

WSUS – Cleanup automatisieren

Hallo zusammen und ein frohes neues Jahr :-)

Mir ist es schon letztes Jahr oft aufgefallen, dass beim Ausführen vom Server Cleanup Wizard innerhalb des Windows Server Update Service (WSUS) die Konsole in ein Timeout läuft. Ich hatte dies immer wieder auf die doch schon in die Jahre gekommene MMC Konsole geschoben. Dieses Jahr wollte ich endlich einmal die Serverbereinigung mittels Skript automatisieren und hatte da auch die Hoffnung, dass die Problematik nur ein GUI Problem sei, welches man über Skripting umgehen könnte. Leider war dies nicht der Fall und die Lösung kommt weiter unten...

Ich will es nicht verheimlichen, dass ich eigentlich alles zusammen kopiert habe. Jedoch will ich mit diesem Blog mehrere Beiträge zusammenfassen um den kompletten Weg einer automatisierten WSUS Bereinigung aufzuzeigen. Natürlich sind am Schluss alle Quellen genannt ;-)

Skript

Als erstes habe ich eigentlich nur nach den notwendigen PowerShell Befehlen gesucht und ich wurde dann auf wsus.de mit einem kompletten Skript fündig.

[WSUS-Cleanup-Skript](#) (Skript als Textdatei) / Link zur Quelle: <http://www.wsus.de/serverbereinigung2>

Das Skript müsst ihr nur noch mit euern Parametern (Servername, Maileinstellungen, Bereinigungsparametern, etc.) anpassen und schon könnt ihr es als Administrator ausführen (UAC lässt grüssen).

Geplanter Task

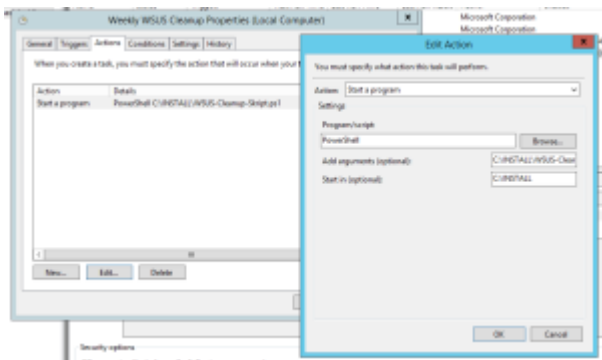
Als nächstes muss ein geplanter Task erstellt werden. Weil es mein erster mit PowerShell war, musste ich mich auch hier

schlau machen.

Fündig wurde ich dann hier: http://www.metalogix.com/help/Content%20Matrix%20Console/SharePoint%20Edition/002_HowTo/004_SharePointActions/012_SchedulingPowerShell.htm

Zusammenfassend sind beim geplanten Task folgende Punkte zu beachten:

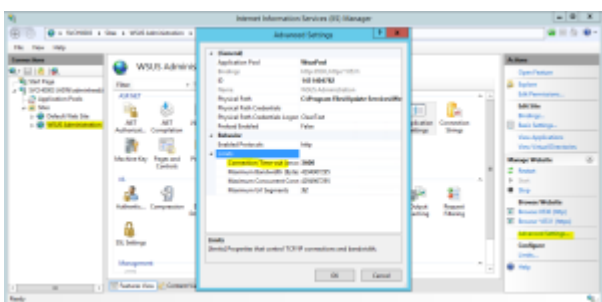
- ausführen ob Benutzer angemeldet ist oder nicht
- Benutzer des Tasks muss lokaler Administrator sein (Run as Admin)
- Ausgeführt wird PowerShell mit dem Skriptpfad als Argument



Timeout Problem

Bei einer grossen Menge an Updates wird der WSUS in ein Timeout laufen:

Um dem Abhilfe zu schaffen müsst ihr dieses Timeout im IIS anpassen:



IIS > Sites > WSUS Administration / Advanced Settings... / Limits > Connection Time-out (seconds)

Ich habe dieses bei meinem Server nun von 180 auf 3600

Sekunden erhöht und seither läuft die Bereinigung ohne Probleme.

Viel Erfolg beim Einrichten :-)

Quellen:

<http://www.wsus.de>

<http://www.metalogix.com>

<http://social.technet.microsoft.com>

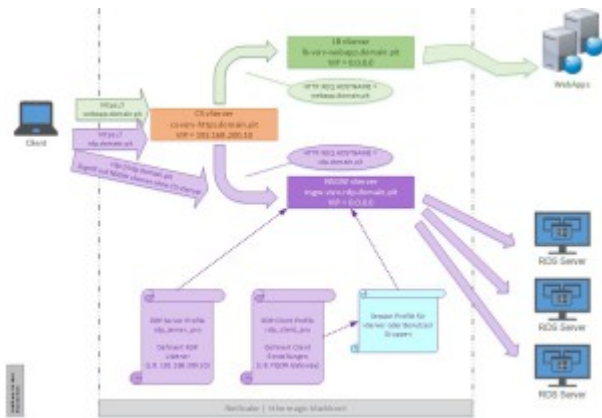
RDP Proxy mit NetScaler 11.x einrichten

Hallo zusammen

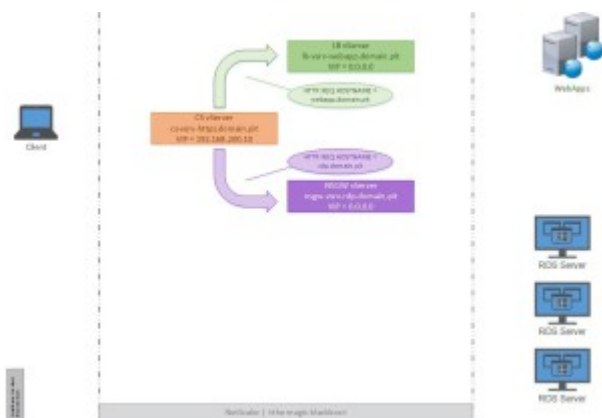
Eine der Neuerungen mit NetScaler 11 war die Einführung des RDP Proxy. Ich habe mich im Rahmen einer Kursvorbereitung ein wenig mit dem Feature auseinandersetzen dürfen/müssen. Nunja was soll ich sagen... es ist eigentlich gar keine Hexerei.

Vorraussetzungen: NetScaler Enterprise (RDP Proxy ist Bestandteil vom Unified Gateway) sowie NetScaler Gateway Universal Licenses

Wie ist funktioniert der RDP Proxy nun? Die komplette Grafik sieht nun so aus:

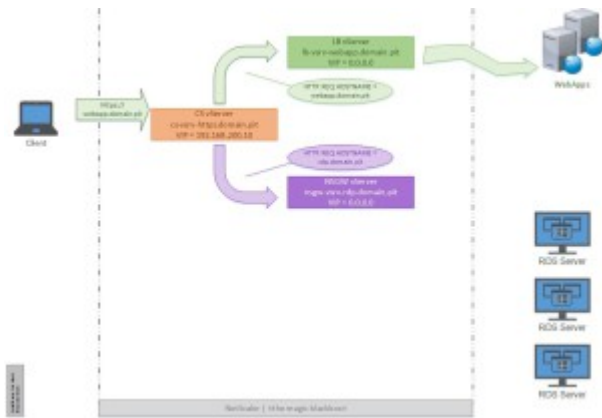


Die Skizze ist mit einer Unified Gateway Installation gezeichnet, wobei der NetScaler Gateway (NSGW) vServer ohne IP (0.0.0.0) hinter einem Content Switching (CS) vServer mit der entsprechenden VIP (z.B. 192.168.200.10) steht. In einer klassischen Konfiguration kann man den CS vServer wegdenken und die VIP ist auf dem NSGW vServer konfiguriert.

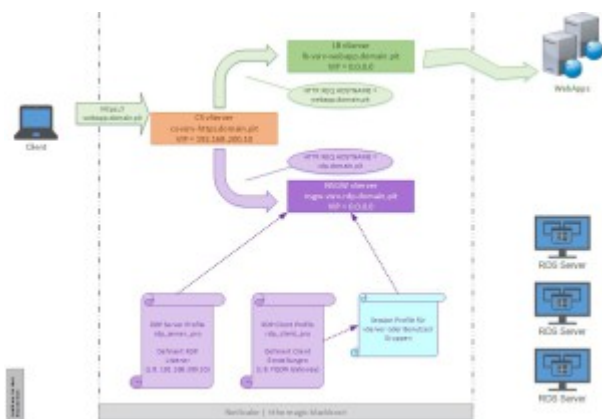


Klassische CS Konfiguration:

- CS vServer hört auf die VIP
- Anfragen auf den Hostname webapps.domain.pit werden auf den Loadbalancing (LB) vServer der Webapplikationen geleitet
- Anfragen auf den Hostname rdp.domain.pit werden auf den NSGW vServer geleitet



Anfrage der Webapplikationen verlaufen ohne spezielle Konfigurationen direkt auf die Webserver der Applikationen.



Mittels RDP Server Profilen (z.B. rdp_server_pro) werden die Listener-Einstellungen wie FQDN unseres Gateway, Port, etc. konfiguriert.

Dieses „RDP Server Profile“ muss an den NSGW vServer gebunden werden, damit dieser neu auch auf Anfragen auf den konfigurierten Port reagiert.

Der Zugriff von extern muss nun nebst dem bekannten 443 Port auch den Port für die RDP Verbindung (Standard: 3389) gewährleistet sein.

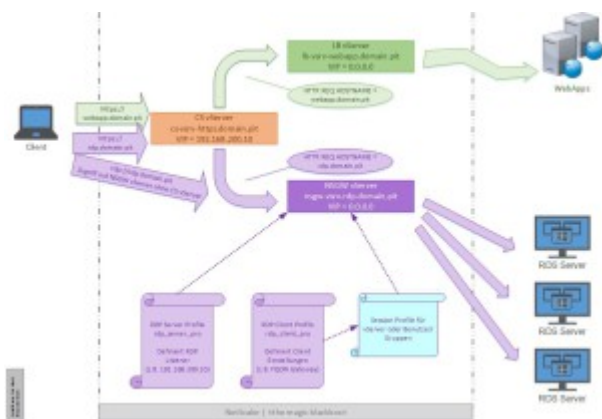
Zugriffe werden nun auf dem NSGW vServer mit der entsprechenden VIP terminiert.

Update 04.05.16: nach neusten Erkenntnissen kann auch der Port 443 für die RDP Verbindung genutzt werden. Dabei muss jedoch Stand heute eine weitere IP verwendet werden. Gemäss Aussage von Citrix soll „Port Sharing“ in einer der nächsten Versionen funktionieren.

Update 08.07.16: Mit der neuen Version 11.1 funktioniert nun

das „Port Sharing“ und man benötigt für einen RDP Proxy nur noch eine IP und kann alle Zugriffe via Port 443 konfigurieren (analog ICA Proxy).

Im „RDP Client Profile“ sind sämtliche für den Client relevanten Einstellungen hinterlegt wie z.B. die FQDN unseres Gateway, welche Mappings wir erlauben, etc. Dieses Profile muss an eine Session Policy gebunden sein, welche wiederum entweder direkt dem NSGW vServer oder einer Gruppe/einem Benutzer zugeordnet ist.



Der RDP Client greift bei einem Aufruf der Verbindung direkt auf den NSGW vServer zu, dieser terminiert die Frontend RDP Verbindung und baut eine entsprechende RDP Verbindung zum gewünschten Backend Computer auf. Et voilà, so funktioniert... vereinfacht gesagt ;-)

Wie wird dies nun konfiguriert? Hier die Schritt-für-Schritt Anleitung in Form von CLI Kommandos (im GUI entsprechend abbilden):

Das RDP Proxy Feature muss separat aktiviert werden, falls nicht im Basis-Setup bereits geschehen:

```
enable feature RDPProxy
```

Nun müssen das RDP Client und Server Profile erstellt werden wobei der SharedKey bei beiden identisch sein müssen. Der Parameter `-rdpFileName` definiert nur, wie die Datei heisst, welche zum Client übermittelt werden soll:

```
add rdp clientprofile rdp_client_pro -rdpFileName pit.rdp -
rdpHost rdp.domain.pit -psk Password1
add rdp serverprofile rdp_server_pro -rdpIP 192.168.200.10 -
psk Password1
```

Nun muss das RDP Server Profile dem definierten NSGW vServer angebunden werden:

```
set vpn vserver nsgw-vsrv-rdp.domain.pit -rdpServerProfileName
rdp_server_pro
```

Als nächstes benötigen wir entsprechende Session Policies in welchen wir den Zugriff erlauben, Clientless auf ALLOW setzen und das entsprechende RDP Client Profile zuweisen. Falls bereits Policies vorhanden sein sollten, können diese ggf. einfach angepasst werden:

```
add vpn sessionAction rdp_prof -defaultAuthorizationAction
ALLOW -clientlessVpnMode ON -rdpClientProfileName
rdp_client_pro
add vpn sessionPolicy rdp_pol ns_true rdp_prof
```

Diese Policies werden nun dem NSGW vServer angebunden:

```
bind vpn vserver nsgw-vsrv-rdp.domain.pit -policy rdp_pol -
priority 100
```

Nach erfolgreicher Anmeldung wählt man den Clientless Access, sofern man nicht automatisch dahin geleitet wurde:



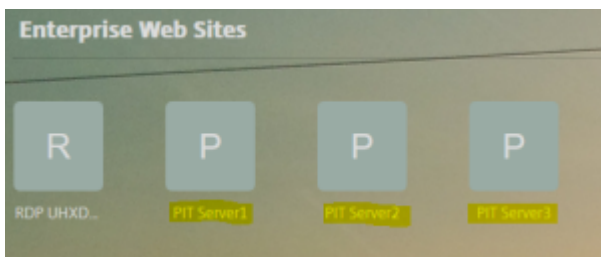
Mit der Eingabe `https://FQDN-des-Gateways/rdpproxy/Ziel` (z.B. `https://rdp.domain.pit/rdpproxy/192.168.100.10`) kann man sich nun eine RDP Verbindung via RDP Proxy aufbauen.

Bequemer für den Benutzer ist es, wenn man die Verbindungen als Bookmark erfasst und dem vServer bzw. den Gruppen zuweist:

```
add    vpn    url    rdp_pitserver1    "PIT    Server1"
```

```
"rdp://192.168.100.11" -clientlessAccess ON
add    vpn    url    rdp_pitserver2    "PIT    Server2"
"rdp://192.168.100.12" -clientlessAccess ON
add    vpn    url    rdp_pitserver3    "PIT    Server3"
"rdp://192.168.100.13" -clientlessAccess ON
bind   vpn    vserver    nsgw-vsrv-rdp.domain.pit    -urlName
rdp_pitserver1
bind   vpn    vserver    nsgw-vsrv-rdp.domain.pit    -urlName
rdp_pitserver2
bind   vpn    vserver    nsgw-vsrv-rdp.domain.pit    -urlName
rdp_pitserver3
```

So sieht ein Benutzer nach erfolgreicher Anmeldung direkt seine Server auf welche er sich mittels RDP verbinden kann:



Damit eine Verbindung erfolgreich aufgebaut werden kann, muss sich der angemeldete Benutzer überhaupt auch ohne Proxy via RDP anmelden können.

Der RDP Proxy macht ein Single-Sign-On und es erfolgt keine Benutzer/Passwort Abfrage mehr.

Nun wünsch ich euch viel Spass beim Nachbauen ;-)

Skript: [NS-RDPProxy](#)