

HOW TO PROTECT DATA FROM NIMBLE-FINGERED STAFF

Protection of company data against malicious IT staff

In the age of data digitisation, the threat of data being misused by your own employees should not be treated lightly: the consequences are incalculable and potentially devastating. Even the latest technology cannot provide complete protection against data misuse. There are, however, two possible approaches to data encryption that can mitigate this risk.

Reto FANKHAUSER*

In recent times there has been an increase in incidents involving theft of critical information, such as bank account details, and sale of the information to criminals or authorities. Such incidents have received a lot of media attention and there is now a general level of awareness that banking data can be a valuable target for theft. Data theft by employees is not restricted to the world of finance, however, and a common feature of most of the incidents uncovered recently is that the information was stolen by internal IT staff.

KPMG's Data Loss Barometer¹ explains the potential risk to an organisation from its employees as follows: "A combination of economic pressure and temptation in the form of offers from criminal organisations let certain employees perceive theft as a viable option." The motivation for data theft may be financial gain, competitive advantage or even sabotage.

IT employees as a potential risk

Depending on their function, IT employees have privileged access to the various IT systems within an organisation. Two roles are of particular relevance when assessing the potential risk associated with IT operations. For database administrators (DBAs): As the managers of the company's data-



Reto FANKHAUSER, Senior Architect and Security Expert, ELCA Informatik AG

bases, they have access to the data stored in the databases for which they are responsible. For system administrators: As they install (and modify) applications, they have access to the configuration parameters of the applications, including credentials for technical database accounts. In addition, they are often able to eavesdrop on network communications (in the same way that network administrators can).

While physical "on paper" data used to be stored in secure, lockable data cabinets, ensuring the security of digitised data raises complex issues and although it is impossible to eliminate the risk posed by IT

employees entirely, it can be reduced. Below, we take a closer look at two procedures to encrypt data that is stored in a structured way, with an emphasis on potential misuse of this data by IT employees.

Encrypting data in the database

Almost all popular storage solutions (including databases) offer the possibility to encrypt the data at rest. Manufacturers highlight the fact that encryption can be performed without the need to adapt existing applications. The data model can be retained, and the encryption is transparent to peripheral systems. Vendors promise conformity with current legislation and policies on data protection at the touch of a button. While hackers may still be able to get hold of a "raw" database file, they will find it worthless because they are unable to decrypt it.

No such built-in mechanism, however, offers adequate protection against misuse by IT employees, who are able to control and manipulate mechanisms such as transparent data encryption. A DBA, for example, possesses the necessary authorisations and tools to decrypt all of the data. System administrators also pose a threat, as they are often authorised to install all kinds of applications, including database tools. All a malicious system administrator requires are credentials as a legitimate database account in order to see the unencrypted

¹ <http://www.datalossbarometer.com/>

Encryption in the database	Encryption in the application
<p>Advantages</p> <ul style="list-style-type: none"> - The applications need not be modified - Data model need not be modified - Functionality provided out of the box by many databases <p>Disadvantages</p> <ul style="list-style-type: none"> - Data outside the database is not protected - Standard: no separation of data and keys (keys in the database), a hardware security model is normally required for such separation - Additional database overhead (reduces performance) - Limited support for encryption algorithms 	<ul style="list-style-type: none"> - Allows improved separation of administrator roles - Encrypted data and keys are stored separately - Data is also encrypted outside the database: transfer, migration and archiving are simpler - Database has no encryption overhead - Scalability: encryption infrastructure can assume the work for multiple applications / databases - Any desired encryption algorithms can be used - No vendor lock-in: data can be migrated easily from one database system to another, irrespective of whether the database system supports encryption <p>Disadvantages</p> <ul style="list-style-type: none"> - Additional communications between the systems - Encryption server requires additional administration - Data models need to be modified - Applications need to be modified

The two encryption approaches compared: advantages and disadvantages

data. Usually a person in this role has access to configuration files from server applications, which need to connect with the database. Since login information is stored within a configuration file, obtaining valid credentials is an easy task.

Further analysis also reveals the technical limits of such “transparent” encryption. Depending on the product, only a limited set of encryption algorithms are available and it is not always possible to encrypt individual tables or columns selectively: some products encrypt all or nothing. If everything is encrypted, all potentially matching data must be decrypted for each query. Given that encryption demands additional resources, it is not practical to encrypt every piece of data, so selective encryption of individual data fields is essential. An additional drawback to this approach is that data is no longer encrypted when it leaves the database, so it is also necessary to encrypt data in transit, e.g. on the network or on portable storage media.

Given these facts, “compliance at the touch of a button” is revealed as an impractical ideal. Yet the need to deal with the peculiarities of individual data and applications cannot be circumvented.

What about application-level encryption?

Data can also be encrypted at the application level before it is written to the database, and decrypted after it is retrieved from the database. The encryption may take place in the application itself, or preferably

the application delegates this task to a “central encryption service.” A centralised service is available for all applications and it scales better as needs increase.

If the data is decrypted on the application layer, it is not necessary to take additional measures to secure the data in transit between the database and the application, as records are transmitted in an encrypted state. Thus, in the event of any sniffing of data traffic on the network, sensitive data remains unreadable. It also means that data can be migrated and transported easily between systems without the need for decryption and subsequent re-encryption. In contrast to the “transparent” in-database encryption approach, the data remains encrypted during the entire migration process.

Compared with database encryption, application-level encryption offers the advantage of making data less susceptible to theft, especially when the potential culprits are employees. Neither DBAs nor system administrators are able to view the data in plain text and even if they have access to a legitimate database account and run queries against the database, all they will see are encrypted records.

This approach is more labour-intensive than database encryption and it influences the data model since current standard encryption algorithms do not retain the input data type. The data model must therefore use the resulting data type for any encrypted columns, rather than the original. The applications must also be adopted so that encryption and decryption are incorporated. Addi-

tional resources are required for ciphering, but this occurs on the application level rather than the database level, the better to suit scalability needs.

There is no silver bullet!

There is currently no magic armour that can protect a company’s data at the flick of a switch against the danger of internal or external data theft. Both of the systems that we have examined offer various advantages and disadvantages.

The database approach requires less effort to implement, but an “all or nothing” strategy carries disadvantages in terms of performance. Data is still vulnerable to disclosure by IT administrators and additional measures are required for securing data in transit.

The application level approach cannot be implemented without a detailed understanding of the data and applications involved and its introduction also affects the data models. A significant advantage to this approach is that it mitigates the risk of data theft by IT employees effectively and it also removes the need for securing data during migrations.

Selecting the appropriate approach for a particular situation requires analysing the data and systems involved, and assessing the effective threats that are present. This minimises the risk of any unpleasant surprises in terms of performance, cost, and level of security. ■ R.F.

*Senior Architect and Security Expert, ELCA Informatik AG (Zurich)