AUTHENTICATION

IDENTIFICATION.

A vulnerable point in any security system is its identification methods : how does it make sure that someone trying to log in as A.B. Smith really is A.B. Smith ? If you can get past this barrier, the system will believe that it knows who you are, and give you free access to all A.B. Smith's privileges. So far as is known, there is only one really reliable way to achieve the required level of security :

KEEP EVERYONE ELSE AWAY.

If no one else can get to your computer system, no one else can steal your data, or do whatever else it is that you're sensitive about. There are several ways to manage this.

- Best : have your own computer, physically inaccessible to anyone else. (That includes electronic accessibility : no network connections.) This is a comparatively expensive way, particularly if your computing requirements are extensive. Microcomputers help a lot, but aren't always the answer – they might suffice if your needs are modest, but still can't handle really large-scale work. And they're rather too portable to be left around, so you have to be careful about theft. This solution works if you're paranoid about security and phenomenally rich; it's usually restricted to military systems.

- An alternative is to share a more elaborate single-user system. You get a better system, but you have to carry your disc packs (or whatever) with you. This is a trip back in time to the pre-monitor-system days – though one might hope for rather more modern hardware. It worked well, in its own way; it doesn't use the machinery very efficiently, but we don't worry about that much now. (Most microcomputers do nothing for most of their lives.) Our microcomputer laboratories worked in this way for a long time before networked computer services were required.

- Simulate a single-user machine on a shared system. The thin end of the wedge – once interference between jobs is physically possible, someone is likely to try it. Even perfectly law-abiding people immediately start to want communications, which opens further loopholes to abuse. This approach merges with the previous category if the shared system is a distributed system sharing network services, such as file store, printing, and communications.

In general, it seems that unless your work is unusually isolated this policy is too xenophobic. Most people need some sort of communications with other people in their group in order to carry on with their work effectively. This became very clear when individual microcomputers began to displace terminals in commercial organisations. The move was originally welcomed by many as a step to freedom from the dictatorial influence of the computer (or, in the commercial world, data processing) centre. It turned out to be also a step to freedom from all the support services which the evil computer centre had been quietly offering – communications, backup, database access, etc. – and chaos reigned until network services helped to some extent to pull the system together again.

- BUT IF THAT DOESN'T WORK –

- then we still have to grapple with the problem we started with : how does the system determine that you are you ? In operating system terms, we must find some way to convince ourselves that someone pressing the keys on some keyboard really is entitled to use the property and privileges associated with some name in the userdata system.

People manage the same authentication process by recognising faces or voices, checking personal knowledge unlikely to be available to an impostor, inspecting identity cards, and such means. We can use much the same means in a computer system, provided that we can adapt it to the much more limited sensory abilities of the computer.

Three classes of technique are used in practice, all based on the person requiring access producing something that no one else should have. This can be :

- Something you *know* – such as a password;
- Something you *own* – such as a card key; or
- Something which is *part of you* – such as a fingerprint.

PASSWORDS.

While the passwords are *supposed* to be secret, all they really prove is that the person granted access to all your resources knows your password, and it's up to you and to the system to make sure that only you know, and that no one else can find out. Notice that the secrecy depends on both parties to the secret :

- People are fallible – passwords can be *given away*, or *stolen*, or *discovered* by trial and error, or *written down*. Or, of course, *forgotten*.

- Systems are fallible – passwords might be stored in legible form, or they might have to appear in command files, where they might be especially vulnerable – and also hard to change.

It almost seems as though there is some sort of principle of conservation of peril connected with passwords, as actions taken to increase security in one respect can lead to more dangerous practices in others. For example, some systems require that all passwords are changed at regular intervals – but to cope with the change people are much more likely to write down this week's password in a convenient place. Similarly, an obvious and easily memorable password is likely to be easy to discover; a less obvious nonsense password is easy to forget.

One of many alternative approaches is to use a pass algorithm rather than a password : everyone using the system has a unique algorithm to work out a proper response to some prompt presented by the system. For example, the system could present you with a letter of the alphabet, to which you might respond with the next two letters in sequence. It seems that this works reasonably well, but the range of simple algorithms which are easy to remember and to work out without writing things down is probably limited.

OTHER METHODS.

Here is an interesting account of real life technology[REQ12] in the field of authentication, covering several methods and discussing their good and bad features.

For more than a year, an intruder rifled through the files of some three dozen computer systems in the U.S. military research complex from the comfort of his West German home, sifting out information on the strategic defense initiative and other defense topics. But in August 1986 Clifford Stoll, manager of a multi-user computer system at Lawrence Berkeley Laboratory in California, noticed a 75-cent discrepancy between the system's two accounting systems, one of which is a public, canned program and the other, a home-grown program only insiders knew about. After rejecting software error as the source, he discovered an account on the canned, public program that did not appear in the homegrown account listing.

Rather than construct barriers to the intruder, Stoll worked with law-enforcement agencies on monitoring the rogue's activities and by mid-1988 had enough information to close in on him. Finally a fictitious file so piqued the intruder's interest that he stayed on the line long enough to be traced. As a result, in March West German police arrested three members of a major computer spy ring that had been stealing U.S. military information for Soviet intelligence agents from Internet, the research computer network built around the Defense Advanced Research Projects Agency's ARPAnet for academic users.

The incident was one of a recent string of computer break-ins, including the case last November in which Robert Morris Jr., a graduate student at Cornell University, Ithaca, N.Y., is alleged to have infected Internet with a self-replicating program called a worm[REQ13].

Morris' worm reproduced itself so many times in so many different computers that it snarled up processing at a number of academic centers for two days.

Raising security consciousness

Many systems would become secure enough if only a few simple access control and data backup measures were implemented. As a first step, experts recommend that any organization relying on data processing issue a corporate policy statement signed by the executive officer outlining what data needs protection and who has responsibility for protecting it.

Merely raising users' awareness of the problem helps; they commonly leave disks or passwords lying about, not realizing that anyone would be interested in taking them. Often they share disks with others and choose easily guessed passwords, such as birthdays, days of the week, or social security numbers. One security consultant found four users with the password *sunshine* in a client's computer system.

Any password found in the dictionary is at risk because if a person won access to files of encrypted passwords – which is how passwords are stored on most computer systems – he or she could run them through dictionary-based decrypting programs. For example, Stoll watched his West German intruder download encrypted password files into his computer, and log back on within a few days using passwords from the files.

Searching for leaks

Once into a computer on a network, an intruder can leverage his break-in to invade other users' files or other computers on the network by guessing passwords, searching for passwords left in files, or trying to attain the omnipotent level of privilege enjoyed by the operating system, which in most operating systems is called supervisor state or system-manager privilege.

This status can be attained by exploiting weaknesses in software added on to a system. The West Germans infiltrated Internet by exploiting several security holes. One well-known hole was in a text editor called Gnu-Emacs, which runs on the Unix operating system. The program allows the user to forward a file to another user, but does not bar moving the file into the systems area, where the operating system is stored. Such a command can normally be executed only by a system manager or other privileged user, but the program did not check the user ID.

The intruder created a file that would grant him supervisor status when executed at the system level. By renaming it, he disguised it as a utility program periodically run by the operating system, and moved it into the systems area. Once his privileges were acquired, he had the system scan files for such defense-related words as *sdi, nuclear, norad,* and *kh-11,* and for passwords, which are often left in computer files. He gained access, for example, to a Cray supercomputer after finding its log-on sequence with account numbers and passwords outlined in a file sent by electronic mail and stored by the recipient – a common practice.

Some companies have begun hiring computer security consultants who take on the role of a hacker trying to gain unauthorized access to the system. Far from the thrilling activity depicted in the movie *Wargames,* hacking seen in this light is very tedious. The consultant checks all suspect programs on the system for any step that could grant him supervisor status.

Without access to the proprietary source code for computer packages on a client's system, the consultant must reconstruct each program's code by reading the instructions in memory. Then, he scans for code that is vulnerable to misuse. Just such an instruction, running in a subroutine on a client's IBM mainframe, had been found by computer security consultant Peter Goldis of Cambridge, Mass., on the day *IEEE Spectrum* interviewed him.

The routine was a common extension to IBM Corp.'s MVS operating system that provided "cross memory services," allowing the user to store data from his program or address space in another address space that he could specify. The operating system keeps users separated from sensitive data by storing data in different memory keys, each with its

own access restrictions. Only the operating system itself or a user (or program) that has been granted supervisor status can move data across memory keys.

Therefore, before allowing a user to perform a cross memory service, the program should check the user's ID, but with the multitude of software vendors, not to mention homegrown software, these security measures sometimes fall through the cracks. "The guy who wrote this program was in a rush or didn't understand what he was doing," said Goldis.

Fortifying access control

Some believe that only a few safeguards will do the trick. "I can't think of a single instance when a hacker penetrated a system that had modest protection," said Courtney. He defines modest protection as denial of access after three unsuccessful password attempts, and on dial-up lines, placement of the access barrier in front of the modem instead of behind it, requiring the caller to enter the account number and password before obtaining a modem tone. In the conventional setup, a trespasser can have his computer keep dialing different numbers until it hits upon a tone, signaling a computer system.

Others, though, believe that password-based access control is inadequate. Rather than requiring only something the user knows, some systems now require the user to possess a card, key, or calculator-shaped token (a unique physical object) in addition to a password to get on the system. A third category of techniques requires a physiological characteristic unique to the user, such as a fingerprint, and encompasses a variety of biometric devices.

One of the more promising token-based access devices is a card the size of a credit card containing a microprocessor, liquid crystal display, and a battery. Synchronized with the host computer, the card generates a new password at set time intervals, say every 60 seconds, using an algorithm known only to the host. Thus, when the user at his terminal keys in a secret personal identification number (PIN) associated with the card and the password shown on the card's display, the host software can verify that the user is in possession of the card.

Because of its simplicity, the dynamic password, as it is called, has been picked up by regional telephone companies, government laboratories, and defense contractors. Rockwell International has been using cards from Security Dynamics, Cambridge, Mass., for more than a year, according to Tipton. The cards work in conjunction with microcomputer-controlled switches made by Micom Systems Inc, Simi Valley, Calif., that manage dial-up access to the company's computer network at nine points around the United States. At each switch, software from Enigma Logic Corp., Concord, Calif., checks the PIN and dynamic password of each caller and then grants access only to lines the caller is authorized to use.

Another class of token-based systems uses challenge/response devices that need not be synchronized to the host. The host issues a random number challenge to a user when he logs on, and the user enters it into his device, which typically resembles a calculator. The device calculates a response using an algorithm known only to the host, and the user enters the number into his terminal for verification by the host computer. This procedure, however, takes more of the user's and the host's time than the dynamic password does. Both challenge/response and dynamic password cards cost between $35 and $60 apiece.

Some tokens, usually those used to limit both computer access and physical access to a sensitive area, have electrical contacts and plug into a reader at the terminal so that the user has to enter only a PIN; the smart card performs the challenge/response procedure or some other log-in sequence. These cards cost around $13 apiece, but readers, which usually also perform encryption, cost about $400 apiece.

Cryptic messages

Because of its cost, encryption has historically been used only for top-security applications in, for instance, intelligence, financial transactions, and personal information in government databases. According to David Chaum, the cryptography group leader at the Center for Mathematics and Computer Science in Amsterdam, the Netherlands, encryption will soon be integrated into

workstations, whereupon costs will drop drastically. The digital signal-processing chips now available in workstations can encrypt data almost as fast as special hardware does.

Biological warfare

Biometric devices measure either a physical or behavioral characteristic to verify identity. Physical characteristics include fingerprints, blood vessel patterns on the retina, and hand geometry – all unique, unchanging human characteristics. Behavioral characteristics include voice, signature dynamics (the order and timing of strokes made during name signing), and keystroke rhythm on a keyboard. In general, physical biometric devices are more expensive and may be considered more intrusive than behavioral devices. Critics are concerned that biometric devices violate privacy and may facilitate employee monitoring.

Behavioral devices are more error-prone than the physical devices because the measured characteristics can vary in each person from day to day. Device performance is judged by the rates at which false acceptances and rejections occur, which can be adjusted on the devices by tightening or loosening the tolerance on the measurement. Most companies that aim for a balance of false acceptances and false rejections have rates under 1 percent, he said.

Miller said biometrics customers initially may have to live with relatively high false rejection rates to ensure that unauthorized users are blocked, but that the rate goes down as users gain more experience with the device. Some companies have their employees practice with the device for a few weeks before giving it full control over access.

There are now about eight biometric devices on the market with another 30 or so under development, but they have not been welcomed on a broad scale for computer access largely because of their expense. The units, which must be linked to each terminal or personal computer, cost $600-$7000 each. According to Miller, the total U.S. biometric market in 1988 was less than $5 million, of which only 16 percent went to computer security. (The rest went to limiting physical access to computers.) That percentage has been rising steadily for the last three years.

A survey of commercial practice in using authentication methods in working with networks and messaging systems[REQ18] showed that, because of the comparative ease of wire-tapping, passwords were regarded as essentially useless. The most highly regarded methods relied on various cryptographic techniques. As more and more computing work relies on communications of one sort or another, it is reasonable to see this a the way of the future. Biometric methods are not even mentioned.

COMPARE :

Silberschatz and Galvin[INT4], Chapter 14.

REFERENCES.

REQ12 : Karen Fitzgerald : "The quest for intruder-proof computer systems", *IEEE Spectrum* **26#8**, 22 (August 1989).

REQ13 : *Communications of the ACM.* **32**, 677-710 (1989) : several papers presented as a special section.

REQ18 : K. Auyong, C.-L. Chee : "Authentication services for computer networks and electronic messaging systems", *Op. Sys. Rev.* **31#3**, 3-15 (July, 1997).