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**

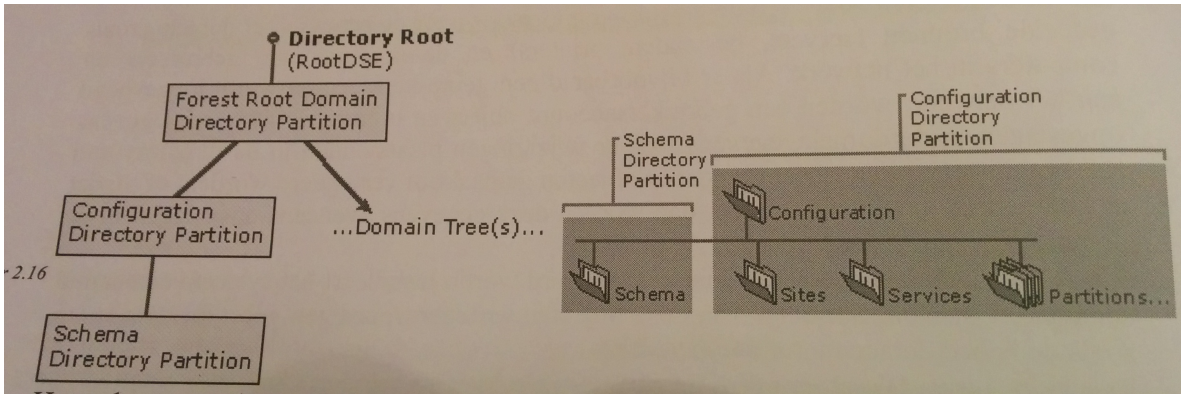
   - beschrijven de fysieke topologie van de directory (bv lijst van alle domeinstructuren, locaties van domeincontrollers en global catalog controllers, sites, ..)
   - Instellingen voor het hele forest worden vertaald naar kenmerken van objecten in de configuratie gegevens. (bv. uPNSuffixes kenmerk houdt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