A multi layered approach to prevent data leakage

Contents

Databases remain one of the least protected areas in the enterprise

Skilled malicious hackers are no longer interested in getting millions of people to open up e-mailed attachments that will then pester everyone listed in an infected machine’s e-mail address book. Instead these people are becoming more business-like, concentrating on opening new streams of revenue for themselves by directly targeting and penetrating networks to grab data that they can use, or sell for profit.

Databases hold much of the most sensitive and valuable datainformation about customers, transactions, financial performance numbers and human resource data to give a few examples. Despite this, databases remain one of the least protected areas in the enterprise. While perimeter and network security measures create a barrier against some type of attacks, there are attack patterns that take advantage of database-specific vulnerabilities.

An open invitation to breach the database Since database management systems are complex, supporting an ever growing set of requirements and platforms, with addition of features they develop gaps in security– vulnerabilities–that are constantly being discovered by users, ethical hackers and unfortunately, non-ethical hackers as well. Such vulnerabilities are reported to DBMS vendors who do their best to patch them, but this is a process that currently takes several months on average, and in some cases years. That time lag is essentially an open invitation to exploit the vulnerability and breach the database.

The scenario reminds me of Willie Sutton, the bank robber. He answered the question why he robbed banks with–that's where the money is. He did not use the approach to stand at street corners to grab money from people passing by. A widely used approach with current ATM systems is limiting the amount that you are allowed to withdraw in each transaction and for each day.

A layered approach to prevent data leakage

A layered approach can be very powerful in preventing data leakage. This approach should start with strong protection at the source, locking down sensitive information in critical databases. This should be combined with a monitoring and blocking capability, at the database query level, that can prevent all internal and external users, including database administrators from accessing data beyond the limit defined in their respective profiles. An enterprise solution should be able to monitor and block the data access volumes and transaction volumes at the application layer, database layer and file system layer. A comprehensive solution should also be able to dynamically escalate threat warnings across the applications, databases and file systems that are part of the dataflow for sensitive information. These different components can then respond with deeper analysis and activate a more restrictive policy for each access request that is targeting sensitive data.

It is usually fairly easy to find and lock down all major databases that store sensitive information like credit card numbers and customer information. This is an important first step since many information leaks–even those that eventually occur via stolen laptops or e-mailing sensitive information–

typically originate with queries to critical databases with sensitive information. This approach can effectively limit the amount of sensitive data that is leaking out from sensitive central data stores to various distributed data stores.

1. New patterns of attack

Just as there are new attackers, there are new patterns of attack. External hacking, accidental exposure, lost or stolen backup tapes, and lost or stolen computers are still significant sources of data leakage. But database attacks are often launched with the active participation of authorised insiders who extract critical data by abusing privileges, hacking application servers and SQL injections. Even well-protected databases may offer applications broad access privileges, beyond those granted to any individual. Access through an application may effectively circumvent infrastructure-based defences. So-called “ home-user” attacks inject SQL commands into otherwise innocuous fields, compromising database security from outside corporate networks. Among the most dangerous avenues of attack, this is also one of the oldest: a trusted but untrustworthy employee applying broad access privileges. Many organisations have formal access policies and processes that govern how and when sensitive data is accessed, but lack practical and cost-effective solutions for detecting or blocking activities that fall outside these policies.

Database attacks are often launched through insiders

Database breaches–often attacks by organized criminals working through authorized insiders–target valuable concentrations of business-critical

information. Business impacts are immediate and profound, and damage to company and personal reputations can last for years. Database breaches are a growing component of IT Risk. There is growing recognition that the “ insider threat”, and specifically the threat posed by users with privileged access, is responsible for a large number of data breaches. According to annual research conducted by CERT, up to 50% of breaches are attributed to internal users. The 2006 FBI/CSI report on the insider threat notes that two thirds of surveyed organizations (both commercial and government) reported losses caused by internal breaches, and some attributed as much as 80% of the damage to internal breaches. It was also reported that 57% of implicated insiders had privileged access to data at the time of breach. It is therefore evident that perimeter and network security measures are not enough to stop such breaches. Driven by the consolidation of valuable information and the professionalization of computer crime, database attacks are often launched through insiders with full authorization to access the information they steal. Both the Computer Security Institute and the FBI survey document the rising incidence of such attacks. Infrastructure security solutions such as perimeter-based defenses, access controls, and intrusion detection can do little against authorized insiders. And the fact remains that no screening and authorization process can be perfect.

Unauthorized behaviors by authorized and unauthorized users

Database attacks represent unauthorized behaviors, by both authorized and unauthorized users. As we have seen, authorized insiders constitute a major threat against information safety and integrity. Barring a perfect screening

process, no permission-based, asset-centric security system can close this fundamental vulnerability.

The problem grows worse

Business enterprises and security companies are in the early stages of their response to the resurgence of threats to their information assets. Yet as they struggle toward solutions, the problem grows worse. More information is made available to customers, partners, and suppliers through Web portals, often linked to critical databases. Companies integrate customer-facing applications such as customer relationship management, service provisioning, and billing more tightly, spreading critical information more widely within and across organizations. In addition more businesses outsource and offshore critical business processes to new “ insiders” who may not meet their own organizations’ internal screening processes. Increasingly automated management of intellectual property, for example in pharmaceutical companies and genetic research, may put corporate assets of significant value in highly-accessible databases Virtually any organization, public or private, is at risk of public embarrassment, financial loss, and government investigation when critical information is stolen or compromised.

More complexity–more issues

Attack an application often enough and you’re bound to find exploitable holes. Databases complicate the issue by being complex beasts that feed information to and from other applications–some vendor-supplied and others

perhaps created in-house or via supplied APIs. The more complex an application becomes, the more likely it is to harbor hidden holes.

2. New security requirements

Security is shifting from protecting the device and learning about individual users to thinking about the policies that I deploy around user interactions and information protection, and having policy management techniques and technologies that give me warnings or block access or activity when it doesn’t conform to what I had prescribed.

Organizations will need multi-layered ways to defend their sensitive information

As the Web has become a ubiquitous operating tool, the risks to businesses have multiplied. If online infrastructures are not protected and have unsecured entry points, companies both large and small are putting their networks at risk. While firewalls are common in every organization, they are no longer sufficient to ward off hackers intent on stealing confidential information. Organizations now realize that they need to have a solid online security policy in place to assure consumers and trading partners that their information is safe.

Blocking based on the volume of data accessed

The defining security requirement for Data-layer security is the ability to detect out-of-policy data access by outsiders or even authorized insiders, through direct access to the database itself, or over networks including the Web. Alerts and blocking based on comparisons with historical patterns of

usage, provides continuous, actionable exception monitoring of transactions that may contain protected data. Solutions must monitor and block out-of-policy transmission of typical patterns such as credit card numbers, Social Security numbers, patient record identifiers and other patterns as defined by enterprise administration.

3. Limitations of traditional approaches

There is a wide array of technologies currently in use for securing databases. As with other areas of IT security, no single tool can provide ironclad defense against all threats and abuses. It is always recommended to employ a combination of tools to achieve adequate security. Traditional perimeter and asset-based defenses won’t work effectively in environment in which perimeters are indistinct and constantly changing, where attacks are marshaled against data, not assets, and where the most likely threats are from fully-authorized insiders with the capacity to circumvent or neutralize defenses.

Perimeter-based defenses offer little protection for critical information
Perimeter-based defenses such as ftrewalls and intrusion-detection

systems are the bedrock of IT security and more necessary than ever, but they offer little protection for critical information stored in databases. First, they are ineffective against attacks by insiders with full authorization to operate inside defended perimeters. When the organization’s trust in its authorized personnel is justified, perimeters are unlikely to provide the same degree of protection as in the past. With the security perils of mobile systems, wireless networks and peer-to-peer “ sharing” networks, high-

capacity USB “ thumb” drives, portable hard drives, and other mobile storage devices, with an array of mechanisms to move information across networks without detection, perimeter defenses can do little.

Identity management and access controls are difficult to design and maintain

Unfortunately, it is very common within enterprises to have group usernames and passwords and to forget to revoke privileges of employees who no longer need them. This mechanism is also exposed to hacking (e. g., SQL injections that escalate privileges). Role-based rather than behavior–based, access controls and permissions are difficult to design and maintain. Among other problems, “ permission inflation” gradually weakens protections over time as individuals acquire new permissions when their job roles change. Access controls also seldom apply to access through applications, for example through SQL injection.

Monitoring using network appliances

Monitoring using Network Appliances can provide alerts (and if used in-line, prevention) on network access to the database, but do not protect against insiders with access privileges / local access. They often require network reconfiguration, and if used in-line slowly create a network bottleneck that cannot handle encrypted traffic or expensive hardware. This is a class of network-based appliances, which monitor network traffic looking for SQL statements, and analyze the statements based on policy rules to create alerts on illegitimate access to the database and attacks. Because the appliance is only monitoring the network it does not have visibility into local

database activity, essentially leaving the database vulnerable to insiders that either have local access or are savvy enough to bypass the appliances. In order to provide adequate monitoring, the appliance must be deployed at every choke point on the network where the database is accessed, encircling the database from all sides.

For mission-critical databases that are often tied into a multitude of applications (ERP, CRM, BI, billing etc.), this significantly raises the cost, which is high to begin with.

Slow and imperfect protection with Intrusion detection and audit

Intrusion detection on database servers can’t resolve authorized from unauthorized queries. On networks, intrusion detection protects only information in transit from very narrow types of attack. While absolutely necessary for an effective IT policy or regulatory compliance program, audits tend to be resource-intensive, consuming a great deal of time and effort and cutting into system performance while underway. And unless audit data is accurate and mapped clearly to data instead of infrastructure, and itself protected from attack, audits offer slow, imperfect protection against internal attack. Unfortunately, neither perimeter defenses, employee screening, nor information-focused security can prevent accidents–as when a laptop containing credit card numbers is misplaced.

Only information-centric security can help managers and auditors understand what information was lost in order to guide notification and remediation efforts.

Native database audit tools

Native Database Audit Tools provide a granular audit trail and forensics of database activity, but come with a serious database performance impact. They offer only after-the-fact forensics and have no prevention capabilities, no separation of duties and are easy to turn off. Most DBMSs come with a set of features that enable granular auditing of every single database activity. These features are seldom used because of the negative impact they have on performance. Furthermore, because they are part of the DBMS, they are administered by DBAs in a way akin to “ letting the cat guard the cream”.

Data eneryption adds an essential level of protection

Database activity monitoring cannot provide protection against privileged insiders or malicious users, but policy driven encryption of database fields can offer good protection of information. Encryption systems should be controlled by a separated policy and also linked to a multilayer protection approach. Data encryption adds an essential level of protection from intruders who manage to break through primary defenses, and also ensures data is seen on a need-only basis as determined by access permissions protecting it from exposure to authorized and unauthorized users. Encryption is a necessity in all situations in which customers can perform (or authorized users are provided access to) transactions involving confidential information stored in a database. Any decent security program must ensure that secure automated encryption management–including secure encryption key protection, aging, and replacement–is implemented across all platforms hosting critical information. The best cryptographic architecture will be

flexible and modular so that it can be easily adapted to various situations across the enterprise. The challenge, as always, is to find the right balance between security and usability. There is no one perfect architecture for all companies, as business policies and associated compliance issues will determine what data needs to be protected and what methods to use. The important thing is to be willing to go beyond compliance basics and develop a workable and comprehensive plan to secure data that suits the needs of your company.

4. Solutions for multi-tiered applications

Privileged access to critical databases. Asset-centric approaches to database security can actually increase risk by wasting time, effort, and focus on solutions unlikely to slow information loss and corruption. Innovations such as multi-level applications, multi-tier storage, and service-oriented architectures (SOA)–often with privileged access to critical databases–raise the complexity and vulnerability of critical data structures. In any of these environments, the mapping of information onto infrastructure assets is complex, and changes constantly. Asset-focused policies, alerts, security logs, and reports are complex and interdependent, and may even be irrelevant for protecting data, and documenting compliance to data-focused policies and regulations.

Who is the real user?

Current data security systems for data at rest, whether they are implemented as separate server appliances, co-located with one or more applications on the same host machine, or co-located with data services

machines such as database servers, operate in real-time, intrusively in-line with the data they protect. When sensitive, encrypted data is requested by applications rather than directly by authenticated users, the “ legitimate user” is frequently no more than the application name itself. Even in cases where an actual username is passed from the application along with the data request, the data security system is “ blind” to whether or not the user is a hacker or has stolen legitimate user credentials.

A behavioral policy can restrict access even if the real user is not identified

Data security systems are not configured to take advantage of application security events detected elsewhere in the environment in the same approximate timeframe. Although correlation with those events is a typical practice when auditing the forensics of events via log files, long after the events have occurred. Some approaches track who the real application user is based on a probability analysis across concurrent processes that are accessing the data. Other solutions can completely track the user but these solutions are application aware–either based on an application API or a plug-in that is specific to each application environment. The behavioral policies restricting access to data are analyzing access patterns and does not require that the real end user is identified.

5. Solutions for Web based applications

Buffer overflows, SQL injection and Cross Site Scripting. Buffer overflows and SQL injection aren’t new, but attackers still manage to make effective use of them to gain access and administrative privileges to databases. Intrusion prevention systems are of use in dealing with buffer overflows. SQL injection

is a popular method of attack, since modern databases utilize SQL–Structured Query Language–to enable users to access and manipulate data stored in a database. The basic procedure for a SQL injection exploit is to provide a valid request at the beginning followed by a single quote and a “;” with an additional request appended which contains the actual command the attacker hopes to implement. By piggybacking the “ bad” code onto the good code it’s possible to trick an incorrectly configured database into carrying out unauthorized executions. Cross site scripting occurs when a web application gathers malicious data from a user. The data is usually gathered in the form of a hyperlink which contains malicious content within it. The user will most likely click on this link from another website, instant message, or simply just reading a web board or email message. Usually the attacker will encode the malicious portion of the link to the site in HEX (or other encoding methods) so the request is less suspicious looking to the user when clicked on. After the data is collected by the web application, it creates an output page for the user containing the malicious data that was originally sent to it, but in a manner to make it appear as valid content from the website.

Latency issues with traditional application firewalls

Web application firewalls are often the easiest way to protect against these sorts of exploits. Code audits in-house, or by an outside expert can also spot and close SQL vulnerabilities. Most application firewalls, whether they are implemented as separate reverse-proxy server machines, co-located with the application on the same host machine, or co-located with network firewall machines, generally operate in real-time, intrusively in-line with the

applications they protect. This introduces latency while the application firewall examines the traffic, logs the activity, alerts IT Operations and/or network firewalls to suspected attacks and passes traffic on to the application. Additional latency is introduced when HT-TPS traffic is examined. For instance, secure socket layer (“ SSL”) protocols used in HTTPS are terminated and decrypted prior to examination; in some implementations, traffic is additionally encrypted again before passing traffic on to the Web, application, and/or database servers for final HTTPS termination. Application firewalls are not configured to take advantage of security events or behavioral anomalies detected elsewhere in the environment in the same approximate timeframe, although correlation with those events is a typical practice when auditing the forensics of events via log files, long after the events have occurred.

Web application firewalls combined with an escalation system

Automated, synchronized threat monitoring and response between the application level and database level provides a highly effective protection against both external and internal attacks. An escalation system that can dynamically switch Web application firewalls between different protection modes is described below.

6. Behavioral policy layers can restriet data access

Control database queries that returns thousands of credit card numbers. Unlike monitoring tools that only inspect inbound database commands; this approach identifies unauthorized or suspicious actions by monitoring traffic both to and from database servers. This allows the

solution, for example, to immediately identify a database query that returns thousands of credit card numbers, thereby deviating from data access patterns.

How to understand the true extent of data theft

A Policy Engine that monitors outbound responses from the database can detect suspicious data access patterns, based on volume of returned records. Data Usage Policies are typically used to detect activities by authorized users that fall outside normal business processes. Information collected by the Data Usage Policy Engine can also be used to understand the true extent of data theft, thus minimizing breach disclosure efforts and costs. A solution may provide access and security exception policies that monitor inbound database commands for unauthorized actions, such as database changes, failed logins, and SELECT operations by privileged users.

Control the amount of data that is accessed

Protection rules control the amount of data that is allowed for each user to be accessed in certain time windows. The item access rates define the number of database records, file blocks or web transactions that is allowed for each connection in a time window. The item access rates can be defined based on the number of rows a user may access from a database column. For example if a query result exceeds the item access rates, the request is blocked before the result is transmitted to the user.

Prevent the result of the query to be accessed by the user

The method for detecting intrusion in a database can be based on an intrusion detection profile, with a set of item access rates, which includes a definitive number of rows that may be accessed in a predetermined period of time for each user. When a query is exceeding an item access rate defined in the profile user authorization the result of the query is prevented from being transmitted to the user.

Data inference policy rules

A variation of conventional intrusion detection is detection of specific patterns of information access known as inference detection. Inference detection is deemed to signify that an intrusion is taking place, even though the user is authorized to access the information. Results from performed queries are accumulated in a record, which is compared to the inference pattern in order to determine whether a combination of accesses in said record match the inference policy, and in that case the access control system is notified to alter the user authorization, thereby making the received request an unauthorized request.

Machine-learning from accepted patterns and past intrusions

The behavioral policies restricting access to data can utilize machinelearning from accepted behavioral patterns and from previous intrusions in order to better predict future intrusions.

7. A multi-layered data defense system

A layered approach to security No single approach to securing a system. will be able to defeat each and every new and innovative intrusion attempt by

insiders and/or outsiders. That’s why we deploy layers of protection. If one or two fail another will withstand the attack, or at least slow down the criminal who is likely to give up and ransack a more vulnerable target. Many crimes, including network attacks, are crimes of opportunity and the easy way in and easy way out becomes the thief’s preferred modus operandi.

Data-layer protection

A Data-layer protection approach monitors all requests for sensitive data access for critical data such as credit card and Social Security numbers, patient identifiers or custom patterns. Comparisons against policy and history identify exceptions and anomalies in real time, and provide a comprehensive audit trail to document compliance. Any genuine long-term solution must be flexible and include options to balance protection level against database performance and other operational needs.

A Multi-layer security advisory framework

A Multi-layer Security Advisory System provides a framework to effectively deal with threats of some classes of attacks. The warning system has 5 risk-of-attack-levels (Threat Levels) which when triggered, initiate specific actions by local servers within the same policy domain. Information about data security events is collected from sensors at different system layers (web, application, database and file system). The Threat Level is propagated to systems that are connected within a data flow. The Threat Level will also adjust for time of day, day of week, and other factors that are relevant.

A Score-card to keep track of usage abnormalities

A score-card is maintained for each subject (user or service account/proxy-user, ip-address, application, process ...) and object (database column, file ...) with a history of processing sensitive data. The score-card summarizes current and historical information about data access patterns for each entity (subjects and users). The score-card also includes a ‘ finger-print’ that reflects historical deviation from acceptable access patterns at the level of s/i/u/d (select/insert/ update/delete) operations. A high score-card value will initiate more extensive analysis before releasing data to the subject. The dynamic and automatic altering of the protection policy between multiple system layers includes modifying the protection policy of data at one or several of the system layers. The modification is performed based on a result of the prevention analysis. The score-card can also keep track of when a remote system need to reconnect to the central system to renew or recharge it’s capability to encrypt and decrypt data. The policy may allow the local system to only operate stand alone for a certain time or processing a fixed number of crypto operations between each host connection and central password renewal. This behavior will act like a rechargeable key box and can automatically shut down the local access to sensitive data in case the local system is stolen, cloned or compromised in some other way.

Escalation in a multi-node security system

This is a method for achieving cooperative processing and control of application-layer security by using loosely and tightly coupled nodes of application firewalls, application monitors and data security enforcement points together with operational and escalation rules. For example a SQL Injection attack at the application layer can automatically switch the Web

Application Firewall from monitoring mode to inline mode to block certain requests. This will provide a dynamic and automatic altering of the protection policy.

Escalation in a mufti-layer security system

The dynamic and automatic altering of the protection policy between multiple system layers includes modifying the protection policy of data at one or several of the system layers. The modification is performed based on a result of the link prevention analysis. For example a SQL Injection attack at the application layer can automatically put the connected backend databases in a higher alert-level (System Threat Level). The higher alert-level can switch to a protection policy that may turn on additional logging and alerting and potentially also block certain requests when the score-card is out of balance.

Balance performance and protection

In meeting the requirements above, Data-layer protection must be flexible to adapt to different requirements related to protection level, performance, scalability and other operational needs. A multi-layer solution can balance performance against the level of protection against internal threats, and can minimize required modifications to the database or associated programs. The Multi Layered Database Security approach to data security provides policy-driven, data protection in real time with customizable balancing between zero performance impact and full protection against internal threat against data at rest.

Selective activation of the intrusion analysis

Access to selected data columns or files can trigger a deeper intrusion analysis process. This is especially advantageous if only a few items are intrusion sensitive, in which case most queries are not directed to such items. The selective activation of the intrusion detection will then save time and processor power.

Dynamically switch between monitor and in-line operation

The Leakage Prevention solution can dynamically block the transaction output results that violate security policies. This can be accomplished by dynamically switching the solution between in-line database gateway operation and operating as a passive monitoring device that initiates other enforcement actions such as transaction blocking, automated logouts of database users, VPN port shutdowns, and realtime alerts.

Conclusion

The proposed Multi Layered Approach to prevent Data Leakage meets many fundamental requirements of organizations to protect their critical data from loss, leakage, and data fraud. Data leakage can be minimized by real-time detection and blocking of leakage of sensitive company information– including analysis of all sensitive data leaving the database, so companies can react immediately to policy violations. Fraud from insiders abusing privileges can be minimized from analysis of behavior against established policies and access history to identify anomalous behavior, even by authorized users, so that organizations can achieve “ defense in depth” for all sensitive data under their care.

The approach can provide protection against poorly-written applications that open vulnerabilities to critical databases and files. The approach can also provide an alternative to some of the frequent patching of critical systems. In addition to dynamically providing minimal and adjustable performance impact, this approach can offer flexibility and dynamic features that can switch to use selected security features when an escalation is needed. To assure timely response, solutions should provide real-time tracking and blocking, not relying solely on alerts or reports after the fact. In addition, audit data should be archived off of the server holding the data, so that the audits themselves are not vulnerable even in the event of a database breach.

About the author: Ulf Mattsson is chief technology officer with Protegrity having created the initial architecture of Protegrity ‘ s database security technology. His extensive IT and security industry experience includes 20 years with IBM as a manager of software development and a consulting resource to IBM’s Research and Development organisation, in the areas of IT Architecture and IT Security. Ulf holds a degree in electrical engineering from Polhem University, a degree in Finance from University of Stockholm and a master’s degree in physics from Chalmers University of Technology www. protegrity. com