1

Securing and protecting IBM DB2 data

Recent news headlines are filled with reports of data breaches and cyber-attacks impacting global businesses of all sizes. The Identity Theft Resource Center[1] reports that almost 5000 data breaches have occurred since 2005, exposing over 600 million records of data. The financial cost of these data breaches is skyrocketing. Studies from the Ponemon Institute[2] revealed that the average cost of a data breach increased in 2013 by 15% globally and resulted in a brand equity loss of $9.4 million per attack. The average cost that is incurred for each lost record containing sensitive information increased more than 9% to $145 per record.

Businesses must make a serious effort to secure their data and recognize that securing information assets is a cost of doing business. In many parts of the world and in many industries, securing the data is required by law and subject to audits. Data security is no longer an option; it is a requirement.

This chapter describes how you can secure and protect data in DB2 for i. The following topics are covered in this chapter:

- Security fundamentals
- Current state of IBM i security
- DB2 for i security controls

[1] http://www.idtheftcenter.org
[2] http://www.ponemon.org/

1.1 Security fundamentals

Before reviewing database security techniques, there are two fundamental steps in securing information assets that must be described:

- First, and most important, is the definition of a company's *security policy*. Without a security policy, there is no definition of what are acceptable practices for using, accessing, and storing information by who, what, when, where, and how. A security policy should minimally address three things: confidentiality, integrity, and availability.

  The monitoring and assessment of adherence to the security policy determines whether your security strategy is working. Often, IBM security consultants are asked to perform security assessments for companies without regard to the security policy. Although these assessments can be useful for observing how the system is defined and how data is being accessed, they cannot determine the level of security without a security policy. Without a security policy, it really is not an assessment as much as it is a baseline for monitoring the changes in the security settings that are captured.

  A security policy is what defines whether the system and its settings are secure (or not).

- The second fundamental in securing data assets is the use of *resource security*. If implemented properly, resource security prevents data breaches from both internal and external intrusions. Resource security controls are closely tied to the part of the security policy that defines who should have access to what information resources. A hacker might be good enough to get through your company firewalls and sift his way through to your system, but if they do not have explicit access to your database, the hacker cannot compromise your information assets.

With your eyes now open to the importance of securing information assets, the rest of this chapter reviews the methods that are available for securing database resources on IBM i.

1.2 Current state of IBM i security

Because of the inherently secure nature of IBM i, many clients rely on the default system settings to protect their business data that is stored in DB2 for i. In most cases, this means no data protection because the default setting for the Create default public authority (QCRTAUT) system value is *CHANGE.

Even more disturbing is that many IBM i clients remain in this state, despite the news headlines and the significant costs that are involved with databases being compromised. This default security configuration makes it quite challenging to implement basic security policies. A tighter implementation is required if you really want to protect one of your company's most valuable assets, which is the data.

Traditionally, IBM i applications have employed menu-based security to counteract this default configuration that gives all users access to the data. The theory is that data is protected by the menu options controlling what database operations that the user can perform. This approach is ineffective, even if the user profile is restricted from running interactive commands. The reason is that in today's connected world there are a multitude of interfaces into the system, from web browsers to PC clients, that bypass application menus. If there are no object-level controls, users of these newer interfaces have an open door to your data.

Some clients using this default configuration have toughened their database security with exit-point solutions from third-party vendors. IBM i exit points allow a user-written program to be called every time that a particular interface (for example, FTP) is used or an event occurs (for example, a profile is created). Security tools that are based on these exit points increase the level of security on a system by locking down interfaces that are not under the control of menu-based or application authority. In addition, exit-point solutions allow clients to implement more granular security controls, such as allowing users access only to the database during certain hours of the day.

Although exit-point solutions can provide great benefits, they are not an alternative to object-level control of your databases. Exit-point solutions help secure interfaces, but they do not completely protect the data that is stored in your DB2 objects. Exit points do not exist for every data access interface on the system. Therefore, if an application starts using an unprotected interface, the only thing protecting your data is object-level access control. When your security implementation totally relies on exit points, then it is also important to track any new data interfaces that appear as IBM delivers new releases and products to ensure that your exit-point solution provides coverage for those new interfaces.

An exit-point solution is a good option for databases with security holes that are caused by a reliance on the default security setup or menu-based control. However, your security work should not stop there. Instead, you must continue to work on a complete database security solution by controlling data access at the object level.

1.3 DB2 for i security controls

As described in 1.2, “Current state of IBM i security” on page 2, object-level controls on your DB2 objects are a critical success factor in securing your business data. Although database object-level security is a strong security feature, some clients have found that object-level security does not have the granularity that is required to adhere to regulatory or compliance policies. A user that is granted object-level access to a DB2 table has the authority to view all of the rows and values in that table.

As shown in Figure 1-1, it is an all-or-nothing access to the rows of a table.

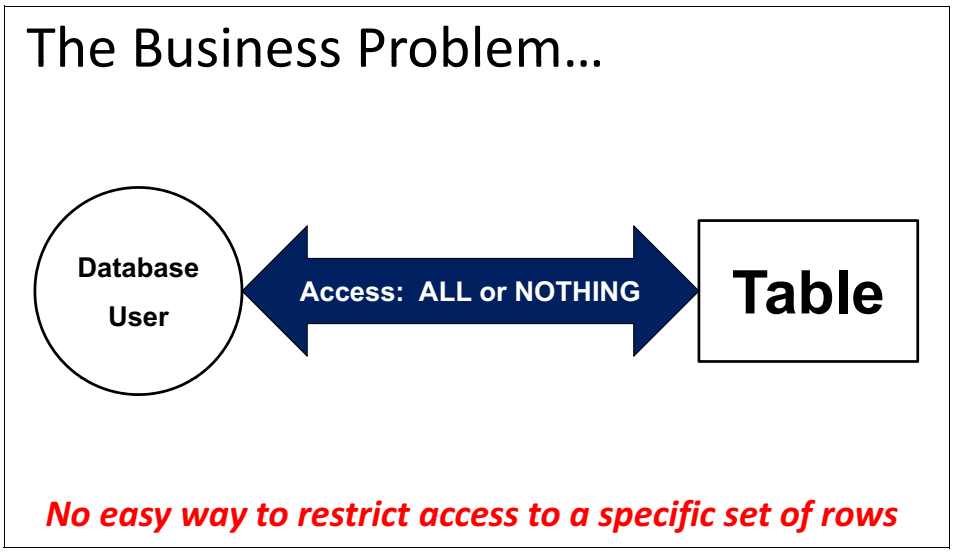

Figure 1-1 All-or-nothing access to the rows of a table

Many businesses are trying to limit data access to a need-to-know basis. This security goal means that users should be given access only to the minimum set of data that is required to perform their job. Often, users with object-level access are given access to row and column values that are beyond what their business task requires because that object-level security provides an all-or-nothing solution. For example, object-level controls allow a manager to access data about all employees. Most security policies limit a manager to accessing data only for the employees that they manage.

1.3.1 Existing row and column control

Some IBM i clients have tried augmenting the all-or-nothing object-level security with SQL views (or logical files) and application logic, as shown in Figure 1-2. However, application-based logic is easy to bypass with all of the different data access interfaces that are provided by the IBM i operating system, such as Open Database Connectivity (ODBC) and System i Navigator.

Using SQL views to limit access to a subset of the data in a table also has its own set of challenges. First, there is the complexity of managing all of the SQL view objects that are used for securing data access. Second, scaling a view-based security solution can be difficult as the amount of data grows and the number of users increases.

Even if you are willing to live with these performance and management issues, a user with *ALLOBJ access still can directly access all of the data in the underlying DB2 table and easily bypass the security controls that are built into an SQL view.

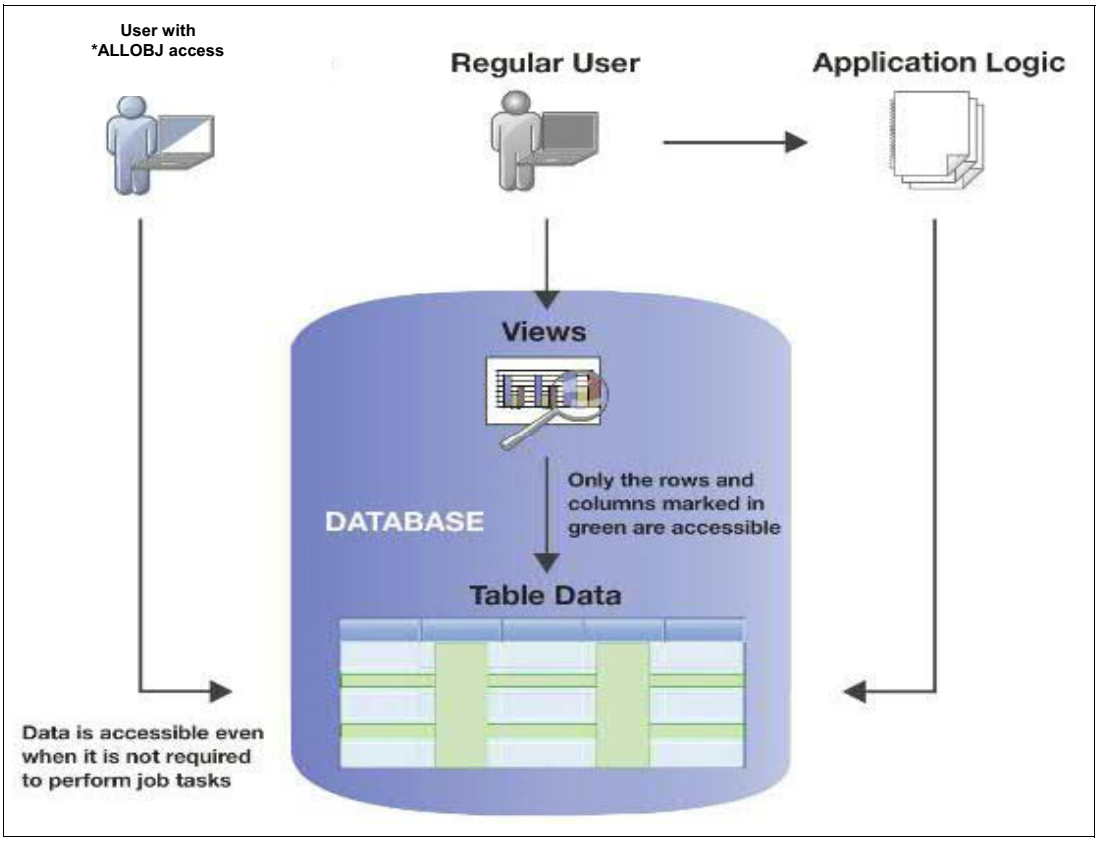

Figure 1-2 Existing row and column controls

1.3.2 New controls: Row and Column Access Control

Based on the challenges that are associated with the existing technology available for controlling row and column access at a more granular level, IBM delivered new security support in the IBM i 7.2 release; this support is known as Row and Column Access Control (RCAC).

The new DB2 RCAC support provides a method for controlling data access across all interfaces and all types of users with a data-centric solution. Moving security processing to the database layer makes it easier to build controls that meet your compliance policies. The RCAC support provides an additional layer of security that complements object-level authorizations to limit data access to a need-to-know basis. Therefore, it is critical that you first have a sound object-level security implementation in place.