FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Besturingssystemen III: Examen

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

Gecompileerd op 27 november 2018

UNIVERSITEIT
GENT

Inhoudsopgave

Hoofdstuk 1

Modelvragen theorie: reeks A

Het examen wordt volledig schriftelijk beantwoord. Indien de student dit wenst, wordt het antwoord onmiddellijk na indienen geëvalueerd, en eventueel gevolgd door enkele vragen ter verduidelijking of aanvulling.

1.1 Structuur van Active Directory gegevens

1. Bespreek de *diverse namen* die alle Active Directory objecten *identificeren.* §2.2.1

   - **Relative distinguished name.** Dit is een identificatie van een object **binnen** een **containerobject**. Een relative distinguished name moet dus **niet uniek** zijn op Active Directory niveau. Dit wordt opgeslagen in het attribuut **cn**.
   - **Distinguished name.** Deze **unieke** identificatie wordt opgebouwd uit de relative distinguished name van het object zelf en van alle RDNs waarvan het object hiërarchisch deel uitmaakt. Dit wordt opgeslagen in het attribuut **distinguishedName**
   - **Canonieke naam.** Dit heeft dezelfde functie als een distinguished name, maar heeft een eenvoudigere representatie. Dit wordt opgeslagen in het attribuut **canonicalName**
   - **GUID.** Elk object heeft een **unieke** GUID. Dit is een 128-bit getal dat niet kan gewijzigd worden. Een GUID wordt aangemaakt bij de **creatie** van een object en kan dan **niet meer aangepast** worden. Dit wordt opgeslagen in het attribuut **objectGUID**.

2. Wat zijn *SPN objecten* ? Bespreek de *aanvullende naamgeving* voor deze objecten. (§2.2.2)

   Security Principal Objects (SPN) zijn objecten die **security IDs (SIDs)** bevatten. Dergelijke objecten worden gebruikt voor het verlenen van toegang tot domeinbronnen en zijn daarom van toepassing op gebruikersaccounts, computeraccounts, groepen en domeinen. Een SID is net zoals een GUID uniek binnen een forest voor elk nieuw aangemaakt object. Bij het verplaatsen en hernoemen binnen hetzelfde domein blijven de GUID en SID behouden van een SPN object. Indien het object naar een andere domein verplaatst wordt, zal enkel GUID hetzelfde blijven. Hierbij wordt het **sIDHistory** attribuut aangevuld met het vorige SID.

3. Enkele veel gebruikte klassen (hiermee worden *attributeschema* en *classschema* objecten niet bedoeld) vertonen nog meer identificerende attributen voor hun instanties. Bespreek deze klassen en attributen.

   - Een **gebruikersaccount** heeft nood aan een extra identificatieattribuut om inlogprocedures door een gebruiker te vereenvoudigen. Een **User Principal Name (UPN)** is een vereenvoudigde waarde (loginnaam) dat uitsluitend gebruikt wordt voor aanmelding. Verder moet een UPN uniek zijn binnen het hele forest en heeft standaard de volgende vorm:

     `RDN@UPNsuffix`

     Hierbij is RDN de RDN van de gebruiker en UPNsuffix één van de volgende alternatieven:
     - De DNS domeinnaam waarin het gebruikersaccount zich bevindt.
     - De DNS domeinnaam van het root domein.
     - Een willekeurige maar een op voorhand gedefinieerde naam.
   - Een **computeraccount** bevat drie extra attributen:
     - **SAM accountnaam.** Dit is gelijkaardig aan het UPN attribuut, maar dient voor compatibiliteit met oudere windows systemen (pre 2000). Standaard bestaat deze naam uit de eerste 15 bytes van de RDN, gevolgd door een $ teken. Deze naam kan op elk moment gewijzigd worden. Deze waarde wordt opgeslagen in het attribuut **samAccountName**.
     - **DNS hostnaam.** Deze waarde bevat standaard de eerste 15 tekens van de RDN en de suffix voor primaire DNS;
     - **Service Principal Name.**

4. In welke *partities* is de Active Directory informatie verdeeld ? Geef de betekenis van elke partitie, hun onderlinge relatie (zowel fysiek als met betrekking tot hun naamgeving), en de replicatiekarakteristieken ervan. (laatste helft §2.2.3)

   - **Domeingegevens.** Deze partitie bevat informatie over alle objecten in het domein (servers, bestanden, printers, accounts, ...). Objecten die aangemaakt of gewijzigd zijn, worden steeds opgeslagen in de domeingegevens. In Active Directory kunnen er meerdere domeinen bestaan in een forest. Deze domeinen vormen een boomstructuur met als wortel de domeingegevens van het **root** domein. Domeingegevens van een bepaald domein worden gerepliceerd tussen alle domeincontrollers van enkel dat domein.
   - **Applicatiepartitie.** Elk van de eventuele applicatiepartities wordt uitgewisseld tussen een eigen deelverzameling specifiek hiervoor geconfigureerde domeincontrollers van het forest, onafhankelijk van domeingrenzen. De applicatiepartitie DomainDNSZones wordt enkel gesynchroniseerd met domeincontrollers binnen een domein die ook DNS server zijn. De applicatiepartitie ForestDNSZones daarentegen zal de domeincontrollers die DNS server zijn op forestniveau synchroniseren.
   - **Configuratiegegevens.** Deze partitie beschrijft de fysieke topologie van de directory. Het bevat onder andere een lijst van alle domeinstructuren, de locaties van de domeincontrollers, de sites en de replicatietopologie. Deze partitie bevat ook de koppeling tussen een partitie en zijn replicerende domeincontrollers in het attribuut **msDS-NC-Replica-Locations** van het overeenkomstige **crossref** object. Enkel voor applicatiepartities kan dit dynamisch gewijzigd worden.

- **Het schema.** Dit is de formele definitie van alle objecten en kenmerkgegevens die kunnen opgeslagen worden in Active Directory. Dit schema is uniek voor alle domeinen in het forest.

Een willekeurige domeincontroller zorgt voor de opslag en replicatie van

1.2 attributeSchema objecten (§2.2.4 en §2.2.5)

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

   - Het attributeSchema bevat alle kenmerken die in het schema voorkomen. Een kenmerk is zelf een object. Zo een kenmerk wordt éénmaal gedefinieerd en kan meerdere malen gebruikt worden bij verschillende klassen, wat voor consistentie zorgt. Een goed voorbeeld hiervan is het description kenmerk. Om op een eenvoudige manier objecten te raadplegen of wijzigen, kan men met het schema snap-in mechanisme werken. **Active Directory Schema** is zo een snap-in, dat eenvoudig in twee vensters zowel de kenmerken als de klassen weergeeft. Dubbelklikken op een item geeft de meest relevante eigenschappen van het object, en de mogelijkheid om deze in te stellen. Er kan ook gebruikt gemaakt worden van de opdracht **dsquery ***. Deze opdracht kan om het even welke kenmerken tonen van één enkel object, of van alle objecten in een container, zowel niet-recursief als recursief.

2. Bespreek de *diverse naamgevingen*, specifiek voor attributeSchema objecten.

   - **Common Name**: De RDN van het attributeSchema object in de Schema container.
   - **GUID**: Dit kan automatisch gegenereerd worden bij de creatie van een nieuw kenmerk. Een attribuut krijgt dan wel een verschillende GUID in verschillende forests. Manueel een GUID instellen kan ook, met bv de **guidgen** of **uuidgen** opdrachten.
   - **LDAP Display Name**: De naam die gebruikt wordt voor LDAP. Deze naam is belangrijk voor programmatische toegang.
   - **Object Identifier**: De interne representatie van een object. Deze identifiers worden verleend door specialie autoriteiten, en zijn gegarandeerd uniek in alle netwerken over de hele wereld. Een Object Identifier bestaat uit een decimale reeks met punten, waarbij de toekenning op een hiërarchische manier gebeurd. Een Object Identifier kan aangevraagd worden bij een regionale ISO vertegenwoordiger. Indien dit niet gewenst is, kan er ook een Object Identifier gegenereerd worden in een Microsoft subtak, met behulp van de opdracht **oidgen**.

3. Bespreek de belangrijkste *kenmerken* van attributeSchema objecten, en op welke waarden die ingesteld kunnen worden.

   - **attributeSyntax en oMSyntax**: De syntax bepaalt het data type zoals: Object ID, Boolean, Integer, DirectoryString zijn enkele van de 26 mogelijkheden. Slechts 18 van deze 26 worden momenteel gebruikt in Active Directory. Het is **onmogelijk** om een nieuwe syntax te definieëren. Het object ID van een syntax wordt geïdentificeerd in de vorm van 2.5.5.x . Sommige Object Identifiers zijn blijkbaar niet te onderscheiden, waarop beroep moet gedaan worden op een bijkomende Integer waarde: mOSyntax.

- **rangeLower en rangeUpper**: Bepalen de lengte- of bereikbeperkingen van kenmerken.
- **isSingleValued**: Geeft aan of een attribuut meerdere waarden kan bevatten (een lijst).
- **searchFlags**: Dit veld bevat binaire informatie, waarbij elke afzonderlijke bit kan ingesteld worden. Veronderstel de vorm $b_{10}b_9b_9b_7b_6b_5b_4b_3b_2b_1$, dan betekent elke bit het volgende:
  - b_1: Deze wordt meestal gezet, zodat eenvoudige indexering van de waarde van het kenmerk geactiveerd wordt, ongeacht van waar het object zich in Acive Directory bevindt.
  - b_2: Indien deze gezet wordt, wordt de waarde van het kenmerk gecbomineerd met de identificatie van de container waarin het object zich bevindt. Dit heet een containerized index, en zijn in staat om snel objecten op te sporen in een specifieke container.
  - b_3: Dit laat Ambiguous Name Resolution toe. Bij opzoekingen van kenmerken waaraan een bepaalde waarde voldaan moet worden, kan i.p.v. (| (kenmerk1=waarde)(kenmerk2=waarde)(kenmerk3=waarde)...) gewoon (anr=waarde) gebruiken.
  - b_4: (staat niets over in de cursus)
  - b_5: Deze bit heeft niets met indexering te maken, maar geeft aan of de waarde van een attribuut behouden blijft bij het maken van een kopie van dit object.
  - b_6: Deze bit instellen versnelt opzoekingen waarn kenmerken metw wildcards vermeld worden. Deze tuple indexen worden best zeldzaam gebruikt, aangezien ze veel resources in beslag nemen.
  - b_7: niet belangrijk, dus niet opschrijven
  - b_8: (staat niets over in de cursus)
  - b_9: niet belangrijk, dus niet opschrijven
  - b_{10}: (staat niets over in de cursus)
- **systemFlags**: Dit heeft dezelfde vorm als searchFlags, een binair formaat $b_{28}...b_5b_4b_3b_2b_1$ waarvan de bits het volgende betekenen:
  - b_1: Deze bit geeft aan of dat het kenmerk gerepliceerd mag worden naar andere domeincontrollers of niet. Attributen die vaak wijzigen zoals lastLogOn en lastLogOff worden niet gerepliceerd.
  - b_2: (staat niets over in de cursus)
  - b_3: Dit geeft aan of een attribuut geconstrueerd is of niet. Een dergerlijk attribuut wordt niet opgeslagen in Active Directory, maar wordt telkens opnieuw berekend op basis van andere kenmerken.
  - b_5: _ToDo: opzoeken
  - b_28: _ToDo: opzoeken
- **isMemberOfPartialAttributeSet**: Bepaalt of een attribuut in de global catalog wordt opgenomen of niet.
- **linkID**: ??

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

   _ToDo: Oplossen

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)

    - De **objectClass** en **objectCategory** attributen laten toe om een klasse te achterhalen. Het attribuut objectClass

1.3 classSchema objecten (§2.2.4 en §2.2.6)

1. Bespreek het *doel* en de *werking* van classSchema objecten.

    - Voor elke klasse is er een classSchema object waarmee de klasse kan ingesteld worden.

2. Hoe benadert Active Directory het mechanisme van *overerving* ?

    - Een klasse die een andere klasse (de superklasse) overeft, neemt de kenmerken van deze superklasse over, inclusief de structuurregels en de inhoudsregels. Deze overerving werkt recursief: de subklasse erft alle gegevens van opeenvolgende superklassen. Een klasse kan echter maar van één superklasse overerven, en eventueel van speciaal hiervoor bestemde hulpklassen, die zelf geen instanties kunnen bevatten. Vanaf Windows Server 2003 is het mogelijk om dynamische objecten te definiëren. Hierdoor is het mogelijk om hulpklassen dynamisch te gebruiken: tijdens de create van een object kunnen extra dynamische hulpklassen gegenereerd worden, die enkel voor die instantie geldig zijn, door aanvullingen van het objectClass attribuut. Na de create kan dit attribuut niet meer gewijzigd worden.

3. Bespreek de diverse *naamgevingen*, specifiek voor classSchema objecten.

   komt overeen met vraag 1.2.2, maar enkel de naamgeving is anders

    - **cn**: De RDN van het attributeSchema object in de Schema container.
    - **schemaIDGUID**: Dit kan automatisch gegenereerd worden bij de creatie van een nieuw kenmerk. Een attribuut krijgt dan wel een verschillende GUID in verschillende forests. Manueel een GUID instellen kan ook, met bv de **guidgen** of **uuidgen** opdrachten.
    - **lDAPDisplayName**: De naam die gebruikt wordt voor LDAP. Deze naam is belangrijk voor programmatische toegang.
    - **governsID**: De interne representatie van een object. Deze identifiers worden verleend door specialie autoriteiten, en zijn gegarandeerd uniek in alle netwerken over de hele wereld. Een Object Identifier bestaat uit een decimale reeks met punten, waarbij de toekenning op een hiërarchische manier gebeurd. Een Object Identifier kan aangevraagd worden bij een regionale ISO vertegenwoordiger. Indien dit niet gewenst is, kan er ook een Object Identifier gegenereerd worden in een Microsoft subtak, met behulp van de opdracht **oidgen**.

4. Bespreek de belangrijkste *kenmerken* van classSchema objecten, en op welke waarden die ingesteld kunnen worden.

    -

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

   _ToDo: Oplossen

6. Hoe en met welke middelen kan het Active Directory schema uitgebreid worden ? Waarom moet je en hoe kan je hierbij *voorzichtig* te werk gaan ? (o.a. §2.2.8, ldifde fractie §2.2.3)

   - Wanneer onvoorzichtig uitbreidingen aan het schema worden toegevoegd, kan dit nefaste gevolgen hebben op domein en forestniveau. Een domein kan beschadigd geraken of zelfs uitschakelen. Bovendien gelden de wijzigingen voor het gehele forest.

     De veiligste manier om uitbreidingen op het schema te ontwikkelen en uit te testen is een geïsoleerd netwerk van een testomgeving. Bovendien worden schemaobjecten beveiligd door Access Control Lists, zodat enkel gemachtigde gebruikers aanpassingen kunnen doen. Verder kunnen ook nog volgende richtlijnen gehanteerd worden:

     – Vermijdt het wijzigen van de attributen van een bestaande klasse.
     – Maak enkel een nieuwe structurele klasse (een subklasse van de superklasse Top) aan als er geen enkel ander object enigszins aan de behoeften voldoet.

     Een schema uitbreiden heeft echter veel potentieel. De aanbevolen manier om het Active Directory op grote schaal uit te breiden is via de **ldifde** opdracht, aangezien één ldifde inputbestand meerdere aanpassingen ineens kan uitvoeren. Zo een inputbestand heeft als formaat het LDAP Data Interchange Format.

1.4 Active Directory domeinstructuren (§2.4.4, laatste paragraaf §2.4.5 en §2.4.6)

1. Wat is de bedoeling van *vertrouwensrelaties* ?

   - Tussen twee domeinen kan er een vertrouwensrelatie tot stand gebracht worden, zodat gebruikers in het vertrouwd domain kunnen geverifieerd worden door de domeincontroller in het vertrouwend domein. Dit wil niet zeggen dat de gebruiker toegang heeft tot de bronnen in dat domein. Windows Server maakt automatisch vertrouwensrelaties aan tussen domeinen en hun kinddomeinen. Deze kunnen niet verbroken worden en zijn bovendien bi-directioneel en transitief. Windows Server maakt ook automatisch vertrouwensrelaties aan tussen de trees van eenzelfde forest.

2. Bespreek de verschillende *soorten* vertrouwensrelaties.

   - **Forest vertrouwensrelatie.** Deze vertrouwensrelatie kan alleen bestaan tussen Windows Server domeinen in hetzelfde forest. Indien de diverse forests minimaal Windows Server 2003 functioneel niveau hebben, dan kan er een bidirectionele en trasitieve forest trust tussen de root-domeinen van deze forests gelegd worden.
   - **Realm vertrouwensrelatie.** Dit is een veralgemeening van een forest vertrouwensrelatie. Deze relatie kan gelegd worden tussen een Windows Server 2008 domein en een willekeurig kerberos v5 realm, onafhankelijk van het besturingssysteem waarop die geïmplementeerd zijn.

- **Verkorte vertrouwensrelatie.** Deze enkelvoudige of bi-directionele relatie kan gelegd worden tussen Windows Server domeinen binnen hetzelfde forest. Deze verkorte vertrouwensrelaties kunnen gebruikt worden om het vertrouwenspad in grote en complexe trees korter te maken. Praktisch is dit enkel nuttig indien het vertrouwenspad minstens een vijftal domeinen overspant, en dan uiteraard nog indien er frequesnt gebruik van gemaakt wordt.
- **Externe vertrouwensrelatie.** Dit is een enkelvoudige relatie waarbij één domein een ander vertrouwt. Verificatieaanvragen kunnen enkel van het trusting domein naar het trusted domein doorgegeven worden. Om een externe vertrouwensrelatie in twee richtingen te leggen, moeten er twee enkelvoudige relaties gelegd worden. Deze relatie is ook niet transitief.

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

   - Om een vertrouwensrelatie aan te maken, moet de domeinnamen en gebruikersaccount met machtigingen om vertrouwensrelaties in beide domeinen te maken, beschikbaar zijn. Een vertrouwensrelatie krijgt een wachtwoord toegewezen, dat bekend moet zijn bij de beheerders van beide domeinen van de vertrouwensrelatie. Dit wachtwoord wordt na het opzetten van de vertrouwensrelatie nooit meer gebruikt. Volgende optionele configuratiemogelijkheden bestaan ook:
     - **Selective Authentication.** Standard worden alle gebruikers van het trusted domein opgenomen in de Authenticated Users groep van het trusting domein. Via selective Authentication moet dit per individuele gebruiker of gebruikersgroep ingesteld worden.
     - **SID Filtering.** Indien SID Filtering aanstaat, wordt enkel rekening gehouden met de SID opgeslagen in het *objectSid* attribuut van de objecten in het trusted domein. Staat dit niet aan, dan verwerkt het trusting domein ook de SIDs opgeslagen in het sIDHistory attribuut.
   - Vertrouwensrelaties kunnen ook aangemaat worden in een *Command Prompt*, met behulp van de **netdom trust** opdracht. Om een overzicht te krijgen van alle vertrouwensrelatie en hun toestand, kan gebruik gemaakt worden van de **netdom query trust** opdracht.

4. Welke verschillen zijn er in praktijk tussen *NT 4.0* en *Windows Server* domeinstructuren ? Bespreek onder andere telkens de noodzaak om meerdere domeinen in te voeren. Bespreek de alternatieve mogelijkheden bij de *conversie van een NT 4.0 domeinstructuur* naar een Windows Server omgeving.

   ToDo: Oplossen

1.5 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):

1. Hoe bepaald wordt *welke servers* een dergelijke specifieke functie vervullen ? *Hoeveel* zijn er nodig (in termen van: *minimaal/exact/maximaal #*, *in functie van* ...), en waarom ?

   _ToDo: Oplossen

2. *Eigenschappen* zoals bedoeling, noodzaak, kriticiteit, inhoud, synchronisatie, voor welke Windows versie(s) van toepassing, ... ?

   _ToDo: Oplossen

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.

   _ToDo: Oplossen

4. Hoe kan achterhaald worden welk(e) toestel(len) de bijzondere functie vervult, en op welke diverse manieren men de *toewijzing* ervan kan instellen, wijzigen en/of ongedaan maken ?

   _ToDo: Oplossen

Hoofdstuk 2

Modelvragen theorie: reeks B

2.1 Active Directory functionele niveaus (§2.4.3)

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

   - **Windows 2000 mixed.** Een forest met dit functioneel niveau stelt geen enkele eis aan het functioneel niveau van de liddomeinen. Een domein met dit functioneel niveau biedt echter de laagste functionaliteit.
   - **Windows 2000 native.** Dit heeft geen impact op een forest. Op domeinniveau legt dit echter de beperking op dat een domeincontroller NT 5+ draait. Lidservers en werkposten hebben deze restrictie niet.
   - **Windows Server 2003.** Een forest met dit functioneel niveau kan enkel domeinen bevatten waarbij hun domein functioneel niveau minstens Windows Server 2003 is. Op domein functioneel niveau laat dit enkel nog Windows Server 2003+ domeincontrollers toe, zonder deze restrictie ook op te leggen aan lidservers en werkposten.
   - **Windows Server 2008.** Een forest met dit functioneel niveau kan enkel domeinen bvatten waarbij hun domein functioneel niveau minstens Windows Server 2008 is. Het biedt echter geen aanvullende functionaliteit. Analoog met Windows Server 2003, zal een domein met functioneel niveau Windows Server 2008 enkel Windows Server 2008+ domeincontrollers toelaten, zonder deze restrictie ook op te leggen aan lidservers en werkposten.

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

   Elk niveau kan ondergebracht worden in het domein functioneel niveau en forest functioneel niveau.

   - **Domein functioneel niveau.**
     - **Windows 2000 mixed.** Geen voordelen, standaardfunctionaliteit.
     - **Windows 2000 native.**

    - Er is de keuze om slechts één global catalog te hebben voor het hele forest, wat voor minder replicatie zorgt. De global catalog bevat een read-only en verkorte inhoudsopgave van elk domein in een groep domeinen. Hierdoor kunnen objecten opgezocht worden, zonder te weten in welk specifiek domein van de directory deze gegevens feitelijk zijn opgeslagen.
    - Transitieve vertrouwensrelaties tussen verschillende domeinen van eenzelfde forest zijn mogelijk. Twee domeinen kunnen een vertrouwensrelatie opstellen, zodat gebruikers in het vertrouwd domein kunnen geverifieerd worden door de domeincontroller in het vertrouwend domein.
    - Domeincontrollers zijn zelf in staat om SPN objecten aan te maken, hiervoor gedelegeerd door de RID master. De RID master is een serverrol dat slechts door één domeincontroller kan vervuld worden. Deze RID master geeft reeksen relatieve SIDs wanneer een domeincontroller zijn RID pool voor 80% heeft opgebruikt.
    - Gebruikers en/of computers kunnen verzameld worden in groepen. Een groep wordt gebruikt om een verzameling van gebruikersobjecten die dezelfde toegangsmachtigingen hebben te groeperen, zodat restricties enkel op deze groep moeten gedefinieerd worden en niet op de individuele gebruikersobjecten.
    - Alle SIDs die in een SPN object in het verleden gehad heeft, worden bijgehouden in het sIDHistory kenmerk.
  - **Windows Server 2003.**
    - Gebruik van aanvullende schema klassen en attributen.
    - Het veranderen van de naam van een domeincontroller, zonder degradatie en promotie.
    - Gebruik van aanvullende opdrachten zoals *redirusr* en *redircmp* om de default Active Directory containers te wijzigen waarin respectievelijk nieuwe gebruikers en nieuwe computers terechtkomen.
    - 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. Een universele groep is een groep die leden kan bevatten uit elk domein van het forest. Restricties op zo een groep is dan ook geldig op elk domein van het forest.
    - Filteren van group policies, nu niet alleen op basis van beveiligingsgroepen, maar ook met behulp van WMI scripts.
  - **Windows Server 2008.**
    - Opnieuw aanvullende schema klasse en attributen.
    - Encryptie van het Kerberos protocol met langere sleutels.
    - Fijnkorrelig wachtwoordbeleid, zodat wachtwoordrestricties niet langer globaal zijn voor het gehele domein, maar specifiek ingesteld kunnen worden voor individuele gebruikers of groepen.
    - Replicatie van DFS namespaces en van de SYSVOL share met behulp van DFS Replication.
- **Forest functioneel niveau.**
  - **Windows 2000 mixed.** Geen voordelen, standaardfunctionaliteit.
  - **Windows 2000 native.** Geen voordelen, standaardfunctionaliteit.

  - **Windows Server 2003.**
    * Het hergebruiken van gedactiveerde attributen en klassen.
    * Dynamische hulpklassen.
    * Dynamische objecten, met een beperkte levensduur.
    * Efficiënte replicatie van de global catalog gegevens.
    * Het veranderen van de naamgeving en de hiërarchische structuur van domeinen in een forest.
    * Transitieve vertrouwensrelaties tussen verschillende forests.
    * Read-only Windows Server 2008+ domeincontrollers.
    * Efficiëntere KCC algoritmen voor de herconstruering van de replicatietopologie.
    * Replicatie van de individuele waarden van multi-valued attributen.
  - **Windows Server 2008.** Geen extra functionaliteit ten opzichte van Windows Server 2003.

3. Hoe kan men detecteren op welk niveau een Active Directory omgeving zicht bevindt ?

   - Het attribuut **msDS-Behaviour-Version** geeft voor zowel op domeinniveau als forest-niveau aan welk functioneel niveau beschikbaar is.

4. Op welke diverse manieren kan men het functionele niveau verhogen of verlagen ?

   - Omschakelen naar een bepaald functioneel niveau gebeurd steeds manueel en kan enkel opwaarts: het is niet mogelijk om een hoger niveau om te vormen naar een lager niveau. Er moet steeds rekening gehouden worden met de restricties die elk niveau oplegt, indien deze niet voldaan zijn heeft het geen zin om te verhogen, en zal dit ook met een gepaste foutmelding getoond worden. Een niveau aanpassen kan enerzijds rechtstreeks, door de attributen van het domeinobject te manipuleren, ofwel via een GUI met behulp van de Active Directory Domains and Trust snap-in, beschikbaar in **domain.msc**.

2.2 Active Directory replicatie (§2.5)

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. Een domein zonder actieve domeincontrollers functioneert niet langer naar gebruikers toe. Doordat in één domein met meerdere domeincontrollers kan gewerkt worden, worden de fouttolerantie en de belastingsverdelingen verberterd.

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.

   - Active Directory maakt gebruik van multi-master replicatie, zodat de directory kan bijgewerkt worden vanaf elke domeincontroller, behalve read-only domeincontrollers. NT

4.0 daarentegen maakt gebruik van een master-slave model met primaire domeincontrollers en back-up domeincontrollers die gebruikt worden om de SAM gegevens, de policies, de gebruikersprofielen en de logon scripts te distribueren. In het master-slave model had slechts één enkele server, de primaire domeincontroller, een wijzigbare kopie van de directory. De andere domeincontrollers werden ingeschakelt voor het afhandelen van aanvragen, inclusief aanvragen van gebruikers voor wijzigingen. In huidige Windows Server systemen zijn alle Windows Server domeincontrollers equivalent. Dit biedt meer fouttolerantie omdat met meerdere domeincontrollers de replicatie kan voortgezet worden als één of meerdere andere domeincontrollers uitvallen.

- Een ander verschil met de replicatietechniek van NT 4 wordt store-and-forward replicatie genoemd. Elke verandering op een domeincontroller wordt slechts uitgewisseld met enkele ander domeincontrollers, die op hun beurt de wijzigingen communiceren met nog enkele andere domeincontrollers. Hoe de domeincontrollers weten naar welke andere domeincontrollers ze hun wijzigingen moeten doorsturen, gebeurd via de **KCC (Knowledge Concistency Checker)** software die op elke Active Directory domeincontroller beschikbaar is. De KCC's proberen steeds een topologie te vormen die ten minste twee paden met elke domeincontroller mogelijk maakt, waarbij elke controller maximaal met drie andere controllers rechtstreeks is verbonden, en waarbij het aantal hops tussen twee willekeurige domeincontrollers hoogstens drie is. Deze replicatietopologie wordt periodiek of via een trigger (bv een domeincontroller wordt toegevoegd) vernieuwd.
- Het laatste verschil tussen NT 4 en Windows Server is de kleinste replicatie eenheid. In NT 4 is dat altijd het volledig object. In forests met Windows 2000 functioneel niveau de volledige waarde van een individueel attribuut, en in forests met minimaal Windows Server 2008 functioneel niveau zelfs de atomaire waarde van het attribuut.
- Windows Server replicatie gebruikt een pull mechanisme. Elke domeincontroller brengt wel zijn intra-site replicatiepartners op de hoogte indien er wijzigingen in de eigen directory aangebracht zijn, maar hij stuurt deze wijzigingen niet op eigen intiatief door. Het opvragen van de gegevens wordt geïnitieerd door de replicatiepartners. Deze methode is zeer foutbestendig. Indien een update mislukt, vraagt de domeincontroller gewoon opnieuw om de gegevens. Verder wordt er niet bij elke wijziging de replicatiepartners verwittigd. De wijzigingen worden gegroepeerd in een bepaald tijdsinterval. Dit heeft als voordeel dat het netwerk minder belast zal worden. Dit wordt propagation damping of replication latency genoemd. Active Directory garandeerd daarom niet dat elke domeincontroller over de meest actuele directory beschikt, maar dat ze uiteindelijk wel zal convergeren, daarom noemt men dit een loose consistency model. Propagation damping is uitgeschakeld voor sommige attributen die te maken hebben met beveiliging. Gewijzigde lockout kenmerken en wachtwoorden worden bijvoorbeeld binnen de 15 seconden gerepliceerd. Dit noemt men urgent replication en wordt in eerste instantie met de PDC emulator uitgevoerd.
- Windows Server Replicatie is multi-threaded: een domeincontroller kan simultan repliceren met diverse partners. Bij replicatie tussen controllers in dezelfde site worden gegevens niet gecomprimeerd om de verwerkingskracht van de domeincontrollers minder te belasten.
- Een obstakel is het replicatieverkeer, dat moet relatief beperkt blijven. Daarom wordt in Active Directory enkel gewijzigde directory gegevens gerepliceerd. Om te voorkomen dat een domeincontroller meerdere malen dezelfde wijziging zal doorvoeren, wordt zijn **USN (Update Sequence Number)** verhoogd. De combinatie van USN en GUID van

een domeincontroller wordt zijn Up-To-Dateness Vector (UTD vector) genoemd. Elke domeincontroller meldt bij elke wijziging aan een object zijn huidige UTD vector aan de andere domeincontrollers, en houdt in de eigen UTD vectortabel de meest recente UTD vector bij die hij van elke andere controller heeft ontvangen, en waarvan hij de wijzigingen heeft verwerkt.

- Een ander probleem doet zich voor wanneer dat hetzelfde kenmerk van een object op meer dan één domeincontroller quasi tegelijkertijd gewijzigd wordt. De domeincontroller die conflicterende wijzigingen detecteert, lost dit op door naar het tijdstip van wijziging te kijken, en de wijziging met het oudere tijdstip te negeren. Zijn de tijdstippen gelijk, wordt enkel de wijziging van de domeincontroller met het hoogste GUID geaccepteerd. Voor sommige objecten, waarvoor de kans op conflicten klein is door bovenstaande methode, is dit nog altijd niet toelaatbaar, en wordt er beroep gedaan op slechts één domeincontroller, de operations master om de wijzigingen door te voeren.
- Men moet vermijden dat een object opnieuw gecreëerd wordt door replicatie wanneer deze verwijdert wordt. Daarom wordt een object niet onmiddelijk uit Active Directory verwijderd, maar eerst als tombstone gemarkeerd en in een hidden container, *Deleted Items*, geplaatst. Na 60 dagen (180 dagen in Windows Server 2008+ domeinen) wordt een object definitief verwijderd.

3. Welke toestellen repliceren onderling in een *forest* ? Welke specifieke gegevens worden hierbij uitgewisseld ?

   - Enkel de domeincontrollers repliceren onderling waarbij zowel de directory objecten en corresponderende kenmerken voor het domein, als het schema en de configuratiegegevens van het forest worden uitgewisseld.

4. Welke impact hebben *sites* met betrekking tot de replicatie van Active Directory gegevens ? Welke andere Active Directory aspecten worden door sites beïnvloed ? (§2.6.1)

   - Replicatiepartners van verschillende sites laten standaard het meldingsmechanisme van gewijzigde UTD vectortabellen achterwege met als gevolg dat er enkel polling kan toegepast worden. Het pollingsinterval staat standaard op 3 uur, maar kan ingekort worden tot 15 minuten.
   - Gegevens worden standaard gecomprimeerd, alhoewel men dit voor elk verbindingsobject kan uitschakelen.
   - Met behulp van sitekoppelingen kan aangegeven worden hoe de verschillende sites onderling, met rechtstreekse fysieke netwerkverbindingen, verbonden zijn. De KCC's genereren automatisch enkel verbindingsobjecten tussen sites als er tussen beide sites een sitekoppeling bestaat. De verbindingsobjecten kunnen dan de feitelijke netwerkverbindingen gebruiken om directory gegevens uit te wisselen.
   - Een aspect dat gewijzigd wordt is de KCC. Om te vermijden dat er meerdere verbindingsobjecten van dezelfde sitekoppeling gebruik maken, waarbij a priori willekeurige domeincontrollers gebruikt worden om informatie tussen de sites uit te wisselen, introduceert men de **ISTG (Inter-site Topology Generator)**. Enkele domeincontrollers krijgen de rol van ISTG, waardoor de functionaliteit van hun KCC wijzigd. De ISTG zorgt ervoor om voor elke partitie hoogstens één verbindingsobject per sitekoppeling aan te maken. De domeincontrollers die van dit uniek verbindingsobject gebruik maken worden

bruggenhoofd servers genoemd. Dit zijn de domeincontrollers waar gegevens tussen sites uitgewisseld worden. Elk paar site heeft dus zijn eigen paar domeincontrollers. Door deze invoering zorgt de ISTG ervoor dat de KCC niet de standaard intra-site richtlijnen volgt van de replicatietechnologie, maar zich optimaal zal aanpassen aan de siteconfiguratie.

5. Hoe wordt bepaald *tot welke site* computers, servers in het bijzonder, behoren ? (laatste paragraaf §2.6.2 en fractie §2.6.3)

   - Computers worden aan sites toegewezen op basis van de locatie in een IP subnet. Een site kan dus beschouwd worden als een verzameling computers in een of meer IP subnetten. Alle computers die geadresseed worden door hetzelfde IP subnet maken automatisch deel uit van dezelfde site.
   - Voor een server wordt de locatie bepaald door de vraag tot welke site in de directory het server object van de domeincontroller behoort. Elke site heeft een container met de naam Servers, die alle domeincontrollerobjecten bevatten die in deze site zijn geplaatst. Tijdens de promotie van een server tot domeincontroller wordt de server automatisch toegevoegd aan de site waaraan het subnet, waartoe de server behoort, is gekoppeld. De site van een server ligt hierdoor vast, en kan enkel manueel veranderd worden met **dssite.msc**.

2.3 Gedeelde mappen en NTFS

1. Welke *configuratieinstellingen* kun je maken tijdens of onmiddellijk na het creëren van gedeelde mappen ? Bespreek het *doel* van elk 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)

   _ToDo: Oplossen

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

   _ToDo: Oplossen

3. 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)

   _ToDo: Oplossen

2.4 Machtigingen op bestandstoegang (§3.3)

1. Welke rol spelen machtigingen bij de beveiliging van bronnen ? Geef een gedetailleerd overzicht van het *algemeen* (op alle windows objecten toegepast) mechanisme van *machtigingen*.

   _ToDo: Oplossen

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.

   _ToDo: Oplossen

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 ?

   _ToDo: Oplossen

4. Op welke *andere objecten* zijn machtigingen van toepassing ?

   _ToDo: Oplossen

5. Wie is *in principe* verantwoordelijk voor het configureren van machtigingen ? Door welke instelling is dit zo vastgelegd ? Hoe kan ervoor gezorgd worden dat enkel *administrators* verantwoordelijk gesteld worden voor het configureren van machtigingen ?

   _ToDo: Oplossen

2.5 Gebruikersgroepen (§4.2.2 en §4.2.3)

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.

   _ToDo: oplossen

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.

   _ToDo: Oplossen

3. Waar en hoe wordt het (volledige) lidmaatschap van een *user* object tot een groep bijgehouden ? Op welke diverse manieren kan men dit lidmaatschap *configureren* ? Op welke diverse manieren kan men de volledige verzameling van objecten, die deel uitmaken van een specifieke groep, of de volledige verzameling van groepen, waar een specifiek object deel van uitmaakt, achterhalen ? (partim §4.1.2 en §4.2.3)

   _ToDo: Oplossen

4. Door *wie* wordt het lidmaatschap van de diverse groeptypes bij voorkeur ingesteld ?

   _ToDo: Oplossen

5. Op welke diverse manieren kan men het beheer van Active Directory objecten, specifieke attributen van groepsobjecten in het bijzonder, *delegeren aan niet-Administrators* ? Bespreek een aantal technieken om dit delegeren *zo eenvoudig mogelijk* uit te voeren. (partim §4.1.2, en §4.4.2)

   _ToDo: Oplossen