

Inhalt

Szenario	1
Das Konzept der SecurityScopes	2
Standard im Active Directory	2
Standardteilung in DomainController und (sonstige) DomainComputer	2
Das Problem	4
Design der Security Scopes	4
Implementierung in eine vorhandene AD-Infrastruktur	6
Aufbau der AdminArea	7
Aufbau der Security Scopes	8
Aufbau zusätzlicher AdminKennungen	10
Zuweisung der Computer zu den Security Scopes	11
Vor der Wirkung der GPO	11
Ab dem Wirken der GPO	12
Probleme	16
Fileserver mit domain-integriertem DFS-Namespace	16
Exchange-Server	17
Privilege Access Management	17
Fazit	17

Szenario

Active Directory ist ein Vertrauensverbund von Computern und Benutzern. Mit einer einzigen Anmeldung gegen einen DomainController kann ein Benutzer auf Ressourcen der Struktur zugreifen – eine Berechtigung vorausgesetzt. Dieses Schema bietet sehr viele Vorteile: von SingleSignOn über zentrale Verwaltung bis zur einfachen Implementierung und Bereitstellung von Standardservices.

Doch auch für einen Angriff gegen die Infrastruktur ist der Standard sehr nützlich: wird eine Identität (ein Benutzer oder ein Computer) kompromittiert, dann kann der Angreifer auf die gleichen Ressourcen zugreifen, für welche die Identität berechtigt ist. Im Umfeld der Standardbenutzer mag das vielleicht kein so großes Problem sein, da diese eher wenige Rechte bekommen. Aber was passiert, wenn eine administrative und damit höher berechtigte Identität übernommen wird? Richtig: der Angreifer kann sich sehr schnell über viele oder vielleicht alle Systeme bewegen und hat passend dazu auch hohe Rechte!

Microsoft empfiehlt natürlich die Anwendung von Least Privilege – einem Prinzip, bei dem die administrativen Kennungen mit so wenig Rechten wie nötig aufgebaut werden. Leider gibt es aber keine fertige Microsoft-Lösung. Und das Standardschema der DomainAdmins und DomainBenutzer ist nicht mehr sicher, denn hier gilt: DomainAdmins haben ihre hohen Rechte auf **jedem** System der Domäne. Also muss eine eigene Lösung her.

Vor über einem Jahr begann ich ein Experiment und entwarf ein Sicherheitskonzept, bei dem ich den DomainAdmins ihre Rechte beschränkte und dafür anderen Kennungen im ActiveDirectory diese Rechte übertrug. Die Rechte sollten nicht mehr auf alle Computer oder Server oder Clients wirken, sondern viel gezielter platziert werden – die Idee meiner Security Scopes (Sicherheitsbereiche) kam auf. Dieses (keineswegs neue) Konzept und meine Interpretation der Umsetzung möchte ich im folgenden WSHoWto vorstellen.

Für ein besseres Verständnis zeige ich zunächst, wie der Standard in einer Microsoft Domain aussieht. Dazu stelle ich dann im Vergleich meine Lösung vor. Zum Abschluss zeige ich, wie die Lösung in eine bestehende Infrastruktur integriert wird und welche Vor- und Nachteile dann zu erwarten sind. Also, lasst es uns angehen!

Das Konzept der SecurityScopes

Standard im Active Directory

Standardteilung in DomainController und (sonstige) DomainComputer

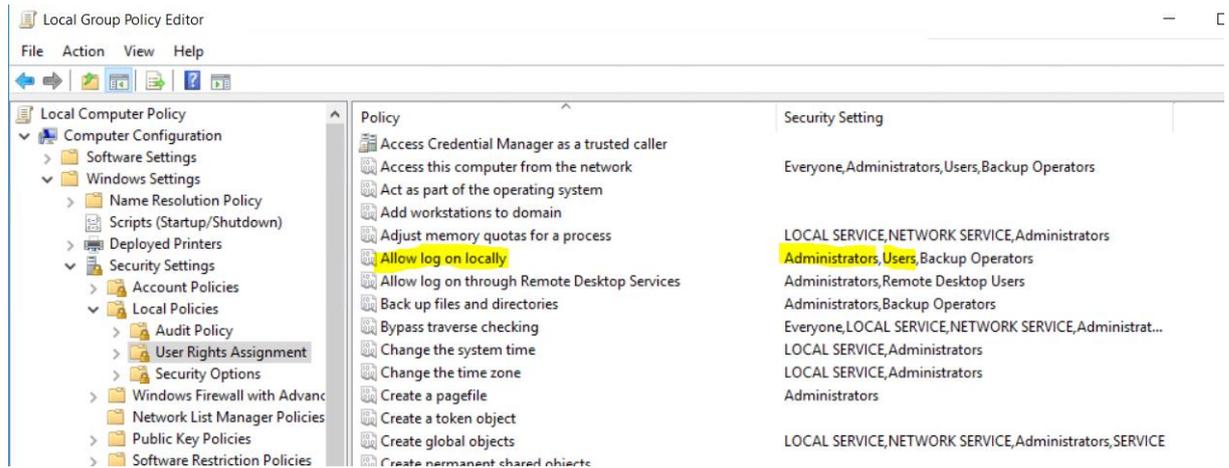
Per Default gibt es bereits in einem Active Directory einen Security Scope – also einen Sicherheitsbereich: die Organisationseinheit der DomainController. Wer genau hinschaut wird erkennen, dass es hier eine Gruppenrichtlinie gibt, die nur auf DomainController wirkt. Diese GPO definiert unter Anderem, dass nur Mitglieder der Gruppe Administrators sich anmelden dürfen:

The screenshot shows the Group Policy Management console. On the left, the tree view is expanded to 'Forest: crashwork:global' > 'Domains' > 'crashwork:global' > 'Domain Controllers' > 'Default Domain Controllers Policy'. The main pane shows the 'Default Domain Controllers Policy' details. Under 'Local Policies/User Rights Assignment', the 'Allow log on locally' policy is highlighted. The list of authorized users includes 'BUILTIN\Administrators', 'NT AUTHORITY\NETWORK SERVICE', 'NT AUTHOF SERVICE', 'NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS', 'BUILTIN\Print Op', 'BUILTIN\Server Operators', 'BUILTIN\Account Operators', 'BUILTIN\Backup Op', and 'BUILTIN\Administrators'.

Für **alle andere** Computersysteme wurde beim DomainJoin definiert, dass sich alle DomainUser in der Gruppe Benutzer befinden:

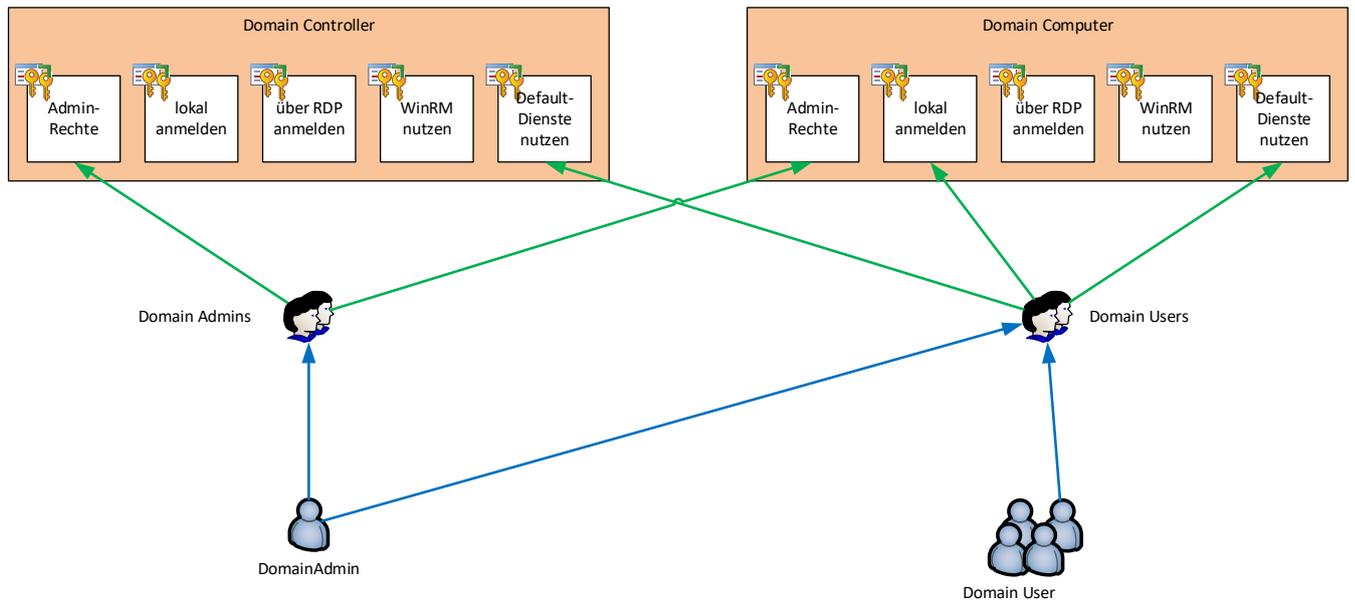
The screenshot shows the Computer Management console with 'Local Users and Groups' > 'Groups' expanded. The 'Users' group is selected. A 'Users Properties' dialog box is open, showing the 'General' tab. The 'Description' field contains the text: 'Users are prevented from making accidental or intentional system-wide changes and can run most'. The 'Members' list includes 'crashwork\Domain Users', 'NT AUTHORITY\Authenticated Users (S-1-5-11)', and 'NT AUTHORITY\INTERACTIVE (S-1-5-4)'. The 'Add...' button is highlighted.

Und die Lokale Richtlinie steuert die Berechtigung dieser Gruppe:



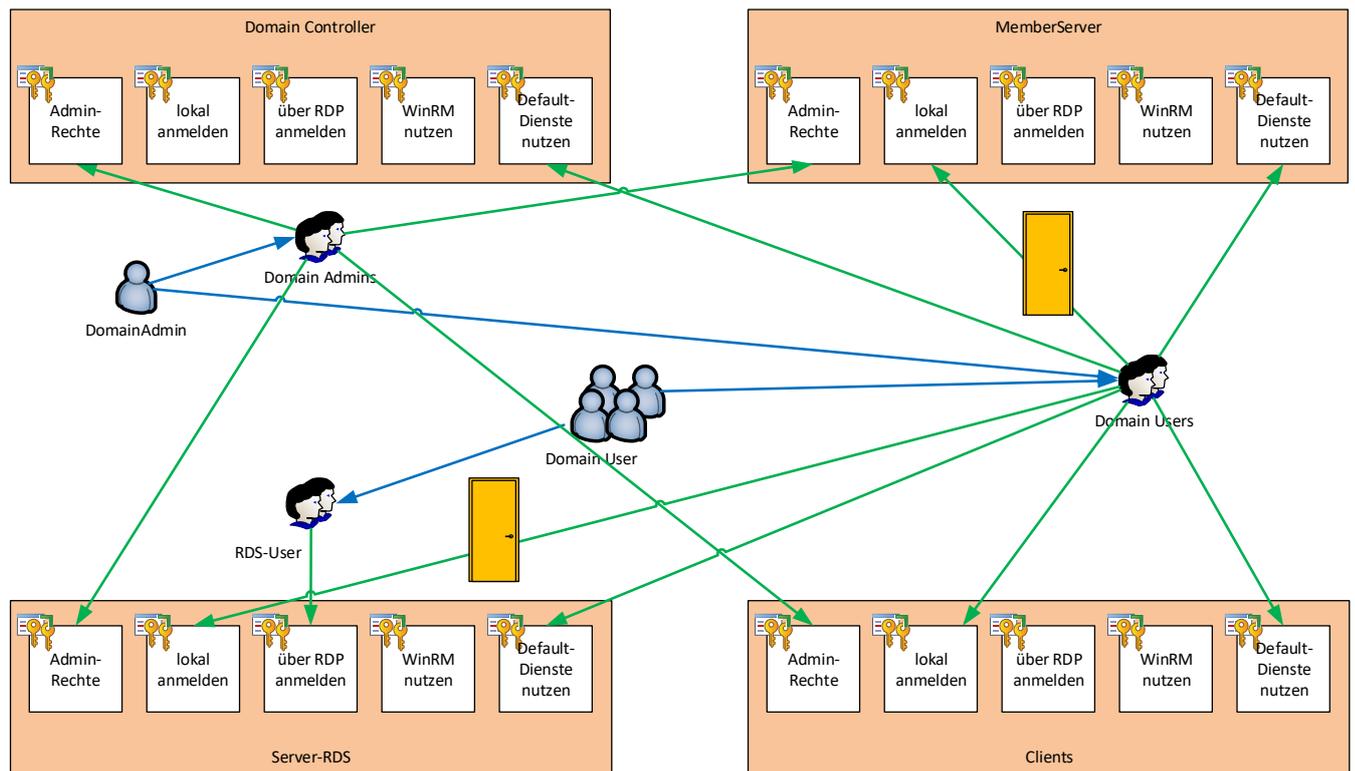
Man erkennt an diesem Beispiel deutlich die Trennung zwischen DomainAdmins und DomainUsers auf den Sicherheitsbereichen DomainController und DomainComputern. Diese Trennung wird natürlich noch in weiteren Berechtigungen sichtbar – ein Beispiel soll aber mal genügen.

Auf einige wichtige Rechte heruntergebrochen könnte das Standardschema also dieses Layout haben:



Natürlich werden sich Standardbenutzer nicht auf Windows MemberServern, sondern nur auf Clients anmelden. Die technische Barriere stellt in der Regel die verschlossene Tür zum Serverraum dar. Stellt man aber einen Windows Server anstelle des normalen Windows Clients an einen Arbeitsplatz, dann kann ein Standardbenutzer damit genauso arbeiten!

Ebenso gibt es vielleicht die RDP-Ausnahmen für RemoteDesktopServices. Somit könnte ein übliches, gelebtes Layout folgendem Bild entsprechen:



Es sind in diesem Schaubild bereits 4 Ebenen definiert:

- DomainController
- StandardServer
- StandardClients
- RDS-Server

Die Isolation basiert auf physischer Trennung, einigen Netzwerkfiltern und der Standardtrennung im ActiveDirectory.

Das Problem

Wo ist nun das Problem? Ausgehend davon, dass die (physischen) Türen nicht überwunden werden können, geht von den Standardbenutzern keine große Gefahr aus, denn sie sind (hoffentlich) gering und **nicht administrativ** berechtigt. Doch die Administratoren sind auf allen Systemen unterwegs! **EIN** kompromittiertes System genügt für einen Angreifer, damit er sich auf **ALLE** Systeme ausbreiten kann – denn so weit kommt ein DomainAdmin nun mal...

Typische Angriffe verlaufen nach diesem Muster:

- Ein Angreifer hat einen Standardbenutzer auf einem Client dazu gebracht, einen Schadcode zu starten (das könnte eine infizierte Datei, ein Link zu einer Website, ein BadUSB-Device, ... sein). Der Angreifer hat nur Standardbenutzerrechte
- Vielleicht gibt es auf dem Client eine Schwachstelle, die der Angreifer ausnutzen kann, um sich lokale Systemrechte zu holen. Dann könnte er mit Tools wie Mimikatz Passwortinformationen auf dem Client auslesen. Wenn er keine interessanten Credentials findet, dann könnte er den Benutzer „ärgern“ und so dafür sorgen, dass sich jemand vom Helpdesk zur Unterstützung anmeldet – so kommt er an die Rechte eines ClientAdmins.
- Nun kann er sich auch auf andere Clients bewegen. Bei einem Client findet er vielleicht die Anmeldung eines Server- oder DomainAdmins. Diese kann er verwenden, um auf einen Server zu wechseln. Wenn die Kennung sogar auf dem DomainController berechtigt ist, dann kommt er direkt zu den DCs
- Das wars dann meist schon – für die Infrastruktur...

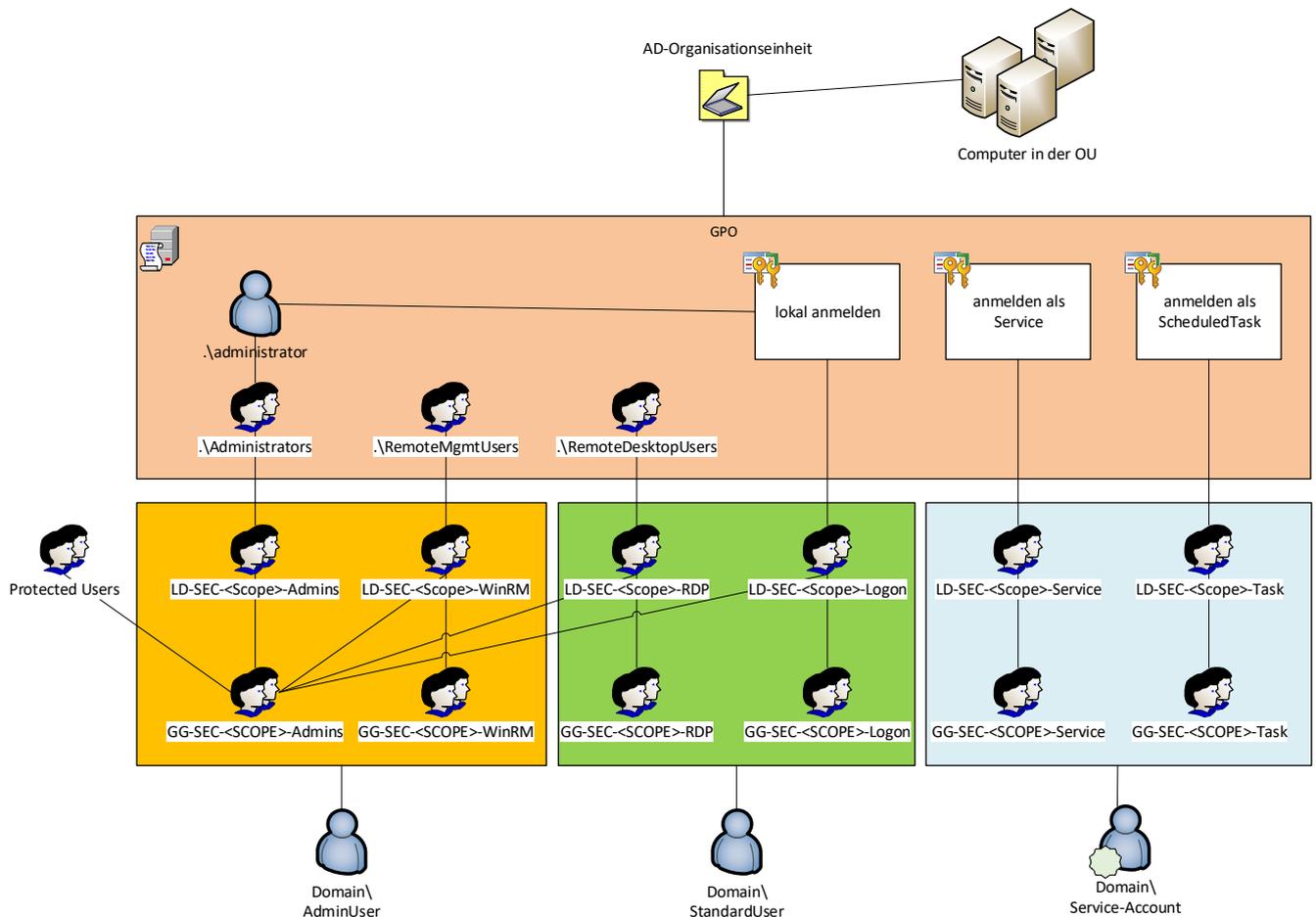
Design der Security Scopes

Nun kann man mit Gruppenrichtlinien und zusätzlichen Sicherheitsgruppen im Active Directory, ein paar Organisationseinheiten und vielleicht ein paar PowerShell-Zeilen (☺) eigene Sicherheitsbereiche definieren. Genau das war mein Ansatz:

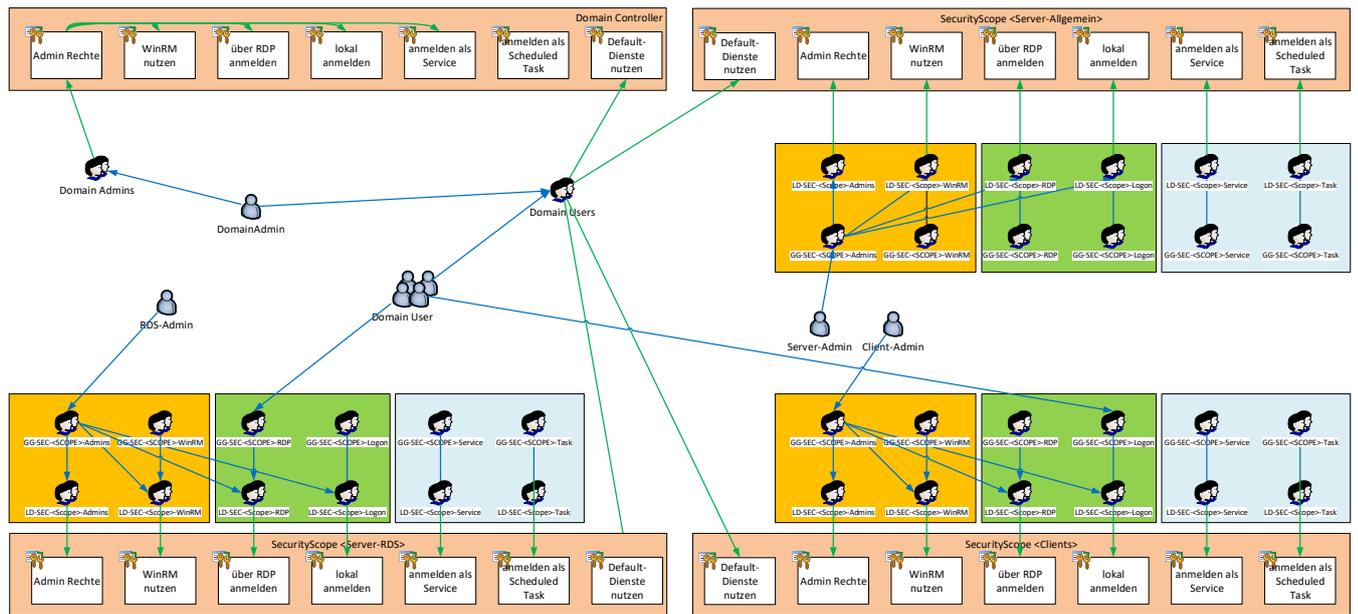
- Ich habe mir 6 übliche Rechte herausgesucht, die ich auf Clients und Memberservern immer wieder an Benutzer delegieren muss und möchte (die Liste ist natürlich erweiterbar):
 - das Recht auf eine lokale Anmeldung
 - das Recht auf eine RDP-Anmeldung
 - das Recht auf die lokale Administration
 - das Recht als Service zu starten
 - das Recht als geplante Aufgabe (Task) zu starten
 - das Recht für einen Verbindungsaufbau über Windows RemoteManagement (WinRM)
- Für jedes Recht habe ich eine Gruppe im Active Directory gebaut.
- Für die Rechtezuweisung habe ich die passende Stelle in einer Gruppenrichtlinie gesucht:
 - Manche Rechte lassen sich über Mitgliedschaften in lokalen Gruppen erreichen
 - Andere Rechte werden in den Sicherheitseinstellungen der GPO definiert.
- Die GPO habe ich auf eine neue Organisationseinheit im AD verlinkt.
- Und in diese OU habe ich die gewünschten Computer des neuen Sicherheitsbereiches positioniert.
- Da meine Regeln die bestehenden Rechte überschreiben, fliegen auch automatisch die DomainAdmins und ggf. die DomainUser raus. Die Berechtigung ist damit erst einmal „genullt“.
- Mit der Mitgliedschaft in den neuen Gruppen kann ich nun ganz genau steuern, wer im AD welche Rechte auf den Zielsystemen bekommt.

Da ich bei meiner Infrastruktur das Berechtigungsmodell AGDLP verwende, habe ich **Rechtegruppen** und **Rollengruppen** im AD aufgebaut und diese sinnvoll ineinander verschachtelt.

Diese Darstellung zeigt mein neues Rechteschema für einen SecurityScope:



Im Vergleich zu meinem ersten Schaubild (die Standard-Infrastruktur) würde dann folgendes Layout entstehen:



Man erkennt (auch wenn es etwas klein ist), dass jeder SecurityScope seine eigenen administrativen Gruppen hat. Ebenso gibt es hier neue administrative Konten, welche die einzelnen Scopes technisch betreuen können – ohne dass es Überlagerungen und somit Sicherheitsüberschneidungen geben muss:

- Wird ein Client kompromittiert, dann kann ein Angreifer keine Maschine übernehmen, auf der ein Serveradmin angemeldet war, denn diese haben hier kein LoginRecht und somit kann auch kein PasswordHash oder Ähnliches hinterlassen werden.
- Ebenso kann auf einem Server kein Hash eines DomainAdmins gefunden werden, denn diese dürfen sich auch nur auf DomainControllern anmelden!
- Ein Angreifer ist also auf seinen bereits erbeuteten Scope beschränkt!

Ein einfaches Prinzip. ☺

Implementierung in eine vorhandene AD-Infrastruktur

Gerade wenn mehrere Sicherheitsebenen benötigt werden, wird eine Scriptlösung die bessere Option zur Anlage darstellen. Daher habe ich mir ein paar PowerShell-Funktionen aufgebaut, mit der

- die Ersteinrichtung der SecurityScopes
- das Aufbauen der SecurityScopes
- die Anlage von neuen AdminKennungen

strukturell und inhaltlich geführt wird. Die einzelnen Funktionen stelle ich in diesem Abschnitt vor.

Meine Lösung besteht dabei aus 2 Scripten. Das erste Script baut im RAM die erforderlichen Funktionen auf:

```

AD-SecurityScopes.ps1 X AD-SecurityScopes-Caller.ps1* X
1 ##### Scriptinfo #####
2 # Scriptreihe: AD-SecurityScopes
3 # Datum: 2017-12-01
4 # Version: V1.12
5 # Programmierer: Stephan Walther
6 #####
7
8 # Funktionen
9 + function Protokoll {...}
23
24 + function teste-ADGroup {...}
32
33 + function erstelle-ADGroups {...}
52
53 + function neues-ADGroupMember {...}
74
75 + function ermittle-SID {...}
80
81 + function erstelle_OUs {...}
105
106 + function erstelle-ADUser {...}
126
127 + function erstelle-AdminArea {...}
294
295 + function erstelle-SecurityScope {...}
415
416 + function erstelle-AdminUser {...}
447
448 # SIG # Begin signature block
449 # MIIII4QYJKoZIhvcNAQcCoIIIOjCCCM4CAQEXCZAJBgUrDgMCGGUAMGkGCi sGAQQB

```

Das 2. Script ist der „caller“. Darin stehen die möglichen Aufrufe der Funktionen des ersten Scriptes. Es dient somit auch zur Dokumentation:

```

AD-SecurityScopes.ps1 AD-SecurityScopes-Caller.ps1* X
1 ##### Scriptinfo #####
2 # Scriptreihe: AD-SecurityScopes
3 # Datum: 2017-10-12
4 # Version: V1.10
5 # Programmierer: Stephan Walther
6 #####
7
8 # lade Funktionen
9 # starte Script AD-SecurityScopes.ps1
10 cls
11
12 # erstelle AdminArea
13 erstelle-AdminArea `
14 -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global" `
15 -ScriptPfad "C:\Admin\SecurityScope"
16
17 # erstelle Scopes
18 erstelle-SecurityScope `
19 -ScopeName "StandardServer" `
20 -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global" `
21 -ScopeOU "OU=StandardServer,OU=MemberServer,OU=crashwork,DC=crashwork,DC=global" `
22 -ScriptPfad "C:\Admin\SecurityScope"
23 -PrefixGPO "GPO-Computer-Security"
24
25 erstelle-SecurityScope `
26 -ScopeName "RDS" `
27 -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global" `
28 -ScopeOU "OU=RDS,OU=MemberServer,OU=crashwork,DC=crashwork,DC=global" `
29 -ScriptPfad "C:\Admin\SecurityScope"
30 -PrefixGPO "GPO-Computer-Security"

```

Aufbau der AdminArea

Ich habe die Komponenten der SecurityScopes gerne separat gespeichert. Damit kann ich leichter Aufgaben im Active Directory an weniger privilegierte Konten delegieren, ohne dass diese ihre eigenen Rechte zu einfach erhöhen können. Diese Ablage nenne ich mal AdminArea. Sie ist einfach eine Organisationseinheit, die separat geführt wird.

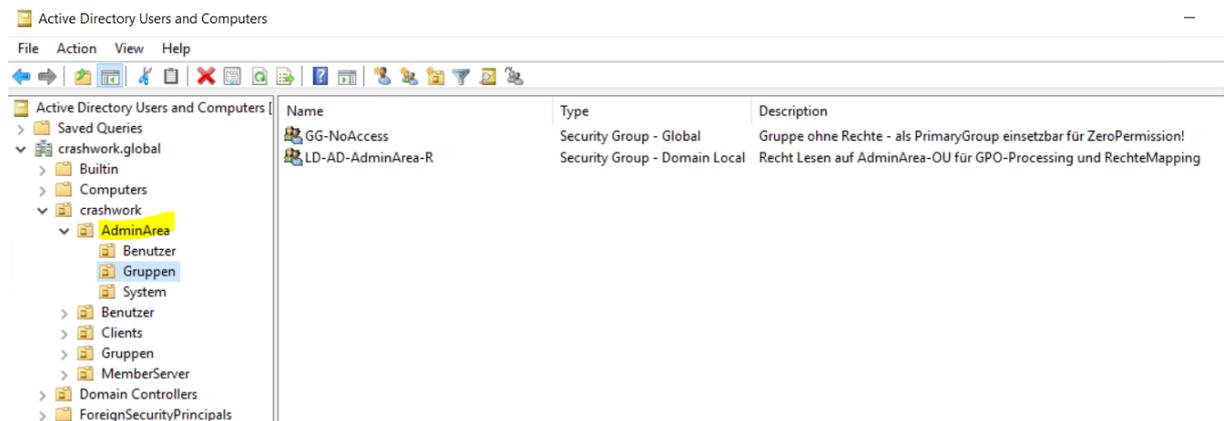
Die AdminArea wird nur einmal aufgebaut. Dabei werden alle OUs erzeugt. Zusätzlich werden einige wichtige Gruppen und Benutzer verschoben. Die Funktion muss dafür nur den Pfad (DN) der OU AdminArea als Parameter bekommen. Alles Weitere wird automatisch ausgeführt:

```

AD-SecurityScopes.ps1 AD-SecurityScopes-Caller.ps1* X
12 # erstelle AdminArea
13 erstelle-AdminArea
14 -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global"
15 -ScriptPfad "C:\Admin\SecurityScope"

konfiguriere AdminArea
  erstelle Organisationseinheiten
    OU=crashwork,DC=crashwork,DC=global gibt es schon
    OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=Benutzer,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=Gruppen,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=System,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
  erstelle Basisgruppen
    Gruppe GG-NoAccess wurde erstellt
    Gruppe LD-AD-AdminArea-R wurde erstellt
  verschiebe SystemGruppen nach 'OU=System,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global'
    'Protected Users'
    'Domain Admins'
    'Schema-Admins'
    'Organisation Admins'
  passe den Account Administrator an
    verschiebe Administrator nach 'OU=Benutzer,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global'
    Gruppenmitglied Administrator -> Protected Users wird erstellt
  konfiguriere ACL der AdminArea
    OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=Benutzer,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=Gruppen,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
    OU=System,OU=AdminArea,OU=crashwork,DC=crashwork,DC=global
  
```

Das Ergebnis kann in der MMC gesichtet werden:



Aufbau der Security Scopes

Im nächsten Schritt werden die Sicherheitsbereiche aufgebaut. Dafür muss man zunächst überlegen, welcher Server zu welchem Bereich gehören soll. Dabei gelten folgende Regeln und Empfehlungen (meinerseits):

Regel & Empfehlung	Beschreibung
Regel: eindeutige Scopes	<ul style="list-style-type: none"> Ein Computer kann nur einem SecurityScope angehören! (Es kann nur eine resultierende GPO geben!)
Regel: gleicher Service == gleicher Scope	<ul style="list-style-type: none"> Computer, welche die gleichen Services anbieten (z.B. mehrere Exchange Server) sollten in dem gleichen SecurityScope liegen. Ausnahmen sind natürlich möglich: ich habe für meine RDS-Infrastruktur 3 SecurityScopes für unterschiedliche Berechtigungen aufgebaut. Hier ist dann aber viel mehr Planung erforderlich.
Regel: Namenskonzept	<ul style="list-style-type: none"> Die SecurityScopes sollten von Anfang an ein festes Namenskonzept erhalten. Die Benennung eines Scopes zu ändern ist sehr unschön!
Empfehlung: Admin für mehrere Scopes	<ul style="list-style-type: none"> Ein Admin kann Mitglied in mehreren SecurityScope-Admingruppen sein → eine Trennung der Computer kann also recht granular erfolgen (ich habe Scopes mit nur einem Computerobjekt! 😊)
Empfehlung: neue OUs	<ul style="list-style-type: none"> Baut bitte neue OUs für die Scopes auf. Meine Funktion mappt die neuen, scharfen GPO auf die OU.

- Das Verschieben der bestehenden Server sollte mit einer Testphase geschehen. Sägt bitte nicht den Ast vom AdminBaum ab...

Wenn das alles klar ist, dann kann der erste Scope gebaut werden. Angenommen, dieser Scope soll alle HyperV-Hosts absichern und administrativ isolieren, dann könnte folgender Funktionsaufruf die Infrastruktur vorbereiten. Die Parametrisierung der Funktion ist dabei recht einfach:

- Name → der Name des Scopes (bitte ohne Sonderzeichen)
- AdminArea → der Pfad zur OU AdminArea
- ScopeOU → der Pfad zu der (bitte neuen) OU. Da kommen dann die Server des Scope rein
- ScriptPfad → die Funktion protokolliert in diesem Pfad
- PrefixGPO → die neue GPO bekommt einen Bezeichner. Das Präfix könnt ihr hier definieren.

So könnte es dann aussehen:

```

AD-SecurityScopes.ps1  AD-SecurityScopes-Caller.ps1* X
46  erstelle-SecurityScope
47  -ScopeName "HyperV"
48  -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global"
49  -ScopeOU "OU=HyperV,OU=MemberServer,OU=crashwork,DC=crashwork,DC=global"
50  -ScriptPfad "C:\Admin\SecurityScope"
51  -PrefixGPO "GPO-Computer-Security"
52

```

```

erstelle SecurityScope HyperV
konfiguriere Gruppen im AD
  Rollengruppen
    Gruppe GG-SEC-HyperV-Admins wurde erstellt
    Gruppe GG-SEC-HyperV-Login wurde erstellt
    Gruppe GG-SEC-HyperV-RDP wurde erstellt
    Gruppe GG-SEC-HyperV-Task wurde erstellt
    Gruppe GG-SEC-HyperV-Service wurde erstellt
    Gruppe GG-SEC-HyperV-WinRM wurde erstellt
  Rechtegruppen
    Gruppe LD-SEC-HyperV-Admins wurde erstellt
    Gruppe LD-SEC-HyperV-Login wurde erstellt
    Gruppe LD-SEC-HyperV-RDP wurde erstellt
    Gruppe LD-SEC-HyperV-Task wurde erstellt
    Gruppe LD-SEC-HyperV-Service wurde erstellt
    Gruppe LD-SEC-HyperV-WinRM wurde erstellt
  definiere Gruppenmitgliedschaften
    Gruppenmitglied GG-SEC-HyperV-Admins -> LD-SEC-HyperV-Admins wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Admins -> LD-SEC-HyperV-Login wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Admins -> LD-SEC-HyperV-RDP wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Admins -> LD-SEC-HyperV-WinRM wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Login -> LD-SEC-HyperV-Login wird erstellt
    Gruppenmitglied GG-SEC-HyperV-RDP -> LD-SEC-HyperV-Login wird erstellt
    Gruppenmitglied GG-SEC-HyperV-RDP -> LD-SEC-HyperV-RDP wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Task -> LD-SEC-HyperV-Task wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Service -> LD-SEC-HyperV-Service wird erstellt
    Gruppenmitglied GG-SEC-HyperV-WinRM -> LD-SEC-HyperV-WinRM wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Admins -> Protected Users wird erstellt
    Gruppenmitglied GG-SEC-HyperV-Admins -> LD-AD-AdminArea-R wird erstellt
erstelle Management-OU
  OU=HyperV,OU=MemberServer,OU=crashwork,DC=crashwork,DC=global
konfiguriere GPO 'GPO-Computer-Security-Scope-HyperV'
  erstelle leere GPO
  kopiere Template
  editiere Template
  importiere Template in die GPO
  entferne Template
  linke GPO auf ManagementOU

```

Man erkennt deutlich das Ergebnis: es werden Gruppen, OUs und die GPO aufgebaut – alles natürlich mit den notwendigen Bezügen untereinander.

Im Active Directory sind folgende Änderungen erkennbar:

Name	Type	Description
GG-NoAccess	Security Group - Global	Gruppe ohne Rechte - als PrimaryGroup einsetzbar für ZeroPermission!
GG-SEC-HyperV-Admins	Security Group - Global	Rolle Administratoren für HyperV
GG-SEC-HyperV-Login	Security Group - Global	Rolle Konsolenlogin auf HyperV
GG-SEC-HyperV-RDP	Security Group - Global	Rolle RDP-Login auf HyperV
GG-SEC-HyperV-Service	Security Group - Global	Rolle 'Logon as a Service' auf HyperV
GG-SEC-HyperV-Task	Security Group - Global	Rolle 'Logon as a Batch Job' auf HyperV
GG-SEC-HyperV-WinRM	Security Group - Global	Rolle 'Remoteverwaltungsbenutzer (WinRM)' auf HyperV
LD-AD-AdminArea-R	Security Group - Domain Local	Recht Lesen auf AdminArea-OU für GPO-Processing und RechteMapping
LD-SEC-HyperV-Admins	Security Group - Domain Local	Recht Administratoren für HyperV
LD-SEC-HyperV-Login	Security Group - Domain Local	Recht Konsolenlogin auf HyperV
LD-SEC-HyperV-RDP	Security Group - Domain Local	Recht RDP-Login auf HyperV
LD-SEC-HyperV-Service	Security Group - Domain Local	Recht 'Logon as a Service' auf HyperV
LD-SEC-HyperV-Task	Security Group - Domain Local	Recht 'Logon as a Batch Job' auf HyperV
LD-SEC-HyperV-WinRM	Security Group - Domain Local	Recht 'Remoteverwaltungsbenutzer (WinRM)' auf HyperV

Dazu gibt es nun eine neue GPO, welche auf der neuen OU gemappt ist:

Der Prozess kann nun mehrmals wiederholt werden, bis einige/alle aktuell benötigten Security-Scopes aufgebaut sind. Das geht natürlich auch nachträglich. ☺

Aufbau zusätzlicher AdminKennungen

Nun fehlt nur noch eine Adminkennung, denn weder die DomainAdmins noch sonst eine andere Identität außer dem lokalen Admin haben ab dem Wirken der GPO Zugriff auf den Server im SecurityScope! Auch dafür gibt es eine Funktion:

```

AD-SecurityScopes.ps1 AD-SecurityScopes-Caller.ps1* X
68     erstelle-AdminUser
69     -ScopeName "HyperV"
70     -AdminUser "ADM-Paul"
71     -StandardUser "Paul.Paulsen"
72     -AdminArea "OU=AdminArea,OU=crashwork,DC=crashwork,DC=global"
73     -ScriptPfad "C:\Admin\SecurityScope"
74
erstelle AdminBenutzer ADM-Paul
ADM-Paul existiert bereits
definiere Gruppenmitgliedschaften
Gruppenmitglied ADM-Paul -> GG-SEC-HyperV-Admins wird erstellt
Gruppenmitglied ADM-Paul -> GG-NoAccess wird erstellt
definiere PrimaryGroup == GG-NoAccess
PS C:\>
    
```

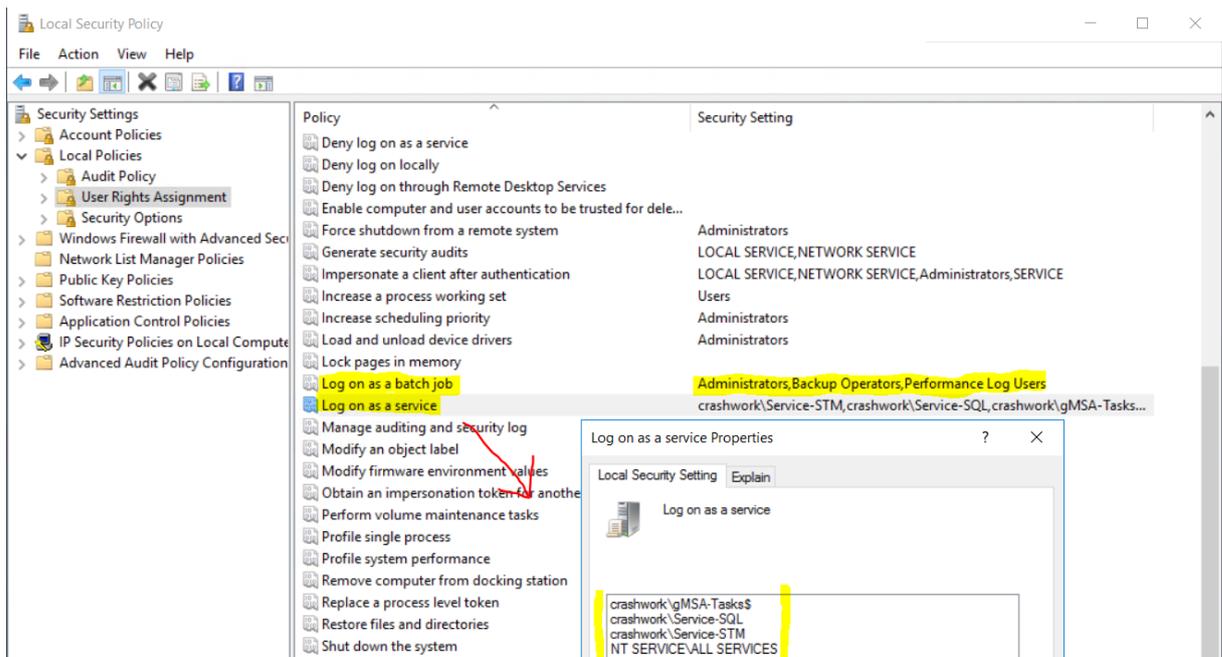
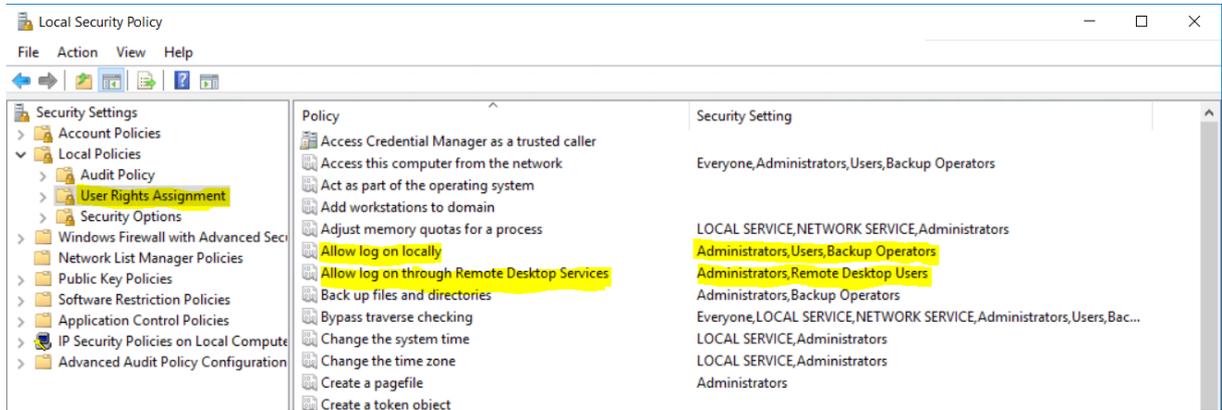
Falls der Benutzer schon existiert, wird er natürlich übernommen. Der Kennung wird dabei die primäre Mitgliedschaft der Gruppe „DomainUser“ entzogen. Stattdessen wird der Account in die Gruppe **GG-NoAccess** aufgenommen. Diese Gruppe wird mit der AdminArea erstellt und wird **NIEMALS** berechtigt. Daher ist der Account dann auch nahezu rechtfrei. Erst durch die zusätzliche Mitgliedschaft in der AdminGruppe des jeweiligen SecurityScopes (oder welche Gruppe auch immer für die Arbeit erforderlich ist – bitte least Privilege anwenden) wird der Benutzer für den neuen Aufgabenbereich berechtigt.

Zuweisung der Computer zu den Security Scopes

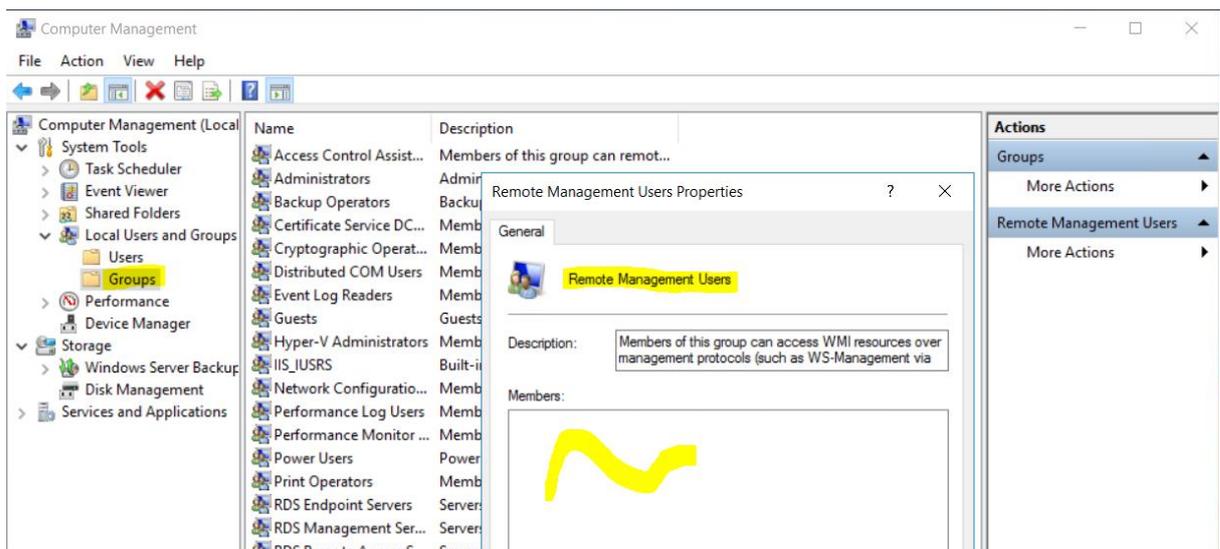
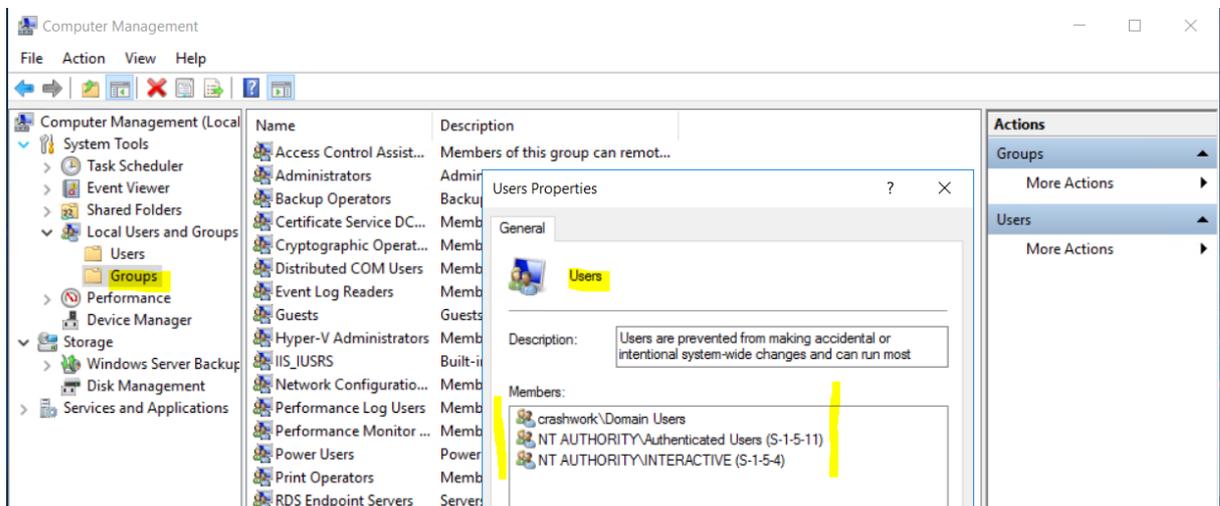
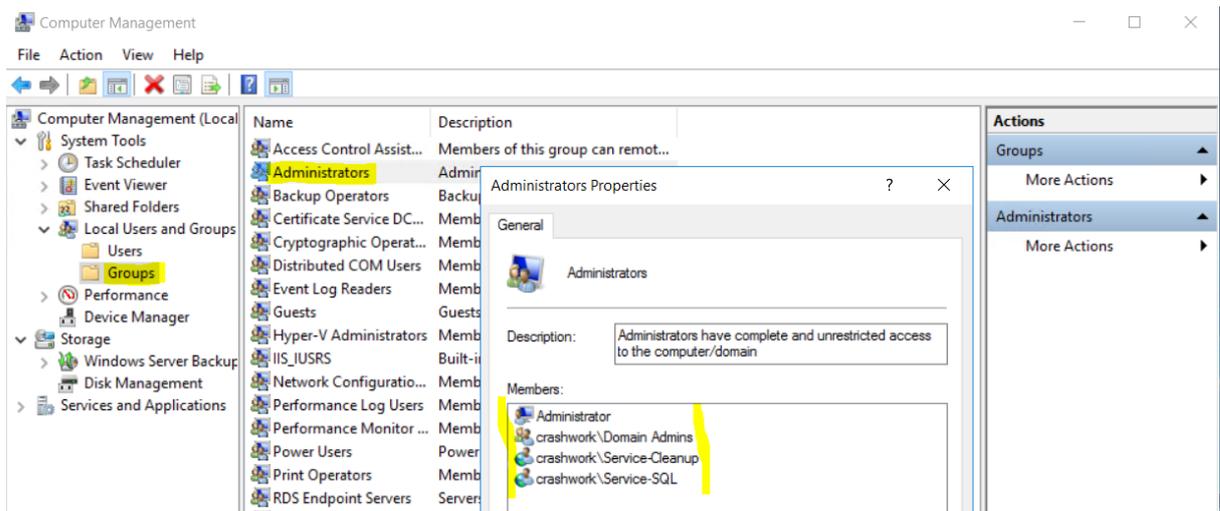
Wenn alle Vorarbeiten abgeschlossen sind, dann kann ein Computer in den Wirkungsbereich der GPO verschoben werden. Alternativ kann natürlich auch die GPO auf einer bestehenden Organisationseinheit verknüpft werden. Das Konzept ist durchaus flexibel. 😊

Vor der Wirkung der GPO

Ein Blick auf die aktuelle Konfiguration kann dabei nicht schaden. Für meine Änderungen sollten vorher die UserRights in der Managementkonsole SecPol.msc gesichtet und protokolliert werden:



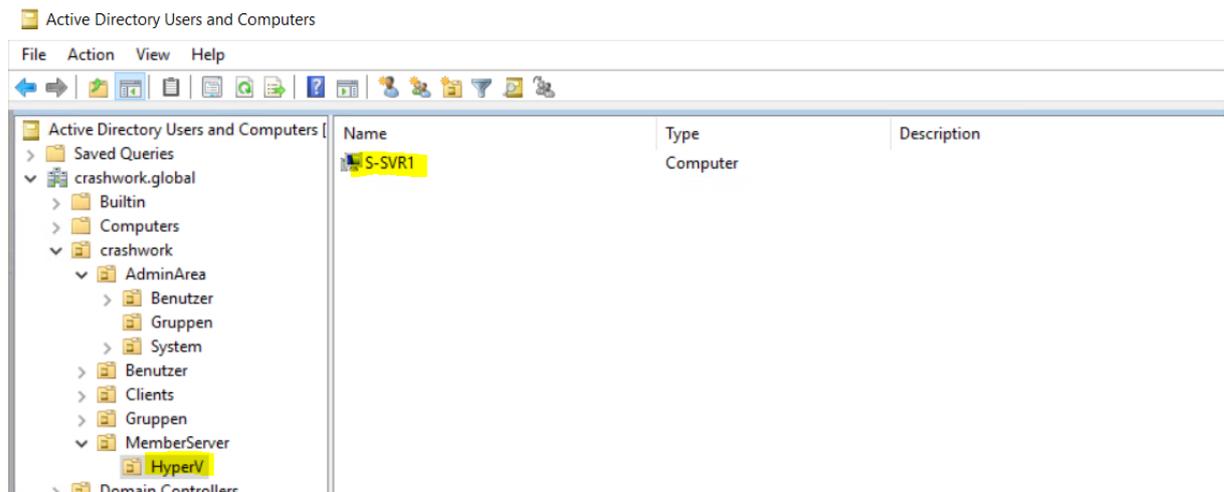
Aber auch die Mitgliedschaften der lokalen Gruppen könnten im Vorfeld modifiziert worden sein:



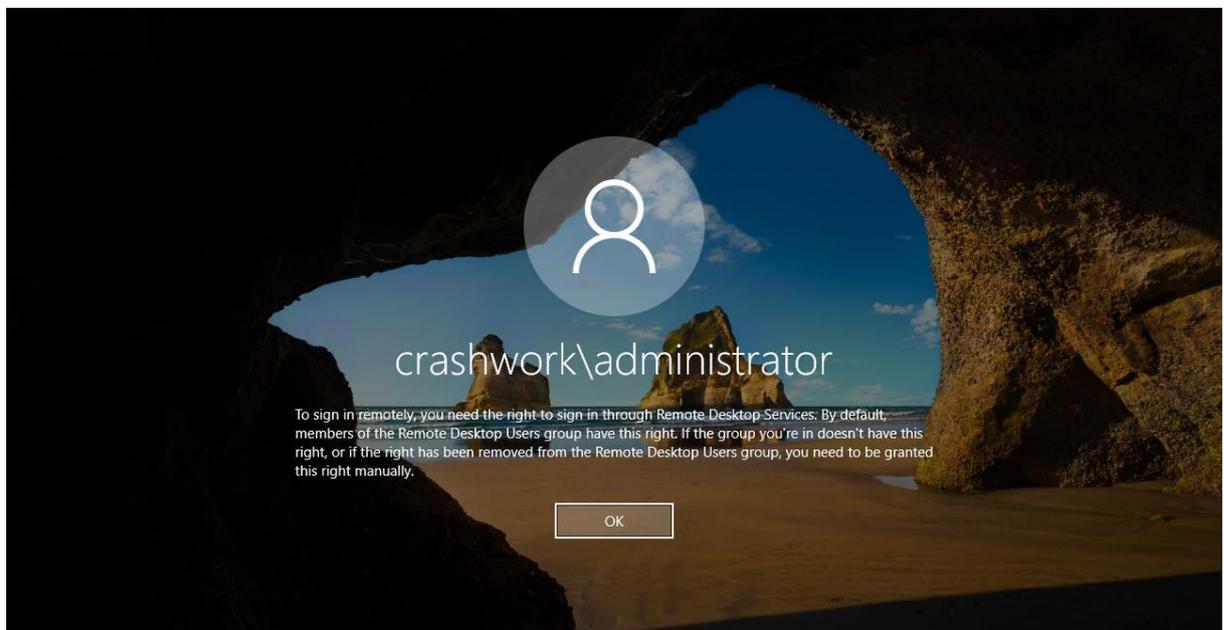
Die Analyse der IST-Situation ist sinnvoll, da diese Einstellungen mit der GPO überschrieben werden!

Ab dem Wirken der GPO

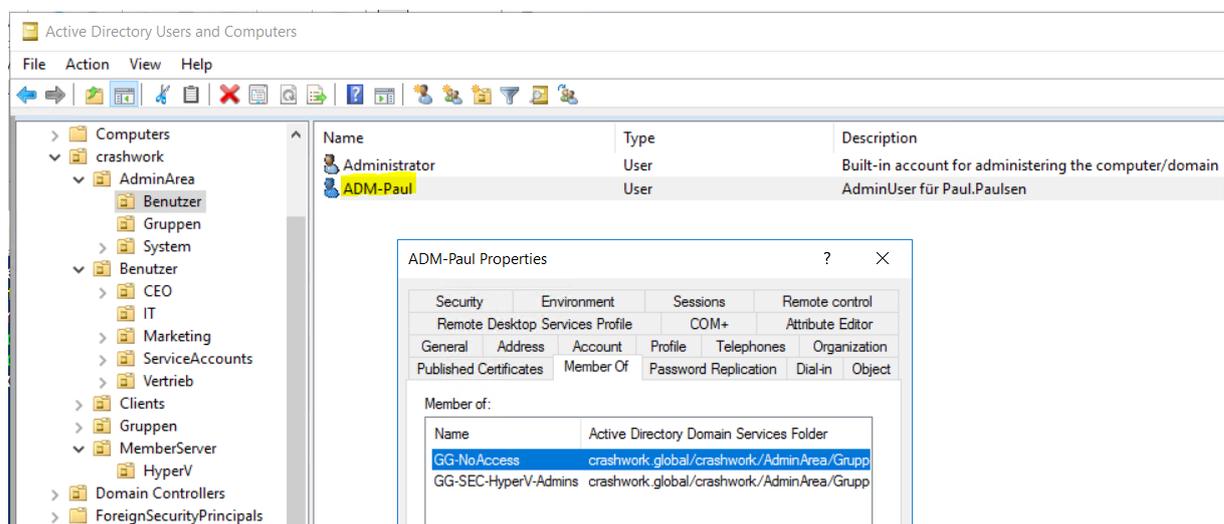
Nun verschiebe ich den Computer in die Organisationseinheit und starte ihn neu:



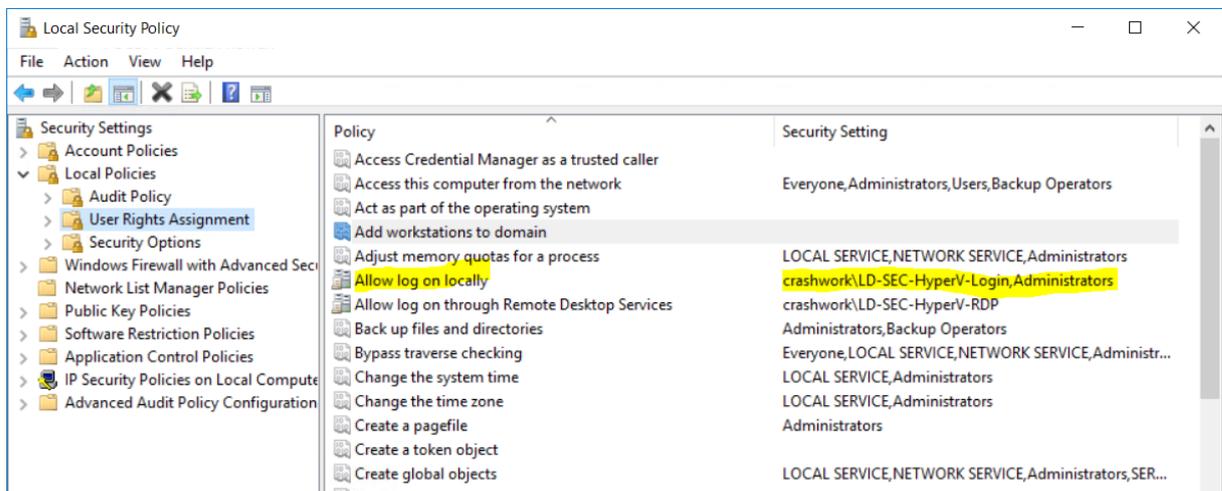
Eine Neuanmeldung mit dem Konto des DomainAdmins wird nun keinen Erfolg mehr haben:



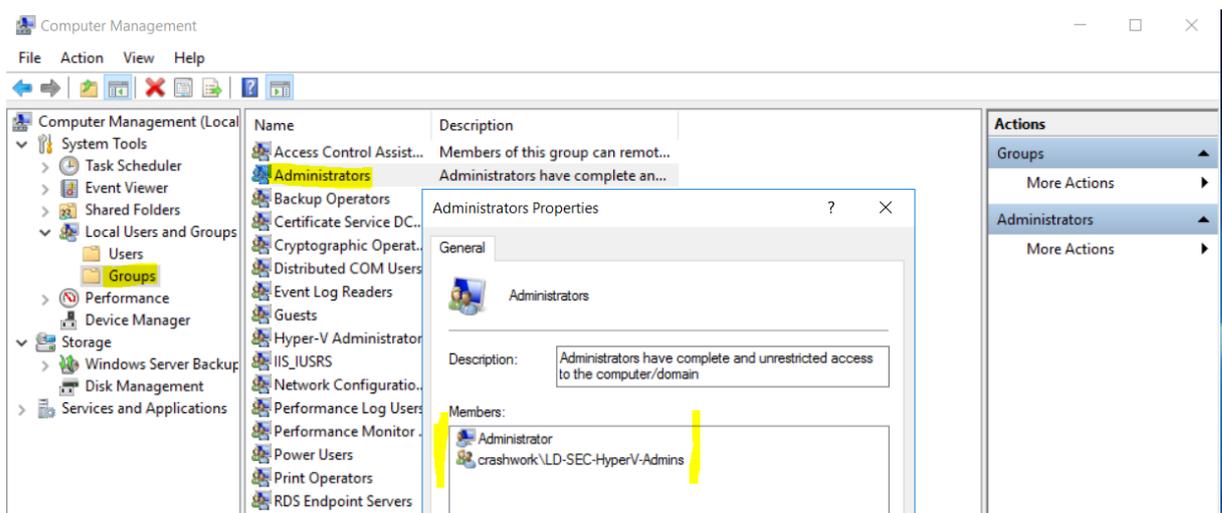
Mit dem richtigen Konto dagegen funktioniert es:



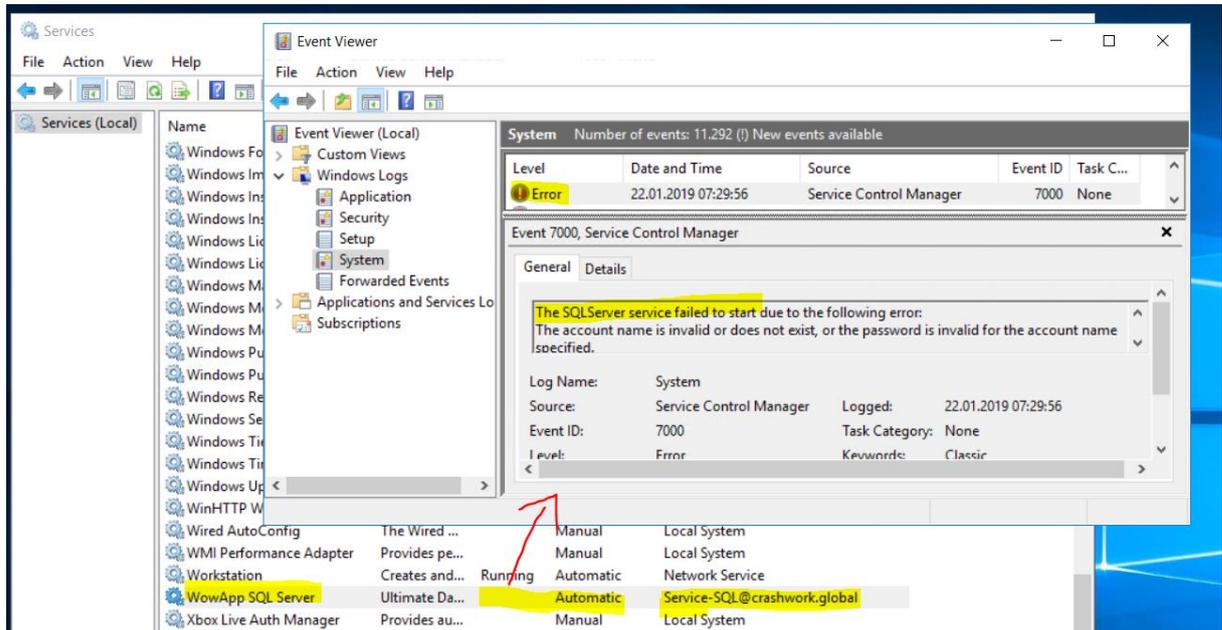
Am Beispiel des Logon-Rechtes kann die Ursache sehr einfach gezeigt werden. Das Recht wurde durch die GPO neu vergeben. Da der AdminAccount **ADM-Paul** über die Gruppenmitgliedschaft **GG-SEC-HyperV-Admins** auf Mitglied in der Gruppe **LD-SEC-HyperV-Login** ist, kann er sich anmelden:



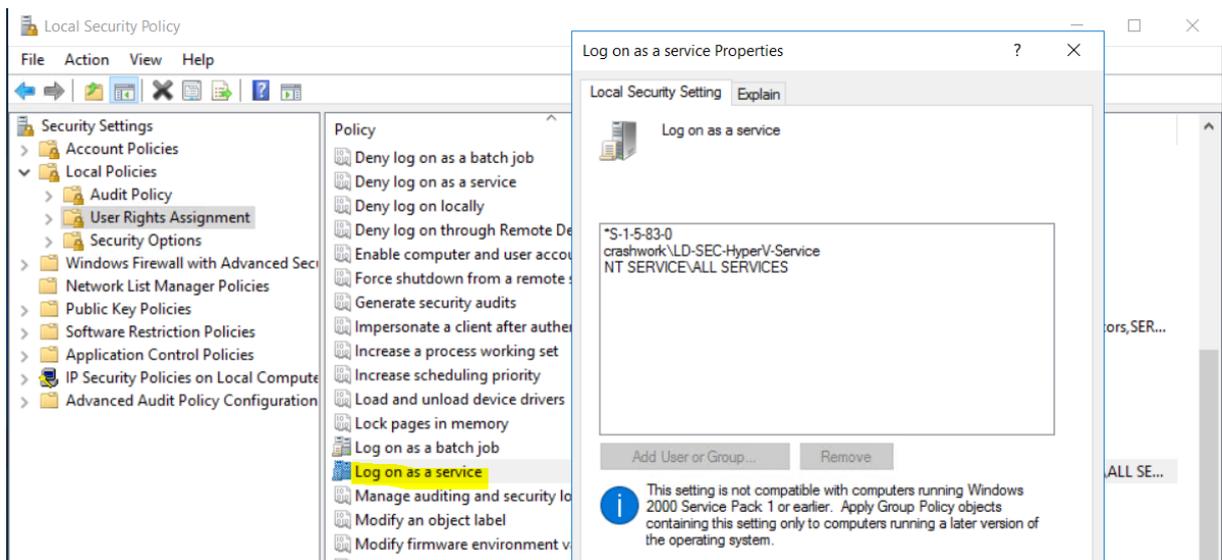
Aber die Administratoren dürfen das doch auch?? Warum kann dann der DomainAdmin sich nicht anmelden?? Ganz einfach: der Account ist nicht länger in der Gruppe der lokalen Admins enthalten (der gelistete Administrator ist das lokale Konto):



Und so zieht sich die GPO durch die gezeigten Konfigurationen. Bereits früher gesetzte Berechtigungen werden überschrieben. Daher sollte **vorher** geschaut werden, welche Identitäten nun in die neuen Gruppen aufgenommen werden müssen. Ein Beispiel finden wir hier: ein Dienst startet nicht mehr...

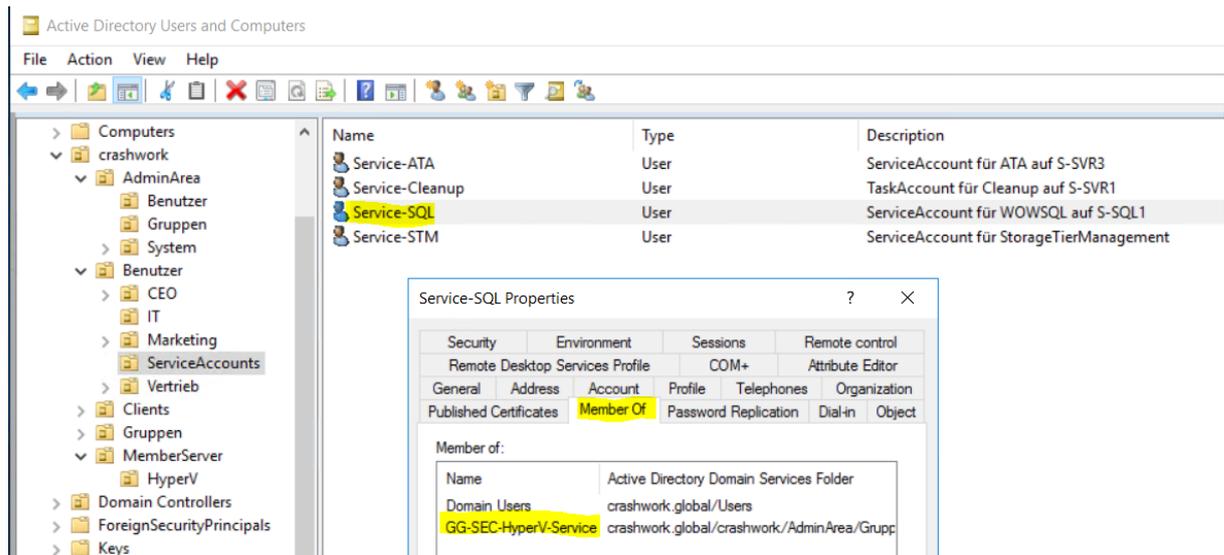


Das ist auch richtig, denn der ServiceAccount hat das Recht nicht mehr:

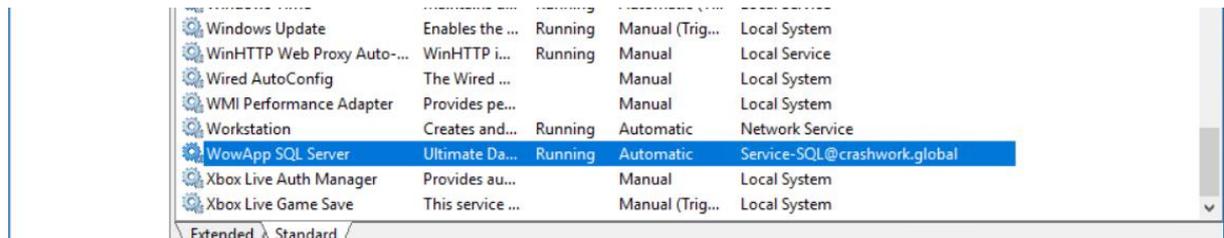


Wer die Konfiguration **vor** der GPO gesehen hat wird feststellen, dass der Benutzer Service-SQL das Recht „Logon as a Service“ hatte. Doch dieses Recht hat jetzt nur noch die neue Gruppe im Active Directory.

Der Account muss also in die neue Gruppe aufgenommen werden:



Dann lässt sich der Service auch wieder starten:



Einige Administratoren sehen dieses Vorgehen nachteilig an. Ich meine aber, dass eine zentrale Vergabe und Definition der Berechtigungen auf lange Sicht viel einfacher zu überschauen ist. Mal ehrlich: hättet ihr vorher gewusst, wo der Benutzer Service-SQL welche Rechte besaß? (Abgesehen von der Description im Active Directory natürlich ☺)

Der Aufbau dieser SecurityScopes ist durchaus mit Aufwand verbunden, kann aber auch gleich als Review und Redesign bestehender Berechtigungen genutzt werden. Und neue Services und Systeme müssen eh sauber plaziert werden!

Probleme

Nach dem Aufbau werden ggf. weitere Zusammenhänge in der täglichen Administration sichtbar. Hier kommen einige Beispiele aus meiner eigenen Infrastruktur. In dieser habe ich mehrere Sicherheitsbereiche definiert:

SecurityScope	AdminKonto	Beschreibung
DomainController	sysadm	alle DomainController (default)
HyperV	stephan-ad	alle Hosts mit der Rolle Hyper-V
MX	stephan-ad	alle Exchange-Server
Standardserver	stephan-ad	alle sonstigen Memberserver: z.B. Fileserver
...

Nach dem Umbau habe ich einige Probleme erkannt:

Fileserver mit domain-integriertem DFS-Namespace

Ich nutze DFS-Namespace, um meine Freigaben zu konsolidieren. Um die Verfügbarkeit auch standortübergreifend zu realisieren, hatte ich den Namespace in das Active Directory integriert. Somit konnte ich mehrere DFS-Namespace-server aufbauen, welche über das AD alle gleich konfiguriert waren.

Das Problem kam auf, als ich den bestehenden Namespace um eine Freigabe erweitern wollte. Die Konsole öffnete ich auf dem Fileserver mit meinem Account **Stephan-AD**, der auf den Fileservern lokal-administrative Rechte hat. Im AD hat er aber fast keine Rechte, denn die DomainController sind ja isoliert. Und anders herum hat der DomainAdmin zwar die Rechte im AD, aber nicht auf den Fileservern. Damit scheiterte die Erweiterung...

Die Lösung war aber recht einfach: ich nahm den Account **Stephan-AD** **temporär** in die Gruppe der **DomainAdmins** auf. Dadurch wurde die Administration der beiden Scopes kurzzeitig verbunden und ich konnte die Änderungen vornehmen.

Exchange-Server

Auch bei Exchange-Servern gibt es eine Abhängigkeit zum Active Directory – wenn man ohne das Split-Modell installiert hat (was der Normalfall ist). Damit hat der DomainAdmin in der Exchange-Infrastruktur alle Rechte, kann aber die Server mit den Exchange-Diensten nicht administrieren.

Meine aktuelle Lösung sieht so aus:

- Ich nutze den **DomainAdmin** für die Exchange-internen Angelegenheiten, also alle Powershell-Befehle und im ECP
- Für die Serveradministration nutze ich den Account **Stephan-AD**. Mit diesem kann ich mich um Eventlogs, Systemupdates und dergleichen kümmern.
- Bei einem Exchange-Update nehme ich den Account **Stephan-AD** in die Gruppe der DomainAdmins und ggf. auch in die Gruppen SchemaAdmins und EnterpriseAdmins auf – je nach Anforderung des CU-Setups.

Privilege Access Management

Da alle meine DomainController auf Windows Server 2016 laufen konnte ich die Funktionsebene der Gesamtstruktur auf Windows Server 2016 anheben und das optionale Feature „PAM“ aktivieren. Dieses Feature erlaubt die temporäre Gruppenmitgliedschaft. Damit kann ich also temporär die Scope-Administration verbinden:

- ich kann als berechtigte Person die administrative Kennung eines Scopes in die Sicherheitsgruppen eines anderen Scopes legen (z.B. den FileserverAdmin in die Gruppe der DomainAdmins aufnehmen → Scope DomainController)
- Dabei definiere ich eine TimeToLive in Minuten.
- Nach Ablauf der TTL nehmen alle DomainController den Benutzer aus der Gruppe.

Innerhalb der TTL kann ich dann Scope-übergreifende, administrative Tätigkeiten wahrnehmen. Das „vergessen“, einen Benutzer wieder aus der Gruppe herauszunehmen ist damit kein Problem mehr.

Fazit

SecurityScopes haben eine Ähnlichkeit zu Netzwerksegmenten: wir wissen heute, dass Server nicht im gleichen Netzwerk wie Clients stehen sollten. Server für die DMZ stellen wir ja auch separat. Zwischen diesen Segmenten verhindern Firewalls unerwünschte Kommunikationen. Und SecurityScopes setzen diese Aufgaben auf der AD-Autorisierungsebene um.

Der Sicherheitsgewinn wird den administrativen Mehraufwand durchaus rechtfertigen!