KPMG Klynveld EDP Audit

Samsom BedrijfsInformatie

COMPACT

KWARTAALBLAD EDP AUDITING

LENTE

1990 / 1

Compact ®
Jaargang 17, nummer 1
Een uitgave van KPMG Klynveld EDP Audit en Samsom BedrijfsInformatie, werkmaatschappij van Wolters Kluwer NV.
Het blad verschijnt 4 x per jaar.

Redactie
D. Steeman RA (hoofdredacteur)
Drs. R.G.A. Fijneman RA
Mw. D. Jansen Heijtmajer RI
A.W. Neisingh RA
Drs. P. Veltman RA

Redactiesecretariaat
Mw. A.M.F. Hofland,
Klynveld EDP Audit,
K.P. van der Mandelelaan 41
3062 MB Rotterdam
Tel.: 010 - 453 47 40
Fax : 010 - 453 47 77

Vormgeving
Bureau Karakter, Delft

Aan dit nummer werkten mee
A.A.J. Breed / Mw. G.J.C. Heikamp / Ing. J.F. Kuperus / Ing. G.H.M. Meijer / Drs. P. Veltman RA / Drs. Ing. J.C. van Winkel RI

Abonnementen
f 185,- per jaar incl. BTW.Losse nummers *f* 50,- incl. BTW. Abonnementen kunnen schriftelijk tot uiterlijk één maand voor de aanvang van een nieuw abonnementsjaar worden opgezegd. Bij niet tijdelijke opzegging wordt het abonnement automatisch met een jaar verlengd.

Abonnementsadministratie
Samsom BedrijfsInformatie
Postbus 4
2400 MA Alphen aan den Rijn
Tel.: 01720 - 6 68 00
Fax : 01720 - 7 59 33

Adreswijzigingen - ook tijdelijke - moeten minstens 8 weken voor de verschijningsdatum bekend zijn.

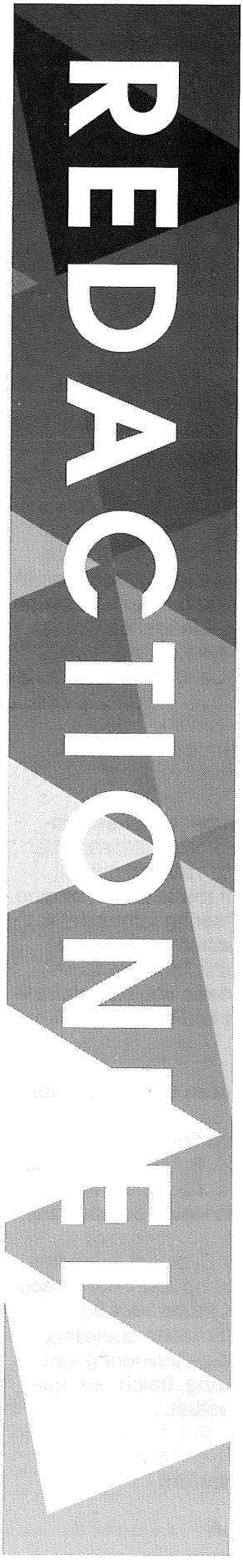
Overname artikelen
Het overnemen en vermenigvuldigen van artikelen en berichten is slechts geoorloofd na schriftelijke toestemming van de uitgever.

Uitgever
H.B. Plas

Lid van de Nederlandse organisatie van tijdschriftuitgevers NOTU

ISSN 0920 - 1645

Inhoudsopgave



Redactioneel

U hebt op dit moment een exemplaar van de eerste vernieuwde uitgave van Compact in handen. Voor KPMG Klynveld EDP Audit is dit een mijlpaal in de geschiedenis. Reeds een aantal jaren werd er van verschillende kanten bij ons op aangedrongen om aan onze voor interne doeleinden bestemde publikatie een wijdere verspreiding te geven. Steeds hebben wij dit afgewezen omdat wij de verplichting om een regelmatig verschijnend periodiek te verzorgen als te belastend voor onze organisatie beoordeelden.

Na rijp beraad hebben wij besloten om de uitdaging te aanvaarden. Na zestien jaren en vijftig Compact-nummers steekt het blad in een volslagen nieuw jasje.

Wat de redactie met dit blad voor ogen staat, is het presenteren van artikelen over het vakgebied EDP-auditing in de ruime betekenis. Dat wil zeggen dat de lezer artikelen kan verwachten over kwaliteitsaspecten van informatietechnologie en over de beoordeling daarvan. Kwaliteit is een algemeen begrip. Het kan nader worden omschreven in termen van betrouwbaarheid, beveiliging, continuïteit, effectiviteit en efficiency. Maar ook in termen als controleerbaarheid, robuustheid, gebruikersvriendelijkheid, onderhoudbaarheid, functionaliteit, etc.

Wij hopen dat Compact in het Nederlandse taalgebied in een behoefte zal voorzien zowel voor lezers als voor schrijvers. Informatie-uitwisseling over het vakgebied zal het niveau van de EDP-auditing verhogen. Potentiële schrijvers stellen wij gaarne in staat hun bijdrage te leveren.

De intentie van uitgever en redactie is om een blad te verzorgen van circa 48 pagina's. Het eerste nummer over besturingssystemen gaat dit aantal te boven omdat recht moest worden gedaan aan de thans in gebruik zijnde belangrijkste systemen.

Wij hopen dat het eerste nummer van Compact in deze nieuwe versie als een interessante bijdrage kan worden beschouwd aan de literatuur over EDP-auditing. De redactie ziet het commentaar van de lezers gaarne tegemoet.

D. Steeman RA

Het blad wil een bijdrage leveren aan de ontwikkeling van het vakgebied EDP-auditing door het publiceren van actuele artikelen op deelterreinen van EDP-auditing en advies, zoals:

- beoordeling automatiseringsorganisaties en -systemen
- risicobeheersing
- telecommunicatie-adviezen
- beveiligingsonderzoeken
- quality assurance
- opleidingen en trainingen
- privacy-wetgeving
- computercriminaliteit en nieuwe regelgeving

Behalve voor EDP-auditors kan dit blad ook interessant zijn voor EDP-deskundigen en gebruikers van informatiesystemen.

De in dit tijdschrift weergegeven meningen mogen niet worden gezien als officiële zienswijze van KPMG Klynveld EDP Audit.

Het blad Compact is met de meeste zorg samengesteld. Niettemin is het niet geheel uitgesloten dat de geboden informatie enkel en alleen door tijdsverloop en/of andere oorzaken minder juist is.

Noch KPMG Klynveld, KPMG Klynveld EDP Audit, noch de redacteuren persoonlijk, noch uitgeverij Samsom Bedrijfs-Informatie bv, deel uitmakend van Wolters Kluwer NV, aanvaarden enige aansprakelijkheid, hoe ook genaamd, uit welken hoofde dan ook voor enig gevolg rechtstreeks of indirect voortvloeiend uit het gebruik van de informatie.

De redactie stelt gaarne ruimte in Compact beschikbaar voor reacties en/of ervaringen van lezers.

Auteurs die overwegen een bijdrage te leveren, wordt verzocht kennis te nemen van de aanwijzing voor auteurs, die bij het secretariaat verkrijgbaar is.

Het operating system kan worden beschouwd als de manager van de resources die te zamen de hardware-infrastructuur van de geautomatiseerde gegevensverwerking vormen. Door deze spilfunctie is het operating system van grote betekenis voor de betrouwbaarheid en beveiliging daarvan. Een onderzoek naar de implementatie van een operating system gericht op deze aspecten, kan derhalve van groot belang zijn voor een organisatie.

Drs. P. Veltman RA

De audit van operating systems

1 Inleiding

In computers vormt het operating system het hart of zenuwcentrum van het gehele systeem voor geautomatiseerde gegevensverwerking. De functie ervan is enigszins vergelijkbaar met die van een verkeersagent die op een druk kruispunt het verkeer in goede banen moet zien te leiden.

Operating systems maken onderdeel uit van elke computer, of het nu gaat om een chip card of om een "number crunching" super computer.

In het verleden hebben vooral de performance-eigenschappen van computers en operating systems de aandacht gehad (kloksnelheid, MIPS, adresseringsmogelijkheden en wat dies meer zij). De vergelijking met de verkeersagent doortrekkend, gaat het hierbij om de snelheid van doorstroming van het verkeer.
Van meer recente datum is de belangstelling voor betrouwbaarheids- en beveiligingsaspecten. Het verkeer moet ter bestemder plekke aankomen en de doorstroming moet met zo min mogelijk ongelukken gepaard gaan.
Enkele fabrikanten hebben bijgevolg een stevige marktpositie weten te verwerven door te voorzien in een behoefte aan computers die gekenmerkt worden door een hoge mate van betrouwbaarheid (ongevoeligheid voor storingen).
De onmiskenbaar toegenomen belangstelling voor beveiliging komt onder andere tot uitdrukking in aankondigingen van leveranciers van computerapparatuur en -programmatuur om de beveiliging van hun produkten op een hoger niveau te tillen.
Dit beleid is gestimuleerd doordat de afnemers van computers zich steeds meer zorgen maken over de kwetsbaarheid van de geautomatiseerde gegevensverwerking (voor computercriminaliteit, virussen, etc.) en de mate waarin zij van de computers afhankelijk zijn.

Het doel van dit artikel is aan te geven hoe een onderzoek naar de implementatie van een operating system kan worden aangepakt. Hiertoe zal eerst, met grove penseel, een schets worden gegeven van de eigenschappen van operating systems die vanuit het oogpunt van betrouwbaarheid en beveiliging van belang zijn.
Tevens zal worden ingegaan op de maatregelen die in de organisatie waar de computer wordt gebruikt, zouden moeten zijn getroffen om te waarborgen dat het operating system op de juiste wijze wordt geïnstalleerd en gehanteerd.
In dit artikel zal slechts terloops aandacht worden besteed aan andere aspecten dan betrouwbaarheid en beveiliging, en zal in hoofdzaak de multi-user (mini- of mainframe-)omgeving de aandacht hebben.

2 Enkele begrippen

Een computer, te zamen met de randapparatuur, kan worden beschouwd als een verzameling (hulp)middelen of "resources" voor de verwerking van gegevens. Het operating system of besturingssysteem kan dan worden gezien als de manager van deze resources, die onder meer de volgende taken voor zijn rekening neemt:
-- "scheduling": het bijhouden van een rooster met taken of "jobs" die moeten worden uitgevoerd en het opstarten en beëindigen daarvan;
-- het toewijzen van resources, bijvoorbeeld intern geheugen en verwerkings-

capaciteit (tijd van de Central Processing Unit - CPU) aan de jobs;
-- het lezen en schrijven van data van en naar randapparatuur (zoals printers en disk- en tape-units).

Dankzij het operating system is de gebruiker in staat de mogelijkheden van de fysieke apparatuur te benutten; de combinatie van hardware en operating system resulteert als het ware in een virtuele machine (niet te verwarren met het operating system "Virtual Machine" van IBM), waarbij de eigenlijke, fysieke eigenschappen van de machine zijn vertaald in een voor de gebruiker toegankelijke vorm.

Het operating system maakt deel uit van de meer omvattende verzameling van systeem- of besturingsprogrammatuur. Onder systeemprogrammatuur wordt hier verstaan alle programmatuur die geen applicatieprogrammatuur is. Applicatieprogrammatuur is de programmatuur die rechtstreeks is gericht op de verwerking van door de eindgebruiker ingevoerde gegevens.
Deze definities leveren geen scherpe afbakening op tussen systeemprogrammatuur en applicatie- of toepassingsprogrammatuur. De grens is dan ook niet nauwkeurig te trekken. Zo zijn in een batch-omgeving bepaalde taken in de applicatieprogrammatuur opgenomen die in een on-line-omgeving zijn overgebracht naar wat wordt beschouwd als een onderdeel van de systeemprogrammatuur.

Applicatieprogrammatuur omvat toepassingen zoals het grootboeksysteem en het salarissysteem.
Systeemprogrammatuur vormt te zamen met de hardware de infrastructuur waarbinnen de applicaties worden uitgevoerd.
Afhankelijk van de leverancier en het type computer bestaat de systeemprogrammatuur uit een min of meer geïntegreerd geheel van de volgende componenten:
-- het operating system of besturingssysteem zelf, ook wel "supervisor" genoemd;
-- programmatuur ter ondersteuning van telecommunicatie en on-line-gegevensverwerking;
-- programmatuur voor de ondersteuning van logische bestandsstructuren. Een geavanceerde uitvoering van dergelijke programmatuur vormt het Data Base Management System (DBMS);
-- bibliotheekprogramma's;
-- hulpprogrammatuur voor programma-ontwikkeling. Programmacode, maar ook "Job Control Language" (JCL) kan met behulp hiervan eenvoudig worden ingebracht en gewijzigd;
-- vertalers, zoals "compilers" en "linkage editors";
-- hulpprogrammatuur voor "job scheduling", "job accounting", etc.;
-- toegangsbeveiligingsprogrammatuur;
-- hulpprogramma's of "utilities" voor performance-metingen, foutdetectie, foutherstel, etc.

Bepaalde componenten,zoals het DBMS worden ook wel subsystemen van het operating system genoemd.

3 Betrouwbaarheid en beveiliging

Het operating system, in combinatie met de hardware en de overige systeemcomponenten, dient te zorgen voor een betrouwbare en beveiligde infrastructuur voor de gegevensverwerking. Deze infrastructuur dient bovendien effectief te zijn, dat wil zeggen: aan te sluiten op het behalen van de doelstellingen van de organisatie als geheel, en efficiënt: gegeven de doelstellingen tegen zo laag mogelijke kosten.

Betrouwbaarheid
In het kader van dit inleidende artikel zal onder betrouwbaarheid van een operating system worden verstaan de mate waarin het in de praktijk (bij gebruik) aan de specificaties voldoet. Een operating system is betrouwbaarder naarmate het minder gauw storingen (afwijkingen van de specificaties) vertoont. Beveiliging, hier gedefinieerd als de maatregelen gericht op de wering van onbevoegden, is niet identiek aan betrouwbaarheid, maar kan er wel toe bijdragen.

Betrouwbaarheid zoals hier gedefinieerd heeft een relatie met de begrippen beschikbaarheid en continuïteit. Continuïteit geeft een dynamisch aspect aan en heeft betrekking op de voortgang van het gegevensverwerkend proces. Beschikbaarheid heeft een statisch karakter en betreft de resources die voor het gegevensverwerkend proces nodig zijn.
Betrouwbaarheid heeft een ruimere betekenis dan uitsluitend beschikbaarheid. Bij de afwijkingen van de specificaties gaat het niet alleen om het uitvallen van het totale systeem (het niet meer beschikbaar zijn), maar ook om afwijkingen (storingen) als:

-- het terechtkomen in een eindeloze programma-"loop", die ingrijpen van de operator noodzakelijk maakt;
-- het verloren gaan van een directory (terwijl de bestanden bewaard blijven). Herstel is dan met behulp van utilities wellicht nog mogelijk;
-- concurrent update, waardoor een mutatie verloren gaat.

Storingen worden veroorzaakt door fouten en kunnen op hun beurt de oorzaak zijn van schade, die weer nieuwe fouten tot gevolg kan hebben. Fouten vinden hun oorsprong in de hardware (bijvoorbeeld stof op een lees-/schrijfkop), software ("bugs" in zowel systeem- als toepassingsprogrammatuur) of in handelingen van gebruikers of operators.

In de hardware van computers is veelal een groot aantal voorzieningen opgenomen voor foutdetectie en -herstel, zoals:
-- toepassing van een pariteitsbit, in combinatie met herverzending als een fout is geconstateerd;
-- duplicatie van bepaalde bewerkingen;
-- "majority polling" of "voting", waarbij een bepaalde bewerking ten minste drie maal wordt uitgevoerd en als resultaat het meerderheidsstandpunt wordt genomen.

Foutdetectie en -correctie op het niveau van het operating system is, evenals bij het pariteitsbit, gebaseerd op redundante informatie, aan de hand waarvan de juistheid van data kan worden vastgesteld en eventueel correctie kan plaatsvinden. Voor foutcorrectie is meer redundantie nodig dan voor enkel foutdetectie, hetgeen weer van invloed is op de performance.
Fouten behoeven niet altijd te leiden tot storingen. Herhaalde pogingen om informatie weg te schrijven op een extern geheugen kunnen uiteindelijk resulteren in een succesvolle schrijfactie, zonder dat de gebruiker iets merkt van de mislukte pogingen. Niettemin is het van belang dat dergelijke fouten worden vastgelegd en geanalyseerd, teneinde te voorkomen dat het aantal fouten onaanvaardbaar groot wordt en leidt tot storingen (bijvoorbeeld lange responstijden of zelfs "abends" (abnormal ends) als de outputhandeling te lang duurt). Tegenwoordig verschaffen de meeste leveranciers een faciliteit voor on-line "remote diagnostics/maintenance", waarbij logging, analyse en verhelping van fouten plaatsvindt door de leverancier.

Indien een fout wel heeft geleid tot een storing, is het belangrijk dat tijdig herstel kan plaatsvinden zonder (te veel) verlies van gegevens. Het mechanisme voor herstel wordt "error recovery" genoemd en is weer gebaseerd op de vastlegging van redundante informatie ("checkpoints", back-up-gegevens). Bij een ernstige storing, waarbij geen onmiddellijk herstel kan plaatsvinden, dient het uitvallende systeem zoveel mogelijk statusinformatie vast te leggen, om herstel in de toekomst te vergemakkelijken ("graceful degradation").

Computers die speciaal zijn ontworpen om een hoge mate van betrouwbaarheid te bieden, worden aangeduid als "fault tolerant"- of "non-stop"-computers. Hierbij zijn extra voorzieningen opgenomen in de hardware en/of software voor foutdetectie en -correctie. Deze computers worden met name gebruikt voor toepassingen waarbij hoge eisen worden gesteld aan de ongevoeligheid voor storingen.

Beveiliging
De mate van beveiliging die een operating system biedt, is van grote invloed op de betrouwbaarheid, daar bij een goede beveiliging de kans op het optreden van al dan niet opzettelijk veroorzaakte fouten wordt verkleind en daarmee de kans op storingen.
Beveiliging is, evenals betrouwbaarheid, een relatief begrip; geen enkel operating system is in absolute zin betrouwbaar of veilig. Anders dan voor betrouwbaarheid zijn voor beveiliging echter verscheidene, min of meer objectieve maatstaven of criteria beschikbaar.

De bekendste criteria voor de beoordeling van de beveiligingsmogelijkheden van operating systems (al dan niet in combinatie met speciale beveiligingsprogrammatuur) zijn afkomstig van het Amerikaanse National Computer Security Center (NCSC). Deze "trusted computer system evaluation criteria" zijn gebaseerd op de klasse-indeling van het (Amerikaanse) Department of Defense en vastgelegd in wat bekend staat als het "Orange Book". Van laag naar hoog worden de volgende klassen onderscheiden: D C1 C2 B1 B2 B3 A1.
Tot voor kort hadden de meeste operating systems en beveiligingspakketten voor multi-user-omgevingen een beveiligingsklasse C (1 of 2). De grote leveranciers hebben nu echter aangekondigd te streven naar indeling in de B-klasse,

waarin enkele reeds zijn geslaagd. Het voornaamste verschil tussen C en B is dat voor de B-klasse beveiligingslabels moeten worden gehanteerd, met behulp waarvan toegangscontrole wordt afgedwongen.

Beveiliging dient zich uit te strekken tot alle resources, waaronder het interne geheugen. Tussen de verschillende processen die in het interne geheugen actief zijn, moeten als het ware muurtjes worden opgetrokken om ongewenste beïnvloeding over en weer te voorkomen. De meeste processen moeten echter om legitieme redenen kunnen communiceren met andere processen; een applicatie die gegevens nodig heeft die op externe geheugens zijn opgeslagen, moet bijvoorbeeld een verzoek kunnen richten aan de "device handler" om deze gegevens op te halen.

Om ongewenste interferentie van processen in het interne geheugen te voorkomen zijn in de loop der tijd verschillende basisconcepten ontwikkeld, die in enigerlei vorm in commerciële produkten zijn toegepast.
Enkele concepten, namelijk "security kernels", "privileged mode" en "capabilities", worden hierna kort besproken.

De gedachte achter "security kernels" is om alle communicatie tussen processen te laten plaatsvinden via een centraal aanspreekpunt, de security kernel. In deze kernel zijn alle beveiligingsrelevante functies afgezonderd. Het gevolg is dat een willekeurig proces slechts in staat is tot communicatie met één centraal proces, dat de bevoegdheid van het aanvragende proces kan beoordelen.

Het risico dat processen ongewenste communicatie tot stand brengen met andere processen kan ook worden beperkt door "normale" processen slechts in staat te stellen in "user mode" (ook wel "problem state") uitgevoerd te worden, terwijl alle geprivilegieerde instructies (waaronder communicatie met andere processen) zijn voorbehouden aan processen in "privileged mode" (of "supervisor mode").

Het "capabilities"-concept kan worden beschouwd als een generalisatie van de "privileged mode"-gedachte. In plaats van twee mogelijke toestanden waarin een proces zich kan bevinden (met bijbehorende bevoegdheden), is er een in beginsel oneindig aantal mogelijkheden. De mogelijke instructies die een proces kan uitvoeren, worden bepaald door de "capabilities" die zijn gedefinieerd voor het domein waarin het proces draait.

4 Systeemprogrammering

Vanuit het oogpunt van betrouwbaarheid en beveiliging van de geautomatiseerde gegevensverwerking is het operating system in tweeërlei opzicht van belang:
-- de inherente eigenschappen van het operating system (in combinatie met de hardware en de overige systeemprogrammatuur);
-- de mogelijkheden om de functionaliteit van het operating system te wijzigen.

De inherente eigenschappen waren het onderwerp van de voorgaande paragraaf. In deze paragraaf wordt nader ingegaan op het beheer van het operating system, in het bijzonder het "change management".

Indien de functionaliteit van het operating system een statisch, niet beïnvloedbaar karakter zou hebben, dan zou het beheer ervan weinig risico's opleveren in termen van aantasting van de betrouwbaarheid en de beveiliging. Doordat echter de werking van het operating system door menselijk ingrijpen kan worden gewijzigd, dienen er maatregelen te zijn getroffen om deze beïnvloedingsmogelijkheden te beheersen, net zoals dat bij applicatieprogramma's het geval dient te zijn.
De functionaliteit van operating systems kan worden beïnvloed door het instellen van opties en parameters (waaronder bevoegdheidsregels) en door het toevoegen van programmacode.

De functie die verantwoordelijk is voor de installatie en het onderhoud van systeemprogrammatuur is systeemprogrammering. Afhankelijk van de omvang van de organisatie zal deze functie zijn ondergebracht bij één van de volgende afdelingen:
-- "systeembeheer": in een mini-omgeving, waarbij de automatiseringsafdeling veelal slechts uit twee of drie personeelsleden bestaat, zullen alle voorkomende werkzaamheden door elk van de personeelsleden kunnen worden uitgevoerd (werkvoorbereiding, operating, applicatieprogrammering, systeemprogrammering, etc.);
-- een aparte afdeling als onderdeel van het rekencentrum of als een geheel

aparte afdeling binnen de automatiseringsorganisatie;
-- een aparte afdeling voor ontwikkeling en een aparte afdeling voor acceptatie (de laatste als onderdeel van het rekencentrum).

Daarnaast komt het voor dat de functie is uitbesteed aan een derde (de leverancier).
De functie systeemprogrammering vormt potentieel een bedreiging voor de betrouwbaarheid en de beveiliging van de geautomatiseerde gegevensverwerking. Wie toegang heeft tot het operating system heeft mogelijk ook toegang tot de resources, programma's en gegevens binnen het systeem. Het gaat hierbij met name om de toegang tot de "security kernel", de "privileged mode", c.q. het domein met de meeste "capabilities". Een aantasting van het operating system heeft belangrijke gevolgen door de centrale plaats die dit inneemt in de geautomatiseerde gegevensverwerking.
Overigens vormt niet alleen de functie systeemprogrammering in dit opzicht een risico; functionarissen die belast zijn met het implementeren van bevoegdheidsregels of competentietabellen (veelal "security officers" genoemd) hebben uit hoofde van hun taken (ook) toegang tot beveiligingskritische functies van het operating system.
Het risico dat door systeemprogrammeurs schade aan de organisatie wordt berokkend - inefficiency, hobbyisme, onbevoegde toegang tot data - wordt versterkt door het esoterisch karakter van hun werk. Hierdoor zullen managers, interne controleafdelingen en accountants niet snel geneigd zijn met hen in discussie te treden over professionele standaards, technische bijzonderheden, etc.

Anderzijds moeten de risico's die uitgaan van de systeemprogrammeur ook weer niet worden overschat. Systeemprogrammeurs zijn technici, met weinig kennis van en belangstelling voor de primaire bedrijfsprocessen, de werking van de applicaties die deze processen vastleggen of sturen en de controles die in de gebruikersorganisatie plaatsvinden.
Daarnaast is er een ontwikkeling gaande dat systeemprogrammering niet meer zozeer het schrijven van Assembler-programma's inhoudt, als wel het instellen van opties en parameters. Door deze ontwikkeling worden systeemprogrammeurs beperkt in hun mogelijkheden en zijn hun verrichtingen beter controleerbaar.

In een organisatie van enige omvang (met een mainframe-omgeving) zijn voldoende maatregelen van functiescheiding te treffen binnen de automatiseringsafdeling om de geschetste risico's tot een aanvaardbaar niveau terug te dringen. In een kleinere organisatie met een mini-omgeving is een voldoende niveau van functiescheiding binnen de automatiseringsafdeling veelal niet realiseerbaar en moet beheersing van de systeemprogrammeringsfunctie vanuit de gebruikersorganisatie plaatsvinden. Speciale aandacht is nodig waar de geautomatiseerde gegevensverwerking een nauwe relatie heeft met de uitgaande geldbeweging, zoals bij het aanmaken van betaaltapes of bij betalingstransacties via datacommunicatie.

5 Audit van het operating system

Algemeen
Gelet op het belang van het operating system (in combinatie met de hardware en de overige systeemprogrammatuur) voor de betrouwbaarheid en de beveiliging van de geautomatiseerde gegevensverwerking, bestaat er bij veel organisaties behoefte om de installatie en het beheer ervan te laten beoordelen door een onafhankelijke of althans onpartijdige deskundige.
Ook de externe accountant die belast is met de controle van de jaarrekening kan er belang bij hebben dat het operating system op betrouwbaarheids- en beveiligingsaspecten wordt onderzocht.
In verband met de benodigde deskundigheid zal deze beoordeling door een technisch geschoolde EDP-auditor dienen plaats te vinden.
De doelstelling en de diepgang van het onderzoek worden bepaald door de opdracht. In de meeste gevallen zal het gaan om een onderzoek naar de betrouwbaarheid en de beveiliging, maar het is denkbaar dat een opdracht wordt verstrekt om de efficiency en/of de effectiviteit van het operating system te onderzoeken.

In geval van een effectiviteitsonderzoek moet de auditor zich een oordeel vormen over de mate waarin het operating system bijdraagt aan het behalen van de doelstellingen van de organisatie als geheel. Tussen een organisatiedoelstelling als winstmaximalisatie of penetratie van een bepaalde markt enerzijds en de mogelijkheden en beperkingen van een

bepaald operating system anderzijds, ligt een enorme afstand. De belangrijkste taak van de auditor zal dan zijn het vertalen van de organisatiedoelstellingen naar eisen voor de geautomatiseerde gegevensverwerking (betrouwbaarheidsniveau, waaronder mate van fouttolerantie, beveiligingseisen, mate van geschiktheid voor real time- dan wel batchprocessing, etc.).

Een efficiency-onderzoek vergt deskundigheid op het niveau van een systeemprogrammeur die ruime kennis heeft van en ervaring met het te onderzoeken operating system.
Aandachtspunten bij een dergelijk onderzoek zijn onder andere de jobscheduler (prioriteitstoekenning, evenwichtige mix van jobs die veel CPU-tijd gebruiken en die veel I/O vergen) en het "paging"-mechanisme (het verplaatsen van "pages" data en programmacode van en naar het externe geheugen).
Een belangrijk hulpmiddel voor efficiëntie-onderzoeken zijn performancegegevens in de systeem-logging.

Aanpak
Een onderzoek naar de betrouwbaarheid en de beveiliging kan het best in een aantal stappen worden uitgevoerd:

1. inventarisatie van de wensen en eisen die door of namens het management zijn gesteld ten aanzien van de functionaliteit (waaronder betrouwbaarheid en beveiliging) van het operating system;

2. inzicht verkrijgen in het operating system (in combinatie met de hardware en andere mogelijke systeemprogrammatuur, in het bijzonder beveiligingsprogrammatuur en utilities);

3. beoordeling van de maatregelen van beheersing van het operating system:
-- personeelsbeleid ten aanzien van systeemprogrammeurs (aanstelling, opleiding, etc.);
-- maatregelen van functiescheiding (verschillende programmeurs voor verschillende componenten van de systeemprogrammatuur, scheiding tussen test en produktie, peer review);
-- programmeer- en documentatiestandaards;
-- gebruikmaking van geëigende installatieprogrammatuur;
-- procedures voor het detecteren, analyseren en verhelpen van storingen;
-- etc.;

4. beoordeling van de wijze waarop het operating system feitelijk is geïnstalleerd. Dit houdt in een beoordeling van de ingestelde parameters en van de wijzigingen en uitbreidingen die op het operating system zijn aangebracht.
Deze stap kan zeer arbeidsintensief zijn en bovendien veel specifieke deskundigheid vereisen. Vaak is het al ondoenlijk om te inventariseren welke wijzigingen zijn aangebracht, laat staan dat op eenvoudige wijze de aard van de wijzigingen kan worden beoordeeld. Voor dit laatste is veelal source code review - van programma's geschreven in Assembler of tegenwoordig ook wel C - noodzakelijk (met alle beperkingen die daaraan kleven);

5. controle of de beheersing van het operating system volgens de voorgeschreven wijze heeft plaatsgevonden. Deze controle betreft onder andere het "problem and change management": worden de procedures zoals beschreven onder 3 inderdaad nageleefd.

Bij *stap 1 (inventarisatie managementeisen)* zal veelal blijken dat door het management geen expliciete, gedetailleerde eisen of normen zijn gesteld aan de betrouwbaarheid en de beveiliging. Het gaat hier bijvoorbeeld om een beveiligingseis als "protection by default" ("geen toegang, tenzij toegestaan"). (Het alternatief is "wel toegang, tenzij verboden".)
De EDP-auditor zal dan op basis van zijn deskundigheid en in overleg met het management moeten komen tot een stelsel van normen waaraan het operating system moet voldoen.
Bij de definitie van betrouwbaarheids- en beveiligingseisen is risicoanalyse onmisbaar: hoe afhankelijk is de organisatie van de functies van het operating system en welke bedreigingen (met welke kans van optreden) zijn er te onderkennen?

Het doel van *stap 2 (inventarisatie functies van het operating system)* is een indruk te verkrijgen van de mogelijkheden en beperkingen van het operating system en van de mogelijkheden om de functionaliteit te wijzigen.
Het is mogelijk dat bij deze stap wordt geconstateerd dat het operating system niet in staat is de gewenste functionaliteit te leveren, bijvoorbeeld een beveiligingsniveau dat voldoet aan de eisen van het NCSC voor klasse B. Het kan daarom gewenst zijn dat de EDP-

auditor reeds bij de besluitvorming rond de aanschaf van het operating system (in combinatie met de hardware) wordt ingeschakeld.
Bij de keus zal overigens niet alleen het beveiligingsniveau een rol spelen, maar ook het hard- en software-beleid van de organisatie in het algemeen en de beschikbaarheid van toepassingspakketten.

In *stap 3 (beoordeling beheersingsorganisatie)* wordt nagegaan of in opzet en bestaan voldoende maatregelen zijn getroffen om de betrouwbaarheid en de beveiliging van het operating system te waarborgen. Deze stap is gericht op de beheersingsorganisatie, maar aan de hand van de te voeren interviews kan tevens een indruk worden gevormd van de installatie (in opzet) van het operating system.
Bij deze stap kan de auditor in grote lijnen dezelfde normen hanteren als gelden voor de ontwikkeling en het beheer van applicaties. Het belangrijkste verschil is dat bij de acceptatie van systeemprogrammatuur de eindgebruiker vrijwel geen bemoeienis heeft en de verwerkingsorganisatie de beslissende stem heeft.
Vanuit het oogpunt van functiescheiding is het gewenst dat ontwikkelen (waaronder parametriseren) en testen van systeemprogrammatuur gescheiden plaatsvinden van de produktieomgeving, terwijl er binnen deze omgeving voldoende kennis aanwezig moet zijn om de te installeren programmatuur te toetsen op de (door het management) gewenste functionaliteit (en de afwezigheid van ongewenste functionaliteit).

In de praktijk wordt bij een majeure wijziging van het operating system vaak een werkgroep of overlegorgaan gevormd, of wordt het bestaande overleg geïntensiveerd. Een nieuwe release kan verschillende gevolgen hebben:
-- een verbetering van de functionaliteit zonder (directe) consequenties voor de betrouwbaarheid of de beveiliging;
-- invloed op de produktieomgeving, waardoor bestaande werkwijzen verandering ondergaan;
-- invloed op andere systeemprogrammatuur, waardoor deze moet worden aangepast;
-- invloed op applicaties, met hetzelfde gevolg.

De samenstelling van de werkgroep en de frequentie waarmee wordt bijeengekomen, dienen te zijn afgestemd op de aard van de wijziging.

Stap 4 (beoordeling feitelijke installatie) is erop gericht een oordeel te verkrijgen over het feitelijk bestaan van de betrouwbaarheids- en beveiligingsmaatregelen in de besturingsprogrammatuur op zeker moment. Afgeleide hiervan is een indruk van de kwaliteit van de beheersingsor-ganisatie.
Toetsingsnormen voor de auditor bij deze stap zijn de eisen die door het management zijn gesteld ten aanzien van de installatie (zie stap 1) en de beheersingsmaatregelen die bij de vorige stap zijn onderzocht.
Over het algemeen heeft het weinig of geen zin om stap 4 uit te voeren als in stap 3 is geconcludeerd dat het beheer te wensen over laat. Een vanuit het oogpunt van betrouwbaarheid en beveiliging adequate installatie van het operating system is dan hooguit een toevalstreffer en er zijn onvoldoende waarborgen dat de installatie bij voortduring betrouwbaar en beveiligd zal zijn.

Het doel van *stap 5 (beoordeling van het functioneren van de beheersingsorganisatie)* is een oordeel te verkrijgen omtrent de werking van de maatregelen gericht op de betrouwbaarheid en de beveiliging van het operating system: hebben deze maatregelen gedurende de onderzochte periode bij voortduring naar behoren gefunctioneerd. Deze stap heeft zowel betrekking op de maatregelen van betrouwbaarheid en beveiliging die zijn opgenomen in de programmatuur van het operating system, als op de beheersingsmaatregelen binnen systeemprogrammering en automatiseringsorganisatie.
De auditor toetst de werking van de getroffen maatregelen aan de opzet van de beheersorganisatie (stap 3) en aan de eerder beoordeelde installatie (stap 4).
Het heeft derhalve geen enkele zin om stap 5 uit te voeren als bij de stappen 3 en 4 tekortkomingen zijn geconstateerd. Er zijn dan onvoldoende maatregelen aangetroffen waarvan het de moeite waard is om de werking ervan gedurende een zekere periode te controleren.

Een bijzonder probleem bij stap 5, dat overigens inherent is aan alle onderzoeken naar de werking van maatregelen gedurende een zekere periode, is dat niet kan worden vastgesteld dat de maatregelen tijdens de onderzochte periode altijd hebben gewerkt. Deze vak-

technische beperking geldt speciaal voor het besturingssysteem; het is vrijwel onmogelijk vast te stellen dat het operating system gedurende een bepaalde periode ongewijzigd heeft gefunctioneerd.

6 Tot slot

In dit artikel is een globaal overzicht gegeven van de functies van een operating system, de taken die moeten worden uitgevoerd om een operating system op de juiste manier te laten functioneren en de wijze waarop een audit van deze functies en taakuitoefening kan worden uitgevoerd.

De functionaliteit van operating systems vertoont een grote mate van diversiteit en derhalve zal voor elke audit (met enige diepgang) een voor het desbetreffende operating system op maat gemaakt audit-programma moeten worden gehanteerd.

Drs. P. Veltman RA
Is na zijn studie bedrijfseconomie gedurende enige jaren werkzaam geweest in de algemene controlepraktijk van KPMG Klynveld, tijdens welke periode hij het post-doctoraal diploma accountancy behaalde, waarna hij de overstap maakte naar de EDP Audit-groep binnen Klynveld. Zijn audit-ervaring ligt op het gebied van besturingssystemen en beveiligingspakketten, informatiesystemen en automatiseringsorganisaties. Hij is docent van cursussen op het gebied van automatisering en controle.

Door de opkomst van het Virtual Machine concept (VM) van IBM krijgt de EDP-auditor in toenemende mate te maken met de beveiligingsaspecten van VM en de invloed die VM kan uitoefenen op de onderliggende besturings- en applicatiesystemen.

A.A.J. Breed

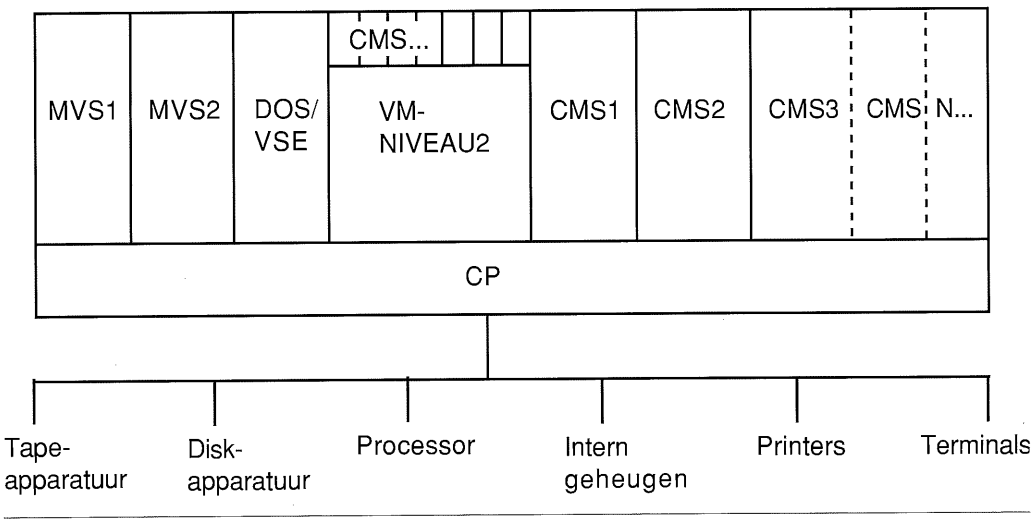
Het Virtual Machine concept van IBM

1 Inleiding

Behalve de reeds alom bekende mainframe-besturingssystemen MVS en DOS/VSE kent IBM een derde besturingssysteem: Virtual Machine (VM). VM is een zogenaamd "host"-besturingssysteem, dat meerdere andere "guest"-besturingssystemen, zoals MVS, (DOS/) VSE of zelfs een tweede niveau VM, naast elkaar op één machine laat functioneren. Dit vindt plaats door simulatie van hardware-componenten in zogenaamde virtuele machines. Elke virtuele machine heeft zodoende een eigen virtuele hardware-configuratie inclusief processor. In werkelijkheid wordt alle hardware gezamenlijk gebruikt. Door deze bijzondere eigenschap kan VM worden gebruikt voor conversie van DOS/VSE naar MVS of scheiding tussen test en produktie op één machine. De laatste tien jaar wordt VM meer en meer gebruikt als een vast besturingssysteem in de range van kleinere systemen (IBM 9370, 43xx) tot zeer grote systemen (IBM 3090-xxx). Om al deze verschillende soorten hardware en systeem-software te kunnen ondersteunen, zijn verschillende versies van VM leverbaar. Zo is naast VM/SP een VM/ XA beschikbaar voor ondersteuning van de Extended Architecture-faciliteit en een VM/HPO voor extra mogelijkheden om de performance van het systeem te kunnen verbeteren.

Dat VM zich heeft ontwikkeld tot een volwassen en levensvatbaar besturingssysteem blijkt, naast het brede scala van zowel IBM- als non-IBM-produkten dat leverbaar is voor VM-omgevingen, uit het groeiende aantal rekencentra dat gebruik maakt van VM. Hierdoor krijgt ook de EDP-auditor in toenemende mate te maken met de beveiligingsaspecten van VM en de invloed die VM kan uitoefenen op de onderliggende besturings- en applicatiesystemen. In dit artikel is getracht duidelijk te maken hoe VM is opgebouwd, welke risico's VM met zich

Figuur 1. Vereenvoudigd voorbeeld VM-architectuur.



meebrengt en welke maatregelen kunnen worden getroffen om deze risico's te beperken.

2 Opbouw VM

De verschillende versies van VM (VM/SP, VM/HPO, VM/XA) bestaan alle uit de volgende onderdelen:

-- Het *Control Program (CP),* dat de werkelijke hardware-componenten bestuurt en zoveel virtuele machines simuleert als gewenst. Op deze virtuele machines kunnen meerdere besturingssystemen worden gebruikt, zoals MVS (XA), VSE, CMS of wederom VM. Met CP kan de apparatuur en aanwezige diskruimte in afzonderlijke delen worden verdeeld. De diskruimte wordt ingedeeld in zogenaamde minidisks. De componenten kunnen door één of meer virtuele machines worden gebruikt. De virtuele machines kunnen met elkaar communiceren via het CP. De systeemoperators van de machines beschikken over CP-commando's waarmee bijvoorbeeld de eigen (virtuele) configuratie kan worden gewijzigd of verbinding met andere virtuele machines tot stand kan worden gebracht (binnen de begrenzingen die in het CP zijn gedefinieerd).

-- Het *Conversational Monitor System (CMS),* dat als "single user-besturingssysteem" door zowel systeemprogrammeurs als gebruikers kan worden gebruikt voor onderhoud van programma's, gegevensbestanden en tekstverwerking. Het beschikt standaard over een tekstverwerker en een tweetal interpretertalen, waarmee geautomatiseerde procedures kunnen worden opgesteld. Een groot aantal van de applicatiesystemen die IBM aanbiedt, zoals het Professional Office System (PROFS) en het Application System (AS), maakt gebruik van de faciliteiten die het CMS-(gast)operatingsysteem ter beschikking stelt.

Daarnaast kunnen de volgende onderdelen worden geïnstalleerd, die kunnen worden gebruikt om extra faciliteiten te verkrijgen:

-- *Group Control System (GCS);* deze multitasking supervisor kan worden gebruikt om subsystemen op te starten die System Network Architecture (SNA) ondersteunen, zoals VTAM en RSCS;

-- *Interactive Problem Control System (IPCS);* dit stelt de gebruiker in staat online eventuele software-fouten of -problemen te analyseren en te rapporteren;

-- *Transparant Services Access Facility (TSAF);* hiermee kunnen VM/APPC-applicaties communiceren met VM/APPC-applicaties in een ander VM-systeem;

-- *Advanced Program-to-Program Communication/VM VTAM Support (AVS);* hiermee worden (APPC)-applicaties in staat gesteld te communiceren met andere APPC-applicaties in een SNA-netwerk. Dit hoeven geen VM/APPC-applicaties te zijn;

-- Diverse overige produkten die kunnen worden toegevoegd aan het VM-systeem, zoals:
. *Virtual Telecommunications Access Method (VTAM/VM);* met dit onder GCS draaiend subsysteem kan onder meer de communicatie van alle terminals met het VM-systeem worden geregeld;
. *Remote Spooling Communications Subsystem Networking (RSCS);* een onder GCS draaiend subsysteem waarmee data kunnen worden verstuurd naar en ontvangen van bestemmingen buiten het eigen VM-systeem;
. *VM/Directory Maintenance (DIRMAINT);* een pakket waarmee gebruikers onder meer in staat worden gesteld zelf passwords te wijzigen. Het beschikt bovendien over bepaalde audit-faciliteiten;
. *Resource Access Control Facility (RACF);* een VM-versie van het beveiligingspakket RACF. Hiermee zijn extra beveiligingsmogelijkheden in een VM-omgeving mogelijk, zoals integrale autorisatiecontrole van beveiligde objecten en uitgebreide logging- en report-faciliteiten;
. *Customer Information Control System/VM (CICS/VM);* een CMS-versie van de teleprocessing monitor CICS/VS, waarmee binnen CMS CICS-applicaties kunnen worden ontwikkeld. Daarnaast zijn interfaces gedefinieerd met de overige CICS-produkten, waardoor vanuit CICS/VM applicaties kunnen worden opgestart in bijvoorbeeld CICS/VS;
. *Structured Query Language/ Data System (SQL/DS);* een onder VM implementeerbaar relationeel Data Base Management Systeem, dat zowel door CMS- als VSE-applicaties (onder andere CICS) te benaderen is.

Control Program (CP)
Het hart van VM wordt gevormd door het

Control Program. Het beheert alle hardware-componenten, het simuleert virtuele machines en het regelt alle verkeer tussen virtuele en reële hardware-componenten, inclusief diskeenheden. Om deze taken te kunnen verrichten dient het te beschikken over informatie die vóór systeemgeneratie moet worden vastgelegd in systeemgeneratiebestanden. Bij systeemgeneratie worden deze bestanden gecompileerd en in object-code opgeslagen en geactiveerd bij het opstarten van het systeem. Deze bestanden zijn te herkennen aan het prefix DMKxxx (bij VM/XA is dit echter HCP xxx). De directory heeft een afwijkende naam, die per systeem kan verschillen.

Beheer van de reële hardware
Het CP beheert alle hardware-componenten van het systeem. Het dient hiertoe te beschikken over informatie over de fysieke of reële hardware.
Het beschrijven van de reële hardware en systeeminitialisatiegegevens vindt plaats in de systeemgeneratiebestanden DMKRIO (RIO staat voor Real Input / Output) en DMKSYS (SYS staat voor SYStem).

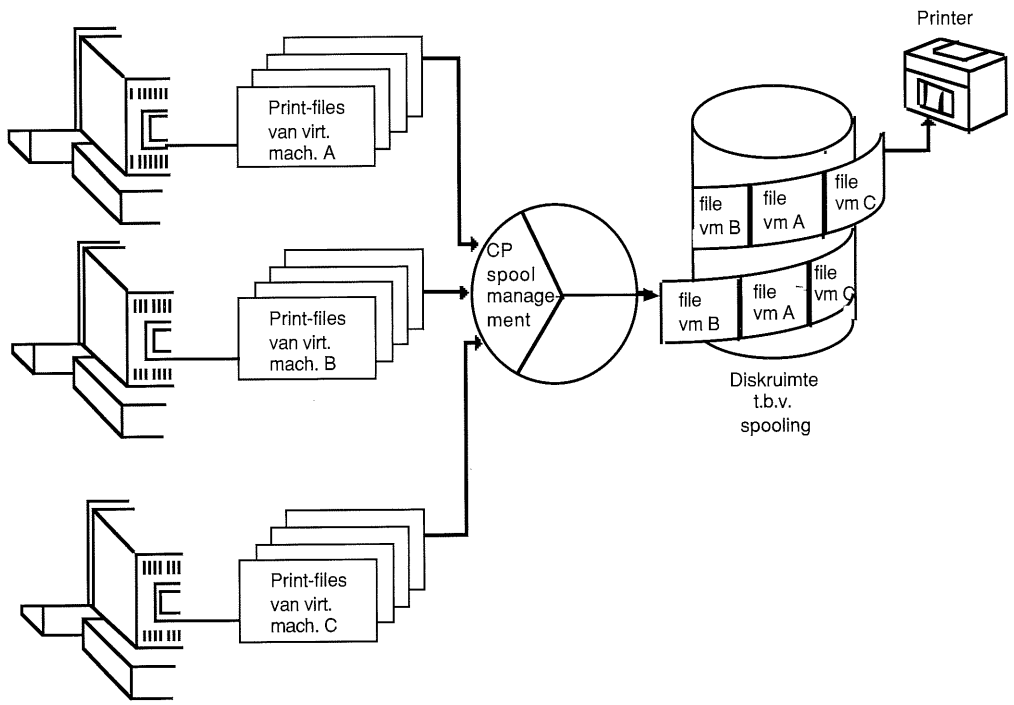
De volgende informatie wordt vastgelegd:
-- Processor type;
-- Omvang intern geheugen;
-- Input/output-configuratie:
. device-adressen;
. device-typen;
. adressen ten behoeve van systeemstart;
. channel-nummers;
. channel-typen;
-- Indeling diskruimte voor:
. CP-programmatuur;
. paging/spooling;
. tijdelijke diskruimte;
. saved systems;
. dump;
. object directory;
. override file.

Simulatie van hardware
Bij de simulatie wordt onderscheid gemaakt tussen zes verschillende componenten:

-- *Processor :* De simulatie van de processor vindt plaats via de verdeling van de totale processortijd in time slices. Iedere virtuele machine krijgt van het CP, afhankelijk van de vastgestelde prioriteit, een aantal time slices toegewezen.

-- *Intern geheugen:* Het CP maakt gebruik van virtueel geheugen, waardoor altijd een deel van het virtuele geheugen op schijf zal staan. Iedere virtuele machine krijgt een stuk virtueel geheugen tot haar beschikking ter grootte van een veelvoud van 4K. Indien een virtuele machine een lees- of schrijfactie onderneemt naar haar "reële" geheugen, zal het CP de vertaling verzorgen naar het

Figuur 2. CP-spool management.



virtuele geheugen en daarna naar het echte reële geheugen.

-- *Readers, punchers, printers:* Het CP maakt gebruik van spooling naar disk om deze hardware te simuleren. De readers en punchers worden gebruikt voor communicatie tussen de aanwezige virtuele machines. Het daadwerkelijke uitprinten vindt pas plaats als de werkelijke systeemprinter vrij is. Tot die tijd staat de print job in de VM-spool area, die wordt beheerd door het CP-spool management (zie figuur 2).

-- *Tape-apparatuur:* Deze apparatuur kan door haar structuur en enorme opslagcapaciteit niet werkelijk worden gesimuleerd en derhalve niet worden geshared door verschillende machines. Op verzoek kan de operator een tape-eenheid aan een virtuele machine toewijzen.

-- *Diskapparatuur:* Het CP verdeelt de reële disks in minidisks. Deze minidisks kunnen verschillen in grootte en functioneren als virtuele diskeenheden. Doordat diskapparatuur ge-shared kan worden, is het ook mogelijk een minidisk door meerdere virtuele machines te laten gebruiken. Hierover straks meer.

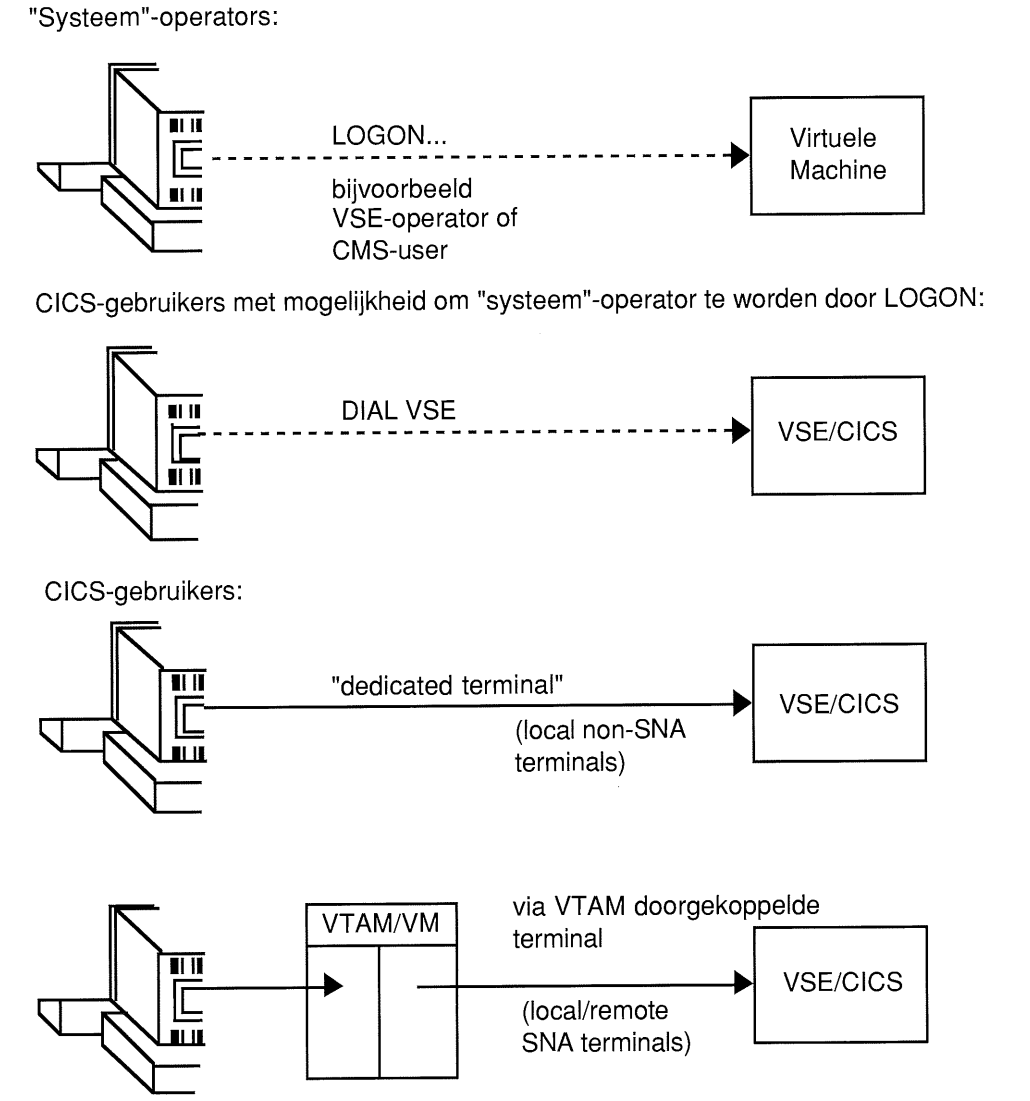
-- *Terminals:* Alle uitgaande lijnen worden beheerd door het CP. Hierbij moet echter een onderscheid worden gemaakt tussen lijnen naar local en naar remote terminals. Local terminals zijn direct met de hardware verbonden (eventueel door middel van terminal controllers) en remote terminals zijn verbonden via communication controllers. Deze communication controllers hebben de mogelijkheid één uitgaande lijn van het CP op te splitsen voor gebruik van meerdere terminals. Als een gebruiker een willekeurige aan het VM-systeem gekoppelde terminal aanzet, krijgt hij in principe het aanlogscherm van VM te zien. Hierna kan de gebruiker contact zoeken met het systeem, waardoor de terminal wordt gekoppeld aan een virtuele machine.

Definitie van virtuele machines
Het definiëren van virtuele machines vindt plaats in de VM-directory. Iedere virtuele machine krijgt hierbij een naam en password. De naam van de virtuele machine zal als user id van de "systeem"-operator gaan dienen. Deze "systeem"-operator kan zowel de operator van bijvoorbeeld een virtuele VSE-machine zijn als een eindgebruiker van een virtuele CMS-machine. De overige gebruikers, bijvoorbeeld VSE/CICS-gebruikers, zijn niet in de VM-directory opgenomen, maar in VSE/CICS. Deze gebruikers kunnen op drie manieren door VM worden geleid:
-- De fysieke terminal van de gebruiker is direct gekoppeld aan de virtuele machine (dedicated terminal), waardoor de gebruiker wordt geconfronteerd met het VSE/CICS-aanlogscherm. Dit is alleen mogelijk voor "local non-SNA"-terminals.
-- Via VTAM-definities is de (SNA-) terminal doorgekoppeld met een virtuele machine (dit kan zowel voor local als voor remote terminals).
-- Via het VM-aanlogscherm dat op de terminal verschijnt, wordt "gekozen" voor de virtuele VSE-machine via het DIAL-commando. De gebruiker komt terecht bij VSE/CICS. Het verschil met de voorgaande wijze is dat de terminal de mogelijkheid had om ook via het VM-scherm aan te loggen als "systeem"-operator van een virtuele machine.

De VM-directory wordt vóór systeemgeneratie vastgelegd, maar kan on-line worden gewijzigd en geactiveerd. Per virtuele machine worden onder meer de volgende gegevens in de directory opgenomen:
-- user id en password;
-- grootte van het virtueel geheugen;
-- toegewezen command classes;
-- minidisk-configuratie;
-- prioriteit;
-- type of virtuele adres van het besturingssysteem;
-- adressen van (dedicated) terminals;
-- V(irtual)=R(eal).

De prioriteit bepaalt de relatieve hoeveelheid tijdseenheden die een virtuele machine toegewezen krijgt. Met de V=R-parameter kan paging overhead van het besturingssysteem van één bepaalde virtuele machine worden voorkomen. De command class(es) bepaalt/bepalen de bevoegdheden die de "systeem"-operator heeft voor het uitvoeren van CP-commando's. Elke virtuele machine kan één of meer command classes bezitten. Ieder commando kan worden uitgevoerd door één of meer classes, waarbij de betekenis van het commando in sommige gevallen afhangt van de command class. De standaardindeling van commando's per class is als volgt:

Class A Primary system operator.
Deze class is onder meer in staat het VM-systeem te activeren.



Figuur 3. Terminal-mogelijkheden.

Class B System resource operator.
Deze class kan de reële en de virtuele hardware-configuratie wijzigen.

Class C Systems programmer.
In deze class bevinden zich commando's waarmee de systeemprogrammeur het VM-systeem kan onderhouden en onderzoeken. Hieronder bevinden zich commando's waarmee het reële geheugen kan worden gemuteerd.

Class D Spooling operator.
Deze class heeft toegang tot het CP-spool management en kan onder meer de volgorde van verwerking van de spoolfiles beïnvloeden.

Class E System analist.
Deze class is in staat besturingssystemen voor virtuele machines te definiëren en te muteren in de System Name Table en heeft leesbevoegdheid in het reëel geheugen.

Class F Service representative.
Deze class is bedoeld voor gebruik door onderhoudstechnici van IBM.

Class G General user.
Hoewel de commando's van deze class slechts betrekking hebben op de eigen virtuele machine, beschikt deze class over een commando waarmee verbinding met minidisks van andere virtuele machines kan worden gelegd.

De standaardindeling van deze command classes is echter te veranderen via een override file. Dit is een optioneel systeemgeneratiebestand (DMKOVR). Een aantal commando's is niet ondergebracht in command classes; deze zijn door alle classes te gebruiken. Hier-

onder vallen commando's zoals LOGON en LOGOFF, waarmee respectievelijk virtuele machines kunnen worden opgestart en afgesloten door de "systeem"-operator.

Het besturingssysteem dat bij het opstarten van de virtuele machine wordt geladen, kan op twee manieren in de VM-directory worden aangegeven:
-- Het type besturingssysteem kan zijn opgenomen bij de virtuele machine. De verschillende mogelijke besturingssystemen staan vermeld in het systeemgeneratiebestand DMKSNT (SNT staat voor system name tabel). De systeemprogrammacode kan hierbij in het geheugen worden ge-shared door verschillende virtuele machines. Dynamische wijzigingen in de systeemprogrammatuur buiten VM om zijn dan niet mogelijk aangezien er sprake is van "bevroren" systeemprogrammatuur.
-- Er kan een (virtueel) adres zijn opgenomen waar de besturingsprogrammatuur op minidisk is opgeslagen. Hierbij wordt het besturingssysteem van disk opgehaald en geladen in de virtuele machine. Hierbij is geen sharing van geheugenruimte, maar wel sharing van systeemprogrammatuur mogelijk. Deze systeemprogrammatuur kan wel door virtuele machines worden gewijzigd zonder tussenkomst van VM.

Voor sommige "guest"-besturingssystemen (bijvoorbeeld VSE en MVS) zijn zogenaamde "handshaking"-faciliteiten met VM beschikbaar. Hierdoor neemt de verwerkingssnelheid van deze "guest"-besturingssystemen sterk toe en wordt het mogelijk CP-commando's uit te voeren. Hiermee wordt het bijvoorbeeld mogelijk te communiceren met andere virtuele machines. Deze handshaking-faciliteiten bestaan voornamelijk voor VSE en in mindere mate voor MVS.

Gebruik diskruimte
Alle in het systeem aanwezige diskruimte wordt gedefinieerd in de VM-directory. VM maakt onderscheid tussen vier verschillende soorten disks:

-- Disks voor gebruik van het CP; dit zijn system owned disks. Deze disks worden gebruikt voor opslag van de CP-systeemprogrammatuur, paging, etc. Ze moeten voor systeemgeneratie worden geïnitialiseerd. Omdat op deze disks ook nog minidisks kunnen worden geplaatst, bestaat de kans dat de ruimte wordt overschreven door deze minidisks. Om dit te voorkomen, worden de system owned diskruimten als "dummy" in de directory geplaatst bij speciale virtuele machines. Deze machines zijn te herkennen door een "$" voor en achter de naam (bijvoorbeeld $DIRECT$ voor de plaats van de objectversie van de system directory). Dit is echter niet verplicht en vindt plaats uit documentaire overwegingen ten behoeve van systeemprogrammering.

-- De disks die ge-shared kunnen worden door verschillende virtuele machines, de echte minidisks. Eén virtuele machine is aangewezen als de eigenaar van de minidisk door het MDISK-statement in de directory. De overige machines kunnen door een LINK verbonden zijn aan deze minidisks. Deze LINK kan statisch zijn (in de directory gedefinieerd) of dynamisch (CP commando class G). De toegang tot de minidisk kan zowel voor de eigenaar (MDISK) als voor andere virtuele machines (LINK) op drie verschillende toegangsniveaus plaatsvinden:
. leesbevoegdheid (R);
. exclusieve lees/schrijfbevoegdheid (W);
. gedeelde lees/schrijfbevoegdheid (M).

Deze letters (R, W, M) geven het primaire toegangsniveau aan waarop de virtuele machine in principe met de minidisk in verbinding wil komen (bijvoorbeeld bij LOGON of LINK). De statements kunnen worden voorzien van een tweede letter (R, W), die het alternatieve toegangsniveau weergeeft waarmee de link wordt uitgevoerd indien een andere virtuele machine de minidisk reeds in gebruik heeft. De uiteindelijke effectuering van de verbinding is afhankelijk van de reeds bestaande verbindingen.

In de directory kan de minidisk bij het MDISK-statement voorzien zijn van een password per toegangsniveau (R, W, M).

Figuur 4. Toegangsmogelijkheden minidisks.

Primair verzoek	R						W						M								
Alternatief verzoek	-			R			-			R			-			R			W		
Bestaande link(s)	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W
Toegewezen link	R	R	-	R	R	R	W	-	-	W	R	R	W	W	-	W	W	R	W	W	W

Wanneer voor een toegangsniveau geen password is gespecificeerd, is geen dynamische link mogelijk. In het andere geval moet de dynamische link worden voorzien van het password. Indien als password ALL is gespecificeerd, kunnen alle class G virtuele machines deze disk linken, zonder een password op te hoeven geven.

-- Minidisks voor tijdelijk gebruik. Deze minidisks worden temporary minidisks genoemd en kunnen door CMS-gebruikers dynamisch worden gecreëerd op system owned disks. Na het verlaten van het systeem worden deze disks voor hergebruik vrijgegeven.

-- Disks voor exclusief gebruik van één virtuele machine; de zogenaamde dedicated "mini"disks. Deze disks zijn niet via statische of dynamische links benaderbaar door andere virtuele machines. De dedicated "mini"disk kan wel in zijn geheel van een virtuele machine worden losgekoppeld en aan een andere machine worden gekoppeld via respectievelijk het DETACH- (class B,G) en het ATTACH-commando (class B).

Conversational Monitor System (CMS)

Het CMS is een single user-besturingssysteem dat alleen als "guest"-besturingssysteem in een VM-omgeving werkt. Na het aanloggen van een CMS-machine zal door het CP de virtuele configuratie worden gegenereerd en het CMS-besturingssysteem worden geladen. CMS is voorzien van een eigen bestandsstructuur en eigen CMS-commando's. Het vormt hiermee de basis voor IBM- en non-IBM-produkten, waarmee de virtuele machine een groot aantal verschillende functies kan uitvoeren. Voor de eindgebruiker achter de terminal is CMS vergelijkbaar met een personal computer met tekstverwerkingscapaciteiten en met besturingscommando's, zoals het printcommando en het directory opvraagcommando.

Ook andere gebruikers, zoals de systeemprogrammeurs, maken gebruik van CMS voor het onderhoud van bijvoorbeeld de VM-systeemgeneratiebestanden en de VM-systeemprogrammatuur. Voor programmeurs zijn er compilers en test- en debug-faciliteiten aanwezig. CMS maakt hierbij gebruik van verschillende subomgevingen, waarbij zowel OS/VS- als DOS/VS-simulatie mogelijk is.

CMS-commando's

De commando's die een CMS-gebruiker ter beschikking staan, zijn in vier groepen te verdelen:

-- CP-commando's; uiteraard gelimiteerd door de command classes van de virtuele machine;

-- CMS-commando's; hieronder vallen in de eerste plaats commando's voor het "normale" omgangsverkeer met de eigen virtuele machine, zoals formatteren, directory opvragen, het versturen van bestanden via de puncher en het besturen van de eventuele tape-eenheid. In de tweede plaats zijn er commando's die men niet direct zou verwachten bij een gemiddelde CMS-gebruiker; hieronder valt bijvoorbeeld het DIRECT-commando, waarmee een nieuwe VM-directory kan worden gegenereerd;

-- Commando's die een procedure opstarten, die bijvoorbeeld is geschreven in REXX of EXEC2, de standaard meegeleverde interpreter-talen. Deze procedures kunnen zelf ook weer CP- en CMS-commando's bevatten;

-- Commando's die CMS in een andere subomgeving brengen. CMS vormt hierbij een platform waarmee verschillende soorten applicaties kunnen worden opgestart. Zo kan bijvoorbeeld de standaard editor (XEDIT) worden opgestart of kan een DOS/VS-omgeving worden gesimuleerd.

Gebruik bestanden en minidisks

CMS heeft net als andere besturingssystemen de mogelijkheid minidisks te formatteren en te voorzien van een label. Na dit proces zijn de disks klaar voor gebruik en kunnen voor bestandsopslag worden gebruikt.

De disks zijn benaderbaar onder de namen A tot en met Z, al naargelang het aantal minidisks in de virtuele configuratie. De namen van bestanden en programmatuur worden geïdentificeerd door een file name, file type en file mode.

In een aantal gevallen verwacht CMS dat een programma of bestand voorzien is van een bepaald file type. Een voorbeeld hiervan is het file type EXEC voor REXX- en EXEC2-programma's.

De file mode bestaat uit twee delen:

-- de letter van de minidisk (A tot en met Z);

-- een cijfer waarmee een onderscheid in soorten bestanden kan worden gemaakt (0 tot en met 6), waarbij sommige cijfers een bijzondere betekenis hebben. Bijvoorbeeld een bestand dat is voorzien

van een 0, kan door een andere gebruiker die de minidisk met leesbevoegdheid heeft ge-linked, niet worden gelezen.

Hoewel het voor CMS-gebruikers mogelijk is minidisks met andere CMS-gebruikers te delen, kan dit in sommige gevallen risico's inhouden. Indien een minidisk tegelijkertijd door twee CMS-machines wordt benaderd met als toegangsniveau gedeeld lezen/schrijven (MW), voorziet CMS niet in een locking-mechanisme met alle risico's daaraan verbonden. Vanaf release 6 kent VM daarom een zogenaamd Shared File System (SFS), waarmee verschillende CMS-gebruikers gemeenschappelijk bestanden kunnen gebruiken, zonder dat ge-linked hoeft te worden. SFS slaat de bestanden op in een groep van minidisks (file pools). Zo'n file pool wordt beheerd door een speciale virtuele machine.
SFS voorziet onder meer in een eigen autorisatie- en locking-mechanisme.

3 Beveiligingsaspecten

Een betrouwbare gegevensverwerking dient te voldoen aan drie eisen:
-- (Produktie)programmatuur mag alleen door geautoriseerde functionarissen worden overgebracht naar de produktieomgeving.
-- Gegevens dienen slechts door geautoriseerde programmatuur te kunnen worden benaderd.
-- Programmatuur mag alleen door of in opdracht van geautoriseerde functionarissen worden uitgevoerd en gewijzigd.

De programmatuur kan hierbij worden verdeeld in applicatie- en systeemprogrammatuur. In een VM-omgeving hebben we echter te maken met minimaal drie niveaus van programmatuur:
-- de VM-systeemprogrammatuur;
-- de systeemprogrammatuur van de virtuele machine;
-- de applicatieprogrammatuur.

Een compleet onderzoek naar de betrouwbaarheid van de gegevensverwerking in een VM-omgeving zou hierdoor inhouden dat deze drie niveaus in beschouwing dienen te worden genomen. Omgekeerd betekent dit dat bij een betrouwbaarheidsonderzoek van een "guest"-besturingssysteem zoals MVS, niet aan de te nemen maatregelen binnen VM voorbij kan worden gegaan. In dit artikel zal echter uitsluitend worden ingegaan op de beveiligingsaspecten van de VM-systeemprogrammatuur en zullen de virtuele machines als applicaties en de verschillende diskbestanden als gegevens worden beschouwd. Dit is mogelijk doordat de virtuele machines in een VM-omgeving als een soort applicatieprogramma's functioneren die kunnen beschikken over bepaalde systeemcommando's (de CP-commando's en de communicatiefaciliteiten via readers/punchers) en directe toegangsrechten tot gegevens (de minidisks).
Doordat het CP slechts een autorisatiemechanisme kent op minidisk-niveau, betekent dit dat we het zicht op de gegevens op bestandsniveau kwijtraken, tenzij voor ieder bestand een aparte mini-disk wordt gedefinieerd. In de praktijk zal een globale bestandsautorisatie plaatsvinden binnen CP en zal een gedetailleerde bestandsautorisatie in het besturingssysteem van de individuele virtuele machines plaats dienen te vinden. CMS kent met het SFS wel de mogelijkheid om op bestandsniveau de autorisaties van verschillende virtuele machines vast te leggen. Het voert echter voor dit artikel te ver om hierop uitgebreid in te gaan.

Aspecten VM-systeemprogrammatuur
De initiële systeemgeneratie vindt plaats via een speciaal "starter"-systeem. Dit is een door IBM geleverd, op CMS gelijkend besturingssysteem, dat wordt geladen vanaf tape en uitsluitend wordt gebruikt voor het initieel genereren van VM. Hiermee kan de systeemprogrammeur aanloggen aan het systeem als virtuele machine MAINT. Vervolgens kan hij de systeemgeneratiebestanden globaal aan de eigen situatie aanpassen, de "system owned" disks initialiseren en VM genereren. Door IBM worden standaard systeemgeneratiebestanden met standaard parameters, user ids en passwords meegeleverd. Nadat VM initieel is gegenereerd, zal de systeemprogrammeur het systeem gaan inrichten en de systeemgeneratiebestanden in detail gaan aanpassen en opnieuw genereren. Hierbij gaat het in hoofdzaak om de volgende vier systeemgeneratiebestanden:
-- DISKMAP; de VM-systeem-directory met minidisks en command classes;
-- DMKRIO; de beschrijving van input/output-configuratie;
-- DMKSYS; de beschrijving van systeemparameters waaronder beveiligingsopties;
-- DMKSNT; de beschrijving van de in het systeem op te nemen "bevroren" besturingssystemen.

Aan deze systeemgeneratiebestanden, die in zowel source- als objectvorm voorkomen, kunnen twee globale eisen worden gesteld:

-- Het is van belang dat de systeemgeneratiebestanden de actuele situatie beschrijven, waarbij overbodige parameters verwijderd en situatieafhankelijke parameters toegevoegd dienen te worden. Als voorbeeld kan worden genoemd het verwijderen van niet gebruikte standaard IBM user ids van virtuele machines. Uiteraard dienen voor de verschillende gebruikers of "guest"-besturingssystemen wel nieuwe virtuele machines te worden toegevoegd.

-- De source-vorm moet gelijk zijn aan de gecompileerde en hierdoor geactiveerde objectvorm. Dit compileren vindt plaats door het uitvoeren van het CMS-commando DIRECT voor de VM-directory en een speciaal CMS-commando (per VM-release verschillend) voor de overige systeemgeneratiebestanden. Dit kan steeds opnieuw plaatsvinden. Hierbij kan worden gespecificeerd welke systeemgeneratiebestanden opnieuw dienen te worden gecompileerd. De virtuele machine die deze commando's laat uitvoeren, dient echter te beschikken over twee "autorisaties".
-- Zij dient te beschikken over het nieuw te creëren systeemgeneratiebestand (de source).
-- Zij dient schrijfrecht te hebben op de system owned diskruimte, waar de systeemgeneratiebestanden in gecompileerde vorm worden opgeslagen.

De systeemgeneratiebestanden worden door IBM gedefinieerd op een minidisk van de virtuele machine MAINT (via het starter-systeem). Toegang tot deze minidisk kan worden ontnomen door bijvoorbeeld géén passwords te specificeren voor de minidisk en géén statische links in de directory op te nemen. Desondanks kan met behulp van de standaard tekstverwerker een eigen systeemgeneratiebestand worden aangemaakt, zodat de tweede "autorisatie" actueel wordt. Het schrijfrecht op de system owned disks wordt tevens in de directory geregeld en dient hier te worden voorbehouden aan geautoriseerde virtuele machines (bijvoorbeeld MAINT).

Aspecten virtuele machines
Indien de virtuele machines als applicaties worden beschouwd, kunnen de volgende eisen worden gesteld:

-- Virtuele machines mogen alleen door geautoriseerde functionarissen worden overgebracht naar de produktieomgeving.
-- Gegevens dienen slechts door geautoriseerde virtuele machines te kunnen worden benaderd.
-- Virtuele machines dienen slechts door geautoriseerde functionarissen te kunnen worden gebruikt en gewijzigd.

De eerste eis betekent in een VM-omgeving dat alleen geautoriseerde functionarissen toegang mogen hebben tot de VM-directory. De toegang tot deze VM-directory wordt in de VM-directory zelf geregeld.

De tweede eis betekent dat virtuele machines alleen die mini- en system owned disks en overige systeemcomponenten mogen gebruiken waartoe zij geautoriseerd zijn. In de eerste plaats zijn communicatiemogelijkheden tussen verschillende virtuele machines via readers en punchers aanwezig. Binnen VM is geen standaardmechanisme aanwezig om deze communicatie te autoriseren. Dit dient in de virtuele machines zelf plaats te vinden. De overige toegang tot gegevens is in de VM-directory vastgelegd door drie maatregelen:
-- de configuratie van de virtuele machine;
-- de specificatie van minidisks en statische links;
-- de aan de virtuele machines toegewezen command classes.

De configuratie bepaalt in hoeverre de virtuele machine gebruik mag maken van de fysieke hardware (bijvoorbeeld de diskruimte of de systeemprinter). De specificaties van mini- en dedicated disks en statische links bepalen onder meer de toegangsrechten van de virtuele machines. De aan de virtuele machine toegewezen command classes bepalen de soorten CP-commando's die door de virtuele machine kunnen worden uitgevoerd. Deze CP-commando's kunnen de toegang van virtuele machines tot systeemcomponenten beïnvloeden. Indien de standaardstructuur van CP-command classes is gewijzigd door middel van de override file (DMKOVR), dient het verkregen resultaat de tweede eis te ondersteunen.

De derde eis betekent dat de virtuele machines en de bijbehorende configuratie inclusief minidisks, beschermd dienen te worden tegen ongeautoriseerde

wijzigingen en gebruik. Hiertoe dienen de virtuele machines en minidisks te zijn voorzien van adequate passwords. Dit vindt centraal plaats in de VM-directory. Het uitvoeren van dynamische links kan hiermee worden geautoriseerd. VM beschikt niet standaard over een faciliteit om gebruikers zelf passwords te laten wijzigen. Dit kan echter plaatsvinden door installatie van bijvoorbeeld het eerder genoemde pakket DIRMAINT. Met dit pakket kunnen extra password-controles voor zowel virtuele machines als minidisks worden ingebouwd, zoals minimale lengte van het password en ongelijkheid aan een aantal generaties voorgaande passwords. VM beschikt wel standaard over een faciliteit waarmee ongeautoriseerde aanlog- en link-pogingen kunnen worden gedetecteerd en gelogged. De parametrisering van deze faciliteiten vindt plaats in het systeemgeneratiebestand DMKSYS.

4 Tot slot

Dat VM meer is dan slechts een conversietool moge nu duidelijk zijn. Het is een besturingssysteem dat enerzijds een bijdrage levert aan de integratie van verschillende verwerkingsomgevingen en anderzijds uitstekende mogelijkheden biedt voor scheiding van bepaalde verwerkingsomgevingen. De integratie kan gebruikt worden voor communicatie tussen systemen en centraal gebruik van gegevens, de scheiding voor bijvoorbeeld het scheiden van een test- en een produktieomgeving. In dit laatste geval kunnen bijvoorbeeld meerdere test- en produktiesystemen op dezelfde hardware worden geïmplementeerd. Dit kunnen besturingssystemen zijn als MVS of VSE, maar ook bijvoorbeeld VM. Deze virtuele VM-machine (second level VM) kan bijvoorbeeld worden ingericht als testsysteem voor nieuwe VM-besturingsprogrammatuur. De mate waarin scheiding tussen de verschillende virtuele machines is aangebracht, blijft echter afhankelijk van de wijze waarop VM is geïnstalleerd en wordt onderhouden door de beheersorganisatie.

A.A.J. Breed
Is sinds 1984 werkzaam bij KPMG Klynveld EDP Audit. Nadat hij in 1989 de AMBI-opleiding had voltooid, heeft hij zijn studie voortgezet via de avondopleiding Economie aan de Universiteit van Amsterdam. Zijn specifieke kennisgebieden liggen op het gebied van de beveiligingsaspecten van IBM-besturingsprogrammatuur en database management-systemen. Daarnaast is hij betrokken bij verschillende systeemonderzoeken en participeert hij als kwaliteitsadviseur in automatiseringsprojecten.

Ing. G.H.M. Meijer

Betrouwbaarheid en beveiliging van het MVS besturingssysteem

Het Multiple Virtual Storage system van IBM is het grootste besturingssysteem voor IBM-mainframes en plug-compatibles. Het wordt veel gebruikt door middelgrote tot zeer grote organisaties voor de besturing van geautomatiseerde administratieve processen die een grote verwerkingscapaciteit vergen.

1 Inleiding

MVS is een besturingssysteem dat veel wordt gebruikt voor de besturing van geautomatiseerde administratieve processen die een grote verwerkingscapaciteit vergen. Het besturingssysteem wordt toegepast door middelgrote tot zeer grote organisaties die in grote mate afhankelijk kunnen zijn van de correcte werking ervan.

In het artikel "De audit van operating systems" (het openingsartikel in dit blad) is aandacht besteed aan de manier waarop een onderzoek naar de betrouwbaarheid en beveiliging van een besturingssysteem op een structurele manier kan worden uitgevoerd. Deze beschrijving verdeelt het onderzoek in vijf fasen, die resulteren in het opdoen van kennis over het te onderzoeken besturingssysteem, de door het management gestelde eisen, de manier waarop in opzet en bestaan aandacht is besteed aan betrouwbaarheid en beveiliging, alsmede de werking van deze maatregelen.

Omdat met name de uitvoering van fase 4 specifieke kennis vereist van het te onderzoeken besturingssysteem, wordt in dit artikel vooral aandacht besteed aan deze fase. Echter, omdat techniek niet los kan worden gezien van organisato-rische aspecten is, zij het beknopt, te-vens aandacht besteed aan fase 3.

Het doel van het artikel is een indruk te geven van de wijze waarop kan worden vastgesteld dat het besturingssysteem MVS op correcte wijze is geïnstalleerd en wordt beheerd. Hierbij zijn drie fasen te onderscheiden, die overeenkomen met de fasen 3 en 4 van bovengenoemd artikel:

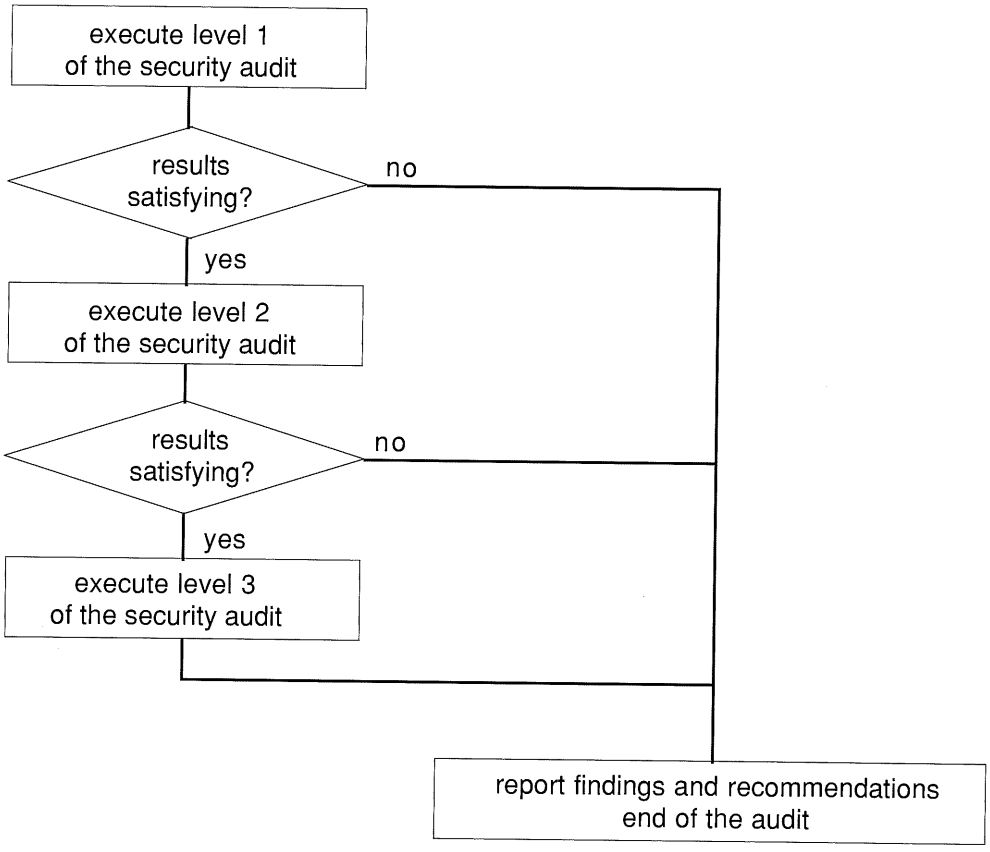
1. beoordeling van de maatregelen van beheersing van het operating system;
2. beoordeling van de belangrijkste beveiligingsopties die het systeem biedt;
3. beoordeling van belangrijke routines, die niet behoren tot het standaard besturingssysteem, maar een belangrijke invloed kunnen hebben op de functionaliteit hiervan.

In het algemeen kan worden gesteld dat de toetsingsnormen voor de auditor de eisen zijn die door het management zijn gesteld. Indien een fase een negatief resultaat van beoordeling oplevert, heeft het in principe geen zin het onderzoek voort te zetten. Het onderzoek kan het best worden afgerond met aanbevelingen voor het niveau waarop in eerste instantie verbeteringen zijn vereist. Een follow-up van het onderzoek kan dan bestaan uit het vaststellen of de aanbevelingen zijn opgevolgd en het verder oppakken van het onderzoek (tussentijdse veranderingen die zich hebben voorgedaan in acht nemend).

Het verloop van het onderzoek is schematisch weergegeven in figuur 1.

2 Karakteristieken van MVS

MVS staat voor Multiple Virtual Storage. Het is het grootste besturingssysteem voor IBM-mainframes en de plug-compatibles. Een grote hoeveelheid batch-, on-line- en time-sharing-processen kan door dit besturingssysteem tegelijkertijd worden bediend. MVS is in staat de "mix" van in uitvoering zijnde processen zodanig samen te stellen en te onderhouden dat het gebruik van de aanwezige computer-resources (vanuit efficiency-oogpunt) optimaal geschiedt.



Figuur 1. Niveaus en diepgang van onderzoek.

Het besturingssysteem kan draaien op de hardware die ontworpen is volgens de IBM 370-architectuur. Deze architectuur kent een aantal versies:
-- de 370-architectuur;
-- de 370/XA-architectuur (Extended Architecture);
-- de 370/ESA-architectuur (Enterprise Systems Architecture).

Het besturingssysteem MVS kent eveneens verschillende versies. In volgorde van ouderdom zijn dit:
-- MVS/SP1, ook wel aangeduid met MVS/370. Dit besturingssysteem kan, technisch gezien, draaien op alle versies van de architectuur, maar men zal het veelal aantreffen op een 370-computer;
-- MVS/SP2, ook wel aangeduid met MVS/XA. Dit besturingssysteem kan draaien op zowel een XA- als een ESA-computer;
-- MVS/SP3, ook wel MVS/ESA genoemd. Dit besturingssysteem kan alleen draaien op een ESA-computer.
IBM heeft nog een aantal andere besturingssystemen dat geschikt is voor de 370-architectuur. De belangrijkste hiervan zijn Virtual Machine (VM en VM/XA) en Virtual Storage Extended/System Product (VSE/SP). Het eerstgenoemde is een "host operating system", met behulp waarvan op één computer meerdere besturingssystemen kunnen worden gedraaid. Het laatste is een kleiner "broertje" van MVS. Vele kleinere organisaties gebruiken VSE/SP voor de besturing van hun informatiesystemen.

3 Betrouwbaarheid en beveiliging

In het hiervoor genoemde inleidende artikel is reeds aandacht besteed aan de begrippen betrouwbaarheid en beveiliging. Hierin werd betrouwbaarheid gedefinieerd als de mate waarin het besturingssysteem aan de specificaties voldoet. Voor MVS is één van die specificaties het "statement of integrity". Dit statement, afgegeven door IBM, luidt als volgt:

"MVS prevents any unauthorized program, using any defined or undefined system interface, from:
-- bypassing store or fetch protection;
-- bypassing OS password, VSAM password, or RACF security checking;
-- obtaining control in authorized state."

Het integrity statement komt er in essentie op neer dat processen in het interne geheugen van de computer onder besturing van MVS, elkaar niet ongeautoriseerd kunnen beïnvloeden. Tevens kunnen deze processen zichzelf niet autoriseren.

Het integrity statement bepaalt tevens dat een proces niet om de "MVS security checking" (toegangscontrole) heen kan zonder dat het daartoe is geautoriseerd. Deze toegangscontrole kan bestaan uit een passwordmechanisme van MVS zelf (een nogal eenvoudige vorm van toegangscontrole) of toegangscontrole met behulp van RACF (een meer geavanceerde vorm van toegangscontrole, een speciaal pakket ontwikkeld en op de markt gebracht door IBM).
Het accent van deze passage van het integrity statement komt echter steeds meer te liggen op toegangscontrole met behulp van RACF, omdat het passwordmechanisme van MVS een te laag niveau van beveiliging biedt en, mede daardoor, nagenoeg niet meer wordt gebruikt.

De interfaces tussen de gebruikersprocessen (bijvoorbeeld applicatieprogrammatuur zoals grootboek- of salarissystemen) en de processen van het besturingssysteem worden door IBM beschouwd als zijnde onder de verantwoordelijkheid van de gebruiker (lees: gebruikende organisatie). Aangezien deze interfaces grote invloed kunnen hebben op de integriteit van het totale systeem, heeft de gebruiker uiteindelijk zelf een grote verantwoordelijkheid ten aanzien van de integriteit.

Uit bovenstaande kan worden geconcludeerd dat de gebruikende organisatie twee elementen voldoende dient te beheersen teneinde een goede basis voor een betrouwbare gegevensverwerking te kunnen creëren (let wel: basis, omdat hier niet wordt gesproken over de beheersing van applicatieprogrammatuur):
-- de interfaces tussen de applicatieprocessen en het besturingssysteem;
-- de toegangscontrole tot gegevens en programma's.

In dit artikel wordt het accent gelegd op het eerste aspect. Het is het primaire doel van de audit van besturingssystemen. Het tweede aspect ligt meer op het vlak van de audit van een logisch toegangsbeveiligingssysteem.

4 Organisatie

Het hoofddoel van te houden interviews met het management en medewerkers van de desbetreffende afdeling is het beoordelen van de opzet van het stelsel van maatregelen met behulp waarvan de organisatie haar besturingssysteem beheerst. De belangrijkste onderwerpen die tijdens deze interviews aan de orde kunnen komen zijn:
-- structuur van de organisatie, alsmede functiebeschrijvingen, standaarden en voorschriften;
-- bestaande procedures binnen de desbetreffende afdeling:
. systeemprogrammering;
. operating;
. werkvoorbereiding;
-- verwachtingen en/of plannen voor de toekomst.

Verder kan in dit stadium reeds worden begonnen met het opvragen van informatie met betrekking tot het te onderzoeken systeem:
-- hardware-configuratie (compleet van het rekencentrum, ook van de niet te onderzoeken systemen, inclusief eventuele koppelingen met andere rekencentra);
-- software-configuratie (overzicht van de geïnstalleerde besturingssystemen, alsmede de ondersteunende besturingsprogrammatuur zoals databases, programmeer-tools, compilers, performance monitors, etc.);
-- planning en status van de logische toegangsbeveiliging.

Procedures voor installation en change and problem management, ondersteund door de techniek, moeten bewerkstelligen dat nieuwe of gewijzigde besturingsprogrammatuur alleen mag worden "gepromoveerd" naar de produktieomgeving indien zij op correcte wijze is getest en zij daartoe is geautoriseerd.

Documentatie van de geïnstalleerde systemen moet beschikbaar zijn. Hierbij dient speciale aandacht te worden besteed aan documentatie met betrekking tot de door de organisatie zelf aangebrachte wijzigingen. Voorbeelden hiervan zijn exits en supervisor calls, die een verandering van functionaliteit van (delen van) het besturingssysteem teweeg kunnen brengen.

5 MVS: een technische verdieping

Authorized Programs

Veel functies van het besturingssysteem zijn "gevoelig". Hiermee wordt bedoeld dat, om geen afbreuk te doen aan de integriteit van het systeem, gebruik van deze functies voorbehouden dient te zijn aan daartoe speciaal aangewezen processen of programma's. In MVS is het gebruik van deze gevoelige functies voorbehouden aan geautoriseerde programma's. Een programma is geautoriseerd indien dit:

-- draait in "supervisor state" of met een "system protection key", beide kenbaar gemaakt in het "Program Status Word" (dit is een gebied in het interne geheugen van de computer waarin de status van het desbetreffende proces wordt bijgehouden), of

-- is geautoriseerd door de "Authorized Program Facility": een dergelijk programma wordt ook wel APF-programma genoemd.

De eerste mogelijkheid is voornamelijk toebedeeld aan de MVS-besturingsprogrammatuur zelf, de tweede is een door de gebruiker te beheersen interface tussen zijn processen en het besturingssysteem. Belangrijk hierbij is echter dat een APF-programma zichzelf in "supervisor state" kan brengen, hetgeen een nog ruimere bevoegdheid geeft.

In dit artikel wordt met geautoriseerd programma dus bedoeld dat het programma gevoelige systeemfuncties mag gebruiken. Het heeft geen betrekking op het al dan niet mogen benaderen van bestanden. Deze laatste vorm van autorisatie wordt bepaald met behulp van logische toegangsbeveiliging.

Authorized Program Facility

De Authorized Program Facility wordt toegepast om ondersteunende besturingsprogrammatuur en/of gebruikersprogramma's te autoriseren, met andere woorden: in staat te stellen gevoelige systeemfuncties te gebruiken.
MVS beschouwt een programma als APF-authorized als aan de volgende voorwaarden is voldaan:

-- Het programma in het interne geheugen is geladen vanuit een library die aan MVS is gedefinieerd als een APF-library.

-- Het programma in deze APF-library is "ge-linked" met een autorisatiecode gelijk aan "1" (AC = 1).

Omdat MVS geen mechanisme biedt om het gebruik van de autorisatiecode te beperken, is het beheer van de APF-libraries de enige mogelijkheid om de Authorized Program Facility te beheersen. De APF-libraries worden aan MVS kenbaar gemaakt tijdens systeeminitialisatie (IPL) door middel van speciale parameters, die zijn vervat in het member IEAAPFxx.

Audit

Gewenste parameter members kunnen worden uitgelijst met behulp van speciale MVS-utilities zoals IEBPTCH en IEBLST.

Met betrekking tot de APF-authorized libraries dient ook aandacht te worden besteed aan de parameter LNKAUTH = LNKLST/APFTAB. Deze parameter geeft aan of alleen de libraries uit IEAAPFxx APF-authorized zijn, of dat aan deze lijst de gehele linklist-concatenation wordt toegevoegd.

Supervisor Calls

Gebruikersprogramma's mogen dus in principe de gevoelige functies van het besturingssysteem niet uitvoeren. Een voorbeeld van zo'n functie is het openen en sluiten van bestanden. Echter een (niet geautoriseerd) gebruikersprogramma heeft deze functies vaak nodig.

MVS heeft een mechanisme dat ongeautoriseerde programma's in staat stelt aan het besturingssysteem te vragen de functie uit te voeren ten behoeve van het aanvragende programma. Dit mechanisme heet een Supervisor Call (SVC). Een ongeautoriseerd programma geeft een "call" naar de supervisor, die ervoor zorg draagt dat de gevraagde functie wordt uitgevoerd. Het aanvragende programma zal wachten tot het antwoord krijgt van de supervisor. Pas daarna kan het weer verder met de verwerking.

MVS maakt onderscheid tussen "IBM-SVCs" en "USER-SVCs". IBM-SVCs behoren tot het standaard MVS-besturingssysteem. USER-SVCs kunnen zijn ontwikkeld door software-leveranciers voor aanvullende (ondersteunende) produkten of door de systeemprogrammeur zelf. SVCs identificeren zich door een nummer. IBM-SVCs zijn genummerd van 0 tot en met 199; USER-SVCs van 200 tot en met 255.

Een SVC is dus een interface tussen ongeautoriseerde en geautoriseerde pro-

gramma's. Daardoor verdient dit mechanisme de speciale aandacht van de auditor. Belangrijkste aspect hierbij is: hoe gaat men om met de USER-SVCs? Welke eisen stelt de automatiseringsorganisatie aan het gebruik van SVCs? Lang niet alle USER-SVCs blijken in de praktijk te voldoen aan daaraan te stellen eisen met betrekking tot integriteit.

USER-SVCs worden kenbaar gemaakt aan het systeem tijdens systeeminitialisatie. MVS heeft hiertoe een speciale set parameters. Hierin is het mogelijk USER-SVCs APF-restricted te maken. Dit wil zeggen dat het gebruik van deze SVC is voorbehouden aan geautoriseerde programma's.

Audit
Het parameter member voor identificatie en activering van USER-SVCs heet IEASVCxx. De hierin opgenomen USER-SVCs zijn automatisch actief (als de routine in het systeem aanwezig is).

Belangrijk aspect van de beoordeling van SVCs is dat software-leveranciers in sommige gevallen een standaard IBM-SVC overschrijven.

Nagegaan dient te worden of de organisatie een evaluatie heeft uitgevoerd teneinde zoveel mogelijk USER-SVCs APF-restricted te maken. Belangrijk aspect in dezen is dat het standaard parameter member (IEASYS00) alle USER-SVCs bevat en dat geen enkele daarvan APF-restricted is.

Een source-code review van SVCs kan van belang zijn, maar alleen op niveau 3 van de audit.

Exits
Het MVS-besturingssysteem, alsmede de meeste ondersteunende besturingsprogrammatuur, is voorzien van "exit entry points". Dit zijn vastgestelde "plaatsen" in de besturingsprogrammatuur waar een systeemprogrammeur zelf programma-coding kan toevoegen. Dit wordt gedaan om te kunnen voldoen aan specifieke eisen of wensen ten aanzien van de werking van het besturingssysteem. De toegevoegde code wordt exit genoemd.

Exits zijn voor het onderzoek van belang om een aantal redenen. Een exit kan, in theorie, de functionaliteit van het besturingssysteem veranderen en draait bovendien in geautoriseerde status.

Audit
Exits kunnen worden geïdentificeerd met behulp van de opstart-routines van de desbetreffende subsystemen. Zo worden JES2 exits geladen met behulp van de JES2 initialisatieparameters (JES2 is een Job Entry Subsystem van MVS). Voor andere subsystemen kan weer gelden dat ze worden geladen met behulp van de Job Control Language die het subsysteem opstart.
Belangrijk is te bepalen wat in opzet de functie is van elke exit. De werkelijkheid kan worden bepaald door de exit te testen (door het submitten van een batchjob kunnen bijvoorbeeld JES2 exits worden getest), of door een source-code review uit te voeren.

I/O-appendages
Een I/O-appendage lijkt in functionaliteit zeer veel op een exit. I/O-appendages zijn bijvoorbeeld bedoeld voor de aansturing van bijzondere I/O-randapparatuur. Voor I/O-appendages is echter onderscheid gemaakt in appendages die slechts mogen worden gebruikt door geautoriseerde programma's en appendages die ook mogen worden gebruikt door ongeautoriseerde programma's. Beide verdienen de aandacht van de auditor, met een extra accent op het laatste type. Het onderscheid wordt aan het systeem kenbaar gemaakt via systeeminitialisatieparameters.

Audit
Qua karakteristieken lijken I/O-appendages veel op exits. Ze kunnen op dezelfde wijze worden beoordeeld, met dien verstande dat I/O-appendages die kunnen worden gebruikt door ongeautoriseerde programma's, zijn opgenomen in het initialisatieparameter member IEAAPP00.

Program Properties Table
Soms is het nodig een programma speciale bevoegdheden te verlenen. Het kan bijvoorbeeld nodig zijn voor een programma een speciaal deel van het interne geheugen te kunnen benaderen. Via de Program Properties Table (PPT) is het mogelijk dit soort bevoegdheden aan programma's toe te kennen.

De PPT wordt in MVS gedefinieerd met behulp van systeeminitialisatieparameters die zijn vervat in het member SCHED00. Door IBM wordt hiervan een standaard geleverd, die na generatie door de systeemprogrammeur kan worden aangepast.

Audit
In de standaard van het parameter member (SCHED00) kunnen meer programma's zijn opgenomen dan voor de specifieke installatie zijn benodigd. Een voorbeeld hiervan is dat ten behoeve van het Job Entry Subsystem veelal twee programma's zijn opgenomen in SCHED00, terwijl maar één hiervan nodig is.

Deze "overtollige" programma's zijn, evenals de door de organisatie toegevoegde programma's van belang. De organisatie dient de inhoud van dit parameter member zorgvuldig te hebben geëvalueerd, en veranderingen hiervan dienen te zijn gedocumenteerd.

Job Entry Subsystem
Het Job Entry Subsystem (JES2 of JES3) is het primaire subsysteem van MVS. Alle processen, behalve de overige subsystemen, draaien met behulp van JES2. De voornaamste taak van dit subsysteem is het afhandelen van de batch-verwerking.
Voor het afhandelen van batch jobs kan een aantal job-classes worden gedefinieerd. Aan iedere job-class kan een prioriteit worden toegekend, evenals een aantal speciale bevoegdheden. Zo is het mogelijk voor job-classes toe te staan dat in de Job Control Language (JCL) systeemcommando's zijn opgenomen. Deze systeemcommando's zijn normaliter voorbehouden aan de operator.

Batch jobs kunnen worden aangeleverd vanaf bijvoorbeeld diskdrives (een submit-commando zorgt ervoor dat een batch job wordt geladen vanaf een diskdrive), "internal readers" en vanuit andere computersystemen. De laatste methode wordt aangeduid met de term Network Job Entry (NJE).

Indien van NJE gebruik wordt gemaakt, dienen de systemen waarmee communicatie plaatsvindt aan JES kenbaar te worden gemaakt. Deze systemen worden "nodes" genoemd. Aan iedere node kunnen bevoegdheden worden toegekend. Deze bevoegdheden kunnen variëren van het slechts mogen aanleveren van jobs tot het genereren van systeemcommando's (met behulp waarvan bijvoorbeeld subsystemen kunnen worden gestopt).

Audit
Alle definities van job-classes, network-job-entry-nodes en internal readers moeten worden geëvalueerd op hun bevoegdheden. Hierbij dient te worden vermeld dat bij default alle bevoegdheden zijn toegekend en dat de organisatie zelf deze bevoegdheden zal moeten intrekken.
De definitie van het Job Entry Subsystem geschiedt met behulp van JES-initialisatieparameters. Het desbetreffende parameter member kan worden gelokaliseerd met behulp van de Job Control Language die dit subsysteem opstart.

Time Sharing Option
De Time Sharing Option (TSO) is een subsysteem onder MVS dat kan worden gezien als een ontwikkelfaciliteit. Hiermee hebben gebruikers een soort eigen virtuele computer waarmee programma's kunnen worden geschreven, gecompileerd en getest. In feite heeft men met behulp van deze faciliteit directe toegang tot bestanden.
Tevens is het mogelijk output op het scherm te bekijken voordat deze naar de printer wordt gestuurd.
TSO wordt met name gebruikt door systeem- en applicatieprogrammeurs. Gebruikers van dit subsysteem worden gedefinieerd in een speciale systeem-dataset, SYS1.UADS. Hierin kunnen per gedefinieerde gebruiker faciliteiten worden aangegeven waartoe hij of zij is geautoriseerd. Voorbeelden van deze faciliteiten (buiten normale "edit"-functies) zijn:
-- het genereren van operator-commando's;
-- het wijzigen of nieuw aanbrengen van definities van TSO-users;
-- het aanleveren van batch jobs;
-- het mogen gebruiken van I/O-devices die "physical mounting" vereisen (een tape moet op de tape unit worden gehangen).

Audit
Tijdens de audit dient te worden vastgesteld welke bevoegdheden aan TSO-users zijn toegekend en of dit in verhouding is met de functies die de desbetreffende gebruikers uitoefenen.

Systeem Logging
MVS biedt een aantal logging-faciliteiten met behulp waarvan het gebruik van het besturingssysteem kan worden gecontroleerd. Een organisatie dient vast te stellen welke logging-faciliteiten benodigd zijn om de werking van het bestaande stelsel van maatregelen en procedures te kunnen controleren.
De verschillende soorten logging in MVS zijn:

-- System Log: bevat job-afhankelijke informatie, alsmede operator commando's en systeemboodschappen. Deze log staat in het bestand SYSLOG.MVS;
-- Een logging van fouten in apparatuur en programmatuur. Deze logging wordt opgenomen in het bestand SYS1.LOGREC;
-- Trace facility. Dit is een zeer gedetailleerde registratie van activiteiten van het systeem, echter slechts over een zeer korte periode. Deze logging (die in het interne geheugen van de computer wordt opgenomen) is voornamelijk bestemd voor engineers van IBM om, in geval van storing, te kunnen vaststellen wat er in het systeem fout is gegaan. Er zijn drie verschillende trace facilities:
. master trace;
. generalized trace;
. system trace;
-- System Management Facility (SMF). Deze logging kan, afhankelijk van de opties die daarvoor zijn gekozen, gedetailleerde informatie bevatten over het besturingssysteem en de verwerking van jobs. Zo kan het openen en sluiten van bestanden door processen worden geregistreerd.
SMF-records worden ook gebruikt als input voor de programmatuur voor doorbelasting van het computergebruik. Ondersteunende besturingsprogrammatuur die een eigen logging bijhoudt, doet dit vaak met behulp van SMF-logging (zoals het beveiligingspakket RACF).

Audit
Van belang is dat de logging-opties het gewenste resultaat hebben. Met andere woorden: de logging moet de volledige informatie opbrengen die de organisatie denkt nodig te hebben.
Voor controle op operator-activiteiten kan het van belang zijn dat de hardcopy-log wordt gecontroleerd door bijvoorbeeld de shiftleader en dat deze wordt geparafeerd.

Systeemgeneratie
Systeemgeneratie is het, met behulp van distributietapes van IBM, creëren van systeem-libraries die voldoen aan de specifieke eisen die vooraf aan het te genereren systeem zijn gesteld. Daarnaast wordt de Input/Output-configuratie gedefinieerd. Er zijn twee typen systeemgeneraties te onderkennen:
-- SYSGEN: het creëren van de systeem-libraries. Dit proces vindt plaats met behulp van het System Modification Program (Extended) (SMP of SMP/E);
-- IOGEN: het definiëren van de I/O-configuratie. Hiertoe dient het I/O Configuration Program (IOCP).

Een combinatie van een SYSGEN en een IOGEN (bijvoorbeeld als voor de eerste keer MVS wordt gegenereerd) wordt ook wel aangeduid met de term "complete SYSGEN".

Tijdens de SYSGEN wordt onder meer de basis gelegd voor de systeeminitialisatieparameters. Na de systeemgeneratie heeft de systeemprogrammeur de mogelijkheid deze parameters aan te passen.

Tijdens de systeemgeneratie wordt het prefix bepaald van de namen van systeem-libraries. De default voor dit prefix is SYS1, maar er kan ook een andere waarde voor worden opgegeven. Indien wordt bepaald dat het prefix voor systeem-libraries SYS3 zal zijn, zal de dataset waarin de TSO-users zijn gedefinieerd SYS3.UADS gaan heten. Deze systeem-datasets komen op een disk pack te staan die dan System Residence Volume (SYSRES-volume) wordt genoemd.

In het IOGEN-proces, dat eveneens geschiedt op basis van SYSGEN-macro's worden bijvoorbeeld disk packs en tape units, auto-answer en auto-call units (modems) gedefinieerd.

Audit
Veelal wordt meer hardware gedefinieerd dan op het moment van systeemgeneratie benodigd is. Dit wordt gedaan om te voorkomen dat systeemgeneratieprocessen te vaak moeten worden uitgevoerd.
Van belang is de naam van de master catalog in SYSCATLG en het prefix voor systeem-datasets. Met behulp van onder andere deze gegevens kan worden bepaald dat de auditor de informatie uit het juiste systeem verkrijgt.

Indien er auto-answer en auto-call units zijn gedefinieerd, dienen de procedures hieromtrent te worden beoordeeld.

Systeeminitialisatie
Reeds is gebleken dat de systeeminitialisatie een grote rol speelt in de integriteit van MVS. Het systeeminitialisatieproces begint met het IPL-commando dat wordt gegeven door de operator. Met dit IPL-commando dient een disk pack te worden opgegeven waar vanaf het bestu-

ringssysteem zal worden geladen (het is mogelijk meerdere SYSRES-volumes te hebben). Het IPL-commando resulteert in het opstarten van het Nucleus Initialization Program. Welke Nucleus wordt geladen is afhankelijk van het SYSRES-volume dat met het IPL-commando wordt gekozen.

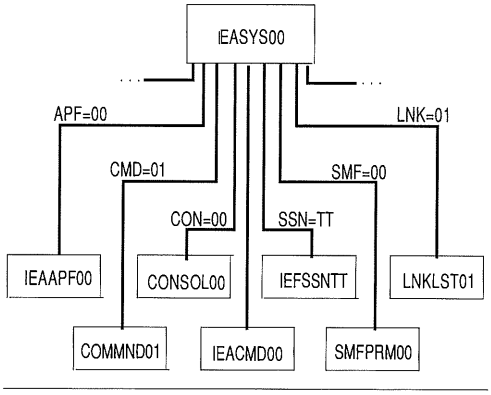
Het opstartende besturingssysteem zal een set initialisatieparameters inlezen, die is opgenomen in de dataset SYS1.PARMLIB. Deze dataset staat op het SYSRES-volume. Iedere MVS-component wordt beschreven of gedefinieerd met behulp van één of meer parameter members.

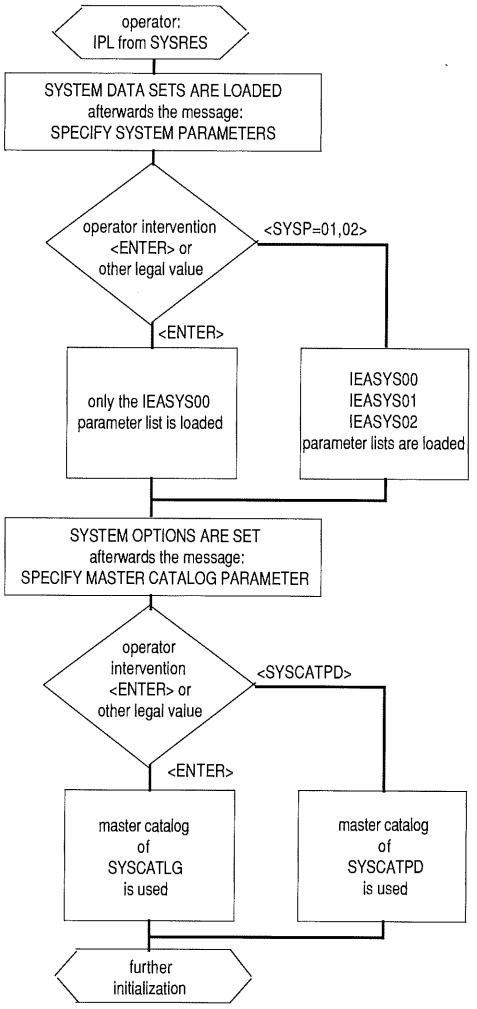
Deze parameter members zijn vervat in een boomstructuur, waarbij één parameter member bovenaan staat, verwijzende naar een aantal parameter members op het onderliggende niveau, etc. Het hoogst gelegen member heet IEASYS00. Het is mogelijk één of meer alternatieve boomstructuren te creëren. Hiertoe wordt dan een ander IEASYSxx-parameter member gecreëerd (bijvoorbeeld IEASYS03). In figuur 2 is een dergelijke boomstructuur afgebeeld.

Tijdens het IPL-proces zal het besturingssysteem de operator vragen systeemparameters te specificeren (SPECIFY SYSTEM PARAMETERS). Indien de operator hierop geen respons geeft zal alleen de boomstructuur onder IEASYS00 worden gebruikt om het systeem verder op te bouwen. Indien de operator SYSP=03 als respons geeft, zal eerst IEASYS00 worden gelezen en daarna IEASYS03, waarbij geldt dat de laatste de eerste overschrijft en/of aanvult.

De volgende vraag die het opstartende

Figuur 2. Boomstructuur van systeem-initialisatie parameters.





Figuur 3. IPL-proces.

systeem aan de operator stelt, is het specificeren van de MASTER CATALOG. De master catalog is een belangrijk onderdeel van het MVS-besturingssysteem. Met behulp hiervan worden (bijna) alle bestanden gelokaliseerd waarmee het systeem zal gaan werken. De naam van de master catalog staat in een SYSCATxx-member in de systeemdataset SYS1.NUCLEUS. Ook voor SYSCATxx geldt dat er meerdere kunnen zijn gedefinieerd. Hiermee heeft de operator de mogelijkheid een alternatieve master catalog aan te wijzen, hetgeen kan resulteren in gebruik van bijvoorbeeld andere datasets. De default voor SYSCATxx is SYSCATLG.

Het verloop van het IPL-proces is schematisch weergegeven in figuur 3.

Een parameter in IEASYSxx geeft de operator al dan niet de mogelijkheid om specifieke parameters te "overrulen".

Deze parameter (OPI=YES/NO) heeft als default YES, hetgeen inhoudt dat operator intervention wordt toegestaan. Handhaving van deze default heeft een grote invloed op de beheersbaarheid van het besturingssysteem. Via de systeem-logging zal dan moeten worden gecontroleerd welke interventions de operator eventueel heeft uitgevoerd en wat de consequenties daarvan zijn ten aanzien van de functionaliteit van het besturingssysteem.

Audit
Zoals vermeld heeft de systeeminitialisatie grote invloed op te beoordelen onderdelen van het MVS-besturingssysteem. De beoordeling van de systeeminitialisatie bestaat voornamelijk uit het beoordelen van de inhoud van de belangrijke systeeminitialisatie parameter members, waarvan de meeste in dit artikel aan de orde zijn gekomen.

6 Beveiliging van systeem-datasets

Nadat de beoordeling heeft plaatsgevonden van de beveiligingsopties van MVS zal moeten worden bepaald welk niveau van logische toegangsbeveiliging deze opties beschermt. Het niet beschermen van deze beveiligingsopties kan resulteren in het ongeautoriseerd veranderen ervan door mensen binnen en buiten de automatiseringsorganisatie.

Audit
Er zal moeten worden vastgesteld welk niveau van beveiliging wordt geboden voor systeem-datasets. De beveiliging van datasets zoals deze door MVS wordt geboden, is van een te laag niveau. Een aanvullend pakket zoals bijvoorbeeld RACF is hiervoor benodigd.

7 Tot slot

De fasen 2 en 3 van de in de inleiding beschreven fasering omvatten het onderzoek van de technische implemen-tatie van het MVS-besturingssysteem. Aangegeven is dat in fase 2 de belangrijkste beveiligingsopties dienen te worden beoordeeld die het besturingssysteem biedt.

Het belangrijkste aanknopingspunt hiervoor zijn de systeeminitialisatieparameters. Hierin staat het grootste deel van de opties die in deze fase van het onderzoek dienen te worden onderzocht.De vraag die voor ieder onderdeel steeds weer zal moeten worden gesteld is: "Waarom"?
De organisatie zal moeten kunnen motiveren waarom zij al dan niet voor bepaalde opties heeft gekozen en welke compenserende maatregelen zij eventueel heeft genomen.
De meeste beveiligingsopties hebben een standaardwaarde (dit is de waarde die aan deze parameters wordt gegeven tijdens het systeemgeneratieproces). Deze standaardwaarden bieden in het algemeen een zeer laag beveiligingsniveau. Dit komt enerzijds doordat dit nodig is om het systeem te kunnen genereren, anderzijds doordat deze beveiligingsdrempels soms een negatieve invloed hebben op de performance van het besturingssysteem. In de praktijk blijkt dat de automatiseringsorganisatie deze standaardwaarden vaak niet heeft aangepast. Soms uit onwetendheid, soms uit overwegingen van efficiency.

Een en ander heeft tot gevolg dat in de praktijk een onderzoek vaak niet verder gaat dan fase 2.

Een betrouwbaar besturingssysteem leidt niet per definitie tot een betrouwbare gegevensverwerking. Het draagt ertoe bij. Aspecten als het beoordelen van de betrouwbaarheid van de logische toegangsbeveiliging en van de ontwikkeling en beheersing van applicatieprogrammatuur zijn namelijk niet aan de orde geweest. Het zijn de volgende stappen die dienen te worden genomen om te kunnen komen tot een oordeel hieromtrent.

Ing. G.H.M. Meijer
Is sinds 1985 werkzaam bij KPMG Klynveld EDP Audit. Hij heeft in 1989 zijn AMBI-opleiding voltooid en studeert momenteel post-doctorale EDP-Auditing aan de Erasmus Universiteit te Rotterdam. Hij heeft EDP-audit-onderzoeken uitgevoerd op het gebied van rekencentra, besturingssystemen, beveiligingspakketten en SWIFT.

UNIX is op een groot aantal verschillende architecturen beschikbaar, vanaf PC's tot IBM-main-frames. UNIX-systemen zijn goed te beveiligen, maar een te hoog beveiligingsniveau resulteert in een onwerkbare situatie. Getracht zal moeten worden een balans te vinden tussen beveiliging en bruikbaarheid.

Drs.ing. J.C. van Winkel RI

UNIX-beveiligingsaspecten

1 Inleiding

In dit artikel wordt een overzicht gegeven van in het UNIX[1]-besturingssysteem opgenomen beveiligingen, en wordt aangegeven op welke wijze de EDP-auditor kan vaststellen of een UNIX-systeem afdoende is beveiligd.

Eerst zal kort worden ingegaan op het UNIX-besturingssysteem. Hierin komen de geschiedenis van UNIX, het universele karakter van UNIX en (een kleine inleiding in) de implementatie van UNIX aan de orde.

Vervolgens wordt een beschrijving gegeven van een aantal mogelijkheden en handvatten die UNIX biedt vanuit het beveiligingsstandpunt. Bij UNIX geldt (wellicht nog meer dan bij andere systemen) dat in principe het systeem potdicht kan worden gemaakt, wat echter een onwerkbare situatie oplevert. Veel moeilijker is het een balans te vinden tussen beveiliging en bruikbaarheid.

Tot slot komt een aantal beveiligingstips aan bod.

2 UNIX

UNIX is een multi-user, multi-tasking-besturingssysteem. Dat wil zeggen dat het systeem meerdere gebruikers onderscheidt (multi-user) met ieder hun eigen werkgebied en privileges. Het systeem zorgt ervoor dat deze gebruikers elkaar niet in de weg zitten.

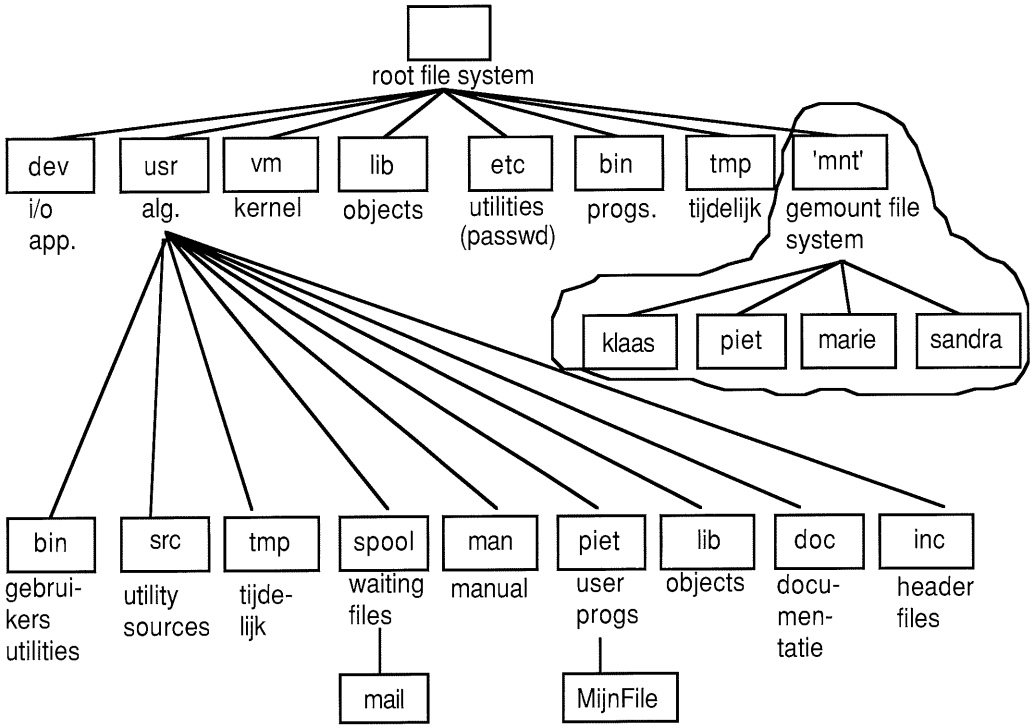
De gebruikers kunnen tegelijkertijd met één of meer taken bezig zijn (multi-tasking). Zo is het mogelijk dat een gebruiker in een tekstverwerkingsprogramma een tekst invoert, terwijl "op de achtergrond" voor deze gebruiker uit een database een aantal records wordt geselecteerd. Weer een andere gebruiker kan tegelijkertijd in een electronic mail-programma bezig zijn, terwijl ook een programma gecompileerd wordt.

UNIX verzorgt ook de verdeling van ruimte van de externe opslagcapaciteit onder de gebruikers. Hiertoe is deze opgedeeld in directories en files. Dit geheel noemt men het file-systeem. In het file-systeem wordt van alle bestanden bijgehouden wie de eigenaar is, en aan wie de eigenaar toestemming heeft gegeven om de bestanden te gebruiken.

Het UNIX-file-systeem is hiërarchisch (zoals ook bij de Apple Macintosh en MS-DOS-machines). Dit wil zeggen dat bestanden opgeslagen zijn onder bepaalde directories, zoals dossiers ook in hangmappen kunnen worden opgenomen. Het is echter ook mogelijk in directories weer directories op te nemen (zoals in een dossier een envelop kan worden opgenomen). Zo wordt in UNIX (meestal) aan alle gebruikers een eigen directory gegeven. Gebruikers kunnen in deze directory zelf weer directories aanbrengen om orde te scheppen in hun bestanden.

De boomstructuur van een schijf kan in zijn geheel onder een directory van een andere schijf worden opgehangen ("mounten"). Hierdoor lijkt het voor de gebruiker alsof er maar één schijf is. De gebruiker hoeft niet eens te weten dat er meerdere schijven zijn. Dit vergemakkelijkt de navigatie door het file-systeem. In de figuren zijn twee file-systemen op-

1 UNIX is een geregistreerd handelsmerk van AT & T Bell Laboratories.



Figuur 1. Een compleet file-systeem, met daarin opgenomen het file-systeem van figuur 2.

genomen: het root file-systeem en een tweede file-systeem dat wordt "opgehangen" onder de "mnt" directory. Hoe UNIX ervoor zorgt dat gebruikers hun bestanden kunnen afschermen voor andere gebruikers, zullen we zien in het hoofdstuk over beveiliging.

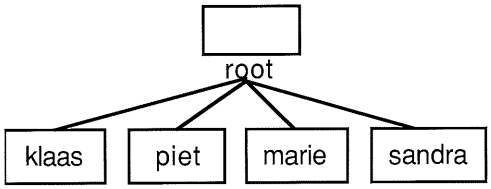
Geschiedenis

Het allereerste begin van UNIX stamt uit 1969. Ken Thompson leidde UNIX af uit het ambitieuze MULTICS-project bij het Massachusetts Institute of Technology (MIT), AT&T Bell Labs en General Electric. In 1975 werd een versie gemaakt die niet meer zoals eerdere versies in Assembler was geschreven, maar (voor 90%) in de taal C.
De programmeertaal C is een blokgestructureerde hogere programmeertaal die derhalve lijkt op talen als Pascal en Algol. Zo biedt C de faciliteiten die van een dergelijke hogere programmeertaal verwacht mogen worden: het kunnen afleiden van nieuwe datatypes uit andere datatypes (data-abstractie), procedures en functies met parameters en de "normale" controlestructuren zoals if, while en case.

Daarnaast biedt C faciliteiten die in Assembler voorhanden zijn. Zo kunnen geheugenlokaties direct worden gewijzigd en bestaan er constructies die direct op de onderliggende processor-architectuur afbeeldbaar zijn. Een programmeur kan bijvoorbeeld de compiler suggereren bepaalde variabelen in registers op te slaan in plaats van in het geheugen, omdat het variabelen betreft die zeer veelvuldig gebruikt zullen worden. Deze aspecten betekenen dat de C-compiler "eenvoudig" zeer efficiënte machinecode kan genereren voor de C-statements.

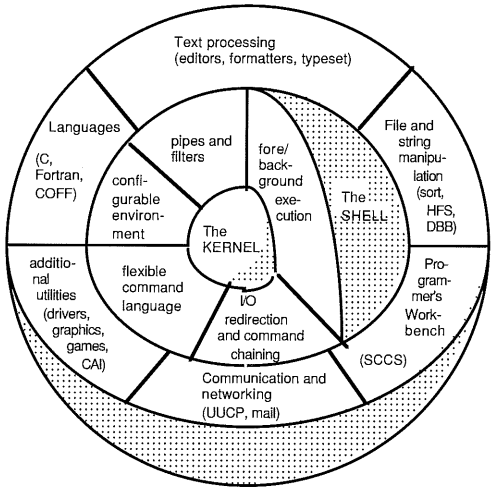
Door de combinatie van de mogelijkheden van Assembler en van een hogere programmeertaal zijn programma's geschreven in C snel in executie, sneller geschreven dan Assembler en overdraagbaar naar andere machines, zolang niet gebruik wordt gemaakt van specifieke machineafhankelijke mogelijkheden.

Figuur 2. Het file-systeem dat in de directory/mnt in het root-file-systeem gemount wordt.



Dit laatste aspect zorgde ervoor dat UNIX nu op een groot aantal totaal verschillende architecturen beschikbaar is: van een kleine PC tot de grote IBM-mainframes en zelfs op CRAY-supercomputers. Daarbij komt dat AT&T Bell Labs een beleid voerde gericht op stimulatie van UNIX op universiteiten. Hierdoor zijn nu vele met UNIX groot geworden afgestudeerden in de informatica op leidinggevende posities aangeland die (wellicht) kiezen voor het systeem waarmee ze bekend zijn.

Opbouw
UNIX is een besturingssysteem dat is opgebouwd rond een relatief kleine kern (de "kernel"). Om deze kernel heen is de shell (schil) gelegd. Dit is wat de gebruiker ziet van UNIX. In de shell tikt de gebruiker commando's in. De shell zal



Figuur 3. De opbouw van UNIX.

de commando's interpreteren en programma's opstarten. Het maakt voor de gebruiker geen verschil of een commando een ingebouwd shell-commando is, een programma uit de standaardprogrammabibliotheek, een zelfgemaakt programma, of zelfs een bestand met daarin weer een aantal shell-commando's (een zogenaamd shell script). De shell zoekt zelf (uit een lijst van alternatieven) uit waar het programma kan worden gevonden en hoe het moet worden uitgevoerd. De programma's uit de standaardbibliotheek zorgen ervoor dat UNIX er uitziet en reageert zoals het doet. De shell bevat namelijk een zeer klein aantal ingebouwde commando's die de shell zelf kan uitvoeren. De meeste commando's zijn in feite programma's die zijn opgeslagen in de standaardprogrammabibliotheek.

Dit laatste geldt ook voor veel beveiligingsaspecten van UNIX. De basisbeveiligingsaspecten van UNIX zitten in de kernel, maar het veranderen van beveiligingsopties wordt gedaan door programma's in de standaardprogrammabibliotheek. Hierin schuilt zowel de kracht als de zwakte van het UNIX-beveiligingssysteem.

3 UNIX-beveiligingen en controle daarop

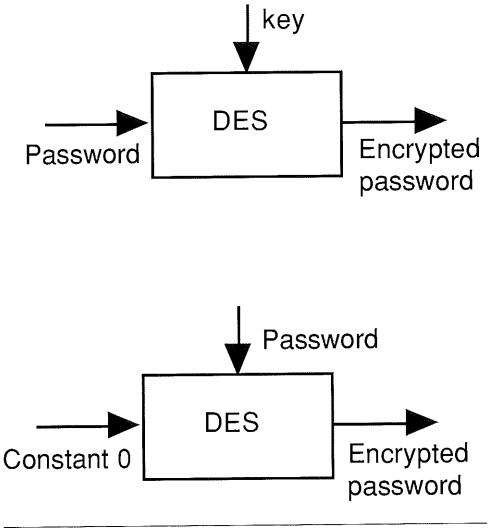
De password file
Het gebruikersbeheer vindt plaats in de password file (naam: /etc/passwd) en de group file (naam: /etc/groups). Beide staan dus in directory /etc. Gedeelten van deze bestanden zijn vrijelijk voor iedereen te lezen, maar alleen door de systeembeheerder te overschrijven (als de systeembeheerder dat goed heeft ingesteld). Onderaan zijn in figuur 4 als voorbeeld twee regels uit een denkbeeldige password file gegeven. De verschillende velden van de regel zijn door dubbele punten gescheiden.

Het eerste veld van de password file bevat de gebruikersnaam. Dit is ook de naam waaronder de gebruiker inlogt.

Het tweede veld is het versluierde wachtwoord van de gebruiker. Deze versluiering vindt zó plaats dat het onmogelijk is uit de versluiering het oorspronkelijke wachtwoord te destilleren. Er wordt gebruik gemaakt van het bekende algoritme DES. Normaal wordt DES gebruikt om met een geheime sleutel een tekst (of een password) te versluieren tot de geëncrypte tekst (of password). Met behulp van dezelfde sleutel is het mogelijk de oorspronkelijke tekst terug te krijgen. Om deze weg terug af te sluiten worden de rollen van de sleutel en het password verwisseld: het password wordt als sleutel gebruikt om een niet-

Figuur 4..

```
jc:cHgyMlO8xNAhY:15:10:Jan.Christiaan.van.Winkel:/usr2/jc:/bin/ksh
jc2:cHgyMlO8xNAhY,c/:15:10:jc.met.expiration:/usr2/jc:/bin/ksh
```



Figuur 5.

geheime constante (bij UNIX meestal 0) te versluieren. DES heeft de eigenschap dat zelfs als men de oorspronkelijke tekst kent (bij UNIX: 0) en het resultaat na versluiering (bij UNIX in de password file opgenomen), het toch niet mogelijk is de bij versluiering gebruikte sleutel af te leiden (het oorspronkelijke wachtwoord).

In figuur 5 is bovendien de normale password-encryptie met DES weergegeven en onderin de UNIX password-encryptie met DES.

Bij het inloggen wordt het ingegeven wachtwoord versluierd, en vergeleken met de opgeslagen versie in de password file. Om het potentiële krakers nog moeilijker te maken, is nog een tweetal aanpassingen gemaakt.

Zo wordt het DES-algoritme 25 keer herhaald uitgevoerd om het proces van password-encryptie te vertragen. Daarnaast genereert het systeem, ongevraagd en onzichtbaar voor de gebruiker, twee karakters (het "salt") die een interne tabel in het DES-algoritme enigszins modificeren. Hierdoor is het niet mogelijk snelle DES-chips te gebruiken, omdat in deze chips de tabel vast is opgeslagen. Hierdoor wordt een inbraakpoging op basis van "brute kracht" bemoeilijkt. De twee karakters worden onversluierd direct na het versluierde wachtwoord in de password file opgeslagen.

Mensen kiezen als wachtwoord vrijwel altijd bestaande woorden of namen. Door alle woorden in de woordenlijst volgens de UNIX-manier te versluieren en het resultaat te vergelijken met de versluiering in de password file, kan een hacker een aantal passwords vinden. Door het toevoegen van het salt moet de hacker alle woorden uit de woordenlijst even zovele malen versluieren als er verschillende salts in de password file voorkomen. Dit maakt het probleem van het kraken van het password zoveel factoren moeilijker als er verschillende salts in de password file staan.

Deze drie maatregelen (DES als eenwegs-algoritme, moedwillig 25 maal vertragen en het salt) moeten ervoor zorgen dat wachtwoorden niet meer uit de password file te destilleren zijn.

Toch werkt deze strategie niet altijd. In het boek *Het Koekoeksei* (zie recensie op pagina 61 van dit nummer), verhaalt Cliff Stoll hoe een hacker toch door de password-beveiliging heen kon komen. Deze hacker was weliswaar al binnengekomen op een bepaalde gebruikerscode, maar wist met een truc aan het systeem het wachtwoord van veel meer gebruikers te ontfutselen. Hiertoe kopieerde hij de pass- word file naar zijn PC, en liet op die PC alle woorden uit een speciaal samengestelde lijst versluieren. De PC kon dagenlang rekenen zonder dat dit door iemand op het systeem dat het slachtoffer was, kon worden gemerkt.

Om deze strategie tegen te gaan, vragen nieuwe versies van UNIX bij het opgeven van een nieuw wachtwoord altijd om een wachtwoord van minstens zes karakters dat bovendien minstens twee letters en minstens één cijfer of speciaal karakter bevat. Hierdoor wordt de gebruiker gedwongen om aan een eventueel bestaand woord een aantal vreemde karakters toe te voegen.

Terug naar de password file. Het is mogelijk direct ná het password (met een komma gescheiden) twee karakters op te nemen uit de verzameling ./a-zA-Z. Deze karakters geven aan met welke frequentie de gebruiker zijn of haar wachtwoord moet veranderen. Het tweede karakter geeft aan hoe lang het password minstens hetzelfde moet blijven. Hierbij staat . voor 0 weken, / voor één week, a voor 2 weken, A voor 28 weken. De tweede password-regel hiervoor (die voor de gebruiker jc2) bevat hiertoe c/ als karakters. Dat houdt in dat jc2 minstens om de vier weken zijn password moet wijzigen (c=4), en dat dat

wachtwoord dan minstens een week hetzelfde moet blijven.

Het derde veld van de password file-regel bevat het gebruikersnummer. De password file koppelt gebruikersnummers aan gebruikersnamen. Intern is een gebruiker alleen bekend onder een nummer. Steeds vindt bij het afdrukken van de gebruikersnaam een vertaalslag plaats van het nummer naar de naam. Als het nummer nul (0) is, worden aan deze gebruiker systeembeheerdersprivileges verleend. Door in de password file te zoeken naar alle gebruikers met als gebruikersnummer nul, is eenvoudig na te gaan welke gebruikers als systeembeheerder kunnen inloggen. (Dit sluit natuurlijk niet uit dat meer mensen het wachtwoord van een dergelijke gebruiker kennen!)

Het vierde veld is het actuele nummer van de groep waarin de gebruiker is opgenomen. Zo is het mogelijk verschillende groepen te onderscheiden, waarvan de leden met de bestanden van medegroepsleden werken, maar niet met die van andere groepen. Het groepsnummer verwijst naar de group file. In de group file wordt het groepsnummer gekoppeld aan een groepsnaam, en wordt vastgelegd welke gebruikers toestemming hebben om naar een andere groep "over te steken". Op deze wijze kan de gebruiker zijn of haar actuele groepsnummer veranderen. Gebruikers die toegang hebben tot de groepen bin en sys kunnen meer dan andere gebruikers, omdat veel programma's en directories bij oplevering van het systeem als groepseigenaar bin dan wel sys hebben. Als die programma's en directories niet voor groeps-access afgesloten zijn, hebben gewone gebruikers zonder systeembeheerdersprivileges toch toegang tot bestanden die normaal niet voor gebruikers toegankelijk zijn.

Het vijfde veld van de password file is puur informatief. Het is een korte omschrijving van de gebruiker. Hierin kan bijvoorbeeld de gehele naam worden opgenomen, maar ook de lokatie waar deze gebruiker zit.

Het zesde veld geeft aan waar in de filesysteemhiërarchie de bestanden van de gebruiker zich bevinden. Dit is de zogenaamde home directory.

Het zevende en laatste veld geeft aan welk programma bij het inloggen opgestart moet worden. Als dit veld leeg is, zal dit programma de standaard-shell zijn, maar het is ook mogelijk hier een programmanaam op te nemen zodat gebruikers automatisch in het vermelde programma belanden. Zodra het programma beëindigd is, wordt de gebruiker uitgelogd.

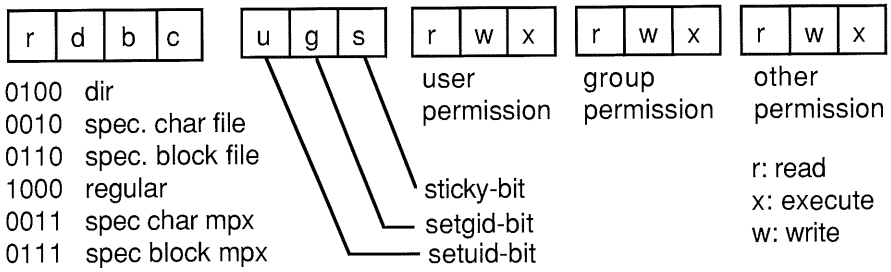
Het is mogelijk in dit zevende veld als shell /bin/rsh op te geven. Dit is de "restricted" versie van de gewone shell. Hiermee is het de gebruiker niet mogelijk:
-- van directory te veranderen (de bewegingsvrijheid wordt zo beperkt);
-- een andere dan de standaardlijst op te geven van directories waarin naar programma's wordt gezocht (zo wordt dus bepaald welke programma's door de gebruiker wel en welke niet te gebruiken zijn);
-- bestandsnamen op te geven die in de directory-hiërarchie niet onder de home directory vallen (hiermee wordt het "zicht" beperkt tot uitsluitend de eigen bestanden);
-- schermuitvoer van programma's naar bestanden weg te schrijven.

Voor shell scripts gelden bovenstaande beperkingen niet. Door aan de gebruiker alleen door de systeembeheerder niet wijzigbare, goedgekeurde scripts ter beschikking te stellen, kan dus "onder toezicht" toch van de volledige faciliteiten van UNIX worden geprofiteerd.
Als programma's faciliteiten bieden om vanuit het programma een tweede shell op te starten, wordt automatisch de restricted shell (rsh) opgestart. Hierdoor kan de gebruiker niet vanuit programma's in de normale shell terechtkomen.

File-beveiliging

Van bestanden worden niet alleen vanzelfsprekende gegevens als de grootte en de aanmaakdatum bijgehouden, maar ook wie de eigenaar is, van welke groep het bestand is en wie op welke wijze aan het bestand mag komen. Voor dit doel wordt een aantal bits bijgehouden.

Deze bits zijn in te delen in drie groepen van drie bits (de rwx-bits), plus een aantal extra bits. De rwx-bits geven aan of de file gelezen (r), geschreven (w) of geexecuteerd (x) mag worden. Merk op dat programma's ook files zijn, net als data bestanden zijn. Deze bits worden bijgehouden voor de eigenaar van de file, voor de medegroepsleden, en voor de



Figuur 6. De zestien bits die de protectie van een bestand regelen. De vier meest linker bits geven het type van het bestand aan. Het sticky bit is opgenomen om performance-redenen, niet als beveiliging.

"rest van de wereld", de gebruikers die niet in dezelfde groep zitten als de eigenaar. Hier een voorbeeld:
rwx-rx-x: De eigenaar mag dit programma (het x-bit staat aan) zowel lezen, overschrijven als executeren. Medegroepsleden mogen alleen lezen en executeren, en overige gebruikers mogen het alleen executeren.
De eigenaar van een file heeft altijd de mogelijkheid om de rwx-bits te veranderen. De rwx-bits voor de eigenaar hebben dus alleen zin als bescherming ten behoeve van de eigenaar zelf.

Als het x-bit aanstaat, gaat de shell ervan uit dat het een programma betreft. Door te kijken naar een speciaal kenmerk ("magic number"), bepaalt de shell of het een lijst van shell-instructies betreft (shell script), of dat het een echt binair programma betreft. Is het eerste het geval, dan zal de shell het operatingsysteem vragen een dochter-shell op te starten die de instructies uit het bestand leest en één voor één interpreteert en uitvoert. Als de shell ontdekt dat het om een binair programma gaat, zal hij de kernel opdracht geven het programma uit te voeren.

Directory-beveiliging
Ook voor directories gelden de rwx-bits, alleen hebben ze hier een bijzondere betekenis.

Het r-bit geeft aan dat de directory kan worden opgevraagd, zoals bijvoorbeeld een telefoonnummer bij 008 kan worden opgevraagd. Als het bit uit staat, wil dat niet zeggen dat bestanden in die directory ontoegankelijk zijn, zoals het ook mogelijk is geheime telefoonnummers te gebruiken, ook al zijn ze niet opvraagbaar bij 008.

Het w-bit geeft aan dat er in de directory mag worden geschreven. Omdat de directory door UNIX wordt gezien als een opsomming van bestanden, wil dat zeggen dat als men toestemming heeft in de directory te schrijven, men dus ook bestanden opgenomen in een directory kan hernoemen, toevoegen of verwijderen. Voor het verwijderen van bestanden heeft men dus geen schrijfprivilege nodig voor de desbetreffende file, maar voor de directory waarin de file is opgenomen!

Ten slotte het x-bit. Dit geeft aan of de directory mag worden opgenomen in het benaderingspad tot een bestand. Als bijvoorbeeld in figuur 1 de gebruiker marie het x-bit voor groepsleden en "others" van de directory /mnt/marie heeft uitgezet, is het voor anderen dan marie niet mogelijk bestanden en directories onder de directory marie te benaderen. Dit is te vergelijken met de mogelijkheid om bij de PTT alle 06-koopnummers uit te laten schakelen. Men kan (als het r-bit aanstaat) wel opvragen welke bestanden in de directory zijn opgenomen (zoals het nog steeds mogelijk is 06-nummers op te zoeken), maar men kan er niet bij. Met het x-bit is het dus mogelijk een hele tak van de directory-boom af te sluiten voor anderen dan de eigenaar en/of de groep.

Device-beveiliging
Devices, zoals printers, terminal-lijnen en tape-stations maar ook de harde schijf en het interne geheugen van de machine, worden door UNIX beschouwd als files. Als men het interne geheugen van de machine wil uitlezen, kan men dit doen door de file /dev/mem uit te lezen.
Omdat het niet gewenst is dat iedereen deze mogelijkheid heeft, gelden voor devices in principe dezelfde beveiligingsmogelijkheden als voor files: het r-bit geeft aan of er van gelezen mag wor-

den, het w-bit geeft aan of er geschreven mag worden. Het x-bit is niet van toepassing.

Executie en het setuid bit
Programma's die door een bepaalde gebruiker worden opgestart, erven de privileges die de gebruiker in kwestie heeft. Zo is het bijvoorbeeld niet mogelijk voor een gebruiker om de password file met een editor te lijf te gaan (als de systeembeheerder de toegang tot /etc/passwd goed heeft afgesloten).

Soms kan het echter wenselijk zijn dat de gebruiker tijdelijk en onder strikte controle van een programma de privileges van een ander krijgt. Neem als voorbeeld weer de password file: Een gebruiker moet zijn wachtwoord kunnen wijzigen, en moet hiervoor dus in de password file kunnen schrijven. Het password-programma krijgt hiervoor tijdelijk niet de privileges van de gebruiker van het programma, maar van de eigenaar van het programma, in dit geval de systeembeheerder. Dit betekent dat het programma toch in de password file mag schrijven.

Om het mechanisme te activeren geeft de eigenaar van het programma aan het besturingssysteem een opdracht die het zogenaamde setuid (set user id) bit aanzet. Bij het opstarten van het programma zal het dan niet de privileges van de gebruiker van het programma krijgen, maar die van de maker van het programma. Bij het uitlijsten van de rwx-bitjes van zo'n programma verschijnt niet een x maar een s op de eerste positie.
Dit mechanisme werkt ook voor een groep. Door het zetten van het setgid (set group id) bit krijgt het programma tijdelijk de privileges van de groep van de eigenaar van het programma.

De extra privileges gelden zolang het programma draait. Zodra het programma beëindigd wordt, zijn ook de privileges van de gebruiker weer normaal. Het is dus niet mogelijk dat een gebruiker door het laten vastlopen van een geprivilegieerd programma voor de rest van de logon-duur extra privileges krijgt.

Het moge duidelijk zijn dat de werking van deze geprivilegieerde programma's goed gecontroleerd moet zijn. Mocht er een fout in een dergelijk programma zitten, dan zou het bijvoorbeeld mogelijk zijn dat een gewone gebruiker plotseling systeembeheerdersprivileges krijgt. Gedurende deze periode zou de gebruiker dan maatregelen kunnen nemen om voortaan op eenvoudige wijze aan deze privileges te komen.
Het setuid bit heeft in het verleden inderdaad tot "gaten" in de beveiliging geleid. Door de lange geschiedenis van UNIX en het loslaten van vele kraakgrage studenten op UNIX zijn vele gaten al gevonden en wordt de kans op gaten steeds geringer.

De Super user
We hebben het eerder in dit artikel al even over de systeembeheerder gehad. In UNIX wordt deze de super user genoemd. De logon-naam van de super user is vaak root, maar een andere naam is ook mogelijk. Het enige wat van belang is, is dat de user id van de super user 0 (nul) is. De namen van alle mensen die super user kunnen zijn, kunnen dus uit de password file gehaald worden door te kijken wie als user id 0 hebben.

Het voornaamste extra privilege van de super user is dat voor hem het file-beveiligingssysteem niet geldt. Met andere woorden: De gebruiker met user id 0 is gerechtigd alle bestanden van wie dan ook, beveiligd of niet, te lezen en te overschrijven.

Daarnaast werkt een aantal programma's alleen als ze door de super user zijn opgestart, doordat in het programma expliciet getest wordt wie het programma heeft opgestart. Hierbij kan worden gedacht aan configuratieprogramma's.
Het moge duidelijk zijn dat degene die super user is op een UNIX-systeem een uitermate machtige positie heeft. Het zal krakers op UNIX-systemen dan ook altijd bekoren om de super user-status te bemachtigen.

Het zoeken naar bestanden met specifieke protectie
Voor de systeembeheerder moet het mogelijk zijn naar bestanden te zoeken die op een bepaalde manier beveiligd zijn. Dat geldt met name voor het zoeken naar specifieke bestanden die niet afdoende beveiligd zijn. In de bibliotheek van standaardprogramma's is het programma *find* opgenomen.
Find zoekt in de directory-hiërarchie vanaf een bepaald punt naar bestanden met bepaalde eigenschappen. Deze eigenschappen kunnen door de gebruiker van *find* met behulp van opties opgegeven worden. De eigenschappen waarnaar kan worden gezocht, zijn onder andere:

-- de naam van het bestand;
-- de protectie van het bestand. Hierbij kan bijvoorbeeld worden opgegeven dat gezocht moet worden naar alle bestanden die zowel het setuid bit aan hebben als dat ze overschrijfbaar zijn voor iedereen;
-- de eigenaar van het bestand (en zo dus naar alle bestanden met het setuid bit aan én met als eigenaar root, de bestanden die draaien met systeembeheerdersprivileges);
-- de grootte van het bestand.

Hieronder een voorbeeld:

find / -perm -4000 -user root -print

drukt de namen af van alle bestanden in het gehele file-systeem die als eigenaar root hebben, en die ook het setuid bit aan hebben staan. Hierbij is 4000 de octale representatie voor het setuid bit in het 16-bits protectiewoord (zie figuur 5).

UNIX en het UUCP-netwerk
Met behulp van UUCP (Unix to Unix CoPy) is het mogelijk op andere systemen dan het eigen systeem te werken en bestanden te benaderen en electronic mail te versturen. Voor de beveiliging van UUCP-communicatie wordt naast het normale UNIX-toegangs- en bestandsbeveiligingssysteem een tweede systeem gebruikt. Een aantal bestanden in de directory /usr/lib/uucp verzorgt deze beveiliging. Bij veel opgestelde UNIX-systemen is aan deze beveiliging te weinig aandacht besteed. Dit is in hackerskringen ook bekend. Bij het hacken van een UNIX-systeem probeert men dan ook vaak via de user id voor UUCP binnen te komen.

Als twee UNIX-systemen via UUCP met elkaar communiceren, zal het actieve systeem (het systeem dat de verbinding initieert) bij het andere systeem op een "normale" lijn inloggen onder de naam UUCP. Hoewel afgedwongen kan worden dat voor deze logon-naam een password benodigd is (zodat niet iedereen kan inloggen onder de naam UUCP), gebeurt dit in veel gevallen niet, omdat vaak veel verschillende systemen op deze user id moeten kunnen inloggen.

De logon shell van de gebruiker UUCP is niet de normale shell, maar het programma UUCICO (Unix to Unix Copy In Copy Out). Dit is ook het programma dat op de initiërende machine draait. Met behulp van een speciaal protocol wisselen de twee programma's nadere identificatiegegevens uit die betrekking hebben op het aangelogde systeem en de gebruiker die de UUCP-opdracht heeft gegeven. Het is mogelijk in de file /usr/lib/USERFILE aan te geven dat een systeem moet worden teruggebeld vóórdat daadwerkelijk data over en weer worden verzonden. Daarnaast kan in de file /usr/lib/uucp/SEQF per bekend systeem worden bepaald dat bij iedere logon een volgnummer dient te worden meegegeven. Het gebelde systeem verifieert dit volgnummer, zodat kan worden gecontroleerd dat het bellende systeem inderdaad degene is die hij beweert te zijn.

Via het UUCICO-protocol worden ook opdrachten uitgewisseld. Deze opdrachten worden door UUCICO uitgevoerd. Hierdoor worden de opdrachten beveiligingstechnisch uitgevoerd door de gebruiker UUCP (de naam waaronder het remote systeem op het lokale systeem is ingelogd). Voor alle files en directories die door UUCICO moeten worden benaderd voor een opdracht geldt dan ook dat ze voor de gebruiker UUCP benaderbaar moeten zijn.

Omdat veel bestanden op een UNIX-systeem vrijelijk door iedereen leesbaar zijn, wordt naast de standaardbeveiliging door UUCICO een tweede controle toegepast. Ook hiervoor maakt UUCICO gebruik van de file /usr/lib/uucp/USER FILE. In deze file is een lijst opgenomen die aangeeft welke delen van de directory-boom door welke gebruiker van welk systeem bij gebruik van UUCP kunnen worden gezien. Op deze wijze is bijvoorbeeld de normaliter vrijelijk leesbare password file niet door UUCP-gebruikers te lezen als in de lijst is opgenomen dat onbekenden alleen de directory /usr/spool/uucp mogen benaderen (de password file staat immers in de directory /etc).

In de file /usr/lib/uucp/L.cmds is een lijst opgenomen welke programma's via een UUCP-verbinding mogen worden opgestart.

4 Beveiligingstips

Tot slot wordt hieronder een aantal beveiligingstips gegeven.

Executeerbare files mogen niet door anderen dan de eigenaar overschrijfbaar

zijn, anders zou het mogelijk zijn in software van anderen wijzigingen aan te brengen. Deze wijzigingen zouden zich met een computervirus zelfs over het hele systeem kunnen verspreiden.

Het is niet voldoende een bestand te beveiligen tegen overschrijven. Als de directory waarin een bestand staat leesbaar is, en de directory is schrijfbaar, is het mogelijk een kopie te maken van het desbetreffende bestand, het originele bestand te verwijderen en het kopiebestand te hernoemen naar de naam van het oorspronkelijke bestand. Hierdoor ontstaat een overschrijfbaar bestand met dezelfde naam als het oorspronkelijke niet-overschrijfbare bestand. Hieronder een voorbeeld:
Het bestand info.text in de directory jan is tegen overschrijven beveiligd, maar directory jan is dit niet.
Kopieer het be-stand info.text naar het bestand info2.text, verwijder info.text, en hernoem info2.text tot info.text. Al deze stappen zijn toegestaan.

Het voorgaande is ook mogelijk indien in het pad naar het bestand toe een directory is opgenomen die overschrijfbaar is. Het is dan immers mogelijk in de desbetreffende directory wijzigingen aan te brengen. Hieronder een voorbeeld:
Stel dat de password file (in de directory /etc) tegen overschrijven is beveiligd en dat de directory /etc ook tegen overschrijven is beveiligd. Als nu de root directory (de top van de directory-boom) niet tegen overschrijven is beveiligd, kan de gehele directory /etc, met alle subdirectories daaronder, vervangen worden door een andere geprepareerde directory.

Maak met het *find*-commando regelmatig een lijst van alle programma's die als eigenaar "root" hebben en ook het setuidbit aan hebben staan. Deze lijst moet zo kort mogelijk zijn; van elk van de programma's op deze lijst moet te verantwoorden zijn waarom ze setuid aan hebben.

Soms wordt gebruik gemaakt van menusystemen om mensen af te schermen van de "normale" shell. De meeste interactieve programma's van UNIX bieden de mogelijkheid om even de shell op te starten. Het gevaar bestaat dat gebruikers op deze wijze onder het moeten gebruiken van het menusysteem uit kunnen.

Het zoekpad voor programma's van de systeembeheerder (root) mag niet '.' (dat wil zeggen de huidige directory) bevatten. Als dit wel het geval is, kan een gebruiker een programma in de eigen directory zetten dat de naam draagt van een veel gebruikt programma uit de standaardprogrammabibliotheek. De systeembeheerder zal dan als hij zich in de directory van de desbetreffende gebruiker bevindt, het programma van de gebruiker opstarten in plaats van het programma uit de standaardbibliotheek. Op deze wijze draait het programma van de gebruiker met systeembeheerdersprivileges!

Daar alle in- en uitvoer-devices en het geheugen als file worden gezien, moet erop worden toegezien dat de beveiliging van deze files goed is opgezet. Anders is het voor gebruikers bijvoorbeeld mogelijk terminal-lijnen van anderen af te tappen (en zo passwords op te vangen) of passwords direct van de harde schijf te lezen, om alle beveiligingen heen. Bij dit laatste moet de gebruiker wel weten hoe een schijf onder UNIX is ingedeeld, maar deze informatie is eenvoudig te verkrijgen.

5 Referenties en literatuurverwijzingen

[MORR78] *Password Security: A case history*, Robbert Morris, Ken Thompson, AT&T Bell Laboratories, 1978.
[UNIX79] *UNIX programmer's manual*, AT&T Bell Laboratories, 1979.
[WINK89] J.C. van Winkel, *The Phenomenon computerviruses reviewed*, Uitgave Nederlands Genootschap voor Informatica, ISBN 90-70621-29-0.

Drs.ing. J.C. van Winkel RI
Was van mei 1984 tot en met februari 1990 werkzaam bij KPMG Klynveld EDP Audit als software engineer. Met ingang van maart 1990 werkt hij bij AT Computing te Nijmegen als UNIX-docent/consultant. Zijn interessegebieden zijn Operating Systems, waaronder UNIX, computervirussen, computerarchitectuur en cryptografie. Hij is lid van de Vereniging van RegisterInformatici (VRI). In 1984 is hij afgestudeerd aan de HIO te Eindhoven en in 1988 aan de Vrije Universiteit Amsterdam met een scriptie over computervirussen.

De AS/400 wordt in het algemeen gebruikt in middelgrote organisaties, waarbij meestal sprake is van een gebrek aan (mogelijkheden tot) functiescheiding binnen de automatiseringsorganisatie. Het beveiligingsmechanisme van de AS/400 biedt echter voldoende faciliteiten om bij een goede implementatie te kunnen steunen op EDP-controles.

Ing. J.F. Kuperus

Aandachtsgebieden bij een AS/400 security audit

1 Inleiding

De in juni 1988 door IBM geïntroduceerde AS/400-computer blijkt een dusdanig verkoopsucces dat EDP-auditors in toenemende mate met deze machine geconfronteerd zullen worden. In eerste instantie geïntroduceerd als vervanging van de S/36- en S/38-computers, blijkt dat de topmodellen ook moeten worden gezien als een reële concurrent van (de IBM 9370 en 4381) mainframes. Dit geldt met name voor gebruikers van een DOS/VSE-besturingssysteem, die produkten zoals het DB2 relationele database management-systeem willen gebruiken. Deze groep staat, wat de IBM-produkten betreft, voor de keuze over te stappen op een MVS-besturingssysteem of te kiezen voor een AS/400. Aan-gezien conversie van VSE naar MVS een kostbare zaak is die jaren kan duren, ligt de keuze van een AS/400 voor de hand.

Veel gebruikers van de logische voorganger van de AS/400 - de S/38-computer - hebben de overstap naar de AS/400 reeds gemaakt of overwegen dit. Daarnaast schaffen ook gebruikers van S/36-computers in toenemende mate een AS/400 aan. Naar schatting heeft reeds tien procent van de zes- tot zevenduizend S/36-gebruikers in Nederland deze overstap gemaakt. Eind 1989 waren wereldwijd reeds meer dan honderdduizend AS/400-systemen geplaatst.

In dit artikel is een beschrijving opgenomen van de AS/400-computer en de mogelijke logische beveiligingen waarmee een betrouwbare gegevensverwerking kan worden bewerkstelligd. Hierbij ligt de nadruk op user profiles en group profiles. Verder zijn de belangrijkste bedreigingen aangegeven en de wijze waarop deze kunnen worden beperkt. Hierbij moet wel worden bedacht dat de AS/400 in het algemeen wordt toegepast in middelgrote bedrijven. In het algemeen is in bedrijven van deze omvang sprake van een minder goede fysieke beveiliging en een gebrek aan functiescheiding binnen de automatiseringsorganisatie. Desondanks is het mogelijk bij een goede implementatie van de door de AS/400 geboden toegangsbeveiliging te steunen op EDP-controles.

De "instapmodellen" van de AS/400 vinden hun toepassing als kantoorcomputer in kleinere organisaties. In dit type organisatie is in het algemeen geen of onvoldoende sprake van functiescheiding en dient men voor het vaststellen van de betrouwbaarheid van de gegevensverwerking terug te vallen op gebruikerscontroles. De in dit artikel beschreven bedreigingen en mogelijke maatregelen hiertegen hebben dan ook betrekking op de grotere AS/400-modellen.

2 Het AS/400-systeem

AS/400-computers worden geleverd in twee typen die echter dezelfde architectuur hebben zodat applicaties onderling uitwisselbaar zijn. De kleinste systeemeenheid - de 9404 - wordt geleverd in de modellen B10 en B20 en is qua capaciteit vergelijkbaar met de kleinere S/36-computers. De grotere systeemeenheid - de 9406 - wordt geleverd in acht modellen. Dit zijn de modellen B30 tot en met B80 en de recent geïntroduceerde modellen B35 en B45. Deze modellen zijn qua capaciteit achtereenvolgens vergelijkbaar met de S/38-, de 9370- en de 4381-modellen van IBM.
De modellen in de 9406-serie zijn in

"racks" gemonteerd en kunnen door het toevoegen van kaarten worden uitgebreid. Ook is het mogelijk een kleiner model uit de 9406-serie uit te breiden tot een groter model door het bijplaatsen van een rack. Voor de overstap van het grootste 9404-model - de B20 - naar het kleinste 9406-model - de B30 - is echter de volledige aanschaf van een model B30 noodzakelijk.
De technische specificaties van deze mo-dellen zijn vermeld in figuur 1.

De systeemeenheid is voorzien van een veiligheidsslot om te voorkomen dat onbevoegden de systeemeenheid aanzetten. Voor de modellen B10 en B20 is er verder een facultatieve voorziening voor de aansluiting op een Battery Po-wer Unit, die het systeem circa tien minuten kan voeden. De modellen B30 en hoger zijn ingericht om te worden aangesloten op een noodstroomvoorziening.

Hardware
De AS/400 is, anders dan bij mainframes het geval is, een volledig geïntegreerd systeem. Dit betekent dat bijvoorbeeld toegangsbeveiligingsprogrammatuur, database management, communicatie-faciliteiten, systeembeheerfuncties en spooling niet als optionele pakketten behoeven te worden geïnstalleerd, maar deel uitmaken van het operating-systeem OS/400.
De gebruiker communiceert met het operating-systeem door middel van commando's - de Control Language - of met behulp van menu's.
Het AS/400-systeem is gebaseerd op drie concepten:

1. *Gelaagde machine-architectuur.*
De AS/400-systeemprogrammatuur is onafhankelijk van de onderliggende hardware-implementatie, waardoor zonder conversie kan worden overgestapt op een nieuwe hardware-technologie. Communicatie met de machine vindt plaats door in utilities en applicaties opgenomen instructies of door gebruik te maken van de Control Language.
2. *Object-georiënteerdheid.*
Binnen een AS/400 is elke eenheid van informatie bekend als een object. De gebruikers zijn hierbij onafhankelijk van de machinestructuur. De AS/400 alloceert ruimte voor een object op een voor de gebruiker onbekende plaats. Aan een object wordt gerefereerd door ingave van naam en bibliotheek. Object-georienteerdheid is van grote betekenis voor het aspect beveiliging, omdat de AS/400 de mogelijkheid biedt om op objectniveau de toegangsbeveiliging te regelen.
3. *Virtuele adresseringstechniek.*
Het AS/400-systeem beschouwt het hoofdgeheugen en het schijfgeheugen als één groot adresseerbaar gebied, daarbij gebruik makend van een apparatuur-onafhankelijke adresseringstechniek. Elk byte in dit totale geheugen heeft een adres en is direct benaderbaar. De gebruiker behoeft alleen de naam op te geven om een object te kunnen benaderen waarbij het systeem verifieert of de gebruiker de gewenste autorisatie bezit. Het is voor een gebruiker niet mogelijk het fysieke adres van een object te bepalen.

Systeem-software
De AS/400 is ontworpen als een interactieve database-machine waarbij het

Figuur 1.

	9404		9406				
	B10	B20	B30	B40	B50	B60	B70
Hoofdgeheugen (Mb):							
. minimum	4	4	4	8	16	32	32
. maximum	16	28	36	40	48	96	96
Schijvencapaciteit (Mb):							
. minimum	630	630	400	400	400	400	400
. maximum	945	945	9600	9600	19200	38400	38400
. aantal schijven/strings	3	3	2	2	4	8	8
Diskettestations	1	1	2	2	2	2	2
Magneetbandstations	1	1	5	5	7	7	7
Lokale werkstations	40	40	120	200	320	480	600
Communicatielijnen	1-8	1-8	2-16	2-32	2-32	2-32	2-48
X.25	6	6	12	12	12	12	12
Token ring adapters	1	1	2	2	2	2	2

operating-systeem (OS/400) zowel batch- als interactieve omgevingen ondersteunt. De belangrijkste functies van het OS/400 zijn:
-- Communicatie-management;
-- Object-management;
-- Data(base)-management;
-- Security management.

Communicatie-management
Met Communicatie-management wordt het geheel van functies bedoeld dat de communicatie en compatibiliteit met andere systemen mogelijk maakt. Communicatie-management is volledig geïntegreerd in OS/400.

Het AS/400-systeem ondersteunt de volgende protocollen en netwerken:
-- Protocollen:
. BSC (Binary Synchronous Communications);
. SDLC (Synchronous Data Link Control);
. TDLC (Twinaxial Data Link Control);
. asynchroon;
. X.25 (Packet Switched);
. Token Ring.
-- Netwerken:
. SNA (System Network Architecture) ;
. X.25,een Packet Switched-netwerk;
. X.21, een publiek toegankelijk gegevensnetwerk;
. Token Ring, een lokaal netwerk;
. APPC/APPN (Advanced Program-to-Program Communications /Advanced Peer-to-Peer Networking). Ondersteund worden de AS/400-implementatie en -extenties van de SNA LU6.2-en PU2.1-architecturen.

Een onderdeel van Communicatie-management is het DDM (Distributed Data Management), waarmee bestanden op een op afstand geplaatst systeem kunnen worden benaderd. Dit systeem dient eveneens DDM te ondersteunen. Daarnaast kan een gebruiker vanuit het ene systeem aanloggen op een ander systeem en gebruik maken van de daar aanwezige programma's en bestanden. Dit is mogelijk door middel van de Display-Station Pass-through. Dit is een onderdeel van de eerder genoemde APPC-functie en maakt het tevens mogelijk een S/36 of een S/38 aan een AS/400 te koppelen.

Object-management
Object-management betreft het geheel van functies waarmee objecten worden opgeslagen en benaderd. Een object wordt beschreven in een object description, waarin de naam, de eigenaar, het type, de grootte en de creatiedatum zijn opgenomen. De eigenaar van een object is de gebruiker die het object heeft aangemaakt. Dit eigenaarschap kan alleen worden overgedragen aan een andere gebruiker door de oorspronkelijke eigenaar of door de security officer (zie later). Bij het benaderen van een object dient een gebruiker geautoriseerd te zijn voor zowel het object als voor de bibliotheek waarin het object is geplaatst.

Data(base)-management
Data-management regelt het verkeer tussen gegevens en applicaties, zorgt ervoor dat de integriteit, zoals vastgelegd in de datadefinities, wordt gewaarborgd en voorkomt concurrent update. Gegevens worden opgeslagen in bestanden (physical files). De wijze waarop een bestand is georganiseerd en de velden van een bestand worden beschreven in bestandsobjecten (file objects). Deze file objects worden geraadpleegd bij het creëren en benaderen van bestanden.

Data-management biedt de mogelijkheid om alle mutaties die op een bestand zijn aangebracht sinds de laatste back-up, in een journal file te bewaren. Bij een beschadiging van een bestand kan, met behulp van de meest recente back-up en de genoemde journal file, dit bestand worden bijgewerkt tot de status van voor de beschadiging.
Daarnaast biedt data-management de mogelijkheid om alle wijzigingen die het gevolg zijn van één transactie, aan te brengen nadat de transactie is voltooid (commitment control). Op deze wijze wordt voorkomen dat, ten gevolge van een foutsituatie, niet alle bij één transactie betrokken bestanden worden bijgewerkt. Indien een gebruiker besluit een transactie ongedaan te maken, zorgt commitment control er eveneens voor dat alle mutaties worden verwijderd (roll-up).

Security management
De AS/400 kent een aantal beveiligingsmechanismen om het gebruik van applicaties, commando's, bestanden en apparatuur te beheren. Onder het hoofdje "Toegangs-beveiliging" wordt op dit onderwerp dieper ingegaan. Hier wordt volstaan met een opsomming van de mogelijkheden:
-- De toegang tot het systeem kan worden beperkt door middel van het ver-

plicht stellen van het invoeren van een gebruikerscode en wachtwoord of door opname van een "job description" voor het draaien van batch jobs.
-- Gebruikers kunnen expliciet worden geautoriseerd om gebruik te mogen maken van bepaalde (invoer/uitvoer) apparatuur.
-- Het gebruik van gegevens, applicaties, commando's en utilities kan worden beperkt door gebruikersbevoegdheden per object vast te leggen.

Naast het kunnen regelen van de (mate van) toegang tot objecten biedt de AS/400 een aantal functies om de continuïteit van de gegevensverwerking te waarborgen en de integriteit van de gegevens te bewaken:

-- *Save- en Restore-commando's.*
Het veilig stellen en inlezen van objecten is mogelijk met behulp van commando's waarvan het gebruik kan worden toegewezen aan bijvoorbeeld operators. Hierbij krijgen operators echter niet de mogelijkheid om objecten te lezen of te wijzigen. In dit verband moet nog de mogelijkheid worden genoemd om tijdens een (onbemande) nachtverwerking een back-up op een gekoppelde schijfeenheid te plaatsen en deze back-up (save file) 's morgens naar een extern medium te kopiëren.

-- *ASP (Auxiliary Storage Pool).*
Een ASP is een groep schijfeenheden waarop een bepaald type object wordt geplaatst. Met behulp van dit mechanisme worden journalen en save files op schijfeenheden geplaatst die fysiek gescheiden zijn van de schijfeenheden waarop de systeemobjecten, bestanden en applicaties zijn geplaatst. Dit is een belangrijk beveiligingsmechanisme, omdat de AS/400 het gehele geheugen als één adresseerbaar gebied ziet en één fysieke schijf in principe delen van systeemobjecten, bestanden en save files zou kunnen bevatten.
Indien deze schijf wordt beschadigd, zal men zowel de originele bestanden als de bijbehorende save en journal files kunnen verliezen. Door het gebruik van ASP gaan ofwel de originele bestanden ofwel de save files en de journaalbestanden verloren. De groep schijfeenheden waarop systeem-objecten, bestanden en applicaties zijn geplaatst, wordt met System-ASP aangeduid. Journaalbestanden en save files worden geplaatst op user-ASP's.

-- *Checksum Protection.*
De informatie op een ASP kan geheel verloren gaan door een beschadiging van één enkele schijf. Om dit op te vangen, zonder het uitvoeren van een tijdrovende recovery, biedt de AS/400 de Checksum Protection-faciliteit. Het principe van Checksum Protection is gebaseerd op het optellen van bits vergelijkbaar met het pariteitsbit op een tape. Van een groep van drie tot maximaal acht gekoppelde en identieke schijfeenheden wordt aldus bij iedere mutatie elke overeenkomstige bitpositie van alle schijven opgeteld en worden berekende pariteitsbits weggeschreven naar een checksum-schijf. Na vervanging van een beschadigde schijfeenheid berekent de AS/400 wat de waarde van elke bit op de vervangende eenheid moet zijn.
Aan Checksum Protection zijn echter extra kosten verbonden. Indien een aantal gekoppelde schijfeenheden wordt opgenomen in een checksum-set, moeten twee extra schijfeenheden worden aangeschaft; één ten behoeve van de opslag van de checksum-informatie en één als reserve-eenheid. Zonder een reserve-eenheid zou Checksum Protection immers weinig zinvol zijn.
Daarnaast heeft Checksum Protection consequenties voor de performance van het systeem. Na iedere mutatie wordt de checksum van de overeenkomstige schijfposities berekend en weggeschreven naar een checksum-eenheid.

Utilities
De AS/400 voorziet in een groot aantal utilities om met het systeem te kunnen werken. Een belangrijke groep van deze utilities zijn de Application Development Tools waarmee applicaties, menu's, gegevens en database-programma's met behulp van menu's kunnen worden onderhouden. Vanuit het oogpunt van de EDP-auditor zijn de SEU en DFU van belang:
-- SEU (Source Entry Utility). Dit is een full screen editor waarmee programmeurs sources kunnen aanmaken en onderhouden;
-- DFU (Data File Utility). Met DFU kunnen database-applicaties worden gecreëerd en onderhouden. Daarnaast biedt DFU de mogelijkheid om rechtstreeks (ongecontroleerd) bestanden te benaderen. DFU geeft de gebruiker volledige toegang tot bestanden waarbij het gevaar aanwezig is dat de informatie in een database inconsistent wordt. Het gebruik van DFU dient niet te worden toegestaan aan eindgebruikers.

3 Toegangsbeveiliging

Het besturingssysteem van de AS/400 is op microcode-niveau geschreven en dit maakt het wijzigen van de systeemprogrammatuur (nagenoeg) onmogelijk. Gesteld kan worden dat toegang tot applicaties en gegevens in een AS/400-omgeving alleen kan worden verkregen volgens regels vastgelegd in de toegangsbeveiligingsstructuur. De EDP-auditor wil uiteraard wel antwoord krijgen op de vraag of door de wijze waarop gebruik is gemaakt van de geboden beveiligingsopties de toegang op een adequate wijze is afgeschermd. Om deze vraag te kunnen beantwoorden is allereerst enige kennis nodig van de werking van de AS/400-beveiligingsmechanismen.

Gebruikersprofielen

In een betrouwbare EDP-omgeving zal iedere individuele gebruiker ten minste moeten beschikken over een gebruikerscode en wachtwoord. Daarnaast mag een gebruiker niet over meer bevoegdheden beschikken dan benodigd zijn voor de uitoefening van zijn functie. Om dit te bewerkstelligen worden de autorisaties van een gebruiker in een AS/400-omgeving vastgelegd in een gebruikersprofiel. De gebruikerscode (naam van een gebruikersprofiel) legt de relatie tussen een gebruiker en de inhoud van het bijbehorende profiel. Een gebruikersprofiel bevat onder andere de volgende informatie:

- naam van het gebruikersprofiel;
- groepsprofiel waartoe een gebruiker behoort (naam, eigenaar en autorisatie);
- wachtwoord;
- publieke autorisatie;
- initieel menu, programma en bibliotheek;
- limited capability-parameter;
- speciale autorisatie;
- gebruikerscategorie;
- specifieke autorisatie (eigenaarschap van en autorisaties tot objecten).

Groepsprofiel. Uit het oogpunt van efficiëntie kan het wenselijk zijn voor bepaalde groepen gebruikers een groepsprofiel te definiëren waarin alle autorisaties worden vastgelegd die voor de leden van deze groep geldig zijn. Een individuele gebruiker heeft alle autorisaties die zijn toegekend aan het groepsprofiel waartoe hij behoort, tenzij een bepaalde autorisatie in het gebruikersprofiel expliciet wordt overschreven. Overigens behoeft een gebruiker niet tot een groepsprofiel te behoren.

Wachtwoord. Wachtwoorden worden encrypted opgeslagen en kunnen door de desbetreffende gebruiker worden gewijzigd. De security officer en de security administrator kunnen wachtwoorden echter overschrijven en op deze wijze gebruik maken van elke gebruikerscode.

Initieel menu, programma en bibliotheek. Het is mogelijk een gebruiker automatisch aan een beginmenu initieel programma of initiële bibliotheek te koppelen.

Limited capability-parameter. In deze parameter wordt vastgelegd of een gebruiker bovengenoemde initiële waarden mag overschrijven. Indien de limited capability-parameter de waarde "yes" heeft, mag de gebruiker de initiële waarden niet overschrijven.
Menu's bevatten een commandoregel die het mogelijk maakt vanuit een menu commando's uit te voeren. Indien de limited capability-parameter de waarde "yes" heeft, wordt een gebruiker tevens verhinderd om AS/400-commando's te gebruiken.
In de velden "speciale autorisatie", "gebruikerscategorie" en " specifieke autorisatie" worden de bevoegdheden van een gebruiker nader vastgelegd. Onderstaand wordt dieper ingegaan op de inhoud van deze velden.

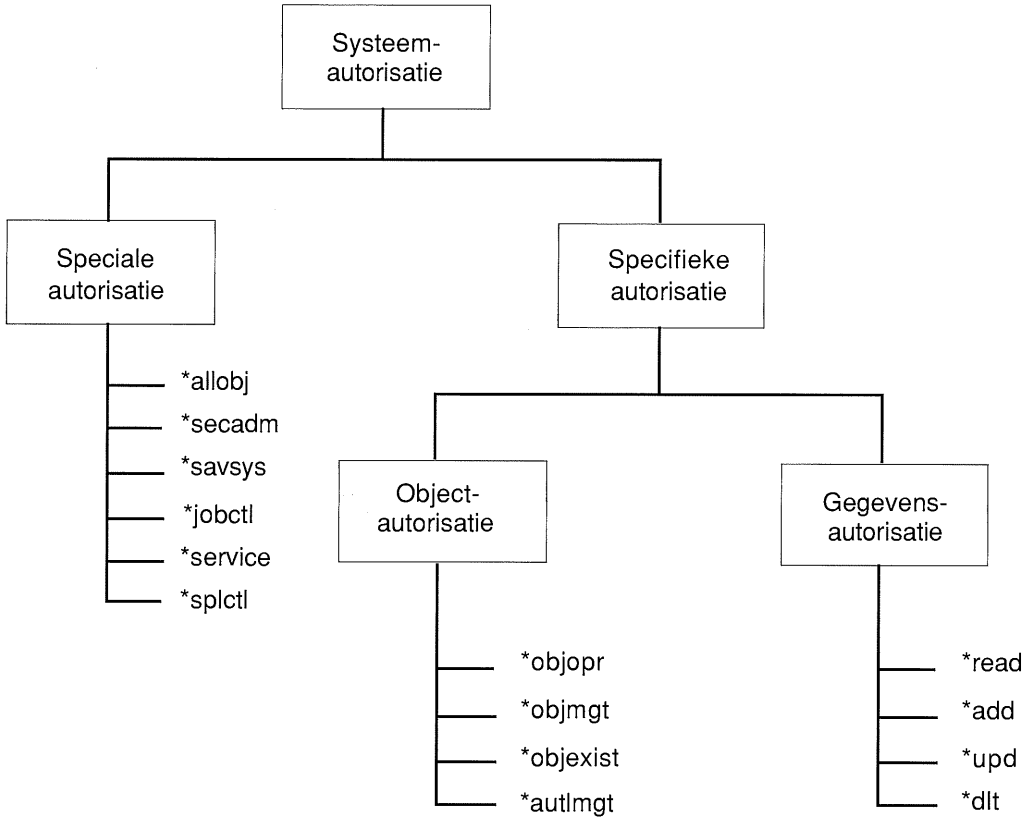
Nadere autorisatie opgenomen in gebruikersprofielen

Bevoegdheden met betrekking tot het gebruik van systeemfuncties en objecten worden in een AS/400-omgeving toegekend door in gebruikersprofielen speciale en specifieke autorisaties op te nemen. De speciale autorisatie die een gebruiker heeft, regelt de bevoegdheden met betrekking tot het gebruik van systeemfuncties. Specifieke autorisatie heeft betrekking op de autorisatie die een gebruiker tot een object heeft.

De wijze waarop autorisaties met elkaar in verband staan wordt met behulp van een schema op pagina 44 verduidelijkt.

Speciale autorisatie

- *allobj (all object authority): onbeperkte toegang tot alle systeemfuncties en tot alle objecten;
- *secadm (security administrator):
  - onderhouden van autorisaties voor documenten en folders;
  - toevoegen en verwijderen van gebruikerscodes;
  - uitgeven en verwijderen van de



Figuur 2.

machtiging om met de autorisaties van een andere gebruiker te werken;
. verwijderen van documenten en folders;
-- *savsys (save system authority): gebruik van save- en restore-commando's;
-- *jobctl (job control authority): mogelijkheid om met invoer- en uitvoer-queues te manipuleren, om printers te starten, subsystemen te stoppen en het systeem te laden;
-- *service (gebruik van service-functies);
-- *splctl (spool control): manipulatie van spool-functies.

Speciale autorisatie heeft geldingskracht voor het gehele systeem en wordt voor iedere gebruiker vastgelegd in een gebruikersprofiel.

Gebruikerscategorie
Het is ook mogelijk een gebruiker in een bepaalde standaard gebruikerscategorie in te delen. Afhankelijk van deze gebruikerscategorie en het gekozen beveiligingsniveau krijgt de gebruiker dan automatisch één of meer speciale autorisaties. De gebruikerscategorie wordt vastgelegd in het gebruikersprofiel. De volgende gebruikerscategorieën worden onderscheiden:
-- *secofr (security officer): voert alle beveiligingsfuncties uit, waaronder het aanstellen van security administrators;
-- *secadm (security administrator): voert beveiligingsfuncties uit, eventueel in een beperkte omgeving;
-- *pgmr (programmer): programmeur;
-- *sysopr (systeem operator): voert systeemoperaties uit, zoals veilig stellen (save) en inlezen (restore). Een systeem operator kan objecten saven en restoren zonder de bevoegdheid te hebben van deze objecten gebruik te maken;
-- *user (gebruiker).

Beveiligingsniveau
Elke individuele gebruiker heeft een persoonlijke gebruikerscode waarmee zijn mogelijkheden zijn vastgelegd. Het gekozen beveiligingsniveau is hierbij van groot belang. Naast het te prefereren beveiligingsniveau 30 kent de AS/400 namelijk de niveaus 10 en 20. Als er voor niveau 10 wordt gekozen, kan ieder met een ad hoc gekozen gebruikerscode aanloggen en is bevoegd alle objecten

te benaderen. Niveau 20 vraagt wel om een vooraf gedefinieerde gebruikerscode en om een wachtwoord, maar ook bij dit niveau kunnen alle objecten worden benaderd, de enige mogelijkheid om gebruikers in hun werkzaamheden te beperken is het koppelen van menu's aan gebruikersprofielen. Op menu's worden dan alleen die mogelijkheden geboden waartoe een gebruiker bevoegd is. Hierbij moet echter worden bedacht dat er altijd de mogelijkheid bestaat dat een gebruiker uit "een menu breekt". Gebeurt dit, dan is het wenselijk dat tenminste de meest kritische gegevens beschermd zijn. Te denken valt aan crediteuren-stambestanden, bestanden met kortinggegevens of met recepturen. Daarom verdient het aanbeveling altijd beveiligingsniveau 30 te kiezen, zodat het mogelijk is van objectbeveiliging gebruik te maken. Opgemerkt wordt dat ook bij beveiligingsniveau 30 gebruikers met de speciale autorisatie *allobj onbeperkt toegang hebben tot alle objecten.

Onderstaand is het verband weergegeven tussen de indeling in gebruikerscategorieën en de daarbij behorende speciale autorisaties. Voor alle duidelijkheid wordt nogmaals opgemerkt dat de hier genoemde bevoegdheden in principe op twee manieren kunnen worden toegekend: via de gebruikerscategorie en via het veld speciale autorisatie.

Gebruikersprofielen kunnen worden gewijzigd door de security officer en door een gebruiker met gebruikerscategorie security administrator en de speciale autorisatie *allobj.

Specifieke autorisatie
Specifieke autorisatie is een combinatie van object- en gegevensautorisatie en bepaalt wat een gebruiker kan doen met een object. Objectautorisatie betreft de handelingen die verricht mogen worden met het totale object (verplaatsen, wissen, hernoemen); gegevensautorisatie geeft aan wat er met de inhoud van een object mag worden gedaan (lezen, toevoegen, verwijderen van records).

Objectautorisatie
- *objopr (object operational):
  de gebruiker mag een object benaderen zoals vastgelegd in de gegevensautorisatie;
- *objmgt (object-management):
  de gebruiker mag de toegangsbeveiliging van het object wijzigen, het object verplaatsen of verwijderen en records toevoegen;
- *objexist (object-existence):
  de gebruiker kan het eigenaarschap van het object overdragen en kan het object verwijderen, saven en restoren;
- *autlmgt (authorization list management):
  geeft een gebruiker de mogelijkheid andere gebruikers toe te voegen aan of te verwijderen uit een autorisatielijst.

Figuur 3.

Beveiligingsniveau 10 en 20

	Speciale autorisatie					
Gebruikerscategorie	*allobj	*secadm	*savsys	*jobctl	*service	*splctl
*secofr	x	x	x	x	x	x
*secadm	x	x	x	x		
*pgmr	x	x	x	x		
*sysopr	x	x	x	x		
*user	x	x	x			

Beveiligingsniveau 30

	Speciale autorisatie					
Gebruikerscategorie	*allobj	*secadm	*savsys	*jobctl	*service	*splctl
*secofr	x	x	x	x	x	x
*secadm		x	x	x		
*pgmr			x	x		
*sysopr			x	x		
*user						

	Objectautorisatie			Gegevensautorisatie			
	opr	mgt	exist	read	add	upd	dlt
*all	x	x	x	x	x	x	
*change	x			x	x	x	x
*use	x			x			
*exclude							

Figuur 4.

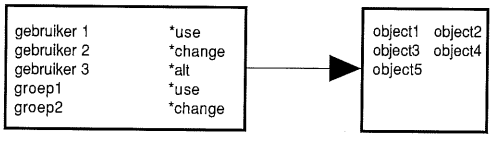
Gegevensautorisatie
-- *read (read):
lezen van gegevens of uitvoeren van een programma;
-- *add (add):
toevoegen van records;
-- *upd (update):
wijzigen van records;
-- *dlt(delete):
verwijderen van records.

De AS/400 kent daarnaast een aantal voorgedefinieerde autorisaties (change, all, etc.) die een samenstel zijn van object- en gegevensautorisatie. Deze voorgedefinieerde autorisaties gelden niet voor de objectautorisatie *autlmgt.
De toegang die een gebruiker tot een object heeft, wordt toegekend door de security officer, een security administrator of door de eigenaar van het object. De gebruiker die het object creëert, is per definitie de eigenaar van een object en heeft hierbij automatisch alle bevoegdheden. Het eigendomsrecht kan door de eigenaar of door de security officer aan een andere gebruiker worden overgedragen.

Autorisatielijst
Autorisatie tot een object kan, zoals reeds vermeld, worden vastgelegd in een gebruikersprofiel en in een groepsprofiel. Daarnaast kan deze autorisatie in een autorisatielijst worden vastgelegd. Een autorisatielijst bevat van een aantal gebruikers en groepen de autorisatie tot de eveneens in de lijst opgenomen objecten. Een voorbeeld is in figuur 5 opgenomen.

Zoekvolgorde
Als een gebruiker toegang wenst tot een object zal het operating-systeem een vaste zoekvolgorde hanteren om de autorisatie te bepalen, waarbij wordt gestopt met zoeken op het moment dat het systeem voor de desbetreffende gebruiker een autorisatie tot het object heeft gevonden.

Figuur 5.



De hierbij aangehouden zoekvolgorde is als volgt:

Gebruikersprofiel:
-- speciale autorisatie * allobj;
-- specifieke autorisatie;
-- autorisatie op de autorisatielijst;

Groepsprofiel:
-- speciale autorisatie * allobj;
-- specifieke autorisatie;
-- autorisatie op de autorisatielijst;

Public (toegekende autorisatie voor alle niet specifiek genoemde gebruikers):
-- publieke autorisatie voor het object;
-- publieke autorisatie voor het object op de autorisatielijst.

4 Aandachtspunten

Bij het beoordelen van een automatiseringsorganisatie zal een EDP-auditor ten eerste vaststellen of de aangebrachte functiescheidingen een betrouwbare geautomatiseerde gegevensverwerking waarborgen. Daarnaast dienen adequate ontwikkelingsmethoden en voorschriften te worden gehanteerd ter waarborging van de kwaliteit van de opgeleverde programmatuur en documentatie. Hierbij is met name de gehanteerde overdrachtsprocedure tussen de test-, de acceptatie- en de produktieomgeving van belang. De indeling in bibliotheken en de daarop aangebrachte objectbeveiliging moet deze scheiding ondersteunen. Ook zal aandacht moeten worden besteed aan continuïteitsaspecten zoals back-up, recovery en uitwijk en de aanwezige fysieke beveiligingen. Specifiek voor de AS/400-computer moet de EDP-auditor daarnaast ten minste aandacht besteden aan onderstaande punten.

Objecten worden meestal per soort in bibliotheken gegroepeerd. Om een object te mogen gebruiken moet men zowel geautoriseerd zijn voor de bibliotheek als voor het desbetreffende object. De EDP-auditor zal zich in eerste instantie moeten vergewissen van de inhoud van de onderscheiden bibliotheken. Per bibliotheek en - in het geval van kritische gegevens en applicaties - per object zal nu moeten worden gecontroleerd welke gebruikers toegang hebben en of deze toegang in overeenstemming is met de taak- en functiebeschrijvingen. Dit is mogelijk door geïmplementeerde objectbeveiligingen uit te lijsten. Hierbij dient men onder het gebruikersprofiel van de security officer te werken, omdat anders niet alle autorisaties zichtbaar worden.

De AS/400 biedt goede toegangsbeveiligingsmogelijkheden, voorop gesteld dat is gekozen voor beveiligingsniveau 30. Bij dit niveau is objectbeveiliging mogelijk, hetgeen bij de niveaus 10 en 20 niet het geval is.

De security officer heeft in feite onbeperkte bevoegdheden, omdat hij zowel de systeem- als de gebruikersbeveiligingen onderhoudt. Daarnaast kunnen vergaande bevoegdheden worden gedelegeerd aan security administrators. Het is dan van belang dat de security officer ten minste elke op het systeem aanwezige bibliotheek kent om de verrichtingen van de security administrator te kunnen bewaken. Om dit te bewerkstelligen moet het "create library"-commando exclusief voor gebruik door de security officer worden gereserveerd.

Het gebruikersprofiel van security officer kan niet worden verwijderd of gekopieerd. Wel is het mogelijk meerdere security officers aan te stellen door van dit gebruikersprofiel een groepsprofiel te maken. Deze in de praktijk uit operationeel oogpunt wel toegepaste methode bemoeilijkt het toezicht op de geïmplementeerde toegangsbeveiliging in hoge mate en moet dan ook als ongewenst worden gezien. Beter is het om een tweede of derde persoon als security administrator aan te wijzen en de gebruikerscode en het wachtwoord van de security officer voor het gebruik in noodgevallen in een verzegelde envelop door het management te laten bewaren.

In de praktijk wordt het gebruikersprofiel van security officer vaak aan een functionaris van de automatiseringsafdeling uitgereikt. Dit zou echter een te grote concentratie van bevoegdheden in de automatiseringsorganisatie betekenen. Het is wellicht beter het wachtwoord van de security officer aan een hooggeplaatste functionaris in de gebruikersorganisatie uit te geven. Binnen de automatiseringsorganisatie kunnen dan meerdere gebruikersprofielen met de bevoegdheid van security administrator worden uitgereikt. Hierbij krijgt elke security administrator niet meer dan de voor de uitoefening van zijn functie benodigde bevoegdheden.

Bij het uitvoeren van een EDP-audit in een AS/400-omgeving is het aan te bevelen alle gebruikers- en groepsprofielen uit te lijsten en deze lijst afhankelijk van de omvang van de organisatie integraal of steekproefgewijs te controleren. Bij het onderwerp "Toegangsbeveiliging" is in dit artikel de betekenis van gebruikerscategorie en speciale autorisaties beschreven. Er moet bijvoorbeeld op worden gelet of een gebruiker met gebruikerscategorie *user niet over de speciale machtiging *allobj beschikt. Een ander punt is de vergelijking van een groepsprofiel met een gebruikersprofiel. Een gebruiker kan ten onrechte zijn ingedeeld bij een groepsprofiel met vergaande bevoegdheden; misschien is in dit geval de door de organisatie gekozen groepsindeling niet correct.
Het is hierbij aan te bevelen het groepsprofiel niet te voorzien van een wachtwoord, zodat gebruikers alleen onder een persoonsgebonden gebruikerscode kunnen aanloggen. Mogelijke ongeautoriseerde activiteiten worden dan per identificerende gebruikerscode in de logging vastgelegd.

Systeem-utilities zoals SEU en DFU bevinden zich in de bibliotheek QIDU, compilers in de bibliotheken QCOBO1, QRPG, QPL1 en QBASIC. Toegang tot dit soort bibliotheken moet expliciet worden geregeld en als publieke autorisatie moet *exclude zijn toegekend.
In de history log worden alle gebeurtenissen vastgelegd die verband houden met het opstarten en sluiten van het systeem, het gebruik van utilities en compilers en ongeoorloofde aanlog-pogingen. De security officer zou deze logging regelmatig moeten controleren. Dit kan gericht geschieden door gebruik te maken van AS/400-commando's of door een CL-programma te schrijven. Interessante zoeksleutels in dit verband zijn

bijvoorbeeld aanlog-pogingen met ongeldig wachtwoord en pogingen om zonder geschikte autorisatie een object te benaderen.

5 Samenvatting

Voor het uitvoeren van een audit van een computersysteem is vaak zeer gedetailleerde technische kennis nodig van alle mogelijke systeemparameters. Men moet welhaast over de kennis van een systeemprogrammeur beschikken om een goede uitspraak te kunnen doen over de betrouwbaarheid van een systeem. Daarbij komt nog het probleem dat zeker de grotere systemen additionele toegangsbeveiligingspakketten kennen die op zich al bijzonder complex van aard kunnen zijn.

De AS/400 is in dit opzicht een stuk eenvoudiger. Alle systeem-software is geïntegreerd en slechts aanwezig op microcode-niveau. De EDP-auditor kan zich voornamelijk beperken tot de vraag of de geïmplementeerde toegangsbeveiliging aansluit op de werkzaamheden van de betrokken functionarissen. Vastgesteld zal moeten worden of toegang tot gegevens slechts mogelijk is door geautoriseerde gebruikers met gebruikmaking van geautoriseerde programmatuur. In dit verband verdienen de objectbeveiligingen en de gebruikersprofielen de meeste aandacht.

De AS/400 is echter een multi-purpose-computer. Dit impliceert dat dit systeem kan worden toegepast als server, als decentraal geplaatste computer of in een netwerk waarin naast andere AS/400-computers ook PC's, S/36, S/38 en mainframes zijn geplaatst. Een en ander maakt duidelijk dat er vele bedreigingen van buitenaf mogelijk zijn en dat de opzet en beveiliging van de datacommunicatie aparte aandacht verdient.

Ing. J.F. Kuperus
Is sinds 1984 werkzaam bij KPMG Klynveld EDP Audit. Hij is betrokken bij het uitvoeren en begeleiden van rekencentra- en systeemonderzoeken in met name de vervoerdersbranche. Hij heeft een brede ervaring opgedaan in het begeleiden van acceptatietests en het opstellen van administratieve procedures bij verschillende ministeries. Zijn interesses liggen met name in de beveiligingsaspecten van kleinere geautomatiseerde omgevingen en hij heeft zich gespecialiseerd in de IBM S/3x en AS/400-lijnen.

**VAX/VMS-systemen worden, al dan niet opgenomen in uitgebreide netwerken, zowel in grote als in kleine organisaties aangetroffen.
Het Virtual Memory System is het besturingssysteem dat op alle VAX-computers, van microVAX tot de VAX 9000, wordt gebruikt. VMS is een operating system dat gelijktijdige uitvoering van multi-user time sharing, batch- en real-time-applicaties toestaat.**

Mw. G.J.C. Heikamp

Beveiligingsaspecten van VAX/VMS-systemen

1 Inleiding

Dit artikel beoogt de EDP-auditor inzicht te geven in de beveiligingsaspecten van het VAX/VMS-besturingssysteem. Na een introductie over apparatuur en systeemprogrammatuur wordt ingegaan op het beveiligingsmechanisme van VMS. Vervolgens wordt ieder onderdeel van dat mechanisme afzonderlijk behandeld. Daarbij komen aandachtspunten naar voren, die bij een beoordeling in beschouwing moeten worden genomen. Tevens wordt kort de beveiliging in een netwerk- en cluster-omgeving aan de orde gesteld.
Het voorlaatste hoofdstuk gaat in op het doel van een VAX/VMS-audit. Als hulpmiddel bij een dergelijk onderzoek kan het pakket "Security Toolkit" goede diensten bewijzen. In het laatste hoofdstuk van dit artikel wordt de functionaliteit van genoemd pakket behandeld.

2 VAX/VMS

VAX/VMS-computersystemen van Digital Equipment Corporation (DEC) worden zowel in het bedrijfsleven als bij de overheid gebruikt. In grote organisaties kan een veelheid van VAXen (en andere computers) worden aangetroffen, onderling verbonden in uitgebreide netwerken. Voor kleine organisaties biedt een microVAX vaak al voldoende capaciteit. De systemen worden voor vele toepassingen gebruikt; hierbij kan worden gedacht aan administratie, produktiebesturing, CAD en wetenschappelijke toepassingen. Momenteel zijn er naar schatting meer dan 6500 applicaties beschikbaar voor VAX-computers. Een bekende toepassing is ALL-IN-1, een geïntegreerd kantoorautomatiserings- en informatiesysteem.

VAX ....

De naam VAX (= Virtual Address eXtension) staat voor een serie "general purpose" computers met een identieke architectuur. Deze is op uiteenlopende manieren in de hardware geïmplementeerd, hetgeen resulteert in een groot aantal computers, van microVAX tot de recent aangekondigde VAX 9000-serie. Met deze laatste serie betreedt DEC de mainframe-markt. Qua prestaties is de VAX 9000 vergelijkbaar met een IBM 3090-180. Afhankelijk van het model kan de verwerking 30 tot 117 maal sneller plaatsvinden dan met een VAX 11/780. De maximale geheugencapaciteit bedraagt 512 Megabyte.

De VAX-familie werd in 1978 geïntroduceerd en is voortgekomen uit de PDP-11. PDP-systemen worden door veel bedrijven ook nu nog gebruikt, met name voor produktiebesturing.
In figuur 1 wordt een overzicht gegeven van de VAX-lijn.

... VMS

Het besturingssysteem VMS (= Virtual Memory System) is een operating system dat gelijktijdige uitvoering van multi-user time sharing, batch- en real time-applicaties toestaat. VMS is in 1986 beoordeeld door het NCSC (National

Figuur 1.

Machine	Geheugencapaciteit	
	minimaal	maximaal
MicroVax II	1 M	16 M
VAX - 11/725	1 M	3 M
VAX - 11/730	512 K	5 M
VAX - 11/750	256 K	8 M
VAX - 11/780	512 K	32 M
VAX - 11/782	1 M	8 M
VAX - 11/785	1 M	36 M
VAX 6xxx	16 M	256 M
VAX 8xxx	16 M	256 M

Com-puter Security Center) en voldeed daarbij aan de criteria nodig voor het verkrijgen van een C2-classificatie. Het Amerikaanse Department of Defence geeft in het zogenaamde "Orange Book" classificaties voor apparatuur en programmatuur. Daarbij worden vier niveaus van beveiliging onderscheiden, A tot en met D, waarbij D het laagste te realiseren beveiligingsniveau aangeeft. De C2-classificatie houdt in dat sprake is van "discretionary protection". Naar VMS vertaald heeft dat betrekking op de aanwezigheid van passwords, de bescherming van bestanden door middel van UIC-protectie (= User Identification Code), de aanvullende bescherming van bestanden met behulp van ACL's (= Access Control List) en de aanwezigheid van audit-faciliteiten. In het vervolg van dit artikel zal daarop verder worden ingegaan.

Voor omgevingen waarin beveiliging van groot belang is, is een update van VMS (VMS/SES = Security Enhancement Service) naar de B-classificatie mogelijk. Deze klasse houdt in dat bescherming moet worden afgedwongen door het systeem. Daartoe is een aantal extra beveiligingsmogelijkheden opgenomen.
Eén van de grootste voordelen van VMS is de flexibiliteit waarmee beveiliging kan worden geïmplementeerd. Deze kan geheel worden afgestemd op de eigen organisatie. Het gevaar is echter niet denkbeeldig dat deze afstemming niet of onvoldoende plaatsvindt. Het is de taak van het management de beveiligingseisen aan te geven. De EDP-auditor kan beoordelen of deze eisen daadwerkelijk in het computersysteem zijn gerealiseerd.

Een bijkomend belangrijk voordeel van VMS is dat het wordt gebruikt op alle VAX-computers, van klein (microVAX) tot groot (VAX 9000). Als een organisatie groeit, kunnen de bestaande applicaties derhalve zonder ingrijpende wijzigingen worden meegenomen naar het grotere VAX-computersysteem.

3 Beveiligingsmechanisme

Algemeen

Beveiliging kan worden gedefinieerd als "maatregelen gericht op de wering van onbevoegden". Het eerste wat een gebruiker merkt van beveiliging is dat hij niet zomaar toegang heeft tot het computersysteem, maar eerst een gebruikersnaam en password moet ingeven. Adequaat password-beheer is noodzakelijk, aangezien men gecontroleerde toegang wenst tot resources (bestanden en programmatuur). De resources vertegenwoordigen namelijk een bepaalde waarde voor de organisatie, waarmee zorgvuldig moet worden omgegaan.

Een operating system heeft als bestuurder van de geautomatiseerde gegevensverwerking ondere andere tot taak deze resources te beveiligen. Daartoe is in het besturingssysteem een mechanisme opgenomen dat de toegang tot het systeem en de daarin opgeslagen gegevens regelt en de organisatie in staat stelt toegangsregels te definiëren en te onderhouden.

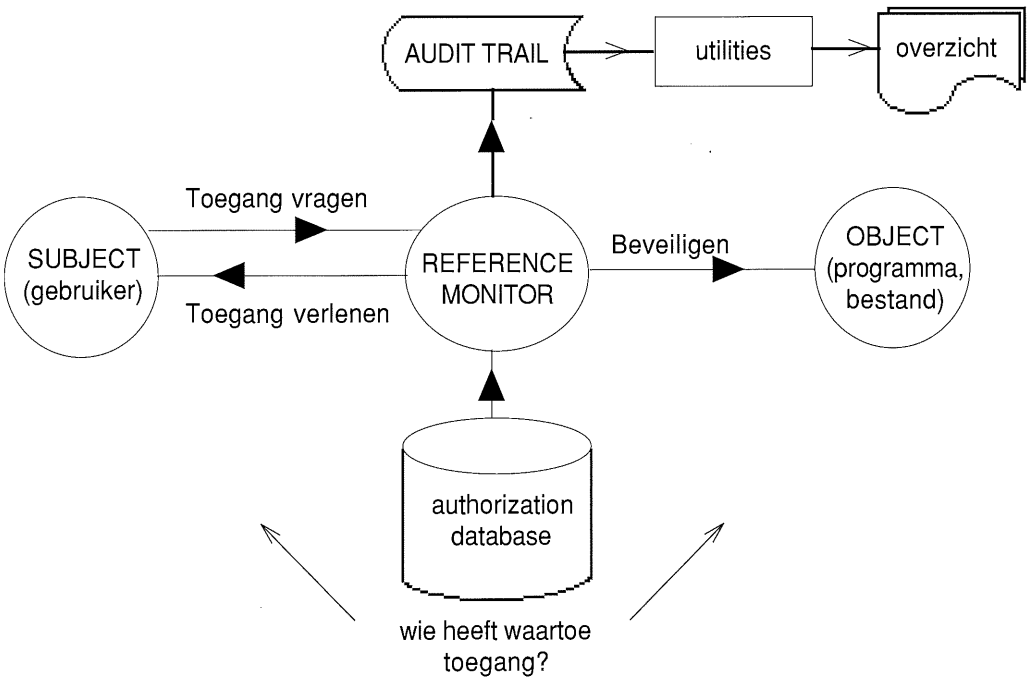
Beveiligingsmechanisme van VMS

Het beveiligingsmechanisme dat in het besturingssysteem VMS is opgenomen, werkt volgens het zogenaamde "reference monitor concept". Dit concept gebruikt voor de beschrijving van een computersysteem de volgende termen: subjects, objects, een "authorization database", een "audit trail" en een "reference monitor mechanism". Deze begrippen zijn in figuur 2 in beeld gebracht en zullen hieronder worden toegelicht.

Een gebruiker kan worden gezien als een subject dat toegang wenst tot een bepaald object. Objecten zijn alle resources (bestanden, programma's, disks, terminals, etc.) in een computersysteem die eventueel bescherming behoeven.

In de "authorization database" staat voor iedere gebruiker aangegeven tot welke objecten hij toegang heeft. Omgekeerd kan voor ieder object worden aangegeven welke gebruikers wel en welke geen toegang hebben. Toegangsrechten op de resources worden daarbij op de volgende wijze onderverdeeld: R(ead), W(rite), E(xecute) en/of D(elete).

De "audit trail" biedt de mogelijkheid iedere gepoogde of daadwerkelijke toegang vast te leggen. Binnen VMS gebeurt dat in een aantal bestanden (bijvoorbeeld de operator log file). Er zijn utilities die het mogelijk maken informatie uit deze bestanden te extraheren.
Het operating system moet afdwingen dat elke aanvraag voor toegang via het "reference monitor mechanism" loopt. Gebruikers met het "bypass" privilege kunnen echter om het beveiligingsmechanisme heen. Het behoeft geen betoog dat met het "bypass" privilege zeer



Figuur 2.

zorgvuldig moet worden omgegaan. Het besturingssysteem dient er, in samenwerking met het "reference monitor mechanism", tevens voor te zorgen dat de "audit trail" en de "authorization database" afdoende beschermd zijn tegen ongeautoriseerde inzage en/of wijzigingen.

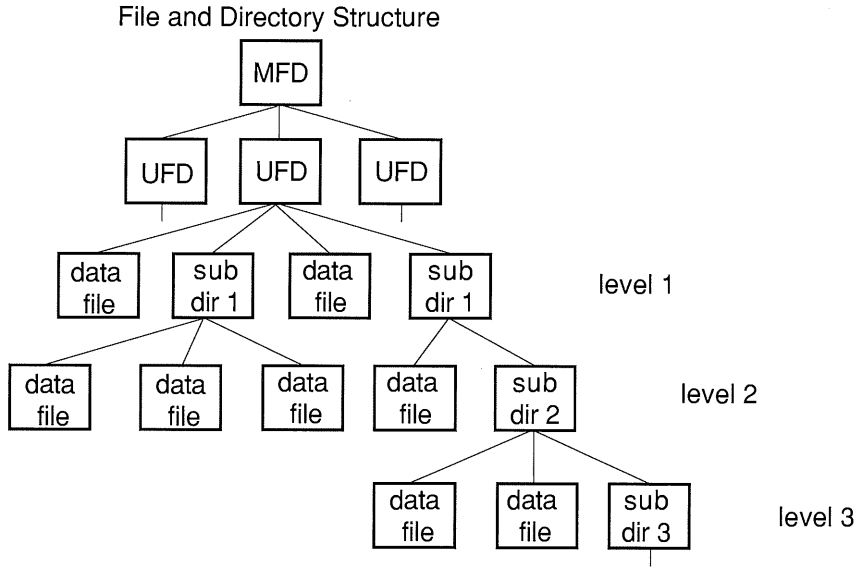
Op de volgende pagina's worden de bovengenoemde onderdelen van het beveiligingsmechanisme nader toegelicht. Allereerst komen de systeeminitialisatie en de "shutdown" van het computersysteem aan de orde. Vervolgens wordt ingegaan op het "file system"; dit kan worden gezien als een ordening van objecten (bestanden). De "authorization database" komt daarna aan de orde. Het volgende onderdeel behandelt de manier waarop de reference monitor bepaalt of een gebruiker (subject) toegang krijgt tot het systeem en uiteindelijk tot een object. Ten slotte wordt ingegaan op de audit trail.

Systeeminitialisatie en shutdown
Het VMS-besturingssysteem wordt opgestart vanaf een console. Op dat moment zijn de logische beveiligingen nog niet actief. Iemand die toegang heeft tot het console kan zichzelf derhalve onbeperkte toegang verschaffen tot het systeem. Het moge duidelijk zijn dat het console te allen tijde in een fysiek beveiligde ruimte dient te staan.

Bij initialisatie wordt door VMS een standaardwaarde aan de systeemparameters toegekend. Deze waarden kunnen door een gebruiker met voldoende privileges (bijvoorbeeld een system manager) worden aangepast aan de voor de organisatie gewenste waarde. Een aantal van deze systeemparameters levert een bijdrage aan het te realiseren beveiligingsniveau (bijvoorbeeld het aantal login-pogingen en de beveiliging van terminals).

Shutdown is op het eerste gezicht vanuit beveiligingsoogpunt niet interessant. Een onverklaarbare shutdown kan echter een aanwijzing zijn dat ongeautoriseerde gebruikers actief zijn. Normaal gesproken kunnen gebeurtenissen met security-implicaties worden vastgelegd. Dit gebeurt echter pas achteraf. Door een shutdown af te dwingen voordat registratie plaatsvindt (bijvoorbeeld door het forceren van een systeemfout), wordt logging en daarmee mogelijke detectie voorkomen.

File system
Het file/directory-systeem van VMS heeft een boomstructuur (zie figuur 3). De top van de boom bestaat uit een disk-volume. Ieder disk-volume heeft een Master File Directory (MFD) die de namen bevat van de User File Directories (UFD's). Iedere gebruiker heeft een eigen UFD. Deze kan be-staan uit bestanden en



Figuur 3. File/directory-systeem van VMS.

directories met een aantal niveaus van subdirectories en daarbinnen weer bestanden. Direc-tories zijn zelf ook bestanden. De inhoud van een bestand kan bestaan uit een source-programma, een object-programma, een executable program, een verzameling gegevens of een tekstdocument.
Op fysiek niveau kan elk bestand uniek worden geïdentificeerd door middel van het zogenaamde file ID. Toegang tot een bestand wordt door het systeem altijd gerealiseerd via het file ID. Gebruikers merken hier weinig van; zij benaderen bestanden via de bestandsnaam.

Het is mogelijk dat in meer dan één directory wordt gerefereerd aan hetzelfde file ID. Dit leidt ertoe dat het desbetreffende bestand "alias" toegangspaden heeft. Er bestaat dan meer dan één naam voor één fysiek bestand. De beveiliging van het bestand kan variëren afhankelijk van het toegangspad. Het is bijvoorbeeld mogelijk dat de primary directory zodanig is beveiligd dat alleen de eigenaar daartoe toegang heeft. De beveiliging van de bestanden in die directory is dan van minder belang, omdat men al op directory-niveau wordt afgewezen. De alias directory kan echter in het geheel niet zijn beveiligd. Op dat moment kan iedereen alle bestanden in de directory benaderen, tenzij de bestanden zelf daartegen afdoende zijn afgeschermd. Ook om andere redenen (met name directe toegang via het file ID, waarbij de directory-structuur wordt omzeild) is het van belang de beveiliging in eerste instantie op bestandsniveau te leggen.
Onzorgvuldige beveiliging van directories maakt het ontstaan van zogenaamde "Worm Holes" mogelijk. Dit houdt in dat een gewone gebruiker (zonder privileges) de inhoud van een bestand van een andere gebruiker (met privileges) kan wijzigen of zelfs eigen bestanden kan plaatsen in een "vreemde" directory. Indien het bestand of de directory toebehoort aan een gebruiker met privileges, kan de "gewone" gebruiker zich derhalve privileges toeëigenen of in ieder geval commando's uit (laten) voeren die zonder privileges niet toegankelijk zijn.

Authorization database
In de "authorization database" zijn van alle gebruikers de bevoegdheden opgenomen in een "user account". In figuur 4 is een voorbeeld van een account afgebeeld; onderstaand wordt een korte toelichting per onderdeel gegeven. Slechts de uit een audit-oogpunt relevante parameters worden besproken.

De user name speelt een rol bij de inlogprocedure. De UIC is bedoeld als een unieke identificatie van een gebruiker. Een gebruiker is altijd lid van een bepaalde groep; de groepen kunnen bijvoorbeeld overeenkomen met de afdelingen van een bedrijf. Het eerste deel van de UIC (300 en ADM) en het account geven de groepsidentificatie weer.

Passwords worden onzichtbaar en "en-

```
UAF>  SHOW JANSEN

Username:       JANSEN                          Owner:      PIET JANSEN
Account:        ADM                             UIC:        [300, 15 ([ADM, JANSEN])
CLI:            DCL                             Tables:     DCLTABLES
Default:        WORK1:[JANSEN]
LGICMD:         LOGIN
Login Flags:    CAPTIVE DISCTLY AUDIT
Primary days:         Mon  Tue  Wed  Thu  Fri
Secondary days:                                 Sat Sun
Primary         000000000011111111112222        Secondary   000000000011111111112222
Day hours       012345678901234567890123        Day hours   012345678901234567890123
Network:        ----------No access ----------              ----------No access ----------
Batch:          ########Full access########                 ########Full access########
Local:          ---------########---------                  ---------########---------
Dialup:         ########---------########                  ########---------########
Remote:         ----------No access ---------               ----------No access ---------
Expiration:                                (none)           Pwdminimum:   6   Login Fails:  0
Pwdlifetime:                             180 00:00          Pwdchange:    1 - APR - 1989  10:30
Last Login:          12 - MAY - 1989  16:04 (interactive),                (none)   (non-interactive)

Authorized Privileges:
  TMPMBX NETMBX
Default Privileges:
  TMPMBX NETMBX
```

Figuur 4.

crypted" opgeslagen binnen de "authorization database". Er kan gebruik worden gemaakt van primary en (als aanvullende beveiliging) secondary passwords binnen één account (user). De system manager kan per gebruiker een minimum-password-lengte bepalen. Bovendien kan de geldigheidsduur van het password worden vastgelegd. Met behulp van de login flags kan een gebruiker worden gedwongen door het systeem gegenereerde passwords te gebruiken.

De meest recente login en het aantal niet succesvolle login-pogingen sinds de laatste wel succesvolle login worden vastgelegd in het user account.
De Command Language Interpreter (CLI) bepaalt welke commandotaal de gebruiker standaard ter beschikking heeft. Normaal gesproken is dit DCL (Digital Command Language). Hiermee kunnen allerlei systeemcommando's, zoals het opvragen van een directory, worden gegeven. Een aantal commando's vereist het bezit van privileges. Indien men een gebruiker aan een menu wil binden en de mogelijkheid om op DCL-niveau te komen wil ontzeggen, kan de gebruiker als CAPTIVE worden gedefinieerd. De commandoprocedure die dit mogelijk maakt, staat vermeld achter LGICMD en wordt bij het inloggen verplicht doorlopen.
Daarnaast kunnen default het device en de directory waarheen een gebruiker tijdens het inloggen wordt geleid (de UFD), worden gedefinieerd. De gebruiker is niet dwingend tot deze directory beperkt. Per login-klasse (bijvoorbeeld local, batch, network, remote en dial-up) kan de toegang worden beperkt. Binnen primary days en secondary days worden twee sets van toegangsuren aangegeven per login class voor de desbetreffende gebruiker (bijvoorbeeld een aantal uren in het weekend en kantooruren door de week).

Authorized privileges bevatten de maximale privileges die de desbetreffende gebruiker tot zijn beschikking heeft. Default privileges zijn de privileges die de gebruiker standaard tot zijn beschikking heeft.

Privileges zijn als volgt te categoriseren:

NONE: geen privileges;
NORMAL: minimale privileges;
GROUP: privileges waarmee invloed kan worden uitgeoefend op leden van de eigen groep;
DEVOUR: staat het gebruik van non-critical system-wide resources toe;
SYSTEM: maakt ingrijpen op de normale systeemactiviteiten mogelijk;
FILES : maakt ingrijpen in de file security mogelijk;
ALL: maakt volledige beheersing van het systeem mogelijk.

Op basis van gegevens in de "authorization database" bepaalt de reference monitor of toegang mag worden verleend. De toegangsbeveiliging is gebaseerd op UIC-protectie en eventueel aanvullend op ACL's. Het principe van deze beveiligingen wordt hieronder besproken. Gebruikers met bepaalde privileges zijn in staat de UIC/ACL-protectie te doorbreken.

UIC-protectie
Voor nieuwe gebruikers van het computersysteem wordt door de system manager een "user account" gedefinieerd in de "authorization database". Iedere gebruiker krijgt daarin een eigen unieke code (UIC). Het formaat is [a,b], waarbij "a" een groepsnummer en "b" een eigenaarsnummer voorstelt. Alle medewerkers van bijvoorbeeld de financiële administratie krijgen hetzelfde groepsnummer. Het eigenaarsnummer daarbinnen is uniek en maakt het mogelijk een bepaalde gebruiker binnen de groep te identificeren.

Als door een gebruiker een object wordt aangemaakt, krijgt dat object de UIC van deze gebruiker. Tevens wordt een bepaald "protection mask" aan het object meegegeven.
Indien nu een willekeurige gebruiker toegang vraagt tot een bepaald object, wordt de UIC van deze gebruiker vergeleken met die van het object. Deze vergelijking resulteert in het indelen van de gebruiker in de volgende categorieën:
system (S): gebruiker in het bezit van alle mogelijke privileges en daardoor in staat het systeem volledig te beheersen;
owner (O): eigenaar van een object;
group (G): gebruiker die tot dezelfde groep behoort als de eigenaar;
world (W): alle gebruikers.

In volgend voorbeeld wordt uitgelegd hoe deze categoriebepaling plaatsvindt. Figuur 5 laat een aantal voorbeelden zien van UIC's van subjecten (gebruikers) en objecten (bestanden). Een vergelijking resulteert in de categorie Owner als het groepsnummer én het eigenaarsnummer van de UIC's gelijk zijn. Group ontstaat als bij vergelijking het groepsnummer van subject-UIC en object-UIC gelijk blijkt te zijn. World resulteert als de hiervoor genoemde vergelijkingen niet opgaan. Daarop is, zoals blijkt, een uitzondering: [1,4] is binnen VMS vrijwel altijd als System gedefinieerd; dit houdt over het algemeen een volledige toegang tot alle objecten in.

Afhankelijk van de categorie waarin de gebruiker is ingedeeld, zijn bepaalde toegangsrechten van kracht: read (R), write (W), execute (E) en delete (D). Deze zijn vastgelegd in het eerder genoemde "protection mask". Dit kan er bijvoorbeeld als volgt uitzien: System:RWED, Owner:RWED, Group:RE, World:R.

ACL-protectie
ACL-protectie biedt de mogelijkheid per object aan een met name genoemde gebruiker expliciet toegang te verlenen of te ontzeggen. Deze vorm van beveiliging is niet verplicht en zal alleen worden gebruikt indien het beveiligingsniveau dat met behulp van UIC's kan worden bereikt, onvoldoende is. Dit is vooral het geval bij bestanden met gevoelige informatie (bijvoorbeeld betalingsbestanden). Het toepassen van ACL's op alle bestanden is niet aan te raden, aangezien de performance wordt vertraagd.
Ten behoeve van de ACL-protectie wordt in de "authorization database" vastgelegd welke gebruikers van het systeem kunnen beschikken over zogenaamde identifiers. Een identifier kan zijn een UIC (unieke identificatie van een individuele gebruiker), een bepaalde loginklasse (alle gebruikers met "Network access") of een bepaalde groep (operators). In de Access Control List van een bestand worden de desbetreffende iden-

Figuur 5.

Object / Subject	[1, 4]	[100, 100]	[100, 120]	[200, 200]	[200, 240]	[200, 260]
[1, 4]	**O**	S	S	S	S	S
[100, 100]	W	**O**	G	W	W	W
[100, 120]	W	G	**O**	W	W	W
[200, 200]	W	W	W	**O**	G	G
[200, 240]	W	W	W	G	**O**	G
[200, 260]	W	W	W	G	G	**O**

S = System **O** = Owner **G** = Group **W** = World

tifiers genoemd. Per identifier kunnen expliciet de toegangsrechten (of het ontbreken daarvan) worden opgesomd.
In een ACL kan een zogenaamd "security alarm" worden opgenomen. Hiermee wordt bereikt dat iedere keer als een bepaald soort toegang tot het object heeft plaatsgevonden, dit wordt gemeld aan de "security operator".

Toegangsbeveiliging
Het doel van toegangsbeveiliging is te voorkomen dat niet-geautoriseerde gebruikers toegang krijgen tot objecten binnen het geautomatiseerde systeem. Toegangsbeveiliging kan in de context van dit artikel worden gedefinieerd als het beveiligen van objecten tegen toegang door onbevoegden.
Zoals reeds is uiteengezet, neemt de reference monitor, op basis van gegevens uit de "authorization database", de beslissing of een gebruiker toegang krijgt. Teneinde deze beslissing te kunnen nemen, wordt een aantal stappen doorlopen:

1. Is de gebruiker bekend binnen het systeem?
De gebruiker geeft bij het "inloggen" een user name op. Deze moet in de "authorization database" vóórkomen; indien dit niet het geval is wordt toegang tot het systeem geweigerd.

2. Is er sprake van een legitieme gebruiker?
De gebruiker moet zich door middel van een password legitimeren. De legitimatie (= authenticatie) is akkoord, indien het ingetypte password overeenkomt met het binnen VMS ("encrypted") opgeslagen password.
Bovenstaande stappen moeten met succes zijn doorlopen, alvorens een gebruiker daadwerkelijk toegang krijgt tot het computersysteem. Het uiteindelijke doel van de gebruiker is echter toegang te verkrijgen tot een object (bijvoorbeeld een programma) binnen het systeem.
Binnen VMS bestaan, zoals wij hebben gezien, twee soorten objectbeveiliging: ACL(Access Control List)-beveiliging en een op de UIC (User Identification Code) gebaseerde beveiliging. De volgende stappen bepalen of de gebruiker toegang krijgt:

3. Is voor het object een ACL gedefinieerd?
Er zijn vier mogelijkheden:
a. Er is geen ACL gedefinieerd.
De gebruiker krijgt dan op dit moment nog geen toegang; de UIC-protectie moet bepalen of toegang kan worden verleend.
b. Er is een ACL gedefinieerd en deze geeft expliciet aan dat de desbetreffende gebruiker toegangsrecht heeft.
De gebruiker krijgt op dit moment toegang tot het object, waarbij de toegangsrechten gelden zoals deze in de ACL zijn opgenomen.
c. Er is een ACL gedefinieerd en deze geeft expliciet aan dat de desbetreffende gebruiker geen toegangsrecht heeft.
De toegang tot het object wordt geweigerd.
d. Er is een ACL gedefinieerd, maar de gebruiker komt hierin niet voor.
Er wordt dan overgegaan naar de volgende stap.

4 Kan toegang worden verleend op basis van de identificatiecode van de gebruiker (UIC)?
Allereerst wordt de categorie van de gebruiker bepaald. Vervolgens wordt gekeken naar het "protection mask" dat bij die categorie hoort. Op basis hiervan wordt de gevraagde (soort) toegang gehonoreerd of geweigerd.
Voorbeeld:
De vergelijking tussen de UIC van de gebruiker en van het object (bestand) resulteert in indeling van de gebruiker in de categorie "group". Het "protection mask" dat voor deze categorie geldt is "G:RWE". De gebruiker krijgt toegangsrechten voor "read", "write" en "execute". Indien het gevraagde toegangsrecht echter "delete" (D) is, wordt dit geweigerd.

5. Is de gebruiker in het bezit van privileges die voldoende krachtig zijn om de UIC/ACL-protectie te doorbreken?
Bepaalde gebruikers, zoals system managers en security officers, hebben voor het uitvoeren van hun taak bijzondere privileges nodig (bijvoorbeeld "readall"). De UIC/ACL-protectie wordt, afhankelijk van het privilege, geheel of gedeeltelijk uitgeschakeld.

Audit trail
Het beveiligingsmechanisme van VMS kan door een organisatie op een bepaalde manier worden ingevuld. Daartoe stelt men eisen waaraan de toegangsbeveiliging tot het systeem en de daarin opgeslagen bestanden moet voldoen. Vastgesteld moet worden of het systeem volgens deze eisen heeft gewerkt. De audit trail legt allerlei gebeurtenissen met security-implicaties vast. Achteraf

dient deze audit trail door een daartoe aangewezen functionaris te worden beoordeeld op basis van de vooraf gestelde eisen. Afwijkingen moeten worden onderzocht en eventueel moeten adequate maatregelen die herhaling voorkomen, worden getroffen.

Hoewel VMS niet over een echte audit trail beschikt, is er een aantal bestanden dat als zodanig kan worden gebruikt.
"Security auditing" kan onder andere op de volgende gebeurtenissen betrekking hebben:
-- het vastleggen van iedere toegangs-(poging) tot een voor de organisatie kritisch bestand;
-- wijzigingen in de "authorization database";
-- break-in-pogingen;
-- installatie-operaties;
-- login-failures;
-- logins per klasse (local, dial-up, etc.);
-- log-outs.

De mogelijkheid bestaat informatie te verkrijgen over het gebruik van system resources. Het regelmatig doorlopen hiervan kan informatie opleveren over ongewone user names, vreemde patronen (bijvoorbeeld een activiteit die alleen door de week plaats hoort te vinden, geschiedt in het weekend), het gebruik van een abnormale hoeveelheid resources, ongewone login-bronnen (bepaalde systemen in het netwerk of terminals) en login-failures.
Het risico is aanwezig dat de audit trail-informatie verloren gaat of wordt gewijzigd. Voor het manipuleren van de audit trail zijn echter privileges noodzakelijk.

Netwerk- en cluster-beveiliging

Netwerk
Het beveiligingsmechanisme van VMS, zoals hiervoor beschreven, werkt in een netwerkomgeving in principe hetzelfde als op een "standalone" VAX. In de praktijk is netwerkbeveiliging echter moeilijker te bewerkstelligen aangezien elk systeem binnen het netwerk een eigen implementatie van de "reference monitor" heeft.
Bij een toegangspoging via het netwerk zijn twee systemen betrokken: systeem A ("source") bevat het subject dat toegang wenst tot een object op systeem B (target). In de "authorization database" van systeem A staat de gebruiker gedefinieerd met zijn bevoegdheden; het object is echter onbekend. Bij systeem B is het object op een bepaalde manier beveiligd; de gebruiker is echter niet bekend in de "authorization database" van systeem B. Systemen A en B moeten nu zodanig samenwerken dat de autorisatie verloopt alsof er sprake is van één systeem met één "authorization database". Om dit te bewerkstelligen wordt over het algemeen gebruik gemaakt van de volgende opties: een standaardnetwerk-account en/of zogenaamde proxy logins.

Het standaardnetwerk-account is een account in de "authorization database" van het target-systeem; over het algemeen is de naam van het account DECNET. Iedere gebruiker van een ander systeem kan toegang krijgen tot het target-systeem via dit account. De bevoegdheden van het account moeten derhalve zeer nauwkeurig zijn gedefinieerd.

Een proxy login stelt gebruikers in staat bestanden te benaderen via een netwerk zonder een gebruikersnaam en password over de lijn te sturen. De autorisatie vindt namelijk plaats op het source-systeem. Op het target-systeem wordt in de "authorization database" de relatie tussen het source user account en het proxy login account (dat op het target-systeem staat) vastgelegd. Een gebruiker op systeem A kan nu gegevens benaderen op systeem B, waartoe zijn proxy account toegang heeft.

Bij de beoordeling van een VAX-computersysteem is het van belang te onderzoeken welke gebruikers toegang hebben tot andere systemen. Nog belangrijker is echter de vraag of gebruikers van andere systemen het desbetreffende VAX-systeem binnen kunnen komen. Indien een VAX immers verbindingen heeft met de buitenwereld, is iedere wereldburger een potentiële indringer. Het vergt een evenwichtig en sluitend stelsel van maatregelen om te voorkomen dat ongeautoriseerde gebruikers toegang krijgen door zich voor te doen als legale gebruiker. Daarnaast dient de mogelijkheid dat gegevens tijdens het transport worden onderschept, zoveel mogelijk te worden voorkomen. Hiertoe kan encryptie in de netwerkprodukten worden toegepast.

Cluster
Een VAX-cluster is een geïntegreerde organisatie van VAX/VMS-systemen met als doel het "sharen" van resources. Een cluster is méér dan een netwerk, maar

niet zo geïntegreerd als een multiprocessor-systeem zoals de VAX 8820. Er zijn twee soorten clusters: een homogeen en een heterogeen cluster. In een heterogeen cluster opereert elk systeem als een afzonderlijk systeem met een eigen reference-monitor; het cluster is net zo veilig als de minst beveiligde VAX binnen het cluster. Een homogeen cluster kent slechts één "authorization database" voor alle systemen en is derhalve qua beveiliging te vergelijken met een "standalone" VAX.

Conclusie
Het feit dat VMS voor authenticatie gebruik maakt van passwords, maakt adequaat password-beheer noodzakelijk. Indien dit niet gebeurt, kan de beveiliging doorbroken worden, aangezien onbevoegden dan de gedaante van een legale gebruiker kunnen aannemen.

VMS heeft verscheidene beveiligingsmogelijkheden. Het beveiligingsmechanisme biedt daarvoor de infrastructuur; de invulling ervan is echter afhankelijk van de organisatie. Gebruikers kunnen als subject minimale bevoegdheden krijgen, of uitgebreide privileges. Het criterium daarbij is dat zij niet méér bevoegdheden binnen het systeem krijgen dan strikt noodzakelijk voor de uitoefening van hun taak. Het zal echter duidelijk zijn dat een oordeel daarover niet los kan worden gezien van de organisatie waar de gebruiker werkzaam is. Voor de beveiliging van objecten (bestanden) geldt iets dergelijks: allemaal, een deel ervan of geen van alle kunnen zij worden beveiligd; ook hier weer afhankelijk van de organisatie. De relaties tussen subject en object, resulterend in bevoegdheden, zijn vastgelegd in de "authorization database" en moeten een afspiegeling zijn van de functies binnen het bedrijf.
Ten slotte kan ook de audit trail, als onderdeel van het beveiligingsmechanisme, worden afgestemd op de beveiligingsinformatiebehoefte van de organisatie.

Aangezien beveiliging niet door VMS wordt afgedwongen, is het van belang regelmatig te controleren of de mate van beveiliging overeenkomt met de eisen die de organisatie hieraan stelt. Het is de taak van de EDP-auditor te beoordelen in hoeverre aan deze eisen wordt voldaan. Hierop wordt in het nu volgende nader ingegaan.

4 Audit

Automatisering leidt ertoe dat taken die voorheen door gebruikers zelf werden uitgevoerd, worden overgenomen door programmatuur. Uit deskundigheids- en efficiency-overwegingen wordt bij verdergaande automatisering het bestaan van een afdeling die de geautomatiseerde gegevensverwerking in goede banen leidt, onontbeerlijk. In veel organisaties zal tevens een afdeling ontstaan die zorg draagt voor het ontwikkelen van nieuwe programmatuur en het zo nodig aanpassen van reeds bestaande programma's.

Gebruikers delegeren taken aan de automatiseringsorganisatie, maar blijven verantwoordelijk voor de betrouwbare uitvoering daarvan. Dit is alleen mogelijk indien:
-- slechts geautoriseerde gebruikers toegang hebben tot het systeem;
-- gebruikers alleen toegang hebben tot die bestanden en programmatuur waarvoor zij zijn geautoriseerd;
-- gegevens alleen worden verwerkt met door gebruikers geaccepteerde programmatuur.

Teneinde het bovenstaande te kunnen waarborgen, moeten bepaalde eisen worden gesteld aan de programmatuur en de (automatiserings)organisatie. Vanuit de systeemprogrammatuur gezien kan de verwerking op twee manieren tot onbetrouwbare resultaten leiden:
1. het besturingssysteem kan worden beschouwd als een normaal programma dat aan bepaalde eisen moet voldoen, bijvoorbeeld scheiding van processen. Deze eisen zijn onafhankelijk van de omgeving waarin het systeem gaat draaien. Indien een fout in het besturingssysteem is geslopen (bug), kan dit leiden tot een onbetrouwbare geautomatiseerde gegevensverwerking;
2. het operating system draait in een bepaalde omgeving. Het is mogelijk het operating system in zekere mate aan te passen aan de eigen specifieke organisatie (systeemparameters, initialisatie, wijzigen van default-beveiligingen). Dit leidt tot een bepaald beveiligingsniveau. Indien dit niveau, gezien de organisatie, onvoldoende is, kan er sprake zijn van een geautomatiseerde gegevensverwerking die niet voldoet aan de hierboven gestelde eisen.

Binnen de organisatie kunnen onbekwaam personeel, slechte hulpmiddelen

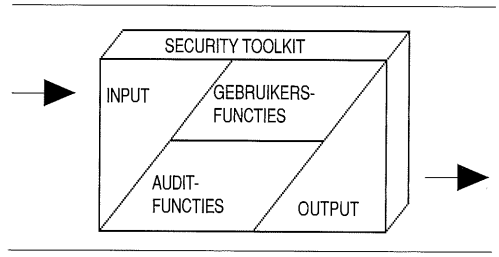
(beschadigde disks), afwezige of onvoldoende nageleefde procedures (gebruikers die passwords opschrijven bijvoorbeeld), een niet optimale omgeving (veel magnetisme), etc. eveneens leiden tot onbetrouwbare gegevensverwerking.
De automatiseringsorganisatie heeft zodanige bevoegdheden gedelegeerd gekregen, dat met haar functioneren de juiste, volledige en tijdige geautomatiseerde gegevensverwerking staat of valt. De bevoegdheden houden, voor wat betreft de verwerkingsorganisatie (= produktie), in principe volledige toegang tot het systeem in. Binnen deze afdeling moet derhalve zoveel mogelijk functiescheiding worden doorgevoerd. Binnen systeemontwikkeling wordt de basis gelegd voor betrouwbare applicatieprogrammatuur. Aangezien dit geen toegang tot het operationele systeem vergt, dient deze toegang ook daadwerkelijk onmogelijk te zijn.

In het voorgaande is een aantal omgevingen te onderscheiden: de omgeving van de gebruiker, van de verwerkingsorganisatie en van de systeemontwikkeling. In produktie draait programmatuur ten behoeve van en geaccepteerd door de gebruiker; deze mag niet ongeautoriseerd worden gewijzigd, noch door produktie, noch door systeemontwikkeling. Bovendien mogen het systeem en de daarin opgeslagen bestanden en programma's slechts door geautoriseerde gebruikers worden benaderd. Dit kan worden gerealiseerd door het implementeren van een adequate logische toegangsbeveiliging, zoals in dit artikel is behandeld. Door de EDP-auditor kan met behulp van kennis over het beveiligingsmechanisme, de toereikendheid van een implementatie worden beoordeeld. De relatie met de (omvang van de) organisatie mag daarbij niet uit het oog worden verloren.
Als hulpmiddel voor de beoordeling van een VMS-systeem kan gebruik worden gemaakt van de nu te behandelen Toolkit.

5 The Security Toolkit: een audithulpmiddel

Inleiding

Naarmate de automatisering voortschrijdt, wordt beveiliging steeds belangrijker. Het vaststellen van de mate van beveiliging van een computersysteem is echter niet zo eenvoudig. Het vergt over het algemeen een vrij grote kennis en er-



Figuur 6.

varing, alsmede een behoorlijke hoeveelheid tijd. Om beveiligingspersoneel met name op dat laatste punt tegemoet te komen, is door Cubic Systems de "Security Toolkit for VAX/VMS" ontwikkeld. Recent is een geheel vernieuwde versie op de markt gebracht (versie 2.0).
Allereerst zal hier de functionaliteit van de Toolkit (gebruikersfuncties en auditfuncties) aan de orde komen. Daarna worden de relaties met de buitenwereld besproken. In het begintraject (input) zijn dit commando's die door de gebruiker rechtstreeks of via een menu worden ingegeven. Het eindtraject (output) bevat de uitvoerresultaten van de Toolkit; deze kunnen eventueel weer als invoer dienen voor andere applicaties. In figuur 6 vindt u dit terug.

Gebruikersfuncties

De term "gebruikersfunctie" is wellicht wat verwarrend: onder gebruikers verstaat de Toolkit een beveiligingsfunctionaris en niet de "gewone" gebruiker van een computersysteem. De functies hebben betrekking op de gebruikers-interface. Hierbij kan men bijvoorbeeld denken aan: on-line help, het opvragen van het Toolkit-menu en het afdrukken van een bestand op het scherm of de printer.

Audit-functies

De audit-functies hebben betrekking op alle gebieden binnen het VMS-operating system die met beveiliging te maken hebben. De desbetreffende gebieden zijn:
-- toegangscontrole van het subject (de gebruiker) tot het systeem;
-- toegangscontrole van het subject tot objecten (bestanden);
-- "authorization database";
-- beveiliging van objecten (bestanden, terminals, disks, e.d.) alsmede systeemparameters die betrekking hebben op beveiliging (bijvoorbeeld het maximaal aantal toegestane login-pogingen);
-- netwerkbeveiliging.
Het zal u niet verwonderen dat vele termen die reeds in het eerste gedeelte van

dit artikel zijn genoemd, hier terugkeren. Eigenlijk doet de Toolkit niets anders dan het in kaart brengen van het op een bepaalde VAX geïmplementeerde beveiligingsmechanisme van VMS.
In het hiernavolgende zullen alle gebieden kort worden besproken. Allereerst komt echter het zogenaamde "summary report" aan de orde.

Summary Reporting
Aan het begin van een audit dient eerst een algemene indruk te worden verkregen van het te onderzoeken systeem. Daartoe wordt algemene informatie opgevraagd, zoals het aantal gebruikers, het aantal bestanden, de systeemnaam en de versie van VMS die wordt gebruikt. Het summary report bevat deze en andere gegevens en geeft bovendien een samenvattend overzicht van de mate van beveiliging van het systeem. Deze samenvatting heeft betrekking op alle hierna te noemen gebieden, zoals de toegangsbeveiliging en de netwerkbeveiliging. Op basis van de samenvatting is het reeds mogelijk eventuele leemten in de beveiliging aan te wijzen. Deze moeten dan verder worden onderzocht.

Access Control
Ter beoordeling van de toegangscontrole tot het systeem kan men onder andere de volgende gegevens opvragen:
-- gebruikersnaam = password;
-- minimum password-lengte;
-- verlopen passwords;
-- levensduur van passwords;
-- mislukte login-pogingen.

Uiteindelijk gaat het om de beveiliging van gegevens die in bestanden zijn opgeslagen. Deze beveiliging kan men onderzoeken door een lijst op te vragen van gebruikers die toegang hebben tot bepaalde (kritische) bestanden.

Authorization Database Reporting
De "authorization database" speelt, zoals we gezien hebben, een belangrijke rol in de toegangsbeveiliging. In deze database staat een beschrijving van alle gebruikers van het computersysteem. Deze beschrijving omvat onder andere de naam van de gebruiker, zijn/haar UIC (user identification code), de eventuele privileges, de geldigheidsduur van het password, etc. Op grond van al deze gegevens bepaalt de reference monitor uiteindelijk of een gebruiker wel of niet wordt toegelaten tot een bepaald object (bestand).

Met behulp van de Toolkit kunnen alle toegangsbevoegdheden van gebruikers in beeld worden gebracht. Daartoe kan bijvoorbeeld een overzicht worden opgevraagd van alle gebruikers die in het bezit zijn van privileges.

Security Reporting
De objecten binnen VMS kunnen worden beveiligd met behulp van UIC- en ACL-protectie. Met behulp van de Toolkit is het mogelijk de beveiliging van objecten als bestanden, terminals en disks te onderzoeken.
Tevens kan men een overzicht krijgen van de waarde van de systeemparameters die vanuit een beveiligingsoogpunt van belang zijn.

Network Reporting
Veel VAX-computers staan opgesteld in een netwerk. In een netwerkomgeving is het van belang te onderzoeken welke gebruikers toegang hebben tot het netwerk. Zo mogelijk nog belangrijker is het te weten, welke gebruikers van andere computers via het netwerk toegang hebben tot de in onderzoek zijnde computer. Het onderzoeken van een netwerk heeft een dimensie meer dan het onderzoeken van een losstaand systeem. Zwakheden die daar wellicht aanvaardbaar zijn, kunnen in een netwerkomgeving verder woekeren en tot belangrijke lekken in de beveiliging leiden.

Input

Menu's en Commando's
Een gebruiker kan de commando's die de Security Toolkit moet uitvoeren via een menu ingeven, direct intoetsen of opnemen in een zogenaamde commandoprocedure.
In het hoofdmenu worden de volgende mogelijkheden geboden:
-- system access reporting;
-- object access reporting;
-- network reporting;
-- security summary reporting;
-- security reporting;
-- identifier reporting.

De Toolkit vertaalt de gekozen menu-opties automatisch in Toolkit commando's. De verwerking kan interactief of batchgewijs geschieden. Tevens is het mogelijk het via het menu opgebouwde commando toe te voegen aan een bestand ter latere uitvoering.
Het gebruik van commandoprocedures kan de EDP-auditor die de werking van

beveiligingsmaatregelen wil vaststellen, veel tijd besparen. De werking van maatregelen wordt immers vastgesteld door regelmatig, op basis van een controleprogramma, het bestaan te onderzoeken. De commandoprocedure dient dan als input voor de Toolkit, die op basis hiervan de gegevens verzamelt die de EDP-auditor nodig heeft voor de beoordeling van de mate van beveiliging.

Network Server
Normaal gesproken wordt de Toolkit geïnstalleerd op de VAX-computer die men wil onderzoeken. Het is echter niet denkbeeldig dat de desbetreffende VAX deel uitmaakt van een uitgebreid netwerk met meer VAXen. Indien men al deze computers wil onderzoeken, houdt dat in dat de Security Toolkit vele malen geïnstalleerd moet worden. Gelukkig is daar iets op gevonden: de network server. Deze maakt het mogelijk alle VAX-computers in een netwerk te controleren vanuit één bepaalde computer, namelijk die waarop de Toolkit is geïnstalleerd. Vanuit deze computer wordt een verbinding opgebouwd met de overige te onderzoeken computers in het netwerk. De werkwijze is verder precies zoals in het voorgaande is beschreven; het enige verschil is dat commando's nu eerst worden getransporteerd naar de andere VAX en daar pas worden uitgevoerd. De resultaten komen weer terug op de thuisbasis.

Output

Uitvoer kan zowel naar het scherm worden geschreven als naar bestanden. Ook is het mogelijk uitvoerresultaten door te sluizen naar andere applicaties ter verdere ver- en/of bewerking. Dit kunnen zowel VMS- (bijvoorbeeld Datatrieve) als PC-applicaties, zoals databases, spreadsheets en word-processors, zijn.

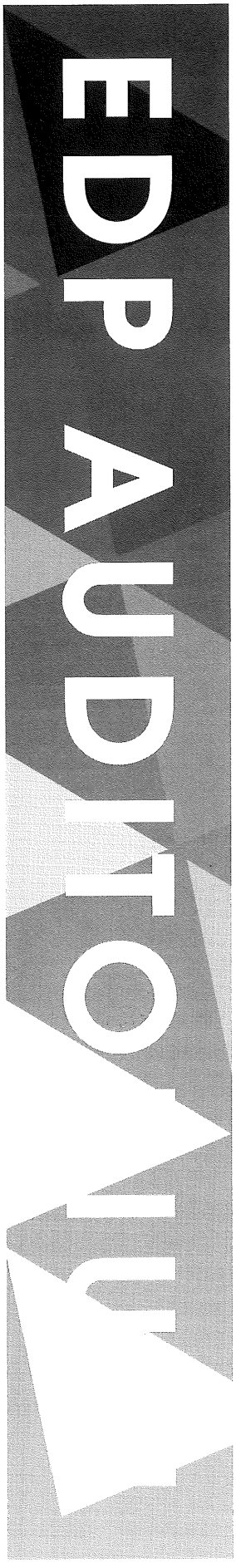
Ervaringen en Conclusie

De eerste versie van de Security Toolkit is door KPMG Klynveld EDP Audit getest. Onder andere naar aanleiding van deze testresultaten is de hiervoor besproken nieuwe versie 2.0 ontstaan. Met deze nieuwe versie zijn binnen EDP Audit ook reeds ervaringen opgedaan.
De ervaringen zijn over het algemeen positief. Met de Toolkit heeft de EDP-auditor een gereedschap in handen, waarmee relatief snel een volledig overzicht kan worden verkregen van de mate van beveiliging van een VAX/VMS-computersysteem. Dit overzicht kan worden samengevat en/of gesorteerd naar eigen inzicht. De Toolkit heeft echter geen intelligentie en is derhalve niet in staat de resultaten te beoordelen. Dit werk dient door de EDP-auditor te geschieden op basis van zijn kennis, algemene normen en eventueel door het management aangedragen normen en altijd in samenhang met de organisatie die wordt beoordeeld.

De Toolkit is overigens niet alleen voor de EDP-auditor bestemd; ook voor system managers, die toch dagelijks met de zorg voor het systeem zijn belast, kunnen er verrassende feiten naar voren komen. Bestanden bijvoorbeeld, waarvan men dacht dat ze goed beveiligd waren, blijken via een omweg wel degelijk toegankelijk voor onbevoegden.

Samenvattend kan worden gesteld dat de Toolkit is wat hij pretendeert te zijn: een gereedschapskist. Zeer waardevol derhalve, maar het uiteindelijke oordeel met betrekking tot de aangetroffen en ontbrekende beveiligingsmaatregelen is aan de vakman of -vrouw die het gereedschap hanteert.

Mw. G.J.C. Heikamp
Is sedert 1985 werkzaam bij KPMG Klynveld EDP Audit. De eerste jaren heeft zij zich beziggehouden met het ontwikkelen van applicatieprogrammatuur ten behoeve van de algemene controlepraktijk en het geven van PC-support en -cursussen aan de gebruiker. Na voltooiing van haar AMBI-studie heeft zij EDP-audits uitgevoerd zowel op het gebied van systeembeoordelingen als rekencentra-onderzoeken. Daarnaast studeert zij in de avonduren economie aan de Universiteit van Amsterdam. Haar specifieke kennis ligt op het gebied van de DEC/VAX-systeemprogrammatuur. Zij is verantwoordelijk voor de research-activiteiten naar de beveiligingsaspecten van VMS. In dat kader heeft zij deelgenomen aan het testen van de Security Toolkit VAX/VMS van Cubic; deze wordt door KPMG Klynveld EDP Audit gebruikt als hulpmiddel bij het uitvoeren van "security audits". Behalve lid van het NGI is zij ook lid van DECUS (Digital Equipment Computer Users Society) en heeft zij zitting in de DECUS Security-werkgroep.



EDP Auditorium

Boekbespreking *Het koekoeksei*

Drs.ing. J.C. van Winkel RI

Het is augustus 1986. Cliff Stoll, een astronoom, komt in dienst bij het Lawrence Berkeley Laboratory (LBL) als medesysteembeheerder van een aantal UNIX-systemen. Tijdens het analyseren van de resultaten van de twee onafhankelijk werkende accounting-systemen (voor het registreren van de gebruikte computertijd) blijkt een verschil van $0.75. Men schuift het probleem van tafel, maar Cliff Stoll is vasthoudend en gaat er, ook om wat meer ervaring te krijgen met UNIX, toch achteraan.

Dit is het begin van een maandenlange affaire. In zijn boek *Het koekoeksei* doet Cliff Stoll op spannende en humoristische wijze verslag van de activiteiten die uiteindelijk leidden tot de arrestatie van de boosdoener(s). Het gaat hier om een gedramatiseerd verslag van de werkelijke gebeurtenissen vanaf augustus 1986. Het bedrag van $0.75 was terug te leiden tot de activiteiten van een kraker op het computersysteem. De kraker had om zijn aanwezigheid op het systeem te verdoezelen een van de twee accounting-systemen buiten werking gesteld. Het andere accounting-systeem was door hem niet ontdekt, zodat de door hem gebruikte computertijd toch werd geregistreerd.
Door het loggen van alle activiteiten die door de hacker werden uitgevoerd en hem de toegang tot het systeem niet te ontzeggen, kwam men erachter dat de kraker met name was geïnteresseerd in militaire gegevens en tientallen computers binnendrong.
Om de kraker te kunnen blijven volgen was veel vindingrijkheid nodig. Zo werden enige tijd printers op alle binnenkomende telefoonlijnen aangesloten om te kunnen bepalen over welke fysieke lijn de hacker binnenkwam. Ook werd een PC speciaal ingericht om op de hackerslijn alle activiteiten gade te slaan, en wanneer de hacker inlogde *tegelijkertijd* via een modem een speciaal nummer te bellen. Zo werd Cliff Stoll steeds opgepiept zodra er activiteiten op de lijn werden gesignaleerd.
Cliff Stoll doet in het boek ook verslag van de ongeïnteresseerdheid van de verschillende "three-letter-word"-instellingen in de Verenigde Staten, zoals de FBI, de CIA en de NSA (National Security Agency). Deze instanties waren niet onder de indruk van de inbraakpogingen omdat het slechts ging om $0.75. Pas vlak voor de ontknoping begonnen de geheime agenten zich ermee te bemoeien, toen de zeer gerichte interesse van de kraker was vastgesteld.
In het boek wordt ook een duidelijk beeld gegeven van de technische mogelijkheden van het traceren van netwerkverbindingen. Het traceren van de verbinding van Californië naar Duitsland kostte enkele seconden, maar voor het traceren van de lokale telefoonverbinding in Duitsland was meer dan een uur nodig. Omdat de hacker zeer voorzichtig was en steeds slechts enkele minuten ingelogd bleef, creëerde Cliff Stoll met zijn medewerkers fake-informatie om de hacker lang genoeg "in de lucht" te houden. Uiteindelijk is de dader gearresteerd. Het bleek om een hacker te gaan met contacten in het Oostblok.

Het boek laat zich lezen als een spannende spionageroman, maar bevat zeer veel achtergrondinformatie over de gebruikte technieken, zowel van de dader als van Cliff Stoll.
Zeer zeker aan te raden!

Het koekoeksei
Cliff Stoll
ISBN 902 694 1676
Uitgeverij Unieboek

Prijs voor afstudeerscripties op het terrein van EDP-auditing

EDP-auditing kan zich verheugen op een snel toenemende erkenning van een vakgebied.
Er zijn inmiddels al drie universiteiten in Nederland die een gerichte EDP-audi-

ting-opleiding verzorgen. Dit is voor KPMG Klynveld EDP Audit aanleiding geweest om te besluiten een prijs van ƒ 5.000 uit te loven aan de student(e/n) die tussen 1 augustus 1989 en 1 augustus 1990 een afstudeerscriptie afrondt (afronden) in het vakgebied van de EDP-auditing en daarbij naar het oordeel van een jury het beste voldoet (voldoen) aan gestelde criteria als interdisciplinaire benadering, verband theoretische concepties en de praktijk, kritische en originele benadering, praktische toepasbaarheid en natuurlijk een heldere betoogtrant.

Sinds 1984 looft KPMG Klynveld Kraayenhof & Co. een dergelijke prijs uit voor Accountancy en KPMG Klynveld Bosboom Hegener voor Organisatie- en Informatiekunde.

KPMG Klynveld EDP Audit wil met deze prijs allereerst onderzoek op haar werkterrein stimuleren. Daarnaast wil zij bevorderen dat studenten in het uitvoeren van het onderzoek hun theoretische kennis aan de praktijk weten te koppelen.

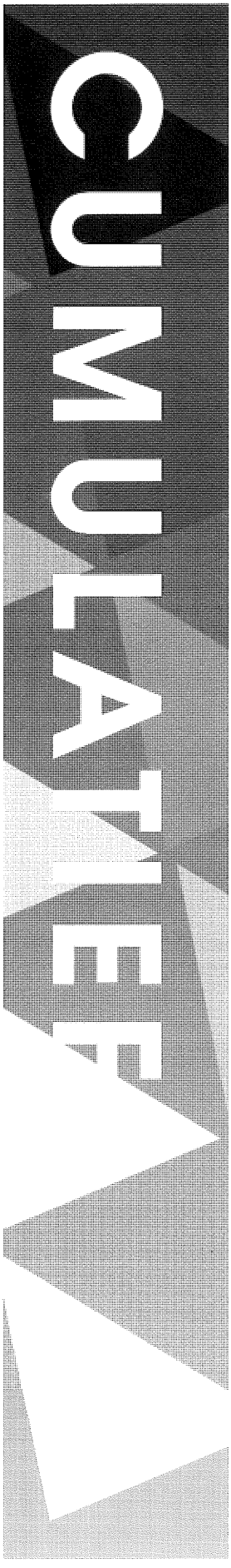
Scripties kunnen op het ruime terrein van de EDP-auditing zijn gericht. Zonder daarbij uitputtend te zijn, kan worden gedacht aan beoordeling en advisering op het terrein van:
-- betrouwbaarheid van de geautomatiseerde gegevensverwerking;
-- beveiliging van de automatisering;
-- telematica/telecommunicatie;
-- juridische aspecten van automatisering;
-- efficiency en effectiviteit;
-- risico-analyse.

Een deskundige jury zal de prijs toekennen. De jury voor de KPMG Klynveld EDP Audit Prijs bestaat uit de leden:
-- mevrouw mr. J.C.N. Couzijn, advocaat-generaal van het Gerechtshof te 's-Gravenhage;
-- de heer prof.dr. J.C. Arnbak, Technische Universiteit Delft (faculteit elektrotechniek);
-- de heer A.W. Neisingh RA, voorzitter KPMG Klynveld EDP Audit;
-- de heer drs. A.C.J.M. Nollen, lid van de Raad van Bestuur van Credit Lyonnais Bank Nederland B.V.;
-- de heer J. van Rijn RA, lid van de Hoofddirectie van de Rabobank Nederland.

Als secretaris van de jury fungeert de heer drs. H.G.Th. van Gils RA, senior EDP-auditor bij KPMG Klynveld EDP Audit.
Uitgebreide informatie en het reglement zijn verkrijgbaar bij het:

Secretariaat KPMG Klynveld Prijzen
Bureau Marketing en Publiciteit
Postbus 72001
1007 TB AMSTERDAM
Tel.: 020 - 5461726

en natuurlijk bij de secretaris van de KPMG Klynveld EDP Audit Prijs, tel.: 020 - 5461618.



Cumulatief

Overzicht van eerder verschenen artikelen in Compact

Een selectie van geactualiseerde artikelen uit de 12 1/2 jaar Compact 1974 - 1986 is opgenomen in het boek *24 over EDP-auditing.* 24 auteurs over EDP-auditing in theorie en praktijk: visies, ontwikkelingen, werkwijzen en oplossingen. Het boek is verkrijgbaar via de boekhandel onder nummer: ISBN 90 14 03648 5.

___42 13e jaargang 86/3 ZOMER 1986

Beveiliging: automatiserings- of organisatieprobleem / *drs. H.C. Kocks RA*

Conversie, Compilatie uit literatuur* / *drs. J. Kuipers*

De microcomputer in de accountantscontrole* / H. Veenman

* De artikelen met een * zijn tevens opgenomen in *24 over EDP-auditing.*

___43 14e jaargang 87/1 WINTER 1986 / LENTE 1987

Internationaal literatuuronderzoek naar computermisbruik in strafrechtelijk perspectief / *mr. V.A. de Pous*

Geïntegreerde gegevensverwerking / *drs. H.C. Kocks RA*

___44 14e jaargang 87/2 ZOMER 1987

Consequenties voor de beheersbaarheid ten gevolge van nieuwe technologische ontwikkelingen / *A.W. Neisingh RA*

Betalingsorganisatie en automatisering binnen de organisatie / *J. ten Wolde RA*

Electronic Banking-systemen in de praktijk / *drs. D.M. Swagerman*

___45 14e jaargang 87/3 HERFST 1987

Escrow-depot voor computersoftware in Nederland / *mr. V.A. de Pous*

Beveiligen tegen computermisbruik / *A.W. Neisingh RA en drs. J. Vossen*

Geïntegreerde gegevensverwerking: Structuur van controle- en beveiligingsmaatregelen in een ADR/DATACOM DB-DC-omgeving / *J.A.W. Winterink RA en drs. R.G.A. Fijneman*

Belangrijke functies van een toegangsbeveiligingspakket / *M.C. Duijm*

___46 14e jaargang 88/1 WINTER 1987 / 1988

SKE, Structured Knowledge Engineering *ing. A. van der Vlist*

Beveiliging bij datatransmissie / *ing. H.A.J.M. Spape*

Electronic Funds Transfer: het elektronisch uitvoeren van betalingen - literatuurstudie / *mw. ing. I.M. van Duin*

___47 15e jaargang 88/2 LENTE / ZOMER 1988 Special van de sectie Software Engineering

De sectie Software Engineering, een inleiding / *H. Veenman*

Software Engineering / *H. Veenman en ing. L.J.M.W. Gielen*

Het testen van software / *O. Kluyt*

UNIX / *ing. A. van der Vlist en ing. J.C. van Winkel RI*

Computervirussen / *ing. J.C. van Winkel RI*

Objects / *ing. L.J.M.W. Gielen*

HyperCard / *J. Schalk*

Programmeertheorie / *J. Schalk*

Het Apple Talk netwerk, een beschouwing / *J.L. Ramos Najera*

PS/2 - OS/2 / *ir. J. de Graaff en drs. D.J.P. Witte*

Elektronisch betalen, de betaalpas / *ing. J. Rotteveel*

___48 16e jaargang 89/1 LENTE 1989

Het uitvoeren van een transactie-analyse / *M.C. Duym*

Software escrow / *R.A. s'Jacob*

Computervirussen. Worm in groot netwerk / *drs.ing. J.C. van Winkel RI*

Beheersaspecten bij gebruik van microcomputers / *J.F.C. van Epen CISA*

The IBM AS/400. A concern to the EDP Auditor? / *H.J. Lijnes*

AS/400 security / *mw. V. Six*

Internationale gegevensstromen: abstract en moeilijk te controleren / *mr. V.A. de Pous*

___49 16e jaargang 89/2 ZOMER 1989

Beveiliging, noodzaak? / *J.L.H. Kooijman RA*

Beveiligingsbeleid formuleren / *drs. R. Schenk*

Informatiebeveiliging in het kader van automatisering / *drs. H.C. Kocks RA en drs.ing. H.A.J.M. Spape RA*

De keuze van beveiligingsmaatregelen in een geautomatiseerde omgeving / *drs. J. Kuipers RA*

De praktische methode voor de analyse van risico's bij automatisering / *ing. C.J.M. Gielen*

Organisatorische beveiliging van de geautomatiseerde gegevensverwerking / *J.C. Boer RA*

Fysieke beveiliging / *J.F.C. van Epen CISA*

Beveiligingsaspecten van computernetwerken / *drs.ing. H.A.J.M. Spape RA*

Logische toegangsbeveiliging / *J. Brinkman*

Beveiliging van de informatie in geautomatiseerde personeelsregistratiesystemen / *J.F.C. van Epen CISA*

___50 16e jaargang 89/3 WINTER 1989

De gevolgen van toepassing van informatietechnologie voor banken / *S. Lelieveldt*

Electronic Data Interchange (EDI) en Elektronisch Betalingsverkeer / *M. Groesz*

Vernieuwing geautomatiseerd verwerkingsproces van het betalingsverkeer bij de Postbank / *drs. C.P. Aland RA en A.H. Kuijlaars RA*

Mogelijkheden tot standaardisatie van de beveiliging van geautomatiseerd giraal betalingsverkeer / *drs. A. Hemelaar RA*

Geautomatiseerd uitgaand geldverkeer en het frauderisico / *drs. H.C. Kocks RA*

Cryptografische beveiliging van elektronisch berichten- en betalingsverkeer *drs. T.P. de Vries*

S.W.I.F.T. en Controle / *drs. P.M. Knuvers en ing. G.H.M. Meijer*