Database activity monitoring

Database activity monitoring (DAM, a.k.a. **Enterprise database auditing** and **Real-time protection**[1]) is a database security technology for monitoring and analyzing database activity. DAM may combine data from network-based monitoring and native audit information to provide a comprehensive picture of database activity. The data gathered by DAM is used to analyze and report on database activity, support breach investigations, and alert on anomalies. DAM is typically performed continuously and in real-time.

Database activity monitoring and prevention (DAMP) is an extension to DAM that goes beyond monitoring and alerting to also block unauthorized activities.

DAM helps businesses address regulatory compliance mandates like the Payment Card Industry Data Security Standard (PCI DSS), the Health Insurance Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act (SOX), U.S. government regulations such as NIST 800-53, and EU regulations.

DAM is also an important technology for protecting sensitive databases from external attacks by cybercriminals. According to the 2009 Verizon Business' Data Breach Investigations Report—based on data analyzed from Verizon Business' caseload of 90 confirmed breaches involving 285 million compromised records during 2008—75 percent of all breached records came from compromised database servers.

According to Gartner, "DAM provides privileged user and application access monitoring that is independent of native database logging and audit functions. It can function as a compensating control for privileged user separation-of-duties issues by monitoring administrator activity. The technology also improves database security by detecting unusual database read and update activity from the application layer. Database event aggregation, correlation and reporting provide a database audit capability without the need to enable native database audit functions (which become resource-intensive as the level of auditing is increased)."[2]

According to a survey by the Independent Oracle User Group (IOUG), "Most organizations do not have mechanisms in place to prevent database administrators and other privileged database users from reading or tampering with sensitive information in financial, HR, or other business applications. Most are still unable to even detect such breaches or incidents."

Forrester refers to this category as "database auditing and real-time protection".[1]

Contents

Common use cases for DAM

Privileged User Monitoring: Monitoring privileged users (or superusers), such as database administrators (DBAs), systems administrators (or sysadmins), developers, help desk, and outsourced personnel – who typically have unfettered access to corporate databases – is essential for protecting against both external and internal threats. Privileged user monitoring includes auditing all activities and transactions; identifying anomalous activities (such as viewing sensitive data, or creating new accounts with superuser privileges); and reconciling observed activities (such as adding or deleting tables) with authorized change requests.

Since most organizations are already protected at the perimeter level, indeed a major concern lies with the need to monitor and protect from privileged users. There is a high correlation therefore between Database Security and the need to protect from the insider threat (http://blog.imperva.com/2011/08/insider-threats-quantifying-the-problem.html). This is a complex task as most privileged users are capable of using sophisticated techniques to attack the database - stored procedures, triggers, views and obfuscated traffic - attacks that may be difficult to detect using traditional methods.

In addition, since targeted attacks frequently result in attackers gaining privileged user credentials, monitoring of privileged activities is also an effective way to identify compromised systems.

As a result, auditors are now demanding monitoring of privileged users for security best practices as well as a wide range of regulations. Privileged user monitoring helps ensure:

• Data privacy, so that only authorized applications and users are viewing sensitive data.
• Data governance, so that critical database structures and values are not being changed outside of corporate change control procedures.

Application Activity Monitoring: The primary purpose of application activity monitoring is to provide a greater level of end-user accountability and detect fraud (and other abuses of legitimate access) that occurs via enterprise applications, rather than via direct access to the database.

Multi-tier enterprise applications such as Oracle EBS, PeopleSoft, JD Edwards, SAP, Siebel Systems, Business Intelligence, and custom applications built on standard middle-tier servers such as IBM WebSphere and Oracle WebLogic Server mask the identity of end-users at the database transaction level. This is done with an optimization mechanism known as “connection pooling.” Using pooled connections, the application aggregates all user traffic within a few database connections that are identified only by a generic service account name. Application activity monitoring allows organizations to associate specific database transactions with particular application end-users, in order to identify unauthorized or suspicious activities.

End-user accountability is often required for data governance requirements such as the Sarbanes–Oxley Act. New auditor guidance from the Public Company Accounting Oversight Board for SOX compliance has also increased the emphasis on anti-fraud controls.

Cyberattack Protection: SQL injection is a type of attack used to exploit bad coding practices in applications that use relational databases. The attacker uses the application to send a SQL statement that is composed from an application statement concatenated with an additional statement that the attacker introduces.[3]

Many application developers compose SQL statements by concatenating strings and do not use prepared statement; in this case the application is susceptible to a SQL injection attack. The technique transforms an application SQL statement from an innocent SQL call to a malicious call that can cause unauthorized access, deletion of data, or theft of information.[3]

One way that DAM can prevent SQL injection is by monitoring the application activity, generating a baseline of “normal behavior”, and identifying an attack based on a divergence from normal SQL structures and normal sequences. Alternative approaches monitor the memory of the database, where both the

database execution plan and the context of the SQL statements are visible, and based on policy can provide granular protection at the object level.

Core features of DAM

As defined by Gartner, "DAM tools use several data collection mechanisms (such as server-based agent software and in-line or out-of-band network collectors), aggregate the data in a central location for analysis, and report based on behaviors that violate the security policies and/or signatures or indicate behavioral anomalies. DAM demand is driven primarily by the need for privileged user monitoring to address compliance-related audit findings, and by threat-management requirements to monitor database access. Enterprise DAM requirements are beginning to broaden, extending beyond basic functions, such as the capability to detect malicious activity or inappropriate or unapproved database administrator (DBA) access." [4]

More advanced DAM functions include:

- The ability to monitor intra-database attacks and back-doors in real time (such as stored procedures, triggers, views, etc.)
- A solution which is agnostic to most IT infrastructure variables - such as encryption or network topology
- Blocking and prevention, without being in-line to the transactions
- Active discovery of at-risk data
- Improved visibility into application traffic
- The ability to offer database activity monitoring in virtualized environments, or even in the cloud, where there is no well-defined or consistent network topology[5]

Some enterprises are also seeking other functions, including:

- Configuration auditing to comply with audits required by the U.S. Sarbanes-Oxley Act
- DLP capabilities that address security concerns, as well as the data identification and protection requirements of the Payment Card Industry (PCI) and other data-centric regulatory frameworks
- Database user rights attestation reporting, required by a broad range of regulations
- The ability to offer database activity monitoring in virtualized environments, or even in the cloud, where there is no well-defined or consistent network topology
- Better integration with vulnerability scanning products

Common DAM architectures

Interception-based: Most modern DAM systems collect what the database is doing by being able to "see" the communications between the database client and the database server. What DAM systems do is find places where they can view the communication stream and get the requests and responses without requiring participation from the database. Database Security Proxy is a non-intrusive method for DAM. The interception itself can be done also at multiple points such as the database memory (e.g. the SGA), at the network (using a network TAP or a SPAN port if the communication is not encrypted), at the operating system level, or at the level of the database libraries.[3]

If there is unencrypted network traffic, then packet sniffing can be used. The advantage is that no processing is done on the host, however the main disadvantage is that both local traffic and sophisticated intra-database attacks will not be detected. To capture local access some network based vendors deploy a probe that runs on the host. This probe intercepts all local access and can also intercept all networked access in case you do not want to use network gear or in case the database communications are encrypted. However, since the agent does not do all the processing — instead it relays the data to the DAM appliance where all the processing occurs — it may impact network performance with all of the local traffic and real-time session termination may be too slow to interrupt unauthorized queries.

Memory-based: Some DAM systems have a lightweight sensor that attaches to the protected databases and continuously polls the system global area (SGA) to collect SQL statements as they are being performed. A similar architecture was previously used by performance optimization products that also used the SGA and other shared data structures.[3]

In the latest versions of this technology a lightweight sensor runs on the host and attaches to the process at the OS level to inspect private data structures. The advantages of this approach are significant:

- Complete coverage of all database transactions — the sensor covers traffic coming from the network, from the host, as well as from back-doors (stored procedures, triggers, views)
- A solution that is agnostic to most IT infrastructure variables - no need to re-architect the network, to open span ports or to worry about key management if the network is encrypted, and this model can also be used to protect databases deployed in virtualized environments or in the cloud

Log-based: Some DAM systems analyze and extract the information from the transaction logs (e.g., the redo logs). These systems use the fact that much of the data is stored within the redo logs and they scrape these logs. Unfortunately, not all of the information that is required is in the redo logs. For example, SELECT statements are not and so these systems will augment the data that they gather from the redo logs with data that they collect from the native audit trails as shown in Figure 3. These systems are a hybrid between a true DAM system (that is fully independent from the DBMS) and a SIEM which relies on data generated by the database. These architectures usually imply more overhead on the database server.[3]

References

1. The Forrester Wave: Enterprise Database Auditing And Real-Time Protection, Q4 2007, October 2007, Jonathan Penn, Katie Smillie, Forrester Research (http://www.forrester.com/rb/Research/wave%26trade%3B_enterprise_database_auditing_and_real-time_protection%2C/q/id/41970/t/2,)
2. Pattern Discovery With Security Monitoring and Fraud Detection Technologies, Mark Nicolett, Avivah Litan, Paul E. Proctor, 2 September 2009, Gartner Inc. (http://www.gartner.com/DisplayDocument?id=1160514)
3. HOWTO Secure and Audit Oracle 10g and 11g, Ron Ben Natan, Ph.D., CRC Press, 2009 (https://www.amazon.com/dp/1420084127)
4. Database Activity Monitoring Market Overview, Jeffrey Wheatman, Mark Nicolett, 3 February 2009, Gartner Inc. (https://www.gartner.com/doc/873513/database-activity-monitoring-market-overview,)
5. Monitor Amazon Aurora Database Activities 21 February 2018, Amazon AWS (https://aws.amazon.com/blogs/database/monitor-amazon-aurora-database-activities-using-datasunrise-database-security)

[1]

1. Database Activity Monitoring- Increased Visibility Into Database Activity (https://www.datasunrise.com/activity-monitoring)
