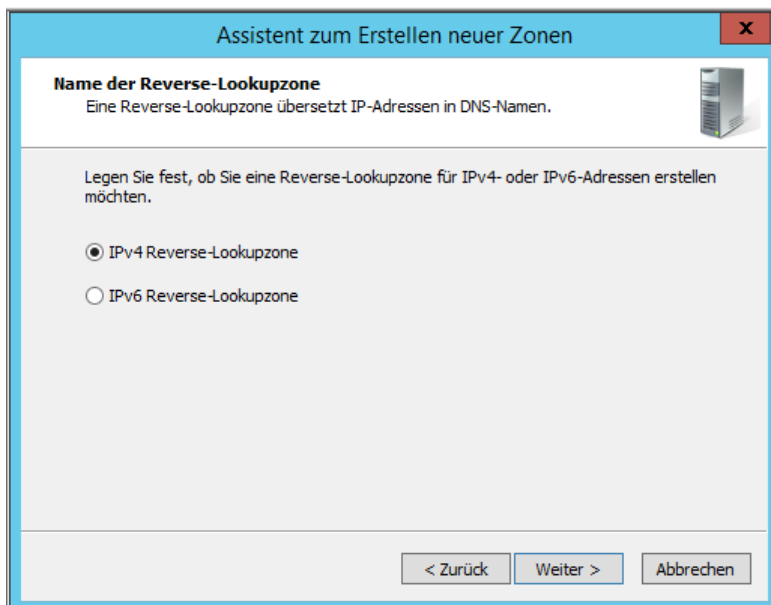
Reverse-Lookupzone im DNS-Server einrichten



Im nächsten Schritt richten wir auf den DNS-Server eine Reverse-Lookupzone ein, diese benötigen wir um IP-Adressen in Namen aufzulösen.



Diese Zone veröffentlichen wir für die komplette Domain.



Assistent zum Erstellen neuer Zonen

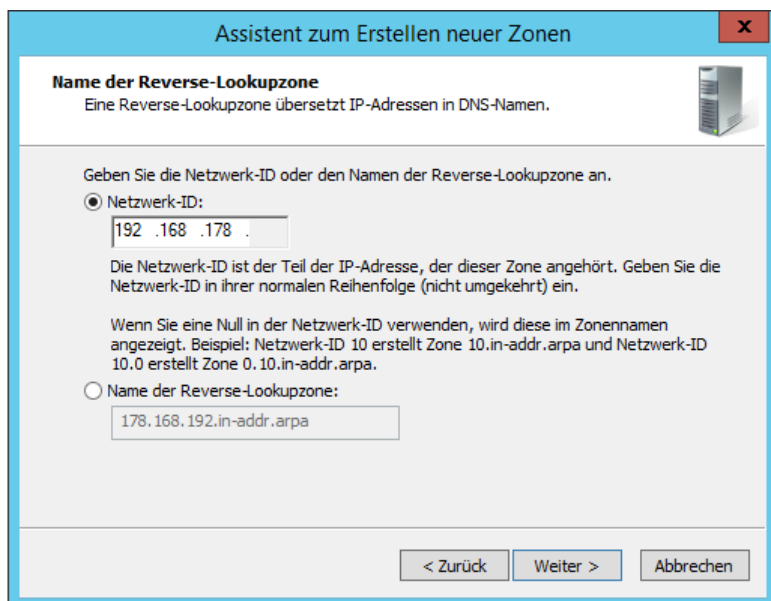
Name der Reverse-Lookupzone
Eine Reverse-Lookupzone übersetzt IP-Adressen in DNS-Namen.

Legen Sie fest, ob Sie eine Reverse-Lookupzone für IPv4- oder IPv6-Adressen erstellen möchten.

☒ IPv4 Reverse-Lookupzone
☐ IPv6 Reverse-Lookupzone

< Zurück Weiter > Abbrechen

Wir richten eine IPv4 Reverse-Lookupzone ein.



Assistent zum Erstellen neuer Zonen

Name der Reverse-Lookupzone
Eine Reverse-Lookupzone übersetzt IP-Adressen in DNS-Namen.

Geben Sie die Netzwerk-ID oder den Namen der Reverse-Lookupzone an.

☒ Netzwerk-ID:
192.168.178.

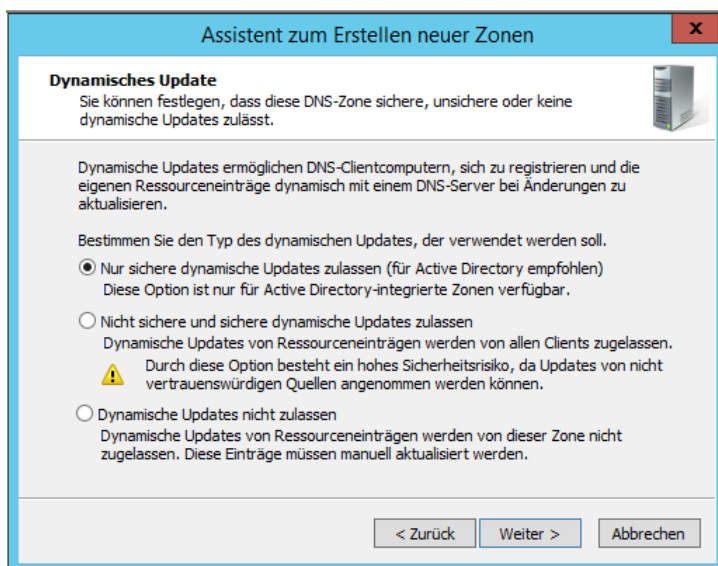
Die Netzwerk-ID ist der Teil der IP-Adresse, der dieser Zone angehört. Geben Sie die Netzwerk-ID in ihrer normalen Reihenfolge (nicht umgekehrt) ein.

Wenn Sie eine Null in der Netzwerk-ID verwenden, wird diese im Zonennamen angezeigt. Beispiel: Netzwerk-ID 10 erstellt Zone 10.in-addr.arpa und Netzwerk-ID 10.0 erstellt Zone 0.10.in-addr.arpa.

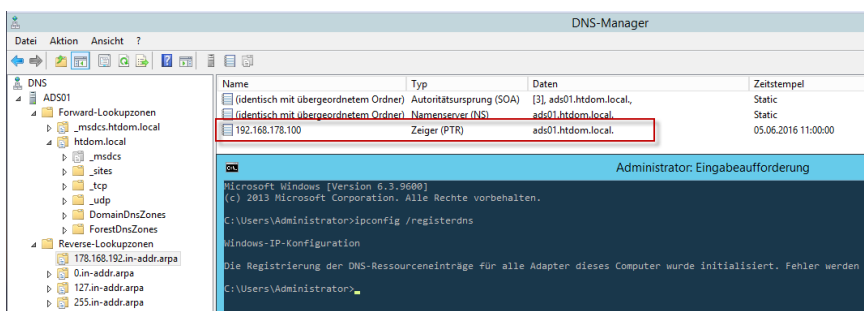
☐ Name der Reverse-Lookupzone:
178.168.192.in-addr.arpa

< Zurück Weiter > Abbrechen

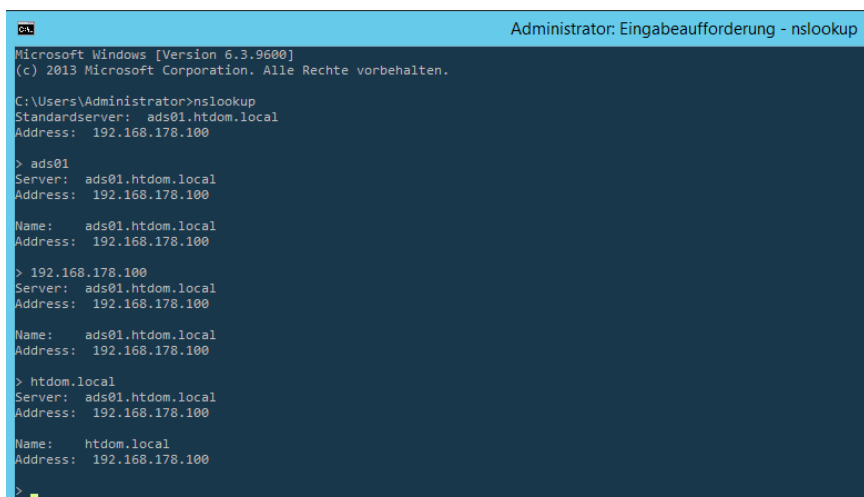
Geben unser Subnetzdaten an.



Wählen nur sichere dynamische Updates zulassen aus.



Nach dem die Reverse-Lookupzone eingerichtet wurde, setzen wir den Befehl **ipconfig /registerdns** ab, um den Domaincontroller selbst in der Reverse-Lookupzone zu registrieren.



Nun können wir per **nslookup** alle Kombinationen (Computernamen, IP-Adresse, Domainname) testen. Wenn wir hier überall ein sauberes Ergebnis zurückbekommen, funktioniert die Namensauflösung einwandfrei.

DNSSEC einrichten (Optional zu Testzwecken)

Zu Testzwecken habe ich mal auf den DNS-Server alle Primären Zonen per DNSSec signiert. Wikipedia und Microsoft schreiben folgendes zu DNSSec.

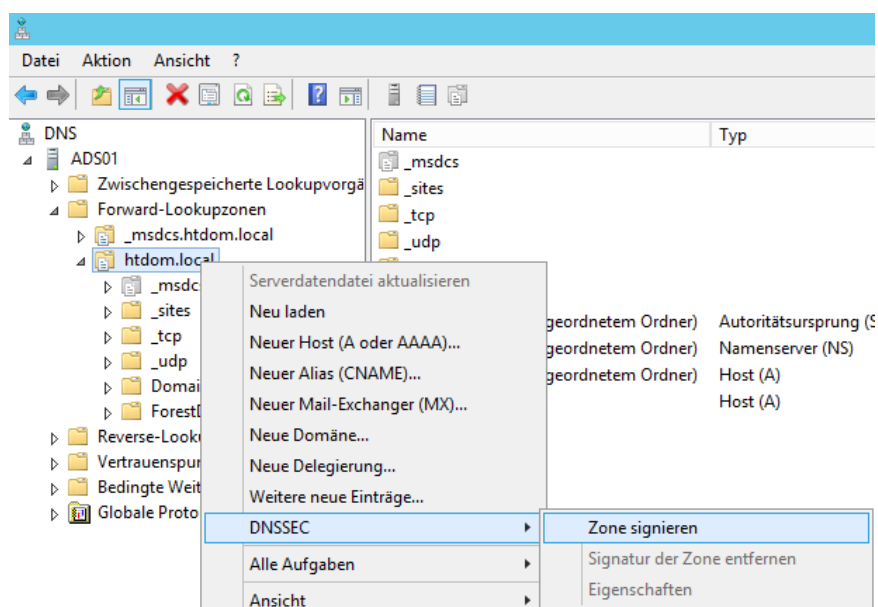
https://de.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

[https://technet.microsoft.com/de-de/library/dn593694\(v=ws.11\).aspx](https://technet.microsoft.com/de-de/library/dn593694(v=ws.11).aspx)

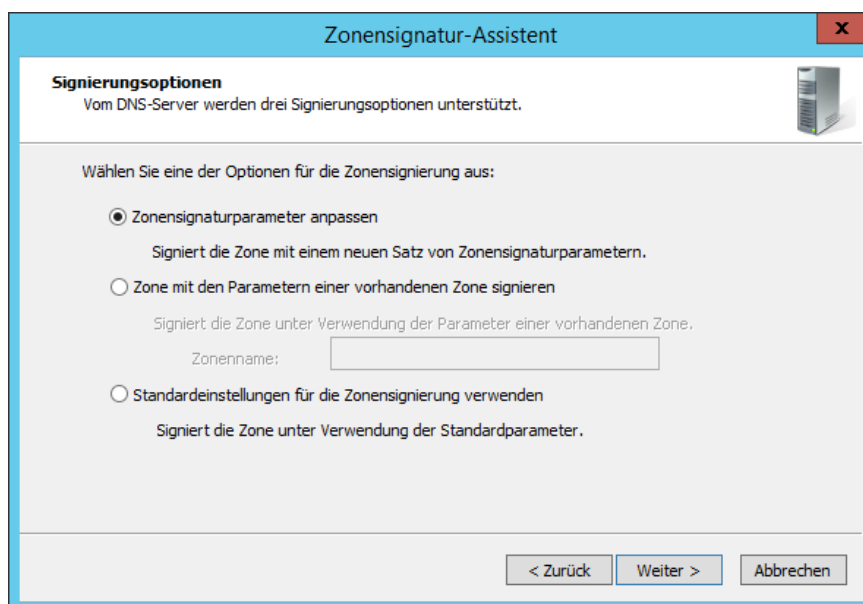
[https://technet.microsoft.com/de-de/library/hh831411\(v=ws.11\).aspx](https://technet.microsoft.com/de-de/library/hh831411(v=ws.11).aspx)

„DNSSEC verwendet ein asymmetrisches Kryptosystem. Der „Besitzer“ einer Information – in der Regel der Master-Server, auf dem die abzusichernde Zone liegt – unterzeichnet jeden einzelnen Record mittels seines geheimen Schlüssels (engl. private key). DNS-Clients können diese Unterschrift mit dem öffentlichen Schlüssel (engl. public key) des Besitzers validieren und damit Authentizität und Integrität überprüfen.“

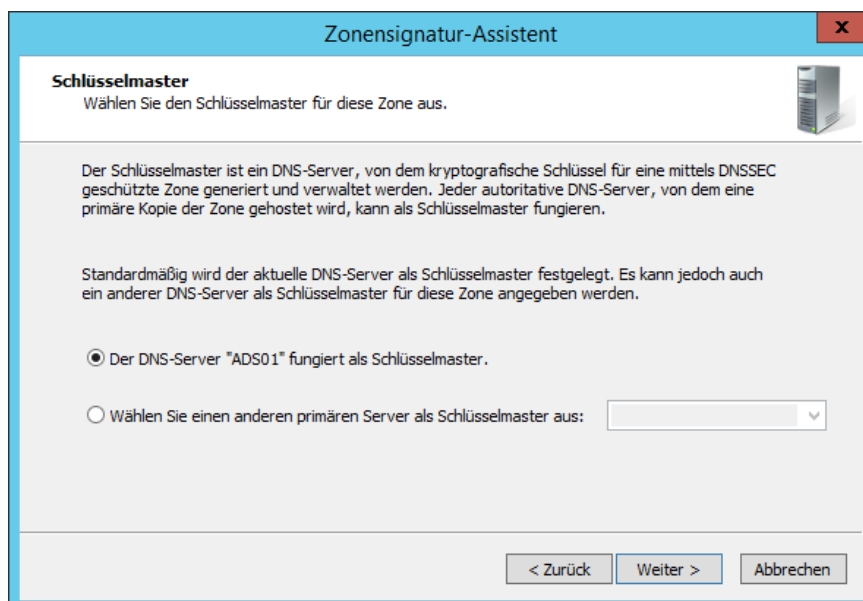
Da ich in der Vergangenheit vieles darüber gelesen habe, wollte ich es mal ausprobieren und mit ein paar Screenshots dokumentieren. Nun muss ich für mich selbst noch herausfinden wieviel mehrnutzen das Ganze hat.



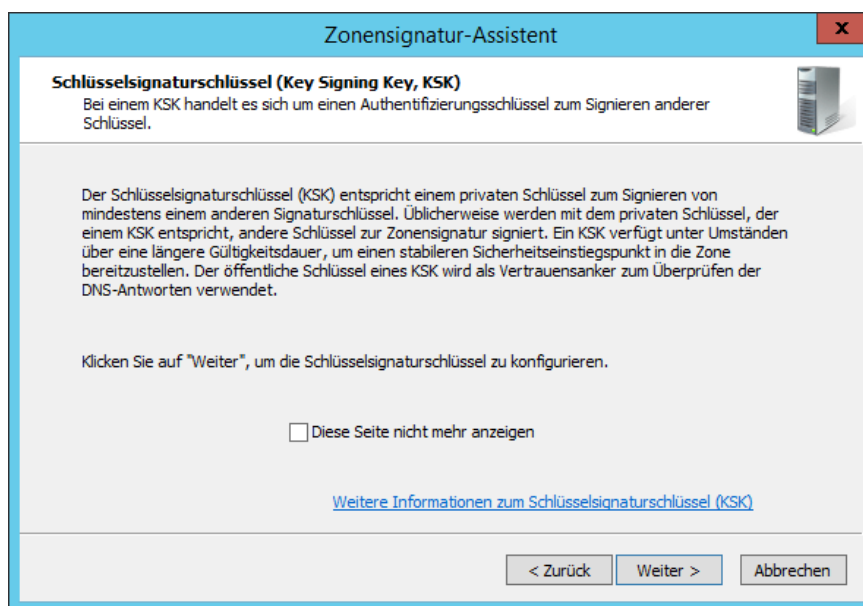
Um nun die Zone zu signieren, klickt man auf den Domaineintrag und wählt im Kontextmenü – **DNSSEC – Zone signieren**.



Da es der erste Versuch ist, möchte ich gerne die Zonensignaturparameter manuell anpassen.

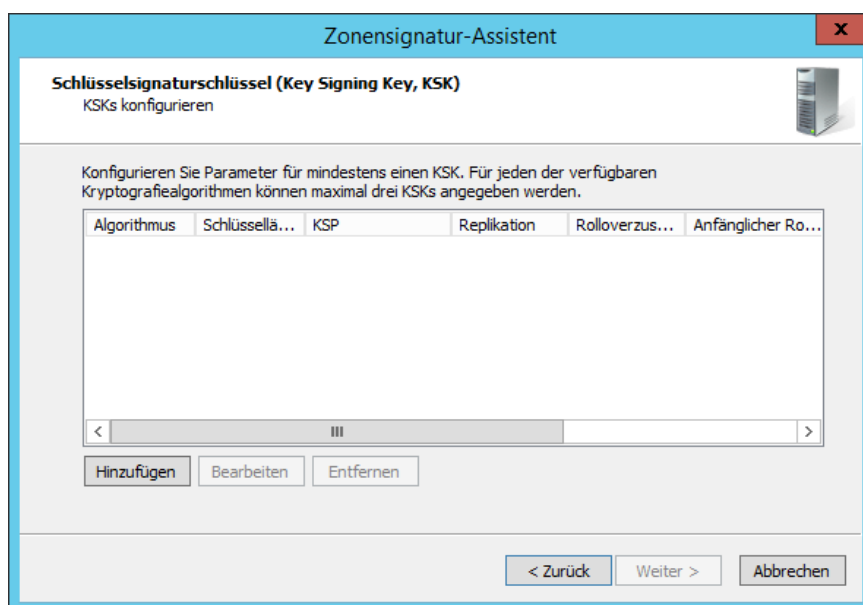


Mein erster Domaincontroller/DNS-Server ist natürlich auch der Schlüsselmaster.

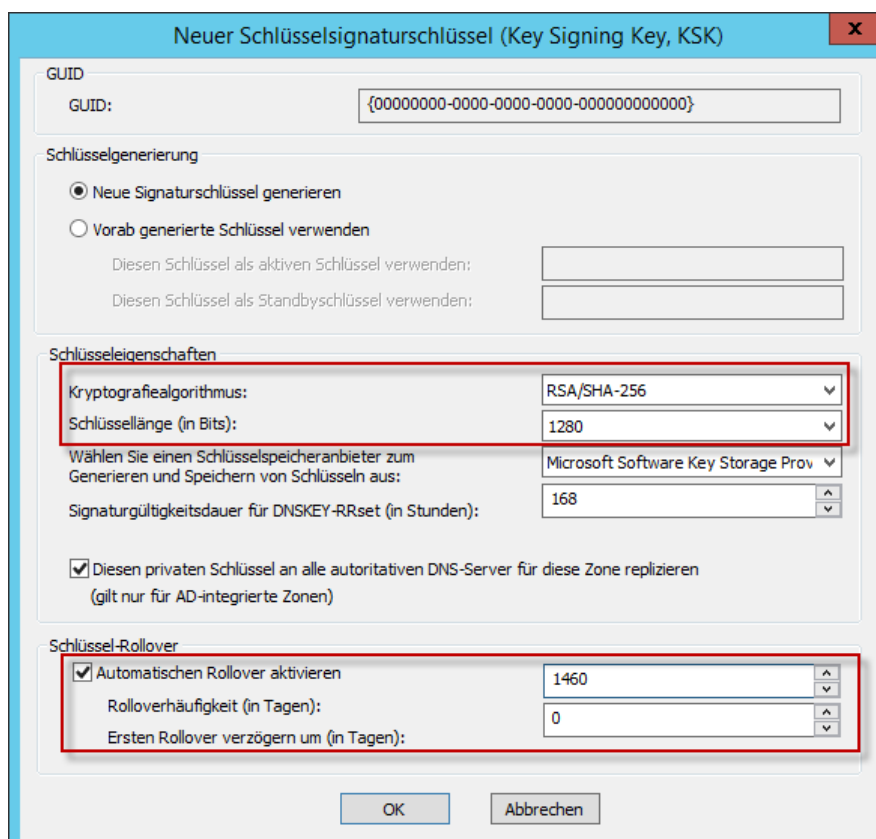


Im nächsten Schritt erstellen wir den **Key Signing Key, KSK**, dieser private KSK Schlüssel wird benötigt um den spätere privaten **Zone Signing Keys (ZSK)** zu signieren.

[https://technet.microsoft.com/de-de/library/dn593682\(v=ws.11\).aspx](https://technet.microsoft.com/de-de/library/dn593682(v=ws.11).aspx)



Hier klicken wir auf Hinzufügen.



Bei den Schlüsseleigenschaften wählen wir für den Kryptografiealgorithmus mindestens **RSA/SHA 256** aus, seit dem 31. Dezember 2015 werden von allen Organisationen empfohlen den SHA-1-Hashalgorithmus durch den SHA-2 zu ersetzen, der SHA-1-Hashalgorithmus wird für ungültig erklärt.

Der **DNSSEC Good Practices Guide** schlägt bei der Schlüssellänge eine Bittiefe von 1280 Bit und maximal 4 Jahre Lebensdauer vor.

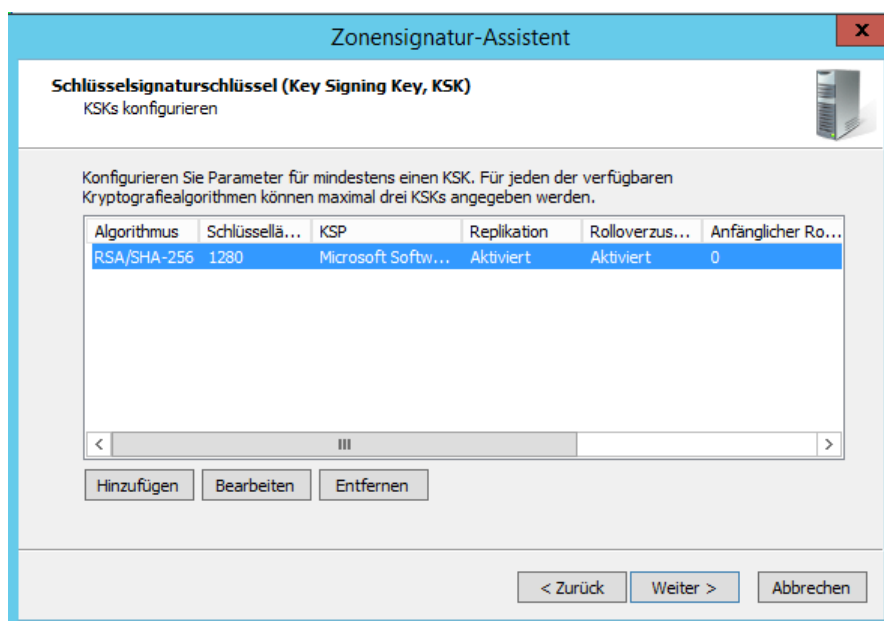
[Good Practices Guide for Deploying DNSSEC](#)

Der Automatische Rollover sollte aktiviert werden, damit der KSK Schlüssel in Regelmäßigen Abständen erneuert wird, in diesem Beispiel können wir den Standard von 755 Tagen belassen oder wir wählen nach Best Practices die vorgeschlagenen 4 Jahre.

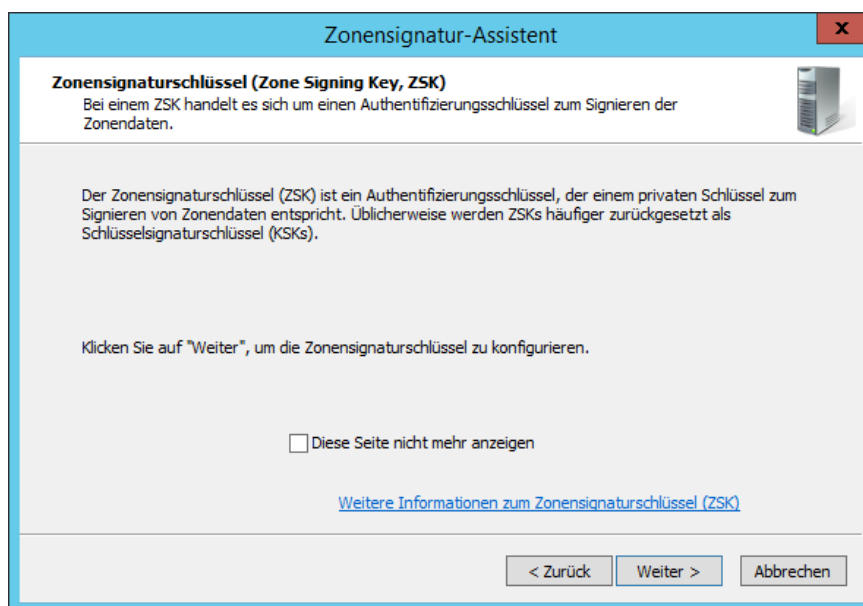
Hier zwei Webseiten die das Thema Schlüssel Rollover beschreiben

<https://securityblog.switch.ch/2013/02/05/algorithm-rollover/>

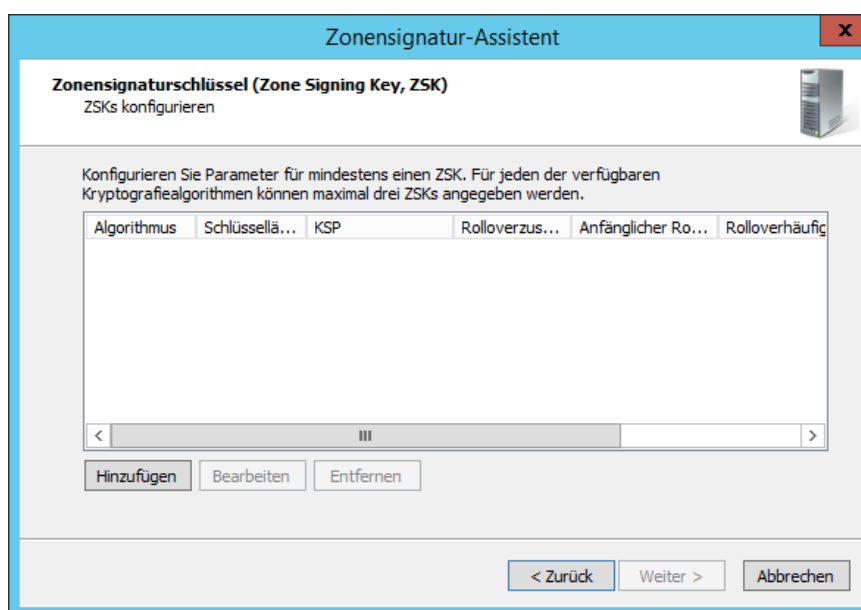
<http://www.heise.de/netze/meldung/DNSSEC-Verfahren-fuer-Schluesseltausch-in-der-Rootzone-festgelegt-3208629.html>



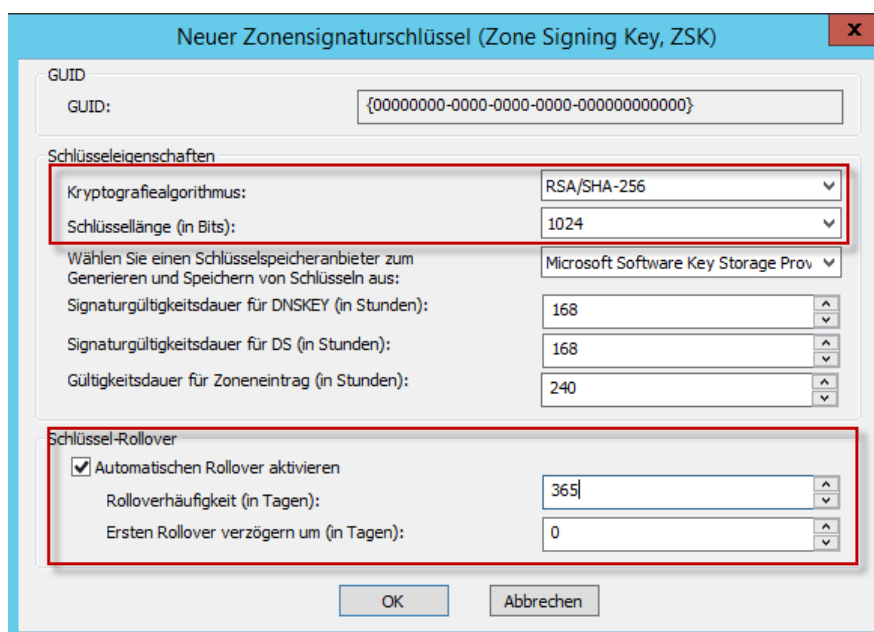
Nachdem der KSK Schlüssel erstellt wurde, klicken wir auf weiter um den Zone Signing Key zu erstellen.



Hier klicken wir ebenfalls auf Weiter, um den Zone Signing Key (ZSK) zu erstellen.



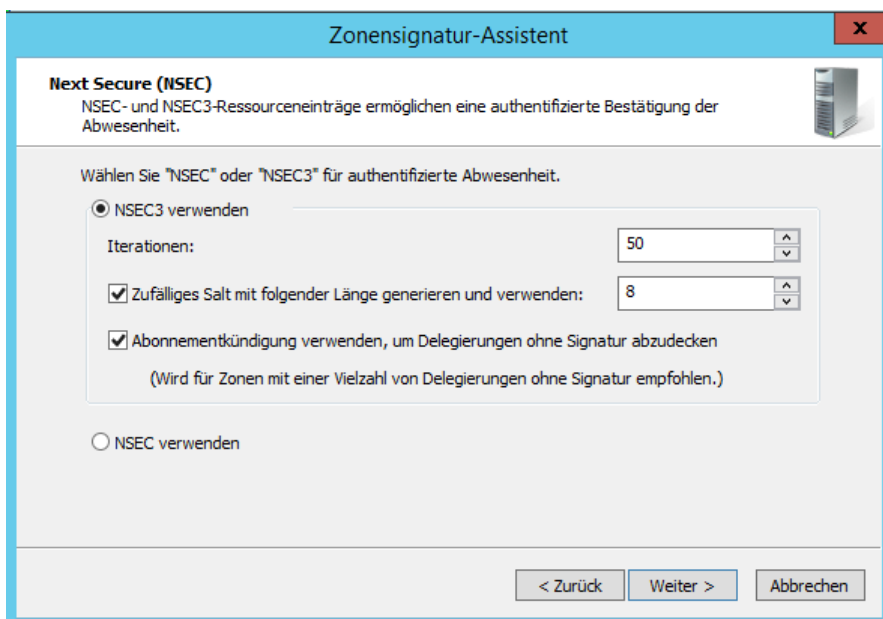
Wir klicken auf Hinzufügen.



In diesem Wizard wählen wir ebenfalls den Kryptografiealgorithmus **RSA/SHA 256** aus mit einer Schlüssellänge von 1024 Bit.

Der Schlüssel Rollover sollte einmal im Jahr stattfinden.

Die Schlüssellänge, die wir hier konfigurieren, beeinflusst das zu sendende UDP-Paket, das derzeit die Größe von 512 Byte nicht überschreiten darf. Umso größer also die Schlüssellänge, umso größer die Datenmenge, die das UDP-Paket senden muss.



Zonensignatur-Assistent

Next Secure (NSEC)
NSEC- und NSEC3-Ressourceneinträge ermöglichen eine authentifizierte Bestätigung der Abwesenheit.

Wählen Sie "NSEC" oder "NSEC3" für authentifizierte Abwesenheit.

☒ NSEC3 verwenden

Iterationen:

☒ Zufälliges Salt mit folgender Länge generieren und verwenden:

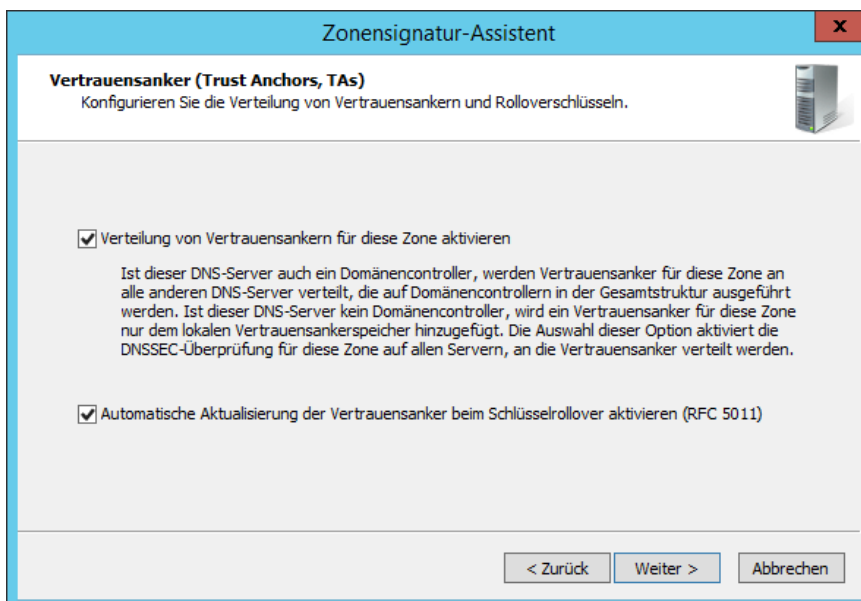
☒ Abonnementkündigung verwenden, um Delegierungen ohne Signatur abzudecken
(Wird für Zonen mit einer Vielzahl von Delegierungen ohne Signatur empfohlen.)

☐ NSEC verwenden

< Zurück Weiter > Abbrechen

NSEC wird benötigt, um nicht-vorhandene DNS Einträge zu beweisen.

https://de.wikipedia.org/wiki/NSEC_Resource_Record



Zonensignatur-Assistent

Vertrauensanker (Trust Anchors, TAs)
Konfigurieren Sie die Verteilung von Vertrauensankern und Rolloverschlüsseln.

☒ Verteilung von Vertrauensankern für diese Zone aktivieren

Ist dieser DNS-Server auch ein Domänencontroller, werden Vertrauensanker für diese Zone an alle anderen DNS-Server verteilt, die auf Domänencontrollern in der Gesamtstruktur ausgeführt werden. Ist dieser DNS-Server kein Domänencontroller, wird ein Vertrauensanker für diese Zone nur dem lokalen Vertrauensankerspeicher hinzugefügt. Die Auswahl dieser Option aktiviert die DNSSEC-Überprüfung für diese Zone auf allen Servern, an die Vertrauensanker verteilt werden.

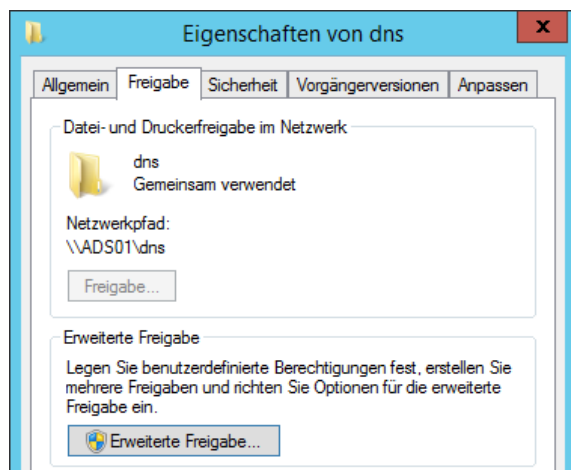
☒ Automatische Aktualisierung der Vertrauensanker beim Schlüsselrollover aktivieren (RFC 5011)

< Zurück Weiter > Abbrechen

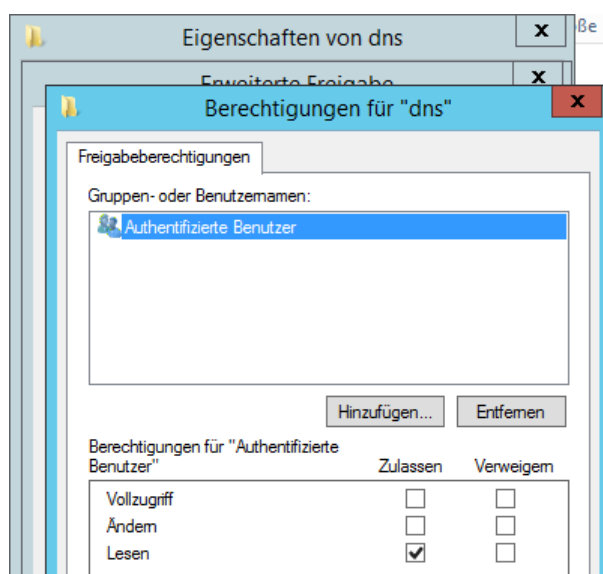
Um das Ganze abzuschließen, klicke ich den Wizard durch.

Vertrauensanker (Trustanchors) im Netzwerk verteilen

Um den Vertrauensanker im Netzwerk verteilen zu können, muss das Verzeichnis **C:\Windows\System32\dns** im Netzwerk freigegeben werden.

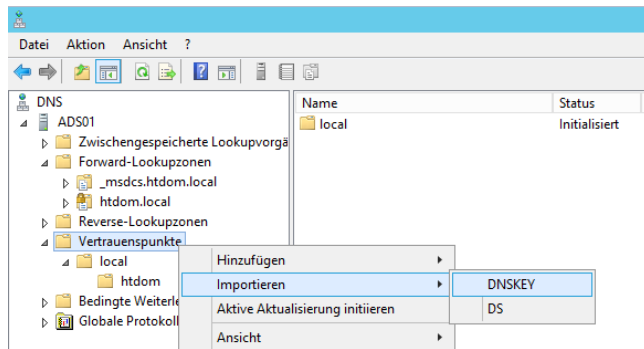


Die Freigabeberechtigungen ändere ich wie folgt, die Benutzergruppe **Jeder** wird gelöscht.

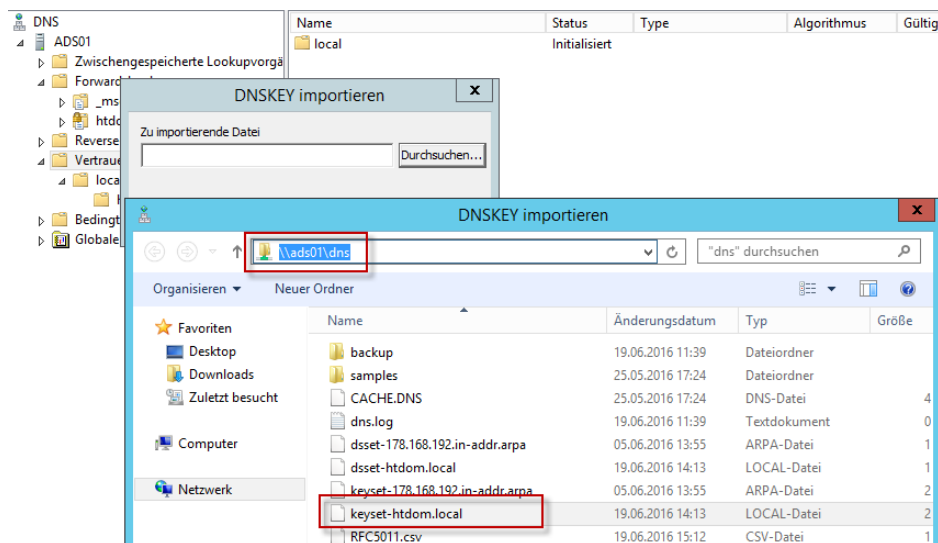


Und dafür füge ich die **Authentifizierten Benutzer** mit „Lesen“ Recht hinzu.

Danach öffne ich die DNS Management Konsole und navigiere zu den Punkt „**Vertrauenspunkte**“



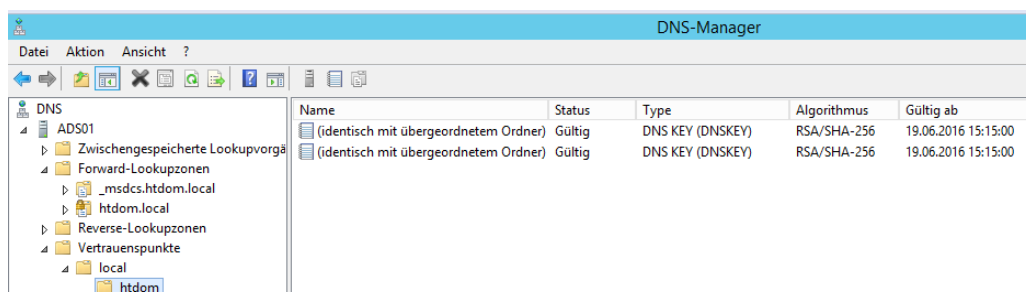
Dort Importiere ich einen neuen DNSKEY.



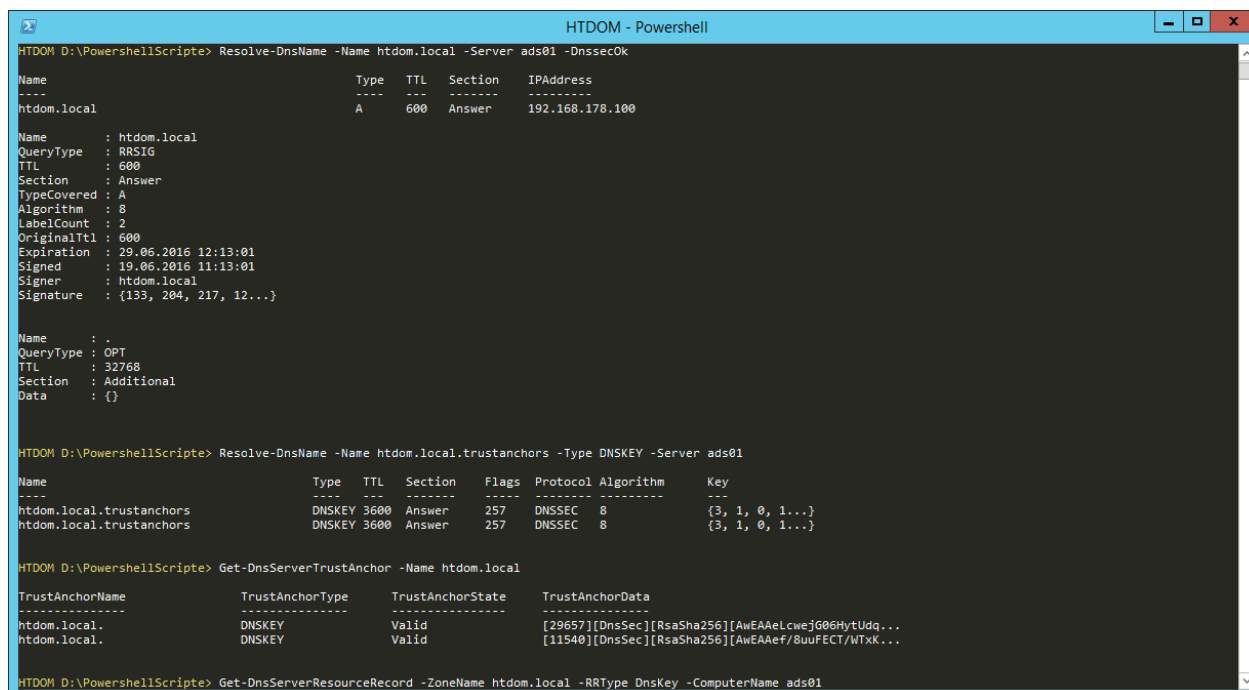
Wir klicken auf Durchsuchen, in der Adresszeile wählen wir den UNC Pfad zu der eben erstellen DNS Freigabe und dort wähle wir die passende **keyset** Datei aus.

[\\ads01\dns\keyset-htdom.local](#)

Nun sollte man in der Ordnerstruktur seine DNSKEYs finden.



Um das Ganze nun testen zu können, öffnen wir eine PowerShell Sitzung und können mit folgenden Befehlen den DNS Server auf DNSSEC überprüfen.



```
HTDOM D:\PowershellScripte> Resolve-DnsName -Name htdom.local -Server ads01 -DnssecOk

Name                                     Type  TTL  Section  IPAddress
----
htdom.local                             A      600  Answer    192.168.178.100

Name      : htdom.local
QueryType : RRSIG
TTL       : 600
Section   : Answer
TypeCovered : A
Algorithm : 8
LabelCount : 2
OriginalTtl : 600
Expiration : 29.06.2016 12:13:01
Signed    : 19.06.2016 11:13:01
Signer    : htdom.local
Signature  : {133, 204, 217, 12...}

Name      : .
QueryType : OPT
TTL       : 32768
Section   : Additional
Data      : {}

HTDOM D:\PowershellScripte> Resolve-DnsName -Name htdom.local.trustanchors -Type DNSKEY -Server ads01

Name                                     Type  TTL  Section  Flags  Protocol Algorithm  Key
----
htdom.local.trustanchors                DNSKEY 3600  Answer    257    DNSSEC    8             {3, 1, 0, 1...}
htdom.local.trustanchors                DNSKEY 3600  Answer    257    DNSSEC    8             {3, 1, 0, 1...}

HTDOM D:\PowershellScripte> Get-DnsServerTrustAnchor -Name htdom.local

TrustAnchorName      TrustAnchorType  TrustAnchorState  TrustAnchorData
-----
htdom.local.         DNSKEY           Valid             [29657][DnsSec][RsaSha256][AwEAAeLcweJG06HytUdq...
htdom.local.         DNSKEY           Valid             [11540][DnsSec][RsaSha256][AwEAAef/8uuFEct/WTxK...

HTDOM D:\PowershellScripte> Get-DnsServerResourceRecord -ZoneName htdom.local -RRType DnsKey -ComputerName ads01
```

- ↳ **Resolve-DnsName -Name htdom.local -Server ads01 -DnssecOk**
- ↳ **Resolve-DnsName -Name htdom.local.trustanchors -Type DNSKEY -Server ads01**
- ↳ **Get-DnsServerTrustAnchor -Name htdom.local**
- ↳ **Get-DnsServerResourceRecord -ZoneName htdom.local -RRType DnsKey -ComputerName ads01**
- ↳ **Get-DnsClientNrptPolicy**
- ↳ **Resolve-DnsName -Name htdom.local -Type SOA -Server ads01 -DnssecOk**
- ↳ **Resolve-DnsName -Name htdom.local -Type DNSKEY -Server ads01 -DnssecOk**

Mehr Sicherheit mit SocketPoolSize und CacheLockingPercent

Um noch ein bisschen mehr Sicherheit in die DNS Konfiguration zu bringen, verändern wir zwei Standardwerte in der DNS Einstellungen.

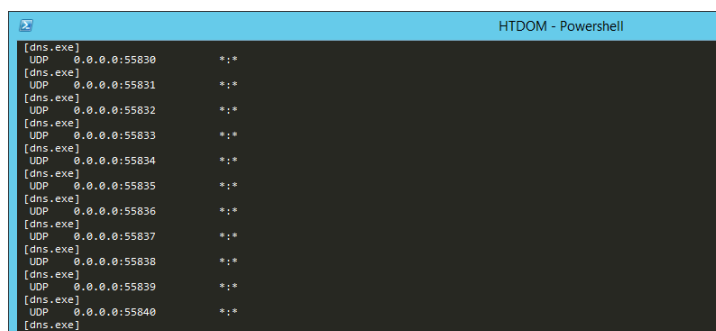
Nach der Grundinstallation eines DNS Servers ist die SocketPoolSize Standardmäßig auf einen Wert von 2500. Dies hat anscheinend in der Vergangenheit dazu geführt das DNS Spoofing Attacken stattfanden, weil Anwendungen und Dienste den gleichen Socketpool benutzten.

[Folgender Microsoft Artikel beschreibt das Sicherheitsproblem](#)

- ↳ `dnscmd /info /socketpoolsize (2500)`
- ↳ `dnscmd /config /socketpoolsize 4000`

Mit folgendem Befehl kann man sehen wie viele offene UDP Ports auf einen Server mit installierter DNS Server Rolle sind.

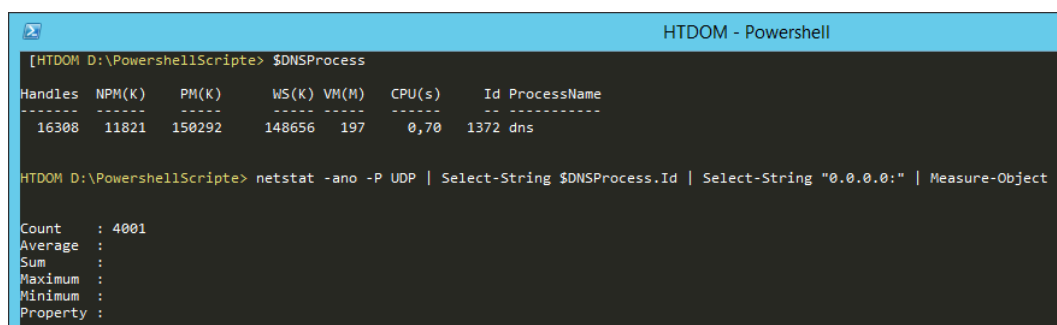
netstat -anb



```
[dns.exe]
UDP 0.0.0.0:55830 *: *
[dns.exe]
UDP 0.0.0.0:55831 *: *
[dns.exe]
UDP 0.0.0.0:55832 *: *
[dns.exe]
UDP 0.0.0.0:55833 *: *
[dns.exe]
UDP 0.0.0.0:55834 *: *
[dns.exe]
UDP 0.0.0.0:55835 *: *
[dns.exe]
UDP 0.0.0.0:55836 *: *
[dns.exe]
UDP 0.0.0.0:55837 *: *
[dns.exe]
UDP 0.0.0.0:55838 *: *
[dns.exe]
UDP 0.0.0.0:55839 *: *
[dns.exe]
UDP 0.0.0.0:55840 *: *
[dns.exe]
UDP 0.0.0.0:55841 *: *
```

Um auszurechnen um wie viel genau es sich handelt, kann man folgendes Statement absetzen.

- ↳ `$DNSProcess = Get-Process DNS`
- ↳ `netstat -ano -P UDP | Select-String $DNSProcess.Id | Select-String "0.0.0.0:" | Measure-Object`



```
[HTDOM D:\PowershellScripte> $DNSProcess

Handles NPM(K) PM(K) WS(K) VM(M) CPU(s) Id ProcessName
-----
16308 11821 150292 148656 197 0,70 1372 dns

[HTDOM D:\PowershellScripte> netstat -ano -P UDP | Select-String $DNSProcess.Id | Select-String "0.0.0.0:" | Measure-Object

Count : 4001
Average :
Sum :
Maximum :
Minimum :
Property :
```

Quelle: <http://www.foxguardsolutions.com/resources/details/a-closer-look-at-windows-dns-ports>

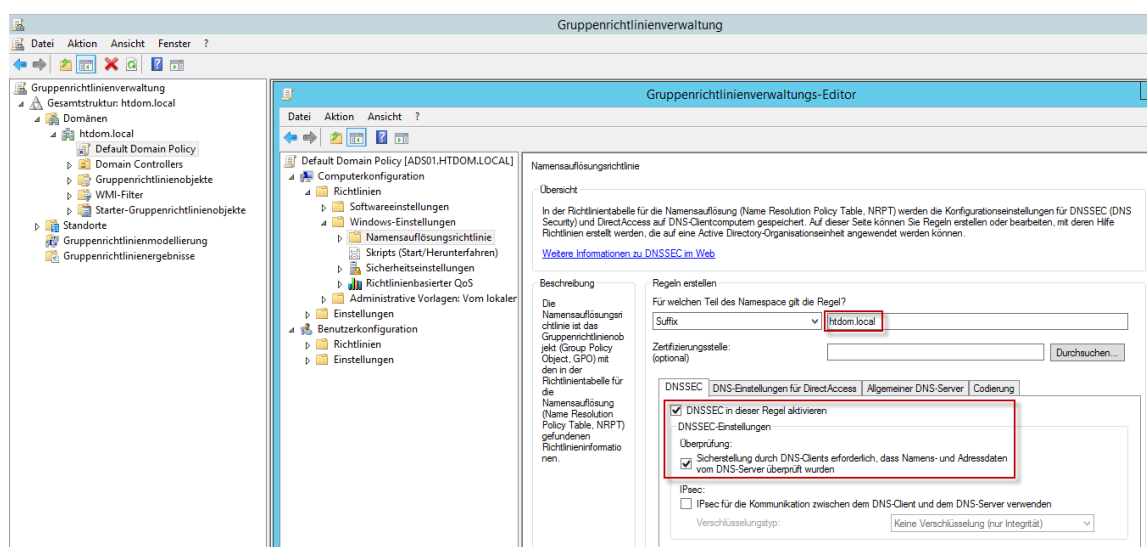
Die zweite Einstellung ist der CacheLockingPercent, diese überschreibt einen gecachden Eintrag wenn die TTL (Time to Life) zu 100% erreicht ist, Best Practice sagt, wir sollen einen Wert von 90 % setzen.

- ↳ `dnscmd /info /cachelockingpercent (100)`
- ↳ `dnscmd /config /cachelockingpercent 90`

oder

- ↳ `(Get-DnsServerCache).LockingPercent`
- ↳ `Set-DnsServerCache -LockingPercent 90`

Gruppenrichtlinie für DNSSEC aktivieren



Mit der Richtlinie „**Name Resolution Policy Table, NRPT**“ wird die Namensauflösung mit DNSSEC Überprüfung erforderlich gemacht.

Entweder wir erstellen eine komplett neue Gruppenrichtlinie oder ändern die **Default Domain Policy**,

Da es hier eine Testdomain ist ändere ich die **Default Domain Policy**, in einer Live Umgebung würde ich hier eine eigene Server Policy anlegen und mit der Domain verknüpfen.

Auch hier bitte aufpassen beim Konfigurieren der Einstellung, hier muss ganz nach unten gescrollt werden, damit die Einstellung auch angewandt werden.

Aktualisieren
Erstellen
Löschen

Erweiterte globale Richtlinieneinstellungen

Richtlinientabelle für die Namensauflösung

Namespace	Zerti...	DNSSE...	DNSSEC ...	DNSSEC (...)	DirectAc...	DirectAc...	DirectAc...	DirectAc...	DNS-Se...	Codierung
htdom.local		Ja	Nein							

Regel löschen
Regel bearbeiten

Anwenden
Abbrechen

Gruppenrichtlinienverwaltung

- ▲ Gesamtstruktur: htdom.local
- ▲ Domänen
 - htdom.local
 - Default Domain Policy
 - Domain Controllers
 - Gruppenrichtlinienobjekte
 - WMI-Filter
 - Starter-Gruppenrichtlinienobjekte
 - Standorte
 - Gruppenrichtlinienmodellierung
 - Gruppenrichtlinienergebnisse

Default Domain Policy

Bereich Details Einstellungen Delegierung

Lokale Richtlinien/Sicherheitsoptionen

Richtlinie	Einstellung
Netzwerksicherheit	
Netzwerksicherheit: Anmeldung nach Ablauf der Anmeldezeit erzwingen	Deaktiviert
Netzwerksicherheit: Keine LAN Manager-Hashwerte für nächste Kennwortänderung speichern	Aktiviert
Netzwerkzugriff	
Netzwerkzugriff: Anonyme SID-/Namensübersetzung zulassen	Deaktiviert

Richtlinien öffentlicher Schlüssel/Verschlüsselndes Dateisystem

Ausgestellt für	Ausgestellt von	Ablaufdatum	Beabsichtigte Zwecke
Administrator	Administrator	01.05.2116 17:33:30	Dateiwiederherstellung

Starten Sie den lokalen Gruppenrichtlinienobjekt-Editor, wenn Sie weitere Informationen zu bestimmten Einstellungen wünschen.

Namensauflösungsrichtlinie

Regelname	Wert
Namespace	htdom.local
Richtlinie	htdom.local
Namespace	htdom.local
Zertifizierungsstelle	Leer
Konfiguration	DNSSEC
DNSSEC (Überprüfung)	Ja
DNSSEC (Psec)	Nein
DNSSEC (Psec-Verschlüsselung)	Keine Verschlüsselung (nur Integrität)
DirectAccess (Psec)	Nicht konfiguriert
DirectAccess (Psec-Verschlüsselung)	Nicht konfiguriert
DirectAccess (Proxyeinstellungen)	Nicht konfiguriert
DirectAccess (Webproxy)	Nicht konfiguriert
DirectAccess (DNS-Server)	Nicht konfiguriert
Allgemeine DNS-Server	Nicht konfiguriert
Codierung	Nicht konfiguriert
Version	1

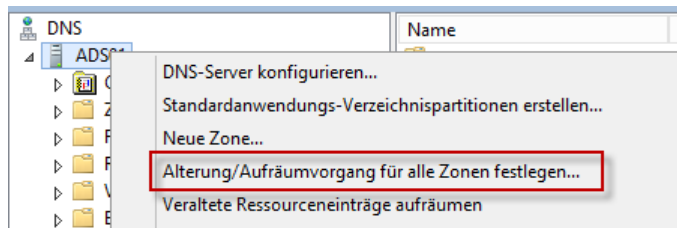
Mit **gpupdate** oder **gpupdate /force** werden die Einstellungen übernommen

Alterung der DNS Serverzonen einstellen

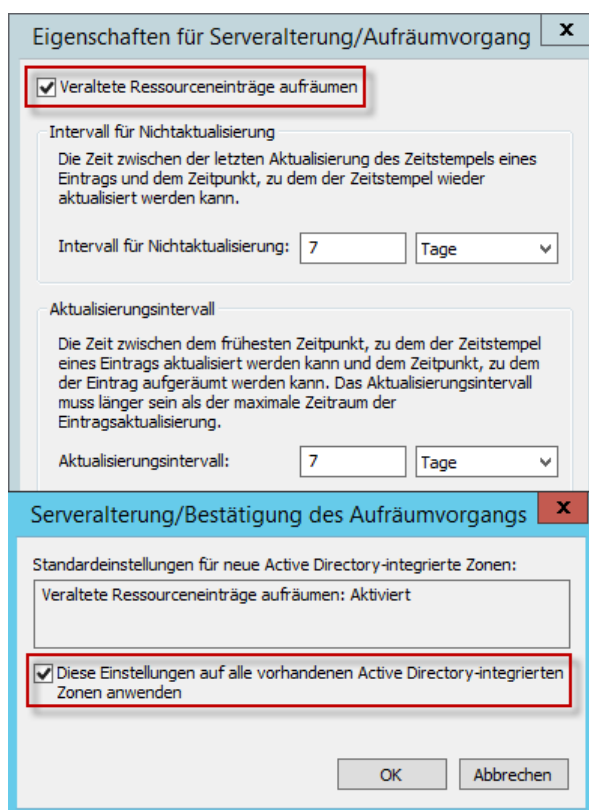
Um den DNS-Server automatisiert zu bereinigen, richtet man die sogenannte Alterung auf Server und Zonen ein. Diese sorgt dafür dass DNS-Einträge mit einem Zeitstempel nach einer gewissen Zeit automatisch gelöscht werden.

Gerade im Zeitalter der Mobilien Geräte, werden immer häufiger Einträge im DNS Server gemacht und diese werden nicht mehr automatisch gelöscht.

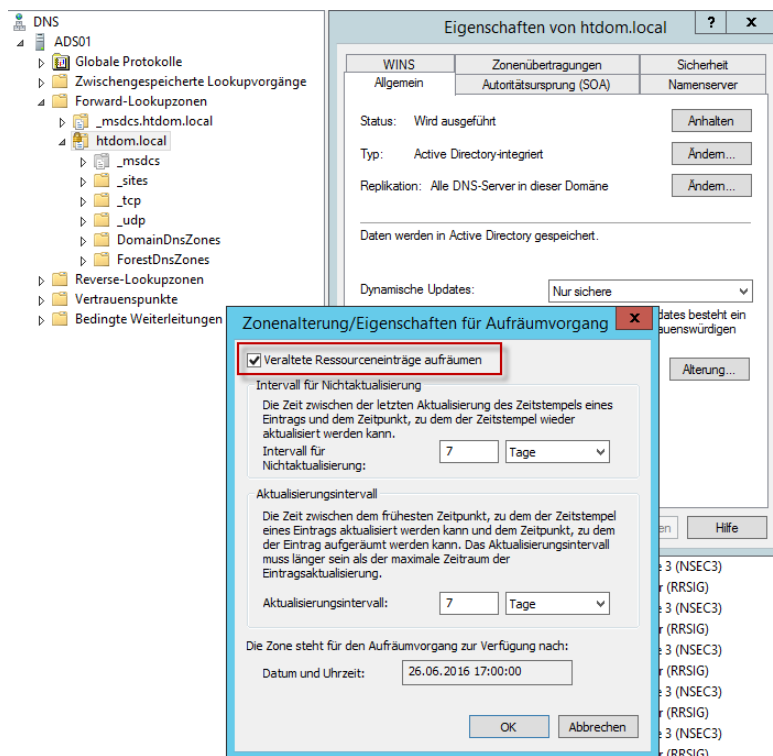
Die Einstellung kann man manuell oder per PowerShell durchführen.



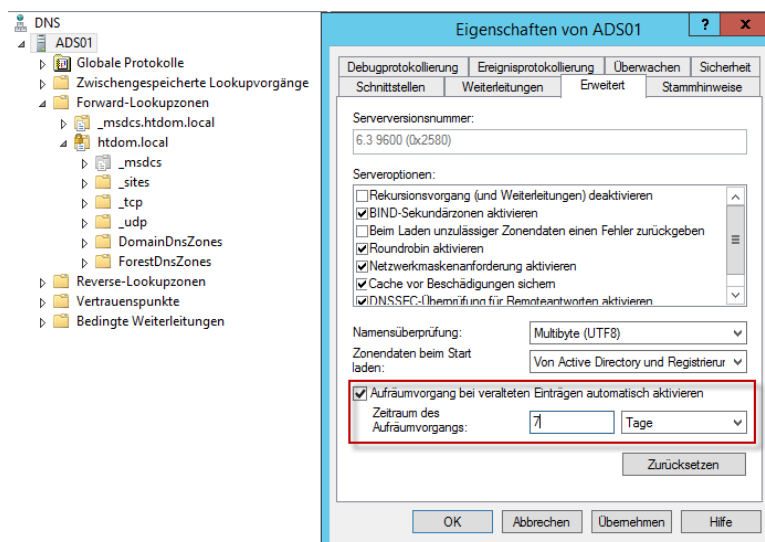
Als erstes klicke man auf den Servernamen und wählt im Kontext Menü „**Alterung/Aufräumvorgang für alle Zonen festlegen**“



Wir setzen das Häkchen und klicken auf OK, danach müssen wir noch das Häkchen für alle vorhandenen Active Directory Integrierten Zonen aktivieren und ebenfalls mit OK bestätigen.



Wenn man sich nun die Eigenschaften der Forward und Reverse Lookupzonen ansieht, wurden hier die Alterungen der Zone aktiviert. (**Bitte kontrollieren**)



Die letzte Einstellung die gesetzt werden muss, ist die Einstellung das der Server nach einer gewissen Zeitspanne die Einträge auch löscht.

Diese Bereinigungsaktion kann man im Ereignisprotokoll (2501/2502) überprüfen.

- ↳ Event ID 2501 = Welche Einträge wurden erfolgreich gelöscht.
- ↳ Event ID 2502 = Es wurde nichts bereinigt

Mit Powershell und dnscmd kann man die Alterung ebenfalls einrichten, diese Befehle sind auf einem Server Core sehr nützlich.

Was unbedingt zu beachten ist, besonders wenn mehrerer DNS Server im Einsatz sind, das nur ein einziger DNS Server die Bereinigung vornimmt, diese Einstellung kann man entweder über PowerShell **-ScavengeServers 192.168.178.100** oder **dnscmd Servername /ZoneResetScavengeServers Zone IP-Adresse** setzen.

Server Aging und Scavenging für alle Zonen einrichten

- ↳ **Set-DnsServerScavenging -ApplyOnAllZones -ScavengingState \$true -RefreshInterval 7.00:00:00 -NoRefreshInterval 7.00:00:00**
- ↳ **dnscmd /config /scavenginginterval 168** (168 / 24 Stunden = 7 Tage)

Scavenging Server für die Zonen einrichten

- ↳ **Set-DnsServerZoneAging -ComputerName \$env:COMPUTERNAME -Name htdom.local -RefreshInterval 7.00:00:00 -NoRefreshInterval 7.00:00:00 -ScavengeServers 192.168.178.100 -PassThru -Verbose**
- ↳ **Set-DnsServerZoneAging -ComputerName \$env:COMPUTERNAME -Name 178.168.192.in-addr.arpa -RefreshInterval 5.00:00:00 -NoRefreshInterval 5.00:00:00 -ScavengeServers 192.168.178.100 -PassThru -Verbose**
- ↳ **Restart-Service DNS**

Was bedeuten denn die angegebenen Werte?

"Intervall für Nichtaktualisierung": Diese Zeitspanne gibt an, wie lange ein Eintrag vom Client nicht aktualisiert werden kann. Wenn ein Client seinen A-Record in der DNS-Zone einträgt, dann wird dieser Eintrag mit einem Zeitstempel versehen. Dieser Zeitstempel kann bis nach Ablauf dieser Frist nicht aktualisiert werden. Wenn aber ein Client z.B. eine andere IP-Adresse erhalten hat, dann kann er diese natürlich im DNS ändern. Diese Einstellung wirkt sich also nur auf den Aktualisierungsintervall des Zeitstempels aus. Den Zeitstempel der DNS-Records kann man in der Standardansicht nicht einsehen. Wenn man den Zeitstempel eines DNS Eintrages einsehen möchte, so muss man vorher in der DNS-Konsole unter "Ansicht" die "Erweiterte Ansicht" aktivieren. Danach kann man durch Doppelklick auf einen Eintrag den entsprechenden Zeitstempel einsehen.

"Aktualisierungsintervall": Innerhalb dieses Intervalls hat der Client die Möglichkeit, den Zeitstempel zu aktualisieren. Wenn dieses Intervall abgelaufen ist, dann wird der Eintrag als veraltet markiert und letztendlich durch den Aufräumvorgang aus der DNS-Zone entfernt.

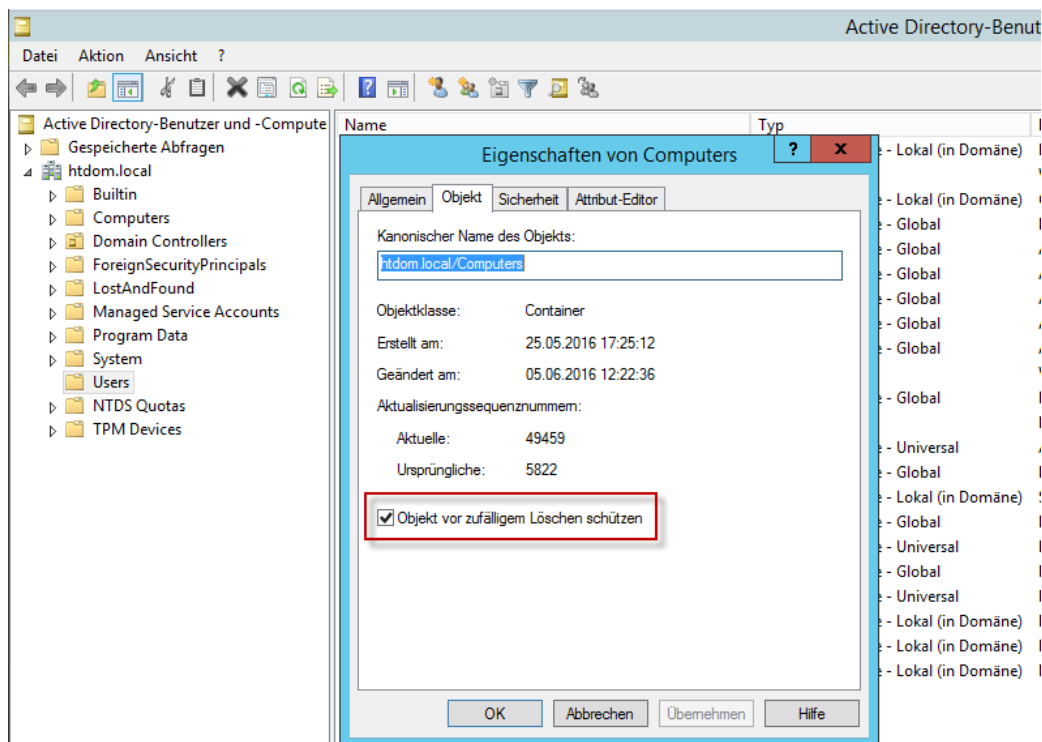
"Aufräumvorgang bei veralteten Einträgen": Dieser Wert gibt an, wie oft der Server das Aufräumen veralteter Einträge vornimmt.

Active Directory Objekte vor versehentlichen löschen schützen.

Der Best-Practice Analyser von Active Directory, klagt nach der Grundinstallation an das alle Objekte vor dem versehentlichen löschen geschützt werden müssen.

Dies realisiert man mit folgenden PowerShell Befehlen.

- ↳ **Get-ADOrganizationalUnit -Filter * -Properties ProtectedFromAccidentalDeletion | where {\$_.ProtectedFromAccidentalDeletion -eq \$false} | Set-ADOrganizationalUnit -ProtectedFromAccidentalDeletion \$true**
- ↳ **Get-ADObject -Filter * -Properties ProtectedFromAccidentalDeletion | where {\$_.ProtectedFromAccidentalDeletion -eq \$false} | Set-ADObject -ProtectedFromAccidentalDeletion \$true -ErrorAction SilentlyContinue**



Zeitserver für den PDC Emulator konfigurieren

Um auf dem ersten Domaincontroller (PDC Emulator) die Zeitsynchronisation mit einer externen Quelle zu konfigurieren, sollte man sich im Internet kurz schlau machen welche Zeitquellen sind den so verfügbar.

Da ich weiß das in Braunschweig mehrere Atomuhren stehen nutze ich auch die Quelle als externer Zeitgeber.

<http://www.zeitserver.de/deutschland/deutsche-ntp-zeitserver/>

<http://www.zeitserver.de/deutschland/ptb-zeitserver-in-braunschweig/>

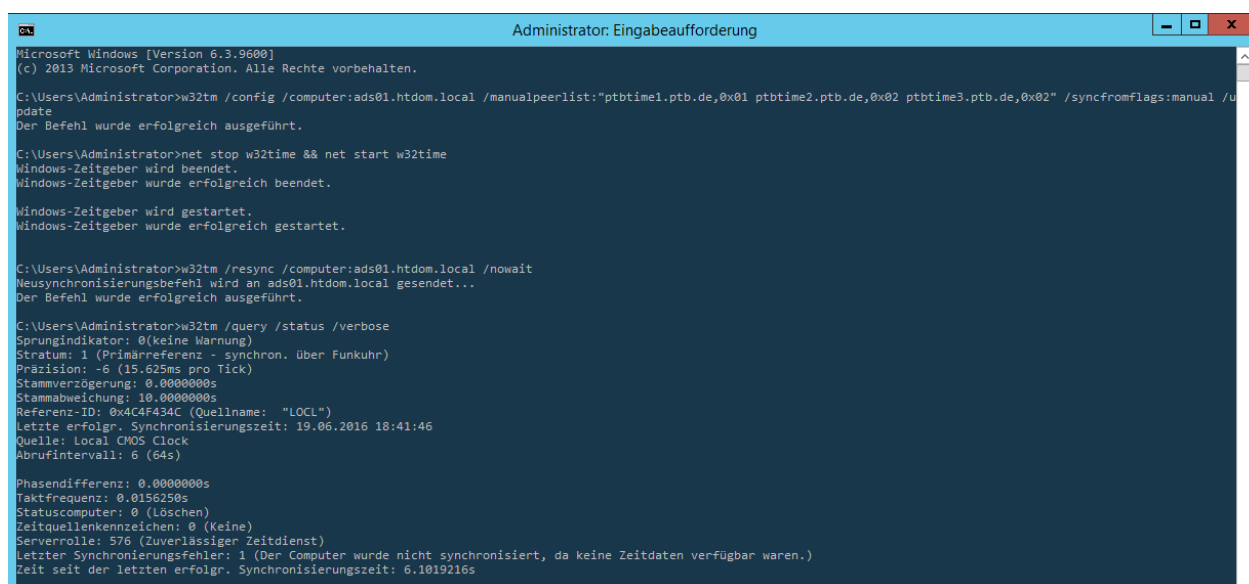
↳ **w32tm /config /computer:ads01.htdom.local /manualpeerlist:"ptbtime1.ptb.de,0x01 ptbtime2.ptb.de,0x02 ptbtime3.ptb.de,0x02" /syncfromflags:manual /update**

- **0x01 SpecialInterval**
- **0x02 UseAsFallbackOnly**
- **0x04 SymmetricActive**
- **0x08 Client**

↳ **net stop w32time && net start w32time**

↳ **w32tm /resync /computer:ads01.htdom.local /nowait**

↳ **w32tm /query /status /verbose**



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

C:\Users\Administrator>w32tm /config /computer:ads01.htdom.local /manualpeerlist:"ptbtime1.ptb.de,0x01 ptbtime2.ptb.de,0x02 ptbtime3.ptb.de,0x02" /syncfromflags:manual /update
Der Befehl wurde erfolgreich ausgeführt.

C:\Users\Administrator>net stop w32time && net start w32time
Windows-Zeitgeber wird beendet.
Windows-Zeitgeber wurde erfolgreich beendet.

Windows-Zeitgeber wird gestartet.
Windows-Zeitgeber wurde erfolgreich gestartet.

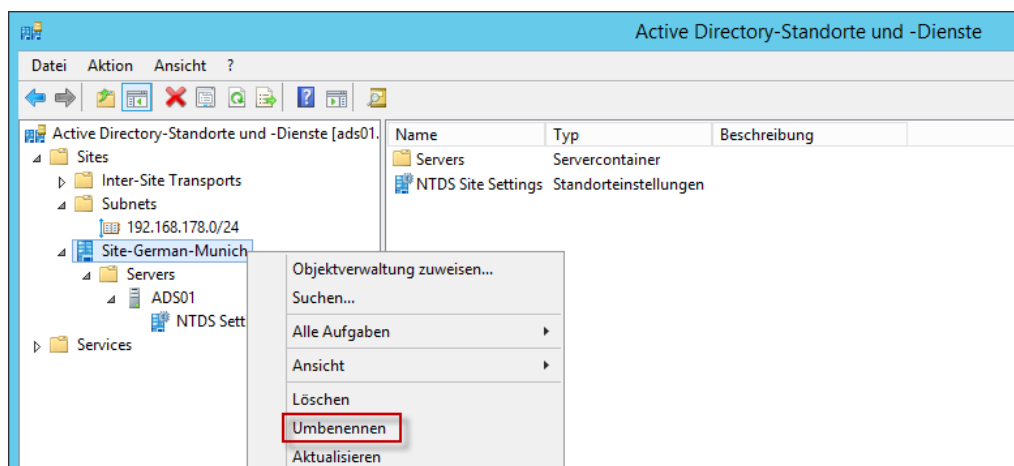
C:\Users\Administrator>w32tm /resync /computer:ads01.htdom.local /nowait
Neusynchronisierungsbefehl wird an ads01.htdom.local gesendet...
Der Befehl wurde erfolgreich ausgeführt.

C:\Users\Administrator>w32tm /query /status /verbose
Sprungindikator: 0(keine Warnung)
Stratum: 1 (Primärreferenz - synchron. über Funkuhr)
Präzision: -6 (15.625ms pro Tick)
Stammverzögerung: 0.0000000s
Stammabweichung: 10.0000000s
Referenz-ID: 0x404F434C (Quellname: "LOCL")
Letzte erfolgr. Synchronisierungszeit: 19.06.2016 18:41:46
Quelle: Local CMOS Clock
Abrufintervall: 6 (64s)

Phasendifferenz: 0.0000000s
Taktfrequenz: 0.0156250s
Statuscomputer: 0 (Löschen)
Zeitquellenkennzeichen: 0 (Keine)
Serverrolle: 576 (Zuverlässiger Zeitdienst)
Letzter Synchronisierungsfehler: 1 (Der Computer wurde nicht synchronisiert, da keine Zeitdaten verfügbar waren.)
Zeit seit der letzten erfolgr. Synchronisierungszeit: 6.1019216s
```

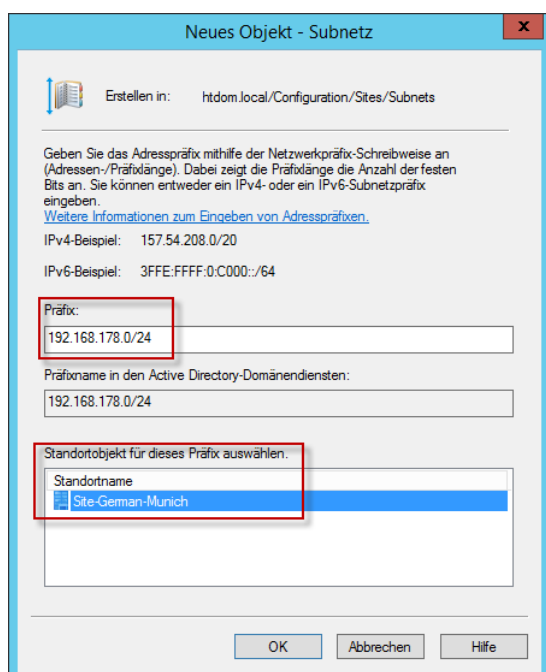
Active Directory – Standort und Dienste

Um die Grundkonfiguration abzuschließen lege ich für meinen ersten Domaincontroller noch ein Subnetz an und benenne die Default Site um. Hierzu öffne ich die MMC **Active-Directory-Standorte und Dienste**.



Hierzu klickt man den Eintrag Default-Site mit der rechten Maustaste an und wählt im Kontextmenü umbenennen.

Im nächsten Schritt legen wir für die Site-german-Munich ein neues Subnetz an, dazu klickt man auf den Eintrag Subnets mit der rechten Maustaste und wählt Neues Subnetz.



Hier vergibt man sein Subnetz in dem sich der Domaincontroller befindet und wählt die passende Site aus.

Was schreibt Microsoft zu dem Thema Standorte:

Ein Standort ist ein Bereich des Netzwerkes, der über schnelle Netzwerkverbindungen mit hoher Bandbreite verfügt. Hierbei handelt es sich per Definition um eine Sammlung von Computern mit schnellen Verbindungen auf Basis von IP-Subnetzen (Internet-Protokoll). Da Standorte steuern, wie die Replikation erfolgt, haben Änderungen, die mit dem Snap-In Active Directory-Standorte und -Dienste gemacht werden, Auswirkungen darauf, wie effizient die Domänencontroller (DCs) innerhalb einer Domäne (aber über große Entfernung) miteinander kommunizieren können.

Ein Standort unterscheidet sich konzeptionell von einer Domäne unter Windows Server 2008, da ein Standort mehrere Domänen und eine Domäne mehrere Standorte umfassen kann. Standorte sind nicht Teil des Domänennamespaces. Standorte steuern die Replikation von Domänenendaten und helfen, die Nähe von Ressourcen zu ermitteln. So wählt beispielsweise eine Arbeitsstation zum Zweck der Authentifizierung einen Domänencontroller (DC) innerhalb des eigenen Standortes.

Zur Sicherstellung, dass die Replikation seitens Active Directory ordnungsgemäß durchgeführt werden kann, wird auf allen DCs ein Konsistenzprüfungsdienst mit Namen Knowledge Consistency Checker (KCC) ausgeführt, der automatisch Verbindungen zwischen einzelnen Computern am gleichen Standort herstellt. Diese Verbindungen werden als Active Directory-Verbindungsobjekte bezeichnet. Der Administrator kann weitere Verbindungsobjekte erstellen oder Verbindungsobjekte entfernen. Wenn die Replikation jedoch an irgendeinem Punkt innerhalb eines Standortes unmöglich wird oder wenn sich ein einzelner Ausfallpunkt ergibt, schaltet sich die Konsistenzprüfung ein und erstellt so viele neue Verbindungsobjekte wie nötig sind, um die Active Directory-Replikation wiederaufzunehmen.....

http://openbook.rheinwerk-verlag.de/windows_server_2012r2/08_002.html#dotpb48200d0-6b44-40ce-af46-dcdd04802812

So das war es erst mal wieder von mir, der Domaincontroller ist nun sauber eingerichtet worden und steht für den zweiten Teil der Konfiguration Serie bereit.

Viele Grüße

Helmut Thurnhofer