

# Theorie vragen BSIII - windows 2014

Andreas De Lille

Augustus 2014

# Inhoudsopgave

|          |   |          |
|----------|---|----------|
| <b>I</b> | <b>Modeling - Reeks 1</b>   | <b>1</b> |
| <b>1</b> | <b>Structuur van Active Directory gegevens</b>  | <b>2</b> |
| 1.1      | Bespreek de diverse namen die alle Active Directory objecten identificeren. (2.2.1)   | 2        |
| 1.2      | Wat zijn SPN objecten ? Bespreek de aanvullende naamgeving voor deze objecten. (2.2.2)  | 3        |
| 1.3      | Enkele veel gebruikte klassen vertonen nog "meer identificerende attributen voor hun instanties. Bespreek deze klassen en attributen.   | 5        |
| 1.4      | In welke partities is de Active Directory informatie verdeeld ? Geef de betekenis van elke partitie, hun onderlinge relatie, en de replicatiekarakteristieken ervan. (laatste helft 2.2.3)  | 6        |
| 1.4.1    | Wat?  | 6        |
| 1.4.2    | Partities   | 6        |
| 1.4.3    | Onderlinge relatie  | 7        |
| 1.4.4    | Replicatie  | 8        |
| <b>2</b> | <b>attributeSchema objecten (2.2.4 en 2.2.5)</b>  | <b>9</b> |
| 2.1      | Bespreek het doel en de werking van attributeSchema objecten. Hoe kunnen deze objecten het best geraadpleegd en gewijzigd worden ?  | 9        |
| 2.1.1    | Doel & werking  | 9        |
| 2.1.2    | Raadplegen & wijzigen   | 10       |
| 2.2      | Bespreek de diverse naamgevingen van attributeSchema objecten.  | 10       |
| 2.3      | Bespreek de belangrijkste kenmerken van attributeSchema objecten, en hoe die ingesteld kunnen worden.   | 11       |
| 2.4      | Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling ? (o.a. 2.2.7)   | 12       |
| 2.5      | Via welke attributen kun je de klasse van een willekeurig Active Directory object achterhalen ? Hoe moet je op zoek gaan naar alle objecten van een bepaalde klasse ? Illustreer aan de hand van relevante voorbeelden. (laatste paragraaf 2.2.6) | 13       |
| 2.5.1    | objectClass   | 13       |

|          |   |           |
|----------|---|-----------|
| 2.5.2    | objectCategory . . . . .  | 13        |
| 2.5.3    | Wat gebruiken? . . . . .  | 13        |
| <b>3</b> | <b>classSchema objecten (2.2.4 en 2.2.6)</b>  | <b>14</b> |
| 3.1      | Bespreek het doel en de werking van classSchema objecten. . . . .   | 14        |
| 3.2      | Hoe benadert Active Directory het mechanisme van overerving ? . . . . .   | 14        |
| 3.3      | Bespreek de diverse naamgevingen van classSchema objecten. . . . .  | 16        |
| 3.4      | Bespreek de belangrijkste kenmerken van classSchema objecten, en hoe die ingesteld kunnen worden. . . . .   | 16        |
| 3.5      | Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling ? (o.a. 2.2.7) . . . . .   | 17        |
| 3.6      | Hoe en met welke middelen kan het Active Directory schema uitgebreid worden ? (2.2.8, ldifde fractie 2.2.3) . . . . .   | 18        |
| <b>4</b> | <b>Active Directory functionele niveaus (2.4.3)</b>   | <b>19</b> |
| 4.1      | Geef de diverse functionele niveaus waarop Active Directory kan ingesteld worden, en welke beperkingen er het gevolg van zijn. . . . .  | 19        |
| 4.1.1    | Domein functioneel niveau . . . . .   | 19        |
| 4.1.2    | Forest functioneel niveau . . . . .   | 20        |
| 4.2      | Bespreek van elk niveau alle eraan gekoppelde voordelen. Geef hierbij telkens een korte bespreking (verspreid over de cursus !) van de ingevoerde begrippen. . . . .  | 20        |
| 4.2.1    | Domein functioneel niveau . . . . .   | 20        |
| 4.2.2    | Forest functioneel niveau . . . . .   | 21        |
| 4.3      | Hoe kan men detecteren op welk niveau een Active Directory omgeving zich bevindt ? . . . . .  | 22        |
| 4.3.1    | Domein functioneel niveau . . . . .   | 22        |
| 4.3.2    | Forest functioneel niveau . . . . .   | 22        |
| 4.4      | Op welke diverse manieren kan men het functionele niveau verhogen of verlagen ? . . . . .   | 23        |
| <b>5</b> | <b>Active Directory domeinstructuren (2.4.4 en 2.4.6)</b>   | <b>24</b> |
| 5.1      | Wat is de bedoeling van vertrouwensrelaties ? . . . . .   | 24        |
| 5.2      | Bespreek de verschillende soorten vertrouwensrelaties. . . . .  | 24        |
| 5.2.1    | expliciet . . . . .   | 24        |
| 5.2.2    | impliciet . . . . .   | 25        |
| 5.3      | Op welke diverse manieren kunnen vertrouwensrelaties gecreëerd en gecontroleerd worden ? Bespreek ook de optionele configuratiemogelijkheden. . . . .   | 26        |
| 5.4      | Welke verschillen zijn er in praktijk tussen NT 4.0 en Windows Server domeinstructuren ? Bespreek de alternatieve mogelijkheden bij de conversie van een NT 4.0 domeinstructuur naar een Windows Server omgeving. . . . . | 27        |
| <b>6</b> | <b>Active Directory server rollen (2.4.7, 2.3 en fractie 2.4.2)</b>   | <b>29</b> |
| 6.1      | Hoe bepaald wordt welke servers een dergelijke specifieke functie vervullen ? Hoeveel zijn er nodig (in termen van: minimaal/exact/maximaal aantal, in functie van ...), en waarom ? . . . . .                            | 30        |

|           |   |           |
|-----------|---|-----------|
| 6.2       | Eigenschappen zoals bedoeling, noodzaak, criticiteit, inhoud, synchronisatie, voor welke Windows versie(s) van toepassing, ... ?  | 30        |
| 6.3       | De eventuele relatie tussen de diverse functies. Vermeld bijvoorbeeld welke functies al dan niet door dezelfde server kunnen vervuld worden, of misschien wel juist wel door dezelfde server moeten vervuld worden.   | 30        |
| 6.4       | Op welke diverse manieren men de toewijzing van elke bijzondere functie kan instellen, wijzigen en/of ongedaan maken ?  | 30        |
| <b>II</b> | <b>Schriftelijk - Reeks 2</b>   | <b>35</b> |
| <b>7</b>  | <b>Active Directory replicatie (2.5)</b>  | <b>36</b> |
| 7.1       | Wat is de bedoeling van replicatie ?  | 36        |
| 7.2       | Hoe wordt dit in Windows Server (ondermeer ten opzichte van NT 4.0) gerealiseerd: bespreek de verschillende technische kenmerken en concepten van Windows Server replicatie, en hoe men specifieke problemen vermijdt of oplost.  | 37        |
| 7.2.1     | Windows NT4   | 37        |
| 7.2.2     | Winows Server   | 37        |
| 7.2.3     | Store-and-forward replicatie  | 39        |
| 7.3       | Welke toestellen repliceren onderling in een forest ? Welke specifieke gegevens worden hierbij uitgewisseld ?   | 40        |
| 7.4       | Welke impact hebben sites met betrekking tot de replicatie van Active Directory gegevens ? Je hoeft hierbij het begrip site op zich niet verder te behandelen. (eerste helft 2.6.1, verschil tussen intrasite en intersite replicatie)  | 40        |
| <b>8</b>  | <b>Active Directory sites (2.6)</b>   | <b>42</b> |
| 8.1       | Welke rol vervullen sites ? Welke Active Directory aspecten worden erdoor beïnvloed ? Bespreek hoe deze aspecten anders gerealiseerd worden indien de toestellen zich al dan niet in verschillende sites bevinden. (ondermeer verschil tussen intrasite en intersite replicatie)                            | 42        |
| 8.2       | Welke relaties bestaan er tussen sites, domeinen, domeincontrollers en global catalogs ? Druk deze relaties ondermeer uit in termen zoals ... een X vereist minimaal/exact/maximaal nr Y's ... . Geef een verantwoording voor elk van deze beweringen. Hoe wordt bepaald tot welke site computers behoren ? | 43        |
| 8.3       | Bespreek de diverse noodzakelijke instellingen om de verschillende aspecten van sites te configureren, en vermeld hierbij telkens: waar en in welke gegroepeerde vorm ze opgeslagen worden, waarom ze noodzakelijk zijn (ondermeer welke andere aspecten er afhankelijk van zijn).                          | 43        |
| <b>9</b>  | <b>Gedeelde mappen en NTFS</b>  | <b>46</b> |
| 9.1       | Welke manier om gedeelde mappen te creëren biedt de meeste configuratieinstellingen aan ? Bespreek het doel van deze diverse instellingen en de belangrijkste eigenschappen en mogelijkheden ervan. (3.2.1, 3.2.2, fracties 3.3.1, 3.4.2, 3.4.3, 3.5 en 3.6)  | 46        |

|           |   |           |
|-----------|---|-----------|
| 9.1.1     | Creatie . . . . .   | 46        |
| 9.1.2     | Configuratie . . . . .  | 47        |
| 9.1.3     | ACL . . . . .   | 48        |
| 9.1.4     | machtingen zijn hiërarchische gestructureerd en overerving is mogelijk  | 48        |
| 9.1.5     | Verschil tussen een security descriptor met lege ACL of zonder ACL .  | 49        |
| 9.1.6     | 2 niveaus voor machtingen . . . . .   | 49        |
| 9.2       | Waar wordt de definitie en (partiële) configuratie van gedeelde mappen opgeslagen ? Hoe kan men deze wijzigen vanuit een Command Prompt ? . . . . .   | 51        |
| 9.3       | Op welke diverse manieren kan men gebruik maken van gedeelde mappen ? (3.2.3) . . . . .   | 51        |
| 9.4       | Geef een overzicht van de belangrijkste voordelen van de opeenvolgende versies van het NTFS bestandssysteem. Bespreek elk van deze aspecten (ondermeer het doel, de voordelen en de beperkingen ervan), en geef aan hoe je er gebruik kan van maken, bij voorkeur vanuit een Command Prompt. (NTFS fractie 1.6, fracties 3.4.1, 3.4.2 en 3.4.4) . . . . . | 52        |
| 9.4.1     | Features vanaf v1.2 . . . . .   | 52        |
| 9.4.2     | Features vanaf V 3.0 . . . . .  | 53        |
| <b>10</b> | <b>Machtingen op bestandstoegang (3.3)</b>  | <b>54</b> |
| 10.1      | Welke rol spelen machtingen bij de beveiliging van bronnen ? Geef een gedetailleerd algemeen overzicht van het mechanisme van machtingen. . . . .   | 54        |
| 10.2      | Bespreek hoe het mechanisme van machtingen specifiek (en op diverse niveaus) toegepast wordt op bestandstoegang. Geef de verschillende soorten machtingen, hun onderlinge relaties, en hoe deze kunnen geanalyseerd en ingesteld worden. Toon hierbij aan dat je zelf met deze configuratietools geëxperimenteerd hebt. . . . .                           | 55        |
| 10.3      | Wat gebeurt er met de machtingen bij het verplaatsen van een bestand ?<br>Wat gebeurt er met de machtingen bij het kopiëren van een bestand ? . . .   | 58        |
| 10.4      | Op welke andere objecten zijn machtingen van toepassing ? . . . . .   | 58        |
| <b>11</b> | <b>Gebruikersgroepen (4.2.2 en 4.2.3)</b>   | <b>59</b> |
| 11.1      | Bespreek in detail het onderscheid tussen de diverse soorten veiligheidsgroepen, ondermeer afhankelijk of het toestel al dan niet in een domein is opgenomen. Behandel hierbij vooral de mogelijkheden en beperkingen. Bespreek ondermeer:  | 59        |
| 11.2      | Hoe en waarom worden deze soorten groepen in de praktijk best gebruikt, al dan niet gecombineerd ? Van welke omstandigheden is dit afhankelijk ? Illustreer aan de hand van concrete voorbeelden. . . . .   | 61        |
| 11.3      | Waar en hoe wordt het (volledige) lidmaatschap van een object tot een groep bijgehouden ? Op welke diverse manieren kan men dit lidmaatschap configureren ? Door wie wordt dit lidmaatschap bij voorkeur ingesteld ? Op welke diverse manieren kan men de volledige verzameling van objecten, die er deel van uitmaken, achterhalen ? . . . . .           | 62        |
| 11.4      | Welke conversieregels gelden er tussen de diverse soorten groepen. Behandel hierbij alle mogelijke combinaties. . . . .   | 62        |

Deel I

**Mondeling - Reeks 1**

# Hoofdstuk 1

## Structuur van Active Directory gegevens

### 1.1 Bespreek de diverse namen die alle Active Directory objecten identificeren. (2.2.1)

#### Naamgeving van object

De namen zijn logisch en hiërarchisch opgebouwd.

Volgende vier namen zijn steeds beschikbaar.

#### 1. RDN - Relative distinguished name

- Voorbeeld: cn = beelzebub
- Is uniek binnen zijn container.
- Denk aan absoluut path (DN) vs. relatief filepath (RDN).
- Wordt opgeslagen in het cn attribuut van het object.

#### 2. DN - Distinguished name

- Voorbeeld: cn = beelzebub, ou= iii, ou=hogent, ou=be (cn = common name, ou = organisational unit)
- Attributed naming, verschillende attribuut=waarde koppels
- Afgeleid van alle container object waarvan het object hiërarchisch deel uitmaakt.
- Uniek over het hele domein.
- Denk aan absoluut path (DN) vs. relatief filepath (RDN).
- Wordt opgeslagen in het distinguishedName attribuut van het object.

### 3. CN - canonieke naam

- !! Niet hetzelfde als de cn van hierboven. Hier cn = canonieke naam ; hierboven cn = common name als waarde van een DN)
- Voorbeeld: hogent.be/iii/beelzebub
- Samengesteld uit de DN, geeft de DN op een eenvoudigere manier weer.
- De meeste hulpmiddelen in active directory tonen de canonieke naam.
- Wordt opgeslagen in het canonicalName attribuut van het object. (en dus niet in het cn attribuut)

### 4. GUID - global unique identifier

- Globaal uniek (zelfs in tijd) getal van 128 bits.
- Kan en wordt nooit gewijzigd.
- Wordt opgeslagen in het objectGUID attribuut van het object.
- Wordt gegenereerd en toegewezen bij het aanmaken van het object.

## 1.2 Wat zijn SPN objecten ? Bespreek de aanvullende naamgeving voor deze objecten. (2.2.2)

### 1. SPN - Security Principal Objects

- Doel: SPN of Security Principal Objects zijn Active Directory objecten die gebruikt worden om toegang te verlenen tot domeinbronnen.
- Zijn van toepassing op computers, gebruikersrekeningen en domeinen.

### 2. SID - Security ID

- Zijn net als guids uniek in tijd; wanneer een object verwijderd en vervolgens terug aangemaakt wordt, zal het een andere SPN krijgen. Hierdoor kan een object nooit machtigingen van een oude account behouden.
- Opgeslagen in het objectSid kenmerk
- Men maakt gebruik van SIDs naast GUIDs om compatibiliteitsredenen.
- hiërarchische string getallen gescheiden door koppeltekens bijvoorbeeld S-1-5-x-y-z-500. Hierbij is S-1-5 een standaard prefix bestaande uit een revision level en een authority identifier. X,y en z zijn 32bit getallen die specifiek zin voor het domain, (Domain Subauthority Identifier), 500 is een relatieve ID (RID) dat naar het feitelijke object verwijst.



- SID blijft behouden als het object verplaatst wordt binnen hetzelfde domein. Als er verplaatst wordt naar een nieuw domein zal de SID wijzigen.
- Wordt gegenereerd en toegewezen bij de aanmaak van het object.
- sIDHistory, houdt alle SIDs bij die het SPN in het verleden had om te vermijden dat een gebruiker na verplaatsing van objecten zijn toegang zou verliezen.

### 3. UPN - User Principal Name

- Doel aanmeldingsnamen van gebruikers vereenvoudigen.
- opgeslagen in het userPrincipalName kenmerk.
- Als de UPN enkel gebruikt wordt voor aanmelding, moet hij uniek zijn binnen het volledige forest.
- Bestaat standaard uit [RDN gebruiker]@[UPN suffix] (zonder [ en ])
- UPN suffix kan vervangen worden door
  - DNS domeinnaam van het domein waar de account zich bevindt of het root domein
  - Mag zelfs een willekeurige naam zijn ook, als hij geregistreerd is met behulp van de Active Directory domeins and Trust snap-in.
- Wordt maar sporadisch gebruikt door compatibiliteitsredenen. Vaak maakt men gebruik van: [NetBIOSnaam van het domein]-[SAM accountnaam]. (zonder [ en ]).

### 4. NetBIOS

- Bestaat standaard uit de meest linkse component in de DNS naam van het domein
- Is niet langer dan 15 letters
- deze naam moet uniek zijn in zijn forest

### 5. SAM accountnaam - Security Accounts Manager

- Moet uniek zijn in het domein
- Wordt opgeslagen in sAMAccountName
- Bestaat uit hoogstens 20 karakters, standaard de eerste 20 bytes van de RDN afgesloten door een \$ <sup>1</sup>.
- Deze naam kan op elk gewenst moment veranderd worden.

---

<sup>1</sup>in de cursus staat er bytes p21, voorlaatste paragraaf, ik zou eerder denken dat het letters zijn

## 6. DNS hostname

- opgeslagen in dnsHostName kenmerk
- standaard eerste 15 bytes van de RDN gevuld door de suffix voor de primaire DNS
- Standaard is de suffix de volledige DNS naam van het domein waar de computer toe behoort.
- Er kan afgeweken worden; meer dan 15 chars en andere DNS naam.

## 1.3 Enkele veel gebruikte klassen vertonen nog ”meer identificerende attributen voor hun instanties. Bespreek deze klassen en attributen.

Komt later aan bod. zaken zoals:

1. LDAPDisplayName
2. Object identifier
3. objectClass (de hiërarchische klassen)
4. objectCategory (de categorie van de klasse van het object)
5. ...

## **1.4 In welke partities is de Active Directory informatie verdeeld ? Geef de betekenis van elke partitie, hun onderlinge relatie, en de replicatiekarakteristieken ervan. (laatste helft 2.2.3)**

### **1.4.1 Wat?**

We noemen de verzameling van alle active directory informatie (objecten en containerobjecten samen met hun meta data (ook objecten)) het gegevensarchief of de directory. Elke domein-controller bevat een kopie van de directory van zijn domein. De informatie is fysiek verdeeld in minimaal 3 categoriën of partities. Cliënt computer houden (uiteraard) geen informatie bij.

### **1.4.2 Partities**

#### **1. Domeinpartities met domeingegevens**

- bevatten informatie over objecten in het domein: gedeelde bronnen (servers, bestanden en printers) en accounts.
- Bij installatie worden er een aantal standaard objecten aangemaakt, een daarvan is de administrator account
- elk domein zit in een aparte partitie, er zijn dus evenveel partities met domeingegevens als dat er domeinen in het forest zijn.
- deze gegevens hebben bijgevolg enkel betrekking op dit domein en worden niet gedistribueerd naar ander domeinen.
- een subset van deze gegevens wordt opgeslagen in de global catalog

#### **2. Applicatie partities**

- bv dns gegevens
- kunnen geen SPN objecten bevatten
- kunnen niet verplaatst worden buiten de applicatie partitie
- beschikbaar vanaf windows server 2003
- zelf partities maken met adsiedit.msc

### 3. configuratie gegevens

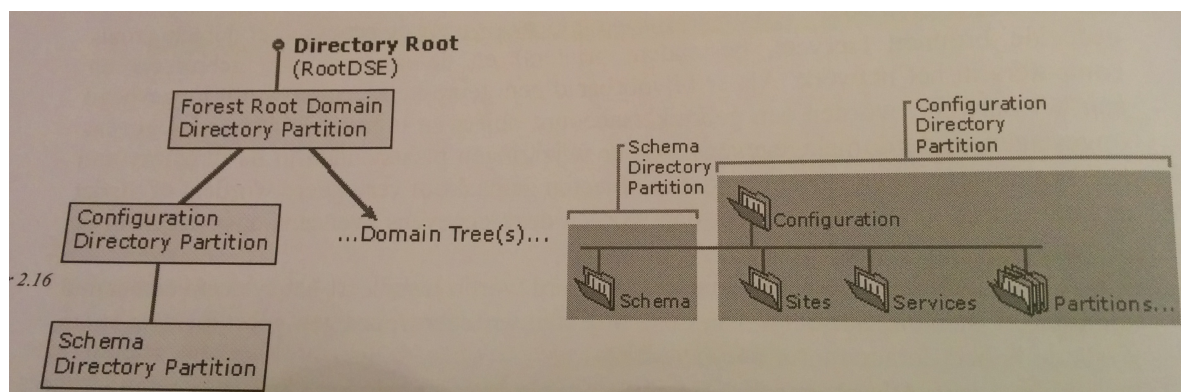
- beschrijven de fysieke topologie van de directory (bv lijst van alle domeinstructuren, locaties van domeincontrollers en global catalog controllers, sites, ..)
- Instellingen voor het hele forest worden vertaald naar kenmerken van objecten in de configuratie gegevens. (bv. uPNSuffixes kenmerk houdt de mogelijke UPN suffixen bij)

### 4. schema

- bevat een formele definitie van alle objecten en kenmerkgegevens die kunnen opgeslagen worden in de directory.
- is uniek voor alle domeinen in het forest.

#### 1.4.3 Onderlinge relatie

- Logische structuur ; boomstructuur



**Figuur 1.1:** onderlinge relatie van de partities, uit de cursus "Besturingssystemen III - Windows Server (J. Moreau)"

- Het forest root domein staat bovenaan en bevat de domein partities samen met de configuratie partitie
- partities kunnen deel uitmaken van een andere partitie, zo kan een domein partitie deel zijn van een hoger liggende domein partitie.
- De schema partitie is een onderdeel van de configuratie partitie
- Applicatie partities kunnen op 3 plaatsen toegevoegd worden
  1. als een afzonderlijke boom in het forest

2. als kind van een domein partitie
  3. als kind van een applicatie partitie
- Fysieke structuur: de schema partitie en de configuratie partitie zijn 2 verschillende entiteiten.

#### 1.4.4 Replicatie

- Elke partitie is een aparte eenheid voor replicatie.
- schema en configuratie gegeven worden gerepliceerd naar alle domeincontrollers in het forest.
- De domeingegevens van een bepaald domein worden gerepliceerd binnen het domein zelf.
- de applicatie partities worden uitgewisseld met een eigen deelverzameling specifiek geconfigureerde domeincontrollers van het forest, onafhankelijk van de domein grenzen. (bv dns gegevens enkel syncen met dns servers)
- een subset van de kenmerken van alle objecten in de domeingegevens van elk domein in het forest worden gerepliceerd naar de globale catalogus.

## Hoofdstuk 2

# attributeSchema objecten (2.2.4 en 2.2.5)

Er zijn verschillende soorten schema's

- **Active directory schema - reële schema** : volledige schema dat de regels van klassen en objecten bevat. bevat 2 soorten definities:
  - attributeSchema objecten: kenmerken, elk kenmerk wordt 1 keer gedefinieerd en wordt daarna gebruikt voor meerdere klassen.
  - classSchema objecten: klassen, de klassen die gemaakt kunnen worden.
- **abstracte schema**: compacte representatie van het gehele schema

### 2.1 Bespreek het doel en de werking van attributeSchema objecten. Hoe kunnen deze objecten het best geraadpleegd en gewijzigd worden ?

#### 2.1.1 Doel & werking

- Kenmerken van klassen zijn zelf objecten in het schema
- beperkingen opleggen
- worden beheerd net als andere objecten
- een kenmerk kan in meerdere klassen hergebruikt worden

### 2.1.2 Raadplegen & wijzigen

- dsquery (tonen)
- via adsiedit.msc
- zelf gemaakte scripts
- ldifde csvde
- verwijderen van items is niet mogelijk, wel isDefunct op true zetten, zodat ze niet meer aangemaakt kunnen worden. Voordeel hiervan is dat ongedaan maken van een (foutieve verwijdering) eenvoudig is.

## 2.2 Bespreek de diverse naamgevingen van attributeSchema objecten.

Voor elk object is ook een viervoudige naamgeving aanwezig.

### 1. CN - Common name

- Niets anders dan de RDN van het attributeSchema object in de schema container.
- bijgehouden in het cn attribuut

### 2. GUID van een kenmerk

- Onafhankelijk van het GUID van een attributeSchema object (duh)
- automatisch gegenereerd indien gewenst
- uniek binnen het forest
- bijgehouden in het schemaIDGUID attribuut

### 3. LDAP display name

- belangrijk voor programmatische toegang
- bijgehouden in het LDAPDisplayName attribuut

### 4. OID - object identifier

- interne representatie
- x.500 ids worden verleend door speciale autoriteiten zoals ITU ANSI en ISO en zijn gegarandeerd uniek in alle netwerken over de hele wereld.
- je kan een tak aanvragen of een unieke genereren in de ms subtak met behulp van de oidgen
- bijgehouden in het attributeID attribuut.

## 2.3 Bespreek de belangrijkste kenmerken van attributeSchema objecten, en hoe die ingesteld kunnen worden.

De 7 belangrijke kenmerken zijn

### 1. **attributeSyntax & oMSyntax**

- bepaald het data type (26 mogelijkheden waarvan 18 in gebruik, bv boolean, integer)
- het is niet mogelijk om nieuwe syntax te definiëren.
- de oMSyntax wordt gebruikt om een bijkomend onderscheid te maken omdat de attributeSyntax alleen niet genoeg blijkt te zijn.

### 2. **rangeLower en rangeUpper**: bereikbeperkingen van een kenmerken

### 3. **isSingleValued** : Of het object een over meerdere waarden heeft

### 4. **searchFlags** : binaire nformatie waarbij de meeste bits bepalen of het kenmerk op een of andere manier geïndexeerd wordt. Indien het kenmerk geïndexeerd is, kan er sneller gezocht worden op dat kenmerk.

- laagste bit: eenvoudige indexering van de waarde van het kenmerk
- tweede laagste bit: waarde van het kenmerk combineren met de identificatie van de container. Dergelijke containerized indexen kunnen snel alle objecten binnen een specifieke container opsporen.
- derde laagste bit: ambiguous name resolution toelaten. Zoeken waarbij minstens een kenmerk uit een verzameling kenmerken een specifieke waarde aanneemt.
- zesde laagste bit; versnellen van opzoekingen waarin kenmerken met jokertekens vermeld worden. deze tuple indexen vergen heel wat resources en worden best in beperkte mate gebruikt.
- vijfde laagste bit: heeft niets met indexing te maken, maar bepaald of de waarde van het attribuut behouden blijft indien men een kopie maakt van het object.

### 5. **systemFlags** Bevat ook binaire informatie

- de laagste bit bepaald of het kenmerk al dan niet gerepliceerd wordt naar andere domeincontrollers. Niet gerepliceerde kenmerken worden gebruikt voor caching of gebruikt bij relatief dynamische kenmerken waarvan de waarde grequent wijzigt zoals lastLogion en LAstLogoff.
- het derde laagste bit van systemFlags wijst op een geconstrueerd attribuut; een attribuut dat telkens opnieuw berekend wordt.



6. **isMemberOfPartialAttributeSet**: bepaald of het kenmerk in de global catalog opgenomen wordt of niet.

7. **linkID**

- Sommige kenmerken vormen koppels bestaande uit forward-link en back-link kenmerken
- De referentiële integriteit te garanderen.
- Enkel de forward link kan aangepast worden, de backlink wordt beheerd door het systeem.
- gebruik door de overeenkomstige attributen van de kenmerken op te vullen met opeenvolgende even en oneven gehele getallen.

## 2.4 Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling ? (o.a. 2.2.7)

1. **classSchema-objecten** Groepen de attributen per klasse en geven dus aan welke klassen er gemaakt kunnen worden.

2. **het abstracte schema**

- abstracte schema
- vereenvoudige interface aan LDAP cliënts door het verbergen van overbodige implementatie details.
- RDN = Aggregate
- high level toegang via ADSI interfaces.
- geeft beperkt aantal kenmerken aan van het class- en attributeSchema.
- kan heel wat werk besparen.

## **2.5 Via welke attributen kun je de klasse van een willekeurig Active Directory object achterhalen ? Hoe moet je op zoek gaan naar alle objecten van een bepaalde klasse ? Illustreer aan de hand van relevante voorbeelden. (laatste paragraaf 2.2.6)**

Hiervoor kan men gebruik maken van 2 mandatory-kenmerken van de top klasse. Deze zijn, doordat ze mandatory zijn in de topklasse, verplicht aanwezig in elk object.

### **2.5.1 objectClass**

Is een multi valued en niet geïndexeerd attribuut dat alle hiërarchische superklassen (op de statische hulpklassen na) bevat.

### **2.5.2 objectCategory**

Is een single valued en geïndexeerd attribuut dat de meest typische vertegenwoordiger uit de verzameling bestaande uit de klasse zelf en alle hiërarchische superklassen.

### **2.5.3 Wat gebruiken?**

- Als de objectCategory is ingesteld met de klasse van het object is dit natuurlijk de beste keuze. De opzoeking laat toe om een beroep te doen op indexeren, wat een stuk performanter is.
- objectCategory ingesteld op een hogere klasse: problemen: we krijgen ook andere deelklassen. Het beste is om eerst de hogere klasse op te halen en dan deze kleinere lijst nogmaals filteren.
- alleen de objectClass selecteren is het traagste vermits er niet geïndexeerd wordt.

## Hoofdstuk 3

# classSchema objecten (2.2.4 en 2.2.6)

### 3.1 Bespreek het doel en de werking van classSchema objecten.

#### 1. doel & werking

- doel: definiëren hoe een klasse er uit ziet.
- voor elke klasse is er een object in het classSchema
- 2 soorten regels
  - (a) structuurregels: hiërarchische relaties tussen de klassen
  - (b) inhoudsregels: kenmerken definiëren die de klasse moet hebben (verwijzen naar attributen van hierboven)
- worden ook opgeslagen als objecten.

### 3.2 Hoe benadert Active Directory het mechanisme van overerving ?

#### 1. basis overerving

- elke klasse, behalve de TOP klasse is afgeleid van een andere klasse
- superklasse of bovenliggende klasse
- subklassen
- overerving: overnemen van alle kenmerken; structuur- en inhoudsregels van de bovenliggende klasse. (!! kenmerken GUID overnemen, maar daarom wordt de waarde van GUID nog niet overgenomen)

- overname is recursief, alle kenmerken van alle bovenliggende klassen overnemen

## 2. meervoudige overerving

- enkel kenmerken overnemen van onmiddellijke superklasse en speciaal bestemde hulpklassen. Deze hulpklassen kunnen zelf geen klassen genereren.
- hulpklassen kunnen zowel dynamisch als statisch gebruikt worden

## 3. statisch gebruik van hulpklassen

- `auxilairyClass` of `systemAuxiliaryClass`
- Wordt vastgelegd in de definitie van de klasse
- statisch en kan niet veranderd worden zonder de definitie van de klasse aan te passen.
- in de definitie van de klasse wordt reeds vastgelegd van welke hulpklassen deze klasse kenmerken kan overnemen. Alle objecten hebben dit kenmerk dus vanzelf.

## 4. dynamisch gebruik van hulpklassen

- beschikbaar vanaf windows server 2003
- `entryTTL` als optioneel attribuut dat gebruikt kan worden om de klasse vanzelf te laten vervallen
- `entryTTL` kan opgefrist worden
- dynamisch kenmerken bijladen door het `objectClass` kenmerken aan te vullen met de klassenaam. Hierdoor erft enkel deze instantie de kenmerken van de hulpklasse
- dit moet wel bij de create van de instantie ingesteld worden vermits het `objectClass` kenmerk na creatie niet meer gewijzigd kan worden.

Of een klasse een structureel, abstracte of hulpklasse is, is af te leiden uit de `objectClassCategory` integer. de waarde van de ints zijn respectievelijk 1, 0 of 2 , en 3. Abstracte klasse kunnen ook geen objecten maken en worden gebruikt om objecten te verzamelen bv `aPersoon` . een abstracte klasse kan enkel een abstracte klasse als superklasse hebben.

### 3.3 Bespreek de diverse naamgevingen van *classSchema* objecten.

Er is voor elke object ook een viervoudige naamgeving:

1. common name
2. GUID
3. ldap displayname
4. object ID

Voor meer informatie kijk bij reeks1 vraag2, daar worden de verschillende namen al uitgelegd.

### 3.4 Bespreek de belangrijkste kenmerken van *classSchema* objecten, en hoe die ingesteld kunnen worden.

2 soorten regels. Als de naam begint met *system* wordt het kenmerk beheerd door het systeem zelf.

1. **Inhoudsregels:** definiëren welke kenmerken een klasse heeft.
  - **postSuperiors & systemPostSuperiors** lijst van verplichte kenmerken. Deze zullen na overerving ook altijd verplicht blijven ook al zijn ze in de klasse zelf als optioneel gemarkeerd.
  - **mayContain & systemMayContain** lijst van optionele kenmerken
  - **rDNAttID** bepaald welk kenmerk van een klasse gebruikt wordt om de RDN van objecten te bepalen. Voor de meeste klassen staat dit kenmerk ingesteld op de waarde.
  - **defaultSecurityDescriptor** bepaalt expliciete machtigingen die gelden voor objecten van deze klassen. Dit kan gebruikt worden om het beheer van deze klasse te delegeren.
  - **systemOnly** indien dit kenmerk de waarde *true* heeft kunnen de structuurregels en inhoudsregels niet gewijzigd worden.
  - **isDefunct** kan gebruik worden om klassen te deactiveren (verwijderen van klassen is niet mogelijk). Doordat klassen enkel gedeactiveerd kunnen worden is het ongedaan maken ervan veel simpeler.

2. **Structuur regels** : Definiëren de mogelijke hiërarchische relaties tussen klassen of objecten.

- **objectClassCategory** : categorie van de klasse bepalen.
  - 1 : structurele klasse
  - 0 of 2 : abstracte klasse
  - 3 hulpklasse
- **objectClass** multivalued niet geïndexeerd. Alle bovenliggende klassen
- **objectCategory**: single valued en geïndexeerd. De meest typische vertegenwoordiger.
- **auxiliaryClass** en **systemAuxiliaryClass** welke hulpklassen de klasse heeft
- **poosSuperiors** en **systemPossSuperiors** bepaald welke andere objecten de klasse kunnen bevatten. Bijvoorbeeld een organizationalUnit kan bijvoorbeeld container zijn voor o.a. user-objecten, dit betekent dat het classSchema-object voor user verwijst naar organizationalUnit.

### 3.5 Welke andere types objecten bevat het Active Directory schema, en wat is hun bedoeling ? (o.a. 2.2.7)

Naast het classSchema-objecten zijn er nog 2 andere beschikbaar:

1. **attributeSchema-objecten** waar de attributen gedefinieerd worden. zie hierboven ergens.
2. **abstract schema objecten** een compacte representatie van het reële schema, zie hierboven.

### 3.6 Hoe en met welke middelen kan het Active Directory schema uitgebreid worden ? (2.2.8, ldifde fractie 2.2.3)

#### risicovol

- risico vol
- schema uitbreiden = wijzigingen geldig in heel het forest
- maak zoveel mogelijk gebruik van overerving om problemen te vermijden en maak dus enkel een geheel nieuwe structurele klasse aan als er geen enkel bestaande klasse voldoet.
- veel potentieel / veel mogelijkheden
- uiteraard kunnen enkel gemachtigde gebruikers wijzigingen doorvoeren
- best een testomgeving gebruiken

#### textbftools

##### 1. ldifde en csvde

- via command prompt
- grootschalige wijzigingen
- uitwisseling gebaseerd op intermediaire tekstbestanden in:
  - ldifde= LDAP Data Interchange Format
  - csvde = comman seperated value

##### 2. LDAP of ADSI interface Wijzigingen doorvoeren via ADSIEdit.msc of eigen scripts.

##### 3. Active Directory schema snap-in

- standaard niet in MMC (Microsoft MAnagement console).
- klassen en attributen in verschillende mappen weergegeven
- na het selecteren van een klasse in het linkerpaneel krijg je een detail overzicht
- dubbelklikken op een kenmerk laat toe om wijzigingen aan te brengen via een dialoog venster.

## Hoofdstuk 4

# Active Directory functionele niveaus (2.4.3)

### 4.1 Geef de diverse functionele niveaus waarop Active Directory kan ingesteld worden, en welke beperkingen er het gevolg van zijn.

#### 4.1.1 Domein functioneel niveau

Het domein functioneel niveau geeft aan welke minimum eis er gesteld wordt aan het besturingssysteem van de domeincontrollers en bepaalt tegelijkertijd welk faciliteiten er beschikbaar zijn. Dit niveau wordt opgeslagen in 2 attributen = ntMixedDomain en msDS-Behavior-Version. Er zijn vier mogelijkheden

##### 1. **Windows 2000 mixed**

- de laagste active directory functionaliteit
- windows 2008+ domeincontrollers niet mogelijk
- standaard gebruikt bij windows server 2000 en 2003

##### 2. **Windows 2000 native** enkel windows server NT5+ domein controllers, werkposten en lidserveren mogen lager zijn

##### 3. **Windows Server 2003** enkel windows server 2003+ domein controllers, werkposten en lidserveren mogen lager zijn.

##### 4. **Windows Server 2008** enkel windows server 2008+ domein controllers, werkposten en lidserveren mogen lager zijn.



### 4.1.2 Forest functioneel niveau

Analoog aan het domein functioneel niveau is er ook een forest functioneel niveau. Dit functioneel niveau bepaald het minimale niveau van een alle domeinen binnen een forest. Dit wordt opgeslagen in het msDS-Behavior-Version attribuut, maar dan van het partitions containerobject van de configuratie gegevens. Er zijn 3 mogelijkheden:

1. **windows 2000 forest** geen enkele eis aan het functioneel niveau van de liddomeinen.  
De standaard instelling
2. **Windows Server 2003** enkel domeinen van 2003+ niveaus
3. **Windows Server 2008** enkel domeinen van 2008+

## 4.2 Bespreek van elk niveau alle eraan gekoppelde voordelen. Geef hierbij telkens een korte bespreking (verspreid over de cursus !) van de ingevoerde begrippen.

### 4.2.1 Domein functioneel niveau

#### Windows 2000 native

- 1 globale catalog voor het ganse forest
- transitieve vertrouwensrelaties tussen verschillende domeinen van eenzelfde forest
- alle domeincontrollers kunnen zelfstandig een aantal SPN objecten aanmaken door de delegering van de RID master, waar er bij het mixed domein steeds beroep moet doen op de PDC emulator master van een domein.
- ruimere mogelijkheden voor configuratie van groepen
- alle sids die een SPN object in het verleden gehad heeft, worden bijgehouden in het SIDHistory kenmerk.

### **Windows Server 2003 niveau**

- gebruik van aanvullende schema klassen en attributen
- naam van een domeincontroller kan eenvoudiger veranderd worden (geen degradatie en promotie meer nodig)
- dagvullende opdrachten: rdirusr en redircmp om de default active directories te wijzigen waarin respectievelijk nieuwe gebruikers en nieuwe computers terecht komen.
- caching op domeincontroller niveau van UPN suffices en het lidmaatschap van universele groepen zodat het niet meer strikt noodzakelijk is dat tijdens het inlogproces een global catalog bereikbaar is
- group policies filteren ook met behulp van WMI scripts.

### **Windows Server 2008**

- aanvullende schema klassen en attributen
- encryptie van het kerberos protocol
- fijnkorrelig wachtwoord beleid
- replicatie van DFS namespaces en SYSVOL share via DFS replicatie (performanter van file replication service)

### **4.2.2 Forest functioneel niveau**

#### **Windows 2000**

- geen enkele eis aan het functionele niveau van de liddomeinen

### **Windows Server 2003**

- hergebruiken van gedeactiveerde attributen en klassen
- dynamische hulpklassen
- dynamische objecten met een beperkte levensduur
- efficiëntere replicatie van de global catalog gegevens, waardoor ondermeer toevoeging van een nieuw kenmerk niet leidt tot een volledige synchronisatie van alle objectkenmerken.
- veranderen van naamgeving en de hiërarchische structuur van domeinen in een forest.
- transitieve vertrouwens relaties tussen verschillende forests
- read only windows server 2008+ domeincontrollers
- efficiëntere KCC algoritmen voor de constructie van de replicatietopologie.
- replicatie van individuele waarden van multi-valued attributen zodat bijvoorbeeld bij verandering van het lidmaatschap van een groep niet de volledige verzameling leden opnieuw moet gesynct worden.

### **Windows Server 2008**

- geen aanvullende functionaliteiten
- wel betere beveiliging

## **4.3 Hoe kan men detecteren op welk niveau een Active Directory omgeving zich bevindt ?**

### **4.3.1 Domein functioneel niveau**

- opgeslagen in 2 attributen
  1. ntMixedDomain
  2. msDS-Behavior-Version
- ingesteld op het domeinobject zelf (bv. dc=iii, dc= hogent , dc = be)

### **4.3.2 Forest functioneel niveau**

- ingesteld op het partitions containerobject van de configuratiegegevens
- wordt bepaald door 1 attribuut: msDS-Behavior-Version

#### 4.4 Op welke diverse manieren kan men het functionele niveau verhogen of verlagen ?

- gebeurd niet automatisch, moet manueel gebeuren
- Kan op 2 manieren
  1. zelf de attributen manipuleren
  2. de Active Directory Domains and Trust snap-in
- Als niet alle voorwaarden voldaan zijn wordt je hiervan op de hoogte gebracht
- verlagen is niet mogelijk
- alle domeincontrollers moeten opnieuw opgestart worden om de wijzigingen door te voeren.
- als je weet dat er geen oudere controllers zijn kun je best het niveau al updaten bij de installatie van de eerste active directory controller zodat alle controllers die erna toegevoegd worden vanzelf het juiste niveau hebben.

## Hoofdstuk 5

# Active Directory domeinstructuren (2.4.4 en 2.4.6)

### 5.1 Wat is de bedoeling van vertrouwensrelaties ?

- Gebruikers van 1 trusted domein ook vertrouwen/ kunnen verifiëren in een ander trusting domein.
- weergegeven et een pijl in de richting van het trusted domein. (Als domein A domein B vertrouwd, is er een pijl van A naar B)
- eenmaal geverifieerd moet er gekeken worden naar de rechten en machtigingen van een gebruiker alvorens hij toegang krijgt tot het andere domein. Deze machtigingen worden per domein toegewezen.

### 5.2 Bespreek de verschillende soorten vertrouwensrelaties.

#### 5.2.1 expliciet

Deze moeten manueel aangemaakt worden

#### forest trusts

- windows server 2003+ (2003 of hoger) nodig
- manueel tussen de rootdomeinen van de forests
- directionele
- transitief
- realms bestaande uit meer dan 2 forests? voor elk koppel een trust maken

### Realm trusts

- veralgemening van forest trust
- tussen windows server 2008+ en willekeurige kerberos v5 realms onafhankelijk van het besturingssysteem waarop die geïmplementeerd zijn.
- bidirectioneel / enkelvoudig
- transitief / niet-transitief

### Verkorte vertrouwensrelaties

- worden gebruikt om het vertrouwenspad in grote en complexe trees korter te maken.
- performanter
- ook shortcut of cross-link genoemd
- in praktijk pas zinvol als het vertrouwens pad 5 of meer domeinen overspant.
- enkelvoudig/bi-directioneel

### Externe vertrouwensrelaties

- een domein vertrouwd het andere
- niet transitief
- altijd enkelvoudig, wil je bi-directioneel dan moet je 2 relaties maken.
- kan niet binnen hetzelfde forest
- bedoeld voor communicatie met externe partners
- met oude NT4 domeinen

#### 5.2.2 impliciet

Deze worden automatisch aangemaakt en beheerd bv tussen rootdomein en de subdomeinen

### 5.3 Op welke diverse manieren kunnen vertrouwensrelaties gecreëerd en gecontroleerd worden ? Bespreek ook de optionele configuratiemogelijkheden.

Enkel de expliciete vertrouwensrelaties moeten zelf geconfigureerd worden.

#### 1. Active Directory and Domains Trust snap)in

- beschikbaar via domain.msc
- rechtermuisknop op domein, properties, trusts-tab, new trusts wizard.
- aan elke vertrouwensrelatie wordt een wachtwoord toegewezen
- aanvullende config is mogelijk en aangeraden vanuit het beveiligingsstandpunt
  - (a) Standaard worden alle gebruikers van het trusted domein opgenomen in de authenticated users impliciete groep van het trusting domein.
  - (b) men kan echter ook voor selective authentication kiezen waardoor dit per individuele gebruiker of gebruikersgroep expliciet moet ingesteld worden.
  - (c) inden men gebruik maakt van SID Filtering (de standaard instelling), dan wordt enkel rekening gehouden met de SID opgeslagen in het objectSid attribuut van de objecten in het trusted domein (en bijgevolg met SIDs waarvan de domain subauthority identifier zeker overeenkomt met die van het trusted domein.
  - (d) indein men SID Filtering uitschakeld, dan verwerkt het trusting domein ook de SIDs opgeslagen in het siDHistory attribuut. Malafide beheerder in het trusted domein met olledige toegang tot het siDHistory attribuut van de objecten in hun eigen domein kunnen langs deze weg zichzelf meer machtigingen en rechten toe-eigenen in het trusting domein.

#### 2. Via command line

- netdom trust : nieuwe relaties toevoegen
- netdom query trust : huidige vertrouwensrelaties opvragen / query'n

## 5.4 Welke verschillen zijn er in praktijk tussen NT 4.0 en Windows Server domeinstructuren ? Bespreek de alternatieve mogelijkheden bij de conversie van een NT 4.0 domeinstructuur naar een Windows Server omgeving.

- NT4 maakt een onderscheid in master domeinen en resource domeinen
  - Master domein of account domein bevat de gebruikers en groepen
  - resource domein biedt bronnen aan zoals printers, shared , ...
  - NT4 domeinstructuren bestaan uit een of meerder master domeinen en meerder resource domeinen. Er wordt een bi-directionele vertrouwensrelatie aangemaakt tussen all masterdomeinen onderling. Daarnaast zijn er enkelvoudige vertrouwensrelaties waarbij elk resource domein elk masterdomein vertrouwd.
- omschakeling moet geleidelijk en evolutionair gebeuren ipv revolutionair.
- windows server kan ervoor zorgen dat het aantal domeinen vermindert, wat het beheer zal vereenvoudigen
  - gebruik maken van ou om domeinen te vervangen
  - oud hebben als bijkomend voordeel dat het verplaatsen van objecten veel makkelijker is.
- beginnen in de root en dan naar beneden werken. eerst het master NT4 domein en dan de resource domeinen. Dit moet zo gebeuren omdat windows server altijd een root domein nodig heeft om te kunnen functioneren.
- indien bedrijfseenheden als aparte organisaties moeten behandeld worden is een forest met aparte trees een zeer goeie oplossing. Elke groep beheerders kan dan een eigen beveiligingsbeleid instellen dat onafhankelijk is van het beleid in andere domeinen. Daarbij worden gebruikers wel best verplaatst naar de domeinen met de bronnen die ze het meeste gebruiken.



- indien er meerdere master domeinen waren , dan was dat om een van volgende redenen:
  - **het netwerk is te groot , een SAM database groter dan 40MB is niet stabiel** Dit is opgelost in windows server, hier neemt een database van 1000 000 gebruikers 20GB in, wat moeiteloos ondersteun kan worden met de huidige database technologieën.
  - **het netwerk heeft verschillende geografische locaties**, aan elkaar gekoppeld door trage verbindingen waarover geen massaal replicatie verkeer tussen domeincontrollers gewenst is. Ook opgelost, zie hierboven.
  - **men wil een specifiek wachtwoord beleid voor verschillende groepen gebruikers** Dit kan opgelost worden door de fine-grained password policies van Windows Server 2008.
  - **het domeinmodel weerspiegelt de organisatie** waarin verschillende bedrijfs-eenheden controle moeten hebben over hun eigen gebruikers en bronnen. Dit is het enige argument om de aparte domeinen in de verschillende sites te behouden. Hierbij krijg je een root domein dat enkel een structural of placeholder domain is.

## Hoofdstuk 6

# Active Directory server rollen (2.4.7, 2.3 en fractie 2.4.2)

Welke vragen moet men zich stellen na de initiële installatie van een Windows Server toestel, in verband met bijzondere functies die de server kan vervullen met betrekking tot Active Directory ? Formuleer bij het beantwoorden van deze vragen telkens (voor zover relevant):

- 6.1 Hoe bepaald wordt welke servers een dergelijke specifieke functie vervullen ? Hoeveel zijn er nodig (in termen van: minimaal/exact/maximaal aantal, in functie van ...), en waarom ?
- 6.2 Eigenschappen zoals bedoeling, noodzaak, criticiteit, inhoud, synchronisatie, voor welke Windows versie(s) van toepassing, ... ?
- 6.3 De eventuele relatie tussen de diverse functies. Vermeld bijvoorbeeld welke functies al dan niet door dezelfde server kunnen vervuld worden, of misschien wel juist wel door dezelfde server moeten vervuld worden.
- 6.4 Op welke diverse manieren men de toewijzing van elke bijzondere functie kan instellen, wijzigen en/of ongedaan maken ?

**Antwoord:** Telkens men een nieuw Windows Server toestel aan het netwerk toevoegt, moet men zich na de initiële installatie ervan enkele cruciale vragen stellen, in het bijzonder in verband met de rol die de server zal vervullen met betrekking tot Active Directory.

**1. Wordt de server al dan niet opgenomen in het domein?**

- een computer waar windows server op draait, maar die geen lid is van een werkgroep of domein wordt een zelfstandige server genoemd. Deze servers kunnen wel bronnen delen binnen het netwerk, maar profiteren niet van de vele voordelen die active directory biedt.
- Om toch te kunnen genieten van deze voordelen besluit men meestal om deze server lid te maken van het domein, hij wordt dan een lidserver genoemd.

**2. Vervult de in een domein opgenomen server al dan niet de functie van domeincontroller?**

- Indien hij de rol van domeincontroller niet vervult, wordt hij een lid- / member-server genoemd.
- een lidserver is niet betrokken bij de replicatie en zal bijgevolg dus ook geen zaken zoals inloggen, beleidinfo of beveiliging voorzien en/of afhandelen.
- meestal zijn die file servers, toepassings servers, database servers, webservers, firewall, routers .. Deze functies worden gegroepeerd in 3 niveaus:
  - (a) **server rollen:** DHCP servers, dns servers, network policy and access services en web servers implementeren primaire serverfuncties.
  - (b) **role services:** complexere servere rollen die optionele componenten bieden. Voorbeelden hiervan zijn windows en unix wachtwoorden synchronisatie.
  - (c) **Features:** Group policy management , powershell , windows server backup features en snmp services die voor meer ondersteunende functies zorgen.
- Dit is configureerbaar via de Add Roles , Add Role Services en Add Features wizards van de Server Manager.
- lidserver blijven een eigen lokale beveiligingsdatabank de Security Account Manager (SAM) behouden. zowel gebruikers die in de AD gedefinieerd zijn als gebruikers die louter in de lokale SAM bestaan , kunnen, gefilterd door het mechanisme van machtigingen en gebruikersrechten , gebruik maken van de faciliteiten van een lidserver.
- men moet er rekening mee houden dat de promotie tot domein controller een aanzienlijke belasting met zich meebrengt.
- ten minste 1 domeincontroller in elke site om prestatie te verbeteren

**3. Als er gekozen wordt voor een domeincontroller, moet ook de functie van globale catalogus ondersteund worden?**

- om replicatie verkeer te beperken wordt er best een globale catalogus per site ingesteld.
- veel controllers = veel replicatie verkeer vs. geen controller in de site: zeer traag.
- Indien er slechts een domein in het forest is, is er geen enkel bezwaar om van alle domeincontrollers een globale catalogus te maken.

**4. Welke domeincontrollers vervullen deze operations master rollen**

- **alle domeincontrollers van een domein zijn nagenoeg equivalent** Een aantal specifieke AD functies, operations master rollen genoemd, kunnen echter slechts door een enkele controller in het domein of het forest vervuld worden.
- **OM rollen Uniek in elk domein**
  - (a) **RID master**
    - een dc maakt SPN objecten aan en gebruikt daarvoor SIDs die hij uit een SIDS pool haalt.
    - indien de SID pool voor 80% gebruikt is zal de dc een nieuwe SID pool aanvragen bij de RID master.
    - De pool bestaat uit 512 RIDs - Relative IDs.
  - (b) **PDC emulator master**
    - volledige emulatie van een NT4 primaire controller in een windows 2000 mixed domain met NT4 backup dc. Volledig transparant voor nt4 gebruikers en users.
    - enkel relevant in een windows 2000 mixed domain met NT4 backup DC's.
    - PDC emulator master krijgt voorgang bij de replicatie van wachtwoord wijzigingen. Als een wachtwoord veranderd wordt, is dit niet meteen op elke dc aanwezig, daarom zal een client bij een afwijzing ook de pdc contacteren vermits deze direct gesynchroniseerd zou moeten zijn.
    - is ook primaire bron voor tijdssynchronisatie.
    - best de RID en PDC door dezelfde controller laten vervullen.

(c) **Infrastructure master**

- Verantwoordelijk voor het bijwerken van verwijzingen vanuit objecten in het eigen domein naar objecten in andere domeinen.
- zie ook forward link, back link, phantom objects
- indien een forward link wijst naar een object in een ander domein, kan het backlink kenmerk van de object niet rechtstreeks aangepast worden. AD lost dit op door in beide domeinen een phantom object te creëren dat doorverwijst naar de DN, GUID en SID van de respectievelijke objecten in de andere domeinen. De infrastructure master van een domein vergelijk continu de kenmerken van zijn phantom objecten met de kenmerken van de corresponderende objecten in externe domeinen, en de kenmerken aan phantom objecten in externe domeinen die doorverwijzen naar eigen objecten met de kenmerken van de objecten. (kijkt dus naar zijn eigen phantoms vs ander domein en vice versa).
- De infrastructure master zal de gegevens uiteindelijk bewerken door de global catalogus te contacteren.
- de die ook global catalog server zijn beschikken al over een kopie van de objecten van andere domeinen en zullen dus geen phantom objecten aanmaken.
- de rol van infrastructure master moet dus ook vervuld worden door een server die geen global catalog is. Anders zijn er nooit verouderde objecten op het systeem. Indien alle controllers global catalog zijn, hebben ze allemaal een globale catalogus en is de rol niet meer van belang.

• **OM rollen die uniek moeten zijn binnen het forest**

- **Schema master** beheert alle bijgewerkte en gewijzigde gegeven voor het schema.
- 1 server mag die maar doen om conflicten te vermijden
- tijdelijk verlies van de schema master is niet merkbaar, tenzij er wijzigingen moeten doorgevoerd worden.

• **Domain naming master**

- beheert het toevoegen en verwijderen van domeinen en applicatie partities in het forest.
- het is de enige DC die de partitions container van de configuratiegegevens kan wijzigen.
- moet een global catalog server zijn.
- gebruikers ondervinden geen hinder als hij even offline is.

- **Wie heeft welke rol?**

- wort bijgehouden in de fSMORoleOwner attributen van vijf verschillende objecten in verschillende partities. (p61)
- een OM masters rol kan van andere domeincontrollers binnen het domein of het forest overgedragen worden.
- indien je een nieuwe forest maakt worden alle single master rollen automatisch toegewezen aan de eerste domein controller in dat domein.

5. **Welke domein controller worden als ODC ingesteld?** Alle nt5 controller beschikken over een equivalente wijzigbare kopie van alle AD partities vanaf windows 2008+ zijn er ook read only domain controllers.

- RODC beperken het replicatie verkeer maar in 1 richting
- men kan dynamisch een RODC filtered attribute set configureren, een verzameling van alle kenmerken die niet naar een RODC gerepliceerd worden.
- Men kan echter voor elke individuele RODC een password Replication Policy configureren die credential caching toestaat voor specifieke gebruikers en computers. Alle andere SPN objecten worden dan doorverwezen.
- RODC kan ook functies vervullen van een globale catalogus of van een DNS server. In dat laatste geval bekommt men een secundaire nameserver.
- een Operations master rol ondersteunen is niet mogelijk als RODC

## Deel II

# Schriftelijk - Reeks 2



## Hoofdstuk 7

# Active Directory replicatie (2.5)

### 7.1 Wat is de bedoeling van replicatie ?

- gebruikers en services moeten op elk gewenst moment vanaf elke computer in het forest toegang kunnen krijgen tot de directory gegevens.
- fouttolerantie en belastingsverdeling opvangen met meerdere dc's
- toevoegingen en wijzigingen op een controller doorgeven aan alle andere via de replicatieservice.
- alle domein controllers nemen deel aan de replicatie en bevatten een volledige kopei van alle directory gegevens voor het eigen domein.
- analoog beschikken alle dc in een forest over heel het directory schema en de configuratie gegevens. Deze info wordt gedistribueerd over het hele forest.
- creatie of modificatie van objecten leidt tot het doorsturen van de gewijzigde gegevens (replicatie)

## **7.2 Hoe wordt dit in Windows Server (ondermeer ten opzichte van NT 4.0) gerealiseerd: bespreek de verschillende technische kenmerken en concepten van Windows Server replicatie, en hoe men specifieke problemen vermijdt of oplost.**

### **7.2.1 Windows NT4**

- Master - slave model met primaire domeincontrollers en backup controllers.
- werd gebruikt om de SAM gegevens , de policies , de gebruikersprofielen en de logon scripts te distribueren.
- Slechts een server, de primaire domeincontroller, had een wijzigbare kopie van de directory.

### **7.2.2 Windows Server**

- multimaster replicatie: directory bijwerken vanaf elke DC (tenzij het een RODC is).
- evolutie van het master - slave model waarbij wel alle dc's ingeschakeld worden om aanvragen af te handelen, inclusief wijzigingen in de directory
- alle dc's zijn equivalent op de Operations master rollen na.
- meer fouttolerantie want meerdere domeincontrollers kunnen verder repliceren indien een controller uitvalt.
- replicatie verkeer beperken en dus enkel gewijzigde gegevens doorsturen.
- **Update Sequence Number (USN)** is een 64bits getal dat per dc verhoogd wordt elke keer hij een of ander object wijzigt.
  - zorgt ervoor dat een wijziging maar een keer wordt gerepliceerd
  - wordt verhoogd bij elke wijziging

- USN met GUID vormt de **Up-To-Dateness Vector (UTD)** van de dc.
  - dc meldt zijn utd telkens als er een wijziging gebeurt.
  - elke dc houdt een utd vector tabel bij met de meest recente utd van elke vector van andere dcs
  - heeft een dc al een wijziging? kijken naar UTD vector tabellen en die vergelijken
- de metadata van elk object die ook in AD opgeslagen zit, houdt voor elk kenmerk o.a. een Property Version Number (PVN) bij samen met de corresponderende UTD vector.
- op basis van de UTD vectortabel kan een controller alle wijzigingen met een bepaalde minimale USN aan een andere controller opvragen . Aan de hand van de metadata kan deze controller precies te weten komen welke wijzigingen hij maar moet opsturen.
- **conflicten**
  - opgelost door enkel de wijzigingen van hoogste GUID te accepteren. Kans op een conflict is nu zeer zeer klein.
  - voor schema en andere zeer kritische zaken, 1 controller aanduiden die wijzigingen kan doorvoeren, de operations master
- **tombstone objecten**
  - na verwijderen van een object wordt het als een tombstone gemarkeerd om te vermijden dat andere dc dit object opnieuw ophalen.
  - na afloop van de tomSoneLifeTime wordt het object effectief verwijderd, dit heeft als gevolg dat als een dc langer de deze tijd niet gesynched is, hij niet meer gesynched kan worden <sup>1</sup>.
  - Deze strict replication consistency kan wel uitgeschakeld worden.

---

<sup>1</sup>De reden waarom weet ik niet, maar ik vermoed dat het is omdat die objecten dan toch opnieuw opgevraagd zullen worden

### 7.2.3 Store-and-forward replicatie

- elke wijziging wordt maar met een andere domeincontroller uitgewisseld.
- de replicatietopologie, de configuratie van de verbindingsobjecten, wordt gegenereerd door de KCC of de Knowledge Consistency Checker.
  - De KCC zoekt zelf de meest optimale topologie.
  - hierbij worden topologieën gecreëerd voor de domeingegevens en voor de schema en configuratie gegevens anderzijds. Ook voor elke applicatie partitie is er een aparte topologie.
  - verbindingsobjecten gelden in een richting, de topologie zal bijgevolg niet uit koppel bestaan die met elkaar repliceren (bij RODC kan dit zowieso al niet)
  - de topologie verbindt elke domeincontroller met minstens 2 paden
  - elke controller is max met 3 andere verbonden
  - het grootste aantal hops tussen 2 domeincontrollers is 3
  - aparte topologie voor forest en domein gegevens
- mechanisme is pull based, een controller meld enkel dat hij wijzigingen heeft, de buurt zal ze nadien ophalen.
- Wijzigingen worden gebundeld en dan periodiek verstuurd (propagation damping), niet ingeschakeld voor attributen die met beveiliging te maken hebben (urgent replication)
- de replicatie is multithreaded en kan dus met verschillende partners tegelijk gebeuren.
- de kleinste replicatie entiteit
  - nt4 : 1 object
  - windows server : 1 kenmerk
  - windows server 2003+ : 1 waarde van een multivalued attribuut

### 7.3 Welke toestellen repliceren onderling in een forest ? Welke specifieke gegevens worden hierbij uitgewisseld ?

- alle DC in een forest beschikken over het directory schema en configuratie gegevens
- tussen een dc met GC van het ene domein en een dc van het andere domein wordt het schema en de config gerepliceerd. (enkel naar een dc in het andere domein<sup>2</sup>) De dc zal ook een subset van zijn domeingegevens naar de GC server repliceren.
- tussen de GC servers van een forest worden subsets van alle domeingegevens naar alle domeinen gerepliceerd.

### 7.4 Welke impact hebben sites met betrekking tot de replicatie van Active Directory gegevens ? Je hoeft hierbij het begrip site op zich niet verder te behandelen. (eerste helft 2.6.1, verschil tussen intrasite en intersite replicatie)

- binnen een site worden gegevens continue en automatisch gerepliceerd.
- bij replicatie tussen sites, ook wel intra-site replicatie genoemd, zijn er heel wat verschillen
  - replicatie partners van verschillende sites laten standaard het meldingsmechanisme van gewijzigde UTD vectortabellen achterwege. er wordt enkel polling toegepast (std. om de 3uur, kan op 15 minuten gezet worden). Hierdoor is Urgent replication niet mogelijk.
  - gegevensuitwisseling wordt standaard gecomprimeerd (40% compression rate), kan wel uitgeschakeld worden per verbinding.
  - sitekoppelingen gebruiken om aan te geven hoe de sites onderling verbonden zijn. KCC genereren automatisch enkel verbindingsobjecten tussen sites als er een sitekoppeling bestaat.
  - gebruik maken van meerdere routes tussen sites om de betrouwbaarheid te verbeteren.
  - geen sitekoppeling = geen verbinding; en dus een nutteloze site (vermits ie geen gegevens krijgt)

---

<sup>2</sup>zie figuur p67

- de topologie tussen sites wordt er gebruikt gemaakt van de Inter-site Topology Generator (ISTG), (dc met grootste GUID), om ervoor te zorgen dat er geen willekeurige dc gekozen worden om informatie uit te wisselen tussen de sites. Anders zouden er meerdere verbindingen gelegd worden tussen sites, maar met verschillende dcs
- ISTG maakt voor elke partitie hoogstens 1 verbindingsobject per sitekoppeling.
- de dc die dit verbindingsobject gebruiken worden bruggenhoofdserver genoemd.
- ISTG en bruggenhoofd servers zorgen ervoor dat de intra-site replicatie optimaal aangepast is aan de siteconfiguratie.

## Hoofdstuk 8

# Active Directory sites (2.6)

best 2.6 eens lezen ook

### 8.1 Welke rol vervullen sites ? Welke Active Directory aspecten worden erdoor beïnvloed ? Bespreek hoe deze aspecten anders gerealiseerd worden indien de toestellen zich al dan niet in verschillende sites bevinden. (ondermeer verschil tussen intrasite en intersite replicatie)

- replicatie kan heel wat bandbreedte opslorpen wat in een WAN omgeving wel tot problemen kan leiden.
- opdeling in sites, een site is een fysieke locatie. Een verzameling subnetwerken die aan LAN snelheid met elkaar kunnen communiceren.
- sites zorgen voor een fysieke structurering van het netwerk en voor het bijhouden van geografische aspecten ervan.
- er is niet noodzakelijk enig verband tussen de fysieke structuur en de domeinstructuur van het netwerk. Logische en fysieke structuren zijn onafhankelijk van elkaar.
- sites zorgen er dus eigenlijk voor om het replicatie verkeer over een wan verbinding te beperken
- replicatie tussen de DCS van verschillende sites wordt zoveel mogelijk beperkt, en moet expliciet worden geconfigureerd.
- de KCC software duidt een inter-site topology generator aan
- deze istg is een dc die toestemming heeft om de sitekopplingen te maken voor replicatie.

- hierbij wordt er hoogstens 1 verbindingsobject aangemaakt per partitie
- de dcs die dit gebruiken worden bruggenhoofden genoemd.

**8.2 Welke relaties bestaan er tussen sites, domeinen, domeincontrollers en global catalogs ? Druk deze relaties ondermeer uit in termen zoals ... een X vereist minimaal/exact/maximaal nr Y's ... . Geef een verantwoording voor elk van deze beweringen. Hoe wordt bepaald tot welke site computers behoren ?**

- een domein overspant meerdere sites.
- minstens een dc per domein nodig, meerdere mag
- per site is er (best) minstens een dc met een GC
- Een dc met gc zal over de sites heen repliceren om het verkeer te beperken.
- per site kunnen er meerdere domeinen aanwezig zijn.
- 1 GC per forest volstaat.

**8.3 Bespreek de diverse noodzakelijke instellingen om de verschillende aspecten van sites te configureren, en vermeld hierbij telkens: waar en in welke gegroepeerde vorm ze opgeslagen worden, waarom ze noodzakelijk zijn (ondermeer welke andere aspecten er afhankelijk van zijn).**

- de meeste config wordt bijgehouden in de AD zelf, namelijk de sites container van de configuratiegegevens.
- sommige instellingen van het replicatiemechanisme worden in het register van elke individuele DC bijgehouden.
- Deze instellingen staan in Parameters subtak van de NTDS service.
- Elementaire configuratie gebeurt met de active directory sites en servers snap-in (ds-site.msc)
- create van sites en toevoegen van lidservers aan sites kan in dssite.msc



- ook sitekoppelingen worden hier aangemaakt, in de inter site transort link container van de snap in
  - rechts op container klikken, ip , new site linke, nieuwe link aanmaken
  - let erop dat de sitekoppeling in tegenstelling tot verbindingsobjecten wel reflexief zijn, ze moeten niet in beide richtingen worden aangemaakt.
- sitekoppelingen zijn niet transitief, bijkomende instellingen zijn:
  - protocol: RPC (betrouwbaar maar meer belasting) of SMTP (vergt minder resources)
  - synchronisatie schema bepaald de tijdstippen waarop de replicatie gebeurd.
  - de bandbreedte en de kost van de verbinding kunnen worden ingesteld, de goedkoopste verbinding die beschikbaar is, wordt gebruikt.
  - het interval tussen verschillend polls
- voor gedetailleerde config moet het objecte van de sitekoppleling rechtstreeks aangepast worden met bv ADSIEdit.msc
- alle verbindingsobjecten zowel inter als intra-site zijn opgenomen in de NTDS settins container vandssite.msc snap-in.
  - de manuele creatie van verbindingsobjecten kan in deze container.
  - meestal is het beter om de KCC deels of volledig zijn gang te laten gaan.
- subnetten vormen de basis voor de indeling in sites
  - elke site is geassocieerd met minstens een subnet en een sitekoppeling
  - subnetten kunnen gecreëerd worden met de dssite.msc
- het bruggenhoof die als enige de gegevens mag repliceren over de sitegrenzen heen, kan expliciet worden geconfigureerd in de properties tabpagina van de server.
- de site covering van een dc bepaald welke sites hij kan bedienen , in het geval dat alle DCs in een bepaald site zijn uitgevallen. Sites krijgen best een eigen global catalog toegewezen. een DC tot global catalog server promoveren kan ook in dssite.msc, meer bepaald in de general tabpagina van de properties tabpagina van de server.
- de site covering van een dc bepaald welk sites hij kan bedienen, in het geval dat alle cs in een bepaald site zijn uitgevallen
- sites krijgen best een eigen GC toegewezen, een dc tot GC promoveren kan in dssite.msc, general tabpagina,properties van de NTDS settins van de overeenkomstige server

- in de properties van NTDS site setting kan de inter-site topology generator voor de site worden gekozen.
- ook universal group membership caching kan hier worden ingeschakeld, dankzij deze optie is het in 2003+ domeinen mogelijk om de inlogprocedure te voltooien zonder een global catalog server te moeten contacteren.
- de cache wordt in ad zelf opgeslagen, maar wordt niet gerepliceerd.

## Hoofdstuk 9

# Gedeelde mappen en NTFS

**9.1 Welke manier om gedeelde mappen te creëren biedt de meeste configuratieinstellingen aan ? Bespreek het doel van deze diverse instellingen en de belangrijkste eigenschappen en mogelijkheden ervan. (3.2.1, 3.2.2, fracties 3.3.1, 3.4.2, 3.4.3, 3.5 en 3.6)**

### 9.1.1 Creatie

Er zijn vier mogelijkheden om shares aan te maken:

**advanced share knop** an de **sharing tabpagina** van de **folder properties** van de **map** die we **wensen te delen** op de **computer** die we **gebruiken**.

- **Share this folder:** geeft aan of de map gedeeld is of niet.
- **share this name:** voegt een share naam toe.
- **share limit the number of simultaneous users:** beperkingen op het aantal gebruikersdat tegelijkertijd verbinding kan maken
- **permissions:** machtigingen
- **caching:** offline mogelijkheden door caching

### **Shared folders MMC snap-in**

- zowel als extensie (compmgmt.msc of stand-alone (fsmgmt.msc)
- rechtermuisklik op systemtools, shared folders, sharesmap , new share, wizard hier moet je ook nog het pad eerst ingeven.

#### **server manager - ServerManager.msc**

- rechtermuisknop op roles, file services, share and storage management, provision share, wizard.
- biedt meer configuratie mogelijkheden.
- achtereenvolgens wordt gevraagd om volgende aspecten te configureren:
  - shared folder location
  - ntfs permissions
  - smb permissions
  - smb settings access based enumeration en client-side caching
  - quota policy
  - filescreen policy

#### **net share - command prompt**

- enkel mogelijk op de locale computer, vanop afstand moet je rmtshare gebruiken
- optie grant identity permission om een machtiging voor een identiteit toe te kennen
- mogelijk op windows server 2008+
- optie delete om share te verwijderen

### **9.1.2 Configuratie**

#### **zichtbaarheid**

- standaard voor iedereen zichtbaar
- verbergen door de naam te laten eindigen op \$
- verborgen shares zijn niet zichtbaar in explorer, maar wel in compmgmt.msc en de server manager.
- na installatie van de windows server zijn er reeds een aantal standaard shares, de meeste zijn verborgen.

## **machtigingen**

- wie heeft toegang tot wat en kan er wat mee doen?
- elk object in de AD of van een NTFS volume, elke registersleutel, elk proces, elke service heeft een security descriptor.
- deze security descriptor bestaat uit Discretionary Access Control List (DACL of ACL), System Access Control List (sACL, logging), SID van de eigenaar en de primaire groep van de eigenaar (wegens POSIX compatibiliteit)

### **9.1.3 ACL**

- Discretionary Access Control List = machtingsset
- verzameling van machtigingen die gelden voor bepaalde groepen of gebruikers + bepaald het type toegang.
- Welke machtigingen gekoppeld kunnen worden is afhankelijk van het type object (een bestand is verschillend van een register key)
- elke toewijzing van machting wordt een machtingsvermelding of Access Control Entry ACE genoemd. Alle ACE vormen samen de ACL. ACE's kunnen aangemaakt worden voor gebruikers en groep van het eigen domein en alle trusted domeinen.
- Zo veel mogelijk machtigingen toepassen op de groepen en niet op individuele gebruikers voor een eenvoudiger beheer.
- de ACL van een object wordt steeds in canonieke volgorde verwerkt door Security Reference Monitor (SRM). Eerst worden alle ACE die machtingen ontzeggen overlopen vervolgens de gene die machtingen toekennen. Ontzeggen is dominant op toekennen.

### **9.1.4 machtingen zijn hiërarchisch gestructureerd en overerving is mogelijk**

- Er zijn 2 soorten machtingen
  1. **expliciete machtingen:** zijn rechtstreeks aan een object gekoppeld (tijdens het maken of achteraf geconfigureerd) en hebben steeds voorrang op impliciete machtingen
  2. **impliciete machtingen:** zijn overgeërfd van de container waarin het object zich bevindt, dit vereenvoudigt het beheer van machtingen en vergoot de consistentie.

### 9.1.5 Verschil tussen een security descriptor met lege ACL of zonder ACL

Een lege ACL laat niemand toe, geen ACL betekent dat iedereen toegang heeft.

### 9.1.6 2 niveaus voor machtigingen

- **Share niveau = SMB machtigingen**

- eerste beveiligingslaag
- hebben overhand op NTFS machtigingen als ze meer beperkend zijn
- bij FAT is dit het enige mogelijke
- 3 machtigingen
  1. full control
  2. read
  3. change
- omzeild door gebruikers die lokaal inloggen.

- **NTFS niveau**

- 2de (beveiliging van mappen) en 3de (beveiliging van bestanden) verdedigingsmuren.
- vaak enkel NTFS machtiging ingesteld en volledige toegang op share niveau.
- Randeffect: gebruikers kunnen onmiddellijk submappen zien, ookal hebben ze geen machtigingen. Dit kan opgelost worden door de Acces Based Enumeration in te schakelen.
- 2 niveaus van machtigingen
  1. laagste niveau - 13 atomaire machtigingen die gebruikt worden voor het hogere niveau. Dit zijn de kleinst mogelij machtigingen
  2. hoogste niveau - moleculaire of standaard machtigingen. Dit zijn 6 veel gebruikte combinaties van atomaire machtigingen.
- instellen van machtigingen op heel veel toffe verschillende manieren zie 3.3.3

- quota's instellen
  - \* opslag capaciteit per gebruiker afdwingen.
  - \* staat standaard uit
  - \* 2 soorten quotas
    1. Volumequotas
      - gelden voor een heel NTFS volume
      - zijn gebaseerd op bestandseigendom (dus SID gebonden en niet per groep.)
      - indien volume meerdere mappen bevat wore de quota's collectief van toepassing op alle mappen.
      - houden geen rekening met eventuele bestandscompressies
      - aanzetten met opdracht fsutil quota of in de explorer bij properties van een volume en dan de quota tab.
    2. map quotas
      - enkel windows server 2008+
      - instellen van quota's op mapniveau
      - gelden collectief, onafhankelijk van de bestandseigendom
      - houden rekening met bestandscompressie
      - vooral zinvol indien de toegang tot de mappen beperkt is tot individuen of groepen.
      - aanzetten via de quota management tak in server manager of de fsm.msc of dirquota opdracht
  - \* voor beide soorten wordt er gebruik gemaakt van 2 drempel waarden
    1. soft threshold : waarschuwing
    2. hard threshold : eenmaal overschreven kan de gebruiker geen bestanden meer opslaan.

- **file screens**

- verhinderen dat bestanden met bepaald extensies kunnen opgeslagen worden
- kunnen reacties triggeren bij poging om het toch te doen
- configuratie analoog met mapquota's , maar nu via de file screening management tak of via filescreen opdracht

- **client-side caching**

- caching van vaak gebruikte netwerk bestanden
- kopieën worden opgeslagen in een map op de lokale HDD met de naam offline Files.
- wijzigingen worden zowel in de cache als op de share doorgevoerd.
- indien de timestamp gelijk is wordt het lokaal bestand geopend, anders wordt het bestand van de share afgehaald/geupdatet.
- de cache blijft beschikbaar als de netwerkshare onbeschikbaar is, zodat de gebruiker kan doorwerken.
- hogere snelheid + betere betrouwbaarheid.
- sync center programma zorgt dat offline wijzigingen worden doorgevoerd zodra de share beschikbaar is. Indien hierbij een conflict optreedt, moet de gebruiker dit oplossen (hij zal de gebruiker dus om een actie vragen).
- bij de configuratie kan men de bestanden vrijlaten aan de gebruiker, alle bestanden laten cachen (voor performantie, bv .exe's wordt steeds lokaal uitgevoerd) of de caching afzetten.

## 9.2 Waar wordt de definitie en (partiële) configuratie van gedeelde mappen opgeslagen ? Hoe kan men deze wijzigen vanuit een Command Prompt ?

zie hierboven en cursus

## 9.3 Op welke diverse manieren kan men gebruik maken van gedeelde mappen ? (3.2.3)

### mappen in het lokale filesystem

- de share krijgt dan een stationsletter toegewezen
- kan dan gebruikt worden alsof het een lokale schijf was
- handig voor shares die veel gebruikt worden.
- via command prompt met de opdracht net use.
- kan ook via pushd deze pakt dan automatisch de laatste beschikbare driveaanduiding.



**9.4 Geef een overzicht van de belangrijkste voordelen van de opeenvolgende versies van het NTFS bestandssysteem. Bespreek elk van deze aspecten (ondermeer het doel, de voordelen en de beperkingen ervan), en geef aan hoe je er gebruik kan van maken, bij voorkeur vanuit een Command Prompt. (NTFS fractie 1.6, fracties 3.4.1, 3.4.2 en 3.4.4)**

De NTFS versies zijn:

1. V1.0, V1.1, V1.2, NT4 en voorlopers
2. V3.0 Windows 2000
3. v3.1 windows xp en hoger

**9.4.1 Features vanaf v1.2**

- beveiliging op bestandsniveau.
  - Gebruikt van security descriptors.
  - vereist voor sommige functies van windows server waaronder de AD
- logging van schrijfactiviteiten
- dynamisch uitbereiden van partities en volumes
- compressie
- spannende volumes (over meerdere fysieke schijven)
- grotere partities zonder performantie degradatie
- hardlinks
- auditing op objecttoegang

### 9.4.2 Features vanaf V 3.0

- reparsepunten en bestandssysteemfilters
- transparante encryptie en decoding
- individuele diskquota op volume niveau
- volumekoppelpunten, mounten van volumes in ntfs mappen
- sparse (ijle) bestanden
  - mogelijkheid om enkel schijfruimte toe te wijzen aan delen van grote bestand waar-naar geschreven werd, enkel nuttige data (vrs van 0) wordt bewaard.
  - dit kan met de opdracht fsutil sparse setflag
- hardlinks : fsutil hardlink create
- junction points (softlinks) naar mappen. File markers voor remote storage service dmv junction point: bestandnaam op server , eigenlijke bestand op tape. via linkd
- in NT6 voor beide de opdracht mklink gebruiken met /h voor hardlink en /d voor symbolic link/. mklink kan voor zowel lokale als remote bestanden gebruikt worden.
- de meeste features worde gerealiseerd door reparse punten
  - wanneer nt5+ een reparsepunt detecteerd wordt de tag van het reparsekenmerk teruggestuurd naar de IO-stack, waar de reparsetag wordt onderzocht door extra bestandssysteemfilters, die bij herkenning nt5+ vertellen dat er een ander stuur-programme moet worden gebruikt dan het standaard NTFS stuurprogramma.
  - dit mechanisme laat toe dat er extra functionaliteit wort toegevoegd aan het be-standssysteem, door microsoft en andere partijen zonder dat het bestandssysteem telkens moet worden herzien.
  - Voorbeelden in de windows server 2008:
    - \* self healing van het volume, zo kunnen corrupties online opgespoord en verbe-terd worden.
    - \* ondersteuning van transacties , meerdere transacties bundelen tot 1 atomair geheel.
    - \* mapquota
    - \* filescreens

## Hoofdstuk 10

# Machtigingen op bestandstoegang (3.3)

### 10.1 Welke rol spelen machtigingen bij de beveiliging van bronnen ? Geef een gedetailleerd algemeen overzicht van het mechanisme van machtigingen.

- machtigingen bepalen wie er toegang heeft tot welke gegevens en wat die er mee kan doen.
- elk object in de AD , elk object van een ntfs volume , elk registersleutel , elk proces en elke service heeft een security descriptor. Deze bevat:
  - een machtigingsset of Discretionary Access Control List (ACL)
  - Een System Access Control List (ACL) die definieert welke acties van de gebruiker gelogd worden
  - de SID van de eigenaar vna het object. Makers van objecten zijn standaard eigenaar, maar kunnen de eigendom overdragen. De eigenaar of beheerder is verantwoordelijk voor het instellen van de ACL.
  - Wegens POSIX compatibiliteit: de primaire groep van de maker.
- Een ACL is een verzameling machtigingen die gelden voor bepaalde groepen of gebruikers.
- door machtigingen in te stellen voor een object, bepaalt de eigenaar of beheerder welke gebruikers en groepen toegangsrechten voor het object hebben. Deze machtigingen bepalen ook het type toegang.
- welke type machtigingen er aan een object gekoppeld kunnen worden is afhankelijk van het type object

- elke toewijzing van machtigingen aan een gebruiker of groep wordt een machtigingsvermelding of access control entry (ACE) genoemd. Hierin wordt de gebruiker of groep opnieuw geïdentificeerd met zijn SID.
- een ACE kan een machtiging ontkennen of toezeggen
- alle ACE 's tesamen vormen de ACL
- best om de machtigingen per groep toe te passen om het beheer simpeler te houden.
- ACE worden in canonieke volgorde behandeld, eerst de ontzeggingen, daarna de toekenningen. Een ontzegging staat dus hoger dan een toekenning.
- Er zijn 2 soorten machtigingen
  1. Expliciete machtigingen : rechtstreeks aan het object gekoppeld
  2. impliciete machtigingen : via de container waartoe het object behoort.
- expliciete machtigingen krijgen altijd voorrang op impliciete
- lege ACL =  $\emptyset$  niemand heeft machtigingen
- geen ACL =  $\emptyset$  iedereen heeft alle machtigingen

## 10.2 Bespreek hoe het mechanisme van machtigingen specifiek (en op diverse niveaus) toegepast wordt op bestandstoegang. Geef de verschillende soorten machtigingen, hun onderlinge relaties, en hoe deze kunnen geanalyseerd en ingesteld worden. Toon hierbij aan dat je zelf met deze configuratietools geëxperimenteerd hebt.

- Hiervoor worden NTFS machtigingen gebruikt deze bestaan in 2 niveaus.
  1. **13 atomaire machtigingen**, dit is het laagste niveau.
    - (a) **Traverse Folder/Execute File:** Traverse Folder geldt voor mappen en laat de gebruiker toe om hulpbestanden die zich meer dan een niveau lager bevinden waartoe de gebruiker toegang heeft, maar niet de tussenliggende mappen toch te gebruiken.
    - (b) **List Folder/Read Data:** Inhoud van een bestand of map tonen.
    - (c) **Read Attributes:** Elementaire bestandskenmerken weergeven.
    - (d) **Read Extended Attributes:** Programma-afhankelijke uitgebreide bestandskenmerken weergeven.

- (e) **Create Files/Write Data:** Nieuwe bestanden in een map aanmaken/Data van een bestaand bestand overschrijven, maar niet aan een bestaand bestand toevoegen.
  - (f) **Create Folder/Append Data:** Nieuwe mappen creëren en data toevoegen aan een bestand.
  - (g) **Write Attributes:** Elementaire bestandskenmerken wijzigen.
  - (h) **Write Extended Attributes:** Uitgebreide bestandskenmerken wijzigen.
  - (i) **Delete Subfolders and Files:**
  - (j) **Delete:** Het object zelf verwijderen.
  - (k) **Read Permissions: NTFS machtigingen weergeven.**
  - (l) **Change Permissions: NTFS machtigingen wijzigen.**
  - (m) **Take Ownership:** eigenaarschap van een bestand of map overnemen. De eigenaar staat buiten alle machtigingen en kan de machtigingen van het object wijzigen.
2. **6 Moleculaire machtigingen :** vormen combinaties van atomaire machtigingen om zo de meest gebruikte combinatie aan te bieden.
- (a) **Full Control:** Alle atomaire machtigingen
  - (b) **Read:** Inhoud, machtigingen en kenmerken van een object bekijken. Submappen waar geen toegang tot is blijven zichtbaar.
  - (c) **Read&Execute:** Identiek aan de Read machtiging, aangevuld met de atomaire Traverse Folder/Execute File machtiging.
  - (d) **Write:** Nieuwe bestanden en submappen aanmaken. Om bestaande bestanden te kunnen wijzigen is de aanvullende Read moleculaire machtiging nodig
  - (e) **Modify: Read&Execute + Write + Delete (atom.)**
  - (f) **List Folder Contents:** enkel voor mappen, laat toe om inhoud van mappen te bekijken ongeacht het bezit van Read toegang.
- de ACL van een object bevindt zich in de security tab van het properties venster. In nt6+ moet er eerst geklikt worden op edit.
  - de permissions tab toont een overzicht van de ACE in volgorde zoals ze door de SRM worden verwerkt. De kolom inherited from geeft aan vanwaar het attribuut werd overgeërfd

- de knop edit opent het venster in edit mode, door een ACE te selecteren in edit mode komt een lijst van atomaire machtigingen tevoorschijn. Speciale combinatie van atomaire machtigingen die niet tot een combinatie van moleculaire machtigingen te herleiden zijn kunnen hier worden ingesteld. In het properties venster wordt dit weergegeven als special permissions. De permissions tab in edit mode laat drie aanvullende mogelijkheden toe:
  1. Allow inheritable permissions : geeft aan of er machtigingen worden overgeërfd van de bovenliggende map. Bij het uitschakelen wordt gevraagd of bestaande overgeërfde permissies van de bovenliggende map moeten worden gekopieerd of verwijderd.
  2. Replace all existing inheritable permissions: zorgt ervoor dat indien ingeschakeld, alle machtigingen op onderliggende object expliciet worden toegepast. Hierbij worden alle machtigingen van de onderliggende objecten vervangen. Reeds gedefiniëerde machtigingen gaan verloren. De actie kan niet ongedaan gemaakt worden.
  3. Add/Remove/Edit : hiermee kunnen ACE worden toegevoegd verwijderd. Ook de atomaire machtigingen kunnen worden aangepast. De keuzelijst apply onto kan overerving selectief laten gelden.
- de auditing tab gebruikt om de sacl van het object in te stellen.
- de owner tabpagina laat toe om de eigenaar van het object te veranderen. Dit kan enkel gebeuren door gebruikers met de take ownership machtiging op het object; die enkel kan worden toegekend door beheerders, eigenaars of gebruikers met Full Control. Overdrachten van eigendom moeten wederzijds aanvaard worden.
- de effectieve permissions tab toont een overzicht van de uiteindelijke effectieve machtigingen zoals ze zouden worden toegekend door de security reference monitor. Dit kan handig zijn voor debugging en probleem diagnose.
- In de command prompt kunnen machtigingen worden bekeken met showacls of perms. Wijzigingen aanbrengen kan met iacs , xacs of subinacl. De eerst laat toe om een lijst van permissions te bewaren in een bestand, om dan later met de optie restore te herstellen.

### **10.3 Wat gebeurt er met de machtigingen bij het verplaatsen van een bestand ? Wat gebeurt er met de machtigingen bij het kopiëren van een bestand ?**

- de gebruiker die de kopie maakt wordt eigenaar van de bestanden.
- binnen hetzelfde ntfs volume : de expliciete machtigingen worden behouden, de impliciete worden vervangen door de impliciete machtigingen van de aankomst locatie.
- indien er gekopieerd wordt naar een ander NTFS volume, vervallen alle machtigingen en worden de impliciete machtigingen van het aankomst volume toegepast.
- naar anders soort volume dan NTFS : alle machtigingen weg
- machtigingen wel behouden ? kopiëren met speciale tools zoals scopy en robocopy.

### **10.4 Op welke andere objecten zijn machtigingen van toepassing ?**

De machtigingen zijn van toepassing op elk object in de AD , elk object van een NTFS volume, elke register sleutel, elk proces en elke service. Deze hebben allemaal een security descriptor en dus de bijhorende machtigingen.

## Hoofdstuk 11

### Gebruikersgroepen (4.2.2 en 4.2.3)

**11.1** Bespreek in detail het onderscheid tussen de diverse soorten veiligheidsgroepen, ondermeer afhankelijk of het toestel al dan niet in een domein is opgenomen. Behandel hierbij vooral de mogelijkheden en beperkingen. Bespreek ondermeer:

- de zichtbaarheid van de diverse soorten groepen
- welke objecten er lid van kunnen zijn
- de onderlinge relaties en de regels voor het nesten van de diverse soorten groepen ? Stel deze relaties eveneens schematisch voor.

**antwoord**



|                  |        | Zichtbaarheid  | Geldigheid                   | Aanwezig op   | Bevat   |
|------------------|--------|--|------------------------------|---|---|
| Domein groep     | lokale | alle lidcomputers (workstations en servers) van het domein | eigen domein                 | dc van eigen domein                                       | gebruikers en groepen uit elk domein van het forest of een ander trusted domein. Lokale groepen van het eigen domein. |
| Globale groep    |        | elk domein van het forest of een ander trusting domein     | volledige forest             | DC en eigen domeinen en GC (enkel naam op GC, geen leden) | Gebruikers en globale groepen van het eigen domein.   |
| Universele groep |        | Alle domeinen van het forest                               | Alle domeinen van het forest | GC van het forest   |   |

## **11.2 Hoe en waarom worden deze soorten groepen in de praktijk best gebruikt, al dan niet gecombineerd ? Van welke omstandigheden is dit afhankelijk ? Illustreer aan de hand van concrete voorbeelden.**

Lokale groepen worden meestal gebruikt om rechten en machtigingen toe te kennen binnen hetzelfde domein, en bevatten eerder andere groepen dan gebruikers. Men creëert een lokale groep, die men toegang geeft tot de bron en voegt andere groepen aan deze groep toe, om te toegang te verlenen. Lokale groepen zijn interessant wanneer de groep enkel zichtbaar mag zijn binnen een domein.

Globale groepen worden gebruikt om gebruikers van een domein te groeperen. Globale groepen kunnen rechten worden toegewezen in een ander domein, bijvoorbeeld door ze toe te voegen aan een lokale groep van het domein waartoe de bron behoort, die toegang heeft tot de bron. Globale groepen zijn minder interessant om aan een bron te koppelen, omdat ze enkel gebruikers en andere globale groepen kunnen bevatten. Lokale groepen kunnen zowel gebruikers als elk soort andere groepen bevatten.

Universele groepen kunnen voor dezelfde doeleinden worden gebruikt als domein lokale groepen, maar zijn dan onmiddellijk zichtbaar in het gehele forest. Dit heeft dus het bijkomend voordeel dat ze niet in elk domein opnieuw moeten gedefinieerd worden. Een mogelijke strategie laat lokale en universele groepen achterwege, en gebruikt enkel universele groepen. Dit heeft echter als risico dat de GC onnodig groot wordt, omdat elke universele groep met al zijn leden in de GC wordt opgenomen. Indien er nog frequente wijzigingen aan het lidmaatschap gebeuren, veroorzaakt dit bijkomend replicatieverkeer. In 2003+ is dit probleem beperkter omdat niet de gehele lijst van leden moet worden gerepliceerd, maar enkel de individuele elementen van het multivalued attribuut. Bovendien is het contacteren van een GC nodig voor het inloggen van een gebruiker, om het lidmaatschap van een universele groep na te gaan.

### **11.3 Waar en hoe wordt het (volledige) lidmaatschap van een object tot een groep bijgehouden ? Op welke diverse manieren kan men dit lidmaatschap configureren ? Door wie wordt dit lidmaatschap bij voorkeur ingesteld ? Op welke diverse manieren kan men de volledige verzameling van objecten, die er deel van uitmaken, achterhalen ?**

De groepen waartoe een object behoort worden opgeslagen in het memberOf attribuut. Omdat het een back-link attribuut is, kan het niet rechtstreeks worden gewijzigd. Objecten programmatisch aan groepen toe te voegen, kan door het member attribuut van het groeps-object aan te vullen. Dit is het corresponderende forward-link attribuut van memberOf.

Om met de grafische interface een gebruiker of groep aan een andere groep toe te voegen, kan het Member Of tabblad van het Properties venster van de gebruiker of groep worden gebruikt.

### **11.4 Welke conversieregels gelden er tussen de diverse soorten groepen. Behandel hierbij alle mogelijke combinaties.**

- Lokale groepen kunnen naar de universele scope worden geconverteerd, op voorwaarde dat ze geen andere lokale groepen bevatten.
- Globale groepen kunnen naar de universele scope worden geconverteerd, zolang de groep geen lid is van een andere groep met globale scope.
- Universele groepen kunnen meestal niet worden omgezet naar anders scopes.
- Veiligheidsgroepen kunnen worden omgezet naar distributiegroepen, en omgekeerd.