Biometric Security

It's all about identification and authentication

Richard Dale

Biometrics makes it possible to authenticate an individual's identity based on his or her unique personal characteristics. Fingerprints are the most widely used biometric, although other technologies use facial geometry, hand silhouettes, and eye scans (based on either the color pattern in the iris or the blood vessel pattern in the retina). To effectively apply biometrics for system security, however, you must solve the fundamental problem of converting existing password-based access control to biometric authentication without imposing the costly burden of modifying application source code.

Identification and authentication (I&A) is basic to security. In I&A, identification tells the system who you are, while authentication provides some evidence that you really are who you claim to be.

Authentication is generally based on one or more of three factors:

- Something you know.
- Something you have.
- Something you are.

Traditional passwords are the obvious example of something you know, and various kinds of hardware tokens implement an approach requiring something you have. Biometric systems, such as a fingerprint or iris recognition, employ something you are.

Passwords have been with us since the dawn of the computer. After providing your user ID to a computer system, you enter your password. In spite of the deficiencies of conventional passwords, the use of passwords for I&A remains nearly universal.

The problems associated with passwords are well known. They can be guessed, stolen, or cracked. In some environments, users deliberately share passwords for their own convenience. A system that uses only passwords to control access cannot authenticate whether the user identified by a password is really the authorized user or someone else who borrowed or swiped the password. Passwords are also costly to administer. Password hassles account for a significant portion of help-desk costs. By some estimates, 40 to 80 percent of all calls to corporate IT help desks deal with passwords and frequently result in resetting forgotten or compromised passwords.

It's a Catch-22. Since passwords are not truly secure, many organizations require 8-character passwords and change them every 60–90 days. However, that makes them even harder for users to remember. Consequently, users are even more tempted to write them down, which only further undermines security. The end result? More calls to the help desk and more IT staff time diverted from development to support.

Hardware tokens implement a methodology based on something you have for identification. Requiring passwords or PINs to gain access to tokens often supplements this approach. Tokens, of course, have been around for years, and depend on one or another cryptographic technique. Traditional strategies include encrypting a timestamp and encrypting a random challenge generated on an authentication server. Cryptographic smartcards and similar devices are gaining wider use as authentication tokens, either by digitally signing a random challenge generated on the server or, as in the case of Windows 2000, actually decrypting the session key generated on a Kerberos server.

To convert an application to use tokens, you typically must modify the application to use either a vendor-supplied API or, in the case of cryptographic tokens, a standard API such as PKCS-11 (Cryptoki) or Microsoft's Crypto API. This is one of the basic problems with tokens—you have to change application source code to implement them.

In addition, portability is both a strength and weakness of hardware tokens. Because of their compact size, tokens can be easily pocketed and carried from one system to another. Thus, if users are required to use tokens both from home and at work, they may leave them behind on a nightstand and arrive at the office unable to logon or access secured files. Hardware tokens are generally at greater risk of physical damage than is typical for most computer equipment. For example, a hardware token may not function properly after being washed and dried with the laundry. These are silly but all too common problems that aggravate users, IT staff, and security administrators. Moreover, tokens left on the user's desk, or a smartcard left inserted in its reader, provide little more security than the password required to access them.

Biometrics for Identification and Authentication

Biometrics provide an alternative approach to I&A by implementing a methodology based on something you are. Biometric I&A has been used for some time in specialized applications, such as physical access control and paperless timecard systems. Use of biometrics as an access control mechanism for computer systems and networks is gaining wider acceptance, in part because of the dramatic decline in biometric hardware prices over the last few years. In fact, according to the International Biometric Group (http://www.biometricgroup.com/), computer security is the fastest growing application of biometrics. Depending on the security requirements of a specific network and set of applications, biometrics can be used as a standalone user I&A mechanism, or security managers can deploy biometrics in combination with other access control tools to provide even stronger user I&A.

Richard is chief scientist of BioconX, which develops biometrics software for network and application security. He can be contacted at rdale@bioconx.com.

Biometric authentication requires a hardware device that captures a physical characteristic of the end user. The most common biometric devices capture fingerprint images. Fingerprint scanners are available from a variety of vendors. Some use an optical system and camera to capture fingerprint data, while others incorporate a solid-state sensor that detects fingerprint ridges that are placed in contact with the sensor itself.

Biometric systems based on a sufficiently unique physical characteristic can perform both identification and authentication in a single step. This key attribute of biometrics is one of its most important advantages. With finger scanning, for example, users can identify themselves to a system simply by clicking a button on the screen, then placing a finger or thumb onto the sensor built into an attached reader or right into their computer mouse or keyboard. They do not have to type in a user ID or password. Users whose biometric matches one of the biometric templates of authorized users are automatically authenticated and logged on.

The Access Control Conversion Challenge

According to the annual survey conducted by the Computer Security Institute (http://www.gocsi.com/) and the FBI's Computer Intrusion Squad, 85 percent of surveyed corporations and government agencies detected computer security breaches in 2001. And that figure doesn't even include virus attacks. Thus, IT security must meet the challenge of strengthening user I&A to improve security.

However, migrating an organization's password-based applications to an I&A scheme based on biometrics is a tough challenge for developers. As noted, most approaches to strong user authentication require that applications be recoded to use an authentication API. This process can be daunting—or even impossible—in environments that are heterogeneous to begin with and that employ applications from third-party vendors that may not offer suitable hooks for implementing effective I&A.

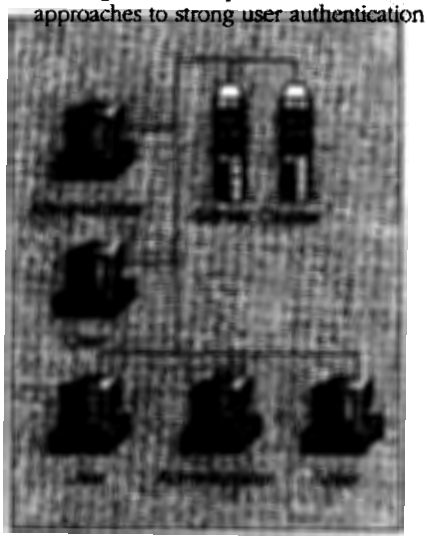

Figure 1: *BioconX architecture.*

A New Semantic

BioconX (the company I work for; http://www.bioconx.com/) tackled this issue by developing a biometric I&A system that does not require organizations to change a single line of source code. BioconX software accomplishes this by using existing password-based systems, while eliminating users' knowledge of their passwords.

Instead of users manually entering logon information, BioconX software identifies the user based on a biometric scan, automatically authenticates user identity, then instantly provides the necessary logon information. Users do not need to know (and, indeed, do not know) the password for signing on to a network or application. Rather than serving as a mechanism for identification via something you know, the password becomes a secret shared only between the BioconX server and the network and applications.

The syntax of the logon remains unchanged, but the semantic is completely different. A traditional password means: "User X has provided password Y." The new semantic means: "BioconX software has scanned the fingerprint or iris of the individual seated at the computer and has determined the person to be user X. User X's password for this specific network or applications for which he is authorized is Y/."

Internally, the software processes fingerprint images by locating the minutiae points; that is, the points where fingerprint ridges terminate or bisect. There are lots of them—typically from 50 to 100 on a fingerprint. The challenge is that there is no way to simply look at a set of minutiae points and come up with an index in the database to match against. Fingers are positioned differently from one scan to the next, they may be applied to the sensor with different pressure, they may be drier on one occasion than on another, and the processing is rarely perfect. Or the software may miss minutiae points that are actually present in the scan, and may introduce minutiae that aren't actually there.

On a small scale, missing a few minutiae points (or adding a couple) isn't a problem because the number of minutiae present on a fingerprint is an order of magnitude greater than the number required to be confident of a match. However, when you're trying to find the right record from a set of several thousand based on data that are fuzzy and incomplete...well, that's a problem.

How It Works

As Figure 1 illustrates, the software consists of:

- A client component that runs on the user's system and is responsible for obtaining biometric information about users from a fingerprint scanner (or other device), interactions with user systems and their applications, and interactions with the BioconX server.
- The BioconX server that maintains a database of users, applications, biometric data and comparison templates, and logon information.
- The Administrator that is used by security administrators to set up and manage user accounts, as well as configure applications for biometric login.

The client component runs on end-user workstations. Client components administer the initial network login via a GINA on Windows NT/2000 systems, or via a network provider on Windows 9x. During logon, a biometric scan is obtained from users. Scan data are forwarded to the BioconX server where they are matched against template records in the database. If a match is found, the software retrieves the user's logon and application information from the database. Back on the user's system, the user is logged on to the network. Applications configured for launch at startup are launched, and logon information (typically, the username and password) is provided to each application as necessary in the background without any user action.

The BioconX server uses one or more biometric databases to map users' physical characteristics and a SQL database as a repository for user and application information. Any SQL that supports an ODBC interface can be used for this purpose. Passwords stored in the database are encrypted. Coherency among multiple servers is maintained by Microsoft Cluster Server. Because of the sensitivity of logon information, organizations deploying BioconX software often house the servers in physically secure locations.

Applications do not need to be launched and logged into at startup. BioconX provides biometrically authenticated login service regardless of when an application is launched. BioconX constantly monitors the system for process initiation and window creation. When a new login window appears for an application that BioconX recognizes, BioconX prompts for a fingerprint (or iris scan) and forwards the information to the server. The server matches the scan against its database and responds to the client with the logon information required

(continued from page 94)
by the application. If users have logged out of an application and want to log back in again, BioconX login services are invoked by clicking a tray icon, and selecting relogin from a pull-down menu.

The BioconX Administrator software is used to setup users and configure applications for biometric login. Like the user tool, the administration tool communicates with the server using an encrypted network connection. Each BioconX user requires a user account, created and administered by the administration tool. To facilitate managing a large population of users, the BioconX Administrator lets users with similar access rights and application requirements be aggregated into user groups.

User setup is carried out using the BioconX Administrator. Biometric data are collected for each device users need to use, and the users' network logins and applications are configured. The administrator automatically synchronizes BioconX user accounts with network operating systems and with Microsoft Exchange. Users can be created, deleted, or modified from within BioconX Administrator, and those changes are automatically propagated to the NT SAM database, Windows 2000 Active Directory, Novell, and Microsoft Exchange. Passwords can be set manually by the administrator or generated randomly. In the case of randomly generated passwords, even the administrator never knows the password assigned to a user account. Similarly, user group memberships can be assigned or modified.

The BioconX Administrator also configures applications for biometric login. This is accomplished by creating templates that define all the information the client requires to recognize, launch, and login to each application. Each template includes a login script that defines the keystrokes required to navigate the application's login screen. The Administrator software lets applications be configured in one of two ways, either by using the Application Wizard or by configuring the application manually. The Application Wizard captures the required information by monitoring a controlled launch and login of an application. The captured information includes the application's executable file name, command-line switches, window titles, and keystrokes required to generate the application's login script.

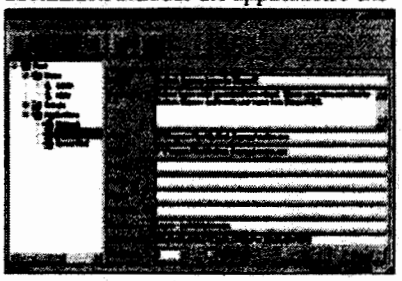

Figure 2: *Typical BioconX Administrator screen.*

When configuring an application manually, administrators provide this same information by entering it into a form; see Figure 2. The scripting language provides facilities for entering text, special characters, and control characters such as Tab, Delete, and Backspace. BioconX logs a user into an application by playing back the sequence of keystrokes defined in the script, while inserting logon information such as user name and password specific to individual users.

The biometric login process just described offers a significant improvement over the security of simple passwords and actually makes the user's life easier, rather than imposing a new burden (such as keeping track of a token). Instead of manually typing user names and passwords, all users need do is click a button and lay their thumb or finger over the scan window on the keyboard or mouse (or look into a camera for an iris scan).

For environments requiring even stronger user I&A, BioconX provides two-factor and three-factor authentication solutions. BioconX two-factor solutions marry something you have with something you are by supplementing the biometric authentication described earlier with challenge/response authentication using a cryptographic token. In challenge/response authentication, the cryptographic token is used to digitally sign a random challenge generated by the server. To authenticate users, the server verifies both the digital signature and biometric scan. BioconX three-factor authentication supplements two-factor authentication by adding a password to the mix.

Setting Up

Setting up a BioconX-based security system is straightforward. For instance, Figure 2 shows the main BioconX Administrator screen with the application configuration window for Outlook Express.

- Name specifies a unique name associated with the application.
- Description provides a brief explanation of the application.
- The five Path edit boxes specify possible paths from which the application can be launched. Multiple possibilities are permitted to support different hosts that use different paths to the same application.
- Caption identifies the caption associated with the login window. The BioconX client uses Caption and Path to identify the specific application being launched.
- Keys to Send specifies the key sequence that the BioconX client sends to the application to login users. In this case, SH{TAB}{DELETE} tabs to the Outlook Express User Name edit box and clears its contents. UN,{TAB} inserts the user's application user name from the BioconX database at the current location, and tabs to the password edit box. PW,{TAB} inserts the password from the database and tabs to the Outlook Express logon OK button. The final {ENTER} sends a carriage return.
- Pause After provides a way to insert a pause between launching an application and inserting logon information.

The BioconX client itself attaches no meaning to the keystrokes it sends to the application. In this example, I use {TAB} and SH,{TAB} to navigate around the window, but the interpretation of this virtual keyboard input is the responsibility of the application and may be different from one application to another. This gives the BioconX client the flexibility to login to mainframe and UNIX applications through terminal emulators, mainframe logins on a 3270 emulator, and standard Windows and web applications.

BioconX Version 3.5 runs on Windows and is written primarily in Microsoft Visual C++, although some user-interface components are written in Visual Basic.

User and application profiles are typically kept in a Microsoft Access database cohosted with the server, although any ODBC data source can be used. Because all database access takes place on the server, the usual performance issues associated with using an Access database in a network application are avoided. The server uses ODBC interfaces to access the database, so it doesn't really know (or care) what database is actually being used. The scripting language is based on the Visual Basic SendKeys method with a couple of extensions. The script language uses a comma to separate individual SendKeys "chunks." To send a comma, use {,}. The strings UN and PW are used to specify the application user ID and password, respectively. The construct {DPL *n*} inserts a delay of *n* seconds.

Conclusion

Biometrics offers a strong authentication alternative to traditional passwords and tokens, and can do so without imposing the burden and cost of application source-code modification. This is accomplished by converting existing password-based access control schemes to biometric I&A while maintaining the syntax of password-based logon. Where security requirements mandate even stronger authentication, biometrics can be used in combination with a token, or with a token and a password.

DDJ