Data Sanitization Techniques

A Net 2000 Ltd. White Paper

Abstract

Data Sanitization is the process of making sensitive information in non-production databases safe for wider visibility. This White Paper is an overview of various techniques which can be used to sanitize sensitive production data in test and development databases. An initial discussion of the primary motivations for data sanitization is given. The remainder of the paper is devoted to a generic survey of the various masking techniques and their individual benefits and drawbacks.

It must be noted that Net 2000 Ltd., (the authors of this paper), sell a software data anonymization tool called Data Masker (http://www.DataMasker.com). However, as the title states, this paper really is a generic survey of data sanitization techniques and there is no further reference to any specific software. If you wish to know more, or have any questions about issues raised in this whitepaper please contact us.

Note: This paper has been superseded by the *Data Masking: What You Need To Know* white paper. We recommend you download that paper as it has expanded and enhanced content.

http://www.DataMasker.com/DataMasking_WhatYouNeedToKnow.pdf

Some keywords which may assist you in finding this document online are:

Data Sanitization, Data Sanitisation, Data Anonymization, Data Scrubbing, Data Scrambling, Data Masking, Data Obfuscation, Data Security, Data Cleansing, Data Hiding, Data Protection Act 1998, Hide Data, Disguise Data, Sanitize Data, Sanitise Data, Gramm-Leach-Bliley Act (GLBA), Data Privacy, Directive 95/46/EC of the European Parliament, Health Insurance Portability and Accountability Act (HIPAA)

Net 2000 Ltd.
http://www.Net2000Ltd.com
Info@Net2000Ltd.com

Table of Contents

Disclaimer

The contents of this document are for general information purposes only and are not intended to constitute professional advice of any description. The provision of this information does not create a business or professional services relationship. Net 2000 Ltd. makes no claim, representation, promise, undertaking or warranty regarding the accuracy, timeliness, completeness, suitability or fitness for any purpose, merchantability, up-to-datedness or any other aspect of the information contained in this paper, all of which is provided "as is" and "as available" without any warranty of any kind.

The information content of databases varies widely and each has a unique configuration. Readers should take appropriate professional advice prior to performing any actions.

Data Sanitization Techniques

Data Sanitization is the process of disguising sensitive information in test and development databases by overwriting it with realistic looking but false data of a similar type.

Why Sanitize Information in Test and Development Databases?

The data in testing environments should be sanitized in order to protect valuable business information and also because there is, in most countries, a legal obligation to do so.

Protecting Valuable Information

Fundamentally there are two types of security. The first type is concerned with the integrity of the data. In this case the modification of the records is strictly controlled. For example, you may not wish an account to be credited or debited without specific controls and auditing. This type of security is not a major concern in test and development databases. The data can be modified at will without any business impact.

The second type of security is the protection of the information content from inappropriate visibility. Names, addresses, phone numbers and credit card details are good examples of this type of data. Unlike the protection from updates, this type of security requires that access to the information content is controlled in every environment.

Legal Obligations

The legal requirements for Data Sanitization vary from country to country but most countries now have regulations of some form. Here are some examples:

United States

The Gramm-Leach-Bliley Act requires institutions to protect the confidentiality and integrity of personal consumer information. The Right to Financial Privacy Act of 1978 creates statutory Fourth Amendment protection for financial records and there are a host of individual state laws. There are also a number of security and privacy requirements for personal information included in the Health Insurance Portability and Accountability Act of 1996 (HIPAA).

The European Union

Directive 95/46/EC of the European Parliament which provides strict guidelines regarding individual rights to data privacy, and the responsibilities of data holders to guard against misuse.

The United Kingdom

The United Kingdom Data Protection Act of 1998 extends the European Parliament directive and places further statutory obligations on the holders of personal, private or sensitive data.

As with most things legal, the details are open to argument. In reality though, if data for which your organization is responsible gets loose and appropriate steps were not taken to prevent that release, then your organizations lawyers could well find themselves in court trying to put their best spin on the matter. However large the legal liabilities are, they could seem trivial in comparison to the losses associated with the catastrophic loss of business confidence caused by a large scale privacy breach.

Any organization that outsources test and development operations needs to be very conscious of the specific laws regulating the transmission of information across national borders.

Data Scrambling Issues

This paper discusses practical techniques which can be used for the masking of sensitive information in test and development databases. Techniques are only one aspect of the solution - there are also a number of issues which must be addressed in order to successfully and securely sanitize the data.

Such issues include large-scale data synchronization requirements such as "*Row Internal Synchronization*", "*Table Internal Synchronization*" or "*Table - Table Synchronization*" – without which the sanitized test data may be rendered unusable. There are also very subtle security issues which must be considered. If such issues as the "*The Isolated Case Phenomena*", "*Where Clause Skips*" and "*Sparse Data*" are not dealt with, then data which seems to have been rendered anonymous, may still remain in an interpretable state.

A separate white paper, devoted to a discussion of these issues, has been prepared by Net 2000 Ltd. and is available for download on the Internet. It should be noted that the names used above are ours and were composed in order to provide a short descriptor for easy reference. As far as we know there is no widely agreed nomenclature.

No attempt is made herein to provide any discussion of data scrambling issues. For a comprehensive discussion please see the companion white paper:

Data Scrambling Issues

http://www.DataMasker.com/datascramblingissues.pdf

Data Sanitization Techniques

Test and development teams need to work with databases which are structurally correct functional copies of the live environments. However, they do not necessarily need to be able to view security sensitive information. For test and development purposes, as long as the data looks real, the actual record content is usually irrelevant. There are a variety of Data Sanitization techniques available – the pro's and con's of some of the most useful are discussed below.

Technique: NULL'ing Out

Simply deleting a column of data by replacing it with NULL values is an effective way of ensuring that it is not inappropriately visible in test environments. Unfortunately it is also one of the least desirable options from a test database standpoint. Usually the test teams need to work on the data or at least a realistic approximation of it. For example, it is very hard to write and test customer account maintenance forms if the customer name, address and contact details are all NULL values.

Verdict: The NULL'ing Out technique is useful in certain specific circumstances but rarely useful as the entire Data Sanitization strategy.

Technique: Masking Data

Masking data means replacing certain fields with a Mask character (such as an X). This effectively disguises the data content while preserving the same formatting on front end screens and reports. For example, a column of credit card numbers might look like:

```
4346 6454 0020 5379
4493 9238 7315 5787
4297 8296 7496 8724
```

and after the masking operation the information would appear as:

```
4346 XXXX XXXX 5379
4493 XXXX XXXX 5787
4297 XXXX XXXX 8724
```

The masking characters effectively remove much of the sensitive content from the record while still preserving the look and feel. Take care to ensure that enough of the data is masked to preserve security. It would not be hard to regenerate the original credit card number from a masking operation such as: `4297 8296 7496 87XX` since the numbers are generated with a specific and well known checksum algorithm. Also care must be taken not to mask out potentially required information. A masking operation such as `XXXX XXXX XXXX 5379` would strip the card issuer details from the credit card number. This may, or may not, be desirable.

Verdict: If the data is in a specific, invariable format, then Masking is a powerful and fast Data Sanitization option. If numerous special cases must be dealt with then masking can be slow, extremely complex to administer and can potentially leave some data items inappropriately masked.

Technique: Substitution

This technique consists of randomly replacing the contents of a column of data with information that looks similar but is completely unrelated to the real details. For example, the surnames in a customer database could be sanitized by replacing the real last names with surnames drawn from a largish random list.

Substitution is very effective in terms of preserving the look and feel of the existing data. The downside is that a largish store of substitutable information must be maintained for each column to be substituted. For example, to sanitize surnames by substitution, a list of random last names must be available. Then to sanitize telephone numbers, a list of phone numbers must be available. Frequently, the ability to generate known invalid data (phone numbers that will never work) is a nice-to-have feature.

Substitution data can sometimes be very hard to find in large quantities. For example, if a million random street addresses are required, then just obtaining the substitution data can be a major exercise in itself.

Verdict: Substitution is quite powerful, reasonably fast and preserves the look and feel of the data. Finding the required random data to substitute and developing the procedures to accomplish the substitution can be a major effort.

Technique: Shuffling Records

Shuffling is similar to substitution except that the substitution data is derived from the column itself. Essentially the data in a column is randomly moved between rows until there is no longer any reasonable correlation with the remaining information in the row.

There is a certain danger in the shuffling technique. It does not prevent people from asking questions like "*I wonder if so-and-so is on the supplier list?*" In other words, the original data is still present and sometimes meaningful questions can still be asked of it. Another consideration is the algorithm used to shuffle the data. If the shuffling method can be determined, then the data can be easily "unshuffled". For example, if the shuffle algorithm simply ran down the table swapping the column data in between every group of two rows it would not take much work from an interested party to revert things to their unshuffled state.

Shuffling is rarely effective when used on small amounts of data. For example, if there are only 5 rows in a table it probably will not be too difficult to figure out which of the shuffled data really belongs to which row.

On the other hand, if a column of numeric data is shuffled, the sum and average of the column still work out to the same amount. This can sometimes be useful.

Verdict: Shuffle rules are best used on large tables and leave the look and feel of the data intact. They are fast and relatively simple to implement since no new data needs to be found, but great care must be taken to use a sophisticated algorithm to randomise the shuffling of the rows.

Technique: Number Variance

The Number Variance technique is useful on numeric data. Simply put, the algorithm involves modifying each number value in a column by some random percentage of its real value. This technique has the nice advantage of providing a reasonable disguise for the numeric data while still keeping the range and distribution of values in the column within viable limits. For example, a column of sales data might have a random variance of 10% placed on it. Some values would be higher, some lower but all would be not too far from their original range.

Verdict: The number variance technique is occasionally useful and can prevent attempts to correlate true records using known numeric data. This type of Data Sanitization really does need to be used in conjunction with other options though.

Technique: Gibberish Generation

In general, when sanitizing data, one must take great care to remove all imbedded references to the real data. For example, it is pointless to carefully remove real customer names and addresses while still leaving intact in stored copies of correspondence in another table. This is especially true if the original record can be determined via a simple join on a unique key.

Sanitizing “formless” non specific data such as letters, memos and notes is one of the hardest techniques in Data Sanitization. Usually these types of fields are just substituted with a random quantity of equivalently sized gibberish or random words. If real looking data is required, either an elaborate substitution exercise must be undertaken or a few carefully hand built examples must be judiciously substituted to provide some representative samples.

Verdict: Occasionally it is useful to be able to substitute quantities of random text. Gibberish Generation is useful when needed but is not a very widely applicable technique.

Technique: Encryption/Decryption

This technique offers the option of leaving the data in place and visible to those with the appropriate key while remaining effectively useless to anybody without the key.

This would seem to be a very good option – yet, as with all techniques, it has its strengths and weaknesses.

The big plus is that the real data is available to anybody with the key – for example administration personnel might be able to see the personal details on their front end screens but no one else would have this capability. This "optional" visibility is also this techniques biggest weakness. The encryption password only needs to escape once and all of the data is compromised. Of course, you can change the key and regenerate the test instances – but stored or saved copies of the data are immediately available under the old password.

Encryption also destroys the formatting and look and feel of the data. Encrypted data rarely looks meaningful, in fact, it usually looks like binary data. This sometimes leads to NLS character set issues when manipulating encrypted varchar fields. Certain types of encryption impose constraints on the data format as well. For example, the Oracle Obfuscation toolkit requires that all data to be encrypted should have a length which is a multiple of 8 characters. In effect, this means that the fields must be extended with a suitable padding character which must then be stripped off at decryption time.

The strength of the encryption is also an issue. Some encryption is more secure than others. According to the experts, most encryption systems can be broken – it is just a matter of time and effort. In other words, not very much will keep the national security agencies of largish countries from reading your files should they choose to do so. This may not be a big worry if the requirement is to protect proprietary business information.

Verdict: The security is dependent on the strength of the encryption used. It may not be suitable for high security requirements or where the encryption key cannot be secured. Encryption also destroys the look and feel of the sanitized data. The big plus is the selective access it presents.

Summary

Given the legal and business operating environment of today, most test and development databases will require some form of Data Sanitization. There are a variety of techniques available and usually several will be required as the format, size and structure of the data dictates.

One key concern not discussed above is repeatability. When designing the data sanitization routines it should be realized that they will eventually become a production process – even if the data is only destined for test environments. In other words, the data will need to be sanitized each and every time a test database is refreshed from production. This means that data sanitization routines that are easy to run and simple to maintain will soon recover any extra development effort or costs.

There are a variety of complex and subtle issues associated with the successful sanitization of test data which have not been discussed in this paper. For a comprehensive overview of this area please view the companion white paper:

Data Scrambling Issues
http://www.DataMasker.com/datascramblingissues.pdf

