De (on)beheersbaarheid van toegangsbeveiliging

Ing. P. Mienes RE en B. Bokhorst RE RA

Waarom wil het nog steeds maar niet lukken om op een efficiënte, effectieve en voor allen goed controleerbare manier te regelen dat toegang wordt verleend op basis van het 'need to access'-principe? Hoe komt het dat ondanks de vele medewerkers die bij dit proces betrokken zijn, auditors nog altijd tot te veel negatieve bevindingen op dit vlak komen? Wanneer komt de tijd dat de organisatie het bijzonder belangrijke proces van logische toegangsbeveiliging adequaat beheerst? Maar vooral: op welke manier kan aan de onbeheersbaarheid een einde worden gemaakt?

Inleiding

Onbeheersbaarheid van toegangsbeveiliging? Jawel, want het is alleszins gerechtvaardigd om te twijfelen aan het 'in control' zijn van deze beheertaak. En dat is niet verwonderlijk: met de klassieke autorisatiearchitecturen is het in de IT-infrastructuur van vandaag nagenoeg onmogelijk om te waarborgen dat autorisaties zijn verleend niet anders dan op basis van het 'need to access'-principe. Security administrators zijn zich bewust van de tekortkomingen, lijnmanagers weten ook dat het proces niet adequaat functioneert, IC-medewerkers raken hun werk moede, en auditors signaleren andermaal dat er 'ruimte is voor verbetering'. Hoe is dit na bijna dertig jaar bouwen aan automatisering mogelijk, en hoe zijn de te ruime autorisaties te verantwoorden tegenover de wet- en/of regelgever en de klanten van de organisatie?

Een veelgehoorde verklaring is dat het historisch zo is gegroeid. Dat is een onmiskenbaar feit, maar geen reden om ons neer te leggen bij het gebrek aan beheersing van de (met name logische) toegangsbeveiliging.

Vanwege
* een nog steeds toenemende graad van automatisering,
* en een daarmee steeds grotere afhankelijkheid van automatisering,
* toegenomen maatschappelijke aandacht voor privacy en fraude,
* toegenomen bedreigingen van buiten af,

en de niet afgenomen verantwoordelijkheid van het management van een organisatie om adequate maatregelen te treffen voor de interne en externe bedreigingen, is het meer dan ooit tevoren van belang dat organisaties hun toegangsbeveiliging daadwerkelijk en voor derden aantoonbaar op orde hebben.

Een relevante vraag hierbij is, in welke omgevingen van het OTAP-model (Ontwikkeling, Test, Acceptatie, Productie) sprake dient te zijn van een adequate toegangsbeveiliging. Doorgaans richt de grootste beveiligingsinspanning zich op het productiesysteem, en vertrouwt men erop dat het change-managementproces waarborgt dat uitsluitend integere applicaties op het productiesysteem worden geïmplementeerd. Als we echter bedenken dat business rules op het ontwikkelsysteem in de applicaties worden geprogrammeerd, dan zal duidelijk zijn dat elk van de omgevingen in het OTAP-model een zodanig niveau van beveiliging verdient dat de integriteit van de applicatie steeds gewaarborgd is. Tenzij in het acceptatietraject een volledige source code review plaatsvindt, begint adequate beveiliging dus al op het ontwikkelsysteem.

Dit artikel richt zich op de tactische en operationele beheersbaarheid van toegangsbeveiliging – met name het beheer van autorisaties – zodanig dat een adequaat beveiligingsniveau wordt bereikt en in stand gehouden op alle systemen en in alle omgevingen die goede beveiliging verdienen.

Achtereenvolgens komen de volgende vragen aan de orde:
* Wat is de problematiek van het huidige beheer van toegangsbeveiliging?
* Welke beveiligingsmodellen kunnen worden onderscheiden, en wat zijn de voor- en nadelen van de verschillende modellen?
* Welke voordelen biedt RBAC (Role Based Access Control) boven de andere modellen, en waarom zou dit de oplossing voor de geschetste problemen zijn?
* Welke lessen zijn geleerd uit eerdere RBAC-implementaties?
* Hoe wordt RBAC procedureel ingebed in het proces van toegangsbeveiliging?

De praktische onbeheersbaarheid van toegangsbeveiliging

Bij de beheersing van de toegangsbeveiliging zijn ten minste de volgende partijen betrokken:
* de eigenaar van het object waar de autorisatie betrekking op heeft;
* de aanvrager van autorisaties, veelal de manager van de betrokkene;
* de uitvoerende, deze implementeert de autorisatie;
* de controleur: internecontrolemedewerker of auditor.

Alle partijen stuiten in de praktijk op voor hen soms onoverkomelijke problemen bij de beheersing van de toegangsbeveiliging, omdat het autorisatiemodel en/of de toegepaste hulpmiddelen binnen het beheerproces de taken van deze partijen niet op een effectieve en efficiënte manier ondersteunen.

Probleem van sommige objecteigenaren

Als ze zich al bewust bezighouden met de beveiliging van hun objecten (er zijn positieve uitzonderingen!), dan is er een groep objecteigenaren die geen winst lijkt te behalen met het verbeteren van de opzet van de beveiliging, namelijk de eigenaren van applicaties. Vaak is het bijvoorbeeld in applicaties al zo geregeld dat gebruikers aan een autorisatieprofiel (≈ rol) worden gekoppeld, waarmee een voor de objecteigenaar redelijk beheersbare situatie bestaat.

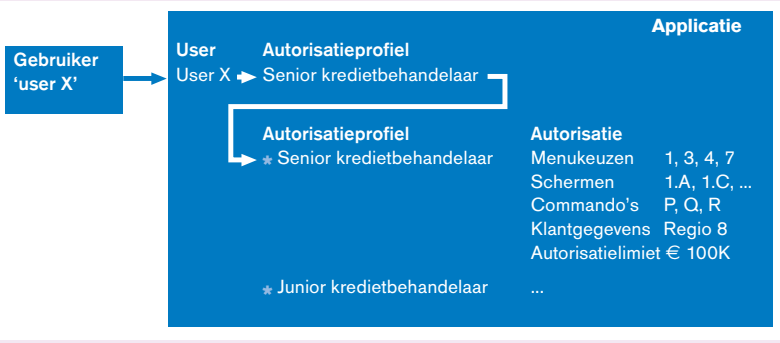
De koppeling van functionele rechten (autorisaties) aan de autorisatieprofielen is vastgelegd in de applicatie en vergt dus op het moment van een autorisatieaanvraag voor een gebruiker geen specifieke aandacht. De koppeling van gebruikers (user-id's) aan autorisatieprofielen vindt veelal plaats op aanvraag van de manager van de gebruiker en met goedkeuring van de objecteigenaar. Voor de eigenaar van dit type objecten werkt dit redelijk goed. Deze vertrouwt op de juiste beoordeling door de manager, en mag hopen dat deze te zijner tijd ook het vervallen van de noodzaak voor de autorisatie meldt ...

Terwijl de eigenaar van een applicatie het begrip 'rol' doorgaans heeft kunnen implementeren in de applicatie, ziet de eigenaar van objecten op het niveau van het operating system en systeembeheerpakketten (systeembestanden, systeemcommando's, utilities, etc.) zich geconfronteerd met het volgende. Op dit niveau is het begrip 'rol' veelal onbekend en niet zonder meer te implementeren. Als we daarbij bedenken dat het op mainframes niet ongebruikelijk is dat ongeveer tien objecteigenaren de verantwoordelijkheid hebben voor de beveiliging van vijftig- tot honderdduizend objecten, dan is duidelijk dat deze objecteigenaren een probleem hebben!

Probleem van de aanvrager

De manager van de gebruiker is de klos: deze moet vaak voor vele tientallen applicaties en voor tientallen servers autorisaties aanvragen, en daarnaast ook netwerkbevoegdheden en fysieke autorisaties. Het is dan aantrekkelijk om de uitvoerende organisatie te vragen de nieuwe medewerker van dezelfde autorisaties te voorzien als een bestaande user-id, van een collega die hetzelfde werk uitvoert. Deze manier van klonen is voor het moment wel effectief, maar op de lange duur niet: vaak heeft de bestaande user-id nog autorisaties vanuit een vorige functie, waardoor de nieuwe medewerker al direct met een te ruime set autorisaties begint.

Voor medewerkers van de exploitatieafdeling van het rekencentrum (systeemprogrammeurs, database administrators, applicatiebeheerders, operators, etc.) is de situatie voor de autorisatieaanvragers vaak nog lastiger te overzien. In die omgeving hebben de medewerkers veelal zelf een sterk sturende rol bij het aanvragen van de



Figuur 1. Voorbeeld van autorisatietabellen in een applicatie.

autorisaties en periodieke controle op toegang tot belangrijke systeembestanden vindt zelden plaats (het valt ook bijna niet te controleren: een exploitatiemedewerker in een mainframe-omgeving heeft vaak duizenden autorisaties, die toegang geven tot tienduizenden objecten).

Probleem in de uitvoering

Het is niet ongebruikelijk dat in een grote organisatie (enkele duizenden medewerkers) tien of meer personen betrokken zijn bij het effectueren van de autorisatieaanvragen voor één nieuwe medewerker of één medewerker die van functie verandert. De applicaties en platforms waar autorisaties moeten worden geregeld, hebben immers verschillende beheerders of security administrators. De coördinatie van al deze aanvragen en gereedmeldingen is bij grote organisaties in een aparte functie belegd. Deze medewerker beoordeelt de aanvragen op juistheid en volledigheid en distribueert de aanvragen naar de verschillende uitvoerders. Na terugkoppeling door de uitvoerders meldt de coördinator de aanvraag als gereed terug bij de manager die de aanvraag indiende.

Hoe zijn de te ruime autorisaties te verantwoorden tegenover de wet- en/of regelgever en de klanten van de organisatie?

Door de betrokkenheid van het grote aantal partijen duurt de succesvolle verwerking van de aanvraag soms wel een week tot enkele weken! Dit geldt vooral voor aanvragen die bij nader inzien toch incompleet waren door het niet vermelden van enkele benodigde autorisaties.

Het operationeel beveiligingsbeheer wordt in de uitvoering geconfronteerd met een nieuw probleem: het zijn niet langer alleen de eigen medewerkers die toegang hebben tot de IT-infrastructuur, maar in toenemende mate ook externe partijen, zowel ingehuurd personeel als gebruikers via externe koppelingen. De hiervoor benodigde autorisaties zijn dynamischer dan de autorisaties voor het 'vaste' personeel. Dergelijke autorisaties kunnen uit efficiency- en tijdigheidsoverwegingen niet meer op de traditionele manier worden beheerd. Andere oorzaken van de dynamiek in de beveiliging zijn reorganisaties en fusies, die beide veel frequenter voorkomen dan

vroeger, toen de beveiligingsmodellen werden ontwikkeld. Welke grote organisaties zijn in staat binnen een jaar de gevolgen van een reorganisatie of fusie te verwerken in de regels voor logische toegangsbeveiliging? Wij kennen er geen. En vaak staat binnen een jaar de volgende reorganisatie al weer voor de deur ...

De oncontroleerbaarheid

Of het nu de objecteigenaar is, de manager, de IC-medewerker of de IT-auditor, elke schakel in het beveiligingsproces is bijzonder geholpen met het bestaan van autorisatiematrices waarin de Soll-positie van de autorisaties is vastgelegd. Een simpele vergelijking met de Ist-positie levert een uitzonderingsrapportage aan de hand waarvan verbeteringen kunnen worden aangebracht.

Autorisatiematrices: *soms* bestaan ze, *vaak* zijn ze niet actueel (met name niet volledig), en *altijd* zijn er verschillen in functie- of rolaanduidingen tussen de diverse autorisatiematrices omdat ze specifiek voor één applicatie tot stand zijn gebracht. Als er al autorisatiematrices bestaan van (applicatiespecifieke) rollen en rechten, dan ontbreekt veelal nog de matrix waarin weergegeven is welke medewerkers welke rol(len) hebben. Hierdoor is een automatische controle op de juistheid van verstrekte autorisaties (de Ist-positie) niet mogelijk.

Kortom: de problematiek van het beheer van toegangsbeveiliging

Het beheer van toegangsbeveiliging is niet gemakkelijk, en het is dus ook helemaal niet verwonderlijk dat er vaak veel mis mee is. Objecteigenaren, aanvragers, uitvoerders en controleurs: ze ervaren allemaal de beheerproblematiek van toegangsbeveiliging. En een oplossing denken ze soms ook gevonden te hebben: rollen. Maar *hebben* ze werkelijk *de* oplossing gevonden?

Automatische controle op de juistheid van verstrekte autorisaties (de Ist-positie) is meestal niet mogelijk.

In het bovenstaande is enkele keren aangegeven dat hier en daar al met rollen wordt gewerkt: inderdaad is dat een beperkte implementatie van Role Based Access Control. Immers, de rechten in de applicatie zijn gekoppeld aan rollen en op hun beurt zijn deze rollen gekoppeld aan user-id's. Op deze manier wordt bepaald welke user-id's welke rechten hebben binnen de applicatie. Het is vanzelfsprekend dat applicaties gebruikmaken van profielen of het begrip 'rol' om rechten aan te koppelen: vijfentwintig jaar geleden al had men door dat dit voor de applicatieontwikkelaar en de administrator veel werk scheelt.

Ook op een andere plek wordt vaak al met rollen gewerkt: de slimme manager (of gedelegeerde) die een autorisatieaanvraag moet opstellen, heeft doorgaans wel een document of spreadsheet met daarin een opsomming van de benodigde autorisaties voor een bepaald type medewerker (rol) in het team.

Omdat het concept van een 'rol' zo vanzelfsprekend is, wordt het eigenlijk overal wel gebruikt, meer of minder bewust, echter zonder de voordelen van het concept ten volle te benutten. Feit is namelijk dat bij audits de volgende bevindingen steevast naar voren komen:

* te ruime autorisaties;
* uiteenlopende autorisatieprocessen;
* onduidelijkheid over aan te vragen en te verstrekken autorisaties;
* onvoldoende beheersbaarheid (waaronder controleerbaarheid) vanwege:
  - geen volledige vastlegging Soll-situatie rollen versus rechten;
  - geen vastlegging Soll-situatie medewerkers versus rollen;
  - geen adequaat overzicht Ist-situatie.

Kennelijk is het zo dat de huidige toepassing van het rolconcept onvoldoende bijdraagt tot een goed beheerste beveiliging. Hoe kan dat? Enkele van de oorzaken daarvoor zijn hierboven weergegeven. Aanvullend kunnen de volgende punten worden onderkend als oorzaken:

* Er bestaat onvoldoende uniformiteit in rolaanduidingen: per applicatie bestaat een unieke set rollen.
* Het toepassingsgebied van de rolaanduidingen is te beperkt: bijvoorbeeld alleen binnen of vlak rond de applicatie; HRM (Human Resource Management) hanteert weer andere functie/rolaanduidingen.
* De diverse autorisatiematrices zijn op vele plaatsen vastgelegd: indien er n verschillende autorisatiematrices zijn, dan zijn er >n vastleggingen.
* Op het niveau van het operating system (OS/390, Unix, NT, OS/400) bestaat het concept van een 'rol' niet of zeer beperkt.

Door de wijze waarop de autorisatiematrices zijn vastgelegd (als ze al bestaan), is het veelal niet haalbaar om een automatische confrontatie met de Ist-situatie uit te voeren, laat staan dat het mogelijk is om de Ist-situatie automatisch (bij) te sturen op basis van de autorisatiematrices.

Organisatie en implementatie van de beveiliging volgens de methode van Role Based Access Control lost – afhankelijk van de reikwijdte en diepgang van de implementatie – de genoemde problemen voor een belangrijk deel op. Als aanloop naar RBAC worden onderstaand de verschillende in de praktijk gehanteerde autorisatiemodellen belicht.

Autorisatiemodellen

Autorisatiemodel-0

In dit model is een gebruiker (geïdentificeerd door een user-id) rechtstreeks geautoriseerd voor toegang tot een object, meestal via de Access Control List (ACL) van het object. Op de ACL van het object is dan gespecificeerd welke toegang (bijvoorbeeld lezen, schrijven, uitvoeren, verwijderen) de user-id heeft tot het object. Het object kan een bestand zijn, commando, utility, directory, een applicatie, maar ook een autorisatieprofiel of transactie binnen een applicatie of een scherm binnen een systeembeheertool.

Opmerking: De lijn met kraaienpoten representeert een n:m (veel-op-veel) relatie tussen de entiteiten. In dit geval betekent het dat een gebruiker aan meer dan één object kan zijn gekoppeld, en dat anderzijds een object aan meer dan één gebruiker kan zijn gekoppeld.

In figuur 3 is in een voorbeeld een tiental van de honderden tot duizenden autorisaties weergegeven die een gebruiker 'user X' heeft.

Dit model bestaat op vrijwel elk platform (OS/390, Unix, NT, OS/400), en wordt ook wel gehanteerd in applicaties en systeembeheertools. Een voordeel van dit model is de eenvoud waarmee een enkele autorisatie kan worden toegekend aan de gebruiker. Een groot nadeel is dat het onderhoud veel inspanning kost: elke nieuwe medewerker moet steeds aan alle benodigde objecten worden gekoppeld – dat zou in theorie duizenden commando's kunnen vergen. Als de medewerker een andere functie krijgt, moeten de verbindingen met de niet meer benodigde objecten stuk voor stuk worden verbroken.

Autorisatiemodel-1 (groepen)

In dit model is een gebruiker geautoriseerd voor toegang tot een object via een autorisatiegroep. Dit houdt in dat de group-id, als identificerend kenmerk van de autorisatiegroep, op de ACL van het object staat, en dat de gebruiker lid is van (gekoppeld is aan) de groep.

Dit model bestaat op de meeste platforms, en wordt vaak toegepast in applicaties voor het groeperen van bevoegdheden.

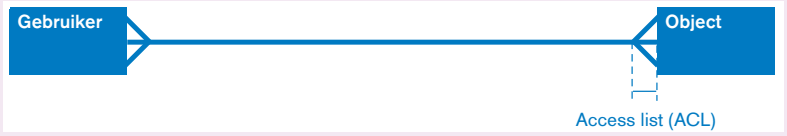
De voordelen van autorisatiemodel-1 in vergelijking met autorisatiemodel-0 worden in het algemeen onderkend, hetgeen de reden is dat autorisatiemodel-1 vandaag de dag het meest toegepaste model is.

De autorisatiegroep in dit model kan worden opgevat als een groepering van gebruikers, maar ook als een groepering van autorisaties. Beide zienswijzen zijn correct, en in de praktijk zijn dit de twee benaderingen bij het ontwerpen van groepen:

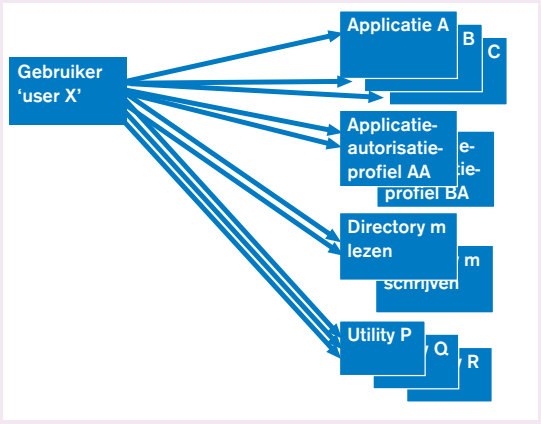
* *Rolgebaseerde autorisatiegroepen* (figuur 5). Gebruikers zijn gekoppeld aan een groep (of groepen) die de rol(len) representeert die de gebruikers hebben binnen de organisatie. Deze benadering leidt gewoonlijk tot een situatie waarbij de gebruiker gekoppeld is aan een beperkt aantal autorisatiegroepen, en de autorisatiegroepen op de ACL's van vele objecten staan. Kenmerkend is ook dat op de ACL's vele autorisatiegroepen staan. In een poging de lengte van de ACL (het aantal group-id's op de ACL) te beperken worden ook wel 'basis' autorisatiegroepen gecreëerd. Dit zijn groepen die de gemeenschappelijk benodigde autorisaties realiseren voor bijvoorbeeld:
  - rollen binnen een afdeling;
  - medewerkers op een bepaalde locatie;
  - vaste medewerkers (versus ingehuurd personeel).

* *Taakgebaseerde autorisatiegroepen* (figuur 6). Gebruikers zijn gekoppeld aan groepen die taken representeren die de gebruikers moeten kunnen uitvoeren vanwege hun rol(len) binnen de organisatie. Deze benadering leidt tot een situatie waarbij de gebruiker gekoppeld is aan veel autorisatiegroepen, en de autorisatiegroepen op de ACL's van een beperkt aantal objecten staan. Op de ACL's staat slechts een gering aantal autorisatiegroepen.

De figuren 5 en 6 geven voorbeelden van de beide benaderingen. Het is van belang om te bedenken dat deze voorbeelden simplificaties van de werkelijkheid zijn. De pijlen onderaan in de figuren geven aan waar de realiteit uitgebreider is.

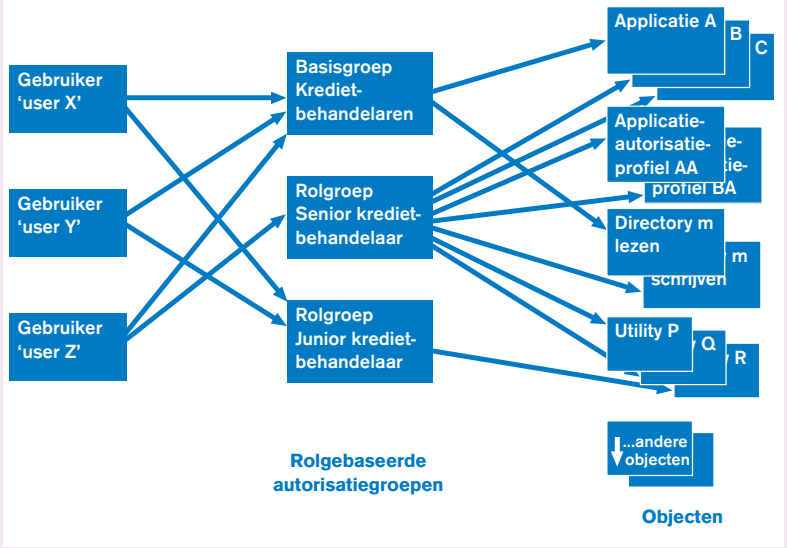


Figuur 2. Autorisatiemodel-0.

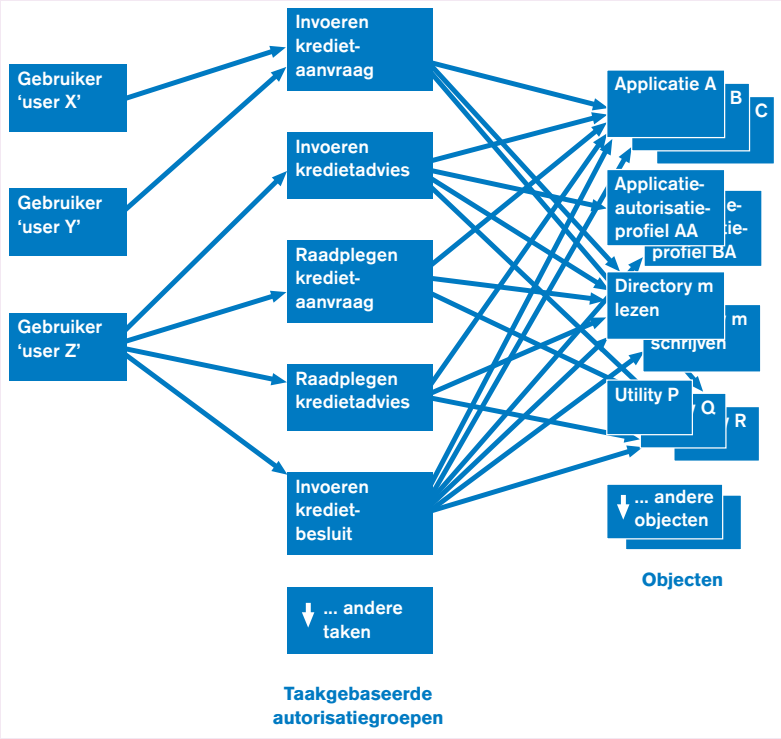


Figuur 3. Voorbeeld van autorisaties volgens autorisatiemodel-0.

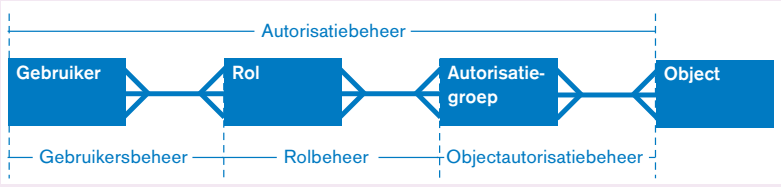
Figuur 4. Autorisatiemodel-1.



Figuur 5. Relaties tussen gebruikers, rolgebaseerde autorisatiegroepen en objecten.



Figuur 6. Relaties tussen gebruikers, taakgebaseerde autorisatiegroepen en objecten.



Figuur 7. Autorisatie-model-2.

Autorisatiebeheerders die claimen reeds een RBAC-implementatie te hebben – refererend aan de rolgebaseerde autorisatiegroepen – hebben niet helemaal ongelijk: autorisatiemodel-1 in combinatie met rolgebaseerde autorisatiegroepen is feitelijk een rudimentaire RBAC-implementatie: de gehanteerde rolbenamingen beperken zich echter tot een specifiek platform of een specifieke applicatie. In het geval dat ook op andere platforms of in andere applicaties rolgebaseerde autorisatiegroepen worden gehanteerd, worden daar bijna altijd andere namen of een andere detaillering van rollen gebruikt. Bovendien blijken de gedefinieerde rolgebaseerde autorisatiegroepen vaak bij nadere beschouwing feitelijk afdeling- of teamgebaseerde autorisatiegroepen te zijn: dit is dodelijk voor de vlotte aanpassing van autorisaties bij reorganisaties! Afdelingen en teams veranderen, rollen vrijwel niet.
Daarnaast is het koppelen van een gebruiker aan een rolgebaseerde autorisatiegroep een technische handeling in een technische omgeving, terwijl het verstrekken van een autorisatie gedreven wordt vanuit de business. Een gebruiker dient niet te worden gekoppeld aan technische dingen, maar aan iets wat kan worden herkend (en gecontroleerd!) vanuit de business: een rol.

De noodzaak voor een platform- en applicatieoverstijgende en (re)organisatieonafhankelijke invulling van het begrip 'rol' begint zich af te tekenen ...

Autorisatiemodel-2 (RBAC)

In dit model is een gebruiker geautoriseerd voor toegang tot een object via een rol en een autorisatiegroep. Dit houdt in dat net als in autorisatiemodel-1 de group-id van de autorisatiegroep op de ACL van het object staat. Aan de andere kant is de autorisatiegroep nu gerelateerd aan één of meer rollen, en is een gebruiker gerelateerd aan één of meer rollen.

Dit is het basis-RBAC-model: gebruikers zijn niet langer rechtstreeks gekoppeld aan autorisatiegroepen. In plaats daarvan komt de relatie tot stand via roldefinities. Op hun beurt zijn de rollen gekoppeld aan taakgebaseerde autorisatiegroepen.

In de regel wordt dit model niet ondersteund op platformniveau, zodat het begrip 'rol' alleen kan bestaan als administratieve entiteit binnen de grenzen van een RBAC-tool.

Opmerking: In het formele RBAC-model wordt niet over autorisatiegroepen gesproken, maar over permissies. Het model is namelijk niet beperkt tot systemen die het begrip (autorisatie)groep kennen: het kan worden toegepast in elk systeem dat al dan niet gegroepeerde rechten/privileges/permissies/autorisaties kent. In dit artikel wordt echter voor de eenvoud het concrete begrip autorisatiegroep gehanteerd.

Autorisatiemodel-2 (RBAC) versus autorisatiemodel-1

In tabel 1 wordt de meest effectieve RBAC-implementatie vergeleken met de meest effectieve niet-RBAC-implementatie:
* autorisatiemodel-2 (RBAC) met taakgebaseerde autorisatiegroepen, en
* autorisatiemodel-1 met rolgebaseerde autorisatiegroepen, het model dat vandaag de dag vaak wordt gehanteerd.

Uit de in tabel 1 gegeven vergelijking van de beide modellen blijkt dat autorisatiemodel-2 (RBAC) op alle fronten leidt tot een betere beheersing van de beveiliging.

Voorbeeld: controleerbaarheid
Waarom is het controleren van beveiliging op basis van autorisatiemodel-1 inefficiënt en kan het zelfs ineffectief blijken te zijn? Een simpel voorbeeld verduidelijkt dit.

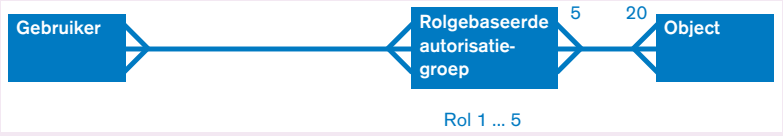
Stel, op basis van een risicoanalyse rond de beheersing van kredietaanvragen is de auditor geïnteresseerd in de beveiliging van een twintigtal objecten die te maken hebben met het raadplegen van een kredietaanvraag. Als eerste vraagt de auditor een overzicht van de twintig ACL's. Laten we aannemen dat vijf rollen toegang hebben tot de twintig objecten om een kredietaanvraag te kunnen raadplegen.

Aspect	Autorisatiemodel-2 (RBAC) met taakgebaseerde groepen (figuur 7)	Autorisatiemodel-1 met rolgebaseerde groepen (figuur 5)
Objectautorisatiebeheer	Omdat de groepen taakgerelateerd zijn, is de objecteigenaar goed in staat de juistheid van de toegang tot het object te beoordelen. De betekenis van de taakgroep wordt namelijk inzichtelijk gemaakt door een functionele beschrijving (attribuut van de taakgebaseerde groep).	De objecteigenaar moet kennis hebben van alle rollen die toegang vragen tot het object, hetgeen vrijwel niet mogelijk is op een betrouwbare en efficiënte manier.
Rolbeheer	Taakgebaseerde groepen worden gekoppeld aan rollen. Deze koppelingen kunnen worden vastgesteld door een manager/teamleider (die de betekenis van de rol goed kent) in samenspraak met een beveiligingsbeheerder (als gedelegeerde van de verschillende objecteigenaren). Hierbij gebruiken zij de hierboven aangehaalde functionele beschrijving van de taakgebaseerde groepen.	Rolbeheer maakt in feite deel uit van het object-autorisatiebeheer.
Gebruikersbeheer	Een gebruiker is gerelateerd aan een rol. Het begrip 'rol' sluit uitstekend aan bij organisatorische begrippen als functie en taak(omschrijving), zodat het koppelen van een gebruiker (user-id) aan een rol goed kan worden overgelaten aan een HRM-medewerker, die immers up-to-date informatie heeft over de rol (!) van de betrokken medewerker in de organisatie.	Een gebruiker is gerelateerd aan één of meer rolgebaseerde autorisatiegroepen. Normaliter voert een security administrator deze technische taak uit, of in een ongunstig geval een systeembeheerder.
Functiescheiding	De functiescheiding in het beveiligingsbeheer is optimaal, omdat het voor de hand ligt om: * gebruikersbeheer, * rolbeheer en * objectautorisatiebeheer door verschillende medewerkers uit te laten voeren.	Over het algemeen verre van optimaal, waarbij alle beveiligingsbeheertaken door de security administrator of de systeembeheerder worden uitgevoerd.
	In een geavanceerde RBAC-implementatie kan ook worden bepaald welke rollen (in de gebruikersorganisatie) uit het oogpunt van functiescheiding niet aan één gebruiker mogen worden toegekend.	Het aan één gebruiker toekennen van rolgebaseerde autorisatiegroepen die uit het oogpunt van functiescheiding onverenigbaar zijn, kan in de regel niet automatisch worden verhinderd.
Beheersbaarheid	Omdat autorisaties worden bepaald door relaties tussen de entiteiten gebruiker - rol en rol - autorisatiegroep, is er een maximum aan transparantie en een minimum aan complexiteit.	Het grote aantal rolgebaseerde groepen op de ACL's bemoeilijkt het verkrijgen van een overzicht.
	Als een medewerker een andere rol krijgt, worden de oude autorisaties op een eenvoudige manier automatisch verwijderd door het RBAC-tool.	Bij het wijzigen van rollen is speciale aandacht nodig voor het (handmatig) verwijderen van oude autorisaties.
Controleerbaarheid	Dankzij het transparante inzicht in de relaties gebruiker - rol en rol - autorisatiegroep is de controle van beveiliging op basis van RBAC relatief eenvoudig.	De controle van autorisaties volgens dit model is in het algemeen inefficiënt of (daardoor) zelfs mogelijk ineffectief (zie voorbeeld in de volgende subparagraaf).
		Vanwege het grote aantal rolgebaseerde groepen op de ACL kan het voor een objecteigenaar/auditor moeilijk zijn om de juistheid van de autorisaties te beoordelen.
		Vanwege het grote aantal ACL's waar de rolgebaseerde groep op is geplaatst, en het technische detailniveau van de betrokken objecten, is het voor een manager/auditor praktisch onmogelijk de juistheid van de autorisaties van de medewerkers te beoordelen.
Multi-platform ondersteuning	Het concept van een 'rol' bestaat als entiteit binnen de grenzen van een RBAC-tool, en betreft autorisaties op alle platforms binnen het werkingsgebied van het RBAC-tool. Integriteit (uniformiteit) van rolnamen is gegarandeerd.	Op de verschillende platforms vindt men in de regel verschillende afzonderlijke implementaties van dit model. Integriteit van rolnamen is niet gegarandeerd, hetgeen de efficiëntie en effectiviteit van het multi-platform beveiligingsbeheer vermindert.
Afdwingen beveiligingsbeleid	Een RBAC-tool kan eenvoudig afdwingen dat niemand méér autorisaties heeft dan de autorisaties die gerelateerd zijn aan de rol die men heeft.	Zonder solide procedures kunnen tools die dit model implementeren niet voorkomen dat gebruikers aanvullende autorisaties verkrijgen.

Tabel 1. Vergelijking RBAC met niet-RBAC.

In figuur 8 is de beveiliging ingericht volgens autorisatiemodel-1 met rolgebaseerde autorisatiegroepen. De auditor moet zich een oordeel vormen over de juistheid van de aanwezigheid van de vijf rolgebaseerde autorisatiegroepen op elk van de twintig ACL's. Dit betekent dat honderd ACL-regels (ofwel relaties tussen autorisatiegroepen en objecten) moeten worden beoordeeld.

Figuur 8. Auditautorisaties voor de taak 'Raadplegen kredietaanvraag' (autorisatiemodel-1).



Figuur 9. Auditautorisaties voor de taak 'Raadplegen kredietaanvraag' (autorisatiemodel-2, RBAC).

In het geval dat autorisatiemodel-2 wordt gebruikt (met taakgebaseerde autorisatiegroepen), zouden alle twintig autorisaties die nodig zijn voor het uitvoeren van de taak 'Raadplegen kredietaanvraag' gecombineerd zijn in één taakgebaseerde autorisatiegroep, zoals in figuur 9 weergegeven.
In deze situatie zou de auditor slechts vijfentwintig relaties hoeven te beoordelen:

* de toegang die de autorisatiegroep 'Raadplegen kredietaanvraag' heeft tot de twintig objecten, en
* de relaties van deze groep tot de vijf rollen.

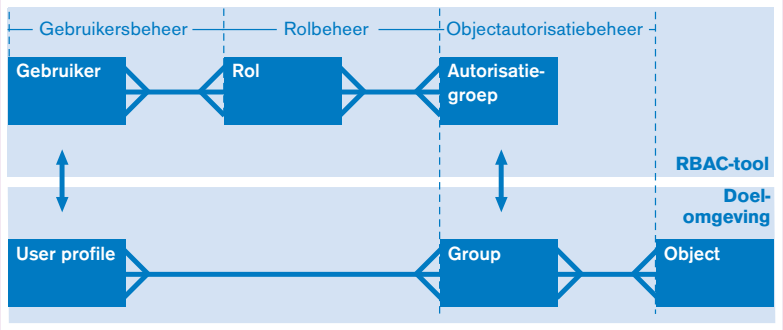
Dit sterk vereenvoudigde voorbeeld geeft de problematiek aan van de controle van autorisatiemodel-1 (met rolgebaseerde groepen) en illustreert dat het auditen (en dus ook het beheer!) van autorisatiemodel-2 (RBAC) belangrijk eenvoudiger is vergeleken met de andere modellen. Het feit dat controle van autorisatiemodel-1 qua omvang snel uit de hand kan lopen vormt ook een reële bedreiging voor de effectiviteit van de controle.

Concrete RBAC-implementatie

Kwestie van beperken

Bij een inventarisatie van RBAC-initiatieven en een verkenning van de markt van RBAC-producten is de kans groot ondergesneeuwd te raken door termen als identity management, provisioning, single signon (SSO), directories, Public Key Infrastructure (PKI), Privilege Management Infrastructure (PMI), Security Assertion Markup Language (SAML), Enterprise Access Management (EAM), etc. Deze veelheid aan concepten kan de RBAC-sponsor in de organisatie de moed in de schoenen doen zinken. Menig leverancier denkt zoveel mogelijk features en gerelateerde pakketten te moeten noemen in de brochures en white papers, omdat het product het anders mogelijk zal afleggen tegen de andere producten met nog meer features en interfaces.

Figuur 10. Mapping van entiteiten tussen het RBAC-tool en de doelomgeving.



Wat voor elk ander project geldt is ook waar voor een RBAC-implementatie: houd het simpel en beperkt. Een eenvoudige RBAC-implementatie, zelfs al is het een papieren implementatie, is vaak voorwaarde voor – of gaat samen met – de bovengenoemde concepten.
Niet voor niets blijven RBAC-initiatieven veelal steken op de ontwerptafel: de ambities zijn te groot en men wil in één keer tachtig procent, liever nog negentig procent 'onder RBAC' brengen. Het is echter goed mogelijk dat bij specifieke RBAC-implementaties de 80/20-regel opgaat: tachtig procent ROI (Return On Investment) met slechts twintig procent van de oplossing.

Mapping van entiteiten

Bij de implementatie van RBAC is het van belang onderscheid te maken tussen entiteiten binnen het RBAC-tool en entiteiten in de doelomgevingen (platforms, applicaties, etc.). Zoals eerder aangegeven bestaat de entiteit 'rol' slechts binnen het RBAC-tool zelf. Anderzijds zal het objectautorisatiebeheer veelal buiten het RBAC-tool plaatsvinden: het plaatsen van een groep op de ACL van een Unix-file, het zetten van een RACF PERMIT op een OS/390-dataset, het koppelen van een transactie in een financiële applicatie aan een autorisatieprofiel, etc.

Figuur 10 geeft een beeld van de begrippen in praktische RBAC-implementaties.

Terwijl de entiteit 'rol' slechts binnen het RBAC-tool bestaat, hebben de entiteiten 'Gebruiker' en 'Autorisatiegroep' een tegenhanger in de doelomgeving: de gebruiker wordt geprojecteerd (gemapt) op een user profile of account, terwijl de autorisatiegroep een representant heeft in de vorm van bijvoorbeeld een RACF-, Unix- of NT-group.

Gebruikersbeheer en rolbeheer vinden dus plaats binnen het RBAC-tool. Vanuit dit tool kunnen (half)automatisch de wijzigingen in de (relaties tussen) user profiles en groups in de doelomgevingen worden doorgevoerd (= provisioning). Objectautorisatiebeheer wordt veelal buiten het RBAC-tool gehouden en in de doelomgeving uitgevoerd, als vanouds.

RBAC-tools

Het bieden van een compleet marktoverzicht van RBAC-tools valt buiten het bestek van dit artikel. Als altijd is de pakkettenmarkt in beweging, en zal een organisatie zelf tot een keuze moeten komen op basis van een zorgvuldig samengesteld pakket van eisen en wensen, en de actuele marktsituatie. Niettemin kunnen, zonder een volledig overzicht te geven, enkele namen worden genoemd

van bekende producten op dit terrein. Het zijn producten die autorisatiebeheer (op basis van RBAC) op de meest populaire platforms ondersteunen, zoals Windows 2000, HP-UX, AIX, OS/390, en in toenemende mate ook in systemen als Lotus Notes en SAP:
* AccessMaster (Evidian);
* AxcessIT (SafeStone);
* *b*hold suite (*b*hold company);
* Control-SA (BMC);
* DirXmetaRole (Siemens);
* eTrust Admin (Computer Associates);
* IBM Tivoli Identity Manager (IBM).

Methode

Eén van de grootste uitdagingen bij de implementatie van RBAC is het definiëren van de rollen en per rol de benodigde autorisaties. Indien de autorisaties zijn gegroepeerd in rolgebaseerde autorisatiegroepen, dienen deze te worden omgevormd tot taakgebaseerde autorisatiegroepen. Onderstaand worden de methoden belicht om te komen van een situatie waarin autorisatiemodel-1 wordt gebruikt met voornamelijk rolgebaseerde autorisatiegroepen tot een situatie met autorisatiemodel-2 met voornamelijk taakgebaseerde autorisatiegroepen.

Onderscheid kan worden gemaakt tussen twee methoden voor het ontwerpen van rollen:
* *top-down- of groene-weidemethode*. Hierbij wordt uitgegaan van de organisatiestructuur, beveiligingsbeleid, functiebeschrijvingen en procesbeschrijvingen om te komen tot een acceptabel detailniveau van roldefinities; een tweede – bijzonder lastige – stap is het bepalen van de autorisaties die nodig zijn voor de aldus totstandgekomen rollen.
* *bottom-up- of 'role mining'-methode*. Hierbij wordt uitgegaan van bestaande autorisaties en aan de hand van interviews met managers vastgesteld welke rollen binnen een afdeling of team bestaan.

Een hybride aanpak is ook goed mogelijk: een combinatie van de top-down en bottom-up methoden, waarbij bijvoorbeeld
* een aantal rollen volgens de ene methode wordt ontwikkeld en een aantal andere volgens de andere methode, en/of
* rollen initieel worden bepaald volgens de bottom-up methode en vervolgens worden aangepast op basis van inzichten die volgen uit de top-down methode.

Bij de RBAC-implementaties die KPMG IRM tot nu toe heeft begeleid – onder andere bij Belastingdienst/Centrum voor ICT (B/CICT) – is steeds bewust gekozen voor de bottom-up methode: zij levert relatief snel resultaat en bij het bepalen van de benodigde autorisaties voor de rollen kan gebruik worden gemaakt van geautomatiseerde hulpmiddelen. Een belangrijke overweging is ook dat het vaak jaren heeft gekost om tot een zodanige set autorisaties te komen dat alles goed functioneert: door geen kennis te nemen van de bestaande autorisaties, met andere woorden door de Ist-situatie te negeren, duurt een RBAC-traject veel langer dan nodig, of worden onnodige beschikbaarheidsrisico's gelopen (omdat bepaalde, toch benodigde autorisaties over het hoofd zijn gezien).

KPMG IRM gebruikt MS-Access om inzicht te krijgen in de actuele autorisaties van gebruikers. Na het in Access importeren van gegevens over gebruikers, (rolgebaseerde) autorisatiegroepen en ACL's van de objecten is het mogelijk op een efficiënte manier inzicht te krijgen in de bestaande relaties tussen deze entiteiten. Met behulp van Access worden relaties gevisualiseerd, waardoor het eenvoudig is om patronen vast te stellen in:
* relaties tussen gebruikers en de in interviews benoemde rollen;
* taakgerelateerde autorisaties, die worden samengebracht in nieuwe taakgebaseerde autorisatiegroepen.

Bij RBAC-implementaties levert de bottom-up methode relatief snel resultaat.

Op basis van dit inzicht wordt de bestaande situatie volgens autorisatiemodel-1 omgevormd tot een situatie volgens autorisatiemodel-2. Gegevens over de nieuwe situatie worden ingevoerd in het RBAC-tool, waarna het betreffende deel van de organisatie of van de IT-infrastructuur onder besturing van het RBAC-tool wordt gebracht.

Procedurele inbedding

Bij het implementeren van het RBAC-beveiligingsmodel dient bovenal ook aandacht te worden geschonken aan de procedurele inbedding van het onderhoud aan entiteiten en relaties. Ten minste de volgende taken dienen te worden belegd:

* *Beheer RBAC-tool:* opvoeren, autoriseren en afvoeren van RBAC-gebruikers (zijnde gebruikersbeheerders, rolbeheerders, objectautorisatiebeheerders, auditors, etc.).

* *Gebruikersbeheer:* opvoeren/afvoeren van gebruikers en het koppelen van gebruikers aan rollen. Deze taak kan worden gedecentraliseerd, zodat men gebruikersbeheerder is voor een deel van de gebruikers en/of een deel van de rollen. De bevoegdheden van een decentrale beheerder worden ingesteld door de beheerder van het RBAC-tool.

* *Rolbeheer:* opvoeren/afvoeren van rollen, het vastleggen van een functionele beschrijving van de rol en het koppelen van rollen aan autorisatiegroepen. Deze taak kan worden gedecentraliseerd, zodat men rolbeheerder is voor een deel van de rollen en/of een deel van de autorisatiegroepen. De bevoegdheden van een decentrale beheerder worden ingesteld door de beheerder van het RBAC-tool.

* *Objectautorisatiebeheer:* In het RBAC-tool: opvoeren/afvoeren van autorisatiegroepen en het vastleggen van een functionele beschrijving van de autorisatiegroep. In de doelomgeving: opvoeren/afvoeren van groups en het koppelen van groups aan objecten. Deze taak kan worden gedecentraliseerd, zodat men objectautorisatiebeheerder is voor een deel van de objecten en/of autorisatiegroepen. De bevoegdheden van een decentrale beheerder worden enerzijds ingesteld door de beheerder

Ing. P. Mienes RE
is senior IT-consultant/ auditor bij KPMG Information Risk Management en is intensief betrokken bij de logische toegangsbeveiliging en het systeembeheer in grote organisaties. Voor KPMG participeert hij onder andere in het Role Based Access Control-project van EEMA.

B. Bokhorst RE RA
is werkzaam als security manager bij de Belastingdienst/Centrum voor ICT. Daarnaast is hij als bestuurslid van het Platform Informatiebeveiliging belast met de begeleiding van werkgroepen op het gebied van beveiligingsnormering.

Een korte versie van dit artikel zal worden gepubliceerd in het periodiek 'Informatiebeveiliging' van het GVIB, nummers 2003-2 en 2003-3.

Gebeurtenis	Betrokkenheid		
	Gebruikers-beheer	Rolbeheer	Objectautori-satiebeheer
a. Medewerker in dienst	V		
b. Medewerker uit dienst	V		
c. Medewerker krijgt (andere) rol	V		
d. In organisatie ontstaat nieuwe rol		V	A
e. Nieuwe functionaliteit toegevoegd aan bestaande (taakgebaseerde) autorisatiegroep		I	V
f1. Nieuwe autorisatiegroep opgericht voor nieuwe functionaliteit			V
f2. Nieuwe taakgroep beschikbaar gesteld aan rol of meerdere rollen		V	A

V Verantwoordelijk A Adviserend I Geïnformeerd

Tabel 2. Verdeling van verantwoordelijkheden.

van het RBAC-tool, en zijn voor een ander deel afhankelijk van toegekende autorisaties in het doelsysteem.

In tabel 2 is voor een aantal administratiegerelateerde gebeurtenissen aangegeven wie hiervoor verantwoordelijk zijn (V), een adviserende functie hebben (A), of geïnformeerd (I) dienen te worden.
De in de tabel genoemde gebeurtenissen hebben naast wijzigingen binnen het RBAC-tool ook wijzigingen tot gevolg in de doelomgeving:

a. In de doelomgeving wordt een user profile aangemaakt.
b. In de doelomgeving wordt de user profile geblokkeerd, dan wel de user profile wordt verwijderd en de relaties tussen de user profile en de groups worden verbroken.
c. In de doelomgeving worden relaties gelegd tussen de user profile en de groups; bij een rolwijziging worden nieuwe relaties gelegd, en de oude relaties met groups worden verbroken voorzover deze niet overlappen met de relaties die nodig zijn voor de nieuwe rol.
d. Het begrip 'rol' bestaat alleen binnen het RBAC-tool, dus deze gebeurtenis heeft geen gevolgen voor de doelomgeving.
e. De bestaande group in de doelomgeving wordt op de ACL van (nieuwe) objecten geplaatst.
f1. Een nieuwe group wordt opgericht in de doelomgeving en op de ACL van (nieuwe) objecten geplaatst.
f2. In de doelomgeving worden relaties gelegd tussen de nieuwe group en alle user profiles die gekoppeld zijn aan de betrokken rol(len).

Bij toepassing van een volwaardig RBAC-tool kunnen de wijzigingen a, b, c, d en f2 (half)automatisch worden doorgevoerd in de doelomgeving. De wijzigingen e en f1 worden in de regel buiten het RBAC-tool om (handmatig) in de doelomgeving uitgevoerd.

Praktijkervaringen

Het NIST (National Institute of Standards & Technology) voert reeds tien jaar onderzoek uit naar RBAC, en komt daarbij tot conclusies die gezien mogen worden. Uit een onderzoek door het NIST bij een verzekeringsmaatschappij kwam naar voren dat tegenover de eenmalige kosten van $78 per medewerker voor de invoering van RBAC $43 per medewerker aan jaarlijkse besparingen staat. De besparingen worden vooral gevonden in de sterk vereenvoudigde administratie van de beveiliging en in afname van improductiviteit als gevolg van het niet tijdig of niet goed realiseren van de benodigde autorisaties.

Frans Calor, projectleider Security Management bij B/CICT: 'De belangrijkste voordelen van RBAC die direct na de invoering zichtbaar waren, zijn enerzijds de sterke vereenvoudiging voor de teamleiders die autorisaties aanvragen, en anderzijds de verbeterde controleerbaarheid, onder meer doordat nu een automatische Soll/Ist-controle mogelijk is. Inmiddels zijn OS/390-autorisaties, fysieke rechten en hoge rechten op NT en Novell onder het RBAC-regime gebracht, en zijn we bezig om de doelgroep uit te breiden van vierhonderd naar vierduizend medewerkers.'

Theo Krens begeleidde vanuit KPMG IRM de implementatie bij een andere organisatie, en memoreert: 'Het belangrijkste probleem dat RBAC bij deze klant heeft opgelost is de beheersbaarheid van de logische toegangsbeveiliging op VME, Unix en NT. Door de invoering van RBAC is behalve de operationele beheersbaarheid ook de controleerbaarheid sterk verbeterd, en er is transparantie ontstaan in de beveiligingsregels. RBAC was ook een stimulans om de procedures rond logische toegangsbeveiliging te vervolmaken.'

Conclusie

Met een eenvoudig rekenschema kan elke organisatie een betrouwbaar beeld schetsen van de kwantitatieve voordelen van de invoering van RBAC. Daarnaast kunnen ook kwalitatieve aspecten een belangrijke rol spelen bij de beslissing om tot (gedeeltelijke) invoering van RBAC over te gaan. Zelfs een qua scope en tooling beperkte RBAC-implementatie kan reeds significante voordelen bieden voor de administratie en controle van de beveiliging.