A Survey on Data Security in Data Warehousing

Issues, Challenges and Opportunities

Ricardo Jorge Santos
CISUC – DEI – FCT
University of Coimbra
3030-190 Coimbra, Portugal
lionsoftware.ricardo@gmail.com

Jorge Bernardino
CISUC – DEIS – ISEC
Polytechnic Institute of Coimbra
3030-290 Coimbra, Portugal
jorge@isec.pt

Marco Vieira
CISUC – DEI – FCT
University of Coimbra
3030-190 Coimbra, Portugal
mvieira@dei.uc.pt

Abstract—**Data Warehouses (DWs) are the enterprise's most valuable assets in what concerns critical business information, making them an appealing target for malicious inside and outside attackers. Given the volume of data and the nature of DW queries, most of the existing data security solutions for databases are inefficient, consuming too many resources and introducing too much overhead in query response time, or resulting in too many false positive alarms (i.e., incorrect detection of attacks) to be checked. In this paper, we present a survey on currently available data security techniques, focusing on specific issues and requirements concerning their use in data warehousing environments. We also point out challenges and opportunities for future research work in this field.**

Keywords: data security; data warehousing; data privacy; data confidentiality; data integrity; data availability; intrusion detection; encryption; data recovery.

I. Introduction

Data Warehouses (DWs) are mainly databases storing consolidated historical and current business data for decision support purposes. The DW reflects the measures and results of the business, as well as how and when it happens. Currently, data is a major asset for any enterprise, not only for knowing the past, but also to aid today's business or to predict future trends [3, 20]. On-Line Analytical Processing (OLAP) and Business Intelligence tools use DWs to produce business knowledge. This makes them a key business asset for any enterprise; DWs are the vault of the enterprise's sensitive business information. Unfortunately, this also makes them an appealing target for malicious inside and outside attackers. Recently published security statistics shows the number of attacks on enterprise data has been continuously increasing [43]. Data security focuses on issues such as confidentiality (or privacy), integrity (including correctness, authenticity and consistency), and availability of data. Confidentiality focuses on protecting information from unauthorized disclosure, either by direct retrieval or by indirect logical inference [14]. Integrity requires protecting data from malicious or accidental changes, including insertion of false data, contamination or destruction of data. Availability ensures data is available to all authorized users whenever they need it. Many data security solutions for databases have been proposed in the past. Some solutions are currently available in main Relational DataBase Management Systems (RDBMS) such as Oracle 11g and MySQL v5, or can be developed and integrated with DWs in a forward manner. Although these solutions have been scientifically proved to be effective, we shall explain why these proposals are unfeasible or, at least, inefficient for usage in DWs, due to specific performance requirements of data warehousing environments. In this paper, we present a survey on today's available data security solutions, focusing on their use for data warehousing scenarios. We present the issues concerning each type of data security solution – data access policies, techniques for enforcing data privacy, intrusion detection, ongoing availability techniques, and methods for recovering from attacks – discussing weak spots and pointing out research opportunities for improving the existing solutions or developing new ones.

The remainder of this paper is organized as follows. In Section 2, we present the existing data security solutions, and discuss the specific issues and requirements for their use in data warehousing environments. In Section 3, we point out the open research opportunities that need to be tackled. Finally, Section 4 presents our conclusions.

II. Data Security Solutions for Data Warehousing

A. *Preventive Data Security Solutions*

Preventive data security techniques are used for protecting data in advance of attacks, such as implementing referential integrity and concurrency constraints, data access policies, data masking and encryption techniques for changing original data values, and checksums for integrity checks on changed data.

Current DataBase Management Systems (DBMS) allow defining referential integrity constraints, data validation rules, role-based access control policies, and comply with ACID requirements, all of which assure data consistency, correctness, and confidentiality, up to a certain point. Checksum techniques have always been used in DBMS for error checking of stored data and detecting data corruptions. Approaches for distinguishing original data from tampered data is using signatures in all records of the DW, as published in [4, 40]. Another approach for detecting correctness errors are the well-known CRC, MD5 and SHA algorithms.

Data masking is an easy way of avoiding disclosure of data by simply changing and replacing original data values. Oracle, for instance, explains current best practices for data masking in their DBMS in [28]. Encryption is an advanced form of data masking and is a widely used technique for enforcing data privacy. Oracle has developed its TDE (Transparent Data Encryption) [27, 29] in versions 10g and 11g of their DBMS, incorporating the well-known standard encryption algorithms AES and 3DES. Oracle 11g TDE encrypts data using a set of master and secondary keys, which can be applied on column and tablespace encryption. These techniques are transparent, not requiring any user source code modifications. If the database tablespace is stolen or copied without clearance, it will not allow any data to be shown correctly, since its content is all encrypted. The MySQL v5 DBMS provides only AES data encryption functions. Although proved efficient in ensuring strong protection, encryption involves several costs:

- Extra storage space of encrypted data;
- Time needed for encrypting sensitive data. Given DW decision support nature, we may assume that almost all of its data is sensitive;
- Overhead in query response time and allocated resources for decrypting data to process queries.

Given the volume of data DW queries typically access, the cost for processing their execution together with decrypting encrypted data usually produces unacceptable response time overheads [37]. We performed an experimental evaluation of the data encryption solutions provided by Oracle 11g TDE, using the well known TPC-H benchmark [35], for measuring the impact on performance for the benchmark's 22 query workload on its 1GB scale database. Although Oracle argues using TDE will only increase response time an average of 5% to 10% [29], in our tests this has shown not to be true. The results show the response time overhead is, on average, much higher than 5%. In fact, it ranges from 30% to 163% for the whole workload, depending on which encryption algorithm is used, as shown in Figure 1. Moreover, the individual query execution time overhead for more than a third of the queries registered 100% to 1000%, as shown in Figure 2.

Currently, all major DBMS supply audit control, backups and tablespace corruption recovery, comply with ACID requirements, allow using standard encryption algorithms and offer extensive authentication, authorization, and access control (AAA) features for defining data access policies for assuring the right users get the right data. Solutions for the inference problem in DWs have also been proposed [1, 39]. However, given the increase of sophisticated attacks and rising internal theft, preventive security techniques and traditional AAA features are no longer enough to protect data [43]. This has lead to the development of reactive data security techniques. These consist on intrusion detection, auto-repair, auto-recovery, and fault-tolerance, among others, which try protecting data from attackers able to bypass preventive security techniques.

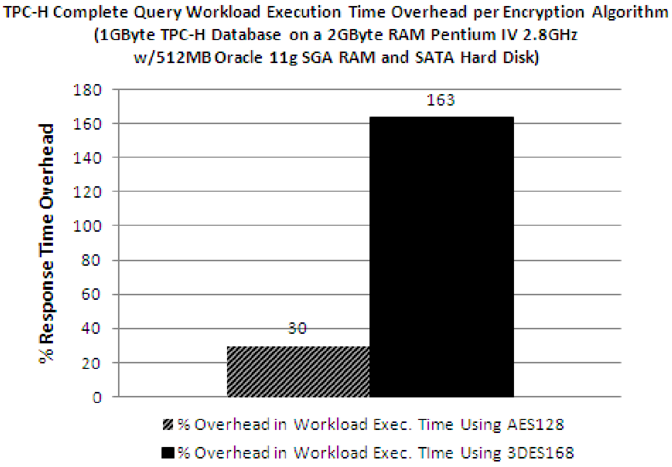

Figure 1. TPC-H Query Workload Execution Time Overhead per Encryption

B. *Reactive Data Security Solutions*

Detecting unauthorized access is the main goal of Intrusion Detection Systems (IDS), based on two general approaches: misuse detection, looking for patterns signaling well-known attacks; and anomaly detection, looking for deviations from normal behavior. Anomaly detection may rely on statistical approaches or predictive pattern generation. Misuse detection is mostly based on detecting predefined attack patterns. In both techniques, Data Mining (DM) is used to reduce human effort and increase detection accuracy [22]. In recent years, DM-based IDS for databases have been developed [5, 6, 10, 13, 16, 19, 21, 26, 33, 34, 44]. Supervising user queries is also a component of IDS. In [7, 8, 15, 41, 42], data mining and/or machine learning approaches are proposed for dealing with SQL injection.

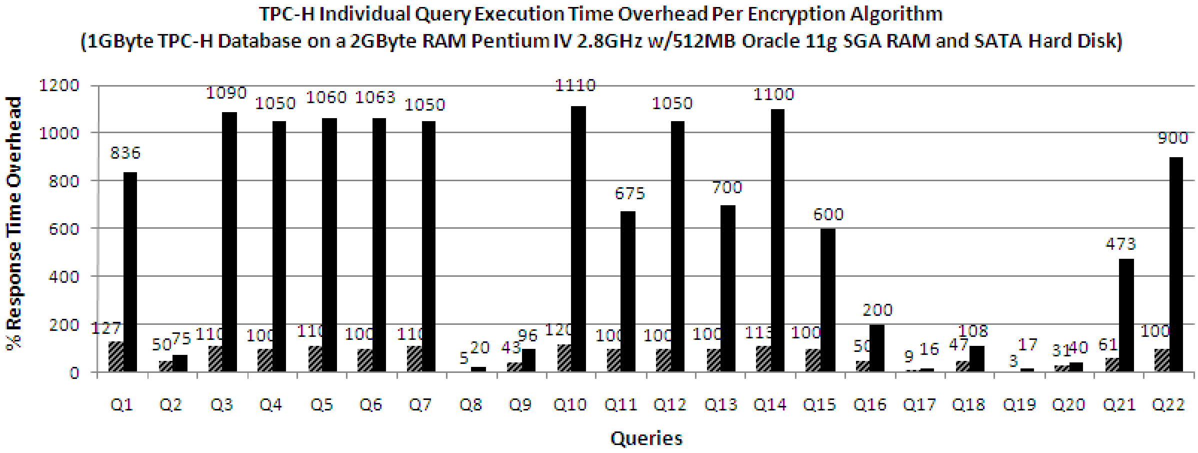

Figure 2. TPC-H Individual Query Execution Time Overhead per Encryption Algorithm

The main tasks in DW data availability involve real-time recovery of corrupted or incorrectly modified data and continuous 24/7 user access. Most solutions solve these issues by replicating data to restore damaged data at any time, allow maintenance interventions avoiding database downtime, and are able to divide query processing efforts in order to avoid data access hotspots. One hardware approach for mirroring data is the application of the well-known RAID architectures [17, 18, 31], on systems where databases lie in centralized servers. However, for optimizing costs, more and more enterprises have been implementing their DWs in low-cost commodity computers, where typically only one disk drive is present and RAID technology is not an option.

Efficient commercial solutions for solving data availability issues in DWs are available today in the market, such as Oracle RAC [30] and Aster Data [2]. Another approach for correcting corrupted data consists on applying error correction codes such as Hamming codes. Data storage systems have been proposed, able to recover from data block corruption, using error correcting codes, replication and remapping of bad blocks, such as [32, 38]. Other systems use these features and add encryption techniques for distributing storage [25]. Architectures for damage assessment and self-healing databases have also been proposed [9, 11, 12, 23, 24]. Although strongly effective for availability purposes, data replication techniques are always an important issue in DWs, given the volume of data and storage size typically involved.

III. Research Challenges and Opportunities

Although standard encryption algorithms are available in today's major DBMS and are able to provide strong data privacy, their impact in database performance makes them unfeasible for usage in DWs. The computational efforts required by algorithms like AES and 3DES have a huge impact on performance, as shown before. Alternatives that minimize overhead in query response time are needed, while being able to achieve a strong level of privacy. Given the speed and simplicity of bitwise operations, perhaps bit-based encryption formulas may provide a way to achieve new efficient and feasible solutions. Of course, if the encryption process is made simpler for the sake of improving database performance, the level of privacy will get weaker. A compromise of the trade-offs must be defined, satisfying the intended level of privacy while minimizing the impact in performance. Other alternative could be the development of query engines that could be able to process the query directly on encrypted data, i.e., without needing to decrypt data.

Row signatures may not be feasible for DWs, because they require reading all columns from a row to verify the signature, which may not be the best approach in terms of performance. Using one signature for each column in each row is an alternative; however, it brings a storage space problem that also influences performance. Given that DW fact tables are typically composed of numerical values and represent 90% or more of DW storage space, a large portion of DW data usually consists on numerical values, a feature that can be used for developing entire-column functions for confidentiality and integrity purposes. A research challenge is to investigate the possibility of having a single signature for validating each column individually and also to validate the entire row at once, while maintaining high database performance. Thus, the main research question in preventive data security for DWs is: *How to improve data masking, encryption and signature techniques for enhancing data integrity and confidentiality, in order to overcome their current computational effort and storage overheads, making them feasible for use in DWs?*

Given the potential damage, detecting malicious intrusions as quickly as possible is critical for taking corrective action. Although recent proposals have enhanced IDS capabilities, they have not been capable of efficiently detecting malicious actions (perceiving intent) after authorized access is granted to users, or to significantly reduce the number of false positives in a highly heterogeneous environment such as DWs [8, 36]. Many decision support queries have an ad-hoc nature, where any form of instruction may be executed or any portion or amount of data may be accessed. This means that it is very difficult to determine "normal user behavior" and "probable attack behavior", making it extremely hard to distinguish between normal, and misuse or abnormal behavior using current IDS. Current IDS are poor at detecting novel attacks in DWs without typically resulting on a very high number of false positives [36]. This frequently leads to wasting an immense amount of time and limited resources on false alarms, thus decreasing confidence in the IDS or even making their usage unacceptable. Under this perspective, more efficient solutions able to reduce the number of false positives are needed that can deal with the specific user requirements of DW environments, without risking database performance. The main research question for DW IDS is then: *How to improve database IDS efficiency and effectiveness in order to distinguish normal users from malicious attackers, in real-time, i.e., while the attack takes place without jeopardizing database performance?*

When using data replication techniques, load balancing for optimizing query performance depends on the nature of the data values themselves. Moreover, given the amount of storage space needed by DWs, technique that enlarges that space, such as data replication, is always an important issue. Restoring data to recover from attacks needs to be done as quickly and effectively as possible, preferably with no server downtime. This is a non trivial task, given that an attack may damage millions of rows or more. Thus, the main research question for data recovery in DWs is: *How to improve existing recovery mechanisms for quickly, efficiently and effectively repairing and/or restoring data, without jeopardizing database availability?*

Little work has been done in proposing a benchmark for evaluating security in databases. The work in [37] is an exception, which proposes a class-based characterization of security mechanisms in database systems and applications. However, it is generic and very broad scoped, making it an incomplete tool for evaluating specific DW security. The main research question here is: *How can we assess the level of security of any given DW?*

Finally, given the increasing usage of open-source solutions in the real-world, the development and assessment of data warehousing security solutions for use in both commercial DBMS, such as Oracle, and open source DBMS, such as MySQL, should be considered.

IV. Conclusions

The currently available data security solutions for data warehousing, discussing their issues and impact in DW performance and scalability requirements, have been presented. We have also shown that these solutions are often inefficient and unfeasible to use in data warehousing environments. DWs function in a well-determined specific environment with tight performance and scalability requirements and, therefore, need specific solutions able to cope with these directives. We have referred their weak spots from the DW perspective, pointing out the research challenges which make ground for significant opportunities in this field, given the lack of specific data warehousing security solutions.

References

[1] Agrawal, R., Srikant, R., and Thomas, D., "Privacy Preserving OLAP", Int. Conf. SIG on Management Of Data (SIGMOD), 2005.

[2] AsterData Systems, "Aster Data nCluster: Always On, for 24x7 Big Data Analytics", http://www.asterdata.com/product/alwayson.php, 2010.

[3] Baer, H., "On-Time Data Warehousing with Oracle Database 10g – Information at the Speed of Your Business", Oracle White Paper, Oracle Corporation, 2004.

[4] Barbara, D., Goel, R., and Jajodia, S., "Using Checksums to Detect Data Corruption", Int. Conf. Extending DataBase Technology (EDBT), 2000.

[5] Barbara, D., Jajodia, S., Wu, N., Stolfo, S., Lee, W., et al., "SIGMOD Record Special Issue on Data Mining for Intrusion Detection and Threat Analysis", SIGMOD Record, Vol. 30, No. 4, 2001.

[6] Bertino, E., Kamra, A., Terzi, E., and Vakali, A., "Intrusion Detection in RBAC-administered databases", 21st Annual Computer Security Applications Conference (AC-SAC), 2005.

[7] Bertino, E., Kamra, A., and Early, J. P., "Profiling Database Applications to Detect SQL Injection Attacks", Int. Performance Computing and Communications Conference (IPCCC), 2007.

[8] Bockermann, C., Apel, M., and Meier, M., "Learning SQL for Database Intrusion Detection using Context-Sensitive Modeling", Int. Conference on Knowledge Discovery and Machine Learning (KDML), 2009.

[9] Bohannon, P., Rastogi, R., Seshadri, S., Silberschatz, A., and Sudarshan, S., "Detection and Recovery Techniques for Database Corruption", IEEE Trans. on Knowledge and Data Engineering, Vol. 15, No. 5, 2003.

[10] Campos, M. M., and Milenova, B. L., "Creation and Deployment of Data Mining-Based Intrusion Detection Systems in Oracle Database 10g", Int. Conf. on Machine Learning and Applications (ICMLA), 2005.

[11] Chakraborty, A., Majumdar, A. K., and Sural, S., "A Column Dependency-Based Approach for Static and Dynamic Recovery of Databases from Malicious Transactions", Int. Journal of Information Security (9), 2010.

[12] Chiueh, T., and Pilania, D., "Design, Implementation, and Evaluation of a Repairable Database Management System", Computer Security Applications Conference (AC-SAC), 2004.

[13] Dia, J., and Miao, H:, "D_DIPS: An Intrusion Prevention System for Database Security", Int. Conf. on Information and Communications Security (ICICS), 2005.

[14] Farkas, C., and Jajodia, S., "The Inference Problem: A Survey", SIGKDD Explorations, Vol. 4, Issue 2, December 2002.

[15] Fonseca, J., Vieira, M., and Madeira, H., "Online Detection of Malicious Data Access Using DBMS Auditing", Latin-American Symposium on Dependable Computing (LADC), 2007.

[16] Hu, Y., and Panda, B., "A Data Mining Approach for Database Intrusion Detection", ACM Symposium on Applied Computing (SAC), 2004.

[17] IBM Corporation, "Understanding RAID level-5", IBM Systems Software Information Center, November 2007.

[18] IBM Corporation, "Understanding RAID level-6", IBM Systems Software Information Center, November 2007.

[19] Kamra, A., Terzi, E., and Bertino, E., "Detecting Anomolous Access Patterns in Relational Databases", VLDB Journal, 17, 2008.

[20] Kobielus, J., "The Forrester Wave: Enterprise Data Warehousing Platforms", Forrester Research Report, Q1 2009.

[21] Kundu, A., Sural, S., and Majumdar, A. K., "Database Intrusion Detection Using Sequence Alignment", Int. Journal of Information Security (9), 2010.

[22] Lee, S. Y., Low, W. L., and Wong, P. Y., "Learning Fingerprints for a Database Intrusion Detection System", European Symposium on Research in Computer Security (ESORICS), 2002.

[23] Liu, P., and Jing, J., "Architectures for Self-Healing Databases under Cyber Attacks", Int. J. Computer Science and Network Security, 2006.

[24] Luenam, P., and Liu, P., "ODAM: An On-the-fly Damage Assessment and Repair System for Commercial Database Applications", International Conference on DataBase Security (DBSec), 2001.

[25] Marsh, M. A., and Schneider, F. B., "CODEX: A Robust and Secure Secret Distribution System", IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, 2004.

[26] Mohan, S. R., Park, E. K., Han Y., "An Adaptive Intrusion Detection System Using a Data Mining Approach", Int. Conf. on Data Mining (ICDM), 2005.

[27] Oracle Corporation, "Security and the Data Warehouse", Oracle White Paper, April 2005.

[28] Oracle Corporation, "Oracle Advanced Security Transparent Data Encryption Best Practices", Oracle White Paper, July 2010.

[29] Oracle Corporation, "Data Masking Best Practices", Oracle White Paper, July 2010.

[30] Oracle Corporation, Oracle Real Application Clusters (RAC), http://www.oracle.com/us/products/database/options/real-application-clusters/index.htm, September 2010.

[31] Patterson, D., Gibson, G., and Katz, R. H., "A Case for Redundant Arrays of Inexpensive Disks (RAID)", ACM Special Interest Group International Conference on Magament Of Data (SIGMOD), 1988.

[32] Prabhakaran, V., Bairavasundaram, L. N., Agrawal, N., Gunawi, H. S., Arpaci-Dusseau, A. C., and ArpaciDusseau, R. H., "IRON File Systems", Int. Symp. on Operating System Principles (SOSP), 2005.

[33] Rao, U. P., Sahani, G. J., and Patel, D. R., "Clustering Based Machine Learning Approach for Detecting Intrusions in RBAC Enabled Databases", Int. J. Computer and Network Security, Vol. 2, No. 6, 2010.

[34] Srivastava, A., Sural, S., and Majumdar, A. K., "Database Intrusion Detection using Weighted Sequence Mining", Journal of Computers, Vol. I, No. 4, 2006.

[35] Transaction Processing Performance Council, "The TPC Decision Support Benchmark H", http://www.tpc.org/tpch/default.asp

[36] Treinen, J. J., and Thurimella, R., "A Framework for the Application of Association Rule Mining in Large Intrusion Detection Infrastructures", Recent Advances in Intrusion Detection (RAID), 2006.

[37] Vieira, M., and Madeira, H., "Towards a Security Benchmark for Database Management Systems", Int. Conf. on Dependable Systems and Networks (DSN), 2005.

[38] Vijayasankar, K., Sivathanu, G., Swaminathan, S., and Zadok, E., "Exploiting Type-Awareness in a Self-Recovery Disk", StorageSS, 2007.

[39] Wang, L., Jajodia, S., and Wijesekera, D., "Securing OLAP Data Cubes Against Privacy Breaches", IEEE Symp. on Security and Privacy (SSP), 2004.

[40] Wang, L., Wijesekera, D., and Jajodia, S., "Cardinality-Based Inference Control in Sum-Only Data Cubes", European Sumposium on Research in Computer Security (ESORICS), 2002.

[41] Wei, K., Muthuprasanna, M., and Kothari, S., "Preventing SQL Injection Attacks in Stored Procedures", Australian Software Engineering Conference (AWSEC), 2006.

[42] Yu, Z., Tsai, J. P., and Weigert, T., "An Automatically Tuning Intrusion Detection System", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 37, No. 2, 2007.

[43] Yuhanna, N., "Your Enterprise Database Security Strategy 2010", Forrester Research, September 2009.

[44] Zhong, Y., and Qin, X., "Database Intrusion Detection Based on User Query Frequent Itemsets Mining with Item Constraints", Information Security Conference (InfoSecu), 2004.