Technische beveiligingsstandaard Unix V 2.0

M. Buijs
P. Buurman
W. van Dommelen
A. Koot
P. Kornelisse
H. Linschooten
N. van Nieuwpoort
P. Veltman
M. Venderbosch

PRAKTIJKREEKS
INFORMATIEBEVEILIGING

tenHagenStam
UITGEVERS

Reeds verschenen in de praktijkreeks informatiebeveiliging:

1 Internet, intranet en beveiliging: het technische kader
2 Wegwijzer voor IT-uitbestedingscontracten
3 Internet, intranet en beveiliging: het organisatorische kader
4 Elektronische werkplekbeveiliging
5 Vier jaar VIR, vloek of zegen?
6 Cryptografie in de praktijk
7 Checklist informatiebeveiliging
8 Technische beveiligingsstandaard Windows NT
9 Oursourcing
10 Beveiliging en Service Level Agreements

P R A K T I J K R E E K S
I N F O R M A T I E B E V E I L I G I N G

Eindredactie: Christian Jongeneel

Ontwerp omslag en binnenwerk: Bottenheft

ISBN 90 440 0195 7

Het Platform Informatiebeveiliging (PI) heeft als doel 'het bevorderen van de beveiliging van alle belangen betreffende gegevensverwerking en gegevenstransport, alles in de ruimste zin van het woord'. Binnen deze doelstelling wordt het ontwikkelen van aanvaardbare richtlijnen voor de praktische inrichting van informatiebeveiliging als essentieel onderwerp gezien. Door het gezamenlijk opstellen van dergelijke richtlijnen kan worden gebruikgemaakt van praktijkervaringen, zodat een doeltreffende richtlijn ontstaat die ook uitvoerbaar is.
De PI-richtlijnen worden in werkgroepverband ontwikkeld onder auspiciën van de Productraad. De Productraad is een speciaal orgaan van de vereniging Platform Informatiebeveiliging, waarvan de leden worden benoemd door het bestuur. De Productraad heeft onder meer tot taak het samenstellen van de werkgroepen, het bewaken van de voortgang, het opstellen van algemeen geldende uitgangspunten, het beoordelen van conceptrapportages en het adviseren van het bestuur omtrent het vaststellen van rapportages.
De deelnemers van de werkgroepen zijn beveiligingsfunctionarissen en EDP-auditors van uiteenlopende bedrijven en instellingen. Zij kenmerken zich door de hoge eisen die zij in hun advies- en controlewerkzaamheden aan organisaties moeten stellen in verband met de sterke automatiseringsgraad en de belangen die met de geautomatiseerde gegevensverwerking zijn gemoeid. Door deze achtergrond vormen de deelnemers een representatieve afspiegeling van de aanwezige IT-beveiligingsexpertise in Nederland en bieden zij een draagvlak om gezag te verlenen aan de ontwikkelde richtlijnen, hetgeen bevorderlijk is voor de acceptatie door het algemene en het IT-management.

De PI-beveiligingsrichtlijnen zijn primair bedoeld voor functionarissen die zijn belast met het implementeren van IT-systemen, zoals systeembeheerders en systeemprogrammeurs. Daarnaast zijn de richtlijnen van betekenis voor de volgende doelgroepen:
– *IT-beveiligingsfunctionarissen* (security officers en administrators). De IT-beveiligingsfunctie binnen een organisatie is verantwoordelijk voor het (doen) treffen van beveiligingsmaatregelen. De richtlijnen bieden hierbij ondersteuning.

– *IT- en algemeen management*. Het management is (eind)verantwoordelijk voor de informatiebeveiliging en geeft hieraan invulling door het (doen) analyseren van risico's en het bepalen van (globale) beveiligingsdoelstellingen. De beargumenteerde keuzen en de managementsamenvatting in de richtlijnen zijn hierbij een handvat.
– *EDP-auditors*. De richtlijnen geven – gemotiveerd – de vereiste beveiligingsmaatregelen aan en de risico's indien niet aan de vereisten is voldaan. Hierdoor kunnen de richtlijnen ook worden gehanteerd als toetsingsnorm bij EDP-audits.

Aldus bieden de richtlijnen enerzijds een handreiking aan beveiligingsfunctionarissen en het algemene en IT-management om een toereikende en evenwichtige beveiliging van de informatievoorziening te implementeren en bieden zij anderzijds een basis aan EDP-auditors voor de normstelling bij de beoordeling van de beveiliging van een IT-systeem.
Deze PI-standaard Unix V2.0 is een algehele herziening van de OTB standaard Unix V1.0.

De auteurs willen de volgende personen bedanken voor hun adviezen: W. Moolendijk (Ministerie van Financiën, belastingdienst), J.F. Bautz (NGIB), C. Dik (Sun Microsystems) en C. van der Zwan (De Nederlandsche Bank).
De volgende personen reviewden het manuscript: T. Pecholt (DSM), J. Seeboldt (Ahold), E. Beijer (Ahold), G. Jonkheer (Ahold), P. Keuling (Ahold), S. de Kopp (Ministerie van Financiën), H. Blankesteijn (Ministerie van Financiën), R. van Slageren (Siemens Nederland), R. Oussooren (Xirion BV), A. Latupeirissa (Xirion BV), O. van der Voort (Defensie Telematica Organisatie), E. Lippe (Fortis Nederland B.V.), J.W. Kroes (Wehkamp B.V.), H. Veerman (ING Nederland), J. Visser (NLUUG), J.C. Winkler (NLUUG), J. van de Graat (AT Computing).

De auteurs:

M. BUIJS, Defensie Accountantsdienst, Auditgroep EDP-Auditing
P. BUURMAN, Xirion BV
W. VAN DOMMELEN, EDP AUDIT POOL
A. KOOT, Ministerie van Financiën, Belastingdienst
P. KORNELISSE, KPMG EDP Auditors
H. LINSCHOOTEN, Hewlett Packard BV
N. VAN NIEUWPOORT, KPMG EDP Auditors
P. VELTMAN, KPMG EDP Auditors
M. VENDERBOSCH, ING Nederland, ITC/Information Security Services, GAK Interne Accountantsdienst

1 Inleiding

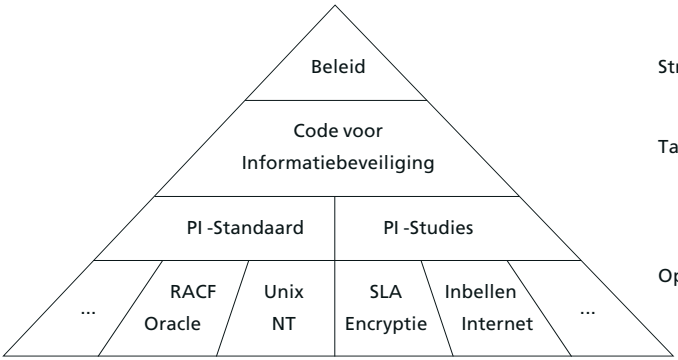
De toenemende integratie van automatisering met de bedrijfsprocessen en de eveneens toenemende complexiteit van automatiseringsoplossingen noodzaken tot voortdurende aandacht voor zowel het vereiste niveau van informatiebeveiliging als de technische realisering hiervan. Ook gezien de ontwikkelingen op het gebied van wet- en regelgeving met betrekking tot informatiebeveiliging is deze aandacht noodzakelijk.
Objectivering van het vereiste niveau van informatiebeveiliging en van de effectiviteit van gekozen technische oplossingen is voor veel organisaties een probleem, doordat slechts in beperkte mate standaarden voorhanden zijn. Beschikbare standaarden kennen ofwel een te beperkt werkingsgebied, of richten zich te veel op de organisatorische kant van de informatiebeveiliging. Door het gebrek aan deugdelijke standaarden zijn organisaties gedwongen zelf oplossingen te ontwikkelen en hieraan veel energie te besteden. De gevolgen zijn suboptimale oplossingen, verspilling doordat vele malen opnieuw het wiel moet worden uitgevonden en moeizame acceptatie door de afwezigheid van geobjectiveerde criteria.
Tegen deze achtergrond is het initiatief ontstaan om in werkgroepverband concrete, geobjectiveerde richtlijnen en normen te ontwikkelen voor de inrichting respectievelijk de beoordeling van de effectiviteit van technische beveiligingsmaatregelen. Deze aanpak heeft de volgende voordelen:

- door uitwisseling van kennis, ervaring en inzicht ontstaat een belangrijk synergie-effect tussen de deelnemers; de deelnemers kunnen elkaar ondersteunen bij de keuze en implementatie van beveiligingsmaatregelen;
- met behulp van de ingebrachte kennis en inzichten kan worden gekomen tot de vaststelling van technische beveiligingsrichtlijnen die op een breed draagvlak kunnen rekenen;
- toepassing van de opgestelde beveiligingsrichtlijnen leidt bij de betrokken deelnemers tot een verhoging van de effectiviteit van de beveiliging, waarbij ook sprake is van een efficiënte en reeds in de praktijk getoetste aanpak.

Context De technische beveiligingsmaatregelen die in de richtlijnen worden beschreven, vormen een onderdeel van de bredere context van het gehele samenstel van beveiligingsmaatregelen om de kwaliteit van de geautomatiseerde informatievoorziening te waarborgen. Beveiligingsmaatregelen binnen deze bredere context zijn bijvoorbeeld beschreven in de publicatie *Code voor Informatiebeveiliging. Een leidraad voor Beleid en Implementatie* [Code]. Deze Code, die vooral in het bedrijfsleven wordt gebruikt, richt zich in het bijzonder op het tactische niveau binnen organisaties en bestrijkt het gehele terrein van informatiebeveiliging. Door het abstractieniveau geeft de Code echter weinig concrete handvatten voor het implementeren van IT-systemen. Hetzelfde geldt voor het besluit *Voorschrift Informatiebeveiliging Rijksdienst*, dat voor de rijksoverheid van toepassing is [VIR].
De PI-richtlijnen kunnen dan ook worden beschouwd als een verdere uitwerking van de Code en de baselinebeveiliging van het besluit VIR en zijn vooral gericht op het operationele niveau binnen organisaties. Daarnaast kan nog een strategisch niveau worden onderkend, dat betrekking heeft op de eindverantwoordelijkheid voor informatiebeveiliging van het topmanagement. De samenhang tussen deze drie niveaus is schematisch weergegeven in figuur 1.

Aangezien voor het tactische niveau van informatiebeveiliging en voor de beleidsmatige en organisatorische maatregelen die op het strategische en tactische niveau moeten worden getroffen, al veel literatuur voorhanden is, wordt hierop in de PI-richtlijnen niet nader ingegaan. Het uitgangspunt van de PI-richtlijnen is dat op dit gebied voldaan is aan de Code voor Informatiebeveiliging en vergelijkbare

FIGUUR 1



De informatiebeveiligingspiramide.

standaarden, hetgeen inhoudt dat er een beveiligingsbeleid is, dat er functiescheiding is tussen ontwikkeling en productie, etc.
Bij de implementatie van een product of architectuur moet een evenwicht worden gevonden tussen risico's en daarmee samenhangend beveiligingsniveau, gebruikersgemak, invoerings- en beheerkosten en gevolgen voor de prestaties van het systeem. De richtlijnen bieden hierbij een praktische leidraad, doordat beargumenteerd wordt aangegeven waarom bepaalde keuzen zijn gemaakt. Door deze aanpak kunnen organisaties de vertaalslag maken naar hun eigen specifieke omstandigheden.

PI-Beveiligingsrichtlijnen De beveiligingsrichtlijnen die in PI-verband zijn en worden ontwikkeld, betreffen de technische maatregelen en voorzieningen die moeten worden getroffen ter waarborging van de integriteit, vertrouwelijkheid en beschikbaarheid van de gegevens die met een IT-systeem worden opgeslagen, verwerkt en/of getransporteerd.
De richtlijnen geven dus aan hoe de (beveiligings)functies van een IT-systeem die relevant zijn voor de genoemde kwaliteitsaspecten, moeten worden ingesteld. Zij bevatten tevens aanwijzingen voor de organisatorische inbedding hiervan, maar primair gaat het om de techniek.
Met de richtlijnen wordt vooral beoogd te bevorderen dat de integriteit, vertrouwelijkheid en beschikbaarheid van de gegevens in voldoende mate is gewaarborgd. Dat wil niet zeggen dat andere kwaliteitsaspecten, zoals efficiëntie (bedieningsgemak, performance), uit het oog worden verloren. Juist door de inbreng vanuit de praktijk wordt gestreefd naar een optimaal evenwicht tussen beveiligingsniveau en praktische realiseerbaarheid. De richtlijnen vormen de neerslag van de gezamenlijke kennis, inzichten en praktische ervaringen van de werkgroep- en PI-leden.
De richtlijnen kunnen betrekking hebben op elke component van de IT-infrastructuur in de breedste zin van het woord (aangeduid als IT-systeem). De IT-infrastructuur betreft het geheel van apparatuur, besturings- en hulpprogrammatuur, faciliteiten voor data-, spraak- en videocommunicatie, alsmede de fysieke beveiligingsfaciliteiten van de geautomatiseerde informatievoorziening. Ook generieke toepassingen en diensten (zoals e-mail en file transfer) vallen onder dit begrip IT-systeem.
De beveiligingsrichtlijnen van PI vallen uiteen in twee categorieën:

– *PI-standaarden*. PI-standaarden zijn technische implementatiehandleidingen voor concrete objecten (IT-producten), bijvoorbeeld een besturingssysteem van een bepaalde leverancier en een bepaalde versie;

– *PI-studies*. PI-studies zijn technische beveiligingshandleidingen voor objecttypen, bijvoorbeeld een bepaalde categorie besturingssystemen, generieke IT-architecturen, zoals een firewall, Internetverbinding of inbelfaciliteit, generieke diensten, bijvoorbeeld directory services en dergelijke.

Bij nieuwe versies van producten en bij nieuwe technologische of maatschappelijke ontwikkelingen bestaat er behoefte aan vroegtijdige risico-inschatting en standpuntbepaling met betrekking tot de invulling van beheer- en beveiligingsaspecten. In die gevallen zijn de richtlijnen meer het resultaat van een researchinspanning dan dat zij – zoals bij bestaande producten en architecturen – zijn gebaseerd op eigen praktische ervaring ('best practice').

Het te bereiken beveiligingsniveau dient te zijn afgestemd op de waarde van het beveiligde belang. Aangezien dit voor elke organisatie verschillend zal zijn, zijn de PI-richtlijnen primair gebaseerd op de beveiligingsmogelijkheden van het IT-systeem, dat wil zeggen op het optimaal benutten van de beveiligingsfaciliteiten die het biedt. Dit wordt het beginsel van goed huisvaderschap genoemd. Bij het opstellen van de richtlijnen wordt echter tevens nagegaan aan welk niveau het systeem redelijkerwijs zou moeten voldoen, gegeven de Code voor Informatiebeveiliging en andere gezaghebbende literatuur, de 'state of the art' van de beveiligingstechniek, de gemeenschappelijke 'common sense' van de werkgroepleden, enzovoort.

Uitgangspunten voor Unix Bovenstaande richtlijnen, met name het principe van goed huisvaderschap leiden tot een zestal uitgangspunten, die gehanteerd worden bij de totstandkoming van de PI-standaard voor Unix-omgevingen. Deze uitgangspunten zijn samengevat in tabel 1 en worden hieronder nader uitgewerkt.

TABEL 1

Kwaliteitsaspecten	Vertrouwelijkheid, integriteit, beschikbaarheid
Basisprincipes	Goed huisvaderschap; beperking van functionaliteit, isolatie, veilige beginwaarden
Unix-omgeving	Geen open verbinding met de buitenwereld Voldoende fysieke beveiliging
Beheer	Één systeembeheerder(sorganisatie)
Unix-versies	Gebaseerd op Linux Generiek; niet leverancierspecifiek
Niet-commerciële tools	Mits integriteit en authenticiteit gewaarborgd zijn

Uitgangspunten van de PI-beveiligingsstandaard voor Unix.

Uitgangspunt 1 De standaard is gericht op het waarborgen van de kwaliteitsaspecten vertrouwelijkheid, integriteit en beschikbaarheid. Unix-omgevingen worden ingezet voor zeer uiteenlopende toepassingen, waarbij elke toepassing zijn eigen eisen op het gebied van beveiliging stelt. In navolging van de Code voor Informatiebeveiliging [Code] worden in deze standaard de kwaliteitsaspecten vertrouwelijkheid, integriteit en beschikbaarheid als norm gehanteerd.
Bij de uitwerking van deze norm is gebruikgemaakt van de algemeen geaccepteerde begrippen subject en object. Subjecten zijn hierbij de actieve elementen in het systeem (zoals processen en gebruikers), objecten zijn de passieve elementen (zoals bestanden, directory's en devices). Gegeven deze subjecten en objecten in een Unix-systeem hebben de kwaliteitsaspecten een exclusief karakter: objecten mogen alleen gelezen, respectievelijk geschreven worden door subjecten, die direct of indirect draaien in opdracht van daartoe bevoegde natuurlijke personen. Deze bevoegdheden dienen in de organisatiestructuur te zijn vastgelegd. Het beginsel van accountability brengt daarbij met zich mee dat alle acties op het systeem tot natuurlijke personen herleidbaar moeten zijn, hetgeen authenticatie van gebruikers door het systeem vereist. Ten slotte moet het systeem de mogelijkheid bieden om controle uit te oefenen op de naleving van bevoegdheden (logging).
Het kwaliteitsaspect beschikbaarheid (continuïteit) komt in deze standaard slechts summier aan de orde, omdat het Unix-besturingssysteem relatief weinig beveiligingsfuncties ter waarborging van dit kwaliteitsaspect biedt. Maatregelen op dit vlak dienen eerder gezocht te worden in de fysieke beveiliging van computersystemen en netwerkcomponenten, de beheerprocessen, het implementeren van een adequaat back-upschema en het invoeren van redundantie op hardwareniveau, zoals het geval is bij non-stop computersystemen en disk mirroring. Zulke maatregelen vallen buiten het bestek van deze PI-standaard.
In de standaard wordt bijzondere aandacht besteed aan maatregelen die noodzakelijk zijn om tegen te gaan dat de integriteit van het besturingssysteem zelf wordt aangetast. In dat geval namelijk kunnen alle op het systeem ingestelde beveiligingsmaatregelen worden omzeild of gedeactiveerd, waardoor de vertrouwelijkheid, integriteit en beschikbaarheid niet langer gewaarborgd zijn.

Uitgangspunt 2 Bij het opstellen van deze standaard zijn drie klassieke beveiligingsprincipes gehanteerd [Saltzer]:

- *Beperking van functionaliteit.* De essentie van dit principe, ook bekend onder de namen 'need to use' en 'least privilege', is dat gebruikers alleen hoeven te beschikken over de functies die ze voor het uitoefenen van hun werk nodig hebben; alle overbodige functies worden uitgeschakeld.
- *Isolatie*. Volgens dit principe, ook bekend onder de naam 'economy of mechanism', dient alle voor de beveiliging relevante functionaliteit in een zo compact mogelijk onderdeel van het systeem te worden opgenomen.
- *Veilige beginwaarden*. Dit principe, dat ook te boek staat als 'fail-safe defaults' of 'safe attribute initialisation', stelt dat variabelen door de systeembeheerder of door de gebruiker vanaf het begin zijn ingesteld op maximale beveiliging, zodat voor het toekennen van permissies of het inschakelen van functionaliteit een expliciete handeling noodzakelijk is.

Uitgangspunt 3 De standaard is gericht op een Unix-omgeving die geïsoleerd en voldoende fysiek beveiligd is. Moderne Unix-omgevingen staan niet meer op zichzelf, maar maken deel uit van uiteenlopende netwerkomgevingen. Deze standaard is gericht op netwerkomgevingen die uit meerdere Unix-systemen en andere platforms bestaan en logisch afgeschermd zijn, bijvoorbeeld met behulp van firewalls, filters die specifieke verkeersstromen doorlaten en andere verkeersstromen tegenhouden. In het bijzonder mag een Unix-omgeving in het algemeen niet zijn blootgesteld aan bedreigingen die voortvloeien uit directe koppelingen met externe computernetwerken. Het belang van netwerkbeveiliging dient benadrukt te worden, omdat de beveiliging van een Unix-omgeving in sterke mate afhankelijk is van de mate van beveiliging van de netwerkomgeving waarin deze is opgenomen. Uitgangspunt is verder dat de desbetreffende Unix-omgeving wordt gekenmerkt door een voldoende mate van fysieke beveiliging.

Uitgangspunt 4 De standaard is gericht op Unix-systemen in netwerkomgevingen die worden beheerd door één organisatorische eenheid. De standaard gaat pragmatisch om met het gegeven dat veel moderne netwerkomgevingen worden gekenmerkt door het ontbreken van de in de mainframewereld gebruikelijke functiescheiding tussen de verschillende beheersfuncties. Is er al sprake van functiescheiding, dan is deze meestal niet ingegeven door overwegingen van interne controle, maar door functionele specialisatie. In deze standaard is daarom uitgegaan van een 'single system manager' – één functie, die overigens door meerdere personen kan worden uitgevoerd. Dit uitgangspunt houdt een vertrouwen ten aanzien van de systeembeheerder in. In omgevingen waarin een dergelijk vertrouwen een (te) hoog risico met zich zou meebrengen, zullen aanvullende preventieve en repressieve maatregelen noodzakelijk zijn. Een adequate logging (en controle daarop) van systeembeheerdersactiviteiten en het op periodieke basis (laten) verrichten van IT-audits zijn hiervan voorbeelden. In deze standaard wordt op enkele punten uitgegaan van functiescheiding op beheerniveau. Hierbij worden functies als functioneel applicatiebeheerder, technisch applicatiebeheerder en gegevensbeheerder onderkend; deze functies kunnen al dan niet verbijzonderd zijn.

Uitgangspunt 5 De standaard is generiek. Ondanks alle inspanningen op het gebied van standaardisatie komen er vele verschillende versies en releases van het Unix-besturingssysteem voor, die soms meer van elkaar verschillen dan wenselijk is. Vanaf de eerste Unix-versie, zoals die eind jaren zestig door Thompson en Ritchie bedacht is, hebben diverse commerciële en niet-commerciële partijen een bijdrage geleverd aan de ontwikkeling van het systeem. Een belangrijke tweedeling was die tussen de versies van AT&T (nu Unix System V Release 4) en van de softwaredistributies van de universiteit van Berkeley (4.xBSD). Beide versies werden initieel kosteloos ter beschikking gesteld aan universiteiten en onderzoeksinstellingen, die het systeem relatief ongecoördineerd en ongestructureerd hebben uitgebreid. Daarnaast bestaan er significante verschillen tussen de Unix-versies van de verschillende leveranciers, die door het toevoegen van functionaliteit steeds enig concurrentievoordeel hopen te behalen. Er zijn vele pogingen geweest om de verschillende versies tot één geheel samen te smeden; de belangrijkste voorbeelden van zulke standaarden zijn de System V Interface Definition (SVID), IEEE Posix en X/Open's Spec1170, recentelijk omgedoopt tot XPG4. Telkens weer is de bij de leveranciers gevoelde commerciële druk om met een 'eigen',

uitgebreidere versie te komen sterker gebleken. Bekende voorbeelden van zulke 'eigen' systemen zijn AIX (IBM), HP-UX (Hewlett-Packard), Solaris en SunOS (Sun Microsystems), OSF/1 (Digital Equipment Corporation), IRIX (Silicon Graphics), SCO Unix (Santa Cruz Operations) en Linux (niet-commercieel). Op dit moment zijn ten minste 200 Unix-varianten in omloop, waarbij van elke variant nog meerdere systeemversies in gebruik zijn.
Door deze enorme diversiteit is er grote behoefte aan generieke normen; het opstellen van zulke normen wordt echter door diezelfde diversiteit belemmerd. In deze PI-standaard moest het niveau van detail van de maatregelen daarom met zorg worden gekozen. Er is zoveel mogelijk gestreefd naar nauwkeurigheid, volledigheid en praktische haalbaarheid, zonder daarbij te pretenderen universeel geldige bedieningsinstructies te kunnen geven. Commando's en bestandsnamen zijn zoveel mogelijk generiek weergegeven, waarbij gebruik is gemaakt van de Unix-variant Linux. Reden hiervoor is dat Linux niet leveranciersgebonden is en zich redelijk conformeert aan de door de internationale Unix-wereld gedefinieerde standaarden (zoals Posix en The Single Unix Specification).
Het opsommen van pasklare Unix-commando's en instellingen is bewust vermeden. Een deskundig systeembeheerder zal de genoemde maatregelen zonder veel problemen naar specifieke Unix-commando's en -instellingen kunnen vertalen. Daar waar relevant zijn leverancierspecifieke maatregelen opgenomen die afwijken van de Linux-standaard. De leverancierspecifieke bijlagen worden apart onderhouden.
Daarnaast zal het raadplegen van specifieke documentatie naast deze technische beveiligingsstandaard noodzakelijk blijven.

Uitgangspunt 6 Het gebruik van niet-commerciële software voor beveiligingstoepassingen wordt aangemoedigd, maar uitsluitend indien de integriteit, authenticiteit en toekomstvastheid van deze software voldoende gewaarborgd is. Bij de beveiliging van moderne Unix-omgevingen genieten niet-commerciële tools een grote populariteit. Tools als Cops, Crack, Tripwire, Satan en TCP Wrapper worden op grote schaal toegepast en het gebruik van deze tools wordt door velen als een basismaatregel gezien. Inderdaad bieden deze en andere tools nuttige beveiligingsfuncties die in de 'standaard'-versies van Unix ontbreken. De beheerder dient zich echter te realiseren dat de integriteit, authenticiteit en toekomstvastheid van zulke tools in veel gevallen niet gewaarborgd zijn. De hieraan verbonden risico's gebieden een voorzichtige opstelling ten aanzien van het gebruik van zulke tools. Volledige zekerheid kan nooit worden verkregen, maar men mag er in redelijkheid van uitgaan dat de tools die gedistribueerd worden door de Computer Emergency Response Teams (CERTs) wel voldoende waarborgen bieden – mits de integriteit en authenticiteit kunnen worden gecontroleerd aan de hand van een checksum.

Opzet van dit boek In hoofdstuk 2 wordt eerst een globaal overzicht geschetst van de beveiligingsstructuur van Unix. De daaropvolgende hoofdstukken gaan steeds in op specifieke beveiligingsaspecten, te weten bevoegdheden van gebruikers (hoofdstuk 3) en objecten (hoofdstuk 4), logging en monitoring (hoofdstuk 5), en netwerk-

diensten (hoofdstuk 6). Het zevende hoofdstuk behandelt enkele zaken die het dagelijkse beheer aangaan. De hoofdstukken 3 tot en met 7 bevatten steeds secties 'normen en maatregelen', waarin de praktische aanwijzingen voor een goede beveiliging gebundeld zijn. Daar waar twee veelgebruikte Unix-implentaties, Sun Solaris en HP-UX, afwijken van de standaard, is dit steeds aan het eind van de betreffende paragraaf aangegeven. Het laatste hoofdstuk biedt enkele afsluitende opmerkingen en een recapitulatie van de belangrijkste te nemen maatregelen. In bijlage 1 is ten slotte aangegeven hoe de hier behandelde onderwerpen zich verhouden tot de Code voor Informatiebeveiliging, terwijl bijlage 2 een beknopt overzicht geeft van op Internet gratis te verkrijgen hulpmiddelen voor het beheer van Unix-systemen.

2 Beveiligingsstructuur van Unix

Dit hoofdstuk geeft een korte beschrijving van het Unix-besturingssysteem, die noodzakelijkerwijs beperkt blijft tot die onderwerpen die vanuit beveiligingsoogpunt relevant zijn.
Unix is aan het eind van de jaren zestig ontwikkeld bij AT&T's Bell Laboratories door Ken Thompson en Dennis Ritchie, twee programmeurs die al eerder hadden gewerkt aan de ontwikkeling van het ambitieuze besturingssysteem Multics. In tegenstelling tot Multics was Unix in beginsel eenvoudig van opzet. In de loop der jaren is Unix echter uitgegroeid tot een complex systeem dat bestaat uit een grotendeels in de taal C geschreven *kernel* en vele applicaties, die veelal onafhankelijk van elkaar zijn ontwikkeld door verschillende partijen. Unix is door veel verschillende leveranciers met kleine of grote verschillen ten opzichte van een niet-aanwezige standaard geïmplementeerd.
Unix staat niet bekend als het meest veilige besturingssysteem; het is oorspronkelijk ontwikkeld voor gebruik in een besloten, vertrouwde onderzoeksomgeving. Omwille van de efficiëntie, de toegankelijkheid en de 'programmeursvriendelijkheid' hebben de ontwerpers er destijds voor gekozen een aantal algemene beveiligingsprincipes te laten varen, waaronder de drie die in de inleiding onder uitgangspunt twee genoemd werden: beperking van functionaliteit, isolatie en veilige beginwaarden. Dit gebeurde in de veronderstelling dat gebruik van het systeem beperkt zou blijven tot kleine, vertrouwde werkomgevingen met een laag risico.
Het beveiligen van een Unix-omgeving is hierdoor een relatief complexe en foutgevoelige aangelegenheid, die een aanzienlijke inspanning vereist, een inspanning die door de introductie van laagbeveiligde netwerktechnologieën en andere uitbreidingen van de functionaliteit sterk is toegenomen en in de praktijk nogal eens wordt onderschat. Daarbij heeft het langdurige gebruik van Unix in academische omgevingen een groot aantal kwetsbaarheden aan het licht gebracht. In dit opzicht onderscheidt Unix zich overigens in positieve zin van vele andere besturingssystemen.

Veel van de oorspronkelijke tekortkomingen in de beveiliging zijn in nieuwe versies door de leveranciers opgelost; sommige Unix-versies

voldoen aan de C2-norm van de Trusted Computer System Evaluation Criteria (TCSEC, ook bekend als het Orange Book) van het Amerikaanse Department of Defense. In de meeste gevallen is de toegevoegde beveiligingsfunctionaliteit echter leverancierspecifiek (proprietary) en voldoet deze niet aan een algemeen aanvaarde standaard, hetgeen de normstelling ten aanzien van deze toegevoegde beveiligingsfuncties in belangrijke mate belemmert. Opgemerkt dient te worden dat niet alle applicaties in de C2 trusted modes van Unix'en draaien. Sommige applicaties verwachten in `etc/passwd` het versleutelde password, zodat ze zelf passwordverificatie van de gebruiker kunnen uitvoeren. Behalve door organisatorische functiescheidingen, die ook technisch geïmplementeerd dienen te worden, kan tevens door middel van technische voorzieningen en de toepassing van beheerhulpmiddelen een hoog niveau van beveiliging worden gerealiseerd. Zo biedt een aantal leveranciers de mogelijkheid om twee beveiligingsniveaus toe te passen.

Essentieel voor informatiebeveiliging zijn de eerder geïntroduceerde subjects en objects. De subjects in een Unix-omgeving komen overeen met processen. Elk proces draait in opdracht van een (menselijke) gebruiker. De objects in een Unix-omgeving komen overeen met bestanden en directory's. De toegangspermissies van subjects tot objects worden weergegeven door middel van protection bits. In de hierna volgende paragrafen wordt eerst nader ingegaan op subjecten (gebruikers en processen), daarna op objecten (bestanden en directory's) en ten slotte op de toegang van subjecten tot objecten. Naast subjects en objects bestaan in Unix ook functies die moeten worden afgeschermd, bijvoorbeeld het zetten van de tijd, het herstarten van het systeem.

2.1 GEBRUIKERS EN PROCESSEN

Een Unix-omgeving wordt doorgaans gebruikt door één of meer menselijke gebruikers (users). Deze gebruikers worden door de systeembeheerder gedefinieerd in het passwordbestand: `/etc/passwd.` Dit bestand, dat voor een ongestoorde werking van het systeem aawezig én leesbaar moet zijn, bestaat uit regels, waarbij elke regel een account beschrijft en een aantal velden bevat, de velden worden gescheiden door een dubbele punt (zie tabel 2).
Een voorbeeld van een regel in het passwordbestand:

```
tuttle:gR8fx./tQ2DmX:425:104:Arch.
Tuttle:/home/tuttle:/bin/ksh
```

TABEL 2

Veld	Betekenis
Username	De naam waarmee de gebruiker bij het inloggen wordt geïdentificeerd
Password	Een bitpatroon dat een versleuteling is van het wachtwoord, versleuteld op basis van een unieke sleutel
User-id (UID)	De gebruikersidentificatie die intern door het systeem gebruikt wordt
Group-id (GID)	De groepsidentificatie die intern door het systeem gebruikt wordt
Comment field	Commentaarveld met vrije tekst; veelal de echte naam van de gebruiker
Home directory	De login directory van het account
Login shell	Door het loginprogramma uitgevoerd programma, veelal een interactieve shell

Velden in het passwordbestand `/etc/passwd`.

HP-UX wordt standaard geleverd met de mogelijkheid om in non-trusted of in C2 level trusted mode te opereren. In de non-trusted mode werkt het systeem over het algemeen zoals omschreven in deze standaard. Echter, zodra gekozen wordt voor de C2 level trusted mode, is onder ander het volgende van belang:

- het wachtwoord van gebruikers is niet meer aanwezig in het passwordbestand, maar is opgeslagen in de zogenaamde Trusted Computer Base. De passworddatabase is dan te vinden onder `/tcb/files/auth/`;
- het passwordveld in `/etc/passwd` wordt gevuld door een '`*`';
- een uniek audit-id voor elke gebruiker wordt gegenereerd;
- alle gebruikers dienen een wachtwoord te hebben;
- de mogelijkheid bestaat om alle of een subset van de systemcalls van een audit te voorzien;
- geselecteerde gebruikers kunnen worden geaudit.

Met behulp van SAM, de 'System Administration Manager' beheertool is deze C2-mode aan of uit te zetten. Verder biedt SAM de mogelijkheid van de zogenaamde Restricted SAM. Aan niet-root-gebruikers kunnen bepaalde beheeronderdelen van SAM worden uitgedeeld, zodat er een bepaalde wijze van rollenscheiding kan worden gecreëerd.

2.1.1 Processen De karakteristieke Unix-omgeving bestaat uit een groot aantal programma's die gelijktijdig worden uitgevoerd. Elk programma in uitvoering (proces) draait direct of indirect in opdracht van een gebruiker of beheerder. In specifieke velden van de procestabel van het besturingssysteem wordt voor elk proces daarom een user-id (UID) en een group-id (GID) bijgehouden. Deze variabelen worden vooraf door de systeembeheerder voor elke gebruiker gedefinieerd. Als de gebruiker inlogt, worden de juiste user-id en group-id in de procestabel van het besturingssysteem ingevuld. De user-id en de group-id vormen daarna de belangrijkste basis voor

toegangsbeveiliging in Unix. Ieder proces heeft één of meer group-id's aan zich gekoppeld. Zie ook tabel 3.
Een voorbeeld van een regel uit de procestabel is:

```
root 1 0 80 Nov 16 ? 9:56 /etc/init -
```

TABEL 3

Veld	Betekenis
UID	De naam van de persoon die het commando runt
PID	Het identificatienummer van het proces
PPID	Het proces ID van het ouderproces
C	Geeft een indicatie van de hoeveelheid CPU tijd die het proces op dit moment gebruikt
STIME	Het tijdstip waarop het proces is gestart
TTY	De terminal die gebruikt wordt voor het proces
TIME	De totale CPU-tijd die het proces gebruikt heeft
COMD	Het commando dat gebruikt werd om het proces op te starten

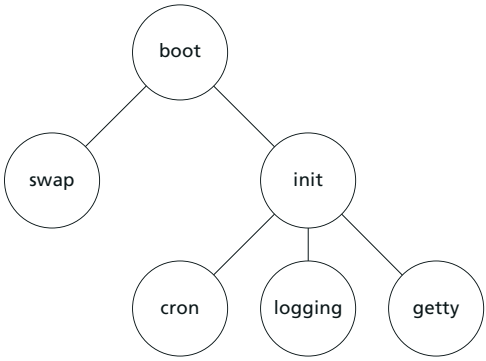
Gegevens in de procestabel ps output (SystemV).

Naast processen die draaien onder 'gewone' user-id's, waaraan geen speciale privileges verbonden zijn, kent Unix-processen die draaien onder de user-id 0; dit user-id wordt veelal aangeduid als de root of de superuser. Deze processen hebben de hoogste privileges: ze zijn geautoriseerd om willekeurige gebieden in het geheugen van het systeem te benaderen, mogen alle system calls aanroepen en zijn vrijgesteld van de toegangsbeveiliging van het file system.
Unix is een pre-emptief besturingssysteem met gescheiden adresruimten. Unix stelt aan de processen slots van processortijd beschikbaar; bij de wisseling van elk slot forceert Unix dat een gebruikersproces tijdelijk wordt onderbroken en een volgend proces de beschikking krijgt over de processor. Daarbij is het voor een proces zonder de daartoe benodigde privileges niet mogelijk het geheugengebied van een ander proces te benaderen.
Een nieuw proces wordt gecreëerd door een al bestaand proces. Het nieuwe proces 'erft' hierbij een aantal eigenschappen van het oude proces, onder andere user-id, group-id en tty. Het PPID van het nieuwe proces is het PID van het oude proces.

System boot Bij het opstarten van de Unix-omgeving (*boot*) wordt een aantal essentiële processen geactiveerd. Dit zijn processen die noodzakelijk zijn voor de werking van het besturingssysteem alsmede processen noodzakelijk voor de werking van de specifieke informatiesystemen. De kernel fungeert als inactieve servicecomponent die wordt aangeroepen door processen.

Het hele start- en stopproces van Unix wordt gecontroleerd door het initproces. Dit is de enige manier waarop in Unix, buiten de automatisch gestarte essentiële processen, initieel processen kunnen worden gestart. Alle Unix-processen zijn uiteindelijk kinderen van init.
Het Unix-bootproces is nogal systeemafhankelijk. Daarom geldt het onderstaand beschreven proces niet voor alle Unix-systemen. Bij het opstarten wordt van de boot-disk het eerste blok (block 0) gelezen. Dit boot-block bevat de software die noodzakelijk is om het besturingssysteem in de lucht te krijgen. Er zijn ook Unix-versies waarbij het boot-block in ROM is opgeslagen.
Bij de system boot kan automatisch een aantal processen worden gestart. Deze processen worden door de systeembeheerder gedefinieerd, bijvoorbeeld bij oudere Unix-systemen in het bestand `/etc/rc`. Vaak zijn dit processen die altijd moeten draaien om de functionaliteit van het systeem en de applicaties te ondersteunen Op moderne Unix-systemen bestaat geen `/etc/rc`-bestand meer. Dit bestand is daar opgesplitst over meerdere bestanden in meerdere directory's. Voor een aantal ingewikkeldere applicaties is het nodig dat er een achtergrondproces (deamon) draait, dat een deel van de functionaliteit uitvoert (bijvoorbeeld een database server).
Gebruikers starten dan een gewoon proces (in de voorgrond) dat communiceert met de deamon om de gewenste acties te laten uitvoeren. Een applicatie kan bestaan uit één proces dat door een gebruiker vanuit een shell wordt geactiveerd, maar ook uit meerdere processen, waarbij één proces bijvoorbeeld een achtergrondproces is dat het feitelijke beheer over een database voert, en andere processen die door de afzonderlijke gebruikers worden geactiveerd. Vervolgens kunnen de gebruikersprocessen communiceren met de daemon die de database beheert om deze te raadplegen of te muteren.

FIGUUR 2



Processen tijdens boot.

In figuur 2 wordt het bootproces weergegeven. Uiteindelijk zijn alle processen gelijktijdig actief en daarmee resident aanwezig. Na de system boot is een Unix-omgeving gereed voor (interactief) gebruik, dat begint met inloggen. Er zijn verschillende manieren om op een Unix-systeem in te loggen. De twee belangrijkste worden hieronder beschreven: via een (asynchrone) terminal en via het netwerk.

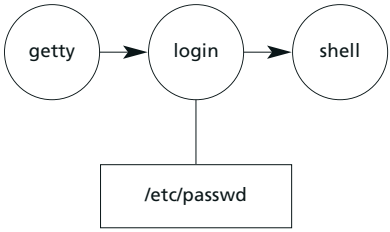
2.1.2 Toegang via (asynchrone) terminals

Na de system boot wordt voor elke actieve terminal het proces `getty` geactiveerd. Vanaf dat moment is het voor gebruikers mogelijk in te loggen en eigen processen op te starten. Op dat moment zijn alle in figuur 2 aangegeven processen actief.
Via het proces `getty` wordt een loginscherm weergegeven. De gebruiker voert zijn username in, waarna het proces `login` wordt opgestart, dat vraagt om het bijbehorende wachtwoord. De gebruiker voert vervolgens het wachtwoord in dat door het loginproces wordt geverifieerd: het wachtwoord wordt versleuteld en vergeleken met het versleutelde wachtwoord zoals dat opgeslagen is in het passwordbestand. Komen beide overeen, dan is de gebruiker 'geauthenticeerd'. Voor de versleuteling wordt gebruikgemaakt van een variant op het DES-algoritme. De versleuteling is zodanig dat gelijke wachtwoorden normaliter een verschillende versleuteling geven.
Het passwordbestand kan door elke gebruiker worden gelezen. Met andere woorden, de versleutelde wachtwoorden zijn openbaar. Deze eigenschap maakt een brute-forceaanval mogelijk: wachtwoorden die in een bepaalde woordenlijst voorkomen, kunnen eenvoudig achterhaald worden door de versleutelde wachtwoorden in het passwordbestand te vergelijken met de versleutelde woorden uit de woordenlijst. Daarom worden wachtwoorden bij veel Unix-versies in een 'shadow password file' geplaatst. In tegenstelling tot het oorspronkelijke passwordbestand is dit schaduwbestand niet algemeen leesbaar.

Als de authenticatie succesvol verloopt, zal `login` de in `/etc/passwd` vastgelegde login shell van de inloggende gebruiker opstarten (zie figuur 3). Doorgaans is de login shell letterlijk een `shell`, met andere woorden: een programmeerbaar, interpreterend commando-interface tussen de gebruiker en de computer. In dat geval kan een bijbehorende profile worden uitgevoerd om gebruikersspecifieke parameters in te stellen. Voorbeelden van zulke profiles zijn `~/.profile`, `~/.login` en `~/.cshrc`; het karakter '~' staat hierbij voor de directory van de gebruiker. In plaats van een `shell` kan ook een menuprogramma, een grafisch interface of een specifieke applicatie

worden opgestart. Vanuit het principe beperking van functionaliteit is het niet wenselijk dat gewone gebruikers een shell tot hun beschikking krijgen.
De processen `getty` en `login` worden nog opgestart onder *user-id 0* (*root*). Na een succesvolle login, start `login` een *shell* voor de inloggende gebruiker op. Hierbij wordt de *user-id* van de inloggende gebruiker - gedefinieerd in het passwordbestand - aan het opstartende proces gekoppeld.

FIGUUR 3



Processen bij inloggen.

Solaris: Asynchrone terminals worden in Solaris gecontroleerd door ttymon, een proces voor alle terminals plus een voor het console device. Ttymon leest de gebruikersnaam van de terminal. Optioneel kan de gebruiker een aantal evironmentvariabelen invoeren achter zijn naam, die dan door login worden geëxporteerd naar de login shell van de gebruiker. Zo kan de gebruiker bijvoorbeeld een ander terminaltype specificeren.

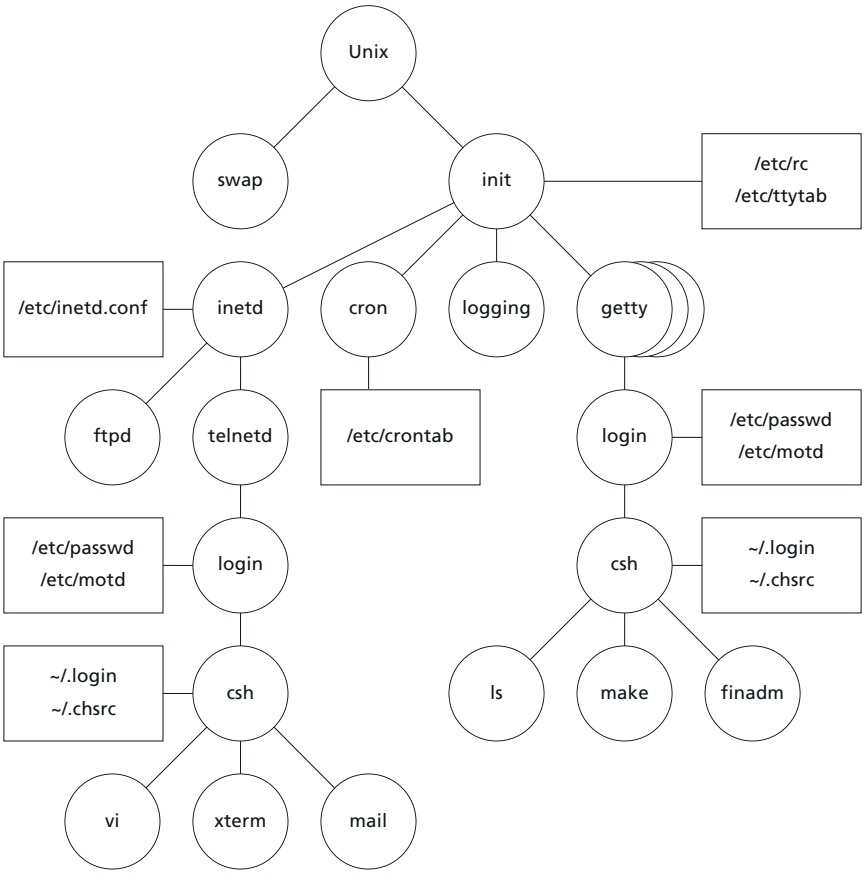
2.1.3 Toegang via het netwerk

In veel Unix-omgevingen wordt bij de system boot een aantal netwerkapplicaties opgestart. Deze applicaties maken het gebruik van communicatienetwerken en protocollen mogelijk. De TCP/IP-protocolfamilie is hierbij verreweg de meest gebruikte. Deze familie omvat naast een aantal essentiële ondersteunende communicatieprotocollen ook een groot aantal standaard-netwerkapplicatieprotocollen. Sommige van deze netwerkapplicaties maken het mogelijk om van afstand op een Unix-omgeving in te loggen (`telnet`, `rsh`, `rlogin`). Andere applicaties maken het bijvoorbeeld mogelijk e-mail te versturen (smtp) of bestanden te transporteren (ftp). Deze netwerkapplicaties komen nader aan de orde in hoofdstuk 6.
Vroeger werden netwerkprocessen tijdens de system boot opgestart

vanuit `/etc/rc`. Tegenwoordig worden de meeste netwerkprocessen, afhankelijk van de specifieke Unix en het soort proces, aangestuurd vanuit een enkel proces, `inetd`. Dit proces 'luistert' naar binnenkomende verbindingen en zorgt ervoor dat de bijbehorende processen worden opgestart. De koppeling tussen het applicatieproces en het netwerkadres van dit applicatieproces (het poortnummer) wordt door de systeembeheerder gedefinieerd, veelal in `/etc/inetd.conf`. Webservers draaien overigens vrijwel nooit via inetd.

Bij toegang via het netwerk is de wijze van authenticatie afhankelijk van de specifieke netwerkapplicatie. Zo maakt telnet gebruik van het loginproces, maar biedt rsh de mogelijkheid om deze vorm van authenticatie te omzeilen.

FIGUUR 4



Voorbeeld van een processtructuur in een Unix-omgeving.

Na het opstarten van het Unix-besturingssysteem, het inloggen van gebruikers en het opstarten van applicaties is dus een groot aantal processen actief. Deze processen worden door het besturingssysteem gerangschikt in een boomstructuur. Figuur 4 geeft een voorbeeld van enkele processen in een operationele Unix-omgeving.

HP-UX: de mogelijkheid is aanwezig om een bestand aan te maken waarmee de toegang via het netwerk verfijnd kan worden aangegeven: `/var/adm/inetd.sec`. Indien dit bestand niet aanwezig is, gedraagt het systeem zich standaard, waarbij de toegang wordt geregeld door de servers. Indien dit bestand wel aanwezig is, dan dienen de regels hierin op de volgende wijze te zijn opgebouwd: <service name> <allow|deny> <host/net addresses, host/net names>. Hierbij is de <service name> gelijk aan de service uit /etc/services of /etc/rpc, terwijl <allow|deny> voor zich spreekt. De rest van de regel kan bestaan uit een enkel systeem of een reeks systemen. Ook wildcards zijn toegestaan. Hiermee kan een extra beveiligingslaag worden opgebouwd binnen de inetdeamon.

2.2 BESTANDEN EN DIRECTORY'S

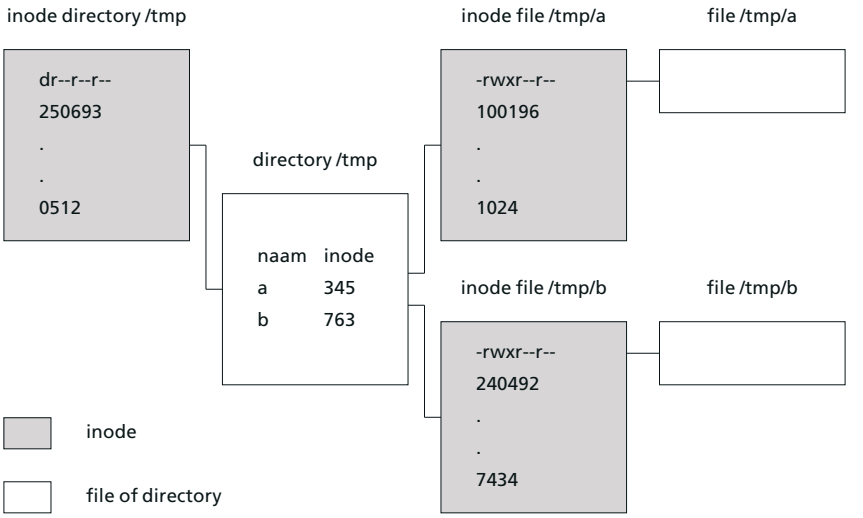
Objecten komen in Unix overeen met bestanden en directory's. Bestanden zijn gerangschikt in een hiërarchisch file system, dat is opgebouwd uit directory's. Eén specifiek onderdeel van de Unix-kernel, het file-subsystem, verzorgt de aansturing van het file system. Per bestand wordt door het file system een inode bijgehouden. In de inode worden de volgende kenmerken van een bestand opgeslagen:

- de grootte van het bestand in bytes;
- het bestandstype;
- de protectiebits die de toegangspermissies aangeven;
- de user-id van de eigenaar van het bestand;
- de group-id van de groep waartoe het bestand behoort;
- het aantal verwijzingen (links) naar het daadwerkelijk bedoelde bestand;
- de datum en tijd waarop het bestand het laatst gelezen en geschreven is;
- de datum en tijd waarop deze inode het laatst door het systeem is gewijzigd;
- het aantal fysieke blokken in het file system dat door het bestand wordt gebruikt.

De naam van een bestand is niet opgeslagen in de inode, maar in het bestand dat de directory representeert. Op deze wijze kunnen één of meerdere alternatieve namen (aliases) of 'harde links' voor een bestand worden gedefinieerd.

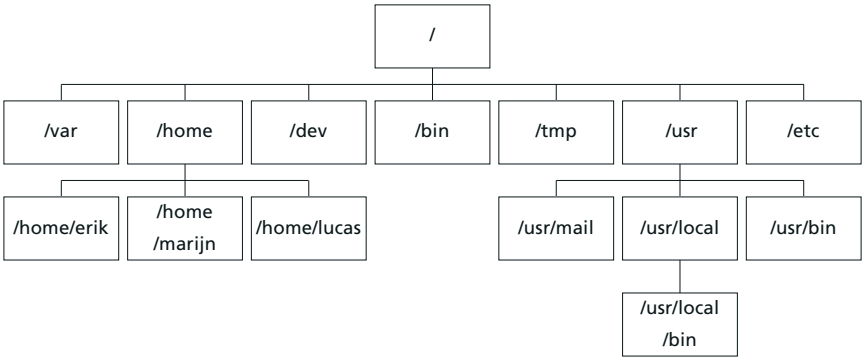
Een directory is een gewoon bestand met daarin een overzicht van de inodes van de bestanden die in de onderliggende directory aanwezig zijn (zie figuur 5). De meeste Unix-omgevingen hebben een standaardindeling van het file system in directory's. Deze indeling is weergegeven in figuur 6 en nader beschreven in tabel 4.

FIGUUR 5



Directory's, bestanden en inodes.

FIGUUR 6



Standaardindeling van het file system.

TABEL 4

/dev	Bevat device bestanden die de fysieke componenten (geheugen, randapparaten) van het systeem representeren. In principe worden bestanden altijd in de directory /dev geplaatst. Deze locatie is slechts 'goed gebruik' en geen technische vereiste
/etc	Bevat configuratiebestanden voor beheeractiviteiten
/sbin	Bevat programma's voor beheeractiviteiten
/bin	Bevat tools en applicatieprogramma's
/usr	Bevat alle gegevens en programma's die noodzakelijk zijn voor de ondersteuning van de gebruiker, maar niet zijn vastgelegd in `etc` en `/bin`.
/tmp	Is een gemeenschappelijke directory die wordt gebruikt voor de opslag van tijdelijke werkbestanden
/var	Bevat logginginformatie en andere variabele bestanden
/home	De plaats van de user directory's

Gebruikelijke invulling van de root directory in een Unix omgeving.

Bestandstypen Unix kent twee hoofdtypen bestanden: gewone bestanden en directory's.
PLAIN FILE (-) Dit is een normaal leesbaar en beschrijfbaar bestand.
DIRECTORY (D) Dit is een normaal leesbaar en schrijfbaar bestand, bedoeld voor de opslag van directory-informatie. Echter, deze bestanden zijn door gebruikers alleen indirect via commando's te veranderen. Dit in tegenstelling tot normale bestanden die rechtstreeks kunnen worden aangepast.

Daarnaast kent Unix links, sockets en pipes:
LINK (L) Is er sprake van een symbolic link naar een inode, dan wordt in een bestand het logische pad naar het daadwerkelijk bedoelde bestand vastgelegd. Bij een harde link wordt in het directory-bestand de verwijzing naar het daadwerkelijk bedoelde bestand aangegeven.
SOCKET (S) Een socket is een datastructuur waarmee informatie van het ene naar het andere proces kan worden doorgegeven. Een socket definieert een eindpunt waarmee het applicatieproces kan communiceren en levert een unieke descriptor terug.
PIPE (P) Een pipe is een first-in-, first-outdatastructuur die wordt gebruikt voor het doorgeven van de uitvoer (stdout) van één proces aan de invoerzijde (stdin) van een ander proces.

Onder Unix worden alle fysieke componenten van het systeem – zoals printers, intern geheugen, swap-geheugen, disks, cartridge readers, printers en modems – gerepresenteerd als bestanden. Er zijn twee typen:
BLOCK DEVICE (B) Een block device is een randapparaat dat gegevens in blokken inleest en wegschrijft, zoals een tape, disk of CD-ROM.

CHARACTER SPECIAL FILE (C) Serieel te benaderen device (per character), zoals een terminal of printer.

Disks Doorgaans is het file system van oudere Unix-systemen opgebouwd uit meerdere file systems, die over meerdere disks en partities op disks zijn verspreid. Tegenwoordig kennen veel Unix-systemen zogenaamde virtual file systems, die over meerdere fysieke disks verspreid kunnen liggen. Unix geeft de mogelijkheid eerst één disk te mounten waarop de root-directory, de Unix-kernel en mogelijk enkele subdirectory's staan opgeslagen. Vervolgens kan binnen het gecreëerde file system met behulp van een mount-point worden aangegeven onder welke directory de additioneel te mounten disk wordt geplaatst. Op deze wijze ontstaat één logisch file system dat in werkelijkheid is opgebouwd uit een verzameling van kleinere file systems.
Het is daarnaast mogelijk om directory's van andere Unix-machines en bijgevolg verschillende file systems te mounten, waardoor gedistribueerde opslag van gegevens een feit wordt (bijvoorbeeld met behulp van Network File System, NFS).

Na het mounten van een file system krijgt een gebruiker de beschikking over de betreffende bestanden en bijbehorende bevoegdheden. Er moeten maatregelen worden genomen om te voorkomen dat een gebruiker een mount uitvoert waarmee hogere privileges worden verkregen dan waar de gebruiker recht op heeft.

Solaris: /bin en /lib zijn symbolische links naar /usr/bin en usr/lib; /usr/lib bevat de meest dynamische libraries. Andere belangrijke directory's zijn /sbin (meerendeels statisch gelinkte programma's voor gebruik tijdens boot en /usr/sbin, programma's voor de system administrator. Op 64-bit Solaris/SPARC bevinden de 64-bit specifieke libraries en executables zich in de 'sparcv9' subdirectory's. Andere belangrijke systeemdirectory's onder Solaris zijn: /devices, /usr/lib, /usr/bin.

HP-UX: de volgende directory's zijn eveneens aanwezig:
- /opt - voor applicaties;
- /etc/opt - configuratiebestanden voor applicaties;
- /stand - kernel, boot loader;
- /lib en /usr/lib - objectcode en bibliotheken (shared- en archived libraries);
- /sbin - essentiele systeemprogramma's (om het systeem te booten en het mounten van file systems);
- /usr/sbin - systeembeheerprogramma's;
- /var/spool - allerlei spool-directory's zoals cron, uucp, mail en lp.

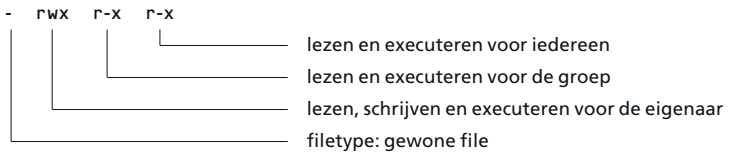
De Logical Volume Manager biedt de mogelijkheid om partities over meerdere schijven te creëren. Elke partitie kan op zichzelf een raw-volume zijn of een file system bevatten.

2.3 TOEGANG VAN PROCESSEN TOT BESTANDEN EN DIRECTORY'S

De toegang van processen tot bestanden wordt onder Unix gerealiseerd door middel van protection bits. Met behulp van deze bits, die specifiek zijn voor elk bestand en iedere directory en die zijn opgeslagen in de corresponderende inodes, wordt aangegeven welke toegangspermissies de eigenaar (user) van het bestand heeft, welke toegangspermissies de leden van de groep (group) van het bestand hebben, en welke toegangspermissies alle andere Unix-gebruikers (other) hebben. Bij deze toegangspermissies wordt onderscheid gemaakt tussen lezen, schrijven en executeren. Naast het zelf gedetailleerd instellen van toegangspermissies ondersteunen veel moderne Unix-varianten tegenwoordig Access Control Lists (ACL's). Omdat ACL's niet gestandaardiseerd zijn bij de verschillende Unix-systemen, kan men er niet zonder meer van uitgaan dat ze werken in een netwerk file system waarin diverse Unix-varianten opgenomen zijn. De protectiebits kunnen worden opgevraagd met het commando `ls -l`; zij worden dan weergegeven door drie groepjes van drie symbolen `r`, `w` en `x` (zie figuur 7).

Het uitvoeren van een bestand kan alleen plaatsvinden wanneer de gebruiker executeerrecht heeft, en als het bestand ofwel een gecompileerd programma is, ofwel een interpreteerbaar script. De beveiliging van directory's is syntactisch identiek, maar semantisch verschillend. Heeft een gebruiker write-permissie op een directory, dan kan hij bestanden in die directory verwijderen, ook al heeft hij

FIGUUR 7



Weergave protectiebits door ls -l (vrij naar [Garfinkel I]).

geen toegang tot de individuele bestanden. Heeft een gebruiker geen execute-permissies voor een directory, dan kan hij de bestanden in deze directory niet benaderen, ook al is hij de eigenaar.
Door het gebruikte mechanisme is de toegangsbeveiliging in Unix minder gedetailleerd dan bij veel andere besturingssystemen, waar de toegangspermissies vaak per subject/objectpaar kunnen worden aangegeven. Bovendien bevinden de toegangspermissies onder Unix zich niet in een centrale database, maar zijn zij verspreid over het gehele file system. Ten slotte leidt de cryptische syntax van de protectiebits ertoe dat het instellen van toegangspermissies een delicate en relatief foutgevoelige operatie is.
Naast de gebruikelijke read-, write- en execute-permissies voor user, group en other kan een drietal speciale permissies worden gedefinieerd. Dit zijn setuid, setgid en het zogenaamde 'sticky bit'. Deze komen in paragraaf 4.3 in detail aan de orde.

3 Gebruikersbevoegdheden

De authenticatie van gebruikers – het identificeren van de gebruiker en het verifiëren of de vastgestelde identiteit de juiste is – is in elk besturingssysteem een essentiële functie. Authenticatie vindt onder Unix net als bij de meeste andere besturingssystemen standaard plaats op basis van gebruikersnamen (usernames) en wachtwoorden. De bevoegdheden van de gebruikers zijn vastgelegd in hun account, die wordt geïdentificeerd via regels in het passwordbestand.
In het bestand `/etc/passwd` wordt de voor beveiliging zo belangrijke relatie gelegd tussen de username en het wachtwoord. Wachtwoorden worden in dit bestand in versleutelde vorm (encryptie) opgeslagen. Een van de redenen voor deze versleuteling is dat veel applicaties vereisen dat het passwordbestand voor iedereen leesbaar is. Door de versleuteling is dit mogelijk, aangezien het bijna onmogelijk is uit het versleutelde wachtwoord het gewone wachtwoord te vinden.
Bij het verifiëren van de identiteit van een gebruiker wordt het volgende gedaan: eerst worden de username en het wachtwoord opgevraagd aan de gebruiker. Het ingetypte wachtwoord wordt niet op het scherm getoond. Dan wordt in het passwordbestand de gebruiker opgezocht en wordt zijn versleutelde wachtwoord opgehaald. Via een algemene routine wordt het ingetypte wachtwoord versleuteld. Dit wordt vergeleken met het opgehaalde wachtwoord. Als al deze stappen goed gaan, is de identiteit van de gebruiker geverifieerd.

3.1 WACHTWOORDEN

In het bestand `/etc/passwd` worden de accounts van het systeem gedefinieerd. Iedere regel definieert één account, met de velden username, password, user-id, group-id, comment, home directory en login shell. Bij het account wordt dus gelijk de voor beveiliging zo belangrijke relatie gelegd tussen de username en het wachtwoord. In de meeste Unix-implementaties bestaan er programma's waarmee gebruikersinstellingen op een eenvoudige manier kunnen worden bijgewerkt; soms is het gebruik nodig. Instellingen die via dit pro-

gramma worden veranderd, voldoen meestal aan de consistentie-eisen die verderop worden gesteld.
Het gebruik van wachtwoorden wordt algemeen beschouwd als de achilleshiel van de toegangsbeveiliging van Unix-systemen. In veel gevallen blijkt een aanzienlijk gedeelte van de wachtwoorden eenvoudig te raden. Daarnaast worden wachtwoorden altijd 'in cleartext' - in leesbare vorm - van het toetsenbord naar de Unix-computer getransporteerd. Het achterhalen van wachtwoorden door het aftappen van de datacommunicatieverbinding is daarom in de meeste Unix-omgevingen een reëel risico.
Omdat de versleutelde wachtwoorden publiekelijk leesbaar zijn, is er een makkelijke aanvalsmethode mogelijk: een hacker versleutelt alle woorden in een uitgebreide woordenlijst en vergelijkt het resultaat met de versleutelde wachtwoorden in het passwordbestand. Om dit risico af te dekken kennen veel Unix-implementaties een 'shadow password file'. De informatie in het oorspronkelijke passwordbestand wordt daarbij meestal in twee delen gesplitst: een gewoon, algemeen toegankelijk passwordbestand zonder wachtwoorden en een volledig afgeschermde shadow password file die de geëncrypte wachtwoorden bevat.
In de shadow password file staan, naast de username en het versleutelde password, nog enkele velden met additionele beveiligingsinformatie. Bijvoorbeeld de informatie over 'password aging' (het geforceerd verlopen van het password). Indien er een shadow password file is, bestaat er ook een programma om de gebruikersinstellingen bij te werken. Vaak is dit geïntegreerd met andere onderhoudsprogramma's voor het systeem.
Wachtwoorden die bekend zijn bij meer personen dan de gebruiker van het account, maken dat het account open staat voor een onbekende groep gebruikers. Hierdoor is het niet meer mogelijk om te identificeren welke persoon via dat account heeft gewerkt. Een ander risico bij een onvoldoende beveiliging is dat willekeurige personen de inhoud van het passwordbestand kunnen wijzigen en nieuwe gebruikers toevoegen.

Solaris: Het versleutelde password staat altijd in `/etc/shadow`. In `/etc/passwd` staat een 'x' in het tweede veld.

3.1.1 Normen en maatregelen

De basisnormen voor het passwordbestand luiden:

- Alleen een bevoegd beheerder dient in staat te zijn de inhoud van het passwordbestand te wijzigen. Daarnaast moet het gewone passwordbestand voor alle applicaties en gebruikers leesbaar zijn.
- Elk wachtwoord mag alleen bekend zijn bij de gebruiker van het desbetreffende account. Het wachtwoord voor root dient uitsluitend bekend te zijn bij bevoegde systeembeheerders.
- Wachtwoorden mogen niet eenvoudig te raden of anderszins te achterhalen zijn. Zwakke wachtwoorden maken het systeem kwetsbaar voor inbraakpogingen. Een 'sterk' wachtwoord bestaat uit ten minste zes karakters, is niet gebaseerd op een bestaand woord of een bestaande eigennaam en bevat naast kleine letters ook lettertekens, cijfers en hoofdletters.
- Wachtwoorden dienen regelmatig veranderd te worden, waarbij oude wachtwoorden niet worden hergebruikt. Dit beperkt de tijd waarin het wachtwoord kan worden gekraakt.
- Het op papier of andere media vastleggen van wachtwoorden door gebruikers dient verboden te worden. Andere gebruikers kunnen die informatie makkelijk vinden.
- Indien mogelijk dient gebruik te worden gemaakt van een shadow password file die niet leesbaar is voor gewone gebruikers.

De bijbehorende basismaatregelen zijn:

- Zorg ervoor dat `/etc/passwd` eigendom van root is.
- Stel de permissies op `/etc/passwd` zodanig in dat alleen root wijzigingen kan aanbrengen, terwijl elke andere gebruiker het bestand kan lezen: zet de permissies op `rw-r—r—`.
- Moedig het gebruik van sterke wachtwoorden aan (awareness) of dwing het indien mogelijk af. Oude wachtwoorden mogen niet worden hergebruikt.
- Eenvoudige manieren om redelijk veilige wachtwoorden te kiezen zijn het gebruik van een bekende tekst zoals een spreekwoord, een liedtekst of een uitdrukking, waarbij de eerste letter van elk woord in het wachtwoord voorkomt, en het kiezen van woorden die zijn samengesteld uit kleinere woorden en één of meer leestekens.
- Laat wachtwoorden verlopen na zes tot dertien weken, zodat daarna niet meer kan worden ingelogd.
- Zet bij tijdelijke accounts de verlooptijd van het wachtwoord minimaal.
- Gebruik, indien mogelijk, een shadow password file.
- Zorg ervoor dat `/etc/passwd` en de shadow password file eigendom van root zijn.

– Stel de permissies op de shadow password file zodanig in dat alleen root wijzigingen kan aanbrengen, terwijl niemand anders het bestand kan lezen.

Als extra maatregel is het mogelijk om de keuze van 'sterke' wachtwoorden af te dwingen met pro-actieve applicaties, zoals `anlpasswd`, `npasswd` en `passwd+`. Daarnaast zijn er diverse commerciële en niet-commerciële tools waarmee de sterkte van de wachtwoorden op een Unix-systeem achteraf kan worden getoetst. Het gebruik van deze tools, die doorgaans gebruikmaken van uitgebreide woordenlijsten en combinatieregels, verdient aanbeveling - mits de authenticiteit, integriteit en toekomstvastheid van deze tools voldoende gewaarborgd is; `crack` is een bekend voorbeeld.
Maar zelfs dan zullen de inherente kwetsbaarheden van authenticatie op basis van wachtwoorden in omgevingen met strenge beveiligingseisen aanvullende authenticatiemaatregelen noodzakelijk maken. De standaardversies van Unix bieden hiertoe echter geen adequate voorzieningen. Wel is een aantal oplossingen in de markt verkrijgbaar. Voorbeelden hiervan zijn pakketten voor wederzijdse authenticatie op basis van cryptografie en systemen voor authenticatie op basis van tokens.

3.2 ACCOUNTS

In Unix moeten de personen die van het systeem gebruik willen maken een account hebben. Een account wordt geïdentificeerd door de username. Direct aan de username is de user-id gekoppeld. De rechten om applicaties op te starten en bestanden te bewerken worden primair via de user-id geregeld. Via de groep, geïdentificeerd door het group-id, kunnen additionele permissies worden geregeld.

3.2.1 Individuele accounts

Naast de user-accounts zijn er nog twee soorten accounts: systeemaccounts en applicatieaccounts. Het verschil tussen al deze accounts ligt alleen in het gebruik. Technisch is er geen verschil. Systeemaccounts worden bij installatie van het operating system gedefinieerd. Ze dienen voor een goede installatie en gebruik van het operating system. Applicatieaccounts zijn administratieve accounts ten behoeve van de applicaties die zijn geïnstalleerd.
Het systeemaccount met user-id 0 heet root; het is het enige speciale account van het systeem: root geniet de hoogst mogelijke privileges.

Processen die draaien onder het rootaccount kunnen onder meer alle geheugenlocaties lezen en schrijven, kunnen elke gebruikersidentiteit aannemen en zijn vrijgesteld van toegangsbeveiliging op het file system. Hoewel het niet verboden is om het account met user-id 0 een andere naam te geven, zullen er toch programma's zijn die hierdoor niet goed werken.
Naast root zijn er onder Unix diverse systeemaccounts aanwezig, met als doel om het eigendom van een groep van gerelateerde bestanden en directory's te regelen (zie tabel 5). Ook het uitvoeren van systeemtaken waar geen rootpermissies voor nodig zijn, valt hieronder (daemon en lp). Met uitzondering van het rootaccount is het niet nodig onder deze accounts in te loggen.

TABEL 5

Systeem *accounts*	Gebruik van het systeemaccount
`root`	superuser: dit account heeft geen beperkingen
`sys`	eigenaar van veel systeembestanden
`bin`	eigenaar van de meeste commando's in `/usr/bin`
`adm`	eigenaar van beheersbestanden in `/var/adm`
`uucp`	eigenaar van beheersbestanden in `/usr/lib/uucp`
`daemon`	account van de systeem-daemon
`lp`	eigenaar van print-bestanden in `/var/spool/lp`

Systeemaccounts.

De useraccounts zijn die accounts waaraan mensen gerelateerd zijn. Deze gebruikers kunnen in het algemeen in verschillende categorieën worden ingedeeld, bijvoorbeeld systeembeheerders, applicatiebeheerders en applicatiegebruikers.
De applicatieaccounts dienen ervoor, zoals de systeemaccounts, om de bestanden van de applicatie te beveiligen en om taken binnen de applicatie met een bekend account uit te voeren.

Solaris: andere systeemaccounts zijn nuucp 9 (account gebruikt voor UUCP) en listen 37 (account voor bepaalde network services).

HP-UX: andere systeemaccounts zijn nuucp 11 (account tevens gebruikt voor UUCP), hpdb 27 (account voor de HP-ALLBASE database) en www 30 (voor het world wide web). Su is te vinden als `/usr/bin/su`.

3.2.2 Normen en maatregelen voor individuele accounts

Als basisnormen voor accounts gelden:

- Zowel usernames als user-id's dienen uniek te zijn. Als user-id's dubbel voorkomen, zijn de bestanden van de ene gebruiker ook van de andere gebruiker. Als usernames dubbel voorkomen, kan de als tweede voorkomende gebruiker niet inloggen.
- Het stelsel van usernames en user-id's dient overeen te komen met de actuele personeelsformatie. Ongebruikte accounts, bijvoorbeeld van ex-medewerkers, dienen gedeactiveerd of verwijderd te worden.
- Voor iedere geïnstalleerde applicatie dient een apart account te worden gebruikt.
- Van ieder account dient ieder veld in het passwordbestand ingevuld te zijn.
- User-id 0 (root) dient uitsluitend in het noodzakelijke geval te worden gebruikt.
- Inloggen als 'root' dient verboden te zijn. Voor reguliere werkzaamheden dient een persoonlijk account te worden gebruikt.

Basismaatregelen zijn:

- Zorg ervoor dat usernames en user-id's uniek zijn.
- Zorg ervoor dat usernames en user-id's één op één overeenkomen met de actuele bemanning van de organisatie.
- Voor iedere applicatie dient een afzonderlijk account te worden gedefinieerd, dat fungeert als eigenaar van de desbetreffende programmatuur en data.
- Verwijder of deactiveer algemene accounts voor gasten.
- Deactiveer ongebruikte accounts of langdurig niet gebruikte accounts, bijvoorbeeld door het plaatsen van '*' in het passwordveld van het passwordbestand.
- Let erop dat de user-id 0 (root) niet gekoppeld is aan een persoon.
- Verbied het direct inloggen als 'root'. Om rootpermissies te verkrijgen dient het gebruik van `/bin/su` verplicht te worden gesteld, welke actie wordt gelogd. Alleen in uitzonderingsgevallen is het toegestaan om in te loggen onder het rootaccount. Hierbij dient per geval autorisatie te worden verkregen en schriftelijk verslag te worden gedaan.
- Zorg ervoor dat ieder account een systeemaccount, een useraccount of een applicatieaccount is.
- Verwijder accounts die na het inloggen een enkel commando uitvoeren, zoals who, date of sync. Verwijder alle bestanden waarvan deze accounts eigenaar zijn. Verwijder ook alle andere referenties naar dit account, bijvoorbeeld bij `cron`. Het beste is

om de accounts compleet te verwijderen. Men zou ook in plaats van de accounts te verwijderen een ongeldige shell (`/bin/false`) kunnen opgeven in het bestand `/etc/passwd`.
- Verwijder het account nuucp. Dit is een account dat voor UUCP (zie hoofdstuk 6) wordt gebruikt.
- Het comment veld in het passwordbestand dient zodanig ingevuld te zijn dat:
  - voor systeemaccounts bijvoorbeeld de identificatie 'system' is ingevuld;
  - voor useraccounts de persoon is geïdentificeerd;
  - voor applicatieaccounts de applicatie en de applicatiebeheerder zijn geïdentificeerd.

Mogelijke extra maatregelen zijn (deze maatregelen kunnen niet gerealiseerd worden met behulp van de standaard Unix-beveiligingsfunctionaliteit):
- Geef ieder soort account zijn eigen interval van user-id's, bijvoorbeeld systeemaccounts van 0 tot 49, systeembeheerders van 50 tot 99, applicatieaccounts van 100 tot 999 en useraccounts vanaf 1000.
- Voor veel omgevingen is het aanbevelenswaardig de systeem- en beheerdersaccounts geheel te blokkeren. De systeembeheerder kan dan slechts via een beheerschil de noodzakelijke taken uitvoeren. Blokkeer in dat geval het rootaccount door het plaatsen van een * in het passwordveld van het rootaccount. Bedenk echter dat in geval van calamiteiten het systeem geboot moet worden vanaf een ander file system waar wel een geldig rootaccount op gedefinieerd is.
- Is functiescheiding op beheerniveau vereist, maak dan gebruik van speciale beheertools, waarmee de bevoegdheden voor het uitvoeren van specifieke beheertaken wordt toewezen aan verschillende functionarissen.
- Maak gebruik van het vier-ogenprincipe door systeemwijzigingen door één functionaris te laten voorbereiden en door een andere functionaris te laten invoeren. Als alternatief kan ervoor worden gekozen om wel wachtwoorden voor root te activeren, maar dan wel twee, te beheren door twee verschillende medewerkers; dit is afhankelijk van de Unix-versie.

3.2.3 Groepen

Onder Unix kunnen accounts worden toegevoegd aan meerdere groepen. Het lidmaatschap van een groep impliceert dat alle accounts in een groep de toegangspermissies van die groep hebben. Naast de informatie over de primaire groep van de accounts, die in `/etc/passwd` wordt vastgelegd, wordt de informatie over het

lidmaatschap van één of meerdere secundaire groepen van de gebruikers vastgelegd in het bestand `/etc/group`. Iedere regel in dit bestand vertegenwoordigt een groep. De regel bestaat uit vier velden die onderling gescheiden worden door het ':'-teken. De betekenis van de velden is weergegeven in tabel 6.

TABEL 6

Naam component	Betekenis
Group name	Naam van de groep
Password	Versleuteld wachtwoord van de groep
Group-id	Group-idnummer
Member(s)	Usernames van de accounts die lid zijn van de groep. De usernames worden van elkaar gescheiden door komma's.

Inhoud van '/etc/group'.

Er is een wezenlijk verschil in het gebruik van groepen tussen System V en 4.xBSD. Onder System V gelden alleen de groepspermissies van de huidige groep. Een account kan wisselen tussen de groepen waarvan hij lid is. De huidige groep van het account wordt bij login gezet op de groep die vermeld staat in `/etc/passwd`. Onder 4.xBSD is het account gelijktijdig lid van alle groepen waarbij het in het `/etc/group` -bestand staat vermeld. Onder 4.xBSD is er tevens de groep wheel, met group-id 0. De leden van deze groep hebben enige systeembeheerpermissies; zij kunnen bijvoorbeeld als enigen het `su` -commando gebruiken om superuser te worden.
In het tweede veld van `/etc/group` kan het wachtwoord van de groep worden vastgelegd. Gebruikers van accounts die niet tot de groep behoren, maar wel het wachtwoord van de groep kennen, wordt op deze wijze de mogelijkheid geboden om over te gaan naar de groep.

HP-UX: het bestand `/etc/logingroup` bevat, indien aanwezig, een gelijke opbouw als /etc/group. Indien dit bestand niet bestaat, dan zorgt /etc/group voor de mogelijkheid om met behulp van het commando newgrp van groep te veranderen. Bestaat /etc/logingroup wel, bijvoorbeeld als link naar /etc/group, dan is de gehele standaard-grouptoegangslijst geldig voor de ingelogde gebruiker. Het gebruik van newgrp is dan niet meer nodig.

3.2.4 Normen en maatregelen voor groepen

Basisnormen voor groepen zijn:
- De indeling in groups dient conform de organisatorische functiescheidingen te zijn.
- Groepen en group-id's dienen uniek te zijn.
- Er mag geen overlap zijn tussen group-id's voor gewone gebruikers en group-id's voor systeemaccounts.

Basismaatregelen voor groepen zijn:
- Zorg ervoor dat iedere group, in verband met de unieke vertaling van group-id naar groupname, een uniek group-id heeft.
- Zorg ervoor dat er geen overlapping is tussen groups met systeembeheerders en groups met gewone gebruikers.
- Definieer voor elke group als wachtwoord een '*', zodat uitsluitend de groepsleden toegang hebben tot de group.

Een mogelijke extra maatregel is: definieer in het geheel geen extra groepen.

3.3 PLUGGABLE AUTHENTICATION MODULES

Het authenticeren van gebruikers op een systeem gebeurt op verschillende wijzen. Het eerste voorbeeld is het standaard Unix-loginprogramma, waardoor authenticatie plaatsvindt door het toetsen van het bij een aanloggende accountnaam ingevoerd wachtwoord aan het in het wachtwoordenbestand bij dat account (in versleutelde vorm) vastgelegde wachtwoord. Ook applicaties kunnen zelf aanvullende authenticatie uitvoeren. Wanneer andere vormen van authenticatie gewenst zijn, zoals authenticatie met behulp van smart cards, dienen de authenticerende programma's opnieuw gecompileerd te worden om deze nieuwe diensten te kunnen bieden.
Een oplossing voor dit probleem is het toepassen van Pluggable Authentication Modules (PAM), een binnen OSF's DCE gedocumenteerde aanvulling (DCE-RFC 86.0). Bij wijzigen van de gewenste authenticatiemethodiek volstaat het om de nieuwe modules te activeren door het wijzigen van een configuratiebestand. Dit configuratiebestand bestaat uit regels die bijvoorbeeld aangeven welke authenticatiemodule voor login moet worden uitgevoerd.

Binnen PAM zijn vier verschillende soorten modules gedefinieerd:
- AUTH de daadwerkelijke authenticatie door middel van bijvoorbeeld wachtwoord, het verstrekken van eventuele Kerberos-tickets;

- ACCOUNT het bepalen of de authenticatie toegestaan is, bijvoorbeeld het verlopen zijn van een account en loginrestricties;
- PASSWORD het instellen van wachtwoorden;
- SESSION het inrichten van de gebruikersomgeving, bijvoorbeeld het mounten van een homedirectory voor de gebruiker.

Het is mogelijk om verschillende modules achtereenvolgens aan te roepen. De modules kunnen op verschillende wijze worden aangeroepen:
- REQUIRED de overige regels in het configuratiebestand worden ook doorlopen;
- REQUISITE bij niet voldoen aan deze regel wordt geen toegang verleend;
- SUFFICIENT het voldoen aan deze regel verleent toegang;
- OPTIONAL deze regel bepaalt niet het resultaat.

Wanneer de volgorde van de regels niet juist is ingesteld, bestaat onvoldoende zekerheid met betrekking tot de authenticatie.
Een voorbeeld van de toepassing van PAM is vervanging van het standaard-loginprogramma door een PAM-module die controleert op het gebruik van sterke wachtwoorden.

Solaris: PAM wordt in Solaris geconfigureerd dmv /etc/pam.conf; de loadable objects staan in `/usr/lib/security` *en* `/etc/lib` *(in geval* `/usr` *niet gemount kan worden).*

HP-UX: PAM is systeembreed geconfigureerd in het bestand `/etc/pam.con`*, of in* `pam user.conf` *voor een enkele gebruiker. De standaard meegeleverde bibliotheek is* `/usr/lib/security/libpam unix.1` *met de mogelijkheid om gebruik te maken van het standaardwachtwoordmechanisme, DCE of een smartcard.*

3.3.1 Normen en maatregelen De basisnorm voor het gebruik van PAM is:
- Gebruik van PAM mag geen verlaging van het beveiligingsniveau teweegbrengen.

Basismaatregelen zijn:
- Implementeer PAM alleen als uitgebreidere authenticatiemethodes worden gebruikt.
- Beveilig elk configuratiebestand in de directory `/etc/pam.d` met de permissie `rw-r--r--` (zie paragraaf 2.3 en 4.2 voor de betekenis van de permissies).

3.4 INRICHTING VAN DE GEBRUIKERSOMGEVING

Als gebruikers volgens de hierboven aangegeven richtlijnen zijn aangemaakt, moet de werkomgeving voor de gebruikers nog worden ingericht. Dit omvat de volgende punten:

- Login shell: het programma dat na de loginprocedure als eerste wordt opgestart.
- Home directory: de directory waar de gebruiker zijn eigen bestanden mag neerzetten.
- Umask: welke permissies toegekend worden aan nieuwe bestanden.
- Path variabele: welke programma's gevonden worden zonder het pad op te geven.

3.4.1 Login shell

Een gebruiker krijgt na inloggen een commando-, menu- of applicatieomgeving tot zijn beschikking via het login shell veld dat voor elke individuele gebruiker gedefinieerd is in `/etc/passwd`. Indien het veld leeg is, wordt een commando-omgeving (shell) via `/bin/sh` opgestart.

Standaard kan de gebruiker, mits deze beschikt over een shell, met `chsh` de eigen loginshell wijzigen. `chsh` laat slechts toe dat programma's die zijn opgenomen in `/etc/shells` kunnen worden geselecteerd. Uiteraard kan een gebruiker vanuit de huidige actieve shell een willekeurige andere shell opstarten.

De diverse shells (`sh`, `csh`, `ksh`) hebben ieder een configuratiebestand (shellrc) in de home directory van de gebruiker. Dit bestand wordt geïnterpreteerd wanneer de shell wordt opgestart. Indien de shell een login shell is, wordt ook een profile geïnterpreteerd. Met deze bestanden kan de gebruiker zijn commando-omgeving verder inrichten. Hij kan echter ook door de systeembeheerder gedefinieerde variabelen voor zichzelf veranderen. De systeembeheerder kan dan ook niet verhinderen dat een gebruiker met een shell buiten de gewenste omgeving kan komen. Met een volledige commando-shell is het mogelijk veel kennis van het systeem te vergaren en veel systeemresources te gebruiken (wat kan leiden tot een denial of services attack).

Met een restricted shell wordt bewerkstelligd dat aan een gebruiker wel een commando-shell ter beschikking wordt gesteld, maar dat de mogelijkheden sterk worden beperkt: het commando `cd` is niet beschikbaar, waardoor de gebruiker niet van directory kan veranderen; bij het aanduiden van bestanden kan er geen pad worden gebruikt waarin een '/' staat; de uitvoer van programmatuur kan niet naar andere bestanden 'geredirect' worden; de PATH variabele kan niet worden gewijzigd.

Een extra beveiliging kan met het commando `chroot` worden bewerkstelligd: de top van het file system ('/') wordt gewijzigd naar bijvoorbeeld de home directory van de gebruiker. Hierdoor wordt voorkomen dat de gebruiker door het file system kan zoeken, zelfs als hij erin geslaagd is een volwaardige shell op te starten. In dit geval dienen wel aanvullende maatregelen te worden getroffen, zoals het aanbrengen van specifieke etc en bin directory's in de nieuwe top van het file system. Met een shell met beperkte bevoegdheden is het, tenzij de mogelijkheden alsnog sterk zijn ingeperkt, meestal mogelijk om enige kennis van het systeem te vergaren of om een denial of services attack uit te voeren.

HP-UX: indien het bestand `/etc/shells` niet bestaat, dan zijn de volgende shells standaard te gebruiken: `/sbin/sh`, `/usr/bin/sh`, `/usr/bin/rsh`, `/usr/bin/ksh`, `/usr/bin/rksh`, `/usr/bin/csh`, `/usr/bin/keysh`.

3.4.2 Normen en maatregelen login shell

Basisnormen voor de login shell zijn:

- Toegang tot applicaties dient gebaseerd te zijn op het principe van beperking van functionaliteit. In het bijzonder dienen gebruikers geen toegang te hebben tot commando-shells of andere programmeerfaciliteiten.
- Om te voorkomen dat gebruikers een commando-omgeving beschikbaar krijgen, dient gebruik te worden gemaakt van menu's. Als alternatief kan ervoor worden gekozen na het inloggen automatisch de applicatie te laten opstarten, waardoor gebruikers geen commando-shell buiten de applicatie om kunnen krijgen.
- Geef gebruikers geen restricted shell. De beperkte beveiliging vereist veel beheer om de gebruiker te beletten om zijn bevoegdheden uit te breiden.

Hieruit volgen deze basismaatregelen:

- Specificeer als login shell in `/etc/passwd` geen commando-shell. Geef in plaats daarvan een specifieke applicatie als login shell op.
- Verpak applicaties in shell scripts waarin als shell-variabele `/dev/null` is gedefinieerd, om tegen te gaan dat gebruikers vanuit de applicatie een commando-shell kunnen bemachtigen.
- Stel op productieomgevingen uitsluitend aan systeembeheerders een commando-shell ter beschikking. Uitzonderingsgevallen dienen door de gebruiker gemotiveerd, door het management

geaccordeerd en door de systeembeheerder gedocumenteerd te worden.
- In `/etc/shells` mogen alleen geautoriseerde shells zijn opgenomen om te voorkomen dat een gebruiker zijn eventueel toegekende commando-shell inwisselt voor een andere shell.
- Gebruik geen restricted shells. Uitzonderingsgevallen dienen gemotiveerd, geaccordeerd en gedocumenteerd te worden. Bovendien gelden daarbij de volgende eisen:
  - maak een zorgvuldige selectie van toegestane commando's en test deze;
  - maak gebruik van het `chroot` -commando;
  - maak de volgende directory's aan onder de home-directory van de gebruiker:
    - `bin`, met gebruikelijke tools als `sh, stty, ls, ed, pwd, cat`;
    - `dev` met de te gebruiken devices;
    - `etc` met daarin de bestanden `passwd, group, profile, utmp, wtmp`;
- definieer in `etc/passwd` alleen user-id's voor de systeembeheerder en voor de chroot-gebruiker zelf; definieer de shell-parameters (.profile) zodanig dat alleen toegestane commando's kunnen worden gegeven. Deze commando's dienen geen mogelijkheid te bieden om bijvoorbeeld een nieuwe commando-shell te activeren.

3.4.3 Home directory

De home directory wordt gedefinieerd in het passwordbestand. De home directory is de privé-directory van de gebruiker. Door het loginproces wordt de omgevingsvariabele HOME op deze directory gezet. De login shell start in deze directory. Diverse programma's gebruiken standaard deze directory om persoonlijke configuratiebestanden neer te zetten.
Indien meerdere gebruikers dezelfde home directory hebben, gebruiken ze dezelfde configuraties en delen ze alle bestanden. Hierbij ontstaan problemen bij het lezen van privé-bestanden en het schrijven van bijvoorbeeld configuratiebestanden.
Indien de home directory niet bestaat of indien de gebruiker geen toegang heeft tot zijn home directory, start de login shell in /.

3.4.4 Normen en maatregelen home directory

De basisnormen voor de home directory luiden:
- Ieder user-account heeft een eigen home directory.
- Ieder applicatieaccount heeft een eigen home directory waarin de bestanden opgeslagen zijn ten behoeve van die applicatie.
- Het systeemaccount root heeft een eigen home directory die op de /-partitie ligt. De overige systeem-accounts hebben als home

directory een directory op de /-partitie. Deze home directory's hoeven niet uniek te zijn.

Basismaatregelen ten aanzien van home directory's zijn:

- Ieder user-account heeft een eigen home directory die als laatste element van het pad de usernaam heeft. Indien de organisatie niet te groot is, kan ieder van deze home directory's een directory op `/home` zijn. Voor organisaties met veel medewerkers kunnen de home directory's gegroepeerd zijn op bijvoorbeeld de eerste letter van de gebruikersnaam of op de groep waartoe de gebruiker behoort.
- Root heeft een eigen home directory op de /-partitie. Zet de home directory van root op `/root` om ervoor te zorgen dat er geen configuratiebestanden in /zelf staan.
- Alle home directory's van user-accounts en van root hebben permissie 700, dat wil zeggen de eigenaar heeft alle permissies, de overige accounts hebben geen rechten.
- Ieder applicatieaccount heeft zijn eigen home directory. Standaard wordt hiervoor in `/opt` de applicatienaam genomen. Hierin staan alle bestanden die bij de applicatie horen. De permissies op de home directory overstijgen in geen geval de algemene permissies die voor het hele file system gelden. Iedere uitzondering hierop (setuid-programma's, wereldschrijfbare directory's) moet gedocumenteerd zijn.

3.4.5 Umask

Bij het aanmaken van bestanden en directory's krijgen deze een permissie die door het creërende programma wordt bepaald. Unix kent de mogelijkheid om aan te geven dat bepaalde permissies niet worden toegekend bij het creëren van bestanden en directory's. Alleen door later expliciet de permissies te veranderen kunnen de oorspronkelijk weggenomen permissies weer worden toegevoegd. De parameter die bepaalt welke permissies worden weggenomen heet umask; dit is een octale waarde.
De default-umaskwaarde wordt ingesteld in het bestand `/etc/profile`. Deze waarde kan worden veranderd via het commando `umask`. In de meeste Unix-versies heeft de default-umask waarde 0022, waardoor zowel de groep als de overige gebruikers schrijfpermissies wordt ontnomen. De instelling 0077 biedt een betere bescherming, omdat dan alleen de eigenaar bij het aanmaken van een bestand of directory permissies voor read, write en execute krijgt. Dit betekent dat niets voor derden is toegestaan, tenzij de eigenaar van de bestanden expliciet toestemming geeft.

3.4.6 Normen en maatregelen umask

Basisnorm voor umask is:

– De umask dient ingesteld te zijn volgens het principe van veilige beginwaarden. Dit houdt in dat een bestand uitsluitend toegankelijk is voor de eigenaar, tenzij expliciet anderszins is aangegeven.

Als basismaatregel geldt:

– Zet umask in `/etc/profile` op 0077: alleen de eigenaar houdt lees-, schrijf- en executiepermissies. Check dat deze waarde in de gebruikersprofiles niet wordt gewijzigd.

3.4.7 Path variabele

Het path is een verzameling directory's die is vastgelegd in de PATH variabele. Als een gebruiker een programma aanroept zonder een pad aan te geven, doorzoekt het besturingssysteem de gespecificeerde directory's in het path naar dit programma. De waarde van het path wordt vastgelegd op twee plaatsen. De waarde voor path in de `/etc/profile` (voor sh en ksh) respectievelijk `/etc/csh.login` (voor csh) specificeert het default path voor Unix. Dit path kan overschreven worden door een persoonlijke waarde, bijvoorbeeld in de persoonlijke shell profile.
Als een directory die voorkomt in het path beschreven kan worden door onbevoegden, dan kan een kwaadwillende gebruiker in zo'n directory een 'Trojaans paard' plaatsen. Bij het aanroepen van een programma wordt daarbij niet het beoogde programma, maar het Trojaanse paard uitgevoerd - met alle permissies van degene die het programma aanroept.
Indien het commando `su` wordt gebruikt om te kunnen inloggen als een andere gebruiker, verandert het path niet. Dit is alleen het geval als via su wordt ingelogd met de optie '-'.

3.4.8 Normen en maatregelen path variabele

Basisnormen voor het gebruik van de path variabele zijn:

– Het path mag geen directory's bevatten die schrijfbaar zijn voor derden. Met name het opnemen van '.' (de huidige directory) in het path kan tot gevolg hebben dat programma's in de huidige directory kunnen worden geactiveerd. In dat geval kan men de systeembeheerder naar een algemeen schrijfbare directory 'lokken' en daar zonder dat hij het beseft een van te voren geplaatst Trojaans paard laten uitvoeren.
– Alle commando's die automatisch uitgevoerd worden, dienen met volledige path-vermelding te worden opgenomen, om er zeker van te zijn dat de juiste commando's worden aangeroepen.
– Het path dient juist gedefinieerd te zijn. Is dat niet het geval, dan kunnen programma's niet gevonden worden indien ze worden aangeroepen

Basismaatregelen die uit deze normen voortkomen zijn:

- Het standaard path voor de eindgebruikers en het path dat gebruikt wordt door beheerders mogen geen directory's bevatten die schrijfbaar zijn voor derden.
- De huidige directory '.' mag nooit in het path van root worden opgenomen.
- Gebruik als root altijd absolute padnamen (zoals `/bin/su, /bin/find, /bin/passwd`) om Trojaanse paarden te voorkomen. Gebruik './' om programma's in de huidige directory op te starten. De systeemdirectory's (zoals `/bin` en `/usr/bin`) dienen in het begin van het path gezet te worden, zodat deze eerst worden doorzocht.
- Start als root nooit een onbekend programma.
- Systeembeheerders die onder root werken, wordt geadviseerd om de volledige directory aan te geven bij het aanroepen van commando's, in plaats van alleen het commando.
- Gebruikers mogen niet zelf hun path wijzigen.

Als extra maatregel kan ervoor worden gekozen in het path geen enkele directory op te nemen. Programma's of bestanden dienen dan bij het aanroepen met de volledige directory te worden aangegeven.

4 Objecten

Met objecten worden in de context van Unix bestanden en directory's bedoeld. Het inrichten van het file system bestaat uit het plaatsen van alle benodigde programmatuur en databestanden op de harde schijf van het systeem en het instellen van de gewenste toegangspermissies op de directory's en bestanden.
In dit hoofdstuk komen achtereenvolgens aan de orde de inrichting van het file system, het instellen van veilige toegangspermissies op alle directory's en bestanden en de speciale permissies van setuid, setgid-bit en sticky bit.

4.1 INRICHTING VAN HET FILE SYSTEM

Het inrichten van een machine begint bij het installeren van het file system. Dit bestaat uit het partitioneren van de harde schijf, het installeren van het operating system en de gewenste applicaties en het verder configureren van de machine (de gebruikers, de services).
De indeling van het file system is geschetst in hoofdstuk 2. Bij de modernere inrichtingen wordt door de fabrikant van Unix al een duidelijke scheiding aangebracht, enerzijds tussen statische en dynamische bestanden, anderzijds tussen machine-eigen bestanden en met meerdere machines deelbare bestanden. Het is aan te bevelen om deze verdeling ook voor de overige applicaties te gebruiken.
Voordat het operating system wordt ingericht, moet eerst bepaald zijn hoe de harde schijf (schijven) wordt gepartitioneerd. De door de fabrikant voorgestelde verdeling van de operating-systembestanden over de partities is gewoonlijk een goed uitgangspunt. Ook kan de File Hierarchy Standard worden gevolgd [Quinlan]. Voor de te installeren applicaties kunnen dan één of meerdere aparte partities worden aangemaakt.
Bij de inrichting moet ook worden bepaald of delen van het file system via NFS (Networked File System) worden geïmporteerd of geëxporteerd. Hierbij kunnen gehele partities aangeboden worden aan andere machines om te gebruiken. NFS wordt verder besproken in hoofdstuk 6.
Nadat de harde schijf in partities is ingedeeld, kunnen de bestanden

van het operating system en de applicaties worden geïnstalleerd. Dit wordt gewoonlijk gedaan met het standaard-installatieprogramma van de fabrikant. Tijdens de installatie van het operating system worden ook de devicebestanden aangemaakt.
Iedere partitie, uitgezonderd /, wordt gewoonlijk tijdens het opstarten gemount. Is dat niet het geval, dan kan het gebeuren dat bij het opstarten één of meer bestanden of programma's niet worden gevonden. Basisprogrammatuur werkt niet meer naar behoren indien `/tmp` of `/var` vol is. Met name logbestanden en tijdelijke bestanden kunnen dan niet worden aangemaakt of uitgebreid.

4.1.1 Normen en maatregelen

Basisnormen zijn:

- Alle partities worden gemount.
- Op systeempartities mogen geen applicatiebestanden staan.

Basismaatregelen zijn:

- Maak partities aan voor de operating-systempartitie(s) en voor de applicatie(s). Geef iedere partitie voldoende grootte. Zorg ervoor dat `/tmp` en `/var` voldoende vrije ruimte hebben. Zorg ervoor dat de overige systeempartities voldoende vrije ruimte hebben om patches te installeren.
- Alle applicaties dienen op `/opt` te worden geïnstalleerd. Applicaties die veel ruimte vragen, kunnen op een nieuwe partitie onder `/opt` worden geïnstalleerd. Indien de applicatie systeemdirectory's gebruikt, leg dan een symbolische link vanuit deze systeemdirectory naar de applicatie partitie. Hierdoor wordt verzekerd dat de applicatie geen operating-systempartitie volschrijft.
- Zorg ervoor dat alle partities bij het opstarten worden gemount. Zorg ervoor dat de volgorde van mounten zodanig is, dat partities die op een andere gemounte partitie worden gemount, ook later gemount worden.

4.2 TOEGANGSPERMISSIES

Unix biedt een primitieve vorm van discretionary access control, waarbij de eigenaar van een bestand of directory zelf de toegangspermissies bepaalt. Hierbij wordt onderscheid gemaakt tussen lezen (read, r), schrijven (write, w) en executeren (execute, x). Deze instellingen zijn bij Unix beperkt tot de volgende categorieën gebruikers: user (de eigenaar zelf), group (de groep waartoe het bestand behoort) en other (alle andere gebruikers van het systeem). De 'rwx-permissies' worden per categorie aangegeven met een octaal getal, waarbij

iedere combinatie van permissies een eigen cijfercombinatie heeft. Naast de 'rwx-permissies' bestaan er het setuid-bit, het setgid-bit en het sticky bit. Deze worden in paragraaf 4.3 behandeld.
Heeft een gebruiker write-permissie op een directory, dan kan hij bestanden in die directory verwijderen, ook al heeft hij geen toegang tot de individuele bestanden. Heeft een gebruiker geen execute-permissie voor een directory, dan kan hij de bestanden in deze directory in het geheel niet benaderen, ook al is hij de eigenaar. Heeft een gebruiker geen read-permissie voor een directory, dan kan hij niet opvragen welke bestanden in de directory staan. Als hij echter de naam van een bestand weet, kan hij die wel bewerken, wanneer de overige permissies hem dat tenminste toestaan.
De toegang tot applicaties wordt onder Unix primair geregeld door middel van bestandspermissies. De execute-permissies van programma's dienen daarom in overeenstemming te zijn met de organisatorische functiescheiding. Daarnaast zijn ten aanzien van programmatuur aanvullende maatregelen noodzakelijk, die in hoofdstuk 3 reeds aan de orde kwamen.

Indien directory's of bestanden meer permissies geven aan meer accounts dan nodig, dan kunnen één of meer bestanden worden veranderd door personen die daartoe niet bevoegd zijn. Ook kunnen personen die daartoe niet bevoegd zijn meer informatie opvragen dan waartoe ze gerechtigd zijn.
Als directory's of bestanden daarentegen te weinig permissies geven aan accounts die hen gebruiken, dan kunnen die accounts niet naar behoren werken.

HP-UX: het zogenaamde HFS file system biedt de mogelijkheid tot het gebruik van ACL's. Als back-up dient het programma `fbackup` te worden gebruikt die dit meeneemt (en `frestore` om de data terug te halen).

4.2.1 Normen en maatregelen

Basisnormen ten aanzien van toegangspermissies zijn:

- De instellingen van de Unix-bestandspermissies dienen in overeenstemming te zijn met de functiescheiding zoals deze in de organisatie is vastgelegd. Is dat niet het geval, dan zijn de vertrouwelijkheid en integriteit van de gegevens niet gewaarborgd. Ook de permissies op directory's dienen ingesteld te zijn in overeenstemming met de organisatorische functiescheiding.
- Bij het instellen van permissies dient te worden uitgegaan van het

principe van veilige beginwaarden: in beginsel dienen permissies voor lezen, schrijven en executeren alleen gedefinieerd te zijn voor de eigenaar van het bestand.
- Per applicatie dient een speciaal applicatieaccount te worden aangemaakt, dat eigenaar is van de applicatiebestanden en dat hierop als enige toegangspermissies heeft; groep- en otherpermissies dienen uitgeschakeld te zijn. Gebruikers dienen uitsluitend via de daartoe geëigende applicatieprogrammatuur toegang tot de applicatiegegevens te verkrijgen.

Hierbij horen de volgende basismaatregelen:
- Stel de permissies van alle systeembestanden en systeemdirectory's zodanig in dat alleen de eigenaar (root of een ander systeem-user-id) over write-permissie beschikt. Dit zijn ten minste de volgende directory's en alle bestanden in deze directory's:
  - `/boot`, `/etc`, `/bin`, `/lib`, `/sbin`, en `/usr`;
  - `/` en `/dev` (alleen de directory);
  - `/var`, met uitzondering van `/var/mail` en `/var/tmp`.
- Zorg ervoor dat de systeemconfiguratiebestanden voldoende leesrechten hebben: in ieder geval `/etc/passwd`, `/etc/group` en `/etc/motd` moeten leesbaar voor iedereen zijn.
- Device bestanden die het mogelijk maken om randapparaten te benaderen (bijvoorbeeld `/dev/rmt` of `/dev/crt`) horen in `/dev` te staan en mogen alleen read- en write-permissies voor root hebben Dit geldt echter niet voor de `/dev/tty` devices, die voor iedereen schrijfbaar moeten zijn, evenals voor `/dev/null`, die voor iedereen op lees- en schrijfbaar moet staan.
- Directory's voor tijdelijke bestanden dienen open te staan voor iedereen die deze directory's gebruikt. In het bijzonder dienen `/tmp` en `/var/tmp` voor iedereen lees-, schrijf- en executeerbaar te zijn.
- Systeemapplicaties die alleen door root gebruikt kunnen worden, moeten van root zijn en mogen geen toegangspermissies voor anderen hebben.
- Stel de permissies van de home-directory's van gebruikers en de bestanden in deze directory's zodanig in dat alleen de eigenaar over toegangspermissies beschikt. Let hierbij vooral op de permissies van de hoger gelegen directory's. Als deze schrijfbaar zijn, kunnen de onderliggende bestanden en directory's toch worden gewijzigd.
- Voor sommige Unix-versies gelden afwijkende regels. Raadpleeg daarom altijd de documentatie van de leverancier. Vertrouw echter niet onvoorwaardelijk op de leveranciers: vaak worden ruimere permissies aanbevolen dan noodzakelijk en staan bij een standaardinstallatie de permissies te ruim.

– Laat applicaties en de daarbij behorende bestanden eigendom zijn van een specifieke applicatiebeheerder. Geef toegang tot deze bestanden via de daartoe geëigende applicatieprogrammatuur.

Extra maatregel: vele Unix-versies bieden Access Control Lists (ACLs), waarmee de toegang van elk subject tot elk object kan worden ingesteld. Zulke ACLs zijn nog steeds leverancierspecifiek; zij behoren niet tot de standaarduitrusting van Unix. Bij het gebruik van leverancierspecifieke maatregelen voor toegangsbeveiliging dient de nodige zorgvuldigheid in acht te worden genomen. Door incompatibiliteiten – bijvoorbeeld tussen opeenvolgende versies of tussen versies van verschillende leveranciers – kunnen onvoorzien lekken in de toegangsbeveiliging ontstaan bij het terugladen of migreren van bestanden. Standaardprogramma's als tar en cpio slaan ACLs niet op.

4.3 SETUID, SETGID EN STICKY BIT

Het setuid-mechanisme wordt veel gebruikt om gewone gebruikers op een (door een programma ofwel subject) gecontroleerde manier toegang te geven tot een object waar de gebruiker normaliter geen toegang heeft. Normaliter draait een proces onder de user-id en group-id van het ouderproces. Hierop bestaat één belangrijke uitzondering: als het setuid-bit van een executeerbaar bestand (een programma) aan staat, krijgt het resulterende proces de user-id van de eigenaar van het bestand, en daarmee alle toegangspermissies van die eigenaar.
Hetzelfde - maar dan met groepen - geldt voor het setgid-bit. De setuid- en setgid-bits zijn in de uitvoer van het commando `ls -l` weergegeven door het symbool '`s`' in plaats van '`x`'. Het is ook mogelijk het setuid-bit aan te zetten zonder de daarbij gebruikelijke execute-rechten te verlenen. In de uitvoer van `ls -l` wordt dit weergegeven door een hoofdletter '`S`'.

Een voorbeeld:

```
144 -rwsr-xr-x 1 root wheel 69082 Oct 8 18:53 mail
152 -r-sr-s—- 1 root wheel 73728 May 21 13:00 rcp
136 -r-sr-xr-x 3 root wheel 65536 Apr 24 1993 chsh
```

Worden de programma's `mail`, `rcp` en `chsh` in bovenstaand voorbeeld door middel van de `exec` system call geladen, dan draait het resulterende proces niet onder de user-id van het proces dat de `exec` heeft aangeroepen, maar onder *user-id 0* (*root*). Voor het programma `rcp` geldt bovendien dat het proces draait onder de rechten van de aan het bestand gekoppelde groep.

Met behulp van het setuid-mechanisme kan men programma's laten draaien onder de user-id van de eigenaar in plaats van de user-id van degene die het programma opstart. Dit wordt aangegeven met het setuid bit: `ls -l` geeft dan de permissies `rwsr-xr-x`. Een vergelijkbare situatie bestaat ook voor het setgid bit, waarbij een programma draait onder het group-id van het programma: permissie `rwxr-sr-x`.
Het setuid-mechanisme wordt gebruikt voor twee doeleinden:

- encapsulatie: hierbij kan een bepaalde verzameling afgeschermde bestanden uitsluitend worden benaderd door een enkel programma, dat draait onder de user-id van de eigenaar van die bestanden;
- communicatie tussen subjects: hierbij dient een setuid-programma, bijvoorbeeld met root als eigenaar, om bestanden aan te maken onder de user-id van een andere gebruiker (su is hiervan het bekendste voorbeeld).

Het setuid-mechanisme wordt vaak gebruikt in combinatie met root. Programma's zijn dan setuid root, hetgeen betekent dat ze worden uitgevoerd onder de permissies van root ongeacht welke gebruiker ze aanroept. Aangezien zulke programma's dan de hoogst mogelijke privileges hebben, is het van groot belang dat de correcte werking van deze programma's is gewaarborgd.
Ook shell scripts kunnen setuid root draaien. Omdat de veiligheid van een shell script vrijwel niet te waarborgen valt, brengt dit grote risico's met zich mee.

Sticky bit Het laatste protectiebit is het sticky bit, dat in oudere versies van Unix werd gebruikt om aan te geven dat een programma na executie in het geheugen moest achterblijven. Indien het sticky bit op een directory is gezet, kan een bestand in die directory alleen door de eigenaar van het bestand uit die directory worden verwijderd. Gebeurt dat niet, dan kunnen ook anderen het bestand verwijderen. Het sticky bit wordt bij het opvragen van de permissies aangegeven met een 't', bijvoorbeeld `rwxrwxrwt`. Bij sommige Unix-varianten heeft het sticky bit echter een andere betekenis. Daarom moet altijd de documentatie van de leverancier geraadpleegd worden.

Solaris: het sticky bit op directory's in Solaris betekent dat niet alleen de eigenaar het bestand kan verwijderen maar ook eenieder met schrijfpermissie op bestand.

4.3.1 Normen en maatregelen

Basisnormen voor de speciale permissies zijn:

- Het setuid-mechanisme dient met de grootste zorg te worden gebruikt, in het bijzonder voor programma's die eigendom zijn van root.
- Gebruik van het setuid-mechanisme dient per geval te worden gemotiveerd.
- De beheerder dient op de hoogte te zijn van de aanwezigheid van alle setuid-programma's.
- Setuid-programma's dienen correct te werken. Met name bij het abrupt afbreken van een deeltaak mag geen onveilige situatie ontstaan.
- Algemeen toegankelijke directory's met daarin bestanden die het eigendom zijn van specifieke gebruikers, dienen met het sticky bit te worden beschermd.

Basismaatregelen die hieruit volgen zijn:

- Gebruik het setuid-mechanisme in principe niet. Motiveer, accordeer en registreer het gebruik van setuid per geval.
- Zorg ervoor dat de correcte werking van setuid-programma's zoveel mogelijk gewaarborgd is. Zet in geen geval setuid op inherent onveilige programma's, zoals mail, write of enige commando-shell.
- Laat geen setuid shell scripts toe.
- Houd een schriftelijk overzicht bij van setuid-programma's. Bewaar dit overzicht niet op het systeem zelf.
- Waarborg dat setuid-programma's geen andere programma's, in het bijzonder shells, opstarten.
- Zet geen setuid- en setgid-programma's op met NFS geëxporteerde file systems. Deze kunnen op een ander systeem worden gecorrumpeerd.
- Zet het sticky bit op `/tmp`, `/var/tmp` en andere gemeenschappelijke directory's met het commando `chmod 1777 <dirnaam>`.

5 Logging en monitoring

Een mechanisme om te kunnen beoordelen of inbreuken op beveiliging hebben plaatsgevonden is logging. Binnen Unix bestaat een veelheid aan logging-faciliteitein. In het kader van deze PI-standaard worden niet alle mogelijkheden uitvoerig behandeld.
Bij activering moet worden nagedacht over verschillende aspecten van logging, bijvoorbeeld:

- welke acties moeten worden gelogd;
- met welke frequentie moeten het logbestand worden beoordeeld;
- hoe lang moeten logbestanden bewaard blijven;
- hoe groot mogen ze worden;
- moeten logbestanden gearchiveerd worden voor later gebruik;
- hoe kan voorkomen worden dat ongeautoriseerde personen toegang krijgen tot de logbestanden.

Om effectief gebruik te maken van logging-informatie moet regelmatig analyse van de bestanden plaatsvinden (monitoring).
Aangezien een veelheid aan gegevens wordt vastgelegd, is het gebruik van effectieve hulpmiddelen noodzakelijk. Te denken valt aan Unix-tools als `grep` en hulpmiddelen die zijn ontwikkeld in Perl (vele vormen beschikbaar via Internet).
Het analyseren van een enkel logbestand biedt onvoldoende houvast om het niveau van beveiliging te kunnen beoordelen. Met name het beoordelen van logbestanden in vergelijking met voorgaande versies van deze bestanden en in relatie tot andere logbestanden, kan hulp bieden bij de beoordeling van een systeem. Dit vergt echter een aanzienlijke investering in menskracht (het feitelijke beoordelen, met de ervaring die nodig is voor die beoordeling) en capaciteit (de ruimte om logbestanden te bewaren).
Logging kent daarnaast echter nog een groot aantal beperkingen. De belangrijkste is dat logging performance en schijfruimte kost. Daarnaast geeft logging niet altijd het gewenste resultaat: de omvang kan belemmerend zijn. En wanneer bijvoorbeeld een Unix-commando door een hacker vervangen is door een ongeautoriseerd programma, is het gebruik van deze ongeautoriseerde programmatuur toch niet rechtstreeks terug te vinden in de proceslogging.

Het uitvoeren van de controlemaatregelen is een arbeidsintensieve taak, die echter voor een deel kan worden geautomatiseerd. Van zulke rapportagetools bestaan zowel commerciële als niet-commerciële versies. Een bijzondere vorm van logging wordt opgeleverd bij het gebruik van performance meetinstrumenten, zoals SAR (System Activity Reporter). Indien zulke metingen worden uitgevoerd, verdient het aanbeveling om de meetgegevens te analyseren op ongewone gebeurtenissen.

Er bestaan globaal twee vormen van logging. De eerste vorm betreft instrumenten die door Unix zelf worden aangeboden. Daarnaast zijn er applicaties die aanvullende logging-faciliteiten aanbieden. Beide komen in de volgende paragrafen aan de orden.

5.1 STANDAARD LOGGING-FACILITEITEN

Unix biedt zelf mogelijkheden voor logging, zoals de Syslog, Account Log en Audit Log. Audit log is in de meeste gevallen alleen beschikbaar wanneer het systeem als Trusted System volgens C2-classificatie geïnstalleerd. Voor elk van deze logging-vormen bestaan specifieke normen en maatregelen, maar hiervan zijn ook enkele in algemene zin te formuleren, met name in het licht van gebruik voor monitoring en audits.

5.1.1 Normen en maatregelen

Basisnormen ten aanzien van logging zijn:

- Er dient beleidsmatig te worden vastgesteld wat de bewaartermijn van logbestanden is.
- Er dient gebruik te worden gemaakt van logbestanden en loggingopties.
- De logbestanden moeten dagelijks door de systeembeheerder worden gecontroleerd.
- Logbestanden moeten beschermd worden tegen ongeautoriseerde toegang.

Als basismaatregelen gelden:

- Activeer de volgende loggingopties:
  - `syslog`;
  - `wtmp`;
  - `utmp`;
  - `btmp`;
  - `audit-log` (indien beschikbaar);
- Plaats (indien mogelijk) logbestanden niet in de directory `/etc`,

maar in de directory `/var/log`, zodat `/etc` als read-only directory kan worden aangemerkt.
- Aan logbestanden mogen alleen records worden toegevoegd. Dit is onder Unix alleen te realiseren door voor logging gebruik te maken van de systeemloggingfaciliteit. Alleen functionarissen met auditrechten mogen de logbestanden lezen.
- Controleer de logbestanden dagelijks op geregistreerde inbreuken op het gedefinieerde beveiligingsniveau. Laat de logbestanden controleren door een ander dan de systeembeheerder, bijvoorbeeld door een beveiligingsfunctionaris of een (interne) controlemedewerker.
- Sla de logginginformatie periodiek op in een archiefbestand.
- Schoon de logbestanden om te voorkomen dat schijven vollopen.
- Bewaar logginginformatie ten minste drie maanden, of zoveel als wordt geëist door wet- en regelgeving.

Extra maatregelen zijn:
- Maak gebruik van een afzonderlijke logserver, waarop de overige geïnstalleerde systemen hun logginginformatie vastleggen.
- Leg de logginginformatie vast op niet-modificeerbare media, zoals papier, worm-disks (bijvoorbeeld een eenmalig beschrijfbare CD-ROM) of een extern computersysteem (laat bijvoorbeeld de logbestanden opslaan op een logserver).

5.1.2 System log

De system log is ontwikkeld binnen de BSD Unix-variant en overgenomen door enkele leveranciers van System V-systemen. De foutmeldingen van verschillende processen worden door het daemon-proces `syslogd` verzameld en in de `syslog` geregistreerd. In het bestand `/etc/syslog.conf` wordt geregistreerd wat door `syslogd` moet worden gelogd. Elke regel in `/etc/syslog.conf` beschrijft waar specifieke (fout)meldingen naartoe geschreven moeten worden. Deze meldingen zijn ingedeeld naar oorsprong, bijvoorbeeld fouten van de kernel en fouten van autorisatieovertredingen. Wanneer geen configuratiebestand aanwezig is, worden alle meldingen naar het standaardboodschappenbestand `/var/log/messages` geschreven. Dit bestand kan snel groeien en zal daardoor minder goed bruikbaar zijn, zeker als ook andere processen dit bestand voeden met logginggegevens.
Daarnaast worden verschillende niveaus van foutmeldingen onderscheiden (`emerg, alert, crit, err, warning, notice, info, debug`). Ook vanuit een applicatie kunnen met `syslogd` specifieke foutmeldingen worden gegenereerd. Het activeren hiervan is echter alleen mogelijk als hiermee in de ontwikkeling van de applicatie rekening is

gehouden. Wanneer dat gewenst is, moet daarom bij de systeemontwikkeling een dergelijke eis worden meegenomen in de fase van de informatieanalyse.
In sommige Unix-versies wordt de `syslog` geopend met het `syslogd` -commando. In dat geval kunnen verschillende parameters worden meegegeven, zoals de interval voor het zetten van tijdstempels als die af moet wijken van de standaard twintig minuten.

HP-UX: syslog schrijft zijn informatie in het bestand `/var/adm/syslog/syslog.log`.

5.1.3 Normen en maatregelen system log

De basisnorm ten aanzien van de system log is:

– Maak zoveel mogelijk gebruik van de loggingfaciliteiten die een operating system zelf bezit.

De hieruit voortvloeiende basismaatregelen zijn:

– Activeer syslog en schrijf de gegevens weg in een eigen logbestand.
– Het logbestand (`/var/log/syslog`) is eigendom van root en alleen beschrijfbaar door `syslogd`.
– Stel `/etc/syslog.conf` zo in dat alle loggegevens naar een logbestand worden geschreven, bijvoorbeeld `/var/log/syslog`. Stuur urgente berichten ook naar het console.
– Laat, indien mogelijk, applicaties op een (instelbare) eigen oorsprong (bijv. `LOG_LOCAL0`) en op verschillende niveaus loggen. Regel het gewenste logniveau in `/etc/syslog.conf`. Laat de meldingen zo volledig mogelijk zijn (procesnaam, process-id, en een duidelijke omschrijving).

Als extra maatregel valt te nemen:

– Log alle gebeurtenissen ook op een permanent medium en/of op een logserver.

5.1.4 User login/logoutinformatie

Gegevens over het inloggen en uitloggen van gebruikers worden vastgelegd in de volgende bestanden, te vinden in de directory `/var/log`:

– `wtmp`;
– `utmp`;
– `btmp`.

In het `wtmp`-bestand wordt door Unix geregistreerd welke logins en logouts hebben plaatsgevonden. Hiervoor wordt de gebruikersnaam, terminalnaam en het in- en uitlogtijdstip opgeslagen. In het `utmp`-bestand wordt door Unix bijgehouden welke gebruikers momenteel op het systeem zijn ingelogd. Naast `wtmp`- en `utmp`-bestanden bestaat er vaak ook nog het `btmp`-bestand. Hierin worden de mislukte inlogpogingen geregistreerd, waarbij onder andere de datum, de tijd en de username worden vastgelegd. Vaak bevat dit bestanden wachtwoorden, die abusievelijk als username zijn ingevoerd. Deze loggingopties zijn standaard actief op een Unix-systeem en kosten nauwelijks performance.
Door de gegevens uit de `wtmp`- en `utmp`-bestanden te vergelijken met een uren- c.q. aanwezigheidsregistratie kan het misbruik van user-id's worden getraceerd. Misbruik van user-ids kan een indicatie zijn voor een succesvolle inbraak of het delen van de autorisaties van een gebruiker. Door analyse van het `btmp`-bestand wordt zichtbaar op welke user-id's mislukte inbraakpogingen zijn ondernomen. Wanneer de `btmp`-bestand voor ongeautoriseerden leesbaar is, zou vertrouwelijke informatie bekend kunnen worden, aangezien abusievelijk ingevoerde wachtwoorden in het bestand kunnen worden aangetroffen.

Solaris logt mislukte logins naar `/var/adm/loginlog`, als dat bestand bestaat. Aangeraden wordt dat bestand te creëren: `touch /var/adm/loginlog, chmod 600 /var/adm/loginlog, chown root /var/adm/loginlog`. De meest volledige informatie over logins staat in `/var/adm/wtmpx` (historisch) en `/var/adm/utmpx` (huidig). De `utmp/wtmp`-bestanden bevatten slechts verkorte informatie en zullen in de toekomst verdwijnen. Extra logging facility is C2 auditing. Dit maakt het mogelijk per gebruiker zeer gedetailleerd te auditen (op system callniveau). C2 auditing wordt geactiveerd door middel van `/etc/security/bsmconv` en gedeactiveerd door middel van `/etc/security/bsmunconv`.

HP-UX: Indien het bestand `/var/adm/btmp` aanwezig is, zullen de mislukte inlogpogingen worden gelogd. Door middel van het commando `lastb` kan dit bestand worden uitgelezen. Aangeraden wordt dat bestand te creëren: `touch /var/adm/btmp, chmod 600 /var/adm/btmp, chown root:other /var/adm/btmp`.

5.1.5 Normen en maatregelen user login

De basisnorm voor user login/logoutinformatie is:

– De standaardlogbestanden `wtmp`, `utmp` en `btmp` dienen door de systeembeheerder geanalyseerd te worden om inbraakpogingen en het misbruik van user-id's te detecteren.

Basismaatregelen zijn:

- De bestanden `wtmp`, `utmp` en `btmp` worden automatisch gevuld door de processen login en logout (exit). De genoemde bestanden mogen niet world-writeable zijn (zet permissies op 644), zodat alleen systeemprocessen de bestanden kunnen muteren en gebruikers niet rechtstreeks de logbestanden ongecontroleerd kunnen wijzigen om bijvoorbeeld sporen van hun aanwezigheid te verwijderen.
- Het `btmp`-bestand mag niet leesbaar zijn voor andere gebruikers dan root.
- Beoordeel de logbestanden periodiek.

5.1.6 Audit log

De meeste versies van Unix beschikken over een audit log, waarmee op commandoniveau logging-informatie kan worden verzameld. De faciliteit is in vrijwel alle gevallen alleen beschikbaar wanneer het systeem als Trusted System volgens C2 classificatie geïnstalleerd. De audit log is de verzameling van logging bestanden van het auditsubsysteem van Unix. De audit logs kunnen worden bewerkt en geanalyseerd met een audithulpmiddel. In het audit log worden gebeurtenissen (inclusief commando's) en system calls gelogd die door gebruikers worden geactiveerd.
Naast de algemene gegevens, zoals het tijdstip van inloggen en de username, is het mogelijk om binnen audit logging aan te geven welke events moeten worden gelogd. Om er zeker van te zijn dat het auditsysteem normaal werkt, draait een deamon in de achtergrond om de verschillende auditparameters te bewaken. Indien bepaalde waarden worden overschreden of delen van het auditsysteem worden verwijderd, zal het daemonproces een melding genereren en het probleem proberen te herstellen.
Indien het bij de gebruikte Unix-versie mogelijk is om per gebruiker aan te geven welke events moeten worden gelogd, moet de logging worden geactiveerd voor beveiligingsrelevante functies, zoals operators, DBA's en beveiligingsfunctionarissen. Als deze mogelijkheid niet beschikbaar is, zal voor het basisniveau de registratie voor alle gebruikers moeten plaatsvinden. Het inloggen hoeft hier niet te worden meegegeven, omdat de registratie van de logins en de logouts al standaard plaatsvindt in het `wtmp`-bestand (zie hiervoor). De volledigheid van de logging is moeilijk te controleren omdat de regels niet zijn genummerd en de tijd tussen registraties variabel is.
Bij de audit-logging moet een duidelijke afweging gemaakt worden tussen beveiliging en performance. Er wordt zeer veel informatie vastgelegd, waardoor het risico bestaat dat het file system vol loopt.

5.1.7 Normen en maatregelen audit log

Basisnorm voor audit logging is:

– Kritische activiteiten van alle gebruikers, alsmede alle activiteiten van kritische gebruikers dienen te worden geregistreerd.

Basismaatregelen zijn:

– Activeer de audit log en stel de bijbehorende autorisaties en bestandspermissies goed in.
– Log (afhankelijk van de specifieke mogelijkheden van het systeem) de events die betrekking hebben op het aanmaken, verwijderen, en muteren van (eigenschappen van) objecten.
– Beoordeel de logbestanden periodiek.

5.1.8 Process accounting

In het `acct`-bestand kan ieder commando dat door een gebruiker wordt ingegeven, worden gelogd. Dit logbestand is in eerste instantie bedoeld voor de doorbelasting van systeemgebruik (process accounting), maar is ook bruikbaar voor de controle op gebruikte commando's. Process accounting registreert niet de parameters die met het opstarten van een proces worden meegegeven. Dit betekent dat de gelogde gegevens niet volledig zijn. Process accounting levert in tegenstelling tot de meeste andere vormen van logging direct informatie over welke commando's door wie zijn gebruikt.
Met behulp van process accounting kunnen inbreuken op verschillende beveiligingsrisico's worden ontdekt, voorzover deze niet afhankelijk zijn van de bij de uitgevoerde commando's toegepaste parameters. Het betreft hierbij zowel het misbruik van bevoegdheden door medewerkers als het misbruik van resources.

Bij process accounting bestaat meestal niet de mogelijkheid om selectief te loggen. Hierdoor heeft het activeren van process accounting met name consequenties voor het beheer en analyse.
De voor beveiliging bruikbare functionaliteit wordt ook door het audit-log geleverd, als dat geactiveerd is, waardoor sommige activiteiten dubbel worden gelogd. Het gebruik van accounting als logginghulpmiddel heeft daarom in die situatie weinig toegevoegde waarde. Indien de Unix-versie die wordt gebruikt de mogelijkheid heeft van audit-logging, dan zou er de voorkeur naar uitgaan alleen gebruik te maken van audit-logging.

5.1.9 Normen en maatregelen process accounting

Bij process accounting geldt als basisnorm:

– Gebruik process accounting alleen voor beveiligingsdoeleinden als de audit-log niet actief is.

Als basismaatregel vloeit hieruit voort:
– Deactiveer process accounting, tenzij andere toepassingen (bijvoorbeeld doorbelasting) de werking ervan noodzakelijk maken.

5.2 APPLICATIESPECIFIEKE LOGGINGS

Buiten de loggingmogelijkheden die Unix zelf biedt, zijn er natuurlijk ook de door afzonderlijke programma's geboden mogelijkheden van logging. De meeste tot de standaard Unix-omgeving behorende programma's bieden min of meer gelijksoortige loggingmogelijkheden. Voorbeelden hiervan zijn `su` en `cron`. Ook de verschillende programma's van derden en zelf ontwikkelde toepassingen kunnen voorzien in loggingfaciliteiten. Applicatieloggings omvatten ook de before en after images van databasetransacties of zeer specifieke audit trails in bijvoorbeeld registratieve informatiesystemen. Raadpleeg de systeemdocumentatie van dergelijke applicaties voor de loggingmogelijkheden. Hier worden alleen enige Unix-specifieke applicatieloggings behandeld.

In de sulog wordt elke aanroep van `/bin/su` vastgelegd (het bestand is te vinden als `/var/log/su`). Daarbij worden onder andere de datum, tijd, terminalnaam en username vastgelegd.
In de message-log worden door BSD-systemen alle consoleberichten gelogd. Een deel van de berichten heeft betrekking op uitgevoerde of mislukte `su` -commando's. BSD kent zelf geen sulog. Het messages-bestand kan in het file system worden gevonden als `/var/log/messages`.

Bij het configureren van de loggingmogelijkheden zou stilgestaan moeten worden bij het bepalen van het bestand waarin de boodschappen worden geplaatst, aangezien de bestanden vol kunnen lopen en analyse bij een grote diversiteit aan applicaties minder makkelijk zou kunnen worden.
Door het gebruik van Setuid-programma's wordt een proces onder de user-id van een ander dan de gebruiker zelf uitgevoerd. Het gebruik van de applicatieloggings is derhalve essentieel om de audit trail te kunnen bewaken.

Solaris: su wordt gelogd via `syslog` en in `/var/adm/sulog`.

HP-UX: su schrijft zijn loginformatie in het bestand `/var/adm/sulog`. De defaultplaats voor de message-log is `/var/adm/messages`. Echter, `cron` dient dan bijvoorbeeld wel van de

volgende regel te worden voorzien: `05,15,25,35,45,55/usr/sbin/dmesg - >> /var/adm/messages`.

5.2.1 Normen en maatregelen

Basisnormen voor applicatieloggings zijn:

– Het gebruik van user-id 0 (root) dient te worden gedetecteerd en geanalyseerd op het onbevoegde gebruik ervan.

Basismaatregelen

– Activeer de beschikbare applicatieloggings, in ieder geval sulog.
– De sulog en messages bestanden mogen niet door andere gebruikers dan root worden gelezen.
– Beoordeel de logbestanden periodiek.

Een mogelijke extra maatregel is:

– Activeer `syslog` voor foutgevoelige of kritische applicaties.

5.3 AUDIT EN MONITORING

Het uitvoeren van de controlerende taak vergt een grote mate van deskundigheid. Mede doordat er geen rechtstreeks verband kan worden gelegd tussen het functioneren van een systeem en de inspanning die met de controle gemoeid is, kan gesteld worden dat de auditfunctie het stiefkindje in een automatiseringsorganisatie is (de audit is de volledige controlerende taak waarvan monitoring, met name van logbestanden, de technische component is). In het kader van deze standaard wordt erop gewezen dat zekerheid met betrekking tot het gerealiseerde beveiligingsniveau zonder een dergelijke functie niet kan worden verkregen.
Het inrichten van de auditfunctie in een Unix-omgeving omvat zowel technische als organisatorische maatregelen. Het inzetten van de juiste mix aan maatregelen zou moeten plaatsvinden op grond van een vastgestelde auditstrategie die op zich weer vanuit het beveiligingsbeleid zou moeten worden bepaald.
Zonder toezichthoudende functie berust het volledige beveiligingsniveau alleen op het vertrouwen in de deskundigheid en goede wil van de beheerdersfunctie. Het nalaten van audit kan ertoe leiden dat opgetreden inbreuken op het beveiligingsniveau of manco's in het beheer van de omgeving onopgemerkt blijven. De risico's van gebrekkige monitoring zijn duidelijk te formuleren:

– Wanneer de logging niet op een efficiënte manier is te beoorde-

len, zal de logging in het geheel niet beoordeeld worden, waardoor het doel van de logging niet wordt gehaald.
- Als logbestanden gedurende langere tijd niet worden geanalyseerd, bestaat de kans dat inbreuk op het beveiligingsniveau onopgemerkt blijft.
- Sommige events worden niet gelogd, waardoor onvoldoende zicht is op het gerealiseerde niveau van beveiliging.

Unix kent van zichzelf geen specifieke auditfaciliteiten. Toch is het van essentieel belang dat regelmatig wordt gecontroleerd op de werking van de getroffen maatregelen. Unix biedt daarvoor wel een groot arsenaal aan hulpmiddelen. Voor de meeste daarvan is echter wel een grondige kennis van de Unix-programmeerhulpmiddelen nodig om de beschikbare gegevens op een effectieve en efficiënte manier te kunnen analyseren.

5.3.1 Normen en maatregelen

Basisnormen voor audit en monitoring zijn:
- Het auditen van een systeem biedt de waarborg dat de werking van de maatregelen is zoals bedoeld (toetsen 'opzet' versus 'werking' van de maatregelen).
- Vanuit het beginsel van functiescheiding dient er een toezichthoudende functie te bestaan.
- Periodiek dient een overzicht verstrekt te worden van afwijkingen van en wijzigingen in de instellingen als beschreven in deze PI-standaard.
- Geautomatiseerde vastleggingen dienen geautomatiseerd te worden geanalyseerd.

Basismaatregelen zijn:
- Er dient minimaal een toezichthoudende functie te worden ingericht die tot taak heeft controle op het beheer uit te voeren.
- De toezichthouder dient te beschikken over de deskundigheid en hulpmiddelen om de functie te kunnen uitoefenen.
- De toezichthouder dient periodiek verantwoording af te leggen over de uitgevoerde werkzaamheden.
- Bepaal het basis auditniveau.
- Bepaal welk niveau van logging noodzakelijk is.
- Voer wekelijks een (geautomatiseerde) controle uit op de in deze PI-standaard gedefinieerde maatregelen. Genereer hiervan een overzicht bestemd voor het management.
- Op basis van de periodieke beoordeling moeten maatregelen worden getroffen om de gesignaleerde beveiligingsincidenten in de toekomst te voorkomen.

– Maak gebruik van automatisch opstartende scripts om de logbestanden te analyseren (maak gebruik van `cron` en `at`).

Denkbare extra maatregelen zijn:
– Maak gebruik van commerciële verkrijgbare hulpmiddelen voor audit.
– Maak gebruik van een logging analysetool die voorziet in een real-timealertfunctie (bijvoorbeeld tklogger of swatch).

6 Netwerkdiensten

Unix-systemen worden op grote schaal ingezet in netwerkomgevingen. Daarbij wordt in de meeste gevallen gebruikgemaakt van het protocol TCP/IP, dat tot de standaarduitrusting van elk Unix-systeem behoort. Daarnaast zijn veel Unix-systemen uitgerust met andere protocol suites, zoals DECNET, OSI, SNA, Netbeui en IPX. Hier worden echter uitsluitend de TCP/IP-protocollen worden behandeld.
Tevens komen de netwerkdiensten aan bod die door TCP/IP ondersteund worden. Daarbij gaat het om toepassingen als Telnet, FTP en SMTP, om de Berkeley r-utilities, om ondersteunende diensten voor netwerkverkeer (NFS, NIS, DNS) en om de grafische gebruikersinterface X-Windows. In de laatste paragraaf komt UUCP aan de orde, een standaard netwerkapplicatie van Unix, die doorgaans echter geen gebruikmaakt van TCP/IP.

HP-UX biedt de mogelijkheid gebruik te maken van de zogenaamde kerberized-netwerk services: ftp, rcp, remsh, rlogin en telnet. Met behulp van het commando 'inetsvcs_sec' kunnen de Secure Internet Services gebaseerd op DCE security services, Kerberos V5, worden beheerd. Standaard wordt de DCE-client meegeleverd. IPSec, om end-to-endnetwerkbeveiliging te verkrijgen, is eveneens aanwezig. Zowel een VPN als op rollen-gebaseerde beveiligingspolicy kan worden geïmplementeerd. RPC en Secure RPC zijn beide mogelijk.

6.1 DE TCP/IP PROTOCOLSUITE

TCP/IP (Transport Control Protocol / Internet Protocol) is een verzameling protocollen die de uitwisseling van informatie over een netwerk mogelijk maakt. Het Internet is op deze standaard gebaseerd. De door TCP/IP ondersteunde netwerkprotocollen staan in tabel 7. De ondersteunende netwerkprotocollen maken deel uit van de kernel van het Unix-besturingssysteem.
Op TCP/IP is een aantal netwerkapplicaties gebaseerd (zie tabel 8). Deze draaien als afzonderlijke processen, die gebruikmaken van de functies van de ondersteunende protocollen. De meeste netwerk-

applicaties worden gestuurd door een centraal proces, inetd. Dit proces beluistert de verschillende poorten en zorgt ervoor dat desgevraagd de bijbehorende applicatieprocessen worden gestart. Deze poortnummers en bijbehorende processen zijn gedefinieerd in `/etc/inetd.conf` (ook andere diensten kunnen door middel van dit configuratiebestand worden gestart).

TABEL 7

Protocol	Functie
Internet protocol (IP)	Adressering, routering
Transport Control Protocol (TCP)	Transport, foutafhandeling, bewaking volgordelijkheid
User Datagram Protocol (UDP)	Transport
Internet Control Message Protocol (ICMP)	Netwerkaansturing

Standaardnetwerkprotocollen in de TCP/IP suite.

TABEL 8

Netwerkdienst / protocol	Functie
Telnet	Inloggen en interactief werken vanaf een ander systeem
File Transfer Protocol (FTP)	Bestandsuitwisseling
Simple Mail Transfer Protocol (SMTP)	e-mail
Berkeley R-utilities (waaronder rwho)	Diverse netwerkapplicaties
Network Information System (NIS)	Applicatie voor het beheer van gedistribueerde omgevingen
Network File System (NFS)	Middleware voor transparante toegang tot bestanden vanaf een ander systeem
Finger	Geeft informatie over gebruikers, login- en logouttijden en mailstatus
Talk	Full-duplexpraatprogramma
Domain Name Service (DNS)	Vertaalt IP-adressen naar Internet-namen en vice versa
X window system	Afbeelden van tekst en grafiek op een 'remote' grafisch display

Netwerkdiensten binnen de TCP/IP context

De adressering in een TCP/IP-netwerk is gebaseerd op IP-adressen en poortnummers:

– IP-adressen worden gebruikt om een specifiek interface (van een computer) in een specifiek netwerk te adresseren. IP-adressen zijn 32 bits groot en opgebouwd volgens een hiërarchische structuur. IP-adressen kunnen zowel permanent als dynamisch worden toegekend. IP-adressen worden doorgaans vertaald naar IP-namen en vice versa. Deze vertaalslag kan op verschillende

manieren plaatsvinden (zie bijvoorbeeld de beschrijving van DNS in paragraaf 6.4.5). IP-adressen zijn voor de gehele internetwereld in beginsel uniek geregistreerd. Er bestaan drie series van IP-adressen die alleen bruikbaar zijn voor het gebruik binnen interne netwerken. Momenteel is overigens een beweging ingezet om de IP-adressering ingrijpend te vernieuwen. In IP versie 6 is de adreslengte 128 bits.
- Poortnummers worden gebruikt om specifieke applicatieprocessen te adresseren. De meeste netwerkapplicaties hebben algemeen bekende poortnummers. Het bestand `/etc/services` beschrijft welke applicaties bij welke poortnummers horen.

In principe is elke netwerkapplicatie zelf verantwoordelijk voor de authenticatie van gebruikers en diensten. De meeste netwerkdiensten verzorgen zelf geen enkele authenticatie. De volgende situaties kunnen worden onderkend:
- er is in het geheel geen authenticatie (dit is de standaardwerkwijze, voorbeeld rwho);
- authenticatie vindt plaats op basis van IP-adressen of hostnames (hierbij bestaat het risico van 'spoofing', dat hieronder wordt uitgewerkt);
- authenticatie vindt, bijvoorbeeld door Telnet en FTP, plaats op basis van wachtwoorden of andere unieke karakterstrings (die voor 'sniffers' leesbaar over het netwerk worden getransporteerd);
- authenticatie vindt plaats door middel van tickets (DCE – Kerberos) of certificaten (PKI-infrastructuur), waardoor een redelijk hoge mate van zekerheid wordt gekregen.

Een bijzondere vorm van netwerkdienst is de Remote Procedure Call, waarvoor het RPC-protocol is ontwikkeld. Door middel van RPC kan transparant een dienst op een ander systeem worden gestart om vanaf dat systeem specifieke taken uit te voeren. Verschillende in deze standaard genoemde diensten maken gebruik van het RPC-mechanisme. Te denken valt aan NFS en de verschillende directory services. Het RPC-protocol zelf maakt gebruik van een zogenaamde 'well known' poort. De diensten die van RPC gebruikmaken, registreren zelf een poortnummer voor de communicatie. De koppeling tussen deze beide poorten wordt geregeld via de Portmapper. Secure RPC is een aangepaste versie van RPC waarin authenticatie met behulp van een PKI-infrastructuur is ingebed. Er bestaan ook versies die voor authenticatie gebruik kunnen maken van een Kerberos-server.

6.1.1 Risico's

Een van de risico's bij het gebruik van TCP/IP is het gebruikmaken van spoofingtechnieken. Dit betekent het nabootsen van netwerkadressen door een ander dan de eigenaar van het betreffende adres, met het doel een vertrouwd adres te stelen. Een van de weinige methoden om 'spoofing' echt effectief te ondervangen is het segmenteren van het netwerk. De routers die het verkeer tussen de verschillende stukjes netwerk routeren kunnen dan met speciale 'anti-spoofing'-filters waarborgen dat adressen uit het ene stukje netwerk niet vanuit het andere stukje worden gebruikt. Daarmee is het risico van 'spoofing' teruggebracht tot de omvang van de betreffende segmenten. Verder kan 'spoofing' met behulp van netwerksniffers gedetecteerd en vervolgens aangepakt.
'Sniffing' is de techniek om het verkeer dat over het netwerk wordt getransporteerd, af te luisteren. Sniffing wordt zowel gebruikt om onderhoud op de technische infrastructuur uit te voeren, als om te analyseren welke gegevens over het netwerk worden getransporteerd. Hierdoor kunnen vertrouwelijke gegevens (bijvoorbeeld niet-versleutelde wachtwoorden) beschikbaar komen bij niet-geautoriseerde personen.

Het gebruik van standaardnetwerkapplicaties binnen het besturingssysteem kan leiden tot een doorbreking van de ingestelde functiescheidingen. Bij het gebruik van standaardnetwerkapplicaties dient grote voorzichtigheid in acht te worden genomen. Er dient hoe dan ook een beveiligingsbeleid te worden geformuleerd, op grond waarvan het specifieke gebruik van netwerkdiensten wordt beveiligd. Het gebruik van netwerkdiensten kan leiden tot het ongewenste resultaat dat het computersysteem wordt blootgesteld aan ongeautoriseerde gebruikers van buiten de eigen vertrouwde organisatie. Hoewel zij op dezelfde manier te beveiligen zijn als de standaard Unix-diensten, leidt de beschikbaarheid van netwerkdiensten ertoe dat met name technisch vaardige buitenstaanders kunnen trachten buiten de reguliere toegangscontrole om binnen te komen. Alleen al het bieden van de mogelijkheid om na te gaan welke accounts op het netwerk aanwezig zijn (bijvoorbeeld `finger` en `rwhod`), kan voor potentiële hackers verleidelijk zijn.

Opgemerkt wordt dat het beveiligen van een lokaal netwerk dat verbonden wordt met andere lokale netwerken of met Internet, alleen goed mogelijk is door gebruik te maken van een combinatie van routers en firewalls. Deze componenten voorzien in het al dan niet doorlaten van IP-pakketten en fragmenten daarvan, TCP/UDP-sessies en de feitelijke data. Deze standaard gaat niet in detail in op deze materie. Zie [Van Dam].

Bij grotere organisaties of bij netwerken met computers met sterk verschillende veiligheidsrisico's zou ervoor gekozen kunnen worden om ook binnen een intern netwerk (een intranet) gebruik te maken van zogenaamde 'firewalls', die processen binnen de verschillende netwerksegmenten van elkaar afschermen en het onderlinge verkeer tussen die verschillende netwerksegmenten bewaken. Op deze manier kan bijvoorbeeld worden voorzien in scheiding van het ontwikkel-, test- en productiedomein, waarna alleen via stricte procedures de overdracht tussen de domeinen zou mogen worden geregeld.
Een goedkope methode om firewallachtige functionaliteiten toe te passen, is het gebruik van het (gratis) pakket `TCP-wrappers`. Een dergelijk systeem kapselt de netwerkdiensten in, waardoor op TCP-pakketniveau beveiliging kan worden geregeld.
Extra aandacht wordt gevraagd voor het invoeren van een adequate wachtwoordpolitiek, die desnoods afgedwongen kan worden met aanvullende tools. Door CERT is geconstateerd dat circa tachtig procent van alle inbreuken op beveiliging in netwerkomgevingen te wijten is aan het onjuist omgaan met wachtwoorden. Het systeem mag nog zo goed beveiligd zijn, tegen een gecompromitteerd account is geen kruid gewassen.

6.1.2 Normen en maatregelen

Voor de toepassing van TCP/IP en netwerkdiensten in het algemeen geldt als basisnorm:

– Het gebruik van netwerkdiensten dient te worden ingericht op grond van een eigen beveiligingsbeleid.

De basismaatregelen zijn:

– De beveiliging van netwerkapplicaties dient gebaseerd te zijn op het principe van isolatie, het principe van beperking van functionaliteit en het principe van veilige beginwaarden. Dit houdt onder meer in dat alle netwerkapplicaties standaard moeten worden uitgeschakeld en slechts bij expliciete toestemming moeten worden ingeschakeld.
– In veel Unix-omgevingen zijn de netwerkprotocollen en -applicaties niet alleen standaard geactiveerd, maar worden zij ook intensief gebruikt, zodat het simpelweg uitschakelen ervan geen realistische optie is. Slechts door het activeren van netwerkapplicaties expliciet te autoriseren kunnen de daaruit voortvloeiende risico's werkelijk zichtbaar worden gemaakt.
– In het algemeen kan gesteld worden dat de in tabel 9 genoemde diensten uitgeschakeld kunnen worden, tenzij uitdrukkelijk vaststaat dat het gebruik ervan noodzakelijk is. In dat geval dient het gebruik ervan gedocumenteerd te zijn.

– Deactiveer deze netwerkdiensten in `/etc/rc` en `/etc/inetd.conf` door ze te kenmerken als commentaar, of neem ze (bij installatie) helemaal niet op in deze bestanden. Motiveer, accordeer en registreer alle uitzonderingen. Tref bij deze uitzonderingen de maatregelen als gedefinieerd in de rest van dit hoofdstuk. Zorg ervoor dat de eigenaar van `/etc/inetd.conf` gelijk is aan user-id 0 (root) en dat de permissies op 644 staan, zodat dit bestand alleen met root-autorisatie kan worden gewijzigd.
– Het bestand `/etc/services` dient als eigenaar user-id 0 (root) te hebben en permissies 644.
– Inventariseer regelmatig welke diensten worden gebruikt en analyseer of het gebruik is geautoriseerd.

TABEL 9

Naam	Poort	Protocol	Naam	Poort	Protocol
systat	11	TCP	exec	512	TCP
netstat	15	TCP	biff	512	UDP
tftp	69	UDP	who	513	UDP
finger	79	TCP	uucp	540	TCP
link	87	TCP	route (rip)	520	UDP
supdup	95	TCP	openwin	2000	TCP
NeWS	144	TCP			

Bij voorkeur af te sluiten services.

Bij wijze van extra maatregelen valt te denken aan:
– Beperkt de risico's van onder meer spoofing en sniffing door maatregelen als:
  • gebruikmaken van eenmalige authenticatiegegevens, zodat hergebruik niet mogelijk is en afluisteren dus geen probleem is - te denken valt aan het gebruik van One-Time Passwords met behulp van bijvoorbeeld een calculator;
  • versleuteling van de getransporteerde authenticatiegegevens op applicatieniveau;
  • versleuteling op een laag tussen de applicatie en de netwerklaag, bijvoorbeeld met behulp van 'Kerberos';
  • encryptie van al het TCP/IP-verkeer op kernelniveau;
  • het fysieke netwerk baseren op glasvezel- in plaats van koperbekabeling of coax, omdat glasvezel extra moeilijk is af te luisteren;
  • gebruikmaken van switches in plaats van hubs;
  • alleen de te gebruiken aansluitpunten voorzien van bekabeling en via een afsluitbare patchkast aansluiten.

Het nadeel van de meeste van deze maatregelen is dat ze ingrijpend en soms bewerkelijk en moeilijk te realiseren zijn, mede omdat alle lokale systemen de betreffende maatregel (met name bij versleuteling) moeten ondersteunen.

– Hanteer hulpmiddelen die de wachtwoordpolitiek ondersteunen, zoals One-Time Passwords en tools die de sterkte van wachtwoorden controleren.
– Realiseer toegangsbeveiliging en wederzijdse authenticatie door functionaliteit aan de applicatie toe te voegen. Kerberos, opgenomen in OSF's Distributed Computing Environment (DCE), is hiervan een voorbeeld.
– Controleer tekortkomingen in de netwerkbeveiliging geautomatiseerd, bijvoorbeeld met behulp van `Satan`.
– Beveilig en reguleer de toegang tot specifieke netwerkapplicaties door functionaliteit aan de ondersteunende protocollen toe te voegen. `TCP-wrappers` is hiervan een voorbeeld. Stel `TCP-wrappers` als volgt in:
  - PARANOID mode;
  - RFC931 optie;
  - negeer alle hosts door 'all:all' in `/etc/hosts.deny` te plaatsen;
  - wrap in ieder geval alle UDP-services.

6.2 TELNET, FTP EN SMTP

Drie veelgebruikte Internettoepassingen die deel uitmaken van standaard in Unix beschikbare applicaties zijn Telnet, FTP en SMTP. Deze worden hieronder behandeld. De functionaliteit van het world wide web (http, java, activex, enzovoort) wordt door Unix niet standaard ondersteund.

6.2.1 Telnet

Met telnet kan de gebruiker via het netwerk inloggen op een 'remote' computersysteem. Net als bij het inloggen via een terminalverbinding vraagt het systeem in dat geval om het invoeren van een username en een wachtwoord. Deze worden onversleuteld over het netwerk getransporteerd, hetgeen een risicofactor is. Een (juridische) drempel tegen het ongeautoriseerd gebruik van een computersysteem is het op het scherm presenteren van een 'Verboden toegang voor onbevoegden'-bericht vóór het inloggen (bijvoorbeeld via `/etc/sttydefs` of `/etc/issue`) en na het inloggen (via `/etc/motd`).
Middels telnet kan een gebruiker via zijn terminal ook een connectie opstarten naar een willekeurige andere dan de standaardtelnetpoort. Hierdoor kan hij bijvoorbeeld rechtstreeks mail sturen via sendmail, waarbij elk veld een zelfbedachte waarde kan krijgen.

Solaris: de login procedure drukt driemaal in boodschap af:
- `/etc/issue`, bij het maken van verbinding, voor het inloggen
- login promt, te veranderen in `/etc/default/telnetd` (2.6 en hoger) of met pmadm/admintool voor direct aangesloten terminals. Deze wordt niet afgedrukt wanneer er verbinding wordt gemaakt met rlogin;
- `/etc/motd`, na het succesvol in loggen.

HP-UX: Om de deamon van telnet een speciale boodschap voor het aanloggen te genereren dient de opt '-b bestandsnaam' te worden meegegeven in het configuratiebestand `/etc/inetd.conf: telnetd –b` bannerbestand.

6.2.2 Normen een maatregelen Telnet

Basisnormen voor het gebruik van Telnet zijn:
- Bij inloggen dient melding te worden gemaakt van het feit dat alleen geautoriseerd gebruik is toegestaan. Dit biedt natuurlijk geen enkele bescherming tegen ongeautoriseerde toegang, maar is een minimale vereiste in het kader van de Wet computercriminaliteit.
- Gebruikers dienen op de hoogte gesteld te worden van de tijdstippen waarop onder hun account is ingelogd.

Basismaatregelen voor het gebruik van Telnet zijn:
- Toon voor het inloggen een bericht waarin duidelijk wordt gemaakt dat alleen geautoriseerde gebruikers mogen inloggen en dat het systeem alleen gebruikt mag worden voor expliciet door het management vastgestelde doeleinden. Een dergelijk bericht dient te worden gedefinieerd in `/etc/sttydefs of in /etc/motd`.
- Toon na het inloggen een bericht waarin staat wanneer voor het laatst is ingelogd en of sprake is geweest van mislukte inlogpogingen. Neem hiertoe het commando `last` op in `/etc/profile` (bij HP-UX is dit een standaard onderdeel van de C2 trusted mode).

Extra maatregelen zijn:
- Tref maatregelen om te voorkomen dat authenticatiegegevens onversleuteld over het netwerk worden verstuurd. Standaard biedt Unix hiertoe echter geen faciliteiten.
- Gebruik `TCP-wrappers` (aanbevolen).

6.2.3 File Transfer Protocol (FTP)

Met het File Transfer Protocol (FTP) kan de gebruiker bestanden van en naar 'remote' computersystemen transporteren. Het gebruik van FTP is aan dezelfde beper-

kingen gebonden als `login` en `telnet`, aangezien eerst een geldige verbinding moet zijn gemaakt door aanloggen.
Het Trivial File Transport Protocol (TFTP) is een primitieve en onveilige versie van FTP, aangezien TFTP geen gebruikmaakt van een authenticatiemechanisme. TFTP wordt echter in een aantal gevallen gebruikt in het bootproces van een werkstation. Daarbij wordt met TFTP van een read-only beveiligd deel van het file system het bootimage voor de werkplek opgehaald, hetgeen op zich geen risico hoeft te zijn.
Met FTP is het mogelijk om directory's en bestanden op een FTP-server ter beschikking te stellen aan FTP-clients of om bestanden te plaatsen op een server. Het anoniem ophalen van bestanden (anonymous FTP) is een veelgebruikte faciliteit op Internet. Een voordeel van het gebruik van anonymous FTP is dat geen transport van wachtwoorden over een netwerk plaatsvindt. Nadeel is dat geen audit trail bestaat. Overigens vraagt de netiquette dat bij gebruikmaking van anonymous FTP de gebruiker het eigen e-mailadres als wachtwoord invoert. Hieraan vallen natuurlijk geen rechten te ontlenen.
In het bestand `~/.netrc` kan de gebruiker aangeven op welke computersystemen met welk wachtwoord moet worden ingelogd.

De FTP-deamon staat toe om een onbeperkt aantal keren te proberen aan te loggen met een onjuiste user-id/wachtwoordcombinatie. Deze daemon is daardoor ten aanzien van authenticatie minder betrouwbaar dan `login`. Hiermee bestaat dus een risico met betrekking tot het via een brute-force-techniek kraken van wachtwoorden. Wanneer FTP onvoldoende is beveiligd, kunnen ongeautoriseerde gebruikers op niet toegestane wijze bestanden van en naar het systeem verplaatsen. Anonymous FTP en TFTP kennen daarbij geen authenticatiemechanisme, waardoor geen audit trail achterblijft. Ditzelfde probleem speelt wanneer gebruik wordt gemaakt van `~/.netrc`, aangezien ook in dat geval authenticatie achterwege blijft. In `~/.netrc` staan de accountnamen en wachtwoorden in onversleutelde vorm. Het gevaar bestaat dat een ongeautoriseerde gebruiker dit bestand leest en de informatie gebruikt om onder andermans account toegang tot een ander systeem te krijgen.

6.2.4 Normen en maatregelen FTP

Basisnormen voor het gebruik van FTP zijn:

- Ongeautoriseerde toegang tot directory's en bestanden dient te worden voorkomen.
- Wachtwoorden mogen door niemand achterhaald kunnen worden.

Basismaatregelen zijn:

- Installeer hulpmiddelen om een lineair oplopende vertraging tussen FTP-inlogpogingen te realiseren, of maak gebruik van tools als de public domain programma's wu-ftpd of proftpd die zodanig in te richten zijn dat een audit trail wordt vastgelegd (zie bijlage 2).
- Het gebruik van FTP is op eenvoudige wijze beter te beveiligen door in `/etc/ftpusers` alle systeemaccounts te vermelden. Hieronder vallen in ieder geval: root, bin, uucp, daemon, news, nobody en alle standaardaccounts. Dit bestand bevat namelijk alle accounts die geen gebruik mogen maken van FTP.
- Sta niet zonder meer anonymous FTP toe. Wanneer de functionaliteit gewenst is, beperk dan de rechten van de (anonieme) gebruikers door het `chroot` -commando, waardoor ze alleen in een afgeschermd deel van het file system terecht kunnen. Bedenk dat binnen dit afgeschermde deel van het file system een gestripte versie van het operating system moet staan (dat houdt in dat een versie van Unix zonder programma's en diensten met hoge autorisaties en risico's beschikbaar moet zijn).
- Toon voor het inloggen een bericht waarin duidelijk wordt gemaakt dat alleen geautoriseerde gebruikers mogen inloggen en dat het systeem alleen gebruikt mag worden voor expliciet door het management vastgestelde doeleinden.
- Wanneer gebruik moet worden gemaakt van TFTP (bijvoorbeeld in het geval van het gebruik van schijfloze werkstations), initialiseer deze service dan in `/etc/inetd.conf`, onder rootrechten met daarbij het chroot commando.
- Controleer periodiek (bijvoorbeeld met behulp van cron) op de aanwezigheid van `~/.netrc` -bestanden, analyseer de herkomst van deze bestanden en verwijder ze.
- Let erop dat de FTP-server geen SITE EXEC (het door de FTP-deamon op de server laten uitvoeren van commando's) ondersteunt.

Extra maatregelen zijn:

- Maak gebruik van TCP-wrappers (zoals `tcp-wrapper` of `logtcp`) om aanvullende toegangsbeveiliging te realiseren.
- Activeer een vorm van logging door het instellen van bepaalde flags (mogelijk in de meeste moderne versies van Unix).

6.2.5 Simple Mail Transfer Protocol (SMTP)

Met behulp van het Simple Mail Transfer Protocol (SMTP) kan de gebruiker e-mailberichten ontvangen en verzenden. Ontvangen berichten komen terecht in een persoonlijke postbus die met een mail-interface kan worden gelezen.

Een veel voorkomende implementatie van SMTP is het programma `sendmail`. Als `sendmail` een bericht ontvangt, wordt nagegaan of het bericht is bestemd voor een gebruiker onder zijn beheer. Blijkt dit het geval, dan wordt het in de postbus van deze gebruiker afgeleverd. Indien het bericht niet bestemd is voor een gebruiker onder beheer van de `sendmail`, dan wordt het bericht doorgestuurd naar `sendmail` op een ander computersysteem. Welk computersysteem dit is, hangt af van de routering die gedefinieerd is in het configuratiebestand `sendmail.cf`.
Hoewel `sendmail` een groot aantal risico's kent, is het in veel gevallen de enige applicatie die de noodzakelijk functionaliteit bevat. Zo kan alleen `sendmail` op een effectieve manier de headers van e-mailberichten aanpassen om te voorzien in de mogelijkheid van maskerade (het substitueren van de feitelijke e-mailhostnaam door een publiek beschikbare hostnaam), waardoor de configuratie van het interne netwerk niet wordt geëxposeerd.
Met de Multimedia Mail Extensions (MIME) kan de gebruiker plaatjes, programma's en andere binaire gegevens met e-mail versturen en ontvangen en, afhankelijk van de applicatie die wordt gebruikt, automatisch laten uitvoeren.

Kenmerkend voor `sendmail` is het herschrijven van headers van een bericht. De hierbij te gebruiken herschrijfregels bevinden zich in het configuratiebestand `sendmail.cf`. Dit bestand bevat een groot aantal andere configuratieparameters en wordt algemeen als onbeheersbaar beschouwd.
De authenticiteit en integriteit van berichten die met SMTP worden verstuurd, zijn niet gewaarborgd. Het is zeer eenvoudig berichten onder een valse naam te versturen. Dit kan bijvoorbeeld door te Telnetten naar poort 25 (de SMTP poort).
Bovendien is `sendmail` niet eenvoudig te parametriseren, waardoor onbewust vaak fouten worden gemaakt.
Door de complexiteit van het programma zijn er verschillende bugs aanwezig. De laatste is nog niet gevonden.

6.2.6 Normen en maatregelen SMTP

Basisnormen voor het gebruik van SMTP zijn:

- De authenticiteit en integriteit van e-mailberichten dienen gewaarborgd te zijn. Dit is echter niet met behulp van alleen de standaardmaatregelen te realiseren.
- `Sendmail` mag alleen onder user-id 0 (root) draaien als dat strikt noodzakelijk is. Met andere woorden: als ook lokale e-mail moet worden afgeleverd.
- Het gebruik van, laat staan het automatisch laten uitvoeren van,

programma's waarvan de authenticiteit en integriteit niet gewaarborgd zijn, is niet acceptabel.

Basismaatregelen voor het gebruik van SMTP zijn:
- Vervang `sendmail` door een betere mailer, bijvoorbeeld Postfix of procmail (zie bijlage 2).
- Configureer sendmail in `sendmail.cf` zodanig dat het niet onder user-id 0 draait. Draai sendmail alleen onder user-id 0 (root), als er ook lokale e-mail moet worden afgeleverd.
- Draai alleen de meest recente versie van `sendmail`. Raadpleeg hiertoe de leverancier en de CERT-site op internet.
- Zet het wizard password in `sendmail.cf` uit (of wijzig de regel in `sendmail.cf` in `OW`, waardoor niet ingelogd kan worden met het wizard password).
- Waarborg dat de gebruikte versie van sendmail geen debug mode ondersteunt door gebruik te maken van de meest recente versie van `sendmail`.
- Zorg ervoor dat elke mailbox eigendom is van een bestaand account. Stel de bestandspermissies van elke mailbox zo in, dat alleen de eigenaar deze kan lezen en schrijven.

Extra maatregelen kunnen zijn:
- Tref extra maatregelen voor het waarborgen van de authenticiteit en integriteit van berichten, zoals Pretty Good Privacy (PGP), Privacy-Enhanced Mail (PEM) of MIME Object Security Standard (MOSS) encryptietechnieken.
- Maak gebruik van een firewall met SMTP-proxyfunctionaliteit om de dienst te beveiligen.

6.3 R-UTILITIES

Met de door de universiteit van Berkeley ontwikkelde R-utilities kan de gebruiker onder andere inloggen op een ander computersysteem (`rlogin`, `rsh`), bestanden kopiëren over het netwerk (`rcp`) en kijken wie er ingelogd is (`rwho`).
Onderdeel van de R-utilities is een generiek mechanisme om de standaardauthenticatie op basis van wachtwoorden uit te schakelen. Door de naam van een 'remote'gebruiker in een configuratiebestand op te nemen, geeft men deze gebruiker toestemming om de R-utilities zonder verdere authenticatie te gebruiken. Door de naam van een 'remote'computersysteem in een configuratiebestand op te nemen, geeft men alle gebruikers van deze systemen toestemming om de R-utilities zonder nadere toestemming te gebruiken.

De betreffende configuratiebestanden zijn `/etc/hosts.equiv` en.`rhosts` bestanden in de home directory's van de gebruikers (`~/.rhosts`). In het bestand `/etc/hosts.equiv` kan men trusted hosts, trusted users en het '+'-teken opgeven:

- Trusted users: alle gespecificeerde 'remote' trusted users mogen zonder wachtwoord de R-utilities gebruiken.
- Trusted hosts: alle gebruikers van gespecificeerde 'remote' trusted hosts mogen zonder wachtwoord de R-utilities gebruiken.
- '+': alle gebruikers van alle 'remote' hosts mogen zonder wachtwoord de R-utilities gebruiken.

Het bestand `~/.rhosts` bevat namen van computers en gebruikers die door de gebruikers als trusted worden beschouwd. Dit bestand is te beschouwen als een persoonlijke aanvulling op `/etc/hosts.equiv`.

Users van de in `/etc/hosts.equiv` genoemde systemen kunnen zonder authenticatie inloggen op het betreffende systeem en daarmee de autorisatiepolitiek omzeilen. Bij onvoldoende afscherming zouden ongeautoriseerd systemen kunnen worden toegevoegd die daarmee wellicht onterecht worden vertrouwd. Doordat geen wachtwoorden over het netwerk worden verstuurd, wordt overigens wel het afvangen van wachtwoorden voorkomen.
Het bestand `~/.rhosts` kan in principe door elke gebruiker zelf worden aangemaakt en is daarmee risicovoller dan `/etc/hosts.equiv`. Elk exemplaar moet aantoonbaar noodzakelijk zijn. Wanneer in het bestand `~/.rhosts`, in de homedirectory van een gebruiker, de accountnaam van een andere gebruiker achter de hostnaam staat aangegeven, kan die andere gebruiker, zonder het opgeven van een wachtwoord, van dit account gebruikmaken.
Activeren van rwhod resulteert (zeker op omvangrijke systemen) in een enorm beslag op resources.

Solaris: Om de.`rhost & hosts. equiv` -faciliteit onklaar te maken voor `rlog ind`, moet de volgende regel `int /etc/pam.conf` worden uitgecommentarieerd:
`rlogin auth sufficient /usr/lib/security/pam rhost_auth.so.1`

6.3.1 Normen en maatregelen

Basisnormen voor het gebruik van de R-utilities zijn:

- Elke gebruiker dient zich met een uniek user-id aan te melden en met een alleen bij die gebruiker bekend wachtwoord te authenti-

ceren. Daarmee dient het gebruik van `/etc/hosts.equiv` en `~/.rhosts` vermeden te worden.
– Gebruik de R-utilities in principe niet.

Als basismaatregelen gelden:
– Gebruik in plaats van deze programma's liever de secure versies, ssh, slogin en scp.
– Motiveer, accordeer en documenteer het gebruik van de R-utilities.
– Gebruik de R-utilities alleen binnen een voldoende fysiek beveiligd netwerk.
– Controleer in dat geval periodiek het bestand `/etc/hosts.equiv` op ongeautoriseerde wijzigingen.
– Leg in richtlijnen vast dat het gebruik van `~/rhosts` -bestanden verboden is.
– Controleer periodiek met behulp van `cron` of het bestand `~/.rhosts` in de persoonlijke directory's van gebruikers aanwezig is. Zo ja, verwijder elk exemplaar. Let speciaal op.`rhosts` -bestanden in systeemdirectory's, en meer specifiek op `~root/.rhosts`. Wanneer deze bestanden toch aanwezig moeten zijn:
  • let erop dat alleen namen van vertrouwde/beheerde systemen genoemd zijn in het bestand `/etc/hosts.equiv`;
  • controleer of de permissies 600 zijn en of Root de eigenaar is;
  • zie erop toe dat er in `/etc/hosts.equiv` geen regel is met alleen een '+' teken;
  • stel vast dat de eerste regel in `/etc/hosts.equiv` niet begint met een '-' teken;
  • let erop dat er geen accountnamen staan vermeld in deze bestanden.

Extra maatregelen zijn:
– Filter de poorten 512 (rexec), 513 en 514 (rlogin en rsh) op de router om toegang van buiten de eigen omgeving te blokkeren.
– In verschillende versies bestaat de mogelijkheid om door het instellen van flags het gebruik van wachtwoorden af te dwingen.
– Maak gebruik van PAM en configureer het bestand `/etc/pam.d/rlogin` zodanig dat het via rlogin inloggen vanaf een niet-beveiligde terminal wordt voorkomen.

In veel gevallen zullen Unix-implementaties over systeemgrenzen heen reiken. Daarvoor zijn aanvullingen nodig om andere computers te benaderen (NFS), bestanden op te sporen (NIS) en netwerkadressen te interpreteren (DNS).

6.4.1 Network File System (NFS)

Via Network File System (NFS) kunnen Unix-computers (clients) bestanden op andere computers in het netwerk (servers) benaderen zonder dat de gebruiker op deze computers ingelogd hoeft te zijn. De client mount hiertoe één of meer file systems van de server, terwijl de server deze file systems voor derden ter beschikking stelt (exporteert).
De te exporteren file systems zijn gedefinieerd in het bestand `/etc/exports` op de NFS-server. Het bestand `/etc/exports` bevat directory-namen waaruit bestanden mogen worden geëxporteerd. Daarbij bestaat bijvoorbeeld de mogelijkheid om de bestanden read-only te exporteren, of de rechten van een superuser bij de export te beperken. Ook het gebruik van secure NFS wordt in dit configuratiebestand ingesteld.
Het account root van een client kan met het commando `mount -a` alle file systems die in het bestand `/etc/fstab` genoemd staan, mounten. In dit bestand kunnen zowel lokale als 'remote' file systems worden genoemd. Ook enkele andere opties kunnen in dit bestand worden opgegeven, zoals het blokkeren van `suid`-rechten.
De communicatie tussen NFS-client en NFS-server is gebaseerd op file handles. Dit zijn unieke, door het systeem gegenereerde, nummers, die elk bestand en directory op schijf identificeren. Bij het mounten van een file system controleert de server of de naam van de client voorkomt in `/etc/exports`. Is de client geautoriseerd om het gespecificeerde file system met de gespecificeerde access mode te benaderen, dan geeft de server de corresponderende file handle aan de client. Met deze file handle kan de client vervolgens de gewenste operaties op het betreffende bestand of directory uitvoeren.

Heeft een programma op een geïmporteerd file system het setuid-bit aanstaan, dan wordt zo'n programma bij aanroep uitgevoerd onder de user-id van de eigenaar. Dit kan leiden tot een ongewenste verruiming van de bevoegdheden.
Wanneer accountnamen, user-id's of group-id's op de gekoppelde systemen niet overeenstemmen, kan de situatie zich voordoen dat een ander dan de eigenaar onbedoeld alle rechten op bestanden verkrijgt.
Een file handle geeft de client toegang tot een bestand of directory op

de server. Deze file handle wordt echter in cleartext over het netwerk getransporteerd. Door het netwerk af te tappen kunnen deze file handles eenvoudig worden achterhaald. Daarnaast kunnen file handles door onvolkomenheden in de generatie van unieke nummers relatief eenvoudig worden geraden.

Solaris: file system exports worden gedefinieerd door middel van sharecommando's in `/etc/dfs/dfstab`; deze commando's worden bij het booten van het systeem, zie `share_nfrs` (1m). Een belangrijke shareoptie is 'nosuid'; deze optie maakt het creëren van set-uid/set-gid bestanden vanaf een client onmogelijk. Solaris heeft volledig support voor Secure RPC.

HP-UX: een belangrijke optie bij het importeren van een share is de optie: nosuid (het niet toestaan van setuid-executie op bestanden).

6.4.2 Normen en maatregelen NFS

Basisnormen voor het gebruik van NFS zijn:

- Gebruik NFS alleen in een vertrouwde omgeving.
- Accountnamen, user-id's en group-id's moeten overeenstemmen op de verschillende te koppelen systemen.

Basismaatregelen zijn:

- Gebruik NFS hoe dan ook alleen binnen een geïsoleerd en voldoende fysiek beveiligd netwerk.
- Tref in dat geval de volgende maatregelen:
  - Waarborg dat accountnamen en user-id's overeenstemmen.
  - Exporteer geen file systems met bestanden die voor iedereen schrijfbaar zijn.
  - Exporteer file systems read-only (zie de opties bij `/etc/exports`, zoals hieronder vermeld).
  - Exporteer geen systeemdirectory's, in ieder geval niet `/`, `/etc` en `/bin8`;
  - Stel het bestand `/etc/exports` als volgt in (alle veranderingen in het bestand hebben alleen effect nadat ' `exportfs` ' opnieuw gestart is):
    - permissies 644 (`rw-r—r—`);
    - stel de eigenaar van het bestand in op user-id 0 (root).
    - gebruik de optie `root=...` niet, aangezien daarmee de root op het systeem waarop het file system is gemount ook als root toegang heeft tot dat file system.
- Sta anonieme gebruikers niet toe. Voorzie ieder record in

`/etc/exports` van de optie `anon=-1`. Deze negatieve waarde leidt ertoe dat een foutcode resulteert indien een anonieme gebruiker een bestand remote wil mounten.
- Maak geen gebruik van de Linux squash-opties of van de secure optie (deze opties bieden slechts schijnveiligheid): zorg dat er geen 'localhost' entry in het bestand staat.
- Maak gebruik van de optie `access` (staat alleen toegang vanaf de vermelde hosts toe).
- Maak gebruik van de optie `ro` (exporteert een file system in read-only mode).
- Stel het bestand `/etc/fstab` als volgt in: voorzie ieder record met een NFS entry in `/etc/fstab` van de optie `nosuid`;
- Maak gebruik van de optie `ro`.
- Voorzie het mount commando van de optie `nosuid` wanneer eindgebruikers zelf kunnen mounten.

Een extra maatregel kan zijn:
- Maak gebruik van Secure NFS in combinatie met Secure RPC. Neem hiertoe contact op met de leverancier.

6.4.3 Network Information System (NIS)

In grote organisaties is het gedistribueerd gebruik van bestanden als `/etc/passwd` in de praktijk moeilijk haalbaar. Om die reden zijn er verschillende methodieken ontwikkeld om het gebruik en het beheer van dergelijke adresboeken (Yellow Pages) mogelijk te maken. De belangrijkste voorbeelden zijn de Directory Service implementaties van SUN (NIS en NIS+), NeXT (Netinfo) en OSF DCE (Distributed Computing Environment). Hier wordt als illustratie een korte beschrijving gegeven van NIS, aangezien dit systeem, dat ook binnen Linux bestaat, als standaard kan worden beschouwd, hoewel de tendens in de richting van DCE zonder meer herkenbaar is.
Centraal in NIS staat de NIS-map. Dit is een centraal bestand op de NIS-server, zoals een passwordbestand. De NIS-clients kopiëren selectief gegevens uit dit centrale bestand. De lokale bestanden worden hiertoe voorzien van het karakter '+'. De lokale software ziet hieraan dat NIS gebruikt moet worden om de inhoud van het bestand van de server op te halen. NIS-maps worden beheerd door een master server. Kopieën van NIS-maps kunnen worden verstrekt aan slave servers, die enerzijds een deel van het NIS-verkeer voor hun rekening kunnen nemen en anderzijds als back-up voor de master fungeren.

Als in de NIS-map `/etc/passwd` superusers zijn gedefinieerd, leidt het kraken van een superuserwachtwoord uit deze groep ertoe, dat

dit account op iedere computer uit het domein kan worden gebruikt.
De IP-adressen kunnen zijn gekoppeld aan een onjuiste symbolische naam of nickname, waardoor een verkeerde computer wordt bereikt.
Als een NIS-server in `/etc/passwd` is voorzien van het (standaard)record +::0:0:::, kan met de loginnaam '+' worden ingelogd als root!

HP-UX biedt zowel NIS als NIS+.

6.4.4 Normen en maatregelen NIS

Basisnorm voor het gebruik van NIS is:

– Aangezien gebruik wordt gemaakt van centraal vastgelegde (dus gedeelde) gegevens, dient er bewust te worden omgegaan met user-id's en wachtwoorden.

Basismaatregelen voor het gebruik van NIS zijn:

– Gebruik de meest recente versie van de directory service. Motiveer, accordeer en documenteer het gebruik.
– Gebruik een afzonderlijk computersysteem als server, zonder gewone accounts.
– Definieer één of meer slave-servers.
– Definieer gebruikers met user-id 0 (root) in het passwordbestand van elk clientsysteem. Gebruik voor de overige loginnamen in `/etc/passwd` een NIS-map.
– Zorg ervoor dat alle +-entries in het passwordbestand op de clients een ⋆ als wachtwoord hebben.
– Zorg ervoor dat het passwordbestand en het group-bestand op de NIS-server geen + in het veld username en groupname bevatten.
– Stel de permissies van de NIS-maps op de server zo in dat user-id 0 (root) de eigenaar is en dat alleen de eigenaar ze kan schrijven.

6.4.5 Domain Name Service (DNS)

DNS (Domain Name Server) is een service die Internetnamen naar IP-adressen vertaalt, en omgekeerd. De Berkeley-implementatie van DNS heet BIND (Berkeley Internet Name Domain server). DNS is een hiërarchische, gedistribueerde database die gebruikmaakt van DNS-servers. Een client kan aan zijn DNS-server een vertaling tussen hostname, IP-adres en aliasname vragen. Wanneer de eigen DNS-server geen vertaling kan geven, kan deze DNS-server op zijn beurt bij zijn eigen DNS-server de vertaling vragen.

Een alternatief voor het gebruik van DNS is `/etc/hosts`, waarin IP-adressen van de werkstations binnen het eigen domein kunnen worden opgenomen. Het beheer hiervan is echter complex op systemen met meerdere servers, aangezien niet gewaarborgd is dat elk `/etc/hosts` -bestand dezelfde inhoud heeft.
DNS werkt zonder authenticatie en is daarom relatief eenvoudig om de tuin te leiden. Daarnaast is DNS geliefd bij hackers omdat het protocol, indien niet zorgvuldig geconfigureerd, waardevolle informatie over de systemen en gebruikers binnen een netwerkomgeving kan verschaffen. DNS kan echter ook zodanig worden geconfigureerd dat zo weinig mogelijk informatie aan de buitenwereld wordt verstrekt, terwijl de systemen op het interne netwerk wel kunnen beschikken over de volledige informatie.
Wanneer de DNS-server niet beschikbaar is, of wanneer de DNS-configuratie onjuist is, zijn de systemen waarnaar verwezen wordt niet bereikbaar.

6.4.6 Normen en maatregelen DNS

Basisnorm bij het gebruik van DNS is:

– Het gebruik van DNS dient met de grootste zorg omgeven te zijn, te meer daar de configuratie van DNS een grote impact heeft op het beheer van het netwerk. Bij een onzorgvuldige inrichting bestaat de kans dat ongewenste informatie over de inrichting van het netwerk beschikbaar wordt gesteld aan de externe omgeving.

Als basismaatregelen gelden:

– Motiveer, accordeer en documenteer het gebruik van DNS. In een klein lokaal netwerk (bestaande uit slechts enkele servers) kan eventueel gebruik worden gemaakt van de mogelijkheid om verwijzingen naar bekende servers op te nemen in het bestand `/etc/hosts`. Wanneer meer dan een handvol servers in het netwerk benaderbaar moet zijn, is DNS feitelijk een noodzakelijk mechanisme om ten minste het beheer te kunnen blijven voeren.
– Gebruik DNS binnen een voldoende fysiek beveiligd netwerk.
– Configureer DNS in de vorm van twee servers (split DNS). Hierbij verzorgt een server binnen het interne netwerk (dat wil zeggen achter een firewall) de verstrekking van adresinformatie aan interne hosts. Een externe DNS-server, die op de firewall of binnen een DMZ (demilitarized zone) draait, zorgt voor de verstrekking van adresinformatie aan de buitenwereld, maar laat hierbij zo min mogelijk informatie over het interne netwerk los.

Extra maatregelen zijn:

- De meest recente versie van BIND bevat (naast meer geavanceerde faciliteiten) mogelijkheden voor authenticatie en autorisatie. Het is raadzaam om gebruik van deze laatste versie te overwegen.
- Het onzichtbaar maken van de interne structuur van het eigen netwerk is onder meer te realiseren door gebruik te maken van Network Address Translation (NAT), een techniek waarbij een vertaalslag plaatsvindt van de adressen binnen het interne netwerk naar een extern bekend netwerkadres. Hiermee wordt effectief bereikt dat de structuur en inrichting van het interne netwerk van buitenaf niet zichtbaar is. Door gebruikmaking van applicaties level proxy servers kan overigens op een eenvoudige wijze automatisch de gewenste adres vertaling worden gerealiseerd.

6.5 X WINDOW SYSTEM

Het X Window System is een grafische gebruikersomgeving die is ontwikkeld door het Massachussetts Institute of Technology (MIT). Met X kunnen applicaties via het netwerk gebruikmaken van een werkstation met een grafisch display met venstertechnieken en een muis. X is een client/serversysteem, waar de betekenis van de woorden client en server iets anders is dan gebruikelijk: de X-client is de applicatie, die meestal draait op een centrale server, terwijl de X-server een programma is dat draait op een grafisch werkstation en het beheer voert over het beeldscherm. De X-server ontvangt van de X-client opdrachten om de grafische gebruikers interface af te beelden en stuurt daarmee het beeldscherm aan. Daarnaast ontvangt de X-server muisbewegingen en toetsaanslagen van de gebruiker en stuurt deze naar de X-client. Voor het beheer van de vensters maakt de server gebruik van een Window Manager, een programma dat de feitelijke grafische gebruikersinterface afbeeldt.
De X-server houdt bij welke clients toegang tot de server hebben. Standaard vindt deze autorisatie plaats door middel van het commando `xhost`. Autorisaties zijn geldig per clientsysteem, hetgeen inhoudt dat alle clientapplicaties op het clientsysteem toegang hebben tot de server. Toegang tot de server houdt hierbij onder andere in dat de client willekeurige delen van het scherm kan kopiëren en wijzigen, en dat hij toetsaanslagen af kan vangen - én kan genereren.
Vanaf versie X11R4 bestaat de mogelijkheid om met het commando `xauth` toegangspermissies per client-gebruiker in te stellen. Hierbij vindt de authenticatie van client-gebruikers plaats door middel van een unieke tekenreeks, het zogenaamde magic cookie. Dit magic cookie wordt onversleuteld over het lokale netwerk getransporteerd.

Een ander risico is dat X-terminals voorzien kunnen zijn van lokale opslagmedia. Ongeautoriseerde personen zouden toegang kunnen krijgen tot gegevens die aanwezig zijn op het werkstation.

Solaris: standaard wordt gebruikgemaakt van 'magic cookie' authenticatie.

6.5.1 Normen en maatregelen

Basisnorm voor het gebruik van X Window is:

– X-terminals moeten minimaal hetzelfde beveiligingsniveau hebben als reguliere werkstations.

Basismaatregelen zijn:

– Gebruik X alleen binnen een geïsoleerd en voldoende fysiek beveiligd netwerk.
– Gebruik, indien mogelijk, `xauth` en magic cookies om toegangspermissies per gebruiker in te stellen en niet het meer kwetsbare `xhost` -programma, aangezien daarmee iedere gebruiker van een systeem toegang kan krijgen tot een server.
– Maak gebruik van een screensaver met wachtwoordbeveiliging.
– Registreer alle verstrekte permissies.

Mogelijke extra maatregelen zijn:

– Overweeg het gebruik van Kerberos in combinatie met X Window.
– Maak gebruik van een hulpmiddel als SSH om informatie versleuteld over het netwerk te transporteren.

6.6 UUCP

Ook UUCP (een afkorting van Unix to Unix Copy Protocol) behoort tot de standaardnetwerkapplicaties. UUCP maakt in de regel echter geen gebruik van TCP/IP, maar van telefoonverbindingen. Er bestaan in de praktijk twee versies van UUCP: Release 2 UUCP en HoneyDanBer UUCP (ook wel BNU UUCP of New UUCP genoemd). Door de toename van het gebruik van de Internet-faciliteiten kan ervan worden uitgegaan dat UUCP in de praktijk nauwelijks meer wordt gebruikt. Tot UUCP behoren onder andere de volgende commando's:

– `uucp` (voor remote copy, filetransfer);
– `uux` (voor remote execution);

– `cu` (voor remote login);
– `mail` (voor remote mail).

UUCP is een store-and-forwardprotocol. Bestanden worden in hun geheel overgestuurd en al dan niet tijdelijk op disk opgeslagen. Elk knooppunt in een UUCP-netwerk kent alleen de systemen waarmee het een directe verbinding heeft; daarom moet een UUCP-gebruiker steeds het gehele pad naar de ontvanger van het bestand opgeven. De UUCP-commando's loggen op het doelsysteem in met behulp van het bijbehorende proces `/usr/lib/uucp/uucico` dat als shell fungeert.
Traditioneel zijn er voor UUCP twee loginnamen gespecificeerd:
– uucp: de eigenaar van UUCP-bestanden;
– nuucp: een loginnaam op een server, waarmee een client-uucico op de server kan inloggen.

Bij alle Unix-versies staat relevante informatie vermeld in de bestanden `Systems`, `Devices`, `Dialers` en `Permissions` in de subdirectory `/usr/lib/uucp`.
In het bestand `Systems` staat onder andere vermeld vanaf welke nodes een verbinding naar deze computer mogelijk is. Aanvullend kan gespecificeerd worden op welke tijdstippen zo'n sessie mag plaatsvinden. Ook staat vermeld met welke nodes een verbinding vanaf de eigen computer mogelijk is. In dit geval dient tevens vermeld te worden hoe deze node bereikt wordt (via een verwijzing naar het bestand devices), hoe daarop moet worden ingelogd (chat-script) en welk login-id en wachtwoord daarbij dienen te worden gebruikt.
Een record in het bestand `Devices` beschrijft een type netwerk dat kan worden gebruikt, waarbij ook vermeld staat (door middel van een verwijzing naar het bestand dialers) welke acties dan ondernomen moeten worden om stap voor stap tot aan de loginsessie van de server te komen.
Het bestand `Dialers` bevat records die ieder een chat-script (voor bijvoorbeeld een modem) bevatten die één barrière neemt, om tot aan de loginsessie van de server te komen.
Het bestand `Permissions` geeft aan welk deel van het file system gelezen mag worden door `uucico`. Deze autorisaties vormen een aanvulling op de betreffende directory- en bestandsautorisaties.

In het bestand `/usr/lib/uucp/L.cmds` worden de door UUCP (in casu `uux`) op het remote-systeem uit te voeren commando's opgenomen. Een van de in dat bestand opgenomen programma's is `rmail`, waardoor in ieder geval de via UUCP verzonden mail op het remotesysteem is te ontvangen.

UUCP is een eenvoudig communicatieprotocol, zonder veel waarborgen. Het risico bestaat dat bij verkeerd adresseren bestaande bestanden worden overschreven op de doelmachine.
Op de machine waarmee contact wordt gezocht, wordt als user-id uucp of nuucp gebruikt. Zeker bij gebruik van `uux` maakt dat een goede bescherming noodzakelijk.

6.6.1 Normen en maatregelen

Voor het gebruik van UUCP is de basisnorm:

– UUCP dient in principe alleen voor maildoeleinden te worden gebruikt. Als UUCP daarvoor niet wordt aangewend, dan dient deactivering overwogen te worden, gezien de complexiteit die de beveiliging met zich meebrengt.

Basismaatregelen ten aanzien van UUCP zijn:

– Ga na of van UUCP gebruik wordt gemaakt. Deactiveer het wanneer van Internet gebruik kan worden gemaakt.
– Gebruik UUCP in principe alleen voor het verzenden en ontvangen van mail. Voorkom in ieder geval het gebruik van `uux`, aangezien daarbij vrijwel ongecontroleerd programma's vanaf een remotemachine kunnen worden uitgevoerd. Motiveer, accordeer en documenteer elke uitzondering op deze regel. In dat geval geldt voor de instellingen van de bestanden `L.sys` (UUCP release 2) en `~/uucp/Systems` (BNU):
  • neem alleen nodes op waarvoor het toegestaan is dat zij een UUCP-sessie met de eigen computer aangaan, en vice versa;
  • gebruik het veld `time` om UUCP-sessies buiten kantoortijd te voorkomen.
– Stel de bestandspermissies van `L.sys` en `~/uucp/Systems` zo in dat alleen de eigenaar ervan (`uucp`) het bestand kan lezen en schrijven.
– Ieder aangesloten computersysteem dient een eigen loginnaam en wachtwoord voor UUCP te hebben. Verwijder het account nuucp.
– Bescherm het account uucp met een wachtwoord.
– Zorg ervoor dat het bestand `L.cmds` als eigenaar user-id 0 (root) heeft.
– Neem in `L.cmds` alleen de noodzakelijk commando's op. Bij twijfel kan het bestand worden leeggemaakt (laat het wel bestaan, anders zal UUCP de defaultinstellingen gebruiken).

Een denkbare extra maatregel is

– Beveilig het gebruik van UUCP met een dial-backfaciliteit.

7 Dagelijks beheer

Naast een adequate toegangsbeveiliging dienen ook de integriteit en de authenticiteit van de gebruikte programmatuur te worden gewaarborgd. Dit hoofdstuk beschrijft basisnormen, basismaatregelen en extra maatregelen om dit doel te benaderen. Hierbij wordt aandacht besteed aan de installatie, het operationele beheer, de automatische verwerking van programma's, de scheiding tussen ontwikkeling en productie en aan security patches. Daarnaast gaat dit hoofdstuk in op twee onderwerpen die betrekking hebben op de toepassing van het Unix systeem: firewalls en de inrichting als server voor (database)applicaties en webservices.

7.1 INSTALLATIE

Het installeren van een Unix-systeem is de eerste stap in de levenscyclus van een operationeel systeem. Nadat een Unix-systeem volgens de bijgeleverde instructies is geïnstalleerd – deze instructies zullen per leverancier verschillen – zal in veel gevallen 'customizing' plaatsvinden. Een onderdeel van deze customizing is het wijzigen van bepaalde beveiligingsinstellingen. Verdere customizing vindt plaats met verschillende tools, waarmee talloze scripts en configuratiebestanden worden aangepast, die verspreid over het gehele file system voorkomen.
Voor het handhaven van de beveiliging van een operationele Unix-omgeving is een gestructureerde aanpak en een juiste en volledige vastlegging van beveiligingsrelevante wijzigingen dus een eerste vereiste.

Het niet of onvoldoende nauwkeurig bijhouden van specifieke aanpassingen leidt ertoe dat geen inzicht bestaat – en geen uitspraak kan worden gedaan over – de beveiliging van het systeem. Dit bemoeilijkt vele praktische aspecten van het beveiligingsbeheer, met name in probleemsituaties en bij het overdragen van taken en bevoegdheden en het afleggen van verantwoording.
Systeemspecifieke instellingen kunnen bij een upgrade van het besturingssysteem of de applicaties verloren gaan, zonder dat dit tijdig wordt gedetecteerd.

De vereiste beveiligingsinstellingen kunnen achteraf worden gewijzigd, bijvoorbeeld om een 'achterdeur' in het systeem in te bouwen. Dit kan gebeuren door het aanbrengen van wijzigingen in de kernel van het besturingssysteem zelf, maar ook in applicaties, configuratiebestanden of shell scripts. Naast het instellen en bewaken van de juiste toegangspermissies (zie hoofdstuk 4) is dus ook het bewaken van de integriteit van programmatuur (zie paragraaf 7.2) en de correcte inhoud van configuratiebestanden van groot belang.

7.1.1 Normen en maatregelen

De basisnormen bij installatie zijn:

- De installatie dient met grote zorgvuldigheid plaats te vinden.
- Er dient altijd een juist en volledig overzicht te bestaan van wijzigingen met betrekking tot de beveiligingsinstellingen van het systeem. Daartoe dient ten minste een schriftelijk verslag van de installatie te worden opgemaakt, waarin alle voor de beveiliging relevante instellingen worden vastgelegd.

Basismaatregelen zijn:

- Voer de installatie volgens de richtlijnen van de leverancier uit. Maak in geval van een upgrade vóór de installatie een back-up van het oude systeem.
- Maak tijdens de installatie een verslag op. Dit verslag dient zowel elektronisch als off-line te worden opgeslagen en minimaal de volgende items te bevatten:
  - de Unix-versie, de leverancier en de bijbehorende utilities;
  - de datum en het tijdstip van de installatie, test en feitelijke ingebruikname;
  - de gehanteerde uitgangspunten ten aanzien van de toewijzing van bevoegdheden aan de verschillende (groepen) gebruikers;
  - de instelling van beveiligingsrelevante systeemparameters;
  - alle permissiebits van systeem- en applicatiebestanden en -directory's;
  - een checksum over de kernel en de belangrijkste tools.
- Bij de instelling van beveiligingsrelevante systeemparameters dient onder meer aandacht te worden besteed aan de volgende aspecten:
  - de gehanteerde regels bij het instellen van toegangspermissies;
  - de toegangspaden tot het systeem;
  - de ondersteunde netwerkapplicaties (`/etc/inetd.conf`);
  - een overzicht van alle setuid-programmatuur, inclusief de motivatie van de noodzaak ervan;
  - een overzicht van `/etc/inittab`;
  - een overzicht van `hosts.equiv` en `~/.rhosts`;
  - de waarde van `umask`.

– Wijzig direct na de installatie alle door de leverancier geïnstalleerde gebruikersnamen en wachtwoorden (zie ook hoofdstuk 4).

7.2 OPERATIONEEL BEHEER

Operationeel beheer omvat verschillende zaken die erop gericht zijn de integriteit van het systeem te waarborgen. Zelfs als de toegangspermissies (zie hoofdstuk 4) binnen een Unix-omgeving naar behoren zijn ingesteld, zijn er mogelijkheden waarlangs de beoogde functiescheidingen alsnog doorbroken kunnen worden. Om de integriteit van systeem- en applicatieprogrammatuur te waarborgen dient voorkomen te worden dat deze door onbevoegden wordt gewijzigd. Dit wordt bereikt door het instellen van de juiste bestandspermissies, het beperken van de permissies van gebruikers en het afschermen van commando shells.
Daarnaast kan de integriteit van systeem- en applicatieprogrammatuur worden bewaakt door het periodiek uitvoeren van integriteitscontroles. Zulke controles zijn doorgaans gebaseerd op controletotalen (checksums) over de inhoud van de betreffende bestanden. Unix biedt standaard geen faciliteiten voor automatische integriteitscontrole op basis van checksums. Het is mogelijk de integriteit van bestanden te bewaken aan de hand van kenmerken als bestandsgrootte, datum van wijziging of eigenaarschap, maar zulke controles kunnen relatief eenvoudig worden omzeild.
Wanneer onvoldoende zicht is op de aanwezige bestanden zou een indringer ongeautoriseerde applicaties, wellicht Trojan horses, kunnen installeren. Versiebeheer is minder goed mogelijk als er geen goede administratie van de geïnstalleerde programmatuur en configuratie daarvan is.

Twee specifieke onderwerpen in het operationele beheer verdienen extra aandacht: opstarten in single-user mode en secure terminal.
Unix kan worden opgestart in single-user mode, waarbij de gebruiker direct beschikt over root-privileges. In single-user mode ligt het hele systeem open voor analyse.
De meeste Unix-varianten bieden de mogelijkheid om de terminals waarop door root ingelogd kan worden, te beperken. Dit geeft de gelegenheid om fysiek te regelen dat slechts vanaf één 'veilige' locatie (zoals een fysiek beveiligd rekencentrum) door root kan worden aangelogd. Door het aanmaken en invullen van `/etc/securetty` kan de systeembeheerder aangeven op welke direct aangekoppelde terminals door root mag worden ingelogd. Sommige versies van Unix beschikken over een voorziening om een scherm te blokkeren als de

terminal enige tijd niet is gebruikt. Ook X-Windows voorziet in deze functionaliteit. Daarnaast bestaat er niet-commerciële software waarmee deze voorziening kan worden gerealiseerd. Unix biedt standaard geen faciliteiten waarmee de gebruiker zich kan overtuigen van de authenticiteit van het loginproces. Met andere woorden: de Unix-gebruiker weet nooit met zekerheid of het getoonde loginscherm een authentiek loginscherm is, of dat sprake is van een door een hacker geïnstalleerd Trojan horse dat erop is gericht accountinformatie (user-id's en wachtwoorden) te kapen.

Met het toenemen van de hoeveelheid servers en werkstations neemt de behoefte aan een eenduidig beheer en exploitatie van de verschillende systemen toe. Daarbij ontstaat ook de ontwikkeling van decentrale plaatsing van serversystemen en de centralisatie van de beschikbare deskundigheid met betrekking tot het beheer van steeds grootschaliger netwerken. Deze ontwikkelingen hebben geleid tot de ontwikkeling van Remote beheer tools. Enkele van deze systemen zijn CA-Unicenter, IBM Tivoli, HP Openview en BMC Patrol. Dergelijke systemen maken in het algemeen gebruik van specifiek voor beheer ontwikkelde protocols (te denken valt aan SNMP, Simple Network Monitoring Protocol). Kenmerkend voor deze tools is ook dat ze meer dan alleen een Unix-platform beheren, ook Netware, NT en OS/390 kunnen in het beheermechanisme worden opgenomen. De beveiligingsaspecten die spelen ten aanzien van deze hulpmiddelen worden in deze standaard niet uitgewerkt.

Solaris: single user boot vraagt altijd om het root password. Het is op Solaris slechts mogelijk om één, geen of alle terminals als secure terminal te definiëren. Dit wordt gedefinieerd in `/etc/default/login`. De basisinstelling definieert `/dev/console` als de enige secure terminal. De volgende zijn mogelijke instellingen in `/etc/default/login`:

- Alleen root logins op `/dev/console` toegestaan
  `CONSOLE=/dev/console`
- Geen root logins toegestaan (basismaatregel)
  `#CONSOLE=`
- Alle root logins toegestaan.
  `CONSOLE=`

Een boodschap weergeven voor het inloggen geschiedt door middel van `/etc/issue`.

HP-UX: Indien het systeem niet in single-user mode mag worden opgebracht, dient het systeem in C2-mode te staan.

7.2.1 Normen en maatregelen

De algemene basisnormen voor operationeel beheer luiden:

- Er dient enige vorm van integriteitscontrole plaats te vinden op essentiële systeem- en applicatieprogrammatuur en configuratiebestanden, met inbegrip van de configuratie van de kernel van het besturingssysteem.
- Alleen geautoriseerde programmatuur mag worden gebruikt.
- Realiseer een adequate change-managementprocedure met goede afspraken over versiebeheer.

Basismaatregelen die hierbij horen zijn:

- Inventariseer essentiële systeem- en applicatiebestanden.
- Registreer de belangrijkste kenmerken van deze bestanden, zoals bestandsgrootte, begin en eind van de inhoud, permissiebits, datum van laatste wijziging en eigenaarschap. Bereken, indien mogelijk, checksums over de inhoud en leg deze schriftelijk vast. Controleer dagelijks of deze kenmerken niet gewijzigd zijn.

Een extra maatregel is:

- Gebruik een betrouwbaar aanvullend product voor integriteitscontrole, zoals Tripwire.

Ten aanzien van single-user mode en secure terminal gelden aanvullend de volgende basisnormen:

- Opstarten in single-user mode dient alleen mogelijk te zijn voor bevoegde systeembeheerders.
- Voor het inloggen dient een bericht getoond te worden waaruit blijkt dat het systeem niet door onbevoegden of voor niet-goedgekeurde doeleinden mag worden gebruikt.
- Is een gebruiker ingelogd, dan dient de terminal niet onbeheerd, maar zeker niet onbeveiligd te worden achtergelaten.

Aanvullende basismaatregelen voor single-user mode en secure terminal zijn:

- Zorg ervoor dat het sleutelwoord '`secure`' uit het bestand `/etc/securetty` wordt verwijderd. In dat geval moet ook bij een boot in single-user mode het root-wachtwoord worden opgegeven (niet mogelijk in HP-UX).
- Vraag, indien bovenstaande maatregel niet mogelijk is, een nieuwe versie van `init` aan de leverancier. Deze versie dient ook bij een boot in single user mode om een rootwachtwoord te vragen.
- Zorg ervoor dat er geen secure terminals in `/etc/securetty` staan, zorg ervoor dat het bestand root als eigenaar heeft en dat de permissies ingesteld staan op 644.

– Toon voor het inloggen een bericht (bijvoorbeeld gedefinieerd in `/etc/motd`) waaruit blijkt dat het systeem alleen door bevoegde medewerkers voor goedgekeurde toepassingen mag worden gebruikt.
– Een scherm dient automatisch te worden geblokkeerd indien een zekere periode geen activiteit heeft plaatsgevonden.

Een mogelijke extra maatregel voor single user mode is:
– Gebruik een fysieke sleutel om het computersysteem tegen onbevoegd opstarten te beschermen.

7.3 AUTOMATISCHE VERWERKING

De achtergrondprocessen (daemons) `at` en `cron` voeren op vastgestelde tijdstippen programma's voor gebruikers uit.

CRON `Cron` is een achtergrondproces dat commando's periodiek op bepaalde tijdstippen uitvoert. In de directory `/var/spool/cron/crontabs` staan tabellen waarin is vastgelegd wat `cron` moet doen en wanneer. `Cron` zelf wordt gestart tijdens het bootproces door een commando in `/etc/inittab`.

AT Met `at` kunnen jobs eenmalig op een bepaald tijdstip worden gestart.

Ook niet-superusers kunnen van `cron` en `at` gebruikmaken. In de directory `/var/cron` bevinden zich de bestanden `cron.allow` en `cron.deny`. Het allow-bestand bevat een lijst van gebruikers aan wie het is toegestaan om een eigen crontab aan te maken. Het deny-bestand bevat een lijst van gebruikers aan wie het niet is toegestaan een crontab aan te maken. De volgende combinaties zijn van toepassing:
– als er geen allow-bestand is, mag iedereen, behalve zij die genoemd zijn in het deny-bestand, een eigen crontab aanmaken;
– als er geen deny-bestand is, mogen alleen zij die genoemd zijn in het allow-bestand, een eigen crontab aanmaken;
– als er geen allow- en deny-bestanden zijn, mag iedereen een eigen crontab aanmaken;
– als zowel het allow-bestand als het deny-bestand aanwezig is, wordt het deny-bestand genegeerd.

Daarnaast is er ook een `at.allow` en `at.deny` bestand met dezelfde eigenschappen.

Wanneer het systeem wordt gestopt en na enige tijd opnieuw wordt gestart, zullen de achterstallige `at-jobs` allemaal tegelijk starten, ter-

wijl de `cron-jobs` blijven wachten tot de klok weer het juiste tijdstip aanwijst. Daarmee zijn de geplande `cron-jobs` in feite dus vervallen. Crontabs zijn per gebruiker in een beschermde directory geplaatst. De jobs worden uitgevoerd onder het privilege van de eigenaar en niet onder het privilege van `cron`.
Bij System V Release 4 is van de aangeleverde `at-job` het setuid-bit automatisch aangezet. Dit voorkomt dat een gebruiker een job van zichzelf met het `chown` -commando 'weggeeft' aan een andere gebruiker, bijvoorbeeld root. De job zou dan te zijner tijd onder de privileges van root door `at` worden uitgevoerd. `Chown` zet het setuidbit altijd uit. Wordt de job 'weggegeven', dan zal dit bit niet aanstaan, waardoor `cron` weigert deze job uit te voeren.
Bij sommige andere Unix-systemen kan `chown` alleen door de superuser worden gebruikt, zodat hier geen risico ontstaat. De directory `/var/spool/at` mag alleen beschrijfbaar zijn voor user-id 0 (root), zodat de andere gebruikers niet via andere wegen dan het `at` -commando jobs kunnen aanleveren. Verder dient er niets in de background te draaien met meer privileges dan absoluut noodzakelijk is. Zo moeten applicatiespecifieke scripts draaien onder de user-id van de applicatie en niet onder de user-id 0 (root). De directory's waar deze scripts in staan, moeten zijn afgeschermd voor gewone gebruikers. De configuratiebestanden van `cron` en `at` moeten eveneens in een beveiligde directory staan.

Door verkeerd gebruik van `cron` en `at`, of het onbevoegd toevoegen van jobs, kunnen ongecontroleerde activiteiten uitgevoerd worden op tijden wanneer geen systeembeheerder aanwezig is. Deze activiteiten kunnen automatische taken hinderen.
Als er geen allow- en deny-bestanden aanwezig zijn terwijl het `cron` -mechanisme en de `at` -faciliteit beschikbaar zijn, heeft dit tot gevolg dat iedereen een `cron` kan aanmaken en hiermee buiten kantooruren programma's kan starten. Dit houdt het risico in dat programma's worden gestart op momenten dat de logging niet actief is.

Solaris: De `cron` -configuratiebestanden staan in `/etc/cron.d`. De standaardinstallatie bevat `at.deny` -bestanden en `cron.deny` -bestanden met een aantal systeemaccounts.

HP-UX: De allow- en denybestanden voor `cron` en `at` zijn te vinden in de directory `/usr/lib/cron`, evenals het logbestand.

7.3.1 Normen en maatregelen

Basisnormen zijn:

- `Cron` en `at` dienen alleen toegankelijk te zijn voor systeembeheerders.
- `Cron` en `at` dienen voor de juiste toepassingen en op de juiste wijze te worden gebruikt.

Als basismaatregelen gelden:

- `Cron` en `at` moeten worden gestuurd via allow-bestanden. Deze moeten dus altijd aanwezig zijn. In de allow-bestanden mogen alleen systeembeheerders gedefinieerd zijn. Uitzonderingen dienen individueel gemotiveerd, geaccordeerd en geregistreerd te worden.
- Activeer de `cron` -log. Deze log geeft aan welke processen door `cron` zijn gestart en op welk tijdstip. Tevens wordt aangegeven onder welke gebruikersnaam het proces loopt, met andere woorden: welke gebruiker het heeft gestart vanuit een crontab. Ook het tijdstip waarop het proces is gestopt wordt in deze log vastgelegd. De log staat in de directory `/var/cron`.
- De uitvoer van foutmeldingen van via `cron` en `at` verwerkte programma's dient via naar een bestand te worden weggeschreven. Indien dit bestand wordt gevuld, dan dient de systeembeheerder hierover automatisch e-mail te ontvangen.
- Bij de aanroep van programma's via `cron` en `at` dient het volledige path van het programma te worden gespecificeerd.

7.4 SCHEIDING TUSSEN ONTWIKKELING EN PRODUCTIE

Een van de aspecten van goed huisvaderschap is het inrichten van gescheiden omgevingen voor systeemontwikkeling, acceptatie en productie. Door het aanbrengen van een dergelijke scheiding wordt voorkomen dat ontwikkelactiviteiten de productie verstoren, dat ontwikkelaars toegang hebben tot productiebestanden en -applicaties, en dat niet formeel goedgekeurde programmatuur in productie wordt genomen.
Er zijn diverse manieren om deze scheiding te realiseren. In sommige gevallen wordt volstaan met het aanmaken van verschillende directory's voor de verschillende activiteiten. Een beter alternatief is het inrichten van afzonderlijke systemen of afzonderlijke lokale netwerken. In dat geval dient wel zorg gedragen te worden voor een adequate beveiliging van netwerkapplicaties (zie hoofdstuk 6).

Aan een slechte scheiding zijn verschillende risico's verbonden. Productiebestanden kunnen gecorrumpeerd worden door acceptatietesten en ontwikkelactiviteiten, wanneer geen goede scheiding van omgevingen en domeinen plaatsvindt. Nieuw ontwikkelde applicaties, die zonder formele acceptatie na een deugdelijke test worden ingevoerd, kunnen veranderingen introduceren die niet eenvoudig terug te draaien zijn.

7.4.1 Normen en maatregelen

Basisnormen inzake scheiding tussen ontwikkeling en productie zijn:

- Er dienen gescheiden computersystemen voor ontwikkeling, acceptatie en productie te zijn. Ontwikkelaars mogen geen toegang tot de productiesystemen of de gebruikersacceptatieomgeving hebben, om te voorkomen dat de productie wordt verstoord, of een geaccepteerde versie door een nieuw ontwikkelde wordt overschreven.
- De integriteit en authenticiteit van nieuwe programmatuur dient door een goede change-managementprocedure gewaarborgd te zijn.
- Programmatuur mag pas na uitvoerig testen en een formele acceptatie in productie worden genomen.

Basismaatregelen zijn:

- Installeer fysiek gescheiden computersystemen voor ontwikkeling, acceptatie en productie. Zorg ervoor dat ontwikkelaars geen toegang krijgen tot de productieomgeving.
- Laat softwareontwikkelaars nieuwe applicaties formeel aanleveren.
- Test en beoordeel de nieuwe applicaties en accepteer ze formeel. Bereken checksums over de geaccepteerde applicatiecode en leg deze schriftelijk vast.
- Draag de nieuwe applicaties formeel over naar het productiesysteem.
- Test de nieuwe applicaties uitvoerig op het productiesysteem en accepteer ze formeel. Bereken daarbij periodiek een checksum over de applicatiecode en controleer of deze checksum overeenkomt met de checksum die is vastgelegd tijdens de acceptatie.

Een aanvullende maatregel is:

- Maak gebruik van firewalls om de verschillende omgevingen te scheiden (zie paragraaf 7.6).

7.5 SECURITY PATCHES

Regelmatig worden in specifieke Unix-versies en applicaties nieuwe lekken ontdekt. In sommige gevallen leidt dit tot een waarschuwing (advisory) van het Computer Emergency Response Team (CERT), een van de vele coördinatiecentra voor het aanmelden van beveiligingsincidenten.
Om zulke lekken te dichten stellen leveranciers regelmatig patches op het besturingssysteem of onderdelen daarvan beschikbaar. Het nauwkeurig volgen en installeren van deze patches is noodzakelijk om de veiligheid van het besturingssysteem zoveel mogelijk te waarborgen en geen beveiligingsachterstand op te lopen. Kwaadwillende lieden lezen altijd de veiligheidswaarschuwingen en bedenken welke mogelijkheden die bieden.

Solaris: leverancier Sun stelt 'Security and Recommended' patches beschikbaar via het Web, op http: //sunsolve.sun.com/. Op deze Website is ook andere beveiligingsinformatie beschikbaar.

7.5.1 Normen en maatregelen

Basisnormen ten aanzien van security patches zijn:

- De systeembeheerder dient CERT-advisories en aanverwante informatiebronnen regelmatig te raadplegen.
- Security patches dienen nauwkeurig gevolgd en geïnstalleerd te worden.

Basismaatregelen

- Houd alle CERT-advisories en aanverwante informatiebronnen nauwkeurig bij. Installeer de noodzakelijke patches, mits de authenticiteit en integriteit ervan voldoende gewaarborgd zijn, bijvoorbeeld met behulp van een afzonderlijk aangeleverde digitale handtekening of een cryptografische checksum. Neem contact op met de leverancier en vraag hem om alle recente security patches.
- Controleer periodiek of alle noodzakelijke patches en de juiste versies van applicaties geïnstalleerd zijn.
- Sommige patches hebben invloed op de instelling van beveiligingsparameters. Controleer daarom na het installeren van een patch altijd of de beveiliging nog aan de vereiste normen voldoet.

7.6 FIREWALLBEVEILIGING

Unix kan ingezet worden als basis voor een firewall. Het systeem heeft op dat moment een 'enkelvoudige' functie en het dient dan zo ingericht te zijn dat het voldoet aan de (beveiligings)normen die gesteld worden aan de dienst die het moet leveren ten aanzien van de bedrijfsbehoeften. Bij het configureren van een Unix-server als drager van een firewallfunctionaliteit zijn enkele extra punten van belang.
In het algemeen kunnen voor een firewall de volgende functies worden onderkend:
- routering (router functie);
- IP-filter niveau (packet filter);
- applicatieniveau (proxy server);
- aanvullende functies als encryptie, content checking en alerting.

Deze functies kunnen zowel door een dedicated-firewallproduct op een afzonderlijke server worden uitgevoerd, als door verschillende producten op verschillende servers. Zowel de firewallprocessen, als de Unix-server dienen een hoog niveau van beveiliging te kennen, aangezien een firewall de vertrouwde (interne) omgeving afschermt van een externe, onvertrouwde omgeving.
Er wordt van uitgegaan dat de firewallapplicaties en de firewallconfiguraties voldoende veilig zijn geconfigureerd. Ook geldt als uitgangspunt dat de server voldoende veilig is geconfigureerd, onder andere dat er geen loginfaciliteiten worden geboden en dat er alleen leesrechten op een afgeschermd deel van het systeem aanwezig zijn.

Een firewall schermt het vertrouwde interne netwerk af van de onvertrouwde buitenwereld. Het compromitteren van de firewall leidt ertoe dat het interne netwerk niet langer beschermd is tegen aanvallen van buitenaf. Te denken valt aan de volgende risico's:
- de beschikbaarheid van het netwerk en netwerkcomponenten is niet langer gegarandeerd (denial of service);
- informatie komt beschikbaar voor onbevoegden: bedrijfsspionage, inbreuk op vertrouwelijkheid;
- informatie wordt ongeauthoriseerd aangepast of zelfs verwijderd: rancuneuze (ex-)medewerkers, inbreuk op betrouwbaarheid.

7.6.1 Normen en maatregelen

Basisnormen voor Unix als basis voor een firewall zijn:
- Een firewall is 24 uur per dag beschikbaar.
- Alleen bevoegde beheerders kunnen inloggen op het systeem.

Basismaatregelen zijn:
- Alleen beheerders voor het Unix-systeem en beheerders van de firewallconfiguratie mogen aanloggen om op systeemniveau hun werkzaamheden uit te voeren en zijn dus bekend in het bestand `/etc/passwd`.
- Alleen beheerders van de firewall hebben toegang tot de configuratiebestanden (rules) van de firewall.
- Alle Unix-services dienen uit te staan, behalve syslogd.
- Alle netwerkdiensten dienen uit te staan met uitzondering van SNMP vanaf het interne netwerk en TCP. Installeer hiervoor bijvoorbeeld TCP-wrappers en/of configureer de controlerende router.
- De firewall-server dient onder een applicatieaccount te draaien.
- De Unix-kernel moet na configuratie opnieuw worden gecompileerd.
- Maak gebruik van SNMP om te monitoren wat de status van de firewall is.

Extra maatregelen zijn:
- De logging van syslog dient zo mogelijk te worden weggeschreven op een logserver.
- Maak gebruik van real-time logging analysetools met een alert functie.

7.7 APPLICATIESERVERBEVEILIGING

Een Unix-systeem kan worden ingezet als applicatie- of databasemanagementsysteem. Het systeem heeft nu een 'enkelvoudige' functie en het dient dan zo ingericht te zijn dat het systeem voldoet aan de (beveiligings)normen die gesteld worden aan de dienst die het moet leveren ten aanzien van de bedrijfsbehoeften. Een dergelijk systeem moet gezien worden als het einde van een verwerkingsketen van aan elkaar gekoppelde systemen, dat wil zeggen de server in een client-serveromgeving. Op enkele plaatsen wordt extra aandacht geschonken aan een applicatieserver, die geplaatst kan worden tussen de client en de DBMS-server, de zogenaamde three-tierarchitectuur.

De databaseserver kan in vele gevallen de spil zijn in een IT-organisatie, waar alle diensten gebruik van maken. In een ERP-omgeving kan dit bijvoorbeeld veelvuldig voorkomen. Om onnodig risico's voor een organisatie te vermijden, dient extra aandacht aan de beveiliging van een dergelijk systeem te worden geschonken. Aan de volgende risico's kan worden gedacht:

– informatie komt beschikbaar aan onbevoegden (bedrijfsspionage);
– informatie wordt ongeautoriseerd aangepast of zelfs verwijderd door rancuneuze (ex-)medewerkers, hackers of bedieningsfouten;
– de applicatie zelf wordt onbruikbaar gemaakt, waardoor de hele organisatie zijn werkzaamheden niet meer kan verrichten, wat geld en mogelijk zelfs klanten kost.

7.7.1 Normen en maatregelen

Basisnormen voor de inzet van Unix op een applicatiesysteem zijn:

– Alleen bevoegde beheerders voor systeem en applicatie hebben toegang tot het systeem om hun werkzaamheden te kunnen uitvoeren en zijn dus bekend in het bestand `/etc/passwd`. De beveiligingsaspecten van de gebruikers van de applicatie dienen door de applicatie te worden ingevuld.
– Extra aandacht dient te worden gegeven aan de (mogelijk) toegevoegde netwerkservices die nodig zijn om de client/server-configuratie mogelijk te maken.
– Indien het systeem toegang heeft tot meer dan één logisch netwerksegment, zoals bij een applicatieserver, dan is het mogelijk dat het systeem informatie kan routeren op netwerkniveau. Dit moet voorkomen worden
– Bij de installatie van een applicatie of database wordt in vele gevallen gebruikgemaakt van aangereikte namen voor bijvoorbeeld de databasebeheerder en soms wordt zelfs een standaardpassword aangereikt, hetgeen onwenselijk is.
– Het kan zijn dat een systeem 'dubbel' moet worden uitgevoerd om de totale beschikbaarheid van een dienst te kunnen verzorgen. De invloed van mogelijke technische high-availability-oplossingen kunnen in het kader van de beveiligingseisen aan het systeem niet wenselijk zijn. De oplossing dient diepgaand te worden onderzocht op mogelijk onverwachte implicaties (systeem dubbel, dubbele netwerken, extra netwerk voor de communicatie tussen deze systemen, extra services).

Dit leidt tot de volgende basismaatregelen:

– Gebruikers van de applicatie mogen geen account op Unix-niveau hebben.
– Alle overbodige netwerkservices dienen uit te staan (zie hoofdstuk 6).
– Het systeem dient zodanig te worden geconfigureerd, dat routing onmogelijk wordt gemaakt: in het bestand `/etc/sysconfig/network` dient de parameter `IPFORWARDING=NO` te worden opgenomen.

– De standaard geconfigureerde gebruiker/beheerderaccounts dienen direct van een nieuw password te worden voorzien.

7.8 WEBSERVERBEVEILIGING

Een webserver is een (Unix-)machine die voor een specifieke taak (het afhandelen van http requests) in een minder veilige omgeving geplaatst is: de machine is namelijk zichtbaar vanaf Internet. De webserver is het Internetgezicht van een organisatie. Dit gezicht mag niet door onbevoegden worden veranderd. Om onnodige risico's voor een organisatie te vermijden, dient extra aandacht aan de beveiliging van een dergelijk systeem te worden geschonken. Aan de volgende risico's kan worden gedacht:
– informatie komt beschikbaar aan onbevoegden; bedrijfsspionage;
– informatie wordt ongeauthoriseerd aangepast of zelfs verwijderd: rancuneuze (ex-)medewerkers en hackers.

7.8.1 Normen en maatregelen

De beveiliging van webservers is een onderwerp op zich. Naast onderstaande zeer globale normen en maatregelen zij daarom verwezen naar [Van Dam].

Basisnormen zijn:
– De webserverapplicatie heeft slechts toegang tot een afgeschermd deel van het file system.
– De webpagina's mogen alleen gewijzigd worden door de beheerder van deze pagina's. Dit betekent dat de webserverapplicatie geen veranderingen mag toelaten.
– Alleen bevoegde beheerders (voor het systeem, de webserver en de webpagina's) kunnen inloggen op het systeem.
– De webserver dient gescheiden te zijn van het interne netwerk. Dit betekent dat de machine op een ander netwerk(segment) is aangesloten. Alle services behalve de webservice staan uit.

Hieruit volgen als basismaatregelen:
– Draai de server in een chrootomgeving onder een apart applicatieaccount.
– Maak het file system read-only voor het webserver applicatieaccount (bijvoorbeeld read-only mounten).
– De webserverapplicatie biedt geen loginfaciliteiten. Gewone gebruikers hebben geen account. De beheerders hebben een sterk wachtwoord, of een loginmethode die veiliger is dan het standaardloginproces.
– De machine kan alleen via HTTP worden bereikt. Alle overige

netwerkservices dienen uit te staan. Nieuwe webpagina's kunnen worden aangeleverd via bijvoorbeeld tapes.
- Het systeem wordt niet gebruikt als file server, database server, print server, enzovoort.

8 Conclusies

Dertig jaar na het ontstaan heeft het besturingssysteem Unix een sterke positie in de informatietechnologie verworven. Het besturingssysteem draait op vrijwel elk type computer: pc, werkstation, midrangesysteem, server, mainframe en supercomputer - en vormt de basis voor een grote verscheidenheid aan toepassingen, die in veel gevallen speciale beveiligingseisen met zich meebrengen.
Kenmerkend is nog steeds dat de Unix-systemen op een 'open' (dat wil zeggen slecht beveiligde) wijze worden afgeleverd door leveranciers, waarbij het aanbrengen van het vereiste beveiligingsniveau een taak van de systeembeheerder is. Deze taak kan de systeembeheerder slechts naar behoren uitvoeren wanneer er, op basis van het in de organisatie geldende informatiebeveiligingsbeleid, procedures en richtlijnen zijn vastgesteld voor de inrichting, het beheer, de beveiliging en de controle van de Unix-omgeving(en).
Het concept van Unix bevat voldoende mogelijkheden voor een systeembeheerder om een stand-alone Unix-omgeving afdoende te beveiligen. Moderne Unix-omgevingen staan echter niet meer op zichzelf, maar maken deel uit van uiteenlopende netwerkomgevingen.

Ondanks de beveiligingsmogelijkheden die Unix zelf biedt, moet men zich ervan bewust zijn dat het Unix-besturingssysteem een aantal fundamentele zwakheden ten aanzien van beveiliging in zich heeft:

- Unix is ontworpen voor gebruiksgemak, niet voor beveiliging. Hierdoor is binnen Unix het beveiligen van gegevens ondergeschikt aan de toegankelijkheid van gegevens.
- Unix security kent slechts twee uitersten: óf men is een gebruiker met beperkte bevoegdheden, óf men heeft de root-bevoegdheid en kan totale controle over het gehele systeem uitoefenen.
- Het 'open' karakter van Unix brengt met zich mee dat de beveiliging van het netwerk waarin de Unix-omgeving is opgenomen extra aandacht vereist.
- De meeste administratieve functies en parameters zijn buiten de (afgeschermde) kernel geïmplementeerd, zodat ze onderzocht, toegepast en/of gewijzigd kunnen worden. Hackers kunnen zich hiermee uitgebreide toegang tot het systeem verschaffen.

Samengevat geldt de stelling: 'Goede hekken zorgen voor goede buren'. Dit betekent dat in een moderne Unix-omgeving goede (security) tools ten behoeve van de systeembeheerders in combinatie met de juiste procedures en richtlijnen benodigd zijn om de Unix-omgeving toereikend te beveiligen.

Unix-systemen kunnen worden ingezet voor meerdere doeleinden, van kleinschalige omgevingen tot omvangrijke client/serverapplicaties, bijvoorbeeld als:
- applicatieserver;
- database server;
- webserver;
- loghost;
- firewall;
- authenticatieserver.

De aard van het gebruik bepaalt in belangrijke mate het vereiste beveiligingsniveau, zowel van het Unix-besturingssysteem als van de overige componenten waaruit de Unix-omgeving bestaat.
Een beveiligingsmaatregel genomen op een bepaalde component sluit de noodzaak/wenselijkheid van maatregelen betreffende een andere component geenszins uit. Het is de kunst om in de mix van mogelijke maatregelen de juiste balans aan te brengen zodat een toereikend beveiligingsniveau kan worden bewerkstelligd zonder 'overkill'.

De leidraad in dit boek biedt de individuele systeembeheerder de keuze om tijdens de werkuitvoering bepaalde maatregelen te implementeren of niet. Ook kunnen bepaalde maatregelen elkaar overlappen, opheffen of versterken. Het is daarom noodzakelijk om binnen de eigen organisatie eerst een vertaalslag te maken van het geldende informatiebeveiligingsbeleid op strategisch/tactisch niveau naar het operationele niveau. Bij de vaststelling of toepassing van de basisnormen voor een bepaalde Unix-omgeving zal altijd een specifieke invulling naar de eigen organisatie moeten worden gemaakt. Na de classificatie van de eigen informatiesystemen en de vastlegging van de eisen/wensen van de eindgebruikers in Service Level Agreements, zal een keuze moeten worden gemaakt voor het gewenste beveiligingsniveau. Dit betekent dat er kan of moet worden afgeweken van de basisnorm, zoals voorgesteld in deze standaard.
Wanneer de keuze is gemaakt voor specifiek op de eigen organisatie toegesneden maatregelen, zullen deze vastgelegd, beargumenteerd, geaccordeerd en geïmplementeerd moeten worden. Documentatie van de specifieke instellingen is nodig om deze te kunnen contro-

leren na wijzigingen op het systeem, het uitvoeren van periodieke (zelf)controles en ten behoeve van het samenstellen van (management)rapportages, Service Level-rapportages en audits.
In veel organisaties zijn de verschillende beheerrollen (zoals technisch beheer, functioneel beheer, productiebeheer) verenigd in één persoon. Vaak is het niet mogelijk vanwege de omvang van de organisatie om hier functiescheiding door te voeren. In deze gevallen dient er binnen de organisatie de keuze te worden gemaakt in welke mate de systeembeheerder het vertrouwen geniet en daarmee wordt bepaald welke aanvullende maatregelen (preventief en/of repressief) dienen te worden genomen om de gebrekkige functiescheiding te compenseren. Binnen deze maatregelen dient een balans te worden gevonden tussen afwezigheid van aanvullende maatregelen en de volledige controle van alle systeembeheerdersactiviteiten. Tabel 10 biedt een overzicht van de belangrijkste risico's en maatregelen die in dit boek aan de orde kwamen.

Bij de implementatie van een Unix-omgeving moet een evenwicht worden gevonden tussen risico's en het daarmee benodigde beveiligingsniveau, het gebruiksgemak, de invoerings- en beheerkosten alsmede de gevolgen voor de prestaties van de Unix-omgeving. Daarnaast moet echter benadrukt worden dat voor het beveiligen van Unix-omgevingen een juiste configuratie van het besturingssysteem wel noodzakelijk, maar niet voldoende is. Om te kunnen voldoen aan de principes van goed huisvaderschap blijven algemene maatregelen, zoals gedefinieerd in de Code voor Informatiebeveiliging, onverminderd noodzakelijk. Voorbeelden hiervan zijn het formuleren van een beveiligingsbeleid, het definiëren van procedures en richtlijnen en het controleren van de naleving hiervan, het realiseren van netwerkisolatie, het realiseren van een adequate fysieke beveiliging, het treffen van personele beveiligingsmaatregelen (onder andere 'security awareness') en het regelen van voorzieningen voor uitwijk, vervanging en/of reparatie van apparatuur bij calamiteiten.

TABEL 10

Onderwerp	Risico's	Maatregelen
Gebruikersbevoegdheden	– misbruik van accounts en/of te ruime bevoegdheden – onjuiste inrichting gebruikers-omgeving met als gevolg ongewenste effecten – misbruik van wachtwoorden – onjuiste inrichting van geauto-matiseerde authenticatiemecha-nismen zoals PAM	– beveiliging van het password-bestand – richtlijnen voor aanbrengen, wijzigen en verwijderen van accounts en indeling in groepen – richtlijnen voor het inrichten van login shells, home directory's, het aanmaken van nieuwe bestanden en de path-variabele – periodiek bekrachtigen autorisaties door eigenaar – beveiliging van het password-bestand – gebruikmaking van shadow password file – password encryptie – richtlijnen ten aanzien van het personeel (o.a. passworddiscipline) – nauwkeurige installatie en beveiliging van desbetreffende programmatuur en bestanden – geen openstaande sessies onbeheerd laten
Objecten	– benadering bestanden door onbevoegden – verkrijging van (te) hoge privileges	– instellen juiste toegangspermissies – richtlijnen en controle op gebruik van setuid en setgid. – toepassen bevoegdhedenmatrix o.b.v. 'least privilege'
Logging en monitoring	– niet of te laat opgemerkte beveiligingsincidenten – ongeautoriseerde raadpleging of wijzigingen in logbestanden – negatieve beïnvloeding van de performance – onvoldoende zekerheid over het juiste beveiligingsniveau	– structurele analyse van de logbestanden – beveiliging logbestanden en bewaking van het gebruik van root-permissie – vaststellen van een audit- en loggingstrategie – documentatie en argumentatie van de gekozen beveiligingsinrichting – periodiek uitvoeren van audits onder andere op basis van vooraf ingestelde logging- en auditopties

→

(VERVOLG) TABEL 10

Onderwerp	Risico's	Maatregelen
Netwerkdiensten	– onjuiste/onvolledige authenticatie – ongewenste verruiming van bevoegdheden – ongeautoriseerde toegang tot het netwerk door buiten-staanders (intern en extern)	– stringente naleving van identificatie/authenticatie richtlijnen – expliciete richtlijnen met betrekking tot gebruik en beveiliging van netwerkdiensten – expliciete autorisatie van het gebruik van bepaalde netwerkdiensten – deactivatie niet gebruikte netwerkdiensten – bewaking van het gebruik van netwerkdiensten met behulp van gespecialiseerde software – gebruikmaking van packet filtering
7 Dagelijks beheer	– onvoldoende inzicht in de beveiliging van het systeem – productieverstoringen vanwege: • uitgevoerde wijzigingen • automatische verwerking jobs	– structurele vastlegging van installatiegegevens en de wijzigingen daarop – documentatie en argumentatie van de gekozen beveiligingsinrichting – documentatie beheerprocedures – versiebeheer applicatie-programmatuur – scheiding van ontwikkel- en productieomgeving – toezicht op gebruik root-bevoegdheid

Overzicht van belangrijkste risico's en maatregelen

BevPr	*Beveiliging van Persoonsregistraties*, Registratiekamer, 1994.
Bruce	Glen Bruce and Rob Dempsey: *Security in Distributed Computing,* Hewlett-Packard Professional Books / Prentice Hall, 1997, ISBN 0-13-182908-4.
CCITS	*Common Criteria for Information Technology Security Evaluation* (parts 1-3), version 0.9, October 1994.
Code	*Code voor Informatiebeveiliging*, herziene versie, Nederlands Normalisatie-instituut, 1999.
CERTau	CERT advisories are available via anonymous FTP from ftp://ftp.auscert.org.au/pub/cert/cert_advisories/*. CERT vendor-initiated bulletins are available via anonymous FTP from ftp://ftp.auscert.org.au/pub/cert/cert_bulletins/*.
Chapman	Brent Chapman and Elizabeth Zwicky: *Building Internet Firewalls*, 1995, O'Reilly & Associates, Inc.
Cheswick	William R. Cheswick and Steven M. Bellovin: *Firewalls and Internet Security*, Addison-Wesley, 1994.
Curry1	D. Curry: *Improving the Security of your Unix System*, SRI International, 1989.
Curry2	David A. Curry: *Unix system security: A Guide for Users and System Administrators*, Addison-Wesley Professional Computing Series, May 1992.
Ford	Warwick Ford en Michael S. Baum: *Secure electronic commerce*, Prentice Hall, 1997, ISBN 0-13-476342-4.
Garfinkel1	Simson Garfinkel and Gene Spafford: *Practical Unix and Internet Security*, O'Reilly & Associates, Inc., 1996, ISBN 1-56592-148-8.
Garfinkel2	Simson Garfinkel with Gene Spafford: *Web Security & Commerce*, O'Reilly Associates, Inc, 1997, ISBN 1-56592-269-7.
Hare	Chris Hare en Karanjit Siyan: *Internet Firewalls and Network Security*, New Riders Publishing, 1996, ISBN 1-56205-632-8.
ITILSec	Jacques A. Cazemier, Paul L. Overbeek en Louk M.C. Peters: *IT Infrastructure Library Security Management,* The Stationary Office voor CCTA, 1999, ISBN 0-11-330014-X.
Lui	Cricket Lui, Jerry Peek, Russ Jones, Bryan Buus, Adrian Nye: *Managing Internet Information Services,* O'Reilly & Associates, Inc., 1994.
Linux	*LINUX Installation guide*, Red Hat 5.1, 1998, ISBN 1-888172-09-6.
Mckusick	Mckusick et. al.: *Design and implementation of the 4.4 BSD Operating System*, Addison-Wesley, 1996, ISBN 0-201-54979-4.
MinEZ	*Kruisverwijzingslijst Code voor Informatiebeveiliging*, Voorschrift Informatiebeveiliging Rijksdienst, Advies Beveiliging van Persoonsregistraties, Ministerie van Economische Zaken, 1996.

Nemeth	Evi Nemeth, Garth Snyder, Trent R. Hein and Scott Seebas: *Unix System Administration Handbook* (second edition, Prentice-Hall, Englewood Cliffs (NJ), 1995.
OpenG	The Open Group (X/OPEN OSF): *Security Survival, A source book from The Open Group*, Editor: Dean Adams, Prentice Hall, 1996, ISBN 0-13-266628-6.
Pipkin	Donald L. Pipkin: *Halting the Hacker, A practical guide to computer security*, Hewlett-Packard Professional Books / Prentice Hall, 1997, ISBN 0-13-243718-X.
Quinlan	Daniel Quinlan (editor), *Filesystem Hierarchy Standard*, Version 2.0, October 26, 1997, www.pathname.com.fhs.
Reid	J. Reid: *Open Systems Security: Traps and Pitfalls*, Proc. Compsec 1995, Elsevier Advanced Technology, 1995.
RFC1244	P. Holbrook, J. Reynolds: *RFC1244-Site Security Handbook*, 07/23/1991.
Saltzer	J. Saltzer and M. Schroeder: *The Protection of Information in Computer Systems*, Proceedings of the IEEE, Vol. 63, No. 9, September 1975.
Stern	Hal Stern: *Managing NFS and NIS*, O'Reilly and Associates, Inc., 1991.
Stoll	Clifford Stoll: *The Cuckoo's Egg*, Pocket Books, 1989, ISBN 0-671-72688-9.
TCSEC	*Trusted Computer System Evaluation Criteria*, Department of Defense 5200.28-STD, 1985.
USecC	*Unix Security Checklist*, European Security Forum, August 1995.
Van Dam	P.P.A. van Dam en anderen, *Internet, intranet en beveiliging: het technische kader*, Ten Hagen & Stam, 1998.
VIR	*Voorschrift Informatiebeveiliging Rijksdienst*, Ministerie van Binnenlandse Zaken, 1994.
Wood	Patrick Wood and Stephen Kochan: *Unix Systems Security*, 1986 Hayden Books.

Deze bijlage geeft aan hoe de in dit boek besproken maatregelen samenhangen met de aandachtspunten uit de *Code voor Informatiebeveiliging* (editie 1999). Per paragraaf in de code wordt aangegeven welke paragraaf in deze PI-standaard daarmee correspondeert. Uit de opzet van PI-standaard ('bottom-up', technisch georiënteerd) volgt dat niet alle Code-elementen worden geadresseerd en/of in een afwijkende volgorde. In een aantal gevallen is de behandeling overigens summier, en zullen de genoemde – al of niet specifieke – Unix-maatregelen op zich waarschijnlijk niet voldoende zijn voor een (uit hoofde van goed huisvaderschap of striktere) beveiliging.

Paragraaf in Code voor Informatievoorziening		Hoofdstuk/paragraaf in PI-standaard	
3	*Beveiligingsbeleid*	1	Inleiding
3.1	Informatiebeveiligingsbeleid		
3.1.1	Beleidsdocument voor informatiebeveiliging		
4	*Beveiligingsorganisatie*		
4.1	De infrastructuur van informatiebeveiliging		
4.1.1	Stuurgroep voor informatiebeveiliging		
4.1.2	Coördinatie van informatiebeveiliging		
4.1.3	Toewijzing van verantwoordelijkheden voor informatiebeveiliging		
4.1.4	Autorisatieproces voor IT-voorzieningen		
4.1.5	Specialistisch advies over informatie-beveiliging		
4.1.6	Samenwerking tussen organisaties		
4.1.7	Onafhankelijke beoordeling van informatie-beveiliging		
4.2	Controle op toegang door derden		
4.2.1	Identificeren van risico's van verbindingen met derden	6	Netwerkdiensten
4.2.2	Beveiligingsvoorwaarden in contracten met derden		
4.3	Uitbesteding van diensten		
4.3.1	Beveiligingseisen in uitbestedingscontracten		
5	*Classificatie en beheer van bedrijfsmiddelen*		
5.1	Verantwoording voor bedrijfsmiddelen		

5.1.1 Overzicht van bedrijfsmiddelen
5.2 Classificatie van informatie
5.2.1 Richtlijnen voor het classificeren
5.2.2 Classificatielabels

---

6 Beveiligingseisen ten aanzien van personeel
6.1 Beveiligingseisen in de functie-omschrijving en bij aannemen van personeel
6.1.1 Beveiligingseisen in de functieomschrijving
6.1.2 Screening van sollicitanten
6.1.3 Geheimhoudingsverklaring
6.1.4 Arbeidsvoorwaarden
6.2 Training voor gebruikers
6.2.1 Opleiding en training voor informatie-beveiliging
6.3 Reageren op beveiligingsincidenten
6.3.1 Het rapporteren van beveiligingsincidenten
6.3.2 Het rapporteren van zwakke plekken in de beveiliging
6.3.3 Het rapporteren van onvolkomenheden in programmatuur
6.3.4 Leren van incidenten
6.3.5 Disciplinaire maatregelen

5.3 Audit en monitoring

---

7 Fysieke beveiliging en beveiliging van de omgeving
7.1 Beveiliging van ruimten
7.1.1 Fysieke beveiliging van de omgeving
7.1.2 Fysieke toegangscontrole
7.1.3 Beveiliging van computercentra en computerruimten
7.1.4 Werken in beveiligde zones
7.1.5 Afzonderlijke ruimten voor aflevering van goederen
7.2 Beveiliging van apparatuur
7.2.1 Het plaatsen en beveiligen van apparatuur
7.2.2 Stroomvoorziening
7.2.3 Beveiliging van kabels
7.2.4 Onderhoud van apparatuur
7.2.5 Beveiliging van apparatuur buiten de onderneming
7.2.6 Veilig afvoeren van apparatuur
7.3 Algemene maatregelen

7.3.1	Clear Desk Policy		
7.3.2	Het verwijderen van bedrijfseigendommen		
8	*Computer- en netwerkbeheer*		
8.1	Bedieningsprocedures en verantwoordelijkheden		
8.1.1	Schriftelijke bedieningsprocedures		
8.1.2	Het beheer van wijzigingen		
8.1.3	Procedures voor het behandelen van incidenten		
8.1.4	Scheiding van functies	1	Inleiding
8.1.5	Scheiding van voorzieningen voor ontwikkeling en voor productie	7.4	Scheiding tussen ontwikkeling en productie
8.1.6	Beheer van externe voorzieningen		
8.2	Systeemplanning en acceptatie		
8.2.1	Capaciteitsplanning		
8.2.2	Acceptatie van systemen	7.1	Installatie
8.2.3	Uitwijkvoorzieningen		
8.3	Bescherming tegen kwaadaardige programmatuur	7.6	Passim, Firewallbeveiliging
8.3.1	Viruscontrole		
8.4	Huisregels		
8.4.1	Reservekopieën maken		
8.4.2	Bijhouden van een logboek	5.1	Standaard loggingsfaciliteiten
8.4.3	Storingen opnemen in een logboek	5.1	Standaard loggingsfaciliteiten
8.4.4	Klimaatbeheersing		
8.5	Netwerkbeheer		
8.5.1	Beveiligingsmaatregelen voor netwerken	6	Netwerkdiensten
8.6	Behandeling en beveiliging van computer-media		
8.6.1	Beheer van verwijderbare computermedia		
8.6.2	Afvoer van media		
8.6.3	Procedures voor de behandeling van gegevens		
8.6.4	Beveiliging van systeemdocumentatie		
8.7	Uitwisseling van gegevens	6 7	Netwerkdiensten, Dagelijks beheer
8.7.1	Overeenkomsten over het uitwisselen van gegevens		
8.7.2	Beveiliging van media tijdens transport		
8.7.3	EDI-beveiliging		
8.7.4	Beveiliging van elektronische post (intern en extern)	6.2.5	Simple Mail Transfer Protocol (SMTP)

8.7.5	Beveiliging van elektronische kantoor-systemen	7.7	Applicatieserverbeveiliging
8.7.6	Publiekelijk toegankelijke systemen (Internet)	6.1	De TCP/IP protocolsuite,
		6.2	Telnet, FTP en SMTP,
		6.4	Netwerk-ondersteunende diensten,
		7.6	Firewallbeveiliging,
		7.8	Webserverbeveiliging
8.7.7	Overige vormen van informatie overdracht	6.3	R-utilities,
		6.4	Netwerk-ondersteunende diensten,
		6.5	X Window System,
		6.6	UUCP
9	*Toegangsbeveiliging voor systemen*	2	Beveiligingsstructuur van Unix,
		3	Gebruikersbevoegdheden,
		4	Objecten
9.1	Eisen voor toegangsbeveiliging		
9.1.1	Beleid voor toegangscontrole		
9.2	Beheer van gebruikerstoegang en -bevoegdheden	3	Gebruikersbevoegdheden
9.2.1	Registratie van gebruikers	3.2	Accounts
9.2.2	Beheer van speciale bevoegdheden	3.2	Accounts
9.2.3	Beheer van gebruikerswachtwoorden	3.1	Wachtwoorden
9.2.4	Controle op toegangsrechten	2.1	Gebruikers en processen,
		3.1	Wachtwoorden,
		3.3	Pluggable Authentication Modules,
		3.4	Inrichting van de gebruikersomgeving
9.3	Verantwoordelijkheden van gebruikers		
9.3.1	Gebruik van wachtwoorden		
9.3.2	Onbeheerde gebruikersapparatuur		
9.4	Toegangsbeveiliging voor netwerken	3	Gebruikersbevoegdheden,
		6	Netwerkdiensten
9.4.1	Beleid voor het gebruik van netwerk-diensten		
9.4.2	Verplichte route		
9.4.3	Authenticatie van gebruikers via externe verbindingen		
9.4.4	Authenticatie van computers	6.4	Netwerk-ondersteunende diensten
9.4.5	Beveiliging van netwerktoegang op afstand	6.1	De TCP/IP protocolsuite,
		6.3	R-utilities,
		6.4	Netwerk-ondersteunende diensten,
		6.5	X Wondw System,
		6.6	UUCP
9.4.6	Segmentering in netwerken		

9.4.7	Beheersing van netwerkroutering		
9.4.8	Beheer van netwerkrouting		
9.4.9	Beveiliging van netwerkdiensten		
9.5	Toegangsbeveiliging voor besturingssystemen	2 3 4	Beveiligingsstructuur van Unix, Gebruikersbevoegdheden, Objecten
9.5.1	Automatische identificatie van werkstations		
9.5.2	Aanlogprocedures voor werkstations	3.4.1 4.1	Login shell, Inrichting van het file system
9.5.3	Gebruikersidentificatie en authenticatie	3.2	Accounts
9.5.4	Wachtwoordbeheersysteem	3.1	Wachtwoorden
9.5.5	Het gebruik van systeemhulpmiddelen	3.3 4.3	Pluggable Authentication Modules, Setuid, setgid en sticky bit
9.5.6	Stil alarm		
9.5.7	Time-out voor werkstations		
9.5.8	Beperking van verbindingstijd		
9.6	Toegangsbeveiliging voor applicaties	2 3 4	Beveiligingsstructuur van Unix, Gebruikersbevoegdheden, Objecten
9.6.1	Beperking van toegang tot informatie	3.2	Accounts
9.6.2	Isolatie van gevoelige systemen		
9.7	Bewaking van toegang tot en gebruik van systemen		
9.7.1	Vastlegging van gebeurtenissen	5	Logging en monitoring
9.7.2	Bewaking van systeemgebruik	5.1	Standaard loggingsfaciliteiten
9.7.3	Synchronisatie van systeemklokken		
9.8	Het gebruik van portable computers en telewerken		
9.8.1	Portable computers		
9.8.2	Telewerken		

10	*Ontwikkeling en onderhoud van systemen*	7	Dagelijks beheer
10.1	Beveiligingseisen voor systemen		
10.1.1	Analyse en specificatie van beveiligingseisen		
10.2	Beveiliging in applicaties	5.2	Applicatiespecifieke loggings
10.2.1	Validatie van invoergegevens		
10.2.2	Controle op de interne verwerking		
10.2.3	Vaststellen van de authenticiteit van een bericht		
10.2.4	Validatie van uitvoergegevens		
10.3	Cryptografische maatregelen		
10.3.1	Beleid inzake de toepassing van cryptografie		
10.3.2	Encryptie		

10.3.3	Digitale handtekeningen		
10.3.4	Onloochenbaarheid van handelingen inzake elektronische documenten		
10.3.5	Sleutelbeheer		
10.4	Beveiliging van systeembestanden		Passim
10.4.1	Beheer van operationele programmatuur		
10.4.2	Beveiliging van testgegevens		
10.4.3	Toegangsbeveiliging voor programma bibliotheken		
10.5	Beveiliging binnen de ontwikkel- en ondersteunende afdelingen		Passim,
		7.4	Scheiding tussen ontwikkeling en productie
10.5.1	Procedures voor het beheer van wijzigingen	7.5	Security patches
10.5.2	Technische controle op wijzigingen in het besturingssysteem		
10.5.3	Restricties op wijzigingen in pakketten		
10.5.4	Clandestiene kanalen en Trojaanse software		
10.5.5	Uitbesteding van software-ontwikkeling		
11	*Continuïteitsbeheer*		
11.1	Aspecten van continuïteitsbeheer		
11.1.1	Het proces 'continuïteitsbeheer'		
11.1.2	Bedrijfscontinuïteit en impact-analyse		
11.1.3	Het schrijven en implementeren van het Continuïteitsplan		
11.1.4	Structuur voor continuïteitsplannen		
11.1.5	Testen, onderhouden en opnieuw beoordelen van bedrijfscontinuïteitsplannen		
12	*Naleving van wettelijke en contractuele voorschriften, procedures, standaards en technische vereisten*		
12.1	Naleving van wettelijke en contractuele voorschriften		
12.1.1	Identificatie van de van toepassing zijnde wet- en regelgeving		
12.1.2	Voorkomen van het onrechtmatig kopiëren van programmatuur		
12.1.3	Beveiliging van bedrijfsdocumenten		
12.1.4	Naleving van de wetgeving inzake bescherming van persoonsgegevens		
12.1.5	Voorkomen van misbruik van IT-voorzieningen		
12.1.6	Regelgeving inzake cryptografische maatregelen		

In deze bijlage worden enkele van de vele security tools genoemd, die verkrijgbaar zijn via Internet. Indien er problemen zijn met het terugvinden van genoemde referenties, dan zijn er enkele plaatsen waar men kan zoeken om alsnog een versie of andere tools te vinden:

http://www.cs.purdue.edu/coast/coast.html

http://www.sites.inka.de/sites/lina/freefire-l/tools.html

Advanced Security audit trail Analysis on uniX, Abdelaziz Mounji ASAX geeft de mogelijkheid om elke vorm van een auditbestand te analyseren door het aangeven van het formaat van het logbestand. Het analyseren van grote seriële bestanden en hier relevante informatie uit halen is altijd een nachtmerrie geweest. ASAX vereenvoudigt de intelligente analyse van seriële bestanden.

http:www.info.fundp.ac.be/~cra/DOCS/asax.html

ftp://ftp.info.fundp.ac.be/pub/projects/asax

anlpasswd, Mark Henderson Een aangepaste versie van Larry Wall's Perl-password programma die het intelligente werk doet in een NIS-omgeving, toestaat het gecosveld aan te passen en ook een gesorteerde lijst van 'bad passwords' controleert.

ftp://info.mcs.anl.gov/pub/systems/

chkacct v1.1, Shabbir Safdar Chkacct werd ontwikkeld als een complement voor de tools COPS en Tiger. In plaats van configuratieproblemen in het gehele systeem te controleren, is het ontwikkeld om settings en beveiliging van het account van een gebruiker te controleren. Door het geven van beschrijvende boodschappen voor de gebruiker kunnen de problemen worden verbeterd. Het kan handig zijn voor de security administrator om bij problemen de gebruiker zelf deze tool te laten gebruiken, in plaats van dat deze administrator in de home-directory's van de gebruiker direct werkt.

http://www.cs.purdue.edu/coast/Archive Indexing.html

ftp://coast.cs.purdue.edu/pub/tools/unix/chkacct

chklastlog, DFN-CERT Chklastlog controleert de bestanden /var/adm/lastlog en /var/adm/wtmp naar tegenstrijdigheden. De 'zap'-tool verwijdert de laatste entry voor een bepaalde gebruiker van /var/adm/wtmp en de entry in het lastlog-bestand. Zijn er andere (niet-verwijderde) ingangen in het wtmp bestand, zal deze tool de ontbrekende ingang in het lastlog-bestand vinden.

http://www.cert.dfn.de/infoserv/dsb/dsb-9404.html

ftp://ftp.cert.dfn.de/pub/dfncert/fixes/chklastlog/

Chkwtmp, DFN-CERT Chkwtmp controleert het bestand /var/adm/wtmp voor ingangen die overschreven zijn met 'nullen'. Indien een dergelijke ingang wordt gevonden, dan wordt de voorafgaande en de volgende ingang aangegeven als indicatie van het tijdstip waarop de verwijdering is gemaakt.

http://www.cert.dfn.de/infoserv/dsb/dsb-9404.html

ftp:/ftp.cert.dfn.de/pub/dfncert/fixes/chkwtmp/

COPS, Dan Farmer COPS is een statisch beveiligingscontroleprogramma dat bekende procedurele (non-bug) problemen binnen een Unix-systeem controleert. In feite neemt het een vingerafdruk van het systeem en rapporteert hierover zijn bevindingen.

ftp://coast.cs.purdue.edu/pub//tools/unix/cops/

Courtney, CIAC Courtney volgt het netwerkverkeer en identificeert de bronsystemen van SATAN probes/attacks. Courtney ontvangt informatie van tcpdump waarbij het aantal nieuwe services wordt geteld dat bij een systeem binnen een bepaald tijdvenster wordt aangevraagd. Wordt een dergelijk systeem op deze manier herkend, dan wordt dit systeem gezien als een potentiële SATAN host.

http://ciac.llnl.gov/ciac/ToolsUnixNetMon.html#Courtney

ftp://ciac.llnl.gov/pub/ciac/sectools/unix/courtney/

Crack, Alec David Edward Muffett Crack is een vrij verkrijgbaar programma dat is ontwikkeld om het standaard 8-karakters DES encrypt password door standaardtechnieken te ontdekken. Het is geschreven om flexibel, configureerbaar en snel te zijn.

ftp://coast.cs.purdue.edu/pub/tools/unix/crack/

Cracklib, Alec David Edward Muffett Cracklib is een bibliotheek bestaande uit C-routines die gebruikt kunnen worden in 'passwd'-

achtige programma's. Het idee achter deze bibliotheek is om gebruikers een password te laten gebruiken dat niet eenvoudig kan worden geraden door bijvoorbeeld de tool 'Crack'.

ftp://coast.cs.purdue.edu/pub//tools/unix/cracklib/

Dig, Steve Hotz, Paul Mockapetris Dig (domain information groper) is een flexibel commando dat kan worden gebruikt om informatie te vergaren over Domain Name System servers. Dig heeft twee modes: direct interactive mode, waarbij een enkele vraag gesteld kan worden en batch mode, waarin een lijst vragen kan worden verwerkt.

ftp://venera.isi.edu/pub/dig.20.tar.Z.

DNSwalk, David Barr DNSwalk is een DNS debugger. Het voert zone transfers uit van aangegeven domeinen en controleert de database op verschillende manieren op consistentie en accuraatheid. DNSwalk heeft perl en dig nodig om te kunnen werken.

http://www.cis.ohio-state.edu/~barr/dnswalk/

fingerd - Mike Shanzer Dit programma levert logging en access control lists voor finger. Hiermee is het mogelijk om finger-verzoeken aan bepaalde systemen (en bepaalde gebruikers als identd wordt vertrouwd) toe te staan en tevens te voorzien van een message of the day bestand.

http://www.foobar.com/ onder resources
http://ftp.foobar.com/pub/fingerd.tar.gz

Gabriel, Bob Baldwin, Ben Dubin, Richard Mahn Gabriel geeft de systeembeheerder een waarschuwing van een mogelijke inbraak via het netwerk door het detecteren en identificeren van de SATAN's probe.

http://www.lat.com/gabe.htm
ftp://ftp.lat.com/gaberial-1.0.tar.Z

ifstatus, David A. Curry Dit programma werkt op een UNIX-systeem en controleert of de netwerkinterface in debug of promiscuous mode staan. Dit kan erop wijzen dat een inbreker bezig is om paswoorden af te luisteren (zie CERTadvisory CA-94:01).

http://www.cs.purdue.edu/coast/archive/Archive Indexing.html
ftp://coast.cs.purdue.edu/pub/tools/unix/ifstatus/

Internet Security Scanner, Christopher William Klaus Internet Security Scanner (ISS) is een van de eerste multi-levelbeveiligingsscanners die beschikbaar kwam. Het werd ontwikkeld om flexibel en eenvoudig over te dragen te zijn naar verschillende Unix-platformen en zijn taak in een acceptabele tijd uit te voeren. Deze tool is voor evaluatiedoeleinden vrij verkrijgbaar.

http://www.iss.net

http://www.iss.net/eval/Eval.html

Klaxon, Doug Hughes Klaxon is een aanpassing aan de sourcecode van rexec, waardoor deze zeer handig is voor het detecteren van aanvallen van portscanners, zoals ISS en SATAN. Optioneel kan het gebruikmaken van IDENT (RFC931) om te achterhalen wie de remote gebruiker is.

http://www.eng.auburn.edu/users/doug/second.html#Security

ftp://ftp.eng.auburn.edu/pub/doug/klaxon.tar.gz

logdaemon, Wietse Venema Logdaemon is een verzameling tools die het resultaat zijn van enkele jaren aanpassen van de BSD-source: (1) rsh en rlogin daemons die de naam loggen van de remotegebruiker en access-controlfaciliteiten heeft in tcp/ip daemon wrapper stijl. (2) ftpd, rexecd en loginsoftware met het loggen van verkeerde inlogpogingen en met ondersteuning van optionele S/Key one-time passwords.

ftp://ftp.porcupine.org/pub/security/index.html

ftp://ftp.porcupine.org/pub/security/

md5, Jim Ellis MD5 - New Message Digest Algorithm is een hash-functie die wordt gebruikt voor de authenticiteit van een bestand, zie RFC 1544.

http://www.rsa.com

ftp://info.cert.org/pub/tools/md5/

md5check, The Regents of the University of California Dit programma controleert of bestaande binaire bestanden overeenkomen met hun cryptografische handtekening.

http://www.agh.edu.pl/pub/security/cert/CA-94:05.MD5.checksums

New COPS Analysis and Report Program (ncarp), Diego Zamboni

Ncarp is een data-analyseprogramma dat overzichten van meerdere COPS-bestanden met onderzoeksresultaten bekijkt en analyseert. Het is gebaseerd op het carpprogramma, dat een onderdeel vormt van het COPS-pakket. Het geeft in essentie dezelfde informatie, maar nu ook rapporten per systeem en informatie hoe om te gaan om de problemen te verbeteren.

http://www.cs.purdue.edu/coast/archive/Archive Indexing.html

ftp://coast.cs.purdue.edu/pub//tools/unix/carp-ncarp/

nmap Nmap is een utility voor netwerkonderzoek en security audit. Het ondersteunt ping scanning (welke hosts zijn aanwezig), diverse port-scanningtechnieken (welke services zijn op welke hosts aanwezig), en tcp/ip vingerafdruk (remote host operating system identificatie). Nmap biedt ook een flexibel doel en port specificatie, decoy scanning, bepalen van de te verwachte TCP sequence karakteristiek, reverse-identd scanning etc.

http://www.insecure.org/

noshell, Michele D. Crabb Dit programma is ontwikkeld om systeembeheerders van extra informatie te voorzien over diegene die proberen binnen te komen op disabled accounts. In tegenstelling tot de meest gangbare manier om '/bin/sync' te gebruiken in het shell-veld in de password entry om een account te disablen, geeft dit programma een informatief alternatief.

ftp://coast.cs.purdue.edu/pub//tools/unix/noshell/src/

npasswd, Clyde Hoover Npasswd is een vervanging van het commando passwd(1). Deze versie controleert of het nieuwe password niet te eenvoudig te raden is.

http://www.utexas.edu/cc/unix/software/npasswd/

passwd+ Dit programma is een vervanging van het systeem commando passwd. Deze versie zal slechte paswoorden niet accepteren.

ftp://ftp.dartmouth.edu/pub/security/

PGP – Pretty Good Privacy PGP beschermt documenten zoals e-mail tegen niet geautoriseerd lezen door gebruik te maken van public key encryptie.

http://www.pgpi.com/

Portmap, Wietse Venema Dit programma vervangt de portmapper waarbij access control wordt toegevoegd in de stijl van de tcp wrapper (log_tcp). Het is een eenvoudig mechanisme om toegang tot NIS (YP), NFS en andere services die geregistreerd zijn in de portmapper, te bemoeilijken. Mogelijk zijn inmiddels de portmappers van de leveranciers beter dan deze versie.

ftp://ftp.porcupine.org/pub/security/index.html
ftp://ftp.porcupine.org/pub/security/

Postfix (of onder de oude naam Vmailer), Wietse Venema Postfix is een alternatief voor Sendmail. Het programma probeert aan de volgende aspecten te voldoen: snel, eenvoudig te beheren en veilig, terwijl postfix tevens voldoende overeenkomstig sendmail is ontwikkeld om gebruikers niet af te schrikken.

http://www.porcupine.org/postfix-mirror/start.html
http://www.postfix.org/

Procmail Procmail is een utility om mail te verwerken. Mail kan gesorteerd worden naar zender, onderwerp, lengte, keywoorden, etc; een ftp-by-mail server kan worden geïmplementeerd en meer mogelijkheden. Procmail is tevens een complete drop-in vervanging voor MDA (Message Delivery Agent).

http://www.iki.fi/era/procmail/mini-faq.html
ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail/

ProFTPD ProFTPD is een FTP daemon, onder GNU Public License, bedoeld om een veilige en configureerbare server te geven onder Unix en Unix-achtige operatingsystemen. Meerdere virtuele en anonymous servers zijn mogelijk en beheer is vergelijkbaar met de Apache-webserver. Deze server biedt vele mogelijkheden, maar is niet primair ontwikkeld voor snelheid.

http://www.proftpd.org/

Raudit, Michele D. Crabb Raudit is een Perl script dat elk .rhosts-bestand van de gebruikers onderzoekt en over zijn bevindingen informatie verschaft. Zonder argumenten zal Raudit info geven over het aantal entries, aantal niet-operationele entries (samenhang met het /etc/hosts.equiv bestand) en het aantal remote entries.

ftp://coast.cs.purdue.edu/pub/tools/unix/raudit.shar

RIACS Auditing Package, Matt Bishop Het RIACS Auditing Package is een bestandsscanningsysteem. Het controleert een file system op mogelijke beveiligings- of accountingproblemen, loopt het file system door en vergelijkt de resultaten met informatie in een master bestand.

http://www.cs.ucdavis.edu/~bishop/programs/tools.html

ftp://nob.cs.ucdavis.edu/pub/sec-tools/binaudit.tar.gz

Rpcbind, Wietse Venema Dit programma vervangt rpcbind, waarbij access control is toegevoegd in de stijl van de tcp/ip daemon wrapper (log_tcp) pakket. Het is een eenvoudig mechanisme om toegang tot NIS (YP), NFS en andere rpc services te bemoeilijken. Toegangscontrole voor systemen gebeurt op IP-adressen en het programma weigert verzoeken door te sturen aan rpc daemons die de oorsprong van het verzoek onderzoeken.

ftp://ftp.porcupine.org/pub/security/index.html

ftp://ftp.porcupine.org/pub/security

Satan, Dan Farmer, Wietse Venema SATAN is de Security Analysis Tool voor Auditing Networks. Het programma verzamelt netwerk-informatie, zoals het type systeem en de beschikbare services op dit systeem. Uitgaande van deze informatie wordt onderzoek gedaan naar mogelijke beveiligingsproblemen. In de zogenaamde 'exploratory mode' gaat het pakket verder op zoek naar mogelijke beveiligingsproblemen op het eigen systeem of zover als het complete netwerk.

ftp://ftp.porcupine.org/pub/security/index.html

ftp://ftp.porcupine.org/pub/security/

skey, Neil M. Haller, Philip R. Karn Het S/KEY one-time password-systeem zorgt voor authenticatie over netwerken die mogelijk te lijden hebben onder eavesdropping/reply aanvallen.

ftp://ftp.bellcore.com/pub/nmh/

smrsh, Eric Allman Smrsh is een zogenaamde restricted shell. Hierbij wordt de mogelijkheid geboden om door middel van een configuratie een expliciete lijst van commando's toe te staan. Als deze tegelijk met sendmail in gebruik is, zal smrsh het mogelijk uitvoeren van commando's door sendmail flink kunnen beperken. Deze tool is onderdeel van sendmail vanaf versie 8.7.1.

SSH, Tatu Ylönen SSH biedt de mogelijkheden om informatie versleuteld, zowel voor het password als data, over netwerken te verzenden via een telnet/rloging/rcp/remsh programma (ssh, slogin, scp). Hiermee is het mogelijk om remote beheer over een netwerk uit te voeren, waarbij ook X11-verbindingen veilig kunnen worden gebruikt. Tevens bestaat de mogelijkheid tot compressie. Dit product is ook commercieel leverbaar.

http://www.ssh.fi/sshprotocols2/

http://www.cs.hut.fi/ssh

STROBE v1.03 Super Optimised TCP port surveyor, Julian Assange

Strobe is een security/networkprogramma dat alle actief luisterende TCP-poorten op remote-systemen onderzoekt. Het maakt gebruik van een algoritme om efficiënt gebruik te maken van de beschikbare bandbreedte.

http://www.insecure.org/nmap/scanners/strobe-1.03.tgz

swatch Swatch: the Simple WATCHdog, een utility voor logbestanden. Swatch is van oorsprong ontwikkeld om boodschappen actief te monitoren terwijl deze worden geschreven naar het logbestand via de syslog utility.

ftp://ftp.stanford.edu/general/security-tools/swatch/

TCPdump TCPdump is een utility voor netwerkmonitoring en is in staat om alle verkeer in een lokaal network te volgen. Hiermee kan ruwe trace data worden afgevangen en verwerkt worden door een performance monitor. TCPdump monitort het netwerk op pakketniveau.

http://www-nrg.ee.lbl.gov/ftp.html

ftp://ftp.ee.lbl.gov/

tcp_wrappers (ook bekend onder TCPD of log_tcp), Wietse Venema

Met dit pakket is het voor een systeembeheerder mogelijk om toegang te controleren voor verschillende netwerkservices zoals SYSTAT, FINGER, FTP, TELNET, RLOGIN, RSH, EXEC, TFTP, TALK en andere netwerkservices. Verder voorziet het in logging van de netwerkservices om aanvallen via het netwerk te kunnen detecteren.

ftp://ftp.procupine.org/pub/security/index.html

ftp://ftp.procupine.org/pub/security/

tiger, Doug Schales Tiger bestaat uit een set script om een Unix-systeem te doorzoeken op mogelijke beveiligingsproblemen, op de manier van Dan Farmers COPS maar dan uitgebreider.
ftp://coast.cs.purdue.edu/pub/tools/unix/tiger/TAMU

tklogger, Doug Hughes Een hulpmiddel om logbestanden in de gaten te houden. Het is geschreven in tcl/tk en kan daardoor eenvoudig worden uitgebreid. Onder meer de logbestanden van de tcp wrappers worden in de gaten gehouden. Tklogger kan in verschillende kleuren de inhoud hiervan in real-time tonen.
http://www.eng.auburn.edu/users/doug/second.html
ftp://ftp.eng.auburn.edu/pub/doug/tklogger

Traceroute - Tracing IP packet routes, Van Jacobson Traceroute is een hulpmiddel voor systeembeheerders om de route te traceren die IP-pakketten volgen vanaf het huidige systeem naar het systeem van bestemming.
http://www-nrg.ee.lbl.gov/
ftp://ftp.ee.lbl.gov/traceroute.tar.Z

tripwire Tripwire is een toolset, inclusief een checksum database, om veranderingen in systeembinaries te detecteren.
http://www.tripwiresecurity.com/
ftp://coast.cs.purdue.edu/pub/tools/unix/Tripwire/

ttywatcher 1.0, Mike Neuman TTY-Watcher is een hulpmiddel om gebruikers op een enkel systeem te monitoren en te controleren. De gebruiker van dit programma is zowel in staat om elke tty te monitoren, als hierop te kunnen meewerken. Ook kunnen individuele connecties gelogd worden in een 'raw' bestand om later te kunnen worden afgespeeld of in een tekstbestand worden opgeslagen.
ftp://coast.cs.purdue.edu/pub/tools/unix/ttywatcher/

X Connection Monitor, der Mouse Dit programma houdt X-verbindingen in de gaten. Het maakt gebruik van RFC931 om de namen van gebruikers te tonen, mits de cliënt RFC931 ondersteunt. Het kan verbindingen onafhankelijk van de client en server bevriezen, vrijgeven of afsluiten. Het is in staat om verbindingen te volgen en

bij het gebruik van bepaalde dubieuze commando's kan het interface worden geactiveerd naar de beheerder.

ftp://coast.cs.purdue.edu/pub/tools/unix/xc/

wu-ftpd, Bryan D. O'Connor Washington University archive-ftpd, beter bekend als wu-ftpd, is een vervanging voor de ftp-daemon. Belangrijk is vooral zijn snelheid. O'Connor is niet meer werkzaam bij Washington University en ondersteunt ook wu-ftpd niet meer.

http://www.academ.com/academ/wu-ftpd

http://www.hvu.nl/~koos/wu-ftpd-faq.html

ftp://ftp.cetis.hvu.nl/pub/koos/wu-ftpd-faq.txt

xinetd v2.1.8, Chuck Murcko Xinetd is een inetd/tcp_wrapper die vele mogelijkheden biedt, inclusief UDP-services, voor logging, verificatie en beheer.

http://www.freebsd.org/ports/security.html

Zap, RokK Industries Dit programma vult de wtmp en utmp entries die overeenkomen met de ingevoerde gebruikersnaam. Het zal tevens de 'last login data' van deze gebruiker door 'nullen' vervangen. Finger naar deze gebruiker toont dus 'Never Logged In'.

ftp://coast.cs.purdue.edu/pub/tools/unix/