



Issue Brief:

Managing Need-To-Know

The problem of controlling internal access

The embarrassing disclosure¹ that medical records for Britain's Prime Minister Gordon Brown and a host of other Scottish luminaries had been illegally accessed is, notwithstanding the political fallout being faced by the UK's National Health Service (NHS), simply another example of the failure of an IT security design to effectively control internal access to sensitive information residing in a widely used database.

In this instance, the database involved is the Emergency Care Summary system which contains both detailed medical and other Personally Identifiable Information on approximately 2.5 million Britons. When NHS established the system three years ago, it guaranteed both the "highest standards of security" and the adoption of strict protocols requiring NHS staff to gain patient consent before accessing their records in non-emergency situations. Despite the fact that an NHS doctor has been charged with violating the Data Protection Act, at this point it appears that the motive for the breach may simply have been curiosity about the public figures rather than financial gain.

The importance of this specific incident – which in the scheme of things appears relatively harmless – is that it provides a high-profile example of a much more serious problem faced by any Enterprise that maintains sensitive information, namely, how to implement internal need-to-know restrictions over specific data components that are resident in broadly accessible data structures.

¹ Dan Goodin, "Prime Minister's Health Records Breached in Database Attack", The Register, March 2, 2009, <http://www.theregister.co.uk/2009/03/02nhs_database_breach.html>



Designing data security around access control

IT security frameworks are overwhelmingly (and appropriately) focused on external threats. These frameworks frequently consist of multiple components: perimeter systems to prevent unauthorized penetration, data encryption algorithms to veil actual content if it is stolen, anti-virus software to protect against vandalism, and various systems intended to send out alerts after a security violation occurs.

While all these security components can be effective deterrents to threats from the outside, none are particularly well designed to address *internal* security issues related to proactively controlling access to sensitive information.

The most effective approach to controlling internal access to sensitive data is, not surprisingly, the blunt instrument of hard coding. Access control is specified through attributes built into the data itself which, in turn, must be matched with attributes of internal users seeking access. While such a rigidly restrictive approach may be appropriate in highly classified environments, the isolated data islands that result are a poor match with the requirements of day-to-day operations for most Enterprises.

For these Enterprises, the underlying challenge in designing effective internal access security is simple and unavoidable: in order to have any value to the Enterprise, data systems must be broadly accessible to internal users. In most instances, the only reason to have the data in the first place is to enable internal users to leverage its value through various Enterprise applications.

Accepting the reality that a large population of internal users must be able to access Enterprise data structures, security issues arise when these broadly accessible repositories contain sensitive data that requires special protection, i.e., security beyond that required by the repository in general. In this regard, the sensitive data will generally fall into definable groups:

- Entire categories of routinely maintained information that are considered private and confidential. The most familiar examples relate to databases that contain personally identifiable information (PII) such as Tax IDs and Dates of Birth.
- Specific records in their entirety that are considered private and confidential, e.g., the health record of Prime Minister Brown.
- Specific data elements in specific identifiable records, e.g., the Prime Minister's Tax ID and mobile phone number.

The security challenge, therefore, becomes designing a solution that is flexible enough to protect *specific components within a data structure* – i.e., fine-grained access control.

Implementing fine-grained access control

The central objective of internal access security is to provide users with access to data *they need to know*, while preventing access to data *they don't need to know*. Given that Enterprise data structures frequently co-mingle sensitive and non-sensitive data, achieving this objective involves a much greater “fine-grained” level of access control: i.e., control over access must be extended to specific data elements inside the data structure itself.



For example, a typical Enterprise database may be secured with controls that limit access to a specified internal group of users. Based on this, internal users will or will not have access to the database depending on their membership in the authorized user group.

The challenge arises when the Enterprise seeks to further limit access to sensitive components *within* the database to those users with a need-to-know. As touched on above, this may involve limiting access to specific categories of data (components such as tables or fields), specific records (actual data elements), or some combination of both. To achieve this objective, fine-grained access control is required.

Access control at the Metadata level

The first step in implementing fine-grained access control is to establish a description of the data structure and its organization, i.e., a Metadata Blueprint. This Metadata Blueprint can encompass a single data structure, as in the database example above or, more realistically within the context of a real-world Enterprise, can be expanded to include all data assets in the Enterprise architecture.

As the Metadata Blueprint is developed, an access control list (ACL) is associated with each of the individual Metadata components. These ACLs specify the user groups with access rights *to that individual component*. Through this process, therefore, fine-grained, component-level access control is extended across all the data structures represented in the Metadata Blueprint.

Access management becomes a process of, first, specifying the internal users in particular user groups and, second, specifying which user groups have access privileges to each Metadata component. Returning to the database example, therefore, significantly greater internal access security can be implemented *without modifying the existing data structure*. Access to sensitive categories of data (e.g., Tax IDs and DOBs) is determined at the Metadata level and can thus be restricted to internal users with a need-to-know such information.

Access Control at the Data Element level

In certain circumstances, such as the incident involving Prime Minister Brown and the other Scottish nabobs, internal access security must be extended over specific high-value data elements. In such instances, fine-grained access control can be achieved through a process designed to separately secure each of the individual high-value data elements through creation of discrete object-level access control.

The first step in this process is to physically remove the high-value data elements from their typical location in the data structure and safeguard them as discrete data element objects in a separate, secure environment. As these data element objects are created in the separate environment, a discrete ACL is associated with each object.

Subsequently, access to these high-value data elements is authorized or denied based on an internal user's membership in a user group specified on the ACL. Once again, access management becomes a process of, first, specifying the internal users in particular user groups and, second, specifying which user groups have access privileges to each discrete data element.

Employing such a process enables fine-grained access control to be implemented over essentially any and all sensitive data. For example, all information related to "well-known or



public figures” such Prime Minister Brown could be removed and separately secured, as could all Tax IDs and DOBs from a customer database or all PII from a traveler’s passport record.

Managing need-to-know within an Enterprise is nothing more than effectively managing internal access to data. Fine-grained access control processes – both at the Metadata and data element level – provide the necessary tools to enable an Enterprise to comprehensively protect its sensitive data assets.

About Eaglehawk Limited

Eaglehawk Limited is a pioneer in advanced enterprise-level data security technology. Eaglehawk’s patented flagship product, SBX Enigma™ (SBX), is a customer-hosted Web Service that provides a secure repository for any Enterprise seeking to manage access to and protect its data.

Accessible to Client Applications through the SBX Application Program Interface (API), SBX is an in-memory, object-oriented data management system. SBX provides advanced security functionality that enables complex Enterprise data environments to centralize and streamline authentication, access management, and audit processes. Key features include:

User Administration Functions

- Comprehensive management over all aspects of user privilege
- Internal security via overlapping, failsafe administrative role structure

Metadata Functions

- Accommodates enterprise-wide data access across multiple systems and types
- Provides fine-grained access control down to individual components within the enterprise structure regardless of data location or type

Data Functions

- High speed in-memory object store securing high-value data elements
- Fine-grained need-to-know access control over individual data elements

Centralized Audit Functions

- Comprehensive, mandatory audit of administrative activities
- Flexible, optional audit of client application/service interactions with SBX
- Generic, as-required audit of *any* independent client application/service event

For additional details, please contact:

William Tice
Eaglehawk Limited
Georgetown, MA

Tel: (978)749-9946
william.tice@eaglehawk.com