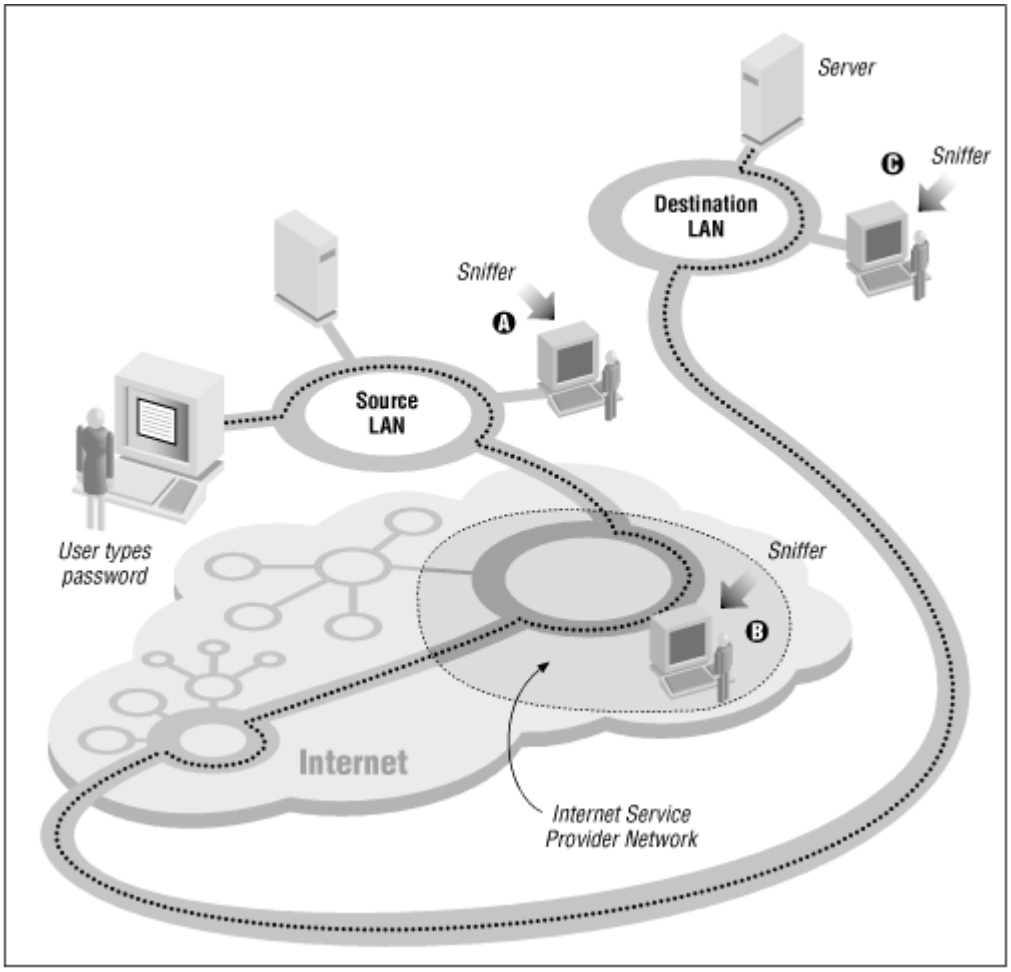
8.7 One-Time Passwords

If you manage computers that people will access over the Internet or other computer networks, then you should seriously consider implementing some form of one-time password system. Otherwise, an attacker can eavesdrop on your legitimate users, capture their passwords, and use those passwords again at a later time.

Is such network espionage likely? Absolutely. In recent years, people have broken into computers on key networks throughout the Internet and have installed programs called *password sniffers* (illustrated in Figure 8.2). These programs monitor all information sent over a local area network and silently record the first 20, 50 or 128 characters sent over each network connection.[12] In at least one case, a password sniffer captured tens of thousands of passwords within the space of a few weeks before the sniffer was noticed; the only reason the sniffer's presence was brought to the attention of the authorities was because the attacker was storing the captured passwords on the compromised computer's hard disk. Eventually, the hard disk filled up, and the computer crashed!

[12] Some sniffers have been discovered "in the wild" that record 1024 characters, or even the entire Telnet session. Sniffers have also recorded FTP and NFS transactions.

Figure 8.2: Password sniffing



One-time passwords,[13] as their name implies, are passwords which can be used only once, as we explained in Chapter 3, *Users and Passwords*. They are one of the only ways of protecting against password sniffers.

> [13] Encryption offers another solution against password sniffing, although it is harder to implement in practice because of the need for compatible software on both sides of the network connection.

Another application which demands one-time passwords is mobile network computing, where the connection between computers is established over a radio channel. When radio is used, passwords are literally broadcast through the air, available for capture by anybody with a radio receiver. One way to ensure that a computer account will not be compromised is to make sure that a password, after transmittal, can never be used again.

There are many different one-time password systems available. Some of them require that the user carry a hardware device, such as a smart card or a special calculator. Others are based on cryptography, and require that the user run special software. Still others are based on paper. Figure 8.3, Figure 8.4, and Figure 8.5 show three commonly used systems; we'll describe them briefly in the following sections.

8.7.1 Integrating One-time Passwords with UNIX

In general, you do not need to modify existing software to use these one-time password systems. The simplest way to use them is to replace the user's login shell (as represented in the */etc/passwd* file; see "Changing the Account's Login Shell") with a specialized program to prompt for the one-time password. If the user enters the correct password, the program then runs the user's real command interpreter. If an incorrect password is entered, the program can exit, effectively logging the user out. This puts two passwords on the account - the traditional account password, followed by the one-time password.

For example, here is an */etc/passwd* entry for an account to which a Security Dynamics SecurID card key will be required to log in (see the next section):

```
tla:TcHypr3FOlhAg:237:20:Ted L. Abel:/u/tla:/usr/local/etc/sdshell
```

If you wish to use this technique, you must be sure that users cannot use the *chsh* program to change their shell back to a program such as */bin/sh*which does not require one-time passwords.

A few versions of UNIX allow the system administrator to specify a program (or series of programs) to be used instead of, or in addition to, the standard password authentication. In these systems, the program(s) are run, one after another, and their return codes are examined. If any exit with an error code, the login is refused. AIX is one such system, and future versions of Solaris are slated to include such functionality.

> **NOTE:** There are many ways to gain access to a UNIX system that do not involve running a shell, such as FTP and NFS. If you use a special shell to implement one-time-passwords, these methods of access will not use the alternative authentication system unless they are specifically modified. You may wish to disable them if you are unable to replace them with versions that use the alternate authentication mechanism.

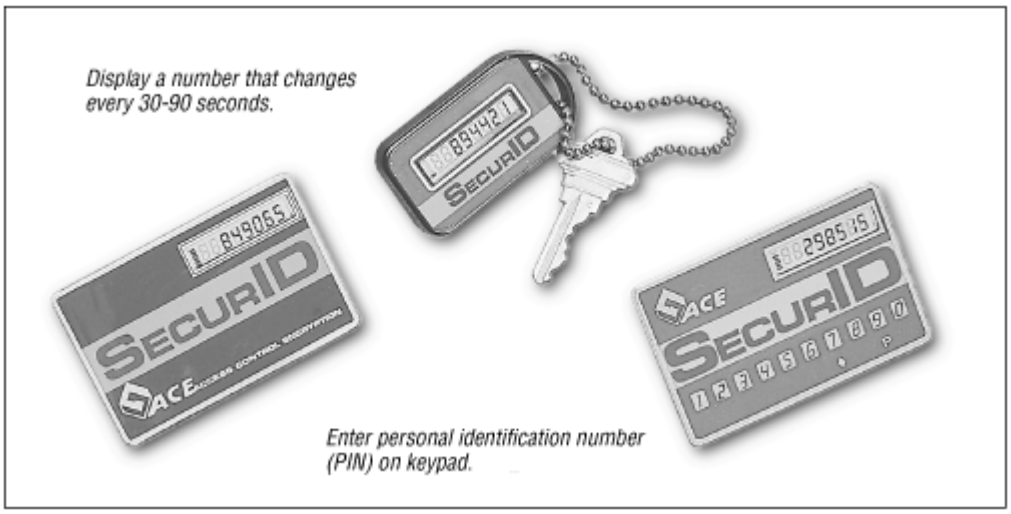
8.7.2 Token Cards

One method is to use some form of token-based password generator. In this scheme, the user has a small card or calculator with a built-in set of pre-programmed authentication functions and a serial number. To log in to the host, the user must use the card, in conjunction with a password, to determine the one-time password. Each time the user needs to use a password, the card is consulted to generate one. Each use of the card requires a password known to the user so that the card cannot be used by anyone stealing it.

The approach is for the card to have some calculation based on the time and a secret function or serial number. The user reads a number from a display on the card, combines it with a password value, and uses this as the password. The displayed value on the card changes periodically, in a non-obvious manner, and the host will not accept two uses of the same number within this interval.

The SecurID shown in Figure 8.3 is one of the best-known examples of a time-based token. One version of the SecurID card is based on a patented technology to display a number that changes every 30-90 seconds. The number that is displayed is a function of the current time and date, and the ID of that particular card, and is synchronized with the server. Another version has a keypad which is used to enter a personal identification number (PIN) code. (Without the keypad, a password must be sent, and this password is vulnerable to eavesdropping.) The fob version shown in the figure provides stronger packaging; it's especially good for people who don't carry wallets or handbags, and carry the device in a pocket. The cards are the size of a credit card and have a small LCD window to display the output.

Figure 8.3: Security Dynamics SECURID cards and fob



A second approach taken with tokens is to present the user with a challenge at login. The SecureNet key shown in Figure 8.4 is a token that implements a simple, but secure, challenge-response system. Unlike the Security Dynamics products, the SecureNet key does not have an internal clock. To log in, the user contacts the remote machine, which displays a number as a challenge. The user types the challenge number into the card, along with its PIN. The key calculates a response and displays it. The user then types the response into the remote computer as her one-time password. The SecureNet key can be programmed to self-destruct if an incorrect password is entered more than a predefined number of times.

Figure 8.4: Digital Pathways SecureNet key card

There are many other vendors of one-time tokens, but the ideas behind their products are all basically the same. Some of these systems also can provide interesting add-on features, such as a *duress code*. If the user is being coerced to enter the correct password with the card value, he can enter a different password that will allow limited access, but will also trigger a remote alarm to notify management that something is wrong.

There are two common drawbacks of these systems: the cards tend to be a bit fragile, and they have batteries that eventually discharge. The cost-per-unit may be a significant barrier for an organization that doesn't have an appropriate budget for security (but they are cheaper than many major break-ins!). And the cards can be annoying, especially when you take 90 minutes to get to work, only to discover that you left your token card at

However, the token approach does work reliably and effectively. The vendors of these systems typically provide packages that easily integrate them into programs such as /bin/login, as well as libraries that allow you to integrate these tokens into your own systems as well. Several major corporations and labs have used these systems for years. Tokens eliminate the risks of password sniffing. They cannot be shared like passwords. Indeed, the tokens do work as advertised - something that may make them well worth the cost involved.

8.7.3 Code Books

A second popular method for supplying one-time passwords is to generate a codebook of some sort. This is a list of passwords that are used, one at a time, and then never reused. The passwords are generated in some way based on a shared secret. This method is a form of one-time pad (see Section 6.4.7, "An Unbreakable Encryption Algorithm").

When a user wishes to log in to the system in question, the user either looks up the next password in the code book, or generates the next password in the virtual codebook. This password is then used as the password to give to the system. The user may also need to specify a fixed password along with the codebook entry.

Codebooks can be static, in which case they may be printed out on a small sheet of paper to be carried by the user. Each time a password is used, the user crosses the entry off the list. After the list is completely used, the system administrator or user generates another list. Alternatively, the codebook entries can be generated by any PC the user may have (this makes it like a token-based system). However, this means that if the user is careless and leaves critical information on the PC (as in a programmed function key), anyone else with access to the PC may be able to log in as the user.

One of the best known forms of codebook schemes is that presented by S/Key. S/Key is a one-time password system developed at Bellcore based on a 1981 article by Leslie Lamport. With the system, each user is given a mathematical algorithm, which is used to generate a sequence of passwords. The user can either run this algorithm on a portable computer when needed, or can print out a listing of "good passwords" as a paper codebook. Figure 8.5 shows such a list.

Unfortunately, the developers of S/Key did not maintain the system or integrate it into freely redistributable versions of */bin/login*, */usr/ucb/ftpd*, and other programs that require user authentication. As a result, others undertook those tasks, and there are now a variety of S/Key implementations available on the Internet. Each of these has different features and functionality. We note the location of several of these systems in Appendix E, *Electronic Resources*.

Figure 8.5: S/Key password printout

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.