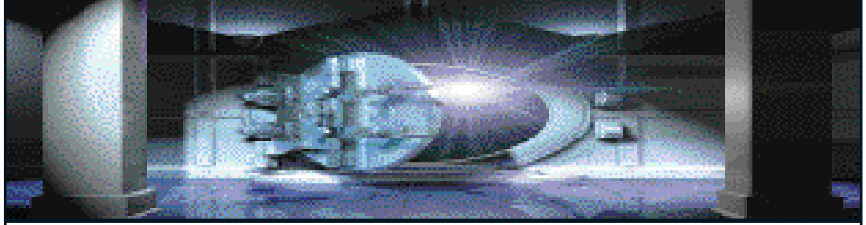

SECURITY PATROL

by Vincent LeVeque

Passwords that Pass Security

QUESTION: Our auditors recently reviewed our AS/400, and their report criticized our system values for setting password characteristics. They want us to increase our password length, expire passwords every 30 days, and prevent people from using ordinary words as passwords. I understand why they think this would improve security, but I know our users, and they will either just write down the passwords or be constantly calling support because they have forgotten their them. Is it necessary to be so strict about passwords?

ANSWER: How sensitive is the information? What are the chances that someone would want to steal or guess a password? How would they get caught if they did? How well do you really know your users? Would they really rebel, or are you just assuming they will? You should know the answers to these questions so that you can develop a realistic password policy. The standards used by auditors represent generally accepted practices and do not take into account your specific needs. Your own standards may be either weaker or stronger than what the auditors recommend. Documenting the risk and the decision behind your organization's password policy will go a long way toward helping you pass an audit.

See the article “Password System Security Values Demystified” (*MC*, May 1998) for more information on setting password characteristics. Many of the password characteristics for which systemwide defaults may be set can be overridden at the individual profile level. You can, for example, set a default password expiration interval of 90 days, but you'll want to set it at 30 days for more powerful accounts, where the risk is higher. Prudent practices generally include the following:

- Set a minimum password length of at least six characters by setting the system value QPWDMIN to 6. This will decrease the chance that trivial passwords will be selected.
- Set a password expiration time between 30 and 90 days by setting the system value to the appropriate time range. Instead of using the systemwide default, individual profiles can be set through the Password Expiration Interval (PWDEXPITV) option of the individual profile, using the Change User Profile (CHGUSRPRF) command.
- Require differences in consecutive passwords so that users are not recycling old passwords. Setting the system value QPWDRQDIF to 6 will prevent the previous six passwords from being chosen as a new password. This will force a certain amount of originality in passwords, making them more difficult to guess.
- Set QMAXSIGN to a number no greater than 5. Someone who cannot type in his password correctly after five attempts has probably forgotten it and should have it reset rather than waste his time trying again and again. A limit of five will also seriously slow down someone trying to guess another person's password.

You are right in stating that there is a trade-off. The very things that make it easy for users to remember their passwords make it easy for others to guess what they are. Changing passwords frequently also makes it more difficult for users to remember nonobvious passwords. These are simple facts of human cognitive psychology. Password strength must take these facts into account. My own belief is that AS/400 passwords should not be easily guessed, but they need not be purely random, nonsensical strings to provide adequate security.

The exact nature and degree of trade-offs varies depending on your security requirements, how your systems are used, and the nature of your user community. These trade-offs between stronger security and other needs are not appropriate for technical staff or management to make on their own. They are basic business decisions that should be made by the managers who are ultimately responsible for your business' information assets.

QUESTION: Can AS/400 passwords be cracked the same way that UNIX passwords are cracked?

ANSWER: The ease with which passwords can be cracked and the ability to do so offline directly affect your password policy. Many times, password recommendations are based on systems whose password files are easily downloaded and broken offline. UNIX and, to an extent, Windows NT fall into this category. For these systems, the strongest-possible passwords are necessary. If you use a standard word from a dictionary, your password will be cracked within a matter of minutes. A random collection of characters may take days, however. You need the strongest-possible passwords to make a dictionary attack as time-consuming as possible.

It is possible to use published APIs (or other methods) to obtain encrypted AS/400 passwords. Recently, postings to the MIDRANGE-L Internet mailing list (*www.midrange.com*) indicated that the encryption used for AS/400 passwords is breakable. The poster demonstrated the possibility but (fortunately) did not post step-by-step directions. I hope that IBM clears this up; this exposure certainly does not put the AS/400 on the same footing as many UNIX systems when it comes to easily cracking passwords. Although you should know that this exposure exists, it should not drive your password policy in the short-run.

Obtaining AS/400-encrypted passwords requires system privileges. This is in contrast with many UNIX systems, where encrypted passwords in /etc/passwd are world-readable, and even with Windows NT, where a boot floppy and a few utilities will permit downloading of encrypted passwords that are present in the Security Account Manager (SAM) file. In both of these cases, breaking passwords is part of a strategy designed to increase the attacker's system privileges. On the AS/400, you need the privileges *before* you can get the encrypted passwords.

Cracking AS/400 passwords is, therefore, more difficult than cracking those of the aforementioned systems, and, even when it is possible, the intruder is not provided the same benefit. My personal belief is that passwords on an AS/400 should be strong enough to prevent a coworker or other knowledgeable party from guessing a person's password within a few attempts. Increasing the difficulty of passwords to the point at which users feel they must write them down to remember them is probably counterproductive. The risk of someone else finding a written-down password is much greater than the risk of someone with privileges downloading and cracking the password file. Your policy should forbid the use of user IDs, user names, family member names, favorite sports teams, or other obvious facts as passwords. PentaSafe (*www.pentasafe.com*) makes a product called PSSecure, which ferrets out obvious passwords in violation of your policy. The program requires privileges to run and uses the authorized IBM APIs.

AS/400 password-cracking is not a high-security risk; however, this could change. All it takes is someone to figure out how to reliably bypass OS/400 operating system protections and then publicize the methods. It's unlikely but not impossible. The use of authorized APIs poses risks as well. If you require these APIs for legitimate purposes, you need to ensure that they are used only for authorized purposes and that they do not become a password-cracking toolkit. Tools, such as the one marketed by PentaSafe, should be controlled properly.

QUESTION: If passwords are so terribly flawed, what can I do to keep my systems secure in the long run?

ANSWER: You are correct. Passwords *are* seriously flawed as a secure authentication method. For the most secure applications, companies have increasingly turned to other means of authenticating users. These means include biometrics and tokens, which provide the user with one-time passwords. The most popular token devices are those made by RSA Security. These tokens generate a numeric password every minute or so. The password is good only once and only for a couple of minutes before it expires. Even an attacker sniffing my network connection could not steal a useful password under these conditions.

I use a SecurID key-chain token to remotely access my employer's network, retrieve email, and access shared files. It's much more convenient than memorizing a random character password. The device is physically about the size of a car-alarm remote control. It has an LCD display, which shows a six-digit number that changes every minute. To remotely access my employer's network, I type a secret four-digit PIN, followed by the SecurID "number of the minute."

I've found a product, made by DetectIT (*www.detect-it.com*), that does allow the use of SecurID tokens for AS/400 logons. When used together, DetectIT-ACF and RSA SecurID provide an interface between the widely used SecurID system and AS/400 native authentication so that the constantly changing code allows you to log on to the AS/400 rather than use a fixed password.

I have no direct experience with DetectIT's products. However, it is good to know that someone out there has tackled the job of integrating a secure password substitute into our favorite midrange computer. If you require something stronger than static password authentication, you should definitely evaluate DetectIT's offerings.

Vincent LeVeque is a senior security engineer for Science Applications International Corporation (SAIC). He can be reached at *vleveque@earthlink.net*.