DATA MASKING: AN INTRODUCTION

With the average cost of a data breach exceeding $3.5 million, it's no wonder that safeguarding confidential business and customer information has become more important than ever. Furthermore, with stricter laws and governance requirements, data security is now everyone's responsibility across the entire enterprise.

That being said, an increasing number of organizations are relying on data masking to proactively protect their data, avoid the cost of security breaches and ensure compliance.

What is Data Masking?

To fully explain data masking and its enterprise potential one first needs to understand how it works. As the name implies, data masking, also known as data obfuscation, is a process enterprises use to hide data. Generally, real data is obscured by random characters or other datum. It covers classified data points from those who don't have permission to view it.

Overall, the primary function of masking data is to protect sensitive, private information in situations where it might be visible to someone without clearance to the information. Imagine a scenario where your organization is working with a contractor to build a database. Masking your data allows the contractor to test the database environment without having access to actual sensitive customer information.

Who Uses Data Masking?

In 2018, enterprise businesses are learning they must incorporate data masking into their security strategy especially in light of the General Data Protection Regulation (GDPR) requirements.

If you are reading this, then you are probably aware that GDPR mandates all businesses that accept data from EU citizens to be in compliance with their governance principles by May 2018. For some enterprises, this has resulted in the need to bolster their security strategies by incorporating data masking best practices.

There are many types of data that can be protected using masking, but some commonly used in the business world include the following:

- PII or Personally identifiable information
- PHI or Protected health information
- PCI-DSS or Payment card information
- ITAR or Intellectual property

All of the above examples are subject to compliance with governance principles.

Data Masking and Security

Data masking is useful in a number of security scenarios. Here are a few of the top reasons that enterprise businesses use data masking:

- **To protect data from third-party vendors:** The sharing of some data with third-party marketers, consultants and others is par for course, but certain information must be kept confidential.
- **Operator error:** Enterprises trust their insiders to make good decision, but data breaches are often a result of operator error and businesses can safeguard themselves with data masking.
- **Not all operations require the use of entirely real, accurate data:** There are many functions within an IT department that don't need real data -- for instance, some testing and application use.

Defining data masking means understanding the important role it plays in your company's overall data security strategy.

Types of Data Masking

There a few different types of data masking to be aware of as you consider next steps. Most experts would agree that data masking is static or dynamic, with one exception - on-the-fly data masking. Here's a look at three main types of data masking:

Static Data Masking

Static data masking refers to the process in which important data is masked in the original database environment. The content is duplicated into a test environment, and can then be shared around third-party vendors or other necessary parties.

Data is masked and extracted in the production database and moved into the test database. While

this may be a necessary process for working with third-party consultants, it's not ideal. That's because throughout the process of masking data for a duplicate database, real data is extracted which can leave a backdoor open that encourages breaches.

Dynamic Data Masking

In dynamic data masking, automation and rules allow IT departments to secure data in real-time. That means it never leaves the production database, and as such is less susceptible to threats.

Data is never exposed to those who access the database because the contents are jumbled in real-time, making the contents inauthentic.

A resource called a dynamic masking tool finds and masks certain types of sensitive data using a reverse proxy. Only authorized users will be able to see the authentic data.

Concerns from dynamic data masking mostly stem from database performance. In an enterprise environment, time is money and even milliseconds have value. In addition to time considerations of running such a proxy, whether or not the proxy itself is secure can be a cause for concern.

On-the-fly data masking

Similar to dynamic data masking, on-the-fly data masking occurs on demand. In this type of data masking, an Extract Transform Load (ETL) process occurs where data is masked within the memory of a given database application. This is particularly useful for agile companies focused on continuous delivery.

Overall, your selection of a data masking strategy must take into consideration the size of the organization, as well as the location (cloud v. on premise) and complexity of the data you wish to protect.

Common Data Masking Techniques

There are a number of techniques that IT professionals can use when masking data. Here's a list of data masking techniques and how they apply to your business:

Encryption

When data is encrypted, authorized users must access it with a key. This is the most complex and secure type of data masking. Here, data is masked via an encryption algorithm.

Character Scrambling

A very basic masking technique is character scrambling. Using this approach, characters are jumbled into a random order so the original content is not revealed. For instance, using character scrambling, an employee who's badge number is #458912 in a production set of data, may read #298514 in the test environment.

Nulling Out or Deletion

Like the name would suggest, when this approach is applied, data becomes null to anyone who isn't

authorized to access it.

Number and Date Variance

When properly executed, number and date variance can provide you with a useful set of data without giving up important financial information or transaction dates. For instance, a data set that offers employee salaries can give you the range in salary between highest and lowest paid employee when masked. You can ensure accuracy by applying the same variance to all salaries in the set, that way they range doesn't change.

Substitution

Substitution effectively mimics the look and feel of real data without compromising anyone's personal information. With this approach a value that looks to be authentic is substituted for the actual value. This effectively hides authentic data, protecting it from breach threats.

Shuffling

Similar to substitution, shuffling uses one data set in place of another. But in shuffling, the data in an individual column is shuffled in a randomized fashion. The output set looks like authentic data but doesn't reveal any real personal information.

Data Masking Best Practices

When it comes to your organization's processes, you want to learn from the best. Below are best practices for creating a strategy that works for data masking within your organization:

- **Find data**: This first step involves identifying and cataloging the various types of data that may be sensitive. This is often carried out by business or security analysts who put together a comprehensive listing of enterprise-wide data elements.
- **Assess the situation**: This phase requires oversight from the security administrator who is responsible for determining if sensitive information is present, the location of the data and the ideal data masking technique.
- **Implement masking**: Remember that for very large organizations, it isn't feasible to assume that a single data masking tool can be used across the entire enterprise. Instead, implementation must take into account architecture, proper planning and a look to future enterprise needs.
- **Test data masking results**: This the final step in the data masking process. QA and testing are required to ensure the masking configurations yield the desired results. If they do not, then the DBA will restore the database to the premasked state, tweaks the masking algorithms and completes the data masking process once more.