DATABASE ACTIVITY MONITORING (DAM)

JOHAN LOECKX

Abstract – Oplossingen voor Database Activity Monitoring registreren alle database queries om op deze manier interne en externe aanvallen op te sporen en het lekken van informatie of misbruiken van privileges te kunnen vaststellen. Doordat ze georiënteerd zijn rond data en redelijk transparant ontplooid kunnen worden, zijn ze een krachtig middel om de veiligheid van uw systemen aanzienlijk te verhogen. Bovendien kunnen ze ook dienen om uw audit trails te verfijnen en de integriteit ervan te verbeteren.

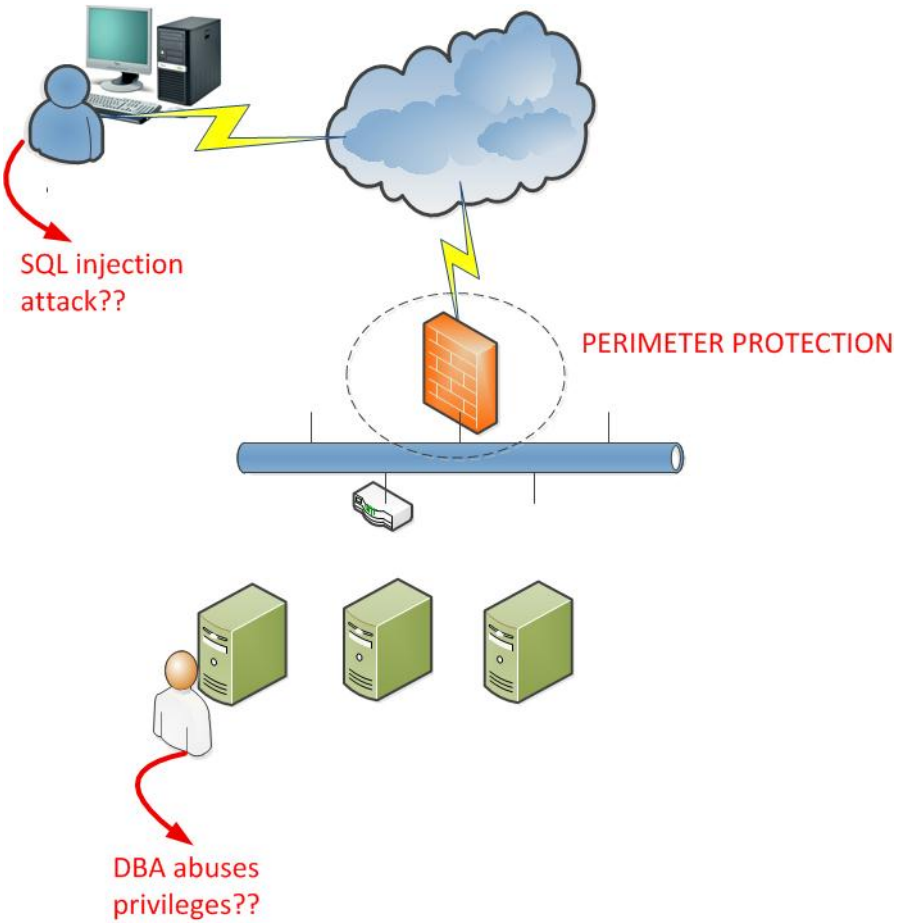
Résumé - Les solutions de Database Activity Monitoring enregistrent toutes les interrogations de bases de données pour détecter les attaques internes et externes ainsi que pour pouvoir constater la fuite d'informations ou l'abus de privilèges. Étant donné qu'elles sont orientées vers les données et qu'elles peuvent être déployées de façon relativement transparente, elles constituent un puissant outil pour augmenter considérablement la sécurité de vos systèmes. Elles peuvent en outre servir à affiner vos pistes d'audit et à en améliorer l'intégrité.

Inhoudsopgave

1. Introductie

Het beschermen van gevoelige gegevens is een uitdagende taak: elke zwakheid in het besturingssysteem, application server, database of zelfgeschreven code kan aangewend worden om schade te berokkenen aan uw systemen of om ongeoorloofde toegang te krijgen tot confidentiële gegevens.

Ondanks het feit dat er vele technieken bestaan om uw systemen te beschermen, blijft de hamvraag: **hoe kunt u zeker zijn dat er geen gegevens lekken?** [1]



Figuur 1: Traditioneel wordt de perimeter goed beschermd. Gevoelige gegevens kunnen echter ook gecompromitteerd worden via bv. SQL injection attacks of misbruik van privileges.

Vaak wordt immers het netwerk goed beschermd aan de perimeter (zie Figuur 1), maar de dreiging zou evengoed van elders kunnen komen: misschien gebruikt een gemachtigd persoon zijn privileges om de gegevens aan te wenden voor ongeoorloofde doeleinden? Misschien wordt gebruik gemaakt van een SQL injection aanval op de applicatie, die een gerechtigd gebruiker is en daarom onopgemerkt blijft? Misschien worden de logfiles gemanipuleerd door de administrator waardoor audits zinloos blijken? Misschien werd een stored procedure of trigger ongemerkt gewijzigd?

DAM of Database Activity Monitoring probeert op deze problematiek een antwoord te geven.

Deze research note is als volgt opgebouwd. Allereerst zullen de functionaliteiten en architectuur van DAM besproken worden. Vervolgens zal gekeken worden hoe zo'n oplossing in de praktijk gebracht kan worden en welke activiteiten het best gemonitord worden. Er wordt een kort overzicht gegeven van de markt om te eindigen met conclusies.

1.1. Context

In deze brede en steeds complexer wordende context is het dan ook niet onzinnig om te **vertrekken van een data-centric approach**. Deze bestaat uit drie componenten die essentieel zijn om tot een effectief veiligheidsbeleid te komen:

- het **beschermen van de data zelf,** bv. door encryptie en Endpoint Data Loss Prevention;
- het **beheer van de toegangen** tot de data (de provisionering van de gebruikers en het bepalen én onderhouden van hun rechten);
- het beschermen van het **transport** van gegevens over het netwerk met bijhorende monitoring & alerting.

Database Activity Monitoring (DAM) houdt nauwgezet bij of de gegevens die toekomen en vertrekken in databanken voldoen aan vooropgestelde policies [2]. Hierbij wordt a priori geen onderscheid gemaakt of de gegevens opgevraagd worden door een gewone gebruiker, een applicatie of een administrator en of dit gebeurde naar aanleiding van een toegelaten of verboden handeling. DAM situeert zich dus grotendeels aan de **"reactieve" zijde** van security omdat ze aan detectie doet, wanneer het (onmiddellijke) kwaad in se eigenlijk al geschied is.

Daar de overgrote meerderheid van gevoelige gegevens zich in databanken bevindt, kunnen we zeggen dat DAM een **krachtig middel is om gegevensverlies tegen te gaan.**

1.2. Wat is "Database Activity Monitoring"?

Database Activity Monitors onderscheppen en registreren alle SQL-operaties quasi in realtime over alle gebruikers, toepassingen en platformen heen. Op deze manier geven ze een **compleet en holistisch zicht op de actuele en historische databaseactiviteit.** Bovendien kan er een policy ingesteld worden waarbij waarschuwingen geleverd worden indien er afgeweken wordt van het beleid.

DAM karakteriseert zich aan de hand van deze kerneigenschappen [3]:

- De werking is onafhankelijk van het databasemanagement-systeem, waarbij de "native" logs niet vertrouwd moeten worden.
- De opslag van deze activiteit gebeurt buiten de database.
- Het combineren van activiteit geschiedt over verschillende DBMS'en en operating systems heen.
- "Segregation of duties"[1] kan gecontroleerd en in bepaalde gevallen zelfs afgedwongen worden.
- Afwijkingen van de opgestelde policy / verwacht gedrag kunnen gedetecteerd worden met de mogelijkheid om alerts te genereren.

Sommige DAM-tools laten ook toe om transacties te blokkeren, maar deze functionaliteit wordt in de praktijk niet zo vaak gebruikt omdat dit de goede werking van de legitieme transacties kan verstoren en applicaties hier bovendien niet altijd mee overweg kunnen [4].

1.3. Situering

Database Activity Monitoring hoort samen in het rijtje met andere **technologieën voor security governance** zoals Security Information & Event Management (SIEM), Privileged Account Management (PAM), etc. [5]. Als we de maturiteit van de technologie beschouwen, zien we dat DAM juist uit de "Trough of disillusionment" klimt en dus op weg is om een gevestigde waarde te worden in het beveiligingsspectrum [6].

DAM-tools gaan verder dan traditionele database audits in de hoeveelheid detail die ze opslaan. Waar een DB audit trail vermeldt dat user *john* "data" heeft gekregen uit de database "customers", zal een DAM-tool weergeven dat *john* 100 000 kredietkaartnummers heeft opgevraagd...

Het is belangrijk te beseffen dat beveiliging van databases verder reikt dan technologie alleen en kadert in een breder geheel van security

[1] Segregation of duties vereist dat er verschillende personen noodzakelijk zijn om een bepaalde taak uit te voeren, om zo fraude en menselijke fouten te elimineren.

governance. Voor een effectieve beveiliging moeten dus ook op dit vlak maatregelen genomen worden [1,7]:

- De configuraties en het onderliggende platform moeten beveiligd worden.
- De gevoelige gegevens moeten in kaart gebracht worden (welke gegevens in welke databases - hierover later meer).
- De zwakheden van alle componenten moeten gescand worden aan de hand van zogenaamde “vulnerability scanners”.

Bovendien werd reeds aangehaald dat gegevens ook preventief beschermd kunnen worden door encryptie, data masking[2] en het instellen van de nodige toegangscontroles.

2. Werking

2.1. Architectuur

Een DAM-tool bestaat uit twee types componenten. Enerzijds verzamelt de tool SQL queries in zogenaamde *collectoren,* anderzijds worden deze in een *centraal punt* veilig opgeslagen (Figuur 2). Dit centrale knooppunt doet dus niet aan monitoring maar verwerkt alle gegevens.

De collectoren zijn op zijn minst in staat om SQL-trafiek op te vangen (het essentiële verschil met SIEM dat niet zo diep binnenin de netwerk-pakketten dringt) en komen aan de nodige gegevens op één van de volgende manieren, waarbij soms verschillende methodes worden gecombineerd [2]:

- ***Network monitoring*** – in deze configuratie luisteren de collectoren actief het netwerk af.

  Het voordeel is dat er geen overhead gecreëerd wordt op de database server en dat er geen wijzigingen moeten aangebracht worden aan de database. Ook is deze manier intrinsiek platform-onafhankelijk.

  Het nadeel is dat er geen lokale activiteit gemeten kan worden en geen interne databasegegevens (indicatoren, statistieken, administratieve settings) opgeslagen kunnen worden.

- ***Remote monitoring*** – de collector krijgt “root”-toegang tot de database, waarop de native auditing geactiveerd wordt.

[2] Data masking is de praktijk om gevoelige gegevens zoals INSZ-nummers of VISA-kaartnummers te vervangen door een geanonimiseerde versie in een gelijkaardig formaat.

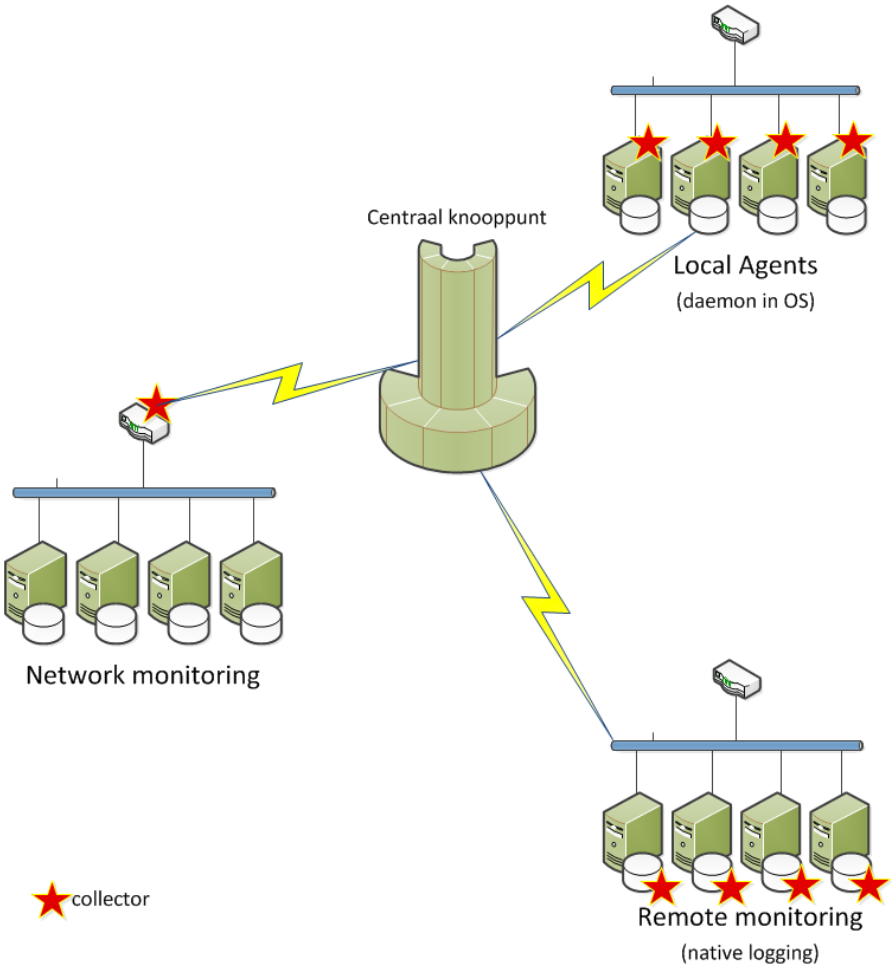
Het grote voordeel is dat alle mogelijke informatie opgeslagen kan worden, inclusief lokale activiteit.

Nadelen zijn de (eventuele) performance-impact van native logging/auditing, de noodzaak om administratieve access toe te laten en het feit dat de configuratie van de database mogelijkerwijs aangepast moet worden.

- ***Local agents* –** er wordt een lokale "daemon" geïnstalleerd die de nodige informatie verzamelt. Deze zijn transparant voor de database en hebben slechts een beperkte performance-impact (3 à 5 %).

  Het nadeel is dat deze werkwijze erg platform-afhankelijk is en een extra daemon geïnstalleerd moet worden.

Het **grote voordeel van een DAM-tool is het feit dat deze transparant voor de bestaande businessprocessen** en -applicaties ontplooid kan worden.



Figuur 2: Er bestaan drie types architecturen voor de uitrol van een DAM-oplossing, waarbij verschillende architecturen gecombineerd kunnen worden. Ofwel wordt het netwerk gemonitord (network monitoring), een agent/daemon geïnstalleerd op niveau van het OS of wordt gebruik gemaakt van de native logging van het databaseplatform.

2.2. Functionaliteit

2.2.1. Events normaliseren en correleren

Met het verzamelen van SQL queries alleen zijn we natuurlijk niets. Juist zoals bij SIEM zijn DAM-tools in staat om **events** komende van verschillende collectoren te **normaliseren en correleren.** Ook kunnen ze **afwijkende patronen van toegang detecteren** (bv. een privilege escalation[3] gevolgd door een SELECT van een grote hoeveelheid gevoelige data).

2.2.2. Policies & alerting

Bovendien is het mogelijk om **policies te definiëren**, die gebaseerd kunnen worden op:

- ***Queries en metadata*** – bv. specifieke queries, of limieten op het aantal teruggegeven resultaten, signatures van queries om SQL injection attacks te kunnen detecteren, toegang van bepaalde users tot bepaalde data, etc.
- ***Activiteit*** **–** aan de hand van historische gegevens wordt een profiel gemaakt van "normale" activiteit en kunnen afwijkingen gemeld worden.
- ***Inhoud*** **–** juist zoals bij EDLP-tools is het mogelijk om alerts te genereren als bepaalde data (bv. een INSZ-nummer of kredietkaartnummer) opgemerkt worden.

Vooral bij grote **complexere set-ups** kan de heuristische optie heel interessant zijn omdat het **opstellen van regels in deze gevallen quasi onmogelijk** is. Voor bepaalde compliance-standaarden (zoals PCI, SOX, ...) en applicaties (bv. SAP) zijn er voorgedefinieerde rule sets beschikbaar, maar ook deze moeten toch nog steeds gefinetuned worden.

2.2.3. Workflow & reporting

Niet alle alerts zijn even prioritair of vereisen dezelfde afhandeling. Sommige alerts zijn van retrospectief belang, terwijl andere onmiddellijke aandacht vereisen. Om deze reden is het dan ook van groot belang om een workflow te kunnen specificeren, en op interactieve wijze te kunnen grasduinen in de relevante security-informatie.

[3] Met "privilege escalation" bedoelt men de acties waarbij een gewone user tijdelijk meer rechten (privileges) krijgt, bv. door het tijdelijk veranderen naar het root account.

2.2.4. Connection Pool User Identification

Applicatieservers maken gebruik van zogenaamde “connection pools” (het kanaliseren van database queries van verschillende applicatieve users door eenzelfde database account) om contact te leggen met de databank. Hierdoor wordt de link tussen de (applicatief) ingelogde gebruiker en de databasetoegang vertroebeld. Door het correleren van applicatieve logs en database logs of door het connecteren met een Web Application Firewall kan het verband toch nog gelegd worden.

2.2.5. Forensic views

Deze feature heeft iets minder te maken met pure security. Forensic views laten toe om de vaakst uitgevoerde queries, de meest voorgekomen fouten en de response times van queries op te vragen, alsook welke gebruikers welke data hebben geraadpleegd etc. Op deze manier is het gemakkelijker inzicht te krijgen in de soorten verkeer en toegangen naar de database.

2.2.6. Content discovery en het opslaan van SQL responses

Sommige tools laten toe om ook het antwoord van SQL queries op te slaan. Zo kunnen bv. alle resultaten van queries uitgevoerd door een bepaalde user bekeken worden, of de resultaten wanneer een bepaalde kolom of een bepaald schema geraadpleegd wordt. Vaak wordt hierin in combinatie data masking toegepast. Deze feature kan interessant blijken voor bepaalde accounts met uitgebreide privileges (gevoelige accounts).

Als er een inbreuk op de policy vastgesteld werd, kan het interessant zijn om te weten welke data nu juist bekomen werden door de potentiële aanvaller. Sommige DAM-tools laten daarom toe om de result sets (die voldoen aan bepaalde criteria) op te slaan, weliswaar ten koste van de nodige performantie-bottleneck. Een mogelijke toepassing is om op deze wijze te weten te komen welke kredietkaartnummers gecompromitteerd werden.

De vraag blijft echter: welke activiteiten moeten we in het oog houden? Met de steeds toenemende hoeveelheid data (de zogenaamde “Big Data”-evolutie) is het aantal events immers niet te onderschatten! Om deze reden wordt aangeraden om bescheiden te starten en zich te concentreren op de meest gevoelige data eerst, zoals bv. INSZ-nummers.

3. DAM in de praktijk

Meer algemeen stelt zich de vraag: hoe beginnen we er nu aan? In de volgende paragraaf wordt het breder kader geschetst waarin Database Activity Monitoring opereert. We bekijken de activiteiten waarop men zich best concentreert om te eindigen met een paar voorbeelden van typisch gebruik.

3.1. Proces bij het aanwenden van een DAM

De **allereerste stap** bij het ontplooien van een oplossing voor Database Activity Monitoring is de zogenaamde **"discovery" van databases**, het in kaart brengen van de databases. In tegenstelling tot wat algemeen aangenomen wordt, zijn deze gegevens niet altijd gekend. Zo zal men databases in development vinden met kopieën van gegevens uit productie, of databasekopieën die voor offline analyse gebruikt worden, etc. DAM-oplossingen bezitten vaak tools om deze discovery te doen, hoewel ze op zich niet strikt deel uitmaken van de tool.

In een **tweede stap wordt elke database-instantie geanalyseerd**, om te detecteren of ze gepatched zijn, of de configuratie veilig is en om te kijken of ze gevoelige gegevens bevatten. Ook wordt er nagegaan of er zwakke wachtwoorden gebruikt worden en of er gebruikers zijn met overdreven privileges. Deze handeling wordt uitgevoerd door zogenaamde Vulnerability Scanners. Steeds vaker zijn dit type tools inbegrepen in het DAM-pakket. Indien niet, kan er gekeken worden naar het aanbod van de typische security vendors zoals Application Security, Fortinet, Imperva, ...

Vervolgens moet er nagegaan worden **wie toegang heeft** tot elk van deze databases. Het is van groot belang dat naast de standaardgebruikers ook de andere toegangskanalen bekeken worden: zo kan een superuser van het systeem toegang krijgen via het besturingssysteem, misschien zijn er connecties via ODBC-connectoren, etc.

Het is op dit moment dat de **database activity monitoring ingeschakeld kan worden om te observeren en analyseren wat niet gekend is,** namelijk hoe worden de databases gebruikt? DAM-tools laten toe om een brede waaier aan vragen te beantwoorden:

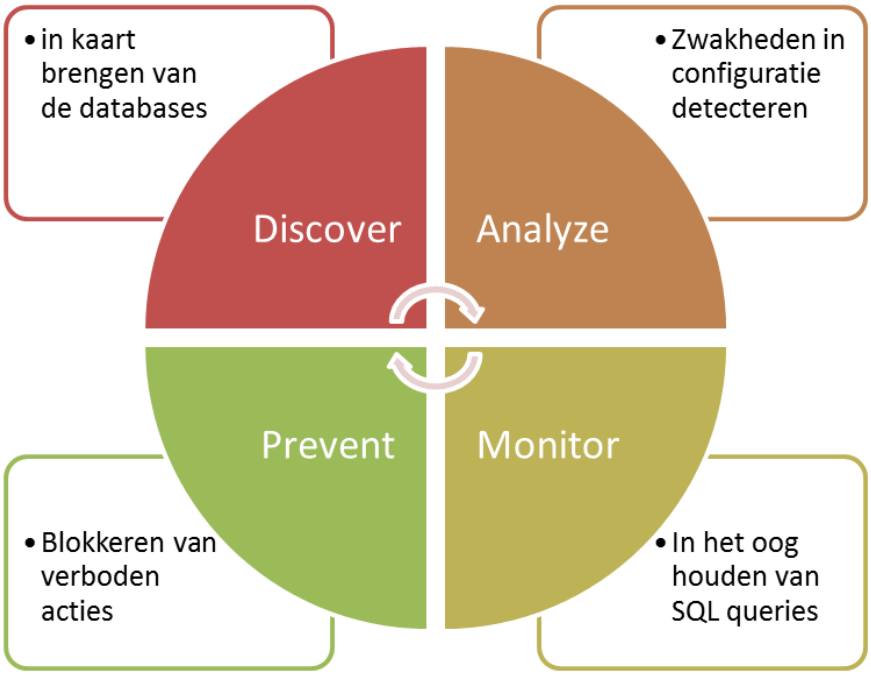
- Misbruiken administratieve users hun privileges om onterecht toegang te krijgen tot gevoelige data?
- Is er een mogelijke SQL injection attack gaande omdat een applicatief account verdacht veel gegevens opvraagt?
- Worden kredietkaartnummers in cleartext doorgezonden?

- Wordt een “slapend account” ineens terug gebruikt wat duidt op een mogelijk veiligheidsprobleem?
- Wordt toegang verschaft tot de database via een account dat normaal geen toegang zou mogen hebben?
- ...

Deze verschillende fases worden symbolisch voorgesteld in Figuur 3.

De gegevens komende uit deze analyse kunnen gebruikt worden om het veiligheidsbeleid strakker te maken (bv. bepaalde users krijgen onnodige toegang tot bepaalde gegevens).

De oplossing kan echter ook dienen om Database Administrators te helpen bij het op punt stellen van queries, debuggen van SQL-fouten, performance logging, etc.



Figuur 3: De verschillende fases / functionaliteiten van een oplossing voor Database Activity Monitoring.

3.2. Te monitoren activiteiten

Om de veiligheid aanzienlijk te verhogen, zal men typisch starten met het bewaken van volgende activiteiten [8,9].

3.2.1. Geprivilegieerde accounts

Om hun werk uit te voeren, hebben **database administrators** typisch gezien **geen toegang nodig tot data [10]**. Daarom moet er nagegaan worden:

- of ze data lezen, wijzigen of verwijderen;
- of ze toegang proberen te krijgen via ongewenste kanalen;
- wanneer ze schemawijzigingen doen;
- of ze gebruikers toevoegen of bestaande accounts wijzigen, om zo meer privileges te verkrijgen via dat account.

3.2.2. Eindgebruikers

Voor eindgebruikers die wel toegang moeten krijgen tot gegevens, is het nuttig om volgende zaken op te volgen:

- toegang tot overdreven hoeveelheden data;
- toegang buiten de “normale” werkuren;
- toegang via ongewenste kanalen.

3.2.3. Developers en technisch ondersteunend personeel

Voor developers, systeembeheerders en ander technisch ondersteunend personeel moet nagegaan worden of ze toegang krijgen tot live production databases: dit zou niet het geval mogen zijn.

Daarenboven kan DAM gebruikt worden om te kijken of de vastgelegde processen volgens plan gebeuren, door bijvoorbeeld na te gaan of er database changes gebeuren op ongewenste momenten of patching buiten de operational windows.

3.3. Voorbeelden van gebruik

Naast een extra instrument om security af te dwingen en inbreuken beter te kunnen detecteren en analyseren, kunnen DAM-tools eveneens gebruikt worden voor audit-doeleinden (bv. tot het bekomen van compliance). Meer concreet kunnen DAM-tools gebruikt worden voor onder andere volgende use cases [10].

- **Waarborgen van de separation-of-duties** – het in het oog houden van alle activiteit van systeembeheerders (queries & resultaten) en specifieke compliance-rapporten genereren.

- **Beschermen tegen geavanceerde applicatieve aanvallen** – bv. het teruggeven van meer dan één INSZ-nummer of kredietkaart-nummer kan wijzen op een aanval. Door te focussen op de data kunnen een hele reeks aanvallen op één punt aangepakt worden.

- **Het ondersteunen van (interne) audit** – de audit trails opgeslagen door DAM-tools zijn beter beveiligd dan de standaard database-auditinformatie omdat ze buiten het bereik vallen van de privileges van de database administrators. Hun datageoriënteerde karakter maakt dat ze de ideale ondersteuning zijn voor bepaalde taken van (interne) auditdiensten.

- **Het bekomen van compliance** – de sterke auditvereisten opgelegd door SOX, HIPAA, PCI, ... voor compliance zijn een van de belangrijkste redenen waarom bedrijven DAM-tools gebruiken.

- **Beschermen van gevoelige gegevens tegen dataloss (intern/extern)** – gevoelige gegevens moeten beschermd worden zowel tegen interne als externe aanvallen. DAM-tools zijn hier een excellent middel toe.

- **Inzicht verschaffen in gegevenspatronen** – DAM-tools zijn de ideale bondgenoot voor DBA's om SQL-fouten te debuggen, of om in het algemeen een beter inzicht te krijgen in de manier, de locatie en het tijdstip waarop gegevens worden opgevraagd / opgeslagen.

- **Fraudedetectie** – fraude gepleegd door werknemers kan gedetecteerd worden door toegangspatronen via enterprise applications te analyseren. Meestal zullen dan gegevens van verschillende databanken samengelegd moeten worden.

4. Marktoverzicht

Hoewel DAM-tools vooral populair zijn omwille van compliance-redenen, merken we toch op dat het geavanceerd gebruik, in de bredere context van information security en security governance, traag maar zeker ingang vindt.

Na een periode van vele *consolidaties* zijn er momenteel een paar grote dominante spelers op de markt. Steeds vaker worden DAM-tools in combinatie gebruikt en gekoppeld/geïntegreerd in andere producten voor security governance zoals Security Information and Event Management (SIEM), Privileged Account Management (PAM) en Data Loss Prevention (DLP). Gartner spreekt sinds begin 2012 van “Database Audit & Protection” eerder dan van DAM.

4.1.1. Belangrijkste spelers

Zonder een uitgebreid overzicht te willen geven, zullen we kort de belangrijkste spelers overlopen. Grotendeels bieden ze een gelijkaardige verzameling van features, meestal als (virtuele) appliances, hoewel niet alle spelers de meest geavanceerde features, zoals het opslaan van SQL responses of het afbreken van bepaalde transacties, ondersteunen.

Let wel dat deze functionaliteiten meestal komen met een extra licentie- en performancekost. Bovendien zijn deze features ook vaak te geavanceerd of te specifiek zodat ze in praktijk niet zo veel bijdragen.

- **IBM InfoSphere Guardium** is de marktleider wat omzet en aantal klanten betreft; het product bevat een uitgebreide set van features.

- **Imperva SecureSphere** volgt IBM op in de markt. De oplossing maakt deel uit van een bredere suite waar o.a. ook een Web Application Firewall (WAF) inzit.
- **Oracle** vertegenwoordigt eveneens een sterke aanwezigheid in de markt, die te danken is aan de sterke positie van de Oracle Database. Hoewel Oracle niet echt één DAM-product heeft, worden de functionaliteiten aardig gedekt door Audit Vault en Database Firewall.
- **McAfee** heeft zich met de overname van Sentrigo DAM een plaats veroverd op de markt van de Database Security. Hoewel het bedrijf nu nog een kleine speler is, maakt het een grote kans om een sterke speler te worden als het zijn sterke en grote customer base weet te overtuigen.
- **Application Security DbProtect** combineert zijn monitoring-product met de vulnerability-scanner en het gebruiksbeheerpakket tot een uitgebreid aanbod.
- **BeyondTrust** bouwt op de assets van Lumigent, een van de eerste spelers in de DAM-markt. Hoewel in het verleden de ontwikkeling wat achter liep op de andere spelers, lijkt dit verschil op heden bijgewerkt.
- **Fortinet**, eveneens met een sterke achtergrond in security, biedt een robuuste oplossing, die qua geavanceerde functionaliteit een beetje achterliep.

4.1.2. Aandachtspunten

Elk van deze spelers heeft zijn eigen voor- en nadelen (gaande van prijs tot maturiteit en integratiemogelijkheden) afhankelijk van de specifieke situatie. Toch geven we enkele aandachtspunten mee waarop gelet moet worden bij het aanschaffen van een pakket voor Database Activity Monitoring [2,4]:

1. Betrek vanaf het begin alle betrokken partners: de DBA's, de veiligheidsdienst, het auditdepartement maar zorg zeker ook voor een vertegenwoordiging van ontwikkeling.
2. Stem af welke databasesystemen als eerste beschermd moeten worden en hoever deze bescherming juist moet gaan (eventueel in lijn met bepaalde compliance-wetgeving).
3. Probeer zeker eerst een test in een realistische omgeving uit te voeren om onrealistische verwachtingen te vermijden (het is mogelijk om niet-intrusieve testen te doen) en te kunnen inschatten welke extra effort nodig is voor het werkelijke deployment (bv. hoe vlot het inbrengen van contextspecifieke regels verloopt).
4. De geavanceerde features worden in praktijk amper gebruikt dus indien dit de eerste kennismaking is met DAM, wegen deze best niet te veel door in de beslissing.

Kijk ten slotte ook naar de integratie in de bestaande businessprocessen: wat gebeurt er als er een alert gegenereerd wordt? Hoe gebeurt de integratie met het usermanagement (wie zijn de sysadmins?), met de CMDB (welke databases?) en het change management?

5. Conclusie

Met de immer stijgende complexiteit van IT-omgevingen wordt het **steeds moeilijker om te achterhalen of er data gestolen worden**, en welke dat dan zijn. Zo worden er kopieën van productiedatabanken genomen voor back-ups of testingdoeleinden en hebben geprivilegieerde accounts vaak overdreven rechten.

Oplossingen voor Database Activity Monitoring gaan radicaal voor **een data-centric approach door alle database queries te registreren**. Op deze manier geven ze een holistisch inzicht in wie waar welke gegevens raadpleegt.

Omdat ze redelijk onafhankelijk van de huidige applicaties ontplooid kunnen worden, zijn ze een **krachtig middel om doeltreffend de veiligheid te verhogen**. Hoewel DAM-tools meestal gebruikt worden omwille van compliance-redenen, zijn ze ook uitermate geschikt om de meest gevoelige gegevens te beschermen (bv. INSZ-nummers) of om de auditmogelijkheden aanzienlijk te verhogen.

De oplossingen die te vinden zijn op de markt zijn reeds erg matuur. Aangezien gegevensdiefstal een steeds groter wordend probleem is, **raden we klanten met gevoelige gegevens dan ook sterk aan om oplossingen voor Database Activity Monitoring aan te wenden.**

De sectie Onderzoek van Smals brengt met regelmaat verschillende publicaties uit over een hele waaier aan topics in de huidige IT-markt. U kan deze publicaties opvragen via het extranet

http://documentatie.smals.be

Of u kan rechtstreeks contact opnemen met het secretariaat van de afdeling "Klanten & Diensten", op het nummer 02 787 58 88.

6. Bijlagen

[1] Jeffrey Wheatman, “Establishing a Strategy for Database Security is No Longer Optional”, *Gartner Research*, 29 November 2011

[2] Rick Mogull, “Understanding and Selecting a Database Activity Monitoring Solution”, *SANS Institute*, April 2008

[3] http://en.wikipedia.org/wiki/Database_activity_monitoring, “Database Activity Monitoring”, Wikipedia

[4] Nigel Stanley, “Database Activity Monitoring”, *IT Director*, 18 February 2010

[5] Kristof Verslype, Bob Lannoy, “Security Information & Event Management (SIEM) - Privileged Account Management (PAM)”, http://documentatie.smals-mvm.be

[6] Jeffrey Wheatman en Mark Nicolett, “Database Activity Monitoring Market Overview”, *Gartner Research*, 3 February 2009

[7] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, “Securing Your Database Server”, *Microsoft patterns & practices*, June 2003

[8] Jeffrey Wheatman, **“**Database Activities You Should Be Monitoring**”**, *Gartner Research,* 14 March 2012

[9] Jeffrey Wheatman, “Ten Database Activities Enterprises Need to Monitor”, *Gartner Research*, 28 April 2010

[10] Elisa Bertino en Ravi Sandhu, “Database Security – Concepts, Approaches, and Challenges”, *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no.1, January – March 2005

[11] Jeffrey Wheatman, "The Future of Database Activity Monitoring", *Gartner Research,* 22 June 2010