Theorie vragen BSIII - windows 2014

Andreas De Lille

Augustus 2014

Inhoudsopgave

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 compabiliteitsredenen.
   - 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 vereevoudigen.
   - 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 compabiliteitsredenen. 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 $ [1].
   - Deze naam kan op elk gewenst moment veranderd worden.

[1] 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 gevold door de suffix voor de primaire DNS
    - Standaard is de suffix de volledige DNS naam van het domein waar de computer toe behoord.
    - 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 domeincontroller 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**

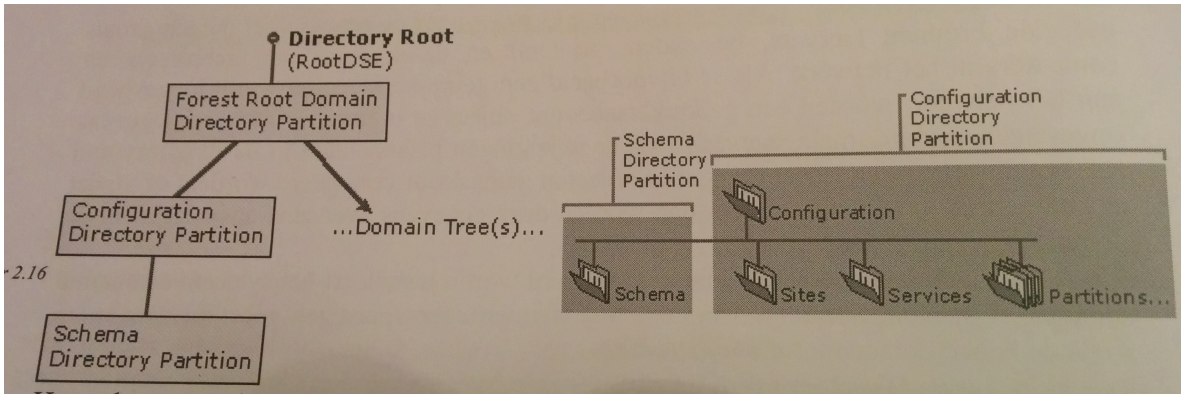
   - beschrijven de fysieke topologie van de directory (bv lijst van alle domeinstructuren, locaties van domeincontrollers en global catalog controllers, sites, ..)
   - Instellingen voor het hele forest worden vertaald naar kenmerken van objecten in de configuratie gegevens. (bv. uPNSuffixes kenmerk houdt 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 ndien 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-obecten, 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 ngo 2 andere beschikbaar:

1. **attributeSchema-objecten** waar de attributen gedefiniëerd worden. zie hierboven ergens.

2. **abstract schema objecten** een compacte representatie van het reëele 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 doorvoerenvia 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 linkerpanel 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-BehaviorVersion. 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 lidservers mogen lager zijn
3. **Windows Server 2003** enkel windows server 2003+ domein controllers, werkposten en lidservers mogen lager zijn.
4. **Windows Server 2008** enkel windows server 2008+ domein controllers, werkposten en lidservers mogen lager zijn.

4.1.2 Forest functioneel niveau

Analoog aan het domein funcitoneel 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 containerobect 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 namesspaces 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 levensduus
- 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 gesynced 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 goee 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 weerspiegeld de organisatie** waarin verschillende bedrijfseenheden 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 initiÃ«le 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):

- **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 ?**
- **Eigenschappen zoals bedoeling, noodzaak, kriticiteit, inhoud, synchronisatie, voor welke Windows versie(s) van toepassing, ... ?**
- **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.**
- **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, toepassingsservers, 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.
   - lidservers 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 performatie 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 verkeerd 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 aatal 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 an 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.
- dc 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 vervult 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 omeinen 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 obecten 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 beschkken 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 bekomt men een secundaire nameserver.
   - een Operations master rol ondersteunen is niet mogelijk als RODC

Deel II

Schriftelijk - Reeks 2