(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
5 October 2006 (05.10.2006)

PCT

(10) International Publication Number
WO 2006/103656 A2

(51) International Patent Classification:
G06F 21/06 (2006.01)

(21) International Application Number:
PCT/IL2006/000380

(22) International Filing Date: 27 March 2006 (27.03.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/665,355 28 March 2005 (28.03.2005) US

(71) Applicants and
(72) Inventors: **GOLDHIRSH, Ofer** [IL/IL]; 17 Haoranit St., Pob 314, 40600 Tel Mond (IL). **POYARKOV, Michael** [IL/IL]; 119 Hapalmach St., 60920 Kadima-zoran (IL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to the identity of the inventor (Rule 4.17(i))*
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*
— *of inventorship (Rule 4.17(iv))*

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** DATABASE SECURITY PRE AND POST PROCESSOR

(57) **Abstract:** Abstract of The Disclosure A novel database pre and post processor appliance and method that function as a go-between mechanism. It is located in line between the database and the database client, stealthy manipulating the traffic between these elements enforcing rules and policies, applying new restrictions and functionalities without disturbing the traffic or affecting the connection. It allows implementing new level of security and functionality to the database and database client relation without the need for assistance from the DBAs or the system programmers. It also allows a real separation of responsibilities between the security administrator and the DBAs and system programmers.

WO 2006/103656 A2

Database Security Pre and Post Processor

CLAIM OF PRIORITY

This application claims priority to U.S. provisional application Ser. No. 60/665355, filed March 28, 2005 entitled "Database Security PreProcessor" by Ofer Goldhirsh and Michael .Poyarkov.

Background of The Invention

Field of The Invention

The present invention relates to a go-between device or a software module that is designed to transparently manipulate traffic between database and application without breaking or affecting the database and database-clients connection and further without substantially affecting database performance. It provides an effective means to externally enforce policies and new methods without the need to change the database or the application. This innovation provides tools to protect databases from all kinds of malicious activities, damage, data theft etc and tools to improve performance, redundancy and load balance and apply new requirement without the need to change the database or the application

Description of the Related Art

Databases are in use in many organizations for a variety of applications. Typically, the database is comprised of many tables. Each table can be accessed by rows, columns, or specific cells. This type of database is usually referred to as a "Relational Database" and is by far the most prevalent. Other, less common types of databases also exist, and this description includes them as well. This access is done by a computer running an application that uses the database and is referred to as the Database Client. This client may be hosting connections from other computers for a different purpose, enabling database access for its clients, while the computer hosting the database itself and managing the access to the database, including security issues, is known as the Database Server.

Many problems arise from the need to protect a database – While the database server is designed for some self-protection, it performs it on a rather superficial level, not going into too much detail. Additionally, the database server, in most cases, does not inspect query answers. Once database access is granted, it will not be revoked, and the usage resulting from that access is not inspected thoroughly. In addition, the very detection of database access (query, stalled procedure, RPC - Remote Procedure Call or any other manner of using or accessing the database, the data held therein or any procedures or methods held therein) of a malicious nature is hardly a straight-forward affair. Such queries can come in many forms – malicious manipulation of the database, non-restricted access, deceptive use of "wildcards", erroneous queries made intentionally to gather information on the database, erroneous operations made mistakenly, users breaching their access rights, etc. Data required for making queries (row name, for example) might contain sensitive information. Queries might be discovered as illegal only in light of user history or profile, and many others.
Furthermore, the malicious or illegal nature of database access might only be discovered by the result the database creates for it. Some queries might be perfectly safe yet because it was made in a testing environment would require the retrieval of false data. Otherwise legal queries might retrieve information that is not to be revealed. There is no current method for prevention of queries by their reply. Many query's malicious nature is only revealed by their reply.

Moreover, current techniques are based on firewall technologies of traffic scanning or SQL query parsing and scanning, and in case of detection of a malicious packet or query, firewall blocking techniques are applied. Recently, a number of new technologies of on-line database traffic monitoring, profiling and inspection of database traffic were introduced. All of those technologies provide profiling, detection and alarming, but their proactive mode follow the firewall techniques of breaking

database/client connections. They do not implement transparent traffic content nor implement new requirements on the database/client connection.
They also do not operate proactively in a stealth mode. Some of them have some proxy functionality.

The firewall blocking techniques are based on dropping packets or sending a reset to the session participants. These blocking techniques cause a break in the connection between the database and the database client. Due to the database and database-client connection nature (impersonation etc.) it typically holds large number of sessions. Dropping the connection stops the service to all users that are using that connection for a relatively long time. Database and database client connection recovery time can last several minutes. This is one of the main reasons that blocking techniques are rarely used on the connection to the database.

In addition, while security measures are the responsibility of the security administrator, if they were to be applied on the Database Server, they also become the responsibility of the Database Administrator, whose main interest lies in high data availability and performance, which tend to conflict with security concerns. If they were to be applied on the database-client (application servers) they are in the domain of the application programmer, whose main interest lies in the application usability, flexibility and performance. Frequent security changes and requirements are extra work with no benefit for them.

Yet, laws have been made to oblige organizations using computerized databases to act in order to protect it, and to be made liable if they fail to do so. Organizations need to react to a growing frequent demand for security changes and requirements. They need to bring all their systems, legacy and new, to the needed level of security. In most cases organizations need to manage these security changes and updates in large numbers of applications. Organizations need to reduce the risks, the cost and the performance impact of the new security requirements.

Currently, conventional designs for the protection of a database are either hosted on the Database Client or the Database Server, or otherwise do not act in a way that does not interrupt the regular data flow, nor do they deal with all the various aspects of database protection.

Conventional security or application changes are made on the database and/or the database client, then retested before deployment and in case of a problem, rollback procedures are enacted on the databases and the applications.

Conventional changes are applicable to a specific application or a specific database or database object.

Conventional performance issues relate to the database (and the storage array)and the database-clients (application servers) performance .

Other solutions do not address the problems discussed above, or provide the novel solutions disclosed by this invention. For example, U.S. Patent Number 5,606,668, Inventor Gil Shwed describes a system for securing inbound and outbound data-packet flow in a computer system. Unlike Shwed's patent which does packet-filtering and dropping or rejecting non-compliant packets, which breaks the connection, the solution herein proposed does data manipulation and preservation of the connection, Patent number 5.835.726, also by Shwed, describes filtering according to networking protocol parameters, unlike the herein proposed, which filters by rules of access to the database set by company policy, not related to the network activity. Shwed's approach is on the basis of from where the network is accessed, whereas the herein proposed solution is dependent upon how the database is accessed, and what are you allowed to see with that level of access. Another approach is described by U.S. Patent Number 6,769,074, Inventor Lev Vaitzblit and describes a system and method for transaction-selective rollback restriction of database objects. This patent deals mainly with data integrity, whereas the herein proposed patent describes a rollback for the dramatic increase in security performance. Another approach is provided by U.S. Patent Application Number 6,999,956 Inventor Ward Mullins, describing a dynamic object database manipulation and mapping system.This patent deals with correlating a transaction on one type of database to another type of database or to an object-programming application. The herein proposed solution deals within the same database, re-routing database objects to another table within the same database.. Another approach is U.S. Patent number 6,996,589, Inventor Jayaram Nandagopal Mysore, which describes a conversion engine between databases. The herein proposed solution deals with conversion of database object names and functions between the database client and the database. itself. Another approach is U.S. Patent Application Number 20050097149, Inventor Lev Vaitzblit, which describes a database audit system. The herein proposed patent deals with enforcing rules via an external device, in addition to the fact that the herein proposed solution was presented provisionally on March 28, 2005, Application number60665355. Another approach is U.S. Patent Application Number 20050120054, Inventor Amichai Shulman, which describes a dynamic learning method and activity/normal behavior profile architecture for providing fast protection of enterprise applications. This solutioin is based on traffic statistics, whereas the herein solution deals with pre-defined corporate policy enforcement, that may not be related to statistics. Another approach is U.S. Patent Application Number 20060059154 Inventor Moshe Robb, which describes database access security. This solution describes a firewall access mechanism, whereas the herein solution describes policy enforcement while preserving the connection. Another approach is U.S. Patent Application number 2006004155 Inventor Stephen W. Blessin, describing a system and method of implementing security on a database. This solution deals with

protecting data parcels by restricting user access to parcels of data in the database, whereas the herein solution applies an external unified policy of permissible data views. Another approach U.S. Patent 5,958,015, Inventor Ziv Dascalu, which describes a network session wall passively listening to communication sessions with use of access rules, breaking the connection. The herein proposed solution does not break the connection. Another approach is U.S. Patent Application 2005237955, a method or system of connecting manipulation equipment between operator's premises and the internet. This solution differs in the location and is unrelated to database content, unlike the herein solution. Another approach is U.S. Patent Application 20050044422, Inventor Craig Cantrell, which describes an active network defense system and method. This solution does traffic manipulation blocks the session, whereas the herein solution preserves and maintains the session and connection.

None of these patents solve the problem of policy enforcement while preserving the connection between the database and the database client, and allows transparently adding new restrictions and functionalities to the relationship between the database and the database client.

In these and in other respects the Database Security Pre and post processor substantially departs from conventional concepts and designs of the prior art and in so doing provides a method and apparatus for the effective protection of databases , the overall system performance and the flexibility, safety, and the cost of security and application changes.

Summary of the Invention

The present invention is a new database pre and post processor that provides effective and comprehensive protection of a database and the data stored in it from damage, theft, or any other malicious activity, that provides flexible and cost effective ways to implement security and application changes, and that further provides improved overall system performance, by transparently off-loading functionality from the database or from the application server.

The general purpose of the present invention, which will be described subsequently in greater detail, is to provide a new database rule enforcement , change implementation and performance pre and post processor that has many of the advantages of conventional database mechanisms mentioned heretofore and many novel features that result in a new database pre and post processor which is not anticipated, rendered obvious, suggested, or even implied by and of the prior art database mechanisms, either alone or in any combination thereof.

To attain this, the present invention generally comprises an electronic computing device or another apparatus designed or able to operate designated program code in order to enforce certain rules, requirements and methods on incoming queries or outgoing data by a variety of methods, policies and strategies, change methods and policies and apply new ones . This apparatus and these methods operate transparent to the database and the database-client and change the content and the structure of the database access without breaking the connection between the database and the database-client. Such device may be connected as a monitor or proxy to the communication line or connection between the Database Client and the Database Server (Fig. 2) or be connected in-line between the Database Client and the Database Server in stealth mode (Fig. 1), so as not to be discovered or noticed by any of the connected devices or apparatuses, not having a designated network address, not being referred to by either server or any other device, and applying all security measures in such a manner not to reveal its existence (Fig. 1) or as hardware embedded in the database-client or as hardware embedded in the database or as a software module embedded in the database-client or as a software module embedded in the database .

There has thus been outlined, rather broadly and in non-descriptive fashion, the more important features of the invention in order that the detailed description thereof may be better understood, and in order that the present contribution to the art may be better appreciated. There are additional features of the invention that will be described hereinafter.

In this respect, before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not limited in its application to the details of construction and development and to the arrangements of components and program methods set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced and carried out in various ways. Also, it is to be understood that the phraseology and terminology employed herein are for the purpose of the description and should not be regarded as limiting.

A primary object of the present invention is to protect data stored in databases, preventing unauthorized or malicious access or manipulation, overcoming the shortcomings of the current solutions.

One object of the present invention is to provide aforementioned protection through a database security pre and post processor, installed with user or manufacturer designated rules for the protection of database, database systems and the data in them, and the prevention of data leaks.

Another object of the present invention is to restrict queries to deny or allow certain modes or styles of usage – certain relations between elements, certain operands, embedded database functions, prepared statements, wherein all may be made mandatory or disallowed.

Another object of the present invention is to prevent data leaks by the inspection of outgoing replies to queries and the post-factum denial of access by the reply – its size, length, the nature of its content or any other parameters defined by specific applications of the invention.

Another object of the present invention is to offload the Database Server by importing its structure, usage rules, security rules etc. and enforcing them itself.

Another object of the present invention is to prevent access to data by enforcing certain application or user specific rules relating to a certain application or user profile, or by way of comparing application or user database access to their respective history and detecting irregularities, anomalies, deviations, exceeding of certain quotas etc.

Another object of the present invention is to prevent the leakage of sensitive data where the database access is made in a development, testing or quality assurance context.In this case the invention might choose to manipulate the results in such a way as not to expose sensitive or confidential data.

Another object of the present invention is to minimize the exposure of sensitive data that may appear in database labels or columns or row names, by creating an "external name" for use in database access, which the invention translates to their corresponding label, column, row or other element, so that the nature of the database may not be revealed.

Another object of the present invention is to provide an auto learning system, which will deduce security rules from the database structure, the relationship between the Database Client and the Database Server, the common and conventional mode of use in the environment etc.

Another object of the present invention is to be able to operate in such a manner as not to be detected by any device or apparatus in the environment or network, so as not to be exposed to malicious activity directed at it, among other reasons.

Another object of the present invention is to supply an apparatus to enable the adding of new restrictions, security measures or functionality to a legacy system without the need for any changes in the database system itself.

Another object of the present invention is to avoid the creation of a data traffic bottleneck by having the common database access pass through without inspection on the way to the database, and to reserve inspection of said access to the replies generated upon their exit from the database. The invention determines if access made is potentially harmful, and in these cases, it will be inspected at the entrance to the database, while non-potentially harmful access will only be dealt with after they achieve a result.. Such post factum dealing may be achieved by a dedicated device.

Another object of the present invention is to allow a security administrator in an organization to apply security policies to a database without the need to involve the database administrator or system programmers whose interests conflict with those of the security administrator.

Another object of the present invention is to execute rules on the line, including, but not limited to, enforcement of proper use, enforcement of syntax, enforcements of certain sets of commands and values, enforcement of certain methods of retrieving the data, enforcement of certain client locations, enforcement of certain dates and times, or different policies for certain locations, dates and times, enforcement of operations only to a certain data unit – a row, a cell, etc. etc.

Another object of the present invention is to prevent queries by a rule based system on the line, terminating the queries without them reaching the database, or enforcing the database to produce an erroneous result.

Another object of the present invention is to prevent malicious queries of the "SQL Injection" variety by way of resolving the query and revealing such queries where a condition will always be true. Furthermore, the invention will attempt to resolve all and any database access to reveal its true nature and implications.

Another object of the present invention is to prevent the leakage of data from the database which might make it vulnerable to attacks. Such data might be, but is not limited to, error messages or acknowledgements of reception, which may reveal database structure or method or otherwise make it vulnerable to attack or theft.

Another object of the present invention is to perform its manipulations, protection, monitoring and prevention in such a manner as not to interfere with the connection between the Database Client and the Database Server, not to break the connection between the two and not to generate any errors or time outs that were not meant as a measure of security. On the whole, the invention is made not to hinder or

break the connection between the Application Server and the Database Server in its effort to protect the database.

Other objects and advantages of the present invention will become obvious to the reader and it is intended that these objects and advantages are within the scope of the present invention.

To the accomplishment of the above and related objects, this invention may be embodied in the form illustrated in the accompanied drawings and the following detailed explanation, attention being called to the fact, however, that the drawings are illustrative only, and the explanation is not an implementation in and of itself, and that changes may be made in the specific construction and design illustrated and described, which may still serve to the function and within the scope of the present invention.

Brief Description of the Drawings

Various other objects, features, implementations and attendant advantages of the present invention will become fully appreciated as the same becomes better understood when considered in conjunction with the accompanied drawings, in which like reference characters designate the same or similar parts or devices throughout the several views and different aspects of the operation, and wherein:

Fig. 1 is the preprocessor connected in stealth mode, in-line between the Database Client and the Database Server.

Fig. 2 is the preprocessor connected in a proxy/monitor configuration, either tapping the connection and resetting in need, or serving as a recognized proxy between the Database Client and the Database Server.

Fig. 3 is the illustration of the method where a malicious query is manipulated by the device to produce a defused one to reach the database, the whole process unnoticed by all parties.

Fig. 4 is the illustration of the method where the device manipulates outgoing content in order to deny illegal database usage,

Fig. 5 is the illustration of the method where the device recognizes and answers a malicious query, with it never arriving at the database.

Fig. 6 is the illustration of the method where the device lets common queries pass through not inspected, marking them for inspection on the way out, and acting if the yielded information is illegal.

Fig. 7 is the illustration of the method where the device delays the query on its way to the Database Server, requiring authorization from another system with which it consults.

Fig. 8 is the illustration of the proper separation of authorities between the System Administrator and the Security Administrator facilitated by the device.

Fig. 9 is the internal structure of the preprocessor handling packets on the path from the Database client to the Database Server

Fig. 10 is the internal structure of the preprocessor handling packets on the path from Database Server to the Database client

Fig. 11 is the illustration of the method where the device applies data hiding procedure

Fig. 12 is the illustration of the method where the device applies data scrambling procedure

Fig. 13 is the illustration of the method where the device applies database objects translations

Detailed Description of the Invention

The preferred embodiment of the present invention is a computing system either specifically designed for the purpose of the invention or a general purpose device including some sort of electronic memory apparatus and preferably two or more network interfaces. The invention may also be implemented on a chip residing on the computer bus, and as a software module that runs on the database client or server. The aforementioned system will be installed with a computer program to employ several methods for the protection of databases, and adding new restrictions and functionalities to the database and database client connection, described hereunder in great detail.

The methods of this invention are designed with several different fields of operation in mind:

- Methods for protection from incoming queries (Fig. 3).
- Methods for prevention of data leaks in outgoing replies (Fig. 4).
- Methods for improving database performance and secured practices (Fig. 6).
- Method for implementing new restrictions and functionalities to the database and database client connection (Fig. 7, 11, 12).
- Method for modification and manipulation of the traffic between the database and the database- client while preserving the database and database-client connection (Fig. 11,12).
- Method for separation of responsibilities between the security administrators and the DBAs and system programmers (Fig. 8).

All of these methods are applied by a designated, dynamic and customizable set of rules, These rules can be set by the designer, operator or client by a secured and authenticated interface. The management interface is set in a different network than that of the interfaces to the Database Server and to the Database Client, as to not expose it to dangers of manipulation by unauthorized entities. Further, proper and advanced security methods will be applied to protect the management interface. Furthermore, the connection to the method or appliance management agent, in certain configurations, can be via the in-line in-band management information to a predefined secured management daemon belonging to the application. All incoming queries and outgoing replies are subject to inspection according to the designated rules, while other rules can be made to affect database behavior and security practices.

The application of these rules by the invention is done in a manner not to interfere with the regular operation of either the application or the database and not to break or reset any connections between the Database Client and the Database Server (Fig.3). The invention controls access (Fig.3-307) to database objects by checking structure, syntax, content and semantics of incoming SQL queries and validity of outgoing data aiming to have malicious or illegal queries never reach the database and to screen illegal answers without causing errors (Fig.5). This is achieved by locally terminating the malicious or illegal query, as seen in Fig. 5. The incoming query (Fig.5-505) is manipulated by the appliance (Fig. 5-502), and returns the manipulated result (Fig 5-506). This result is in a form that won't break the connection. It may contain generic database error message or any other information in order to preserve the connection. The invention deals with all aspects of maintaining and operating a connection – generating error messages for illegal queries before the database, or otherwise forcing an error message from the database by manipulating the query, deleting illegal content from replies or replacing replies with blank content and any other method relating to the non-interruption or regular activity by the security measures.

The herein solution takes all necessary measures to the issue of preserving the connection and the correct and legal data frame format. When manipulation of either incoming database access or outgoing data is done, the data frame is actually re-built, creating new, legal headers, length fields, counters, CRC values, etc. so that the new frame will not be erroneous and will pass to the database or application as a legitimate query or reply, as shown in Fig. 3. The packet (Fig.3-305) sent out from the database client (Fig.3-301) is manipulated by the appliance (Fig.3-302) and forwarded with full compliance to the session rules and the application rules. Also, as shown in Fig. 4. the database (Fig4.-403) sends its results. The results are inspected by the appliance (Fig.4-402) and forwarded to the database client (Fig.4-405), fully complying with the outgoing data rules and the session rules. Much care is also given to have the appliance context-aware when handling outgoing data, connecting it with the database access that produced it, in order to produce a legitimate database answer. The manipulated database access data frame reaching the database server must be as legitimate, correct and legal as any standard query made by the application, and the manipulated reply data frame reaching the application server must also be as legitimate, correct and legal as any standard reply made by the database. Special care must also be attended to preserving any special rules, regulations, standard or formats unique to the application and also industry standard ones. In Fig. 11, we demonstrate how data that leaves the database (Fig. 11-1101) is converted, lines are dropped from the outgoing data (Fig. 11-1104), and reformatted by the appliance (Fig.11-1102) to a table (Fig.11-1105), fully complying with the expected format of the client (Fig.11-1103), while applying, in this specific example, the data-hiding restrictions. All in all, the manipulated data frames produced by the appliance must appear no different than regular data frames in the system in which it is installed.

By its nature as a rule-based, customizable appliance, the invention can be used to protect and employ other rules and methods that are not included in this description, yet will become obvious and can be deduced from its other methods by many a person skilled in the art. The description therein takes in consideration and under its protection all those applications and methods that are possible within its rule-based system.

Incoming database access is inspected by the system, parsed into individual objects and separate operands, and resolved according to the individual object values and the connecting operands. The database access is then subjected to the system rule-based inspection. The method for rule-based inspection of incoming database access can be defined to sense and scrutinize database access by a variety of outlooks and degrees:

- Syntax

  a rule can be made to allow or disallow only certain query, stalled procedure and RPC syntaxes. The use of prepared statements may be made mandatory, always true objects may be purged from queries, use of constants may be disallowed or made mandatory, certain operands or characters may be made mandatory or disallowed, and any other limitation or enforcement of syntax may be applied to the database access.

- Semantics

  Usage may be limited to only certain database objects: databases, tables, columns rows and others or any combination of which.

- Context

  Usage may be limited according to user-specific, group-specific or system-wide context – in light of previous activity, recent activity, usual activity, or any other nature of query context.

- Value

  Values used may be restricted to certain ranges to prevent exploitation of database faults or data leakage. Other limitations or enforcements of values may be applied to the database access.

- Procedure

  Database access may be confined to certain methods. For example, a bank teller may only be able to get information by ID, and not by name. Other applications of this are also possible, either user or group specific or system-wide. Enforcing specific and designed database access procedure can be done by a variety of parameters (certain fields, certain tables, certain operands or operators) and groupings (users, time of day, usage history).

- Aggregation-Restriction

Some data might only be accessed row by row, not having its sum returned, while other data might only have its sum returned, not allowing the access of specific rows. This obviously depends on the nature of the data. Group operations of data (sum, average, etc.) may also be allowed or disallowed. For some types of data, whole table queries may not be allowed. Other methods for protection pertaining to the grouping of data may also be applied.

- Origin-Restriction

The access of data might be limited or excluded from certain geographical locations, IP addresses, networks, computers, users or any other feasible identification of origin. Security policy might also depend on location, user, etc.

- Time&Frequency

Database access may be limited to certain times or day, or to certain idle times between database accesses. Security policy might also change on frequent database accesses or on certain times.

- Importing&Consultation(Fig.7)

Database pre-and post-processor might withhold inspection of database access awaiting confirmation from an outside source – Biometrical devices, user organizational hierarchy, or any other outside source, as shown in the example in Fig.7. A query that is generated by the database client (Fig. 7-704) is sent (arrow #1) towards the database, is intercepted by the appliance (Fig.7-703), and suspended. A consulting query is sent by the appliance (Fig.7-703) to a consulting server or system (Fig.7-702). When the results are returned (arrow#3) the query is forwarded to the database or locally terminated, according to the results received from the consulting system.

- Pre- and post-processor might also import the security policy, schemes and measures from the database server itself, implementing them in the pre- and post-processor.

Other methods of protection from, and the scrutinizing of, incoming queries, or any database access are to be made obvious or reached by deduction by any person skilled in the art. The intent of the invention is to apply those methods as well and these methods are to be implemented in the invention and encapsulated within this description.

Upon the returning of an answer, it will, in turn, will also be inspected by the pre- and post-processor, subject to another set of rules (Fig.4). While all outgoing answers will be inspected, some queries of common types (e.g. SELECT queries) may not go through inspection on the way to the database, only inspected when the answer is returned (Fig.6).

Most database queries are used for retrieving information. In order to avoid performance degradation, all the retrieve queries are immediately forwarded to the database (fig.6-605,606), and processed by the appliance (Fig. 6-602) parallel to the database (Fig.6-603). The results (Fig.6-607) are inspected by the appliance, and the appropriate answer (Fig.6-609) is sent to the database client (Fig.6-601). We may use this technique also with commit queries, but in this case, additional actions need to be taken, in case of malicious or improper queries, in order to roll back the database to its prior state, as shown in Fig6-608. This categorization of common types may or may not be implemented as a separate hardware device, and may or may not be a customizable element of the pre- and post-processor.

Database answers (any data leaving the database, as a result of database access) may be inspected by a variety of outlooks and degrees:

- Table

  Table answers may be restricted in a number of ways: No complete database table may be returned, table size may be limited, the number of rows or columns may be restricted, other limitations on table answers may also be implemented, as shown in Fig 11.

- Column/Row

  Certain columns may not be allowed to be returned, and certain column data might be subject to manipulation, to protect sensitive data for QA purposes. In case of sensitive data in column, row or table name, database object translation mechanism may be employed, where a database element may have an "outer name" and an "inner name", as shown in Fig.13. Other limitations or manipulations of column or row answers may also be applied.

- Value

  Certain ranges may be applied to answers, as to protect specific, sensitive objects. Data values might also be (randomly or not) manipulated to protect sensitive data used for QA purposes.

- Message

  The pre- and post-processor may prevent outgoing database error messages or other database messages, as those may jeopardize its safety and reveal its inner workings mechanisms and security policy.

- Context

  Data returned may be limited by user or group specific or system-wide context, based on history, usual usage or best practices. This may be used to prevent the disguising of data theft as common queries, or a whole table being stolen row-by-row.

- Taboo

  Certain values, rows, columns or tables may not be returned, no matter what query was

made. The method provides for a definition and prevention of the leaks of this "taboo" data, as shown in Fig.12.

Other methods of scrutinizing, screening and manipulating outgoing data are to be made obvious or reached by deduction by any person skilled in the art. The intent of the invention is to apply those methods as well and these methods are to be implemented in the invention and encapsulated within this description.

The invented apparatus and methods also allow for several other benefits more general in nature, not directly related to the inspection of incoming database access and outgoing data:

- Any Database or Syntactical Object Translation

  To protect sensitive data in database elements names and structure, the invention supplies a method to use an "outer name" for use in queries and answers, and an "inner name" for use within the database. The "outer name" may or may not have any logical significance, and will serve to keep the sensitive "inner name" hidden, as shown in Fig13. The "inner name" could potentially point to a different database, system or manifestation. The translation system could further be used to facilitate migration or topology changes, and other benefits arising from re-directing and translating of database and syntactical objects.

- Legacy-Systems-Enhancements

  Serving as an external mechanism, the database pre- and post-processor may be used to facilitate new security measures and policies and apply new features and requirements in legacy systems or in systems otherwise sensitive to changes, with no need for any modification in the system itself. This may be also used for added functionality, as described in the "Importing&Consultation" method, or for improving performance, as described in the following "Database Offloading" method.

- Database-Offloading

  By taking the security responsibilities from the database server to the database pre- and post-processor, it serves to offload the database server, improving its performance. The invention also supplies a method to import the security policy of the database, taking more of its load. The appliance may offload other functionalities from the database in order to improve the database performance.

- Stealth-Mode-Operation

  The invention may serve in one of three functions – as a monitor device (Fig.2), tapping in on the connection, logging all activity and sending a TCP reset in case of malicious activity, as a known and active proxy device, standing as a mediator in all transactions, or

as an in-line stealth mode device (Fig.1), having no network or hardware address, undetectable to any inner or outer network device, client or server. In such a mode the device's existence is unknown to any attacker, and it is not exposed to any form of attack.

- Separation-of-Authorities(Fig.8)

By supplying a tool for enforcing security policies on the database, the invention supplies a method for the Security Administrator of an organization to deal with database security without having to go into the Database Administrator realm of responsibilities and conflicting interests. The same is true in the case of the system programmers.

An example of packet handling is described in Fig. 9 and Fig.10. Please note that Fig. 9 and Fig.10 represent only examples of a preferred embodiment of the herein solution's dealing with the packet path from database client to the database. They are not intended to limit the invention to one preferred embodiment. To the contrary, they are intended to cover alternatives, modifications and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

The following are the elements that describe the process of the packet path from the database client to the database. The movement in the flow can be ascertained by inspecting the arrow directions in the diagram.

Fig 9 - 901. The element of the system collecting from the network media packets sent by the Database Client

Fig 9 - 902. The element of the system parsing collected packets and dispatching said packets to other elements of the system for further processing

Fig 9 - 903. The element of the system parsing SQL language statements carried by the collected packets

Fig 9 - 904. The element of the system taking decisions about operations performed on the said packets and preparing data needed by these operations. The examples of operations are:

a. Forward packet to the Database
b. Modify packet
c. Drop packet
d. Replace packet with one or more different packets and send these packets to the Database Server
e. Send one or more packets to the Database Client

Fig 9 - 905. The element of the system holding and maintaining set of rules used by the Decision Engine and data used for operations mentioned in description of element 94.

Fig 9 - 906. The element of the system rebuilding packet which may be safely forwarded to the Database Server without breaking connection between Database Client and the Database Server

Fig 9 - 907. The element of the system injecting into the network media packets destined to the Database Server or Database Client

The following are the elements that describe the process of the packet path from the database to the database client. The movement in the flow can be ascertained by inspecting the arrow directions in the diagram.

Fig 10 - 1001. The element of the system collecting from the network media packets sent by the Database Server

Fig 10 - 1002. The element of the system parsing collected packets and dispatching said packets to other elements of the system for further processing

Fig 10 - 1003. The element of the system interpreting Database response to the Database Client carried by the said packets

Fig 10 - 1004. The element of the system taking decisions about operations performed on the said packets and preparing data needed by these operations. The examples of operations are:

a. Forward packet to the Database
b. Modify packet
c. Drop packet
d. Replace packet with one or more different packets and send these packets to the Database Client
e. Send one or more packets to the Database Server

Fig 10 - 1005. The element of the system holding and maintaining set of rules used by the Decision Engine and data used for operations mentioned in description of element 104.

Fig 10 - 1006. The element of the system rebuilding packet which may be safely forwarded to the Database Client without breaking connection between Database Server and the Database Client

Fig 10 - 1007. The element of the system injecting into the network media packets destined to the Database Client or Database Server

As to a further discussion of the manner of usage and operation of the present invention, the same should be apparent from the above description. Accordingly, no further discussion relating to the manner of usage and operation will be provided.

With respect to the above description then, it is to be realized that the optimum hardware and program code implementation of the invention, including methods for improving performance, adhering to industry standards, interfacing with currently available database servers, supplying with a management console, including variations in implementation and use, are deemed readily apparent and obvious to one skilled in the art, and all equivalent implementations to those illustrated in the drawings and described in the specification are intended to be encompassed by the present invention.

References Cited

U.S. Patent Documents

5,606,668	February, 1997	Shwed
5,835,726	November, 1998	Shwed , et al.
6,769,074	July, 2004	Vaitzblit
6,823,460	November, 2004	*Hollander* , et al.
6,804,778	October, 2004	Levi , et al.
7,010,700	March, 2006	Foss , et al.
7,010,524	March, 2006	Galindo-Legaria , et al.
6,999,977	February, 2006	Norcott , et al.
6,999,956	February, 2006	Mullins
6,996,589	February, 2006	Jayaram , et al.
5,958,015	September, 1999	Dascalu
6,434,618	August, 2002	*Cohen* , et al.
6,625,650	September, 2003	Stelliga
6,751,677	June, 2004	*Ilnicki* , et al.
6,735,594	May, 2004	Zimowski , et al.
6,636,894	October, 2003	Short , et al.

U.S. Patent Application

20050097149	May, 2005	Vaitzblit, Lev ; et al.
20050120054	June, 2005	Shulman, Amichai ; et al.
20060059154	March, 2006	Raab; Moshe
20060041555	February, 2006	Blessin; Stephen W. ; et al.
20040250098	December, 2004	Licis, Norman D.
20050015510	January, 2005	Rhee, Jai-hyoung
20050237955	October, 2005	Shapira, Yair ; et al.
20050044422	February, 2005	Cantrell, Craig ; et al.
20050028013	February, 2005	Cantrell, Craig ; et al.
20030072265	April, 2003	Seaborne, Andrew Franklin ; et al
20020059435	May, 2002	Border, John ; et al.

Claims

What is claimed is:

1. A computer system a method or apparatus that provides for the off-loading of database and database system security, administration, management and performance, wherein such database and database system security, administration, management and performance activities operate independent of said database or said database system or any database application, and which uses stealth data manipulation provided by a spoofing protocol, which said stealth data manipulation occurs during pre and post-processing of traffic between or among two or more computing devices, all without interrupting data flow, degrading data flow performance or changing traffic integrity.
2. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents unauthorized or malicious access or manipulation of said database or said database system, comprising a database tool as define herein-

   a computing means having installed therein a database security pre and post processor means;

   an operating system means for operating said computing means and database security preprocessor means;

   an application means running on said pre and post processor means wherein said application means analyzes and handles incoming database access or other data, in accordance with user or manufacturer designated rules;

   a second application means running on said pre and post processor means wherein said application means analyzes and handles outgoing database answer or other data, in accordance with user or manufacturer designated rules;
3. A computer system, a method or apparatus according to Claim 1, installed with user or manufacturer designated rules to provide protection of a database or database system, or the data contained in said database or said database system, and to further prevent data leaks from said database or database system, comprising a database tool
4. A computer system, a method or apparatus according to Claim 1, wherein said computer system restricts queries to deny or allow certain modes or styles of usage, including certain relations between element, certain operands, embedded database functions, prepared statements, all of which may be made mandatory or disallowed by said computer system in connection with said database or said database system, comprising a database tool
5. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents data leaks from said database or said database system, comprising a database tool prevent data leaks by the inspection of outgoing replies to queries and the post-factum denial of access by the reply – its size, length, the nature of its content or any other parameters defined by specific applications

6. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents access to data and enforces certain application or user specific rules relating to a certain application or user profile, or compares application or user queries to their respective query history and detects irregularities, anomalies, deviations, that exceed certain quotas etc., comprising a database tool
7. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents unauthorized or malicious manipulation of said database, or said database system, comprising a database tool
8. A computer system, a method or apparatus according to Claim 1, wherein said computer system minimizes the exposure of sensitive data that may appear in database labels or column or row names, by creating an "external name" for use in the query, which said computer system would then translate to their corresponding label, column or row, so that the nature of said database or database system may not be revealed, comprising a database tool
9. A computer system, a method or apparatus according to Claim 1, having installed therein an auto learning system means that deduces security rules from said database or said database system structure, the relationship between the Database Client and the Database Server, the common and conventional mode of use in said database or database system environment, comprising a database tool
10. A computer system, a method or apparatus according to Claim 1, having installed therein a means to enable the adding of new restrictions, security measures or functionality to a legacy database system or to a legacy application or applications running within said legacy database system, without the need for any changes in said legacy database or said legacy database system itself, comprising a database tool
11. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents the creation of a traffic bottleneck through having common queries pass through said computer system without inspection on the way to said database or database system, and to reserve inspection of said queries to the way out of said database or database system, comprising a database tool
12. A computer system, a method or apparatus according to Claim 1, wherein said computer system allows a security administrator to apply security policies to a said database or said database system without the need to approach said database administrator or said database system administrator, comprising a database tool
13. A computer system, a method or apparatus according to Claim 1, wherein said computer system, in order to execute rules on the line, enforces proper use, enforces syntax, enforces of certain sets of commands and values, enforces certain methods of retrieving the data, enforces certain client locations, enforces certain dates and times, or different policies for certain locations, dates and

times, and further enforces operations only to a certain data unit such as, without limitation, a row, a cell, comprising a database tool

14. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents queries by a rule-based system on the line, terminating the queries without them reaching the database, or enforcing the database to produce an erroneous result, comprising a database tool
15. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents malicious queries, including without limitation the "SQL Injection" variety, by way of resolving the query and revealing such queries where a condition will always be true, comprising a database tool
16. A computer system, a method or apparatus according to Claim 1, wherein said computer system prevents the leakage of data from the database which might make it vulnerable to attacks, such data might be, but is not limited to, error messages or acknowledgements of reception, which may reveal database structure or method or otherwise make it vulnerable to attack or theft, comprising a database tool
17. A computer system, a method or apparatus according to Claim 1, where said computer system manipulates, protects, monitors, inspects and prevents in such a manner as to not interfere with the connection between the Database Client and the Database Server, to not break the connection between the two and not to generate any errors or time outs that were not meant as a measure of security and generally to not hinder or break the connection between the Application Server and the Database Server in said computer system's efforts to protect the database. comprising a database tool
18. A computer system, a method or apparatus according to Claim 1, to provide protection of a second computer system or a plurality of other computer systems, where said computer system prevents the leakage of sensitive data from said second computer system or plurality of computer systems, where such system could manipulate a query made in a development, testing or quality assurance context without exposing sensitive or confidential data contained in said second computer system or plurality of computer systems, comprising a database tool
19. The method or apparatus according to claim 1, further specifying the position of an apparatus as a communication device just before the database server.
20. The method or apparatus according to claim 1, further specifying the position of an apparatus as a communication device just after an application server.
21. The method or apparatus according to claim 1, further specifying the position of an apparatus as a co-processor installed on the database server bus.
22. The method or apparatus according to claim 1, further specifying the use of such method as one installed on the database server itself.

23. The method or apparatus according to claim 1, further specifying the use of such method as one installed on the application server itself. Acting when accessing either an internal or external database.
24. The method or apparatus according to claims 22 or 23, wherein the traffic is intercepted in the system call level.
25. The method or apparatus according to claims 22 or 23, wherein the traffic is intercepted in the communications stack level.
26. The method or apparatus according to claim 1, further comprising a method of having the method or apparatus act in a fashion invisible to other network or software elements.
27. The method or apparatus according to claim 1, wherein said processing is used to inspect and manipulate database access.
28. The method or apparatus according to claim 27, wherein said database access manipulation may imply changing the query, stalled procedure or remote procedure call, adding or removing elements, function or objects, or changing, adding or removing any part of the query, stalled procedure or remote procedure call.
29. The method or apparatus according to claim 27, wherein said database access manipulation is done without breaking the connection between the database client and the database server.
30. The method or apparatus according to claim 27, wherein said manipulation is used to restrict database access mode, style or structure.
31. The method or apparatus according to claim 30, further using the restriction to demand or deny certain relations between elements, certain operands, use of certain database tables or objects, use of prepared statements or use of embedded database functions.
32. The method or apparatus according to claim 27, wherein database access is resolved to detect any malicious or harmful nature before reaching the database.
33. The method or apparatus according to claim 27, wherein processing is used to enforce proper use, syntax, certain sets of commands and values, certain methods of retrieving data, certain client locations, certain dates and times, or different policies for certain locations, dates and times, and allow operations only to a certain data unit.
34. The method or apparatus according to claim 33, further comprising a method to allow the creation of virtual tables and data units destined for use by certain applications or users, limiting the scope of data visible to the application or user, independent of application or database.
35. The method or apparatus according to claim 1, wherein said processing is used to inspect and manipulate replies made by the database.
36. The method or apparatus according to claim 1, wherein said processing is used to inspect and manipulate database traffic in both directions.

37. The method or apparatus according to claim 35, wherein said inspection and manipulation is used to prevent data leakage.
38. The method or apparatus according to claim 37, wherein said prevention is performed by parameters defined in the specific application, these may include size, length or nature of the content.
39. The method or apparatus according to claims 27, 35 and 36, further comprising a set of rules according to which the manipulation is done.
40. The method or apparatus according to claim 39, further comprising a method for setting, changing and updating the rules.
41. The method or apparatus according to claim 36, where rules are enforced in respect to the user or application accessing the database, based on defined profiles or by comparing to user or application history and quotas, detecting deviations and irregularities.
42. The method or apparatus according to claim 35, wherein manipulation is used to mask, randomly or systematically change data exiting the database according to parameters set in the method.
43. The method or apparatus according to claim 42, wherein said parameters include the nature of the environment or system where the information is destined to arrive at.
44. The method or apparatus according to claim 35, wherein said manipulation is used to prevent the database from inclosing information on its structure, in error messages, acknowledgements, or any other reply created by it.
45. The method or apparatus according to claim 36, wherein said manipulation is used to translate names referring to database objects, functions or tables.
46. The method or apparatus according to claim 45, wherein said translation is performed on the basis of an internal or external conversion table.
47. The method or apparatus according to claim 45, wherein said translation is used to mask sensitive data, hide database structure, redirect from one database or table to another to facilitate database changes and migration, changes in topology or to provide load balancing services.
48. The method or apparatus according to claim 36, wherein said manipulation is done in a store-and-forward fashion.
49. The method or apparatus according to claim 48, wherein the method or apparatus itself fabricates the forwarded content.
50. The method or apparatus according to claim 36, wherein common database access is passed not inspected, flagging their reply for inspection and post-factum denial and rolling back the database to the previous state. The process is facilitated within the method or apparatus – caching the query, invoking the denial and performing the roll back.
51. The method or apparatus according to claim 50, where a dedicated apparatus performs the inspection.

52. The method or apparatus according to claim 36, wherein queries might be terminated on the line or within the method, never reaching the database.
53. The method or apparatus according to claim 52, wherein such termination is followed by a method to force the database to evoke an error message.
54. The method or apparatus according to claim 53, wherein the pre and post processor fabricates the error message.
55. The method or apparatus according to claim 39, further comprising a method for the automatic creation of the rule set by an automated learning mechanism which will deduce rules from the database structure, the relationship between the database client and the database server, the common and conventional mode of use in the environment and other structural and behavioral patterns.
56. The method or apparatus according to claim 39, further comprising a method for importing the rule set from the database server's usage and security rules for implementation in the method or apparatus itself.
57. The method or apparatus according to claim 39, further comprising a method for importing or deducing the rule set from an existing identity management, role, rule, directory or access control systems.
58. The method or apparatus according to claim 36, further comprising a method for reserving judgment on a query or reply until receiving further information or authentication from another system.
59. The method or apparatus according to claim 58, wherein said information is solicited by the method or apparatus
60. The method or apparatus according to claim 1, wherein the method or apparatus are used to provide additional functionality, security related or otherwise, to an existing configuration.
61. The method or apparatus according to claim 1, wherein the pre and post processing abilities are used to offload activity from the database server.
62. The method or apparatus according to claim 1, wherein the pre and post processing abilities are used to optimize database access.
63. A method for separating the responsibilities of the database administrator and the security administrator. This separation is facilitated by the creation of a management device for the application. This device may enable the definition of the rule set in claim 39, the virtual tables in claim 34, the masking mechanism in claim 42, the translation table in claim 46 or other definitions relating to the nature and functionality of the method or apparatus.
64. The method to claim 63, wherein the separation is used to facilitate a double-inspection method for changes in the database.

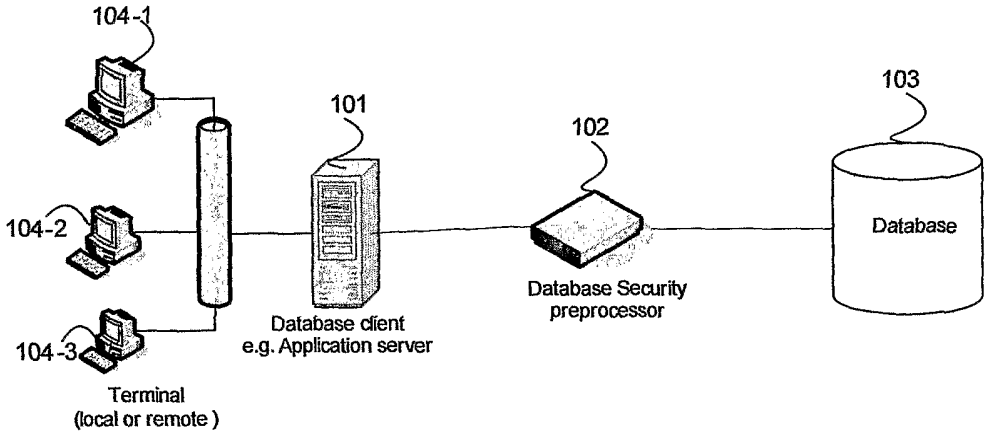

Fig. 1
transparent configuration

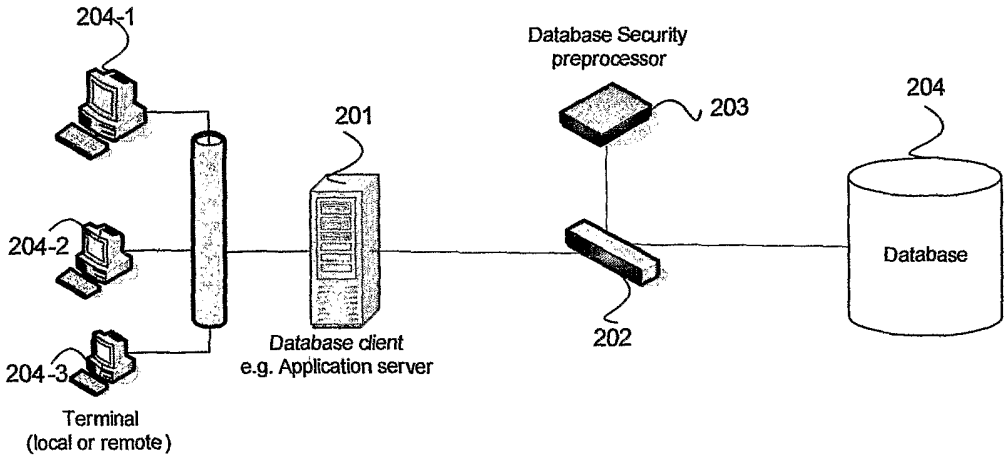

Fig. 2
Proxy and Monitor
configuration

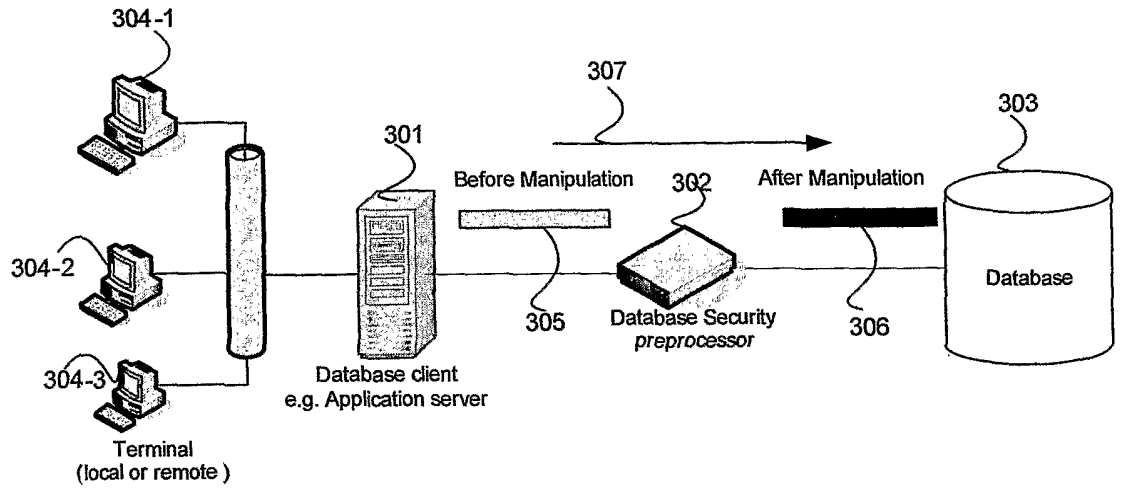

Fig. 3
Manipulate frames from Database clients(e.g. application servers) to database while preserving database client connections

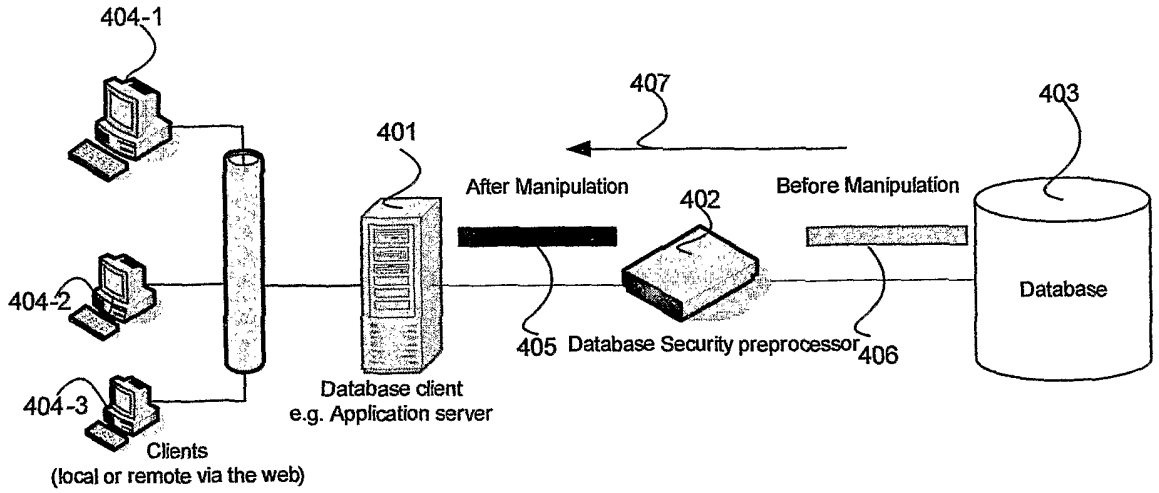

Fig. 4
Manipulate frames from database to database clients(application server) while preserving database client connections

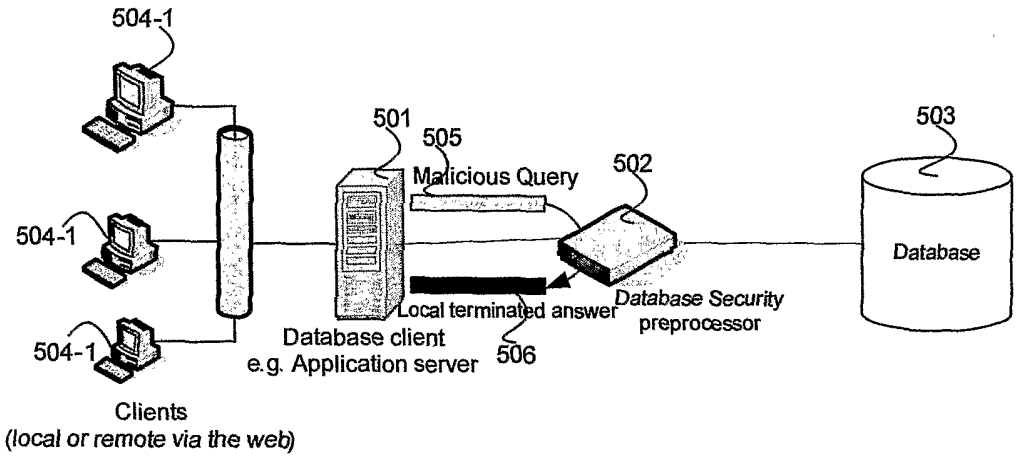

Fig. 5
Local termination while preserving database client(e.g. application database) connections

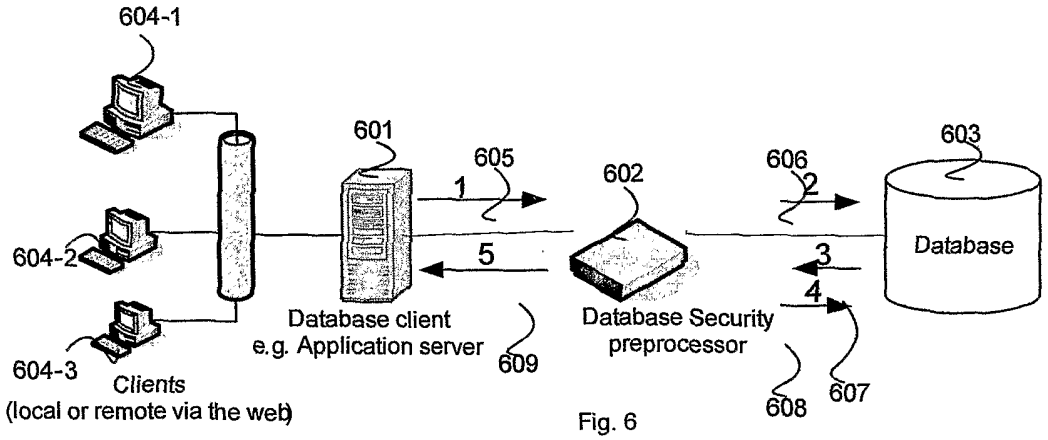

Fig. 6
Common query speedup by query pass through and inspection of the reply. Roll-back to database in case of illegitimate reply and manipulate the result returned to the Database client(e.g. application server)

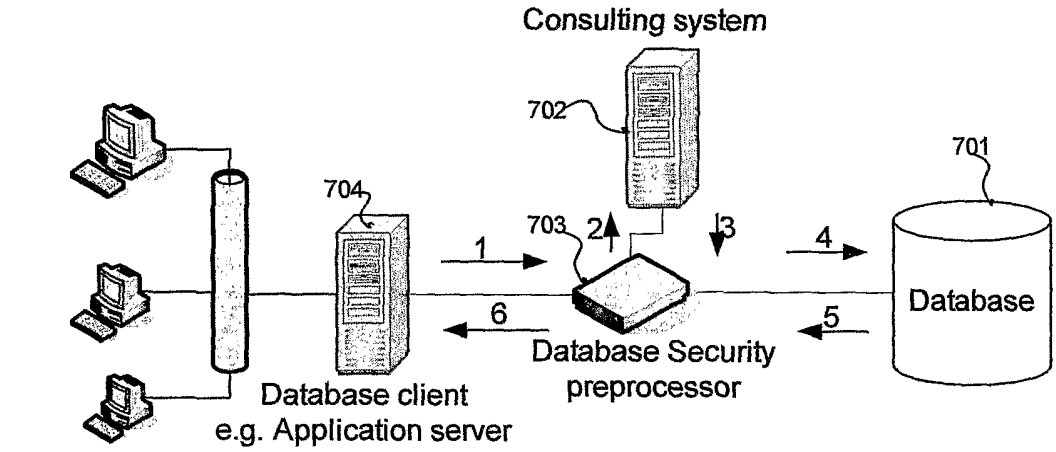

Fig. 7
Query delay at the database security preprocessor and consulting with another application, system or service

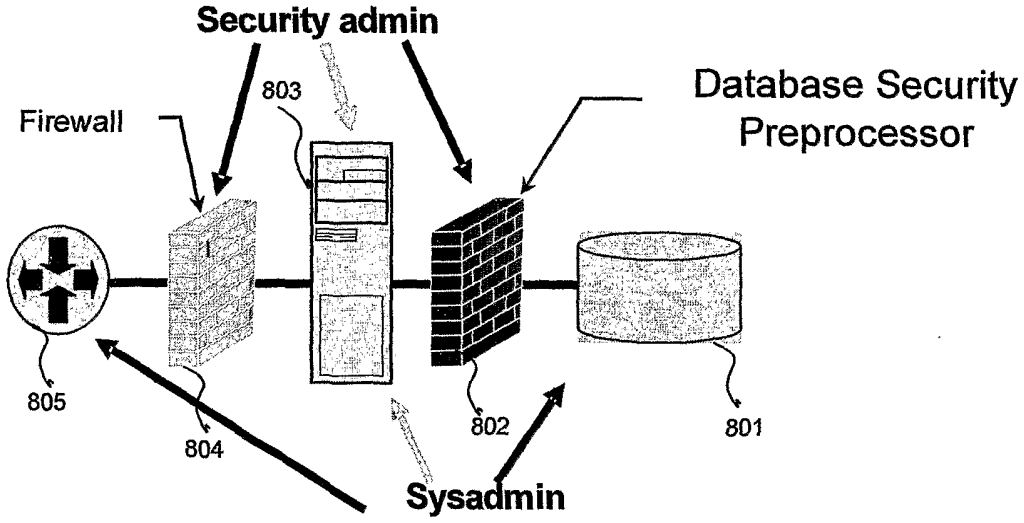

Fig. 8
Separation of Authorities

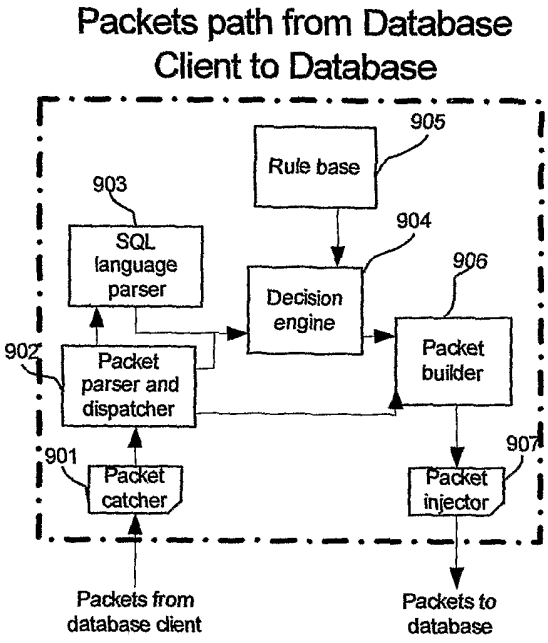
Packets path from Database Client to Database
905
Rule base
903
SQL language parser
904
Decision engine
906
Packet builder
902
Packet parser and dispatcher
901
Packet catcher
907
Packet injector
Packets from database client
Packets to database

Fig. 9

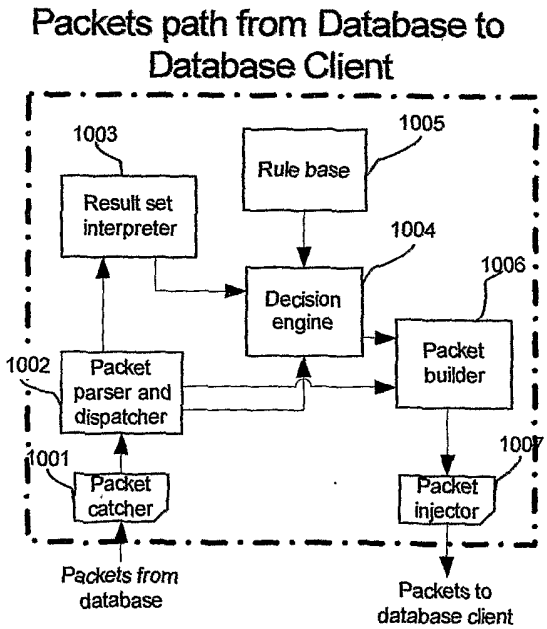
Packets path from Database to Database Client
1003
Result set interpreter
1005
Rule base
1004
Decision engine
1006
Packet builder
1002
Packet parser and dispatcher
1001
Packet catcher
1007
Packet injector
Packets from database
Packets to database client

Fig. 10

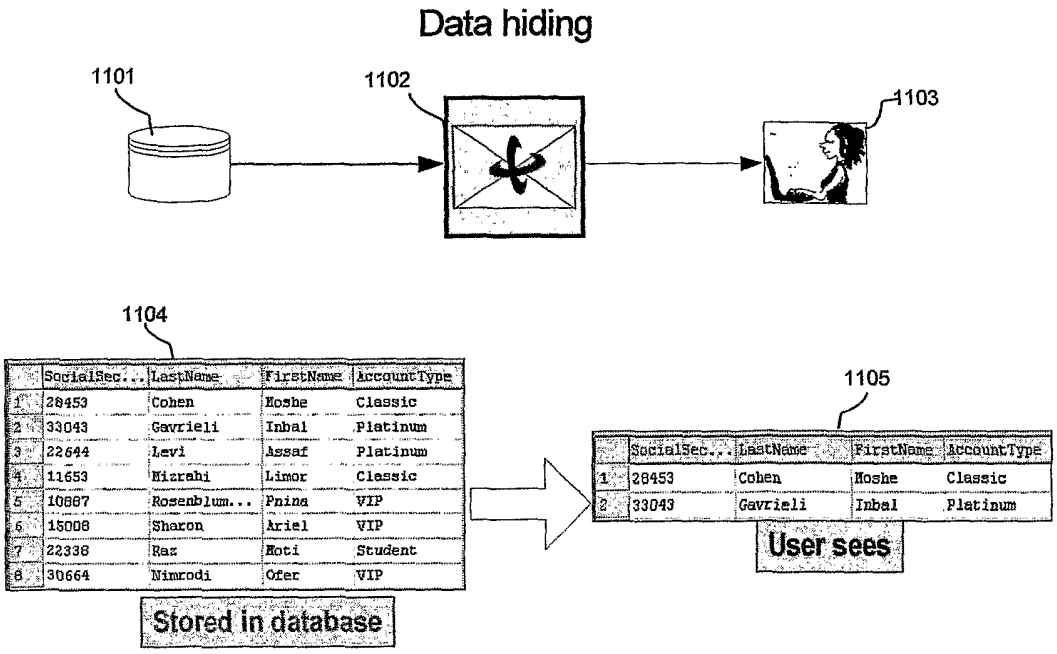
Data hiding
1101
1102
1103
1104
1105
SocialSec... LastName FirstName AccountType
1 28453 Cohen Moshe Classic
2 33043 Gavrieli Inbal Platinum
3 22644 Levi Assaf Platinum
4 11653 Mizrahi Limor Classic
5 10887 Rosenblum... Pnina VIP
6 15008 Sharon Ariel VIP
7 22338 Raz Moti Student
8 30664 Nimrodi Ofer VIP
Stored in database
User sees

Fig. 11

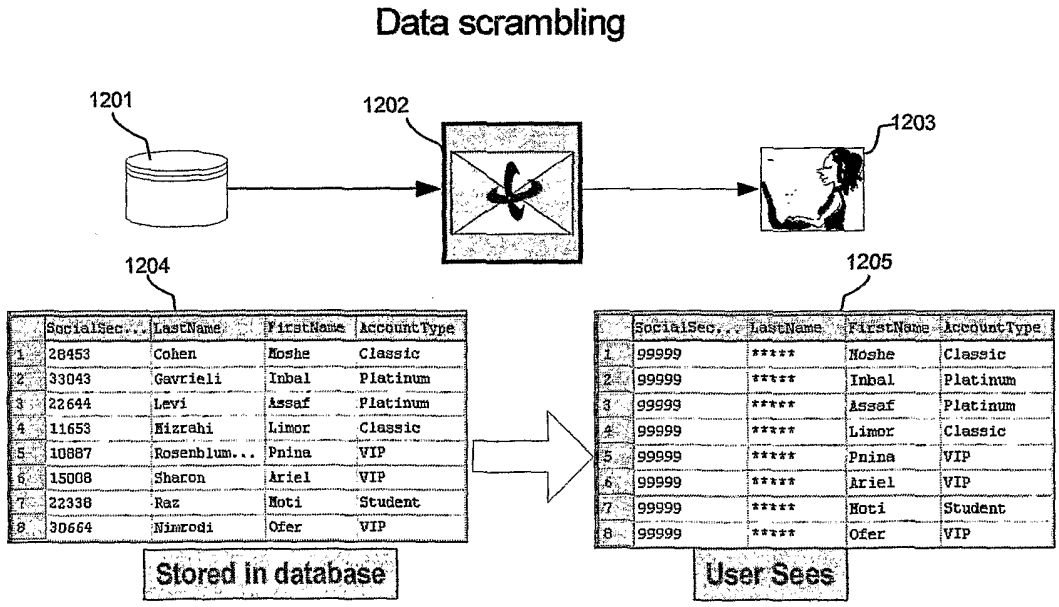
Data scrambling
1201
1202
1203
1204
1205
SocialSec... LastName FirstName AccountType
1 28453 Cohen Moshe Classic
2 33043 Gavrieli Inbal Platinum
3 22644 Levi Assaf Platinum
4 11653 Mizrahi Limor Classic
5 10887 Rosenblum... Pnina VIP
6 15008 Sharon Ariel VIP
7 22338 Raz Moti Student
8 30664 Nimrodi Ofer VIP
Stored in database
1 99999 ***** Moshe Classic
2 99999 ***** Inbal Platinum
3 99999 ***** Assaf Platinum
4 99999 ***** Limor Classic
5 99999 ***** Pnina VIP
6 99999 ***** Ariel VIP
7 99999 ***** Moti Student
8 99999 ***** Ofer VIP
User Sees

Fig. 12

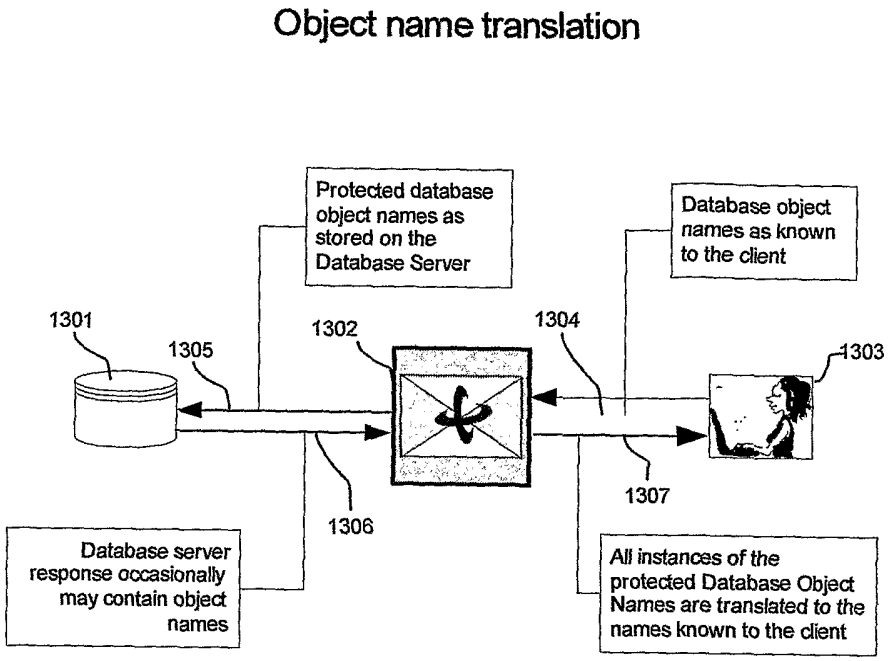
Object name translation
Protected database object names as stored on the Database Server
Database object names as known to the client
1301
1305
1302
1304
1303
1306
1307
Database server response occasionally may contain object names
All instances of the protected Database Object Names are translated to the names known to the client

Fig. 13