Masking of Data For ERP Test Environment

Shantanu Thengdi, Saurabh Nair, Nikhil R. Deshmukh, Kirti Kawley and Manali Narnaware

Student, Department of Computer Science & Engg, Rajiv Gandhi College of Engg. & Research, Nagpur University, Maharashtra, India

E-mail:saurabhunair@gmail.com
Contact no. : +91 8446069690

Abstract— Since the last few years, many companies are migrating their posting on cloud-based solution, thus increasing the access to confidential data like personal information, payroll data, customer-information, etc. to hosting provider team. Also even in case of in-house hosting, development has access to testing/development server which is cloned copy of Production to be sent to Non-production. It's very critical to help companies to secure their data by means of Scrambling/Masking data so that development and hosting team can continue to work without really having access to real data. There is a need to develop application that is scalable and cost effective and help clients secure their data. We have suggested various ways of masking data in such a way that the masking preserves the format and consistency of data by enabling data privacy.

Keywords — Confidential Data, Non-Production, Scrambling, Masking, Testing, Data Privacy, Substitution.

INTRODUCTION

Data masking enables organizations to generate realistic and fully functional data with similar characteristics as the original data to replace sensitive or confidential data. This contrasts with encryption or Virtual Private Database, which simply hides data instead of masking it, and the original data, can be retrieved with the appropriate access or key after encryption. Data masking does not allow the original sensitive data to be retrieved or accessed. Names, addresses, phone numbers, social security number and credit card details are examples of data that require protection of the information content from inappropriate and unauthorized visibility. Live production database environments contain valuable and confidential information—access to this information is tightly controlled and highly restricted. However, each production system usually has cloned development copies, and the restrictions on such test environments are less stringent. This greatly increases the possibility that the data might be used inappropriately for personal gains. Data masking can modify sensitive databases records so that they remain usable for testing, but do not contain important confidential or personally identifiable information which causes privacy concerns. Yet, the masked data used for testing resembles the original in appearance to ensure the integrity of the application [1].

Companies and agencies of all types can gather, store, and process large amount of data. Improving business process using analytical and data mining tools to retrieve information from the data is the primary objective of gathering such data. Organizations run the risk of compromising sensitive information when copying Production data in to Non-Production environments for the purposes of application development, testing, or data analysis [4]. Data Masking helps reduce this risk by irreversibly replacing the original sensitive data with fictitious data so that production data can be shared safely for use in non-production environment. An increasing number of enterprises are depending on data masking to actively secure organizational data, ameliorate data security measures and lower costs associated with data breach. Data masking protects data by de-identifying sensitive information contained in non-production environments and enables enterprises to extend their traditional security platforms. It masks sensitive or confidential data so that it can be replicated safely to non-production systems. Using previously built or customized complex masking techniques; IT organizations can safeguard the original information characteristics (data types, formats, etc.) and maintain data and referential integrity. For application development and testing realistic data is required. Usually, development teams are given copies of production environment databases that are created using internally developed scripting. However this method is not fully secure since real data with sensitive information could fall in wrong hands [3]. While testing an online banking system, application tester can manipulate customer records and as a result can view names, addresses, social security numbers, phone numbers and other private information of individual causing great security and privacy concerns.

Architecture

Contrary to the proposed system, the architecture of the project is rather less complex, although there is ample scope for improvement and extension to each and every module associated with the same.

There can be various phases derived out in order to get the basic idea behind the Architecture as follows :–

A. Authentication
B. Fetching
C. Algorithm Execution
D. Delivery

A. *Authentication*

Authentication is the process by which it is ensured that the person or system requesting access to a piece of information has valid authority to access it. In private and public computer networks (including the Internet) this method is used. A basic username-password system is provided for authenticating the user and letting him enter into the system.

B. *Fetching*

A successful authentication paves way for fetching of all the tables in the database. On the whole, tables get fetched for selection in order to get them a valid participation in the next phases, listing the tables along with their key relations (e.g.: Foreign key).

C. *Algorithm Execution*

Once the user scrambles / masks the specified table or even rows / columns, the algorithm fed to the system gets triggered and the data is manipulated for further use.

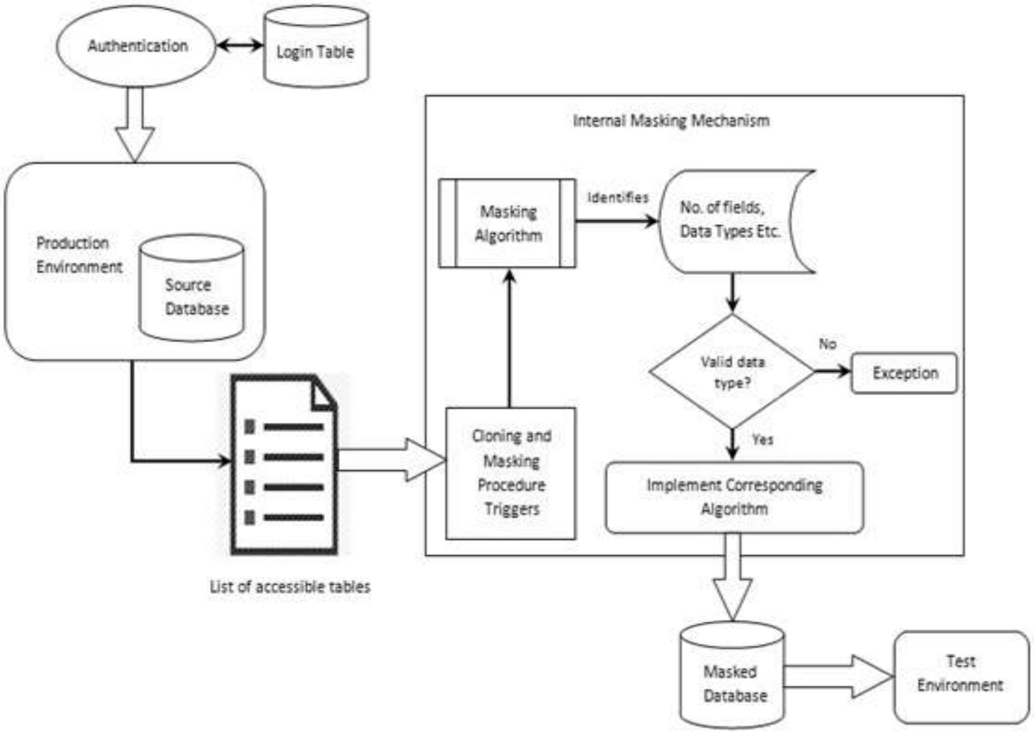
Authentication
Login Table
Production Environment
Source Database
List of accessible tables
Internal Masking Mechanism
Masking Algorithm
identifies
No. of fields, Data Types Etc.
Valid data type?
No
Exception
Yes
Implement Corresponding Algorithm
Cloning and Masking Procedure Triggers
Masked Database
Test Environment

Fig. 1 : System Architecture

Fig. 1 System Architecture

D. *Delivery*

Finally, the Masked Database is sent to the Test Environment, which may then implement various new developmental and analytical processes with data (masked) in hand as good as original data without compromising the security.

Different types of Data Masking Techniques

A. *Substitution*

Substitution is one of the most effective methods of applying data masking and being able to preserve the authentic look and feel of the data records. Substitution is the technique used to cover or mask original values with a exact substitute for it. Substitution is used for encryption of data where there is a need for providing the user with such data that looks authentic but is not useful to him. Using substitution various data and data types can be altered. It is very much similar to encrypting data with an old encryption cipher. There are various methods of substitution which can be used based on provided circumstances. This technique can be used in places with large customer information or in a banking system that stores values regarding client's transactions [1]. The example of such a system can be of a table in database having customer records with the customer's name, last name and sex provided, by using substitution we can replace the first few characters of names of male customers with particular character and do the same with the first few characters of the female customer's name. Substitution can be done on various database fields like zip code, salaries, social security number and even for addresses [15].

B. *Shuffling*

Shuffling is a data obfuscation technique that derives a new value for the current set of value by replacing it with the values that are being masked in the column. Though data shuffling is somewhat similar to substitution it has different approach to masking the data. But it is much easier to gain knowledge about the original information by placing a particular scenario on the data set. It will also be a cause of concern if the original algorithm for ciphering the shuffle of data is deciphered. The shuffling technique has some real advantages over substitution in certain scenarios where there are certain requirements for mask to be provided in such a way that it affects only required fields of data [4]. Example of shuffling can be of a social security number which has 111-222-888 format before shuffling and after shuffling it can become 888-111-222.

C. *Encryption*

Broadly, speaking there are three types of cryptographic algorithms: secret key algorithms, public key algorithms and hashing algorithm. Secret key algorithms are symmetric in the sense that both participants in the communication share a single key. In contrast to a pair of participants having a private key that is shared with no one else, a public key is published so it is known to everyone. In case of data masking we can apply encryption on the data in the fields by using any of the available encryption techniques. There are various standardized algorithms which can be used while applying masking operation. The various encryption algorithms are DES, MD5, SHA, 3DES, etc. [13].Use of these algorithms help in masking of the data in a highly secured manner as these algorithms apply many iterations of encryption on the data.

D. *Number and Date variance*

Numeric and date variance technique can be used to mask various numeric and date related field in the table which are of importance. The numeric and date variance can be used to make the date or a numeric value seem different from its original value [1]. If the numeric variance is applied on a salary column for a +20/-20 variance then the data obtained is still useful for the range of salaries that are paid to the employees and can still be used for testing applications. The same can be done on a field having dates where one can increase or the decrease the current date by +40/-40 days for masking. This can prevent anyone with improper authorization from viewing correct salaries or date of birth of other recipients.

E. *Nulling out or Deletion*

Another technique for masking data is use of null and deletion operation. Though the null value prevents anyone from viewing important data but it certainly fails the requirements of the data used for testing. The nulling out method is generally used on fields which are either highly important or are too complicated to mask. Another problem with nulling is that it exposes to anyone who wishes to penetrate the system that masking has been applied on the dataset. Deletion is the other method for masking though it is very rarely applied.

Methodology

Scrambling/Masking of sensitive and confidential data requires an extensive research on zeroing upon a single strategy. The most common yet effective approach is discussed as follows using fig 2. :

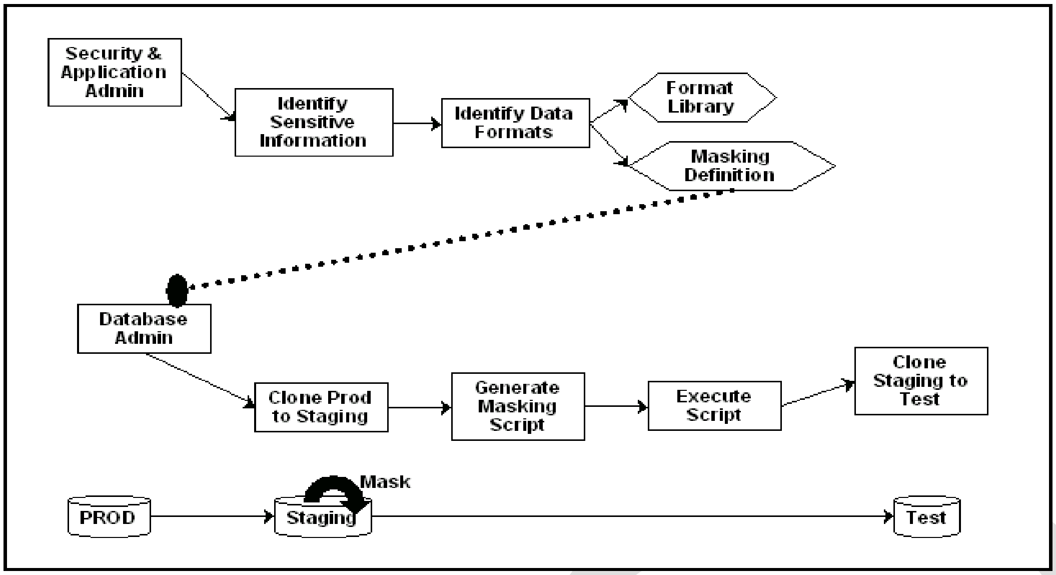

Fig 2. Data Masking Methodology

1. Security and Application Administrator carries out tasks such as Authentication, Firewall Protection, etc.

2. Sensitive Information is identified and selected for masking by the user.

3. Data Formats / Contents are modified with the use of Format Library and Masking Definition.

4. Database Administrator surveys the output and surveillance is carried out for clone production from Live Production environment.

5. Masking Script is generated in case DBA or any concerned authority requires decryption or de-masking algorithm provided secrecy of confidential information.

6. Script is executed on clone data, which is the most important step.

7. Cloned data is provided to the Test Environment for Test and Development purpose.

8. Masked data is used by this environment to carry out developmental tasks.

Result

Production Environment, with the help of the proposed Data Scrambling/Masking application, masks the cloned data and this data is sent to the Non-Production Environment which in a way, does two things; Format and data confidentiality, both get preserved at the same time. Plus the Non-production environment can make changes to the as-good-as-original data and test, develop and analyze new modules to it.

Conclusion

Summarizing the discussed points as a whole, we can say that Data Security is the need of the hour and to achieve it in an ERP system which requires a good stability involves various databases and relationships, we can conclude that Intra-Enterprise data transfers must have secure Scrambling and Masking Systems in order to emphasize confidentiality and security in the company. The system discussed is a primitive version of what can be extended as a whole for a large-scale ERP system and can be thought of as a feasible (cost-effective and labor-saving) option for Data Masking.

Thus, with incremental support and extension to the said module, this tool has the ability to serve as an ERP companion for data masking and scrambling during interaction of Database with Test Environment.

REFERENCES:

[1] Ravi Kumar G K ,Manjunath T N, Ravindra S Hegadi, Umesh I M : "A Survey on Recent Trends ,Process and Development in Data masking for testing", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011, p-535-544.

[2] Adam N. R. and Wortmann, J. C. 1989."Security-Control Methods for Statistical Databases: A Comparative study," ACM Computing Surveys (21:4), pp. 515 –556

[3] Ravi Kumar G K, Dr. B. Justus Rabi, Manjunath T N, Dr. Ravindra S Hegadi, Archana.R.A : "Design of Data Masking Architecture and Analysis of Data Masking Techniques for Testing", International Journal of Engineering Science and Technology (IJEST), ISSN : 0975-5462 Vol. 3 No. 6 June 2011.

[4] Ravi Kumar G K, Dr. B. Justus Rabi, Manjunath T N, Dr. Ravindra S Hegadi, Archana.R.A "Experimental Study of Various Data Masking Techniques with Random Replacement using data volume",(International Journal of Computer Science and Information Security (IJCSIS), Vol. 9, No. 8, August 2011

[5] Liew, C. K., Choi, U. J., and Liew, C. J. 1985."A Data Distortion by Probability Distribution,"ACM Transactions on Database Systems (10:3), pp.395-41

[6] Domingo-Ferrer J., and Mateo-Sanz, J. M. 2002."Practical Data-Oriented Micro aggregation for Statistical Disclosure Control," IEEE Transactions on Knowledge and Data Engineering (14:1), pp. 189- 201.

[7] Xiao-Bai Li, Luvai Motiwalla by "Protecting Patient Privacy with Data Masking" WISP 2009

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

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

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

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

[12] V. Radha and N. H. Kumar, "EISA – An Enterprise Application Security Solution for Databases", Int. Conf. on Information Systems Security (ICISS), S. Jajodia and C. Mazumdar (Eds), Springer LNCS 3803, 2005.

[13] U. T. Mattson, "Database Encryption – How to Balance Security with Performance", Protegrity Corporation Technical Paper, 2004.

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

[15] AMBROSIA, V. G., BUECHEL, S. W., BRASS, J. A., and PETERSON, J. R., 1998, An integration of remote sensing, GIS, and information distribution.

[16] Greengard, S.1996. "Privacy: Entitlement or Illusion?" Personnel Journal (75:5), pp. 74-88.