Data-centric security

Jennifer Bayuk, independent security consultant and ICASA lecturer

The authoritative control objectives for access to data have always been something along the lines of: ***"Confirm that user access rights to systems and data are in line with defined and documented business needs, and that job requirements are attached to user identities….Ensure that critical and confidential information is withheld from those who should not have access to it."*** [1]

This standard was relatively simple to follow when all enterprise data was stored on a mainframe. However, with each emerging distributed technology and corresponding threat, this standard has been getting ever more difficult to achieve in practice.[2]

The recognition that data is increasingly vulnerable to a wide variety of attacks has led information security professionals to adopt a military strategy for protecting it called 'defence in depth'.

Defence in depth is illustrated in Figure 1. The idea is that data should be secured by enclosing it in layers of security. Protection mechanisms at each layer restrict access in different ways.

"But even when all the layers are in place, some of these architectures are more secure than others"

The first layer is the user desktop. Once users authenticate to a desktop, they must then be able to reach the network where the data resides. On the network, there will be a server that has operating system controls that protect applications. Applications have their own access controls. Once authenticated to an application, permissions or entitlements within that application would determine whether a user may be granted access to data via controlled processes like database management systems.

The basic idea is sound and thousands of such architectures have been developed in the past few decades. But even when all the layers are in place, some of these architectures are more secure than others.

A tale of two access paths

Figure 2 compares the access paths used by two applications, A and B. Both applications require desktop authentication at the user level and provide direct access from user workstations to the network via firewall rules. But application A allows direct authentication to the application from the network, whereas application B requires the user to log into an operating system on a server prior to authenticating to the application layer.

Taken as a pure defence-in-depth exercise, it would seem that the extra authentication layer makes application B more secure. However, any security architect will tell you that it is actually application A that has the more secure architecture. This is because application B provides the potential for the user to directly access the server, which contains the software and database access used by the application.

Authentication to the server potentially allows the user access to view files and initiate processes at the operating system level that may gain access to data while bypassing user application entitlements. Where a user must access the application directly from the network, as in the path taken by application A, the user is prevented from bypassing the entitlements built into the application.

In addition to the data access paths taken by applications A and B, Figure 2

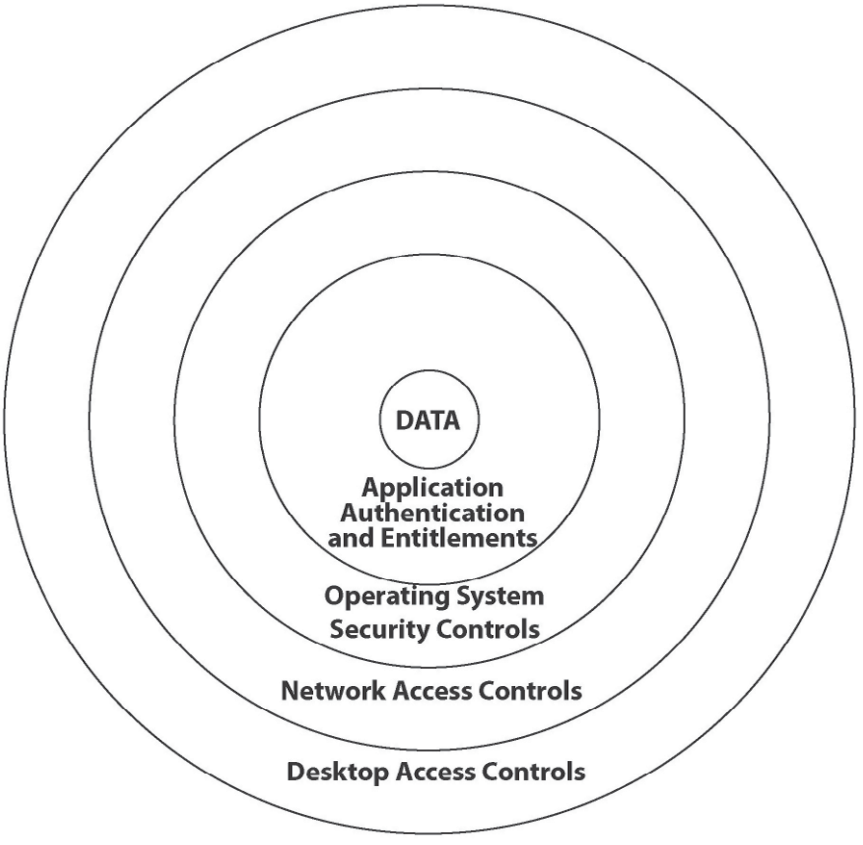

Figure 1: Layers associated with defence in depth

contains a third path, labelled 'Remote Access'. Many organisations place great emphasis on ensuring that it is possible to provide defences for data in at least the five levels depicted in the diagram. Nevertheless, many access paths reach data from virtually unknowable sources. This can happen via authentication from the internet, and firewall rules that allow authenticated remote users the same network access through firewalls as insiders.

For example, it is common for network access paths to be open directly to data, ostensibly for use by administrators, on the grounds that those who maintain data must have access to it. This extends from databases to operating systems, where access to the configuration of the database allows data to be within reach as well. It also often falls within the job responsibilities of application support departments to have direct access to application data. Outsourcing presents yet another significant potentially justified job function scenario.

Weaknesses in network-level access

In practice, such administrative access paths are left open at the network and not the user level. This reduces all the layers of defence in depth to one or two. Even where encryption is employed at the disk or electronic media level, administrators store and manage the keys, and any authorised access uses mechanisms that automatically retrieve the decryption keys to make the decryption process seamless to the end user.

Given that a user at one level is not restricted from presenting credentials of another user at the next level, it is common to see hacking attempts within the internal network on paths that are not required to be open by any application. There are also situations in which applications themselves make use of administrative-level logins for

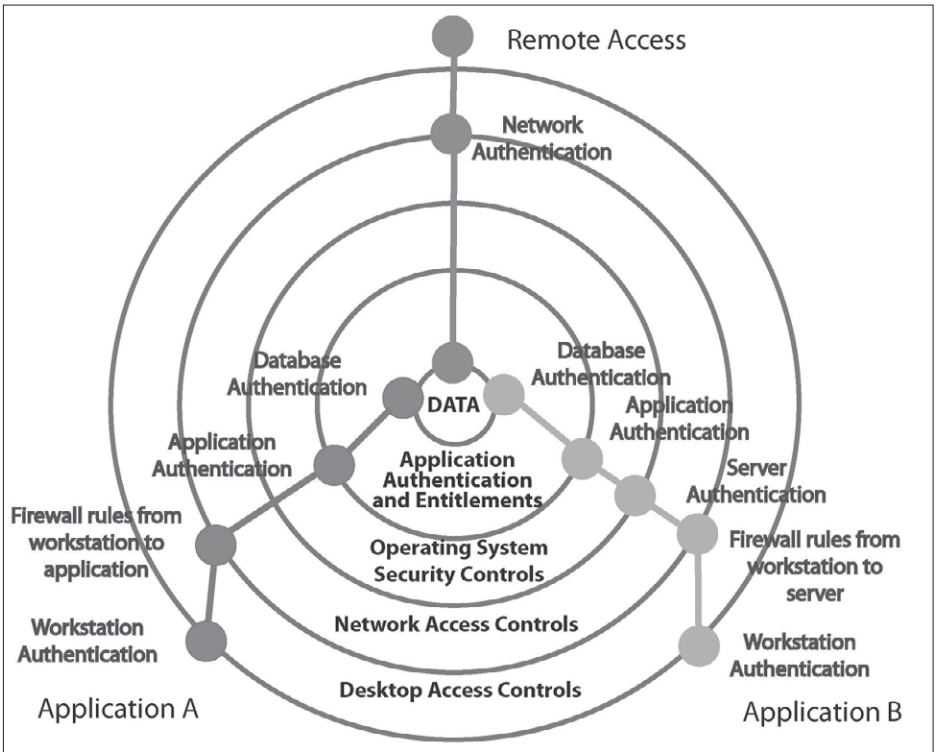

Figure 2: Alternate data access paths

data access, and these are sometimes not safeguarded as well as they should be. Passwords don't change, and people who leave the firm retain knowledge of the password.

Moreover, even non-administrative users may be authorised to have direct data access. Power users somehow manage to implement authorised business processes in which they personally extract data from databases and reformat it for delivery within and outside the organisation. Their activities are often beyond the business's IT department's ability to control. Where these power users have the authority to send data to third parties via insecure methods, just allowing them to have data they need to do their jobs creates security vulnerabilities and regulatory compliance issues.

"Data-centric security starts with a hard look at what data the business must protect and why, or an exercise in information classification"

This situation has resulted in a back-to-basics approach to information security; a data-centric approach. Data-centric security starts with a hard look at what data the business must protect and why, or an exercise in information classification. It creates holistic business data control requirements. Project managers are assigned to approach these requirements as they would any other type of business requirement. The result is a business analysis of data usage that results in data handling requirements. These in turn result in security technology requirements.

Separation of data

The data-centric approach has been a subject of research for over two years, and many cite its inception in a seminal IBM paper: Data-centric security, Enabling business objectives to drive security.[3] The authors wrote: "We propose to link security services directly to business processes by relating security services directly to the data they implicitly protect — a relationship that is often obscured by the presentation of security as an end in itself." It also advises that infrastructure zones

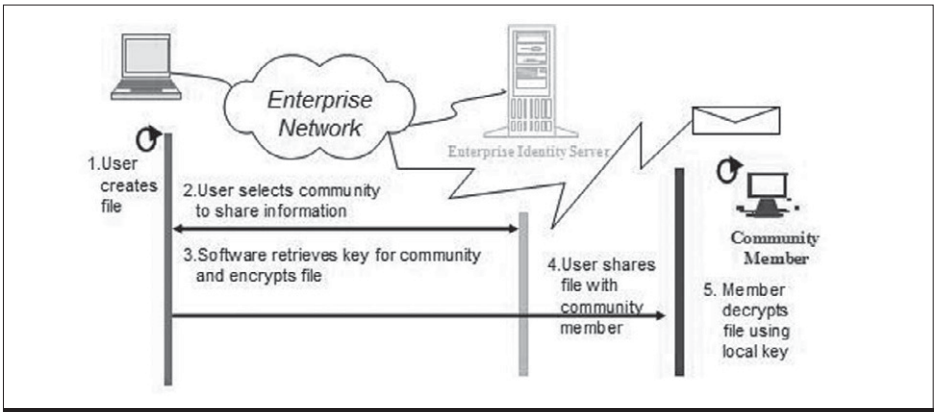

Figure 3: Discretionary digital rights management

be created to further minimise access at the network level.

Data-centric security requirements also dictate that businesses separate data not just at the information classification, but also at the use case level. For example, the Payment Card Industry Security Standards require identity data to be encrypted when it is combined with the account number, but not if it appears by itself.[4]

Often, a data-centric business analysis results in requirements to allow application access to data but to lock out as many IT support teams as possible. Solutions to this issue require technology that will maintain classification labels and prevent even authorised users from sharing data in an unauthorised manner. The only available technologies that exist to implement this approach involve encryption at the data rather than the disk level (and indeed, the PCI standard specifies data field level encryption).

Although many vendors offer pseudo-solutions that achieve much less, to truly practice data-centric security would be to adopt enterprise-wide methods of separating data from infrastructure, labelling all of it, and maintaining the segregation between authorised use cases, and unauthorised use cases.

However, practitioners are constrained by the type of solutions currently on the market. These may be classified generally into two categories: File and Field. Within these two categories are at least two implementation varieties. For example:

Within a data-centric File approach there are:

- Discretionary digital rights management (DDRM)
- Mandatory digital rights management (MDRM) or network tethering

Within a data-centric Field approach there are:

- Database level encryption
- Application field level encryption

Each of these approaches is illustrated with a figure and a description of the pros and cons with respect to implementing a data-centric security model. The figures are in the form of sequence diagrams. They show the points in the network that the data touches as columns. The steps in the process of user data access are numbered with a short description of what technically occurs in each step. The sequences illustrate how the technology works to implement the approach.

Discretionary digital rights management

DDRM is a technology wherein files are encrypted and assigned access control lists. As illustrated in Figure 3, at the time a user creates a file, the user must specify which members of the community with whom the file may be shared. These are typically predefined groups, but may also be individuals.

Implementation of this approach requires a server that stores encryption keys for all the users who are members of the community. When the file is created, it is encrypted in such a way that only the members specified upon creation can decrypt the file. Usually the implementation allows for an administrator to decrypt all files in case any given user has trouble with the software and cannot retrieve the encrypted data. The administrative key is called the 'additional decryption key' or 'corporate decryption key' and it is automatically added to the access list for every file.

The pros of this approach include the fact that it allows a group of users to collaborate in securing a shared set of highly sensitive files. It allows mobile users to access information remotely with minimal exposure to theft, and the file level encryption technology that it uses is mature.[5]

However, there are some downsides. It lets users decide when to encrypt information, so it is possible that some information which should be encrypted is either intentionally or accidentally not actually encrypted. Mobile users may also copy information to unencrypted media, such as laptops that are connected to the internet, or hand-held USB devices.

Finally, administrators have super-decryption keys, and often also administer group membership, so data access is not necessarily restricted to the select user community.

Network tethering

Network tethering is also a file level technology. It is referred to as mandatory digital rights management because, once access rights are set by an administrator, a user does not have the discretion to share or save the file in a way that anyone not authorised to see it will be able to.

As depicted in step 3 of Figure 4, this method requires specialised software that allows encryption keys stored on the server to be opened

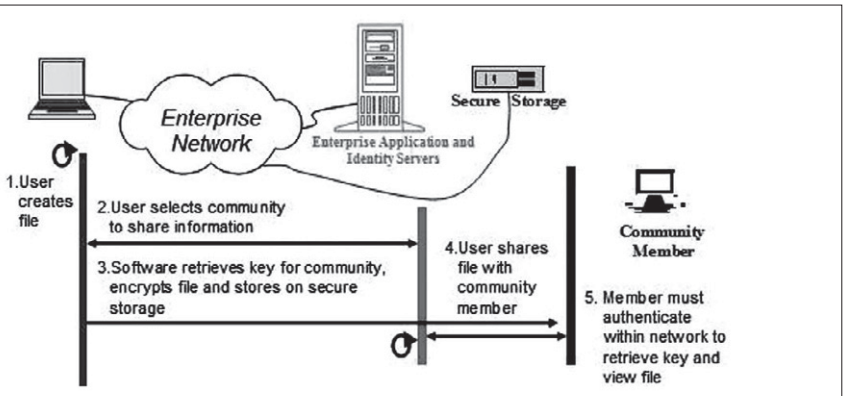

Figure 4: Network tethering

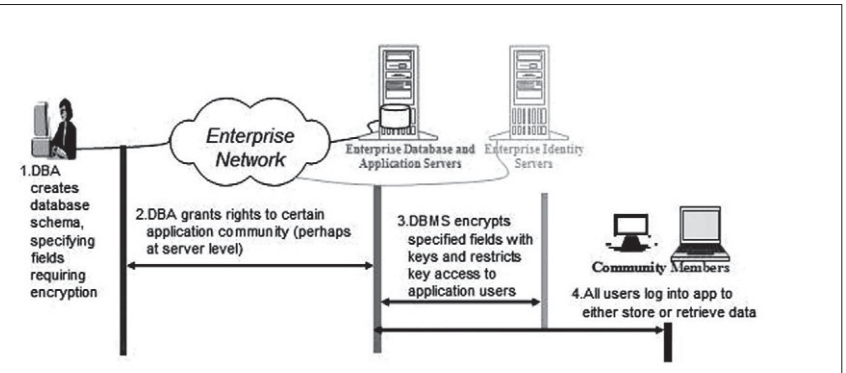

Figure 5: Database level encryption

only when the user is connected to the network.

The pros of this approach are that it allows a department authority to specify with which groups of individuals certain types of information should be shared. Users cannot arbitrarily add individuals to access lists, and it allows mobile users to access information remotely while data never leaves the network. Used correctly, it does not expose data to internet threats. And, as with DDRM, the file-level encryption technology that it relies on is mature.

This approach suffers from some of the same downsides as DDRM. Administrators still have super-decryption keys, for example. However, the specialised software may make it very difficult for mobile users to copy information to unencrypted media, making it superior to a DDRM approach. But screen shots must necessarily be exposed, and cut-and-paste features are hard to disable, especially given the unpredictable variety of remote devices that are allowed to be on the network.

Database level

Database level is a field level technology implemented by database management systems. As illustrated in Figure 5, off-the-shelf functionality within the DBMS allows a database administrator to specify which database fields should be encrypted and which sets of users can access those fields.

Keys are managed within the database itself and database utilities allow keys to be changed. Users may be part of the groups of DBMS users that access the data or they can be restricted to the application by having one application login accessible only after users authenticate to the application via other means.

This approach carries several benefits. User access to data can be restricted to application functionality by giving the application the only database login with database decryption capability. For example, applications can restrict the amount of data a user can decrypt with a single operation to prevent users from copying whole files or unencrypted data in bulk.

Database-level access also allows a department authority to specify among which groups of individuals certain types of information should be shared, and users cannot arbitrarily add individuals to access lists. And correct implementation does not rely on correct user behaviour or application code.

However, database administrators and application support staff still have keys to the kingdom (though their access may be audited). And anyone with direct DBMS login access that is in a group with access to the keys may still download data in bulk. Given the overhead of user-level audit on DBMS queries, it is not likely that the access would be audited.

"Although they do provide some protection when used in conjunction with a secure application architecture, defence-in-depth strategies have failed the test of time in securing data in an internet-connected, ubiquitous file-sharing world"

Finally, when encrypting database fields that are used by multiple applications, reports that include the data may be difficult to generate because select queries that rely on matching data across tables may be difficult if not impossible.

Application field level

Application field level encryption is illustrated in Figure 6. In this approach, an application developer designs an application programming interface (API) that is the only method that data can be entered or retrieved from an encrypted data store. The encryption API uses keys that are not accessible to the DBMS administrator, and are accessible to the application only in the production run-time environment. The database

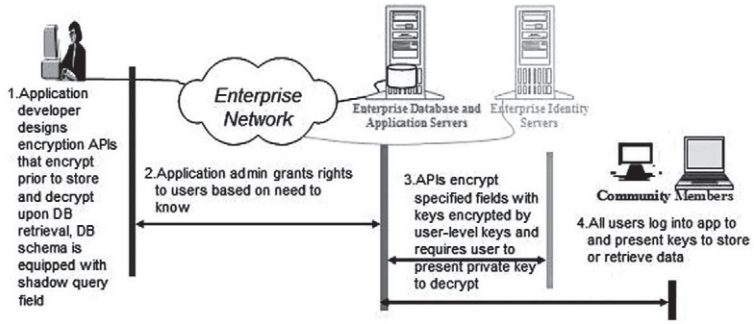

Figure 6: Application field level encryption

contains a shadow query field that allows the API to specify a unique record in the database without allowing the real data to be part of the user query.

Again, this approach allows a department authority to specify who the data is shared with, and it also restricts data access to the application, without exception. The DBA can also be prevented from accessing decryption keys by storing them on alternative technology, so administrative access to data would require multiple administrators to collude to violate policy.

However, the correct implementation relies on correct application source code. A rogue developer could allow excessive access to data by putting back doors in the code. However, they could not grant access to data to anyone that did not have access to the application.

Where some database fields are encrypted that have utility beyond a single application, all applications and reports that use them must rely on the shadow field to specify records. If the shadow field becomes corrupted (perhaps via a bug in the application source code), the only way to recreate it would be to decrypt and recreate all the encrypted records and shadow fields.

Summary

In summary, although they do provide some protection when used in conjunction with a secure application architecture, defence-in-depth strategies have failed the test of time in securing data in an internet-connected, ubiquitous file-sharing world. Encryption itself is still not the silver bullet because authorised users often have unfettered access to unencrypted data. In today's enterprises, data leakage is the rule not the exception.

However, information security professionals are on the hook to change this situation. Currently, they have few off-the-shelf tools at their disposal. However, data-centric technology is a meaningful buzzword that, if it evolves according to its current vision, could provide technology building blocks that would allow security professionals to label data and restrict data access to authorised use cases.

Currently, these efforts will all be customisations, as few vendor products currently deny data access at the database level and fewer restrict the access of administrators. To reduce the amount of customisation necessary in the future, security practitioners should foster vendor recognition of the application-field definition model. Though no implementation may be perfect, a consensus on requirements will help products evolve toward a data-centric vision.

About the author

Jennifer Bayuk is an independent consultant on topics of information confidentiality, integrity, and availability. She is engaged in a wide variety of industries with projects ranging from oversight policy and metrics to technical architecture and requirements. Jennifer has a wide variety of experience in virtually every aspect of information security. She was a chief information security officer, a security architect, a manager of information systems internal audit, a big 4 security principal consultant and auditor, and a security software engineer.

Jennifer frequently publishes on information security and audit topics. She has lectured for organisations that include ISACA, NIST, and CSI. She is certified in Information Systems Audit (CISA), Information Security Management (CISM), Information Systems Security (CISSP), and IT Governance (CGEIT). She has her masters degrees in Computer Science and Philosophy.

References

1. Control Objectives for Information Technology Version 4.1, IT Governance Institute, DS5.3 and DS11, 2007: p 118 and 143.
2. Bayuk, J, Stepping Through the InfoSec Program, ISACA, 2007:
3. Bilger, O'Connor, Schunter, Swimmer, Zunic, Data-centric security. Enabling business objectives to drive security, December 2006. IBM Global Services.
4. Data Security Standard Requirements and Security Assessment Procedures version 1.2, Payment Card Industry Security Standards Council, October 2008.
5. File encryption technology methods have been in well known and in practical use since the mid-nineties. Kaufman, Charlie, Radia Perlman and Mike Speciner, Network Security, Private Communication in a Public World, Prentice Hall, 1995. These have been incrementally improved, and catalogued by the Information Technology Laboratory at the US National Institute of Standards and Technology, see Federal Information, Processing Standards Publication 140-2, Security Requirements for Cryptographic Modules, Most recent update 12-03-2002.