

# Security Information Management

*Auteur: Matthijs Koot* > Matthijs heeft in juni 2005 zijn hbo-opleiding Informatica afgerond aan de Informatica Communicatie Academie te Arnhem. Zijn afstudeeropdracht omvatte een onderzoek naar enkele beveiligingsmaatregelen (waaronder Security Information Management) en werd uitgevoerd in opdracht van de cluster Security Management van Univé Verzekeringen.

Een moderne ICT-infrastructuur omvat uiteenlopende componenten die meldingen (kunnen) genereren, waarbij een deel van die meldingen vanuit beveiligingsoogpunt interessant kan zijn - webservers, file servers, anti-virus en anti-malware systemen, routers, firewalls, VPN appliances, IDS/IPS sensors, et cetera. Door meldingen van verschillende componenten te centraliseren en analyseren kan een meer holistisch beeld worden verkregen van de beveiligingsstatus van een infrastructuur. Het implementeren van deze maatregel werkt ondersteunend bij het aantonen van compliancy met Section 404 van Sarbanes-Oxley, omdat het een onderdeel is van de interne controlemechanismen om de integriteit van de informatievoorziening te bewaken.

## Consolidatie en normalisatie

Consolidatie betreft in deze context het centraliseren van meldingen van heterogene systemen. Een melding is een set van eigenschappen waarmee de afzender een bepaald feit over zichzelf kenbaar wil maken, omdat het beleid van een organisatie - vertaald naar configuratie-instellingen van ICT-componenten - dat voorschrijft. Om meldingen van verschillende bronnen in verschillende formaten te kunnen centraliseren zal elke melding eerst moeten worden *genormaliseerd*.

Onder *normalisatie* wordt de vertaling verstaan van een inkomende melding naar een (standaard)formaat dat het SIM systeem begrijpt.

Noot: Voor de goede orde, onder normalisatie wordt niet zoals bij databases verstaan het verwijderen van redundante informatie.

Met de opkomst van nieuwe technologie als web services, Bluetooth en draadloze netwerken neemt het aantal mogelijkheden voor buitenstaanders om in te breken op ICT-omgevingen steeds verder toe. Hoewel preventieve maatregelen vanwege hun beschermende aard de eerste keuze zijn in de beveiligingsarchitectuur voor het afdekken van dergelijke bedreigingen, is het verstandig om in een gehele beveiligingsarchitectuur ook aandacht te besteden aan detectieve en correctieve maatregelen. Zodoende kan bij het ontbreken of omzeilen van een preventieve maatregel tijdig worden onderkend dat er iets ongewenst gaande is. Eén voorbeeld van een detectieve maatregel is het toezicht houden op de infrastructuur door beveiligingsgerelateerde meldingen van verschillende componenten te consolideren, correleren en interpreteren. Die maatregel staat in Amerika al langer bekend onder de term 'Security Information Management' (SIM) en is het onderwerp van dit artikel.

Daarbij zijn twee aandachtspunten:

1. syntaxis
2. semantiek

De syntaxis betreft vooral de volgorde van attributen en de gegevenstypen die worden gebruikt om feiten te beschrijven in meldingen; daarbij worden attributen 'gemapt' naar het standaardformaat van het SIM systeem.

voor elk formaat waarin binnen een infrastructuur meldingen kunnen worden gegenereerd: syslog, Windows Event Log, Checkpoint LEA, RDEP/POP et cetera. De focus van het SIM-systeem wordt daardoor beperkt tot die componenten waarvoor expliciet ondersteuning wordt aangeboden, waarmee het beoogde doel - een *totaalbeeld* van de infrastructuur - eigenlijk voorbij wordt gestreefd. Er

### Voorbeeld van syntaxis normalisatie

Gegeven een melding A, bestaande uit een veld 1, 2 en 3:

veld 1 mapt naar het algemene attribuut 'bron IP' met gegevenstype 'IP-adres';  
veld 2 mapt naar het algemene attribuut 'bron poort' met gegevenstype 'poortnummer';  
veld 3 mapt naar het algemene attribuut 'gebeurtenis' met gegevenstype 'string'.

De semantiek betreft de betekenis van attributen en is bij normalisatie een complexer probleem dan de syntaxis. Afhankelijk van bijvoorbeeld de topologische context van de afzender en de betekenis die een fabrikant aan een dergelijk attribuut geeft, kunnen zelfs simpele attributen in betekenis verschillen. Een melding uit een productieomgeving met een hoog beveiligingsniveau heeft een andere betekenis dan een melding uit een test- acceptatie- of ontwikkelomgeving. Die context moet bekend zijn bij een SIM-systeem om relevante meldingen van irrelevante meldingen te kunnen onderscheiden.

Verder impliceert de normalisatiestap het gebruik van een aparte wrapper

zijn enkele ontwikkelingen op dat gebied, waaronder de open standaarden IDXP en IDMEF [1,2] (specifiek voor IDS/IPS systemen), de proprietair-standaarden SESA van Symantec [3] en SDEE van ICSA Labs [4] (voor beveiligingssysteem in het algemeen; ondersteund door o.a. Cisco en ISS) en bepaalde ontwikkelingen binnen het *Common Information Model* van de Distributed Management Task Force (voor systeembeheer in het algemeen). Voorlopig zal het echter noodzakelijk blijven om verschillende wrappers te gebruiken.

## Schaalbaarheid

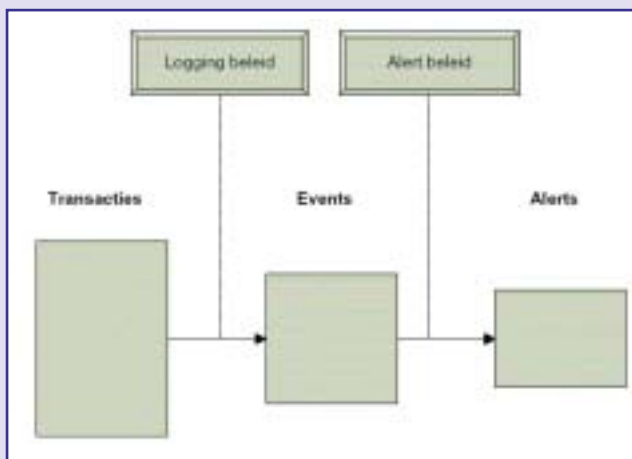
De omvang en hoeveelheid van de meldingen verschilt per component en

is vooral afhankelijk van configuratie-instellingen en gebruikslast. De ICT-componenten die vanuit beveiligings-oogpunt interessant zijn - zoals productieservers en firewalls - hebben vaak een hoge gebruikslast en genereren veel meldingen. Het centraliseren van ál die meldingen zou veel overhead veroorzaken op het netwerk. In de architectuur van een SIM-systeem zullen dan ook maatregelen zijn geïmplementeerd om het bandbreedtegebruik te reduceren (gefedereerde aggregatie, correlatie of interpretatie, compressie, et cetera). Het aantal meldingen per seconde is zelfs een metriek die wordt gebruikt bij productselectie; het SIM systeem moet immers in staat zijn om álle meldingen die op een infrastructuur worden gegenereerd af te handelen. Het is daarom niet ongebruikelijk dat een hiërarchische architectuur wordt gebruikt, waarbij aggregatie en filtering zo dicht mogelijk tegen de componenten worden uitgevoerd en tussenliggende manager nodes slechts een beperkt deel van de meldingen voor verdere analyse doorsturen naar hogere manager nodes (binnen deze context is elk systeem dat een rol speelt bij de IDS/IPS of SIM-architectuur een *node*; de centrale console is de *root node*, waaronder *manager nodes* kunnen worden ondergebracht omwille van schaalbaarheid; de eindsystemen zijn 'normale' nodes).

### Meldingen: events versus alerts

Tot nu toe werd alleen gesproken van *meldingen*. Het onderwerp dat in dit artikel wordt behandeld heeft echter verdere differentiatie naar transacties, events en alerts. Een *transactie*, zoals hier bedoeld, is de kleinste logische verwerkingseenheid waarvan een systeem of component het voorkomen kan registreren of laten registreren, typisch in een logboek. Een geregistreerde transactie heet een *event*; een event kan dus neutraal zijn en hoeft op zichzelf niet te wijzen op een aanval of inbraak. In het *logging beleid* staat gespecificeerd welke transacties wel en eventueel welke transacties niet moeten worden geregistreerd. Het logging beleid wordt primair opgesteld vanuit de tactische bedrijfsprocessen, waarbij rekening wordt gehouden met externe factoren - normenkaders van toezichthouders, wet- en

regelgeving, security baselines et cetera. Bepaalde events of combinaties van events zijn significant en geven aanleiding tot handelen; daarvan wordt tijdens de *loganalyse* een *alert* gegenereerd. Welke (combinaties van) events een alert veroorzaken staat gespecificeerd in het *alerting beleid*. Het alerting beleid wordt primair opgesteld vanuit de operationele processen, waarbij de keuzes vooral worden bepaald door relevantie en impact op die operationele processen. Zie het onderstaande schema.



Analyses van events en alerts kunnen allerlei bedrijfsprocessen ondersteunen - voor capaciteitsbeheer kan de analyse gericht zijn op de gebruikte systeembronnen (er zou een alert kunnen worden gegenereerd wanneer de harde schijf van een server 95% vol is); voor beschikbaarheidsbeheer kan de analyse gericht zijn op o.a. de bereikbaarheid van componenten (er zou een alert kunnen worden gegenereerd als een router is uitgevallen); voor kostenbeheer kan de analyse gericht zijn op het gebruik per afdeling, et cetera. In dit artikel staat de analyse centraal als functie voor beveiligingsbeheer.

### Correlatie

Correlatie is het op verschillende manieren analyseren van de (geconsolideerde) events of alerts om verbanden te ontdekken. Correlatie heeft verschillende doelen (zie de tabel).

Dan Gorton maakte in zijn licentiaat thesis [4] onderscheid tussen correlatie van events en correlatie van alerts:

*"Intrusion event correlation refers to the interpretation, combination, and analysis of neutral events from all available sources, about target system activity for the purposes of intrusion detection and response."*

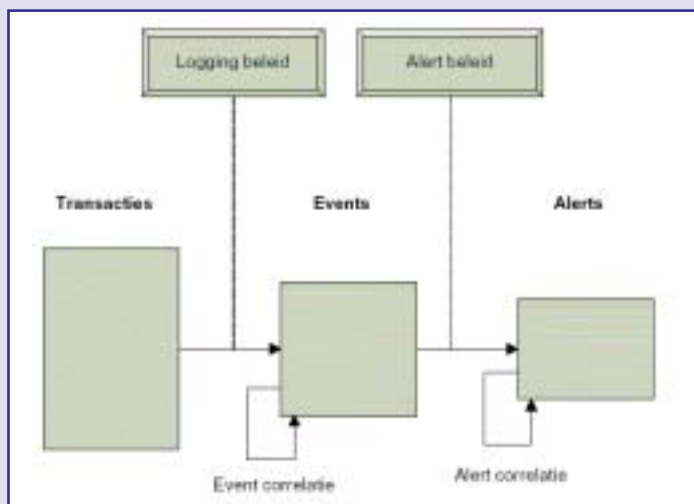
*"Intrusion alert correlation refers to the interpretation, combination, and analysis of intrusion alerts, together with information external to the intrusion detection system, with the purpose of intrusion alert refinement and intrusion scenario building."*

#### Doelen van correlatie

(Aldus Hervé Debar in [3])

1. reductie van het aantal meldingen;
  - wegfilteren van de eerste ruis (false positives);
  - samensmelten van meldingen die betrekking hebben op dezelfde gebeurtenis;
  - groeperen van gelijksoortige meldingen;
  - logische verbanden leggen tussen meldingen;
2. verbeterde diagnose;
  - type activiteit (welk soort aanval? geen aanval?);
  - relevantie (is de infrastructuur überhaupt kwetsbaar?);
  - verificatie (is de aanval geslaagd?);
3. activity tracking;
  - is er informatie uitgelekt naar de aanvaller?
  - welke informatie is beschikbaar over de aanvaller?

Samengevoegd met het eerdere schema: componenten (IDS alerts van verschillende netwerk sensors) en correlatie



Vanaf hier zal de focus liggen op correlatie van *alerts*, omdat uit die stap (in theorie) de meest waardevolle informatie kan volgen: de reconstructie van een hack scenario dat heeft plaatsgevonden. Bij het correleren van alerts zijn er twee belangrijke aandachtspunten:

1. temporal proximity (tijd);

- verhoudingen in timestamps van verschillende alerts kunnen wijzen op causaliteit; deze eigenschap kan helpen bij detectie van multi-step intrusions (bijv: eerst heeft de aanvalder een covert channel opgezet, toen een interne webserver gekraakt via een exploit, daarna een Distributed Denial of Service (DDoS) robot gestart);

2. spatial proximity (ruimte);

- correlatie van alerts met hetzelfde doelwit (bijv. doel-IP, doel-poort);
- correlatie van alerts met dezelfde vermoedelijke veroorzaker (bijv. bron-ID, user-id, proces-id);
- correlatie op basis van de afkomst van de alert (topologische context en host versus netwerk sensor).

Om de *temporal proximity* te kunnen bepalen dient de tijdsynchronisatie op de verschillende componenten vrij nauwkeurig te zijn; van elk component dat niet synchroon loopt in tijd zullen bij correlatie geen verbanden kunnen worden ontdekt in tijd.

Er wordt veel onderzoek verricht naar correlatiemechanismen, waarbij zowel wordt gekeken naar correlatie van homogene als heterogene bronnen. Er is vooral materiaal beschikbaar over correlatie van alerts afkomstig van netwerk-

tussen alerts van netwerkcomponenten en hostcomponenten (IDS alerts van host en netwerk sensors).

Huidige theorieën rondom correlatie vallen uiteen in twee categorieën:

1. alert clustering;
2. intention recognition.

Bij alert clustering worden alerts gecorrelleerd op basis van overeenkomsten in attributen, zoals bron IP, doel poort, proces-id en user-id. Het doel van alert clustering is *root cause analysis*, dat zoveel wil zeggen als het achterhalen van de gebeurtenissen die ten grondslag liggen aan een reeks van gegeneerde alerts. Door gebruik van clustering kan het aantal alerts dat de console bereikt aanzienlijk worden verminderd, zoals Klaus Julisch aantoonde in [5]; de overgebleven alerts zijn bovendien van hogere kwaliteit.

Bij intention recognition (ook bekend als *attack plan recognition*) wordt geprobeerd om op basis van alerts te herleiden (of voorspellen) wat de intentie van de tegenstander is. Als de beheerder tijdig weet wat de tegenstander van plan is (multi-step attack) kan hij tijdig adequate maatregelen nemen om escalatie te voorkomen. Enkele voorbeelden zijn correlatie op basis van voorgedefinieerde aanvalsscenario's, correlatie op basis van pre- en postcondities en state transition analysis (bij *state transition analysis* worden aanvallen gemodelleerd als een verzameling van transities die samen een pad vormen van een 'ongekraakt' tot een 'gekraakt' systeem - zie [6, 7]).

In de komende paragrafen zullen enkele vormen van correlatie worden toegeleucht. De verschillende vormen van correlatie zijn vaak complementair. De mechanismen die in dit artikel worden besproken zijn alle vormen van *knowledge-based correlation*; bij dergelijke mechanismen is kennis vereist van kwetsbaarheden en aanvalsscenario's. Bij *behaviour-based correlation* is zulke kennis niet vereist, maar worden statistische methoden als Bayes en Granger-Causality gebruikt om de causaliteit van meldingen te bepalen. Er zijn echter nog geen bruikbare implementaties van deze vormen van correlatie, daarom blijven ze hier verder buiten beschouwing. De geïnteresseerde lezer wordt doorverwezen naar een onderzoeksrapport dat in 2004 is opgesteld door de vier IDS-goeroes Valeur, Vigna, Kruegel en Kemmerer [8].

Alert clustering

Bij alert clustering worden alerts gegroepeerd op basis van overeenkomsten in hun attributen, waardoor *attack threads* ontstaan. Eén thread bevat alle alerts die aan één aanval gerelateerd lijken te zijn. Elke inkomende alert wordt vergeleken met alle bestaande threads maar wordt alleen toegevoegd aan de thread die het 'best' overeenkomt. Belangrijke vragen zijn welke attributen moeten worden vergeleken, hoe de overeenkomst wordt bepaald en hoe zwaar elk attribuut meeweegt bij die vergelijking. Voorbeelden van te vergelijken attributen zijn de locatie en naam van de betrokken sensor, bron en doel IP-adressen, bron en doel poorten, de bron en doel user-id's, het soort aanval en de tijd. Elk attribuut heeft een eigen metriek op basis waarvan de mate van overeenkomst wordt bepaald. Bij IP-adressen kan worden vergeleken met subnets, bij poortscans kan worden gekeken naar overlap van poorten of TCP vlaggen, bij tijd kan worden gekeken naar nabijheid, et cetera.

Voor verschillende aanvallen wordt voor verschillende attributen een verschillende mate van overeenkomst verwacht. Die verwachte waarde kan vervolgens meetellen bij de clustering. Het lijkt bijvoorbeeld redelijk om van een SYN flood te verwachten dat er géén overeenkomst is in bron-IP (IP spoofing is immers essentieel bij zo'n aanval - anders zou de aanval met een simpele

ACL zijn af te weren); van een portsweep wordt verwacht dat de overeenkomst in bron-IP en doel-poort groot is, maar dat de overeenkomst in doel-IP klein is.

De clusters worden ten slotte samengesmolten (*alert fusion*), waarna een meta-alert wordt gegenereerd die de hele cluster vertegenwoordigt.

De kwaliteit en effectiviteit van alert clustering is afhankelijk van parametrisatie van het algoritme; de *similarity matrices* en *similarity expectations* moeten door een expert worden ingesteld, waardoor de mens en zijn kennis hierbij belangrijke succesfactoren zijn.

### Voorgedefinieerde scenario's

Aan deze vorm van correlatie liggen vooraf gedefinieerde aanvalsscenario's ten grondslag. Zulke scenario's kunnen ofwel handmatig door beveiligingsexperts worden gespecificeerd, ofwel automatisch worden aangeleerd door data mining van 'training data sets' (afgetapt hacking verkeer).

Enkele voorbeeldscenario's staan in de onderstaande tabel. Het gaat bij deze voorbeelden om *sjablonen* voor aanvallen; de werkelijke waarde van de attributen verschilt per aanval.

De inkomende alerts worden geanaly-

Scenario Language of LAMBA. Evenals bij alert clustering kan alleen worden gecorreleerd op basis van voorkennis - in dit geval kunnen alleen aanvallen worden herkend waarvan een voorgedefinieerd scenario bekend is; onbekende scenario's worden niet herkend.

### Pre- en postcondities

Correlatie op basis van pre- en postcondities (ook wel *prerequisites* en *consequences* genoemd) - is gericht op voorwaardelijke relaties tussen alerts. Een preconditione specificeert een noodzakelijke voorwaarde voor een geslaagde aanval. Een postconditie specificeert het mogelijke resultaat van een geslaagde aanval (en wordt daarom ook wel *gevolg* of *consequentie* genoemd). Het correlatiealgoritme zoekt naar paren van alerts waarbij de postconditie van de ene alert overeenkomt met de preconditione van een andere alert; daardoor ontstaat een (*may-*)*prepare-for* relatie (mits de chronologische volgorde van de alerts dat toestaat).

Pre- en postcondities worden - net als aanvalsscenario's - als predikaten gespecificeerd. De combinatie van een alerttype, preconditione en postconditie wordt hyper-alert genoemd. Een hyper-alert wordt als volgt gespecificeerd: (*fact, prerequisite, consequence*)

Het veld *fact* specificeert het soort informatie dat het alerttype betreft; het veld *prerequisite* specificeert de voorwaarden waaraan moet worden voldaan, wil de aanval succesvol zijn; het veld *consequence* specificeert de gevolgen van een succesvolle aanval.

Stel, een tegenstander is voor DDoS doeleinden op zoek naar Sun Solaris systemen die een kwetsbare versie van de Sadmin service draaien (dit voorbeeld komt uit [9]). Bij exploitatie van die service verkrijgt de tegenstander root-rechten en kan een DDoS daemon worden geïnstalleerd. De hyper-alert voor exploitatie van de Sadmin service wordt als volgt gespecificeerd:

Hyper-alert Type *SadminBufferOverflow* = (*{VictimIP, VictimPort}, ExistHost(VictimIP)^VulnerableSadmin(VictimIP), GainRootAccess(VictimIP)*)

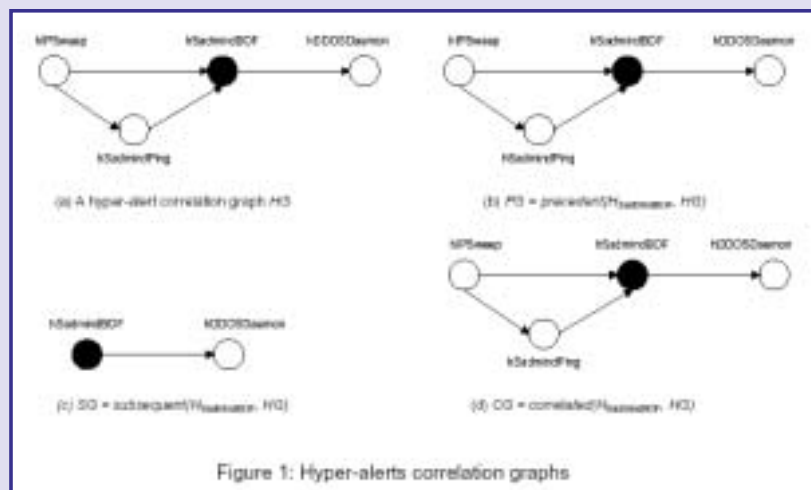
De *facts* zijn het doel-IP en de doel-poort. De *prerequisites* zijn dat het doelsysteem bestaat en een kwetsbare versie van Sadmin draait. De *consequence* is dat de tegenstander root-rechten verkrijgt op het doelsysteem.

De aanval bestaat uit verschillende stappen die allemaal alerts veroorzaken. Correlatie van die alerts resulteert in een gerichte acyclische graaf (een *graaf* is een schematische voorstelling van de werkelijkheid, bestaande uit een verzameling punten waarvan er sommige verbonden zijn), waarvan de nodes meldingen zijn die (*may-*)*prepare-for* relaties kunnen hebben met leave nodes. Na afloop van de aanval zijn vier meldingen gegenereerd, waarbij de correlatie resulteerde in graaf (d) uit het onderstaande figuur:

Scenario	Indicatie van kenmerkende attributen
Eén tegenstander valt één aanval uit op één doelwit	Zelfde bron-IP, doel-IP en soort aanval
Eén tegenstander die aanvallen uitvoert vanaf en op één doelwit	Zelfde bron-IP en doel-IP
Een gedistribueerde aanval op één doelwit.	Zelfde doel-IP en soort aanval.
Eén tegenstander die dezelfde aanval op meerdere doelwitten uitvoert	Zelfde bron-IP en soort aanval

seerd vanuit deze voorgedefinieerde scenario's, waarbij alerts die samen een bepaald scenario lijken te vormen worden gecorreleerd. Nadeel aan deze manier is dat onbekende aanvalsscenario's niet zullen worden opgemerkt. Recentelijk is in dat kader onderzoek gedaan naar *plan recognition and prediction* op basis van causale netwerken. Van die theorie zijn echter nog geen bruikbare implementaties beschikbaar.

De aanvalsscenario's worden gespecificeerd in declaratieve talen (predikaten) zoals *chronicles formalism*, *Attack*



Uit (d) blijkt dat de aanval als volgt is uitgevoerd:

1. tegenstander pingt een IP reeks (hIPSweep);
2. tegenstander controleert de systemen die antwoorden op aanwezigheid van Sadmind (hSadmindPing, optioneel);
3. tegenstander exploiteert Sadmind (hSadmindBOF, instantie van het hyper-alert type *SadmindBufferOverflow*);
4. tegenstander prepares-for installatie van een DDoS daemon (hDDOSDaemon).

Stap 4 betreft een voorspelling (!); de postconditie van hSadmindBOF vervult in dit voorbeeld de preconditionie van hDDOSDaemon (root-rechten op een systeem). Naarmate de complexiteit van de aanval toeneemt en er meer hyper-alert typen zijn, groeit ook de resulterende graaf. Het resultaat kan een graaf zijn die uit duizenden, zo niet tienduizenden nodes bestaat.

Deze benadering van correlatie is gelijktijdig onderzocht door verschillende onafhankelijke groepen, waaronder Templeton en Levitt (JIGSAW, 2000-2001), Cuppens (MIRADOR, 2002) en Ning, Cui en Reeves (TIAA, 2002-2005).

Evenals bij alert clustering en aanvalsscenario's kan alleen worden gecorrigeerd op basis van kennis - in dit geval kunnen alleen voorgedefinieerde aanvallen worden herkend; onbekende aanvallen worden niet herkend.

### Conclusie

Security Information Management is de discipline die orde moet gaan scheppen in de chaos van de vele beveiligingsgerelateerde meldingen die in een grote, heterogene infrastructuur worden gegenereerd. Door

meldingen van verschillende bronnen - host, netwerk, applicatie - te consolideren naar een centrale repository kan daarna middels verschillende vormen van correlatie kennis worden verkregen over intrusions en andere policy violations. Bij het consolideren worden zowel de syntaxis als semantiek van meldingen genormaliseerd. Er zijn geen standaarden voor die normalisatie, waardoor wetenschappelijk onderzoek naar correlatie tussen host-, netwerk- en applicatiecomponenten wordt bemoeilijkt. De huidige inzichten van correlatie bieden alleen praktische mogelijkheden voor correlatie op basis van kennis; alleen bekende aanvallen, bekende patronen, bekende scenario's worden herkend. Er wordt weliswaar onderzoek gedaan naar verschillende vormen van statistische correlatie waarmee onbekende aanvallen zouden kunnen worden herkend, maar dat onderzoek heeft nog niet geleid tot bruikbare implementaties. Om correlatie goed te kunnen laten geschieden is tijdsynchronisatie tussen de verschillende componenten essentieel. De huidige stand van correlatietechniek biedt goede mogelijkheden voor herkenning van bekende aanvalspatronen, maar nog niet voor onbekende aanvalspatronen (voor zover die niet via inductie uit bekende patronen kunnen worden herleid).

Als SIM goed wordt opgezet en voorziet in goede correlatiefuncties wordt een bestaande beveiligingsarchitectuur uitgebreid met een waardevolle detectieve maatregel. Een goede driver voor implementatie van SIM is de bijdrage die het holistische toezicht levert voor compliance met Section 404 van Sarbanes-Oxley. Anticiperend op toekomstige wet- en regelgeving wordt ook niet-beursgenoteerde ondernemingen aanbevolen om SIM alvast in overweging te nemen.

### Bronvermeldingen

- [1] IETF draft standard - Intrusion Detection Message Exchange Format (IDMEF):  
<http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-14.txt>
- [2] IETF draft standard - Intrusion Detection Exchange Protocol (IDXP):  
<http://www.ietf.org/internet-drafts/draft-ietf-idwg-beep-idxp-07.txt>
- [3] Hervé Debar, 2004, "Intrusion detection alerts"  
<http://seclab.cs.ucdavis.edu/seminars/Herve-slides.pdf>
- [4] Dan Gorton, 2003, "Extending Intrusion Detection with Alert Correlation and Intrusion Tolerance":  
[http://www.ce.chalmers.se/~daane/LicentiateThesis\\_031209.pdf](http://www.ce.chalmers.se/~daane/LicentiateThesis_031209.pdf)
- [5] Klaus Julisch, 2003, "Clustering Intrusion Detection Alarms to Support Root Cause Analysis":
- [6] Udo Payer, 2004, "Realtime Intrusion-Forensics: A First Prototype Implementation":  
<http://www.terena.nl/library/tnc2004-proceedings/papers/payer.pdf>
- [7] Richard Kemmerer, 1995, "State Transition Analysis: A Rule-Based Intrusion Detection":  
<http://www.csl.sri.com/papers/stat-paper/stat-paper.ps.gz>
- [8] Fredrik Valeur et al, 2004, "A Comprehensive Approach to Intrusion Detection Alert Correlation"  
<http://www.cs.ucsb.edu/~rsg/Hidra/Papers/correlation.pdf>
- [9] Peng Ning et al, 2002, "Constructing Attack Scenario's through Correlation of Intrusion Alerts"  
<http://discovery.csc.ncsu.edu/~pning/pubs/ccs02.pdf>