User authentication : a state-of-the-art review.

Coley, John A.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

USER AUTHENTICATION:
A STATE-OF-THE-ART
REVIEW

by

John A. Coley

September 1991

Thesis Advisor: Moshe Zviran

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (*If applicable*) 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION 8b OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number
11 TITLE (*Include Security Classification*) USER AUTHENTICATION: A STATE-OF-THE-ART REVIEW	
12 PERSONAL AUTHOR(S) Coley, John A.	
13a TYPE OF REPORT Master's Thesis 13b TIME COVERED From To	14 DATE OF REPORT (*year, month, day*) September 1991 15 PAGE COUNT 124
16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	

17 COSATI CODES			18 SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	Access Control; User Authentication; Passwords; Information System Security

19 ABSTRACT (*continue on reverse if necessary and identify by block number*)

Access control of computing systems is considered a key issue among Information Systems managers. There are different methods available to computing systems to ensure proper authentication of a user. Authentication mechanisms can use simple user-generated passwords or complicated combinations of passwords and physical characteristics of the user (i.e., voice recognition devices, retina scanner, signiture recognition devices, etc.).

This thesis looks at the various authentication mechanisms available to a security manager. It describes how different authentication mechanisms operate and the advantages and disadvantages associated with each mechanism. It also reports on several commercially available software products that support the user authentication process. Finally, a discussion of password use in the military environment and the unique requirements of the Department of Defense.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Moshe Zviran	22b TELEPHONE (*Include Area code*) (408) 646-2489	22c OFFICE SYMBOL AS/ZV

Approved for public release; distribution is unlimited.

User Authentication:
A State-of-the-Art
Review

by

John A. Coley
Lieutenant Commander, United States Navy
B.S., Mississippi State University 1977

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Department of Administrative Sciences

ABSTRACT

Access control of computing systems is considered a key issue among Information Systems managers. There are different methods available to computing systems to ensure a proper authentication of a user. Authentication mechanisms can use simple user-generated passwords to complicated combinations of passwords and physical characteristics of the user (i.e., voice recognition device, retina scanner, signature recognition device, etc.).

This thesis looks at the various authentication mechanisms available to a security manager. It describes how different authentication mechanisms operate and the advantages and disadvantages associated with each mechanism. It also reports on several commercially available software products that support the user authentication process. Finally, a discussion of password use in the military environment and the unique requirements of the Department of Defense.

TABLE OF CONTENTS

I. INTRODUCTION

The proliferation of computer technology has bred opportunities for ill-intentioned individuals to violate the integrity and validity of computer-based information systems (IS). At the same time, a growing dependence on computer-based information systems creates an urgent need to collect information and render it accessible. (Zviran and Haga, 1990a)

The developments in computers and communications technologies in the last two decades has made most organizations susceptible to misuse or abuse of computer-based information systems. While information systems can provide an improvement in an organization's functioning and enhance its services, they can also expose that organization to greater risks as they become more dependent on information resources. Recent surveys of top IS management issues indicate that security is considered a top concern (Brancheau and Wetherbe, 1987; Dickson et al, 1983). Therefore, the amount of information security that an organization requires to protect its computing facilities and information resources is a key management issue (Wilson et al., 1990).

It is believed that only about 15 percent of computer crime ever comes to the attention of law enforcement agencies (Carroll, 1987). According to recent studies, American

businesses lose $3-5 billion each year in computer fraud (Lewis, 1987). In England, a report issued by the Audit Commission indicates that companies are still ignoring the threat posed by poor computer security (Lauchlan, 1991). This report recounts that 40% of all breaches of security reported involve computer fraud (Lauchlan, 1991).

Access control ensures that unauthorized users do not gain entry into a computer system, as well as preventing a legitimate user from performing a function inside the system that he/she is not allowed to do (Wood, 1983). An IS manager can approach access control with external and internal methods. External mechanisms include such methods as making physical access difficult by use of guards, locks, or some type of token (Ahituv et al., 1987). Internal controls are aimed to prevent unauthorized tampering with data. These controls are designed to prevent users from accessing segments of memory to which they are prohibited. While access control is one way of providing internal security and control, there are other specific approaches that can be used in conjunction with access controls to prevent an unintended intruder (Denning and Denning, 1979).

The focus of this thesis is to survey the various user authentication mechanisms for improving computer security. Chapter II is an overview of user authentication and a description of authentication mechanisms and their importance. Chapters III and IV deal with traditional and advanced

password mechanisms respectively. These chapters describe the characteristics of user-generated passwords, machine-generated passwords, passphrases, authentication by word association, cognitive passwords, and authentication servers and discuss their advantages and limitations. Chapter V is an evaluation of traditional password and advanced password mechanisms. Chapter VI discusses current password use in practice. Alternative authentication mechanisms are discussed in Chapter VII, to include automatic call-back procedures, authentication servers, token and smart card use, and information on biometrics. Chapter VIII includes a discussions on password encryption and the use of various commercially available software products that can assist an IS security manager in strengthening an organization's authentication procedures. Finally, Chapter IX deals with password use in the military. This chapter focuses on National Security Agency (NSA) guidelines for password use, as well as similarities and differences between civilian and military applications.

II. USER AUTHENTICATION OVERVIEW

A. AUTHENTICATION OVERVIEW

User identification is the process by which an individual identifies himself or herself to a computer-based information system as a valid user. User Authentication is the procedure by which a user establishes that he or she is indeed that user, and has the right to use the system or portions of it. A simple authentication system can effectively prevent the casual prowler from poring through the system. (Hutt, 1988)

Most operating systems have applied stringent security measures to lock out illegal users before they can access system resources. User authentication mechanisms are an important ingredient in these security schemes.

Authentication mechanisms are divided into three categories:

- What they know, such as a password or an encryption key
- What they possess, such as a token or a capability
- Something about you, such as a picture or a fingerprint (Wood, 1983; Spender, 1987; Weiss, 1990)

Authentication usually operates in the following manner: a user enters some piece of identification, such as a name or an assigned user-ID. This identification can be available to the public (e.g., when it also serves as the user's e-mail

identification) or easy to guess (e.g., a user's first name). Thus, it does not provide security for the system. To authenticate a user, the system requests further information (e.g., a password). If the authentication information matches that on file for the user, the user is granted access to the system. A mismatch leads to a denial of access (Wood, 1983). Use of ATMs is an example of a combination of something a user has (a plastic card) and something a user knows: the Personal Identification Number (PIN), which serves as a password.

B. AUTHENTICATION MECHANISMS

Several authentication mechanisms are available to computer users. The most common mechanism used today is a password. Traditional password mechanisms fall into two categories: user-generated or system-generated. More advanced password schemes include passphrases, associative passwords, cognitive passwords and authentication servers. Alternative authentication mechanisms include automatic call-back procedures, authentication servers, tokens and smart cards, and biometric devices.

Some authentication procedures are purposely slow. However, while it is not inconvenient for a legitimate user if the authentication procedure takes 5 to 10 seconds, a brute force attack on a system that requires 5 or 10 seconds per attempt makes this class of attack generally infeasible. (Pfleeger, 1989)

There are other ways to discourage unauthorized users. If someone fails to log-in after several attempts, it is common for the system to disconnect, forcing a user to reestablish a connection with the system. This method will slow down a penetrator from attacking the system.

In a more secure environment, stopping a penetrator may be more important than tolerating users' mistakes. In such cases a system may limit password entries to three tries, assuming that legitimate users can type their passwords correctly within three tries. After a third successive password failure, the account is disabled, and can only be reenabled by the security administrator. This action, while inconvenient due to denial of services, may help in identifying accounts that may be the target of attacks by penetrations.

The underlying assumption in password authentication assumes that only a user to whom the password belongs knows it. However, passwords can be guessed, deduced, or inferred. One method of uncovering a users password is to simply ask him/her for it. Other passwords have been obtained by watching a user typing in his/her password. This causes flaws in the authentication process. (Menkus, 1988)

Advanced password mechanisms help eliminate the way an intruder can obtain a password. By requiring more than a single password, advanced authentication mechanisms provide a level of security that is desirable to many IS organizations.

In addition to passwords, there are several alternative mechanisms available to assist user authentication. These include tokens, smart cards and biometric devices such as handprint detectors, voice recognizers, and identifiers of patterns in the retina. Although expensive and in some cases still experimental, these devices are useful in very high security situations. (Wilson, 1987)

The next chapters discusses the various authentication mechanisms, providing definitions, examples, and analysis of their advantages and disadvantages.

III. TRADITIONAL PASSWORD MECHANISMS

A. PASSWORD OVERVIEW

A basic access control routine in any computer operated system is to ensure proper user authentication. The most commonly used authentication method is the password. Developers of password access control packages have used an analogy of a fence or wall protecting a valued physical asset. Once a user has presented the correct password, he or she essentially passes over this logical fence and gains access to the information system (Wood, 1987). In a large number of computer systems, passwords are the first line of defense against unauthorized persons trying to gain access to computer resources. Sometimes it might be the only line of defense. As such it is imperative that this defense be as formidable as possible (Wood, 1983).

Passwords are formally defined as "any sequence of letters, numbers, special symbols, or control symbols,...non-printing, that are used to authenticate a computer user's identity". (Riddle et al., 1989)

In some cases a user chooses a password, while in others a password is assigned by the system. The composition of passwords varies from system to system. The effectiveness of a password is dependent on the balance to be struck between

the ease with which it can be remembered and the difficulty with which it can be guessed by an unauthorized party. In theory, the only person who should know the user's password is the user. (Riddle et al., 1989)

Passwords are considered to be of limited usefulness as protection devices because of the relatively small number of bits of information they contain. However, despite horror stories associated with password use, researchers say that passwords can provide ample security if managed and handled properly. (Betts, 1985)

An underlying goal of password security is to cause minimum inconvenience to users. As the first line of defense, a password security system should prevent unauthorized entries as well as preventing authorized users from engaging in unauthorized activities. (Morris and Thompson, 1979; Jobusch and Oldehoeft, 1989; Riddle et al., 1989)

Passwords should be hard to guess and hard to determine exhaustively. A fundamental dilemma of password selection is that easily remembered passwords are easy to guess; the hard-to-remember passwords get written down and therefore can be misused or stolen. (Highland, 1990)

In the case of an occasionally mistype of passwords, a user should receive a message of INCORRECT LOG-IN at which point the log-in procedure should be repeated to gain access to the system. Even the worst typist should be able to log-in successfully in three to five attempts. (Menkus, 1988)

Operating systems often encourage users to change their passwords regularly. Password aging, or the enforcement of a maximum password lifetime is one method of automatically forcing users to change their passwords. Such mechanisms can typically enforce a minimum and maximum amount of time between password changes. The regularity of password change is usually a system parameter, which can be changed for a given installation.

While password aging may seem like a good idea, many argue that it is counter-productive. Users do not like to change passwords; systems requiring them to do so may cause frustration. Mechanisms that do not warn of an upcoming expiration of the password can actually decrease security, as such a mechanism may suddenly demand that a new password be set. Such practices will probably not result in the best password choice, and most likely will be written down as well. Systems supporting minimum lifetimes can actually stop users from changing their passwords. Minimum lifetimes are primarily used to keep users from "cheating" the aging system, by changing to a temporary password, and then back to the old one (Jobusch and Oldehoeft, 1989).

Changes should be made periodically, depending on the security classification of the information to which they afford access. Passwords to special control information should be used once only. Passwords to confidential information should be changed daily. Passwords to private

information should be changed weekly. Other passwords can be changed as desired, but this should be done no less frequently than once every six months. (Carroll, 1987)

Auditing, when used with password mechanisms, is used to record events that occur during authentication attempts. Information collected by auditing software includes:

- successful log-in and log-out information
- unsuccessful log-in information
- successful password changes
- unsuccessful password changes
- number of currently active sessions. (Jobusch and Oldehoeft, 1989)

Using this information provides a method of detecting a perpetrator using a stolen account, as well as attempted breakins. Audit information can also be used to deactivate a port or a username if a high rate of authentication failure is detected. (Jobusch and Oldehoeft, 1989)

Monitoring of a password system should not be left to those times when serious violation occurs. Even though this operation might be repetitive and boring, it is essential to maintaining security. (Highland, 1990)

B. USER-GENERATED PASSWORDS

Since a password has to be remembered, people tend to pick simple passwords (Pfleeger, 1989). If a user is picking a

password, he or she is probably not choosing a word completely at random. Most likely a user's password is something meaningful to him or her. People typically choose personal passwords, such as the name of a spouse, a child, a brother or sister, a pet, a street name, or something similar.

It would be easy to select a password by picking two short words and separating them by punctuation, digits or control characters. Here are a few examples:

rich$gal,

poor*boy,

out#sick,

home%run

big6!mac.

Another route to strong passwords is to select a common and easily remembered phrase, eliminating the spaces between words and truncating after the required number of characters. Here are some examples: "He is a dud" becomes "heisadud," "Peter Piper" becomes "peterpip," and "floppy disk" becomes "floppydi."

Foreign languages also work very well as passwords. For example, try "thank you" or some other phrase in the foreign language. Or truncate a translation as necessary. (Highland, 1990)

Practical guidelines regarding password selection include:

- Use more then A-Z. When using A-Z, there are only 26 possibilities per character. Adding digits expands the number of possibilities per character to 36. Using both upper and lower case letters plus digits expands the number of possibilities per character to 62.

- Choose long passwords. The combinatorial explosion of passwords begins at length 4 or 5. Choosing 6-character or longer passwords makes it less likely that a password will be uncovered.

- Avoid actual names or words. Theoretically there are about 300 million "words" (i.e. any combination of characters) of length 6, but there are only about 150,000 words in a good collegiate dictionary, ignoring length.

- Choose an unlikely password. In order to remember the password easily, you want one that has special meaning. However, you don't want someone else to be able to guess this special meaning.

- Change the password regularly. Even if there is no reason to suspect that the password has been compromised, change is advised.

- Don't write it down.

- Don't tell anyone. (Pfleeger, 1989)

User-generated passwords are popular because they are conceptually simple, relatively inexpensive, easy to administer, and user friendly. (Wood, 1990)

Disadvantages of user-generated passwords are many. First, if a user makes the password as secure as possible, he or she tends to write it down so as not to forget it. By doing so a user is leaving it open to compromise. Second, if a user does not put effort into selecting a password by choosing a familiar name or trivial association it makes it

easy for an intruder to figure out. Third, even if a good password is chosen, if a user keys it in slowly or allows someone to watch as it is keyed in, it is then subject to compromise. And finally, there are many ways for an intruder to unfold an operating system in order to find the password table and decipher it or use some method to capture the password. (Ahituv et al., 1987)

C. MACHINE-GENERATED PASSWORDS

A password generator is a program that creates strings to be used for machine-generated passwords. Such programs are made available on systems in an effort to ensure "good" password choices. How to design a password generator that produces passwords that are both difficult to guess and easy for a user to remember is not immediately apparent. While it is easy to generate random strings to be used as passwords, they most likely will not be easy for a user to remember. Also, password generators that are not sufficiently random in the method in which they select passwords may be limited in the number of passwords they can generate. (Jobusch and Oldehoeft, 1989)

A password generator that can produce pronounceable passwords is desirable, as these passwords are more likely to be remembered than a random string of characters. Such a system depends on a set of rules that "define" what pronounceable means. A sample set of rules may include:

- a consonant must be followed by a vowel of any type
- a vowel may be followed by a vowel of a different type, or by a consonant
- never have more than two consecutive vowels of any type. (Jobusch and Oldehoeft, 1989)

Typically, a random number generator is used to select random letters or groups of letters that are considered pronounceable. These groups of letters are then linked together to form the password. While the resulting "word" may not be recognizable, it should be pronounceable in the way it is constructed. (Jobusch and Oldehoeft, 1989)

Machine-generated passwords are attractive because they can ensure that passwords are relatively strong (Wood, 1990). It is far easier to maintain control over password selection in a system wherein passwords are machine-generated and assigned to users than one in which users may select their own passwords (Highland, 1990).

Unfortunately, the disadvantage is that such passwords are difficult for a user to remember and because of that they run a greater risk of compromise (Highland, 1990; Wood, 1990). This difficulty leads to two problems. First, a user who forgets his or her password must bother the system manager for a new one. Secondly, one important security dictum says, "Never write down your password." In machine-generated password systems there is no effective way of keeping users from violating the dictum (Highland, 1990; Wood, 1990).

Machine-generated passwords are also problematic because the results of some pseudo-random number generators used to produce them are readily reproducible by an informed opponent who has access to the algorithm employed. To make matters worse, in an attempt to generate pronounceable words, many machine-generated password mechanisms significantly reduce the number of possible passwords. This reduced pool from which to pick machine-generated passwords in turn reduces the required effort to guess a valid password. (Wood, 1990; Bishop, 1991)

Because this fact of human nature, some security directors have decided to stop using machine-generated passwords and go back to user-selected ones (Highland, 1990).

D. PROBLEMS WITH TRADITIONAL PASSWORDS

There are several problems associated with traditional password mechanisms. First, passwords can be intercepted at the point of entry. With most systems the password is not displayed as it is entered, so an interceptor must watch your fingers as opposed to the screen (Betts, 1985). Even by watching your fingers, if the interceptor can not get all the characters he maybe able to pickup the pattern for entry. Even such information as left-right-right-shift-left-number would substantially reduce the possible passwords to try (Avarne, 1988).

Second, passwords are in the clear from the moment they are entered until the moment they are accepted by the host

computer. Therefore, if someone has tapped into the communication line in a network, they can intercept a user's password. Tricking novice users with false log-in programs that steal passwords is another way of abusing insecure communication lines. Such programs point out another transmission concern: how do users know if they are really communicating with the host? Outside of using secure and/or encrypted communication lines, the insecure transmission of password information can be a serious weakness in any password mechanism. (Jobusch et al., 1989)

Traditional passwords discussed in this chapter have three fundamental weakness. They are:

- They can often be guessed.
- They are entered in the clear.
- They are used more than once. (Avarne, 1988)

All of these problems occur because a traditional password is static. Frequent change of passwords are desirable. To foil these kinds of attacks, the password has to be changed almost immediately after it is used. Few human users can handle so frequent password changes and so only change passwords when required by the system. (Pfleeger, 1989)

The password schemes examined in the next chapter try to alleviate these weaknesses. Chapter IV deals with more advanced schemes for password security including passphrases and question and answer schemes. Question and answer schemes include word association and cognitive passwords.

IV. ADVANCED SCHEMES FOR PASSWORD SECURITY

Advanced password security schemes are more than just a string of several alpha-numeric characters. They may include a series of questions and answers or may be a passphrase that a user must recall in order for the authentication process to be completed. Figure 2. depicts the various types of advanced password schemes.

The advantage of an advanced scheme is it can provide better security for computer systems than traditional password schemes (Smith, 1987; Zviran and Haga, 1990b).

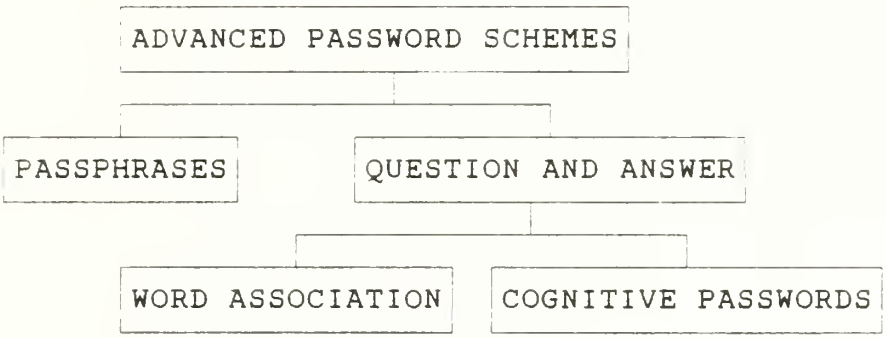

Figure 2. Advanced Password Schemes

A. PASSPHRASES

A passphrase is one form of authentication that is just a longer version of a password. It involves arbitrary selection of easily remembered, but very likely to be meaningless, three or four word phrase from previously defined lists of adjectives, nouns, verbs and such. An example of a passphrase might be "big girls drink wine." A passphrase is used as though it were a single password. (Menkus, 1988)

When users select their own passwords, they are more likely to remember them, but chosen passwords can often be easily guessed. Thus, if passwords are generated with the use of a pseudo-random number generator that include both letters, numbers, and control characters, they are considerably more secure, although unpopular with users. Passphrases then seem to be a relatively attractive compromise between ease-of-use and high security. (Wood, 1983)

Passphrases are easily to remember because they are user selected and are more than just as random collection of characters. But there are problems associated with passphrases. Even though a user may recall the passphrase without writing it down, a frequent user may become upset at the prospect of typing such a long string of characters every time they desire to use the system (Porter, 1982). Therefore systems should not require a minimum password length that is unreasonably long. Passphrases may in fact be an inadequate means of authentication for frequent users (too long) but can

prove useful to those who experience long intervals between log-ons.

Another problem with passphrases is the same as with simple passwords, passphrases may need to be checked for triviality; choices like "mary had a little lamb" will most likely be guessed (Porter, 1982; Jobusch et al., 1989).

B. WORD ASSOCIATION

The practical need of computer security is to identify users quickly and reliably by some process which does not handicap effective computer use. For that purpose a test is needed with several characteristics. The test must elicit individualistic responses from different users, so it can identify any particular user quickly. The test should be based on easily remembered material, so that it imposes little memory burden on a user and can be reliably administered. And ideally the test, by its nature, should elicit user interest and cooperation rather than resentment. The key to meeting these requirements is to devise a test that enlists the cooperation of users themselves. (Smith, 1987)

One promising approach for verifying user identity would be a question and answer test called word association. But it must be a personalized test whose contents are specified by each individual user, rather than one general test applied by the computer to all users. Word association can be unique to an individual, if they are chosen for that purpose (i.e.

avoiding such common associations as "black-white"). Word association could be quite strong, and if chosen carefully could be remembered without particular effort. And if each user is permitted to specify his/her own association, there would be very little user resistance expected to this kind of testing.

Word association would work in the following manner. At initial enrollment a new user is asked to provide the computer with a list of 20 cues (words or phrases) along with a response that the user associates with each one. The computer would store those cue-response association safely away. Then on subsequent attempts to access the computer, the computer would select a cue at random and challenge the candidate user to give the stored response, repeating that process as necessary to confirm the user's claimed identity. Depending upon the computer assessment of risk a user might be required to give one response or several, but presumably not all 20. (Smith, 1987)

The cue list would look different for every user. A cue list and response list could be generated by a user in less then 30 minutes. The critical decision in creating a list is to choose a context in which associations are already established. Probably most people will find it easier to remember associations if they choose single integrating context for their entire list, although that is not strictly necessary to the process.

Users should select single word responses for greater certainty of memory and for greater ease of entry in response to displayed cues.

Most users have rich associations with people and places, and thus it is probable that most spontaneously created cue lists would have proper names as their specified responses. However, that is not strictly necessary to the process; as a response the user could choose any word that he/she is reasonably certain to remember. It is not necessary that a response be "correct" in any sense except that it matches whatever response was initially stored for that cue. To pass this test, the user does not have to remember the state flower of Alaska or any other factual data. Note also that a users personal association will not necessarily correspond to general stereotypes. (Smith, 1987)

Smith conducted tests six months after users provided a list of cues and responses to test the ease with which word association could be remembered. When asked to reproduce their cue list most people had problems recalling the list. One person could not remember any of the cues. The others each managed to remember some, although not with complete accuracy.

These people were then each shown their original cue lists and asked to recall the correct responses. They remembered almost all responses correctly and with little difficulty,

averaging 94 percent correct, indicating persistent strength of their word associations even after a six-month interval.

Twelve months later (i.e. 18 months after list creation), the same people were tested again. They were first asked to recall the cues that they had devised, which they could do with only partial success. They were then asked to recall responses to their original cue lists, which they could still do with reasonably good accuracy, averaging 86 percent correct. (Smith, 1987)

Good recollection of such word association could be expected on several grounds. The responses were cued, which should aid recall. The material to be remembered was generated originally by each individual tested, and self-generation is know to aid recall (McFarland et al., 1980; Slamecka and Graf, 1978). Most responses reflected personal rather than impersonal associations, which also aids recall (Rogers et al., 1977).

Assessing memory after such long periods of disuse represents an extreme test. In reality, a user's memory would be tested more frequently, perhaps as often as daily or even more frequent with certain users. Thus through repeated use many users would come to remember the cues as well as the responses. On the other hand users need not remember the cues for associative testing to work. Certainly a user would not need any printed record of his/her cue list, nor any display of the entire list except perhaps when changing it. And there

would be no need to display for any purpose the stored responses that have been specified for cues.

User may at times forget the proper response to a particular cue. If that were to happen, the user presumably could authenticate his/her identity by responding correctly to several other cues, however many the computer logic might impose in the interest of adequate security.

Authenticated users should be able to change their cue lists from time to time, just as they might change a password. Thus if a user noticed that a particular association was not easily remembered, he/she might choose to change it. Or perhaps the computer could keep track of user errors and draw a troublesome association to a user's attention for potential revision. It is possible that with changing interests a user might wish to change a list simply because it has become boring.

Considered overall, the potential value of word association for user authentication seems promising. This method imposes only modest demands on the user, and it would require relatively little computer logic to implement. Simple interactive software routines that would accomplish the basic functions of initial user enrollment, subsequent cue-response exchange for authenticating user identity, and occasional user revision of stored cue lists and associated responses would be needed. More complex software might be needed for risk assessment, flexibly controlling the stringency of

authentication procedures. But a simple matching logic will suffice to process the word associations on which user authentication is based. (Smith, 1987)

C. COGNITIVE PASSWORDS

Cognitive passwords as a security scheme evolved from Smith's (1987) work with word association. Instead of challenging a user with a single list of word association cues, cognitive passwords challenge a user with a set of five randomly selected cognitive questions out of a pre-selected set of questions and answers (Zviran and Haga, 1990b).

Cognitive password systems and word association systems are similar in that they both ask a user to provide the data upon which their passwords are based. The password challenges consist of fact-based and opinion-based cognitive data that only a user should know. A fact-based question asks a user something that the user knows but which is a fact independent of any feeling about it. For example, "What is the name of the elementary school that you last attended?". An opinion-based cognitive item asks for an opinion about something: "What is your favorite type of music?" or "What is your favorite flower?"

A cognitive password system will combine both user-generated and system-generated characteristics. It is system-generated in that the security administrator creates question that would be used to generate a response from a user. The

exact response to these questions would be entirely user-generated. Hence, a cognitive password system is basically an access quiz. If the user responds correctly to a series of questions concerning him/herself, he/she would then be authorized access to the computer system. (Zviran and Haga, 1990b)

There are several advantages associated with a cognitive password system. First, since the cognitive password is significant to the user, but not readily associated with him or her, it is easy for a user to remember, but difficult for an intruder to guess. Second, the responses may be of such length that a brute force attack would be thwarted. And finally, a cognitive password system requires several questions to be answered correctly, so this layering adds an additional degree of security.

There are also several disadvantages associated with a cognitive password system. First, users of the traditional password system have a tendency to forget a single password, therefore remembering many cognitive passwords would seem to be harder for the user. Secondly, it is unlikely that a user would remember all of his responses so establishing an acceptable miss percentage may be difficult to do. If it is set to low, an intruder may penetrate the system; if it is set to high, authorized users may be denied access. (Haga et al., 1989)

V. EVALUATING PASSWORD MECHANISMS

Several studies have been conducted to determine the memorability of traditional password and advanced password mechanisms and their susceptibility to guessing by someone close to the user. These studies are discussed in the following section.

The second section of the chapter discusses empirical evaluation of password usage and the composition of those passwords.

A. COMPARISON OF PASSWORD MECHANISMS

Beedenbender (1990) conducted research into the recall of traditional passwords and advanced password schemes. Several different questionnaires were used with the intent either to verify information from previous studies or to justify conclusions about new areas of study.

The respondents answered two versions of the original questionnaire and a significant other (spouse, close friend or family member) completed a second form of the questionnaire. The questionnaire asked for the respondent's sex, years of computer usage, types of computers with which they were experienced and a respondents identifier (i.e., Social

Security Number (SSN)). The SSN was used to tie all the questionnaires together with the correct respondent. (Beedenbender, 1990)

The second part of the questionnaire asked the respondents to create a password of up to eight alphanumeric characters. The test group was asked to memorize and safe guard this password. They were then asked how they devised this password. The second part of the questionnaire contained a unique, eight character, system-generated password. Fifty-five of the questionnaires had a system-generated random alphanumeric password while the other forty eight questionnaires had a system-generated pronounceable password.

The second part of the questionnaire asked the respondents to create a passphrase consisting of any combination of up to 80 alphanumeric characters. There was no minimum requirement for the passphrase. The respondents were urged to memorize and safeguard this passphrase like any other password. They were then asked how they derived this passphrase.

The questionnaires were identical in their third part. In this section, 20 open response questions ask for items of information that were described as cognitive passwords. The first group consisted of six personal facts assumed known only to the respondent or someone socially close to the respondent. The second group asked 14 opinion-based questions (i.e., favorite fruit, favorite food, etc.).

The final part of the questionnaire requested the respondent to come up with a list of 20 word associations. The respondents were not required to use a central theme throughout nor was there any limitation or minimum number of alphanumeric characters in either the cues or responses. The respondents were then asked to copy just the cues onto another questionnaire to see if a socially close person would be able to figure out the responses. (Beedenbender, 1990)

Three months after the initial questionnaires were completed, the respondents were asked to recall the password they selected, the system-generated password, and the passphrase they supplied. They were also asked the method of recall used.

In the identical version of the cognitive password section, the same respondents were asked the same questions again.

In the identical version of the word association section the respondents were asked to regenerate their list of 20 cues and responses. As soon as the respondents had generated as many associations from memory as possible, they were given a list of their initial 20 cues. They were then asked to generate as many responses as they remembered. If, at his point, they were still unable to remember their responses, they were given the central theme, if any, to aid them correctly remembering their responses.

The final section of the questionnaire requested the respondents to rank the various password methods by ease of memory. The respondents were then asked to rank the methods by how they liked them.

Another questionnaire was then given to the significant-other. The first part of the questionnaire asked for the respondents SSN and the relationship of the significant-other to the respondent. The second part of the questionnaire repeated the 20 cognitive password questions. The significant-other was asked to indicate what they thought the respondents would answer to each of the questions. They were asked to complete the form without help from the respondents.

The final part of the significant-others questionnaire asked them to determine the responses to the cues written down by the respondents from the word association portion of the initial questionnaire. After attempting to figure out the correct responses without aid from the respondent, the significant-other respondent was given a second chance. This time the respondent would inform the significant-other if there was a central theme to the association and if so, what it was.

In both the cognitive password and word association sections it was assumed that if someone socially close to the respondent was unable to figure out the correct responses, then the chances of an intruder figuring out the responses would be slim.

Over the three month period only 27.2% of the respondents could recall the password that they had created themselves. As in previous research studies, this research showed that as password length increased it became more difficult to remember. (Beedenbender, 1990)

Similarly only 12.7% of the respondents could recall their system-generated alphanumeric password. However, the respondents assigned a system-generated pronounceable password, 37.5% were able to recall it. It should be pointed out that not one respondent was able to remember the random alphanumeric password on his own. Among those who did recall it, 85.7%, had written it down.

Only 21.4% of the respondents were able to remember their passphrases. Most of the respondents, 77.7%, choose passphrases consisting of fewer than the minimum recommended thirty characters (Porter, 1982). This did not help them in recalling their passphrase.

After the three month period, the respondents recalled an average of 74% of their cognitive passwords. Two respondents were able to recall all 20. The recall average of the fact-based cognitive passwords was 83%. The opinion-based cognitive password recall was some what less, only 74.8%. The people socially close to the respondents could guess no more than an average of 38% of the respondents' cognitive passwords. Only a few significant-others could legitimately guess more than 10 out of 20 responses. Two significant-

others could not guess any of the 20 responses correctly. (Beedenbender, 1990)

The guessing of fact-based cognitive passwords showed the significant-other able to correctly respond to 44.8% of the questions while averaging only 32.5% for the opinion-based cognitive passwords.

The notion that people more socially close to the respondents are better guessers than those even slightly removed, was found to be true. The average number of correct guesses for family members was 60%, while spouses were 41% and friends were 23.5%. (Beedenbender, 1990)

On average, the respondents recalled 69% of their word association. Seven respondents recalled all 20 responses and almost a third remembered 90% or more of their responses. While there was success at the high end of the spectrum, there was a fairly uniform distribution of respondents remembering from 30%-90%. This distribution may be explained by the fact that respondents were given a free reign in making up their word association. Unlike the cognitive password section, in which all the respondents answered the same questions, the word associations had various degrees of difficulty depending upon how challenging each respondent decided to make them. (Beedenbender, 1990)

Even with the wide variance, the average success rate was over twice that of the user-generated password method. In comparison with the overall success rate of cognitive

passwords, word associations were not as great (69% to 74%). However, there were twice as many respondents scoring 90% or more correct responses on the word associations than on the cognitive passwords.

The significant-others, on average, could guess only 25.5% of the correct responses. Seventeen significant-others could not guess even one response correctly. A small percentage of significant-others (10.3%) were able to guess ten or more responses correctly. When significant-others knew the central theme the success rate improved to 33%. There were still six significant-others who could not guess any correct responses.

Even with the theme, the significant-others failed to guess as many correct responses (33% to 38%) as they had in the cognitive passwords section. Also, unlike cognitive passwords, social closeness made no significant difference in the ability of the significant-other to figure out the responses. (Beedenbender, 1990)

When ranking the various methods as to how easy they were to remember, the respondents clearly chose user-generated passwords as the one that they thought was easiest. However, this method was one of the worst for recall by the respondents. Other than this, the rankings generally reflected how the respondents actually did in recalling their passwords from the different methods. (Beedenbender, 1990)

In another study, Zviran and Haga (1990) conducted tests of the memorability of cognitive passwords and their

susceptibility to guessing by people close to the users. At the same time they tested the recall of system-generated passwords (random alphanumeric seven-character strings) and user-generated passwords.

The study included the use of three similar versions of a self-administered questionnaire. The primary respondents in the study, called user-respondents, answered a first questionnaire to determine their age, sex, years of computer usage, the types of computer which they have used (mainframe, stand-alone micro or micro linked to mainframe) and the last four digits of their Social Security Number (SSN). The SSNs were used to hide the identity of the respondents while allowing for a method of linking future questionnaires together.

The second part of the questionnaire asked the user-respondents to create a password of up to eight alphanumeric characters. They were urged to memorize and safeguard it like any other password. They were then asked how they devised it. Finally, in this part of the questionnaire they were assigned a system-generated unique seven-character password. These passwords were constructed of random combinations of letters and numbers. The respondents were asked to memorize and safeguard this password as well.

The third section of the questionnaire asked for twenty open-response, cognitive items. This information fell into two groups. In the first group were six items that asked for

personal facts that assume only a respondent or someone socially close to a respondent would know. For example: elementary school attended, first name of favorite uncle, first name of best friend in high school, mother's maiden name, first name of first boyfriend/girlfriend or father's occupation. In the second group were 14 opinion-based items that asked each respondent to declare favorites. For example: favorite music, favorite color, favorite flower, favorite vegetable and favorite dessert. Again it is assumed that only someone close to the respondent would know the responses. (Zviran and Haga, 1990b)

The first questionnaires were completed and three months later the respondents were given a second questionnaire. This questionnaire repeated the same 20 question from the previous form. It also asked the respondents to recall the password they created and the system-generated password they were assigned. They were then asked if they had recalled the passwords from memory or had resorted to writing them down.

The significant-other version of the questionnaire asked for only two items of identifying data: the last four digits of the user-respondent's SSN and the relationship of the significant-other to the respondent. The remainder of the questionnaire repeated the 20 cognitive question from the first two forms. Each significant-other respondent was asked to complete the questionnaire without help from their user-respondent friends or spouses. They were asked to guess what

he/she thought his or her user-respondent would answer to each question. They were also asked to answer only questions that they were confident of their responses, leaving blank those items where they would have to guess widely. This was done in order to see how well the significant-others could guess the responses of the user-respondents. Assuming that if people who were socially close to the users showed deficient knowledge of personal data, then someone who was socially distant from the same user would be unlikely to guess cognitive passwords. (Zviran and Haga, 1990b)

The results showed that after the three month interval, respondents were better able to recall conventional passwords that they created than they were at recalling passwords that were assigned to them. A minority of the respondents wrote down the self-created passwords while most of them wrote down the passwords assigned in order to aid in remembering the passwords.

The average number of correct recalls by the user-respondents on all cognitive data questions was 82 percent. That equates to 15 to 17 correct responses. Compare the level of these responses with responses for the two type of conventional passwords recalled over the same period and the best response was 35 percent for self-generated passwords. On the cognitive data continuum, that equates to only 7 correct matches. No respondent scored that low on cognitive data.

The significant-other questionnaire showed an accuracy of only 27 percent correct responses of all cognitive items. The correct responses were skewed towards the low end of the scale. Only one person was able to guess 10 out of 20 cognitive questions correctly. Moreover, a comparison of the profile of this distribution with that of the user-respondents showed no overlap between user-respondent responses and significant-other responses.

The significant-others were able to guess only 37 percent on average of fact-based items. It is assumed that significant-others would know fact-based items better than they would know opinion-based items. That appears to be true since significant-others guessed only a third of opinion-based items. Even though they are precisely the people who should know better than anyone the personal facts about user-respondents.

Assuming that significant-other are in the best position to possess personal knowledge about the user-respondent then the accuracy of personal knowledge will decrease if even the slightest social distance is introduced. This was proven out when the number of correct guesses by spouses were compared to correct guesses by friends. The spouses guessed correctly on 29 percent of the items while the friends guessed correctly on only 16 percent of the items. (Zviran and Haga, 1990b)

Over the three month period, the study showed that the recall of cognitive items was noticeably better than the

recall of either the self-created or assigned passwords. Cognitive passwords were recalled 82 percent of the time while self-generated and system-generated passwords were recalled 35% and 23% respectively.

B. EVALUATION OF PASSWORD USE

Several studies have been conducted to assess usage patterns of traditional passwords.

Morris and Thompson (1979) conducted experiments to determine typical users' habits in the choice of passwords when no constraint is put on their choice. The results are disappointing. In a collection of 3,289 passwords Morris and Thompson found, 15 were single ASCII character; 72 were strings of two ASCII characters; 464 were strings of three ASCII characters; 477 were strings of four alphamerics; 706 were five letters, all upper-case or all lower-case; 650 were six letters, all lower-case. An additional 492 passwords appeared in various dictionaries, name lists, and the like. A total of 2,831 or 86 percent of this sample of passwords fell into one of these classes.

There was overlap between the dictionary results and the character string searches. The dictionary search alone, which required only five minutes to run, produced about one third of the passwords. (Morris and Thompson, 1979)

In another study conducted in 1987 at Syracuse University using the university's timesharing system, 6226 user-selected

passwords used to authenticate 7014 computer user identities were compared. Researcher found that if left on their own, people are lazy about passwords, relying on easy to remember passwords such as initials, pronouns, nouns, or mnemonics, and unless forced to, people do not change their passwords on a regular basis. People prefer three to five character passwords to seven or eight character passwords. Only a small number of users seek complex passwords using number and letters in all eight positions. Only 15% of the passwords are repeated. The majority of passwords are as unique as the people who created them. About 30% of all the passwords that are user created use a true English word; an additional 10% can be assigned a part of speech based on the English word found in them. If two- or three-character passwords are excluded, about 44% of the passwords use a true English word while approximately 60% can be assigned a part of speech. (Riddle et al., 1989)

In a study conducted by Zviran and Haga (1990) of 997 self-generated passwords they found that 80.1 percent of the passwords consisted of alphabetic characters only, 13.7 percent of alphanumeric characters, 5.5 percent of numeric characters, and 0.7 percent of ASCII characters. The average number of characters in a password, calculated from the password lengths in the study, was six. Thirteen percent of the passwords consisted of eight characters, 14 percent of seven characters, 25 percent of six characters, 24 percent of

five characters, 17 percent of four characters, and 5.6 percent of three characters. (Zviran and Haga, 1990)

One survey, conducted recently at a government agency, found that 43% of the agency's 1,500 employees used two-character passwords (probably their initials), and over 25% used a single character. Compliance with good practice is no better in business. A survey of 50,000 users in several private companies revealed that about 20% used single character passwords. (Highland, 1990)

VI. PASSWORD USE IN PRACTICE

This chapter will expose a reader to the practice of password usage in commonly used computing environments. The system access control for three major operating systems on mainframes/minicomputers will be discussed (i.e., IBM's MVS, DEC's VMS/VAX and Unix).

A. MULTI-USER OPERATING SYSTEMS

Security in multi-user operating systems covers three areas: system access control, file access control, and audit. System access control is concerned with the identification and authentication of users when they first establish contact with the system. This includes both interactive access through terminals along with access through network protocols and batch access through jobs. File and database access controls are concerned with controlling access to both programs and data files by authorized users. Detection of unauthorized access attempts and verification of authorized access is controlled by the audit function. (Courtney, 1991)

1. IBM's VMS

IBM's MVS operating system for the System/370 mainframes was a successor of the System/360 operating system, and was principally designed to support commercial batch

processing in a closed environment. Although the first version in 1976 already contained the time-sharing option (TSO) software, this interactive user interface was still dedicated to preparing batch jobs, submitting them and inspecting the results of their execution. In fact, TSO was subordinate to batch in those days. Owing to the "closed shop" characteristics and the emphasis on batch production, access control was not one of the major topics during the design of those early non-RACF (resource access control facility) systems. (Paans, 1991)

The security mechanisms initially integrated in System/370 and MVS were as follows:

- Both the hardware and operating system allowed a distinction to be made between authorized system software and unauthorized user programs (supervisor vs. problem-program mode, key in storage, virtual vs. real storage, address spaces, etc.).
- When logging on to TSO the user had to provide a user ID and a password. Both were stored in clear text in the system library **SYS1.UADS**.
- The system data set **PASSWORD** could contain read and/or write passwords for data sets.
- For data sets controlled by the virtual storage access method (VSAM) it was possible to define passwords which were stored in the catalogues.
- An expiration date for a data set could be defined. (Paans, 1991)

Although there are security mechanisms allowing at least some control over users, many computing centers do not really use them because of lack of interest in security, or

used them in such a way that they are rendered ineffective. (Paans and Bonnes, 1983)

Initially MVS contained an interface for a security package, which was later filled in by RACF. The first versions of RACF were designed to allow a gradual move from an unprotected environment to a protected environment, and only controlled those subjects (persons, users) and objects (resources, data sets) explicitly defined. For each subject and each object one had to create an RACF profile describing the authorities of the subjects, the access requirements for the objects, the relations, etc. In those days security was sometimes characterized as "nothing is protected unless explicitly specified", especially because many centers did not bring all users under RACF control. They were usually satisfied when the most important users and resources were protected, and allowed the remaining users to work as non-RACF users and to access unprotected data sets. (Paans, 1991)

While the password was first in clear text in the system library **SYS1.UADS**, with RACF it was moved to the RACF database and was scrambled via a masking algorithm. Although this provided more security, the password still remained in clear text in the terminal status block (TSB) in virtual storage legible to many users, and, moreover, the masking algorithm was easy to compromise. Hence the design of TSO was improved to remove the clear text password from the TSB, and RACF was extended with data encryption standard (DES) for

passwords. This was RACF 1.6 in 1984, introduced a one-way encryption via DES and so storing encrypted passwords which could not be decrypted (Mclellan, 1986; Paans, 1991).

Moreover the system administrator was now allowed to specify rules for password syntax and usage. He now may issue the following parameters:

- A list of previous passwords is maintained and the user is not allowed to select one of them when specifying a new password.
- After a specified number of days a password is marked expired and has to be changed by the user during his or her next log-on.
- If a user forgets his password and attempts to guess it, RACF will revoke (freeze) his userid after he exceeds the specified threshold.
- Up to eight syntax rules can be specified for new passwords. For each position, one may indicate whether alphabetic, alphanumeric, numeric, vowel, non-vowel and constant character are allowed. (Paans, 1991)

RACF at the level 1.6 and higher provides sufficient support to force the users to use passwords in a responsible and secure way. Keeping in mind that a trivial password such as a user ID or the user's Christian name can only be selected once, and thereafter has to be followed by at least twenty three other passwords. Moreover, forcing the user to insert at least one numeric in the password inhibits the use of names of persons and brands of cars. And, after all, the hacker has only three to five chances to guess the correct combination of userid and password. Elementary statistics show that the probability of a hit is negligible in such an environment as

long as there are reliable procedures for initial passwords and password resets. With modern RACF and a security minded attitude by those in charge of the system, hackers have no realistic chance to breach the security. (Paans, 1991)

2. DIGITAL EQUIPMENT CORP.'s VAX/VMS

When DEC released its initial version of VMS in 1978, it protected the password file with an encryption algorithm called AUTODIN 2 CRC. That algorithm performed a hash on the password and then stored the 32-bit hash rather than the password itself. (Mclellan, 1986)

It became, however, a tempting target for cryptanalysts and by 1980, two different methods had been discovered to invert the algorithm and decode the password.

DEC realized that it should not have encrypted the password alone, and that 32-bit hash was too short to avoid "aliases" (identical encrypted passwords for different users). It also realized that the AUTODIN 2 CRC encryption algorithm, which executed in 140 microseconds, was too fast, allowing brute-force decryption schemes to work.

After three years of using AUTODIN 2, DEC shifted (with VMS version 2) to the so-called Purdy algorithm for encrypting its password authorization file in its VMS version 2. DEC also changed the encryption procedure to use a 64-bit hash of the password, plus user's name, plus 16-bit random

"salt" that was also stored in the user's User Authorization File (UAF) record. (Mclellan, 1986)

The mix of these three elements, plus the addition of a random salt, meant that any brute-force attack had to target each user's password individually--rather than try a particular password guess across the entire authorization file. In addition to making the encrypted password user-specific, DEC had--with the Purdy algorithm--shifted to a crypto system that was almost 100 times slower than the AUTODIN 2 CRC. (Mclellan, 1986)

The Purdy algorithm in a VAX has no "key." It is simply an inherently irreversible mathematical scheme based on the difficulty of factoring large numbers--the same class of problems at the heart of the widely publicized RSA "public key" crypto system.

DEC exhibited considerable independence in choosing the Purdy algorithm over government-approved forms of cryptography--specifically the Data Encryption Standard (DES) algorithm promoted since 1975 by the National Security Agency (NSA) and the National Bureau of Standards (NBS).

By avoiding DES, DEC successfully stepped out of the class of code users who were overly dependent on government approval. By doing so, DEC escaped the impact of the NSA's announcement in 1985 that DES was now so old and too widely used to be trusted any longer. Also, in relying on a cipher that used prime-number factoring as its coding principle, DEC

chose a scheme that is actually mathematically similar to the crypto devices the NSA is now promoting for new, "stronger-than-DES" crypto security. (Mclellan, 1986)

DEC's VMS has achieved a National Security Agency (NSA) rating of C2 which provides the capability of defining who can and cannot use the system, what they can access, and why. It also provides a strong audit capability to ensure control is maintained while retaining the flexibility needed in a general purpose operating system. (Candia, 1990)

DEC's latest release, VMS version 5.4 includes additional password controls. For security managers worried about sophisticated users bypassing minimum password length requirements, the algorithm has been changed. For sites with local requirements for password hash algorithms, there is now a means of specifying one's own algorithm. (Kilgallen, 1991)

For most sites, however, the password history and password screening features are the most significant improvements. By default, VMS will retain a history of hashed values of users passwords, and prohibit the user from choosing a new password which has been used in the past. There is limited amount of space (adjustable by the system manager) for storage of old passwords. By default, this will hold several years worth of passwords, even if they are changed as often as once per month. For the malevolent user who decides to change his or her password many times to overflow the space, there is

no safety. In that event, VMS forces that particular user to use a machine-generated password. (Kilgallen, 1991)

Password screening prevents users from choosing new passwords which are found in an on-line dictionary of North American English words. In addition to dictionary screening, VMS V5.4 also supports site-specific exits during password selection, so that local tests can be made as to the suitability of passwords. This could be used to enforce a requirement that all passwords include both letters and numbers, or that no password start with the letter A, or any other restriction.

It is imperative that security managers ensure that the new security features on VMS V5.4 are actually being used. At many sites system managers have decided to exempt some or all users from the new password requirements. Viewed from the security perspective, that is ridiculous; but it still happens.

Of course, system managers have the ability to override these password restrictions and assign trivial passwords to themselves or to certain favored individuals. The fact that password restrictions are normally in place does not eliminate the need to periodically run password-guessing tests against each machine. (Kilgallen, 1991)

3. UNIX

Unix systems make use of a modified version of the DES algorithm. As with IBM systems, Unix systems put a password into the DES key port in order to make it a one-way encryption system. But as with DEC with the Purdy algorithm, Unix avoids both government crypto controls and the inherent risk of being part of a large group that is using a standard crypto algorithm--inevitably a choice target for hostile cryptanalysts. (Mclellan, 1986)

The features that make Unix machines easy to use can also make them vulnerable to attack. Hence, Unix has acquired a reputation for weak security (Lonsford, 1990).

The parameters and practices for user IDs and passwords set up by the IS staff are the most important criteria for securely running any computer system. These controls are even more critical under Unix.

In Unix, a user creates an ID, also referred to as an open ID or open account, that requires no log-in password. Users of open-IDs must be assigned to single-user workstations that have no outside connections. Thus, an open-ID would be unacceptable for a multi-user system or a network. (Lonsford, 1990)

Two features that can create problems similar to the open user ID are the "trusted host" and "trusted user." From the trusted host, all remote log-ins are accepted without a password. A trusted user is a single user who is allowed to

log-in from his or her local system to another without supplying a password. A list of the system's trusted hosts is located in the file **/etc/hosts.equiv**.

In the home directory of every trusted user is a file called **.rhosts**, which lists the systems from which the user can log in without a password. Often the user configures each account so that he or she may log in from the other hosts without a password. Attackers have exploited this by gaining access to the user's account on one system, then attempting to log-in to the systems named in the **.rhosts** file. (Lonsford, 1990)

Most operating systems, such as IBM's MVS or DEC's VAX/VMS with resource access control facility (RACF), have split up the various access privileges and allotted them to categories. For example, in order to make backups, the operator must have the ability to read any dataset on the system. That privilege might be called READALL. The security administrator, who sets up the system rules and file protections, would have a privilege on his personal account called **security**. Unix, however, has no such distinctions; it's all or nothing when it comes to system privileges.

Unix has only one privileged ID or account, which has all privileges. It is common practice at Unix sites to share this superuser ID, called **root**, among the system

administrators simply by sharing the password. Of course, sharing an ID and password creates an accountability problem. (Lonsford, 1990)

As a remedy, many Unix systems require the privileged user to log into a personal account first, the use the **setuser** command to log in again as **root**. Thus, the privileged user must know two passwords to become **root**. A byproduct of the **setuser** command is an audit record that tracks who logged in as **root**.

Obviously, these procedures are only good if the users who are allowed to sign on as **root** are trustworthy. A user who can bypass or override such security controls has the keys to the entire system. Any user who is signed on as **root**, the highest level of Unix privilege, can delete the log records on all but the most secure systems, This is true, not only of Unix, but VMS, MVS and other operating systems.

Once the IS staff have secured user-IDs, they should turn to passwords. Most operating systems provide a way to set requirements for passwords. (Lonsford, 1990)

Unix, however, has no built-in password-screening facility. Nor does it have a place to add one easily--leaving users to there own choices. Without guidance from IS most users will pick an easy password and stick with it. Fortunately, some new versions of Unix and commercial add-on packages, provide password generators that can improve basic Unix ID and password administration.

Another vulnerability in Unix's password administration is that the file containing user-IDs and passwords, called /etc/passwd, is publicly readable. Although the passwords in the file are encrypted, the encryption routine is readily accessible. Encrypting a guess at a password and comparing it with those in the password file is a simple matter. Newer versions of Unix, such as Sun Microsystems Inc.'s SunOS version 4.0, and AT&T's System V release 3.2 and System V/MLS, have addressed the problem by moving the passwords from /etc/passwd into a shadow file that is readable only by root. (Lonsford, 1990)

Deciding which files on the system are critical is key to determining how to structure Unix's file transfer mechanisms and remote access features. These files should, of course, include the operating system and configuration files, password file and any shared program files, including both source and executable program files.

Unix provides several features to control and monitor remote access. Unix can, for instance, limit remote commands to prevent remote system users from controlling the central system. The exact controlling mechanism depends on the flavor of Unix in use. Some Unix systems restrict the use of commands by specific remote nodes; some have restrictions that include all remote nodes. Regardless of the mechanism, the IS staff should decide which commands and directories should be accessible to remote users or disallow all remote commands.

The administrator should also be sure he/she has enabled the Unix feature that automatically generates audit trails of remotely initiated activities, and it should be reviewed regularly. (Lonsford, 1990)

VII. ALTERNATIVE AUTHENTICATION MECHANISMS

This chapter discusses alternative security techniques to traditional and advanced password schemes. Such alternatives include hardware features, tokens, smart cards, and biometric devices (See Fig. 3). Several of these methods are used in conjunction with traditional passwords or PINs.

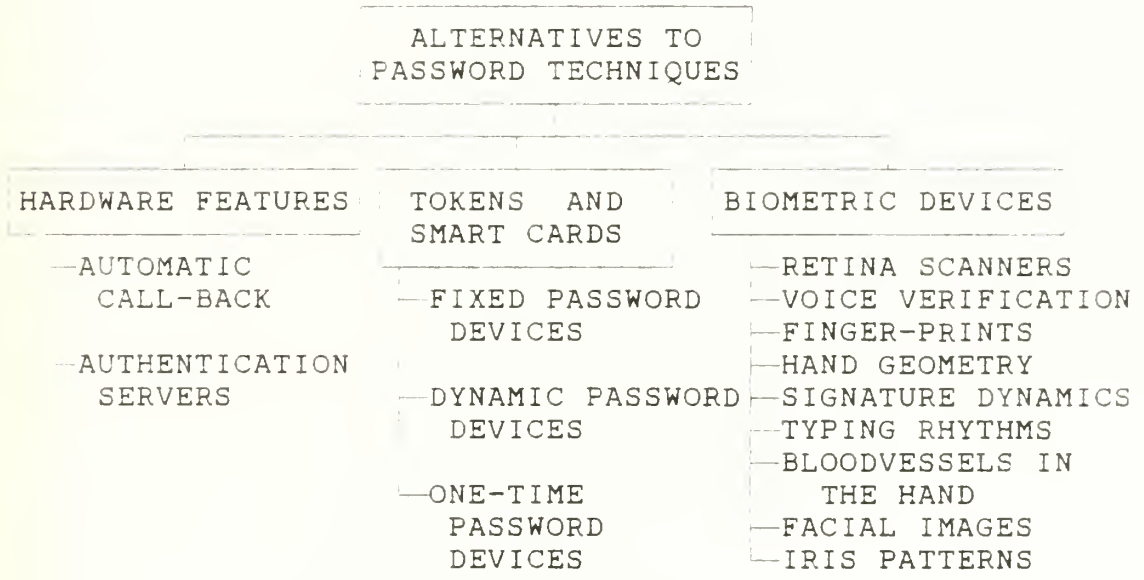

Figure 3. Alternative Authentication Mechanisms

A. HARDWARE FEATURES

1. Automatic Call-back

One early solution to the end-user authentication problem in a network environment was the call-back device (Murray, 1983; Wilson, 1987).

With an automatic call-back system, an authorized user dials a computer system. After a user identifies himself to the system, the computer breaks the communication line by hanging up on the user. It then compares the user and telephone number to an internal list and calls the user back at a predetermined number.

All dial-back accomplishes is to change the telephone number the hacker must attack. The effectiveness of the call-back system is based on the assumption that it is calling back an end-point of the network. The problem arises when a PC user has left his PC hooked up to his desk telephone in order to conduct business from another remote location. To break in, all a hacker may need to do is call up that user's office telephone and do a logical execution of the PC function keys until finding the one which automatically dials up a departmental minicomputer. If a user has programmed all his/her host sign-on codes into the PC, the hacker performs another logical execution of the function keys, causing the minicomputer to automatically dial into the host. And the host, after recognizing the call on one of its call-back

lines, hangs up and calls the remotely controlled minicomputer back (Murray, 1983; Wilson 1987).

In a closed network system where the end-points and telecommunications paths of the network are known, an automatic call-back system can be a useful security device. But in today's open network system the automatic call-back system as a means of security is limited. (Wilson, 1987)

2. Authentication Servers

Working in a network environment poses additional threats to security since penetrating a machine may enable a penetrator to compromise other network-connected computers as well. Authentication servers are one way of authorizing users to all machines on a network. An example of such a mechanism is Kerberos, an authentication mechanism for untrusted workstations developed at the Massachusetts Institute of Technology for their Project Athena network of workstations. (Jobusch and Oldehoeft, 1989)

The Kerberos system is a "trusted third-party authenticator" meaning the network clients using the system trust the server's "judgement as to the identity of each of its other network clients to be accurate." The authenticator server maintains a database of its network clients and their private keys. Using these private keys along with session keys generated by the server, tickets identifying network

clients are created and used as evidence of authentication. Kerberos can be used to authenticate all network services.

When a user identifies himself as a Kerberos client by entering a user name, the user name is sent to the authentication server, along with a request for ticket-granting service. The authentication server first checks to see if it knows about the network client. If so, the authentication server generates a random session key to be used during communications between the network client and the ticket-granting server. The authentication server then forwards information on the network client to the ticket-granting server, encrypted with a key known only to the authentication and ticket-granting servers. A copy of this ticket is then sent to the network client, encrypted in the network client's private key which was derived from the user's password, known only by the network client and the authentication server. The network client then asks the user for the password. The entered password is then used to decrypt the response from the authentication server. The ticket contained in the response is then the information that the network client needs in order to request network services.

While the Kerberos method is complicated, it achieves the primary goal of authenticating untrusted workstations and their users. (Jobusch and Oldehoeft, 1989)

More sophisticated measures than passwords and hardware features are needed to control the security problems

in this new, open network environment. The solution for the commercial environment is a sign-on security mechanism, independent of network configuration, that unequivocally identifies the specific authorized user seeking system access. This enhanced security can be accomplished using dynamic password devices and one-time password devices. These devices are discussed in the following section.

B. TOKENS AND SMART CARDS

The major challenges to increasing security in information systems have been cost and convenience (Weiss, 1990). What is needed is to provide a cost-effective increase in the level of security without burdening the user or the security administrator; system users need to maintain the convenience, portability, and flexibility of a simple password; and must exponentially increase system security at the same time. An effective way to accomplish these goals is to supplement the traditional password with an authentication mechanism (e.g. tokens) (Weiss, 1990).

An example of a token as an authentication mechanism is a bank ATM card. It requires a user to insert the card into a "card reader" at the ATM terminal--which reads data stored on the card's "magnetic tape" and then demands a second identifier: the user's memorized PIN to verify access. The ATM card along with the PIN ensures that the user is authenticated properly. (Weiss, 1990)

Any strategy to develop a creative solution should emphasize three overriding criteria: greatly increased security, end user convenience, and flexibility. This criteria led to the following goals:

- Any new scheme must be at least several orders of magnitude more secure than existing technologies.
- The convenience, portability, and ease of use associated with passwords should be maintained.
- No additional equipment should be required at the physical terminal.
- If a token were used, it should be as convenient to carry as credit card, (hopefully very similar in size).
- A token should strongly resist counterfeiting.
- The computer access token should support added value by: being adaptable as an ID badge; provide over-the-phone authentication; allow for additional uses such as physical access control and encryption key generation. (Weiss, 1990)

One solution is to design a credit card-size device which can electronically "display" a code unique to an individual. If such a card could change its display every 30 or 60 seconds--and a synchronized host computer was programmed to accept that card's displayed code only while it was being displayed, and then but once--then the risk of electronic eavesdropping or casual observation evaporates. (Weiss, 1990)

About 20 vendors currently market such hand held devices, each of which contains a microprocessor, battery and LCD readout and ranges in price from $30 to $100 per unit. Four vendors' product, however, lead the market: Enigma Logic's

Multisync and Access Card; Racal-Guardata's Watchword Generator; Digital Pathways' Securnet Key; and Security Dynamics' SecurID. Token software, which can reside on a mainframe, minicomputer or personal computer, is customized for each installation and thus ranges in cost. (Highland, 1990)

Enigma Logic's Multisync and Access Card, Racal-Guardata's Watchword Generator, and Digital Pathways' Securnet Key produce devices that are about the size of a small calculator with a numeric keypad and use a "challenge-response" strategy. The user logs on to his/her terminal using a PIN and the computer response with a "challenge"--a single digit or series of digits on the terminal screen--which the user keys into the token. The handheld device then performs a computation on the challenge based on an algorithm assigned specifically to that token. When the token displays the results, or "response," the user enters it into the terminal's keyboard. Meanwhile, the host has performed the same computation. If both responses match, the user's identity is verified. (Highland, 1990)

Security Dynamics' device is the size of a credit card and operates on a random-number basis. When the system is set up, a starting number, or "seed," is assigned to the token and recorded on the host. To access the host, the user first enters his/her PIN and then the random number generated by the device, which changes every 60 seconds. The host verifies the

authenticity of the PIN and then refers to its reference table to find the seed as well as the date and time that the seed was put into the token. Using an algorithm, the computer determines what number the token should have displayed and compares it with the number entered. (Highland, 1990)

Even if a device is lost or stolen, other built-in features inhibit illegal access. The software for each token allows only a certain number of log-on attempts before locking out a user.

Some token software also includes an audit trail and built-in alarm that alerts the security administrator or host operator of illegal access attempts. Some software can be customized to provide data on files accessed as well as exception reports.

While tokens have been available for more than a decade, early releases were somewhat unreliable. Battery failures and other malfunctions wreaked havoc on systems. While recent improvements have made these devices more acceptable for general use, these devices pose some drawbacks.

One problem is the tiny keyboards on the challenge-response devices. For any one with medium-size fingers, it is very difficult to push in a number on the half-centimeter-square numbers without hitting the key next to it, about 4mm away. Many people resort to using an implement such as a pencil eraser.

The problem is not just reduced accuracy. On the challenge-response type of token, a user has only a limited amount of time to key in the challenge to the device and response to the host. If the user exceeds the time limit, he/she is automatically logged off. If the user misses three times, the system locks him/her out. (Highland, 1990)

While some token software can be adjusted to lengthen the response limit, too lengthy a duration will compromise security. Limiting time is another good method of screening an intruder who is inexperienced with the token.

Battery life is another concern. A typical battery will last five years, but the security administrator should always keep a log to anticipate replacements. On units with embedded batteries, the entire token must be replaces.

The more severe problem posed by faulty or worn-out batteries is that the user cannot access the system. In other cases, a user might forget or misplace the token, or it might be stolen. In the last case, the security administrator needs to deactivate the user's account.

Because it is inevitable that an employee will at some point leave a token in another pocket or purse, the security administrator must keep spare tokens available.

As with passwords, the use of tokens can overlook the fact that most computer-related crimes or errors are committed by authorized users. Software may someday be available to support tokens, adding a third layer to the access control

system. They could protect highly classified files by challenging any user attempting access. If tokens are assigned only to users who should see these files, tokens could be used to screen unauthorized access. (Highland, 1990).

1. Fixed Password Devices

A "dumb" token, such as a credit card carries a fixed password. This is communicated whenever the token is read. If the host system reads the token directly (electrically), the data can be hidden from the user. The user can not memorize it create copies or communicate it to others. Thus one principal consequence of using an authentication device is that users need not be trusted with the authenticating data. The key, of which there is one copy, remains in the user's hand, where its presence can be observed, rather than in the head, where it cannot be checked. (Spender, 1987)

This is only true if the device is secure from interrogation by anything other than its host system. There must be no other way of reading the data from the token or otherwise copying it. The host can add further security by updating the data every time the device is used and creating cross-checked audit trials in both device and host. In general, devices can be made more secure if they have additional "smarts" such as read/write memory or a microprocessor. Then the token can demand identifying data from a user and/or the host before operating correctly.

A direct reading token needs a reader, such as a magnetic stripe reader, at the user's terminal. An alternative is to equip the device with its own reader, i.e. engineer it as a calculator-like password generating device. When questioned by the host, a user inputs that question into the token, which generates an answer which is then passed back to the host. Though if the question is always the same, a user will be able to record the answering password and use it without having the token. (Spender, 1987)

2. Dynamic Password Devices

In the past several years, a major development in the computer security field has been dynamic password security systems. Computer users seeking to verify their authorized identities through dynamic password systems do not know the value of the password by which they gain entry. They may have memorized a portion of a required password but the remainder can only be obtained from a hardware password issuing device, which displays a different password each time it is used (Bosen, 1986; Avarne, 1988).

Some of these devices derive their dynamic passwords by encrypting combinations of the current date and/or time (or elapsed time between two successive usages). Others encrypt their own prior usage history. Most are capable of encrypting random-number challenges issued by security logic within the protected computer resources. Some encrypt random flashes of

light emanating from the surface of a CRT driven by appropriate security software. (Bosen, 1986)

If the token has an on-board microprocessor it will be able to process data. The microprocessor can be used to hide the authenticating data. One widely adopted method is to use the authenticating data as an encryption formula or cryptokey. A user can be presented with an unexpected text, such as a random number. This is entered on the token's calculator-like keyboard and the encrypted text read off its display. The encrypted reply to the host's random challenge is passed back to the host which then determines the cryptokey used. In this way the hosts knows the token's identity and, by implication, which user is accessing the system. The host's task is simplified if it knows all registered cryptokeys and simply establishes which of these, if any, has been used.

Provided the token's encryption technology is sound, this lock and key interaction keeps the authenticating data secret. It is only revealed to the host in the complex cryptographic relationship between the challenge and the response. Randomizing the challenge prevents a user, or anyone else, from responding correctly without actually using the correct token. (Bosen, 1986)

There are alternatives to the challenge/response approach. Any piece of changing data shared by both the token and the host, such as the clock time of the user/host interaction, can be encrypted. This type of token needs a

clock synchronized with the host's. The advantage of synchronized token is that the challenge data does not need to be entered, it is already known to the token. Since the user does not need to enter the challenge, this approach may seem more user-friendly.

The two mode, full challenge/response and synchronized, create different implementation problems, system risks and user benefits. Some of the commercially available tokens offer both modes, others only one. (Spender, 1987)

Dynamic passwords are entirely unpredictable and so cannot be guessed. They are only used once, so are of no use to an attacker, if they are intercepted. (Avarne, 1988)

Several disadvantages are associated with dynamic passwords. First, if the token is lost or stolen it could be used by an intruder to access the system. Secondly, the expense of outfitting each employee with a token may be cost prohibitive (Bosen, 1986; Avarne, 1988).

3. **One-Time Passwords**

Another variation of dynamic passwords are one-time passwords. One-time passwords use a credit card-sized device which can electronically "display" a code unique to an individual. The card is designed to change its display every 30 or 60 seconds--and a synchronized host computer is programmed to accept that card's displayed code only while it

is being displayed, and then only once--this eliminates the risk of electronic eavesdropping or casual observation. (Weiss, 1990)

The use of this card requires what appears to be two passwords: one classic and conventional, the PIN; the other is the displayed card-code.

They are, of course, different. They share only convenience, portability, and ease of use. The second ID validator, the changing and unpredictable 4 to 8-character card-code displayed on the card's LCD screen, becomes concrete evidence (without a card reader) that the card (token) is at that point and time available to the user. It is a coded representation of the possession of an uncounterfeitable token. A user's eyes are the card reader, the existing terminal keyboard is the ID entry device, the code entered is a password that is not a tradition password. (Weiss, 1990)

One of the strengths of this password scheme is that the codes generated and displayed by the card can't be known ahead of time, memorized, loaned, or even guessed by anyone. There is no pattern; prior codes become irrelevant.

A user simply reads the displayed alphanumeric characters off of the card, types them in--and now, any PC or dumb terminal captures two ID authenticators (Weiss, 1990).

The one-time password produced by the tiny computer within the ID card--can replace the memorized password in

identifying the user to an Information System. It can also be added as a second authenticator to supplement the first. For example, the system at the host demands, first, an assigned, secret and memorized password, and only second, the card-code which, at that moment, is being displayed on the ID card's LCD screen. Together they form a pass-code. (Weiss, 1990)

The use of this second independent token-based identifier vastly increases the certainty of end-user authentication. Security managers no longer need to worry so about one co-worker learning another's password, the headache of password administration is greatly reduced. Outsider and hacker-related threats virtually disappear; and internal threats--always more prevalent and serious are controlled because audit trails offer solid accountability.

The pass-code requires two independent elements--something known--the memorized password; and something possessed--the ID card and its displayed card code. A lost card becomes useless without its complementing password. Similarly, the memorized password is worthless without the uncounterfeitable card token to generate the one-time card code. (Weiss, 1990)

The proliferation of networks has created great and valid concern about the security of passwords transmitted in clear-text. A new technology is available which protests the secrecy and integrity of your passwords without the expense and complications of full network encryption.

The great strength of the one-time password generated by a hand held ID card lies in its computational unpredictability. The displayed card code is, in part, the result of a mathematical process known as a one-way function (with data loss). There is no known way to reverse the calculation, or predict or compute a fraudulent card code, even with the card in hand for study. Absolutely nothing transmitted over the network will ever allow an intruder to later-or sequentially-gain illicit access to a protected system.

To give full protection to the memorized password, the first of the two independent ID authenticators suggested, Security Dynamics has developed a SecurID "P Card" (pin pad). Still the size of a credit card, the P card has pressure-sensitive keys built into it. A user enters his memorized PIN into the card, and the displayed sum of the two separate authentication codes can then be transmitted over an open line with full assurance that an eavesdropper gains absolutely nothing if the resulting PASSCODE is intercepted. (Weiss, 1990)

A constant added to a random number produces only a random sum. No clear-text password is ever transmitted; nor is the user's memorized password ever stored within the card. The card does not compare or validate the entered PIN; it simply adds it to the next random number generated by the card. (Weiss, 1990)

4. Relating Tokens and Users

The ergonomic issues are important because identity is associated with the token, not the user to whom it has been issued. Most corporate employees are getting used to carrying credit-card style photo-ID badges. These may double as physical access control devices and control the users' movement about a secure plant, operating with direct readers; magnetic stripe, bar code, tuned eddy-current proximity devices and smart cards. Such established behavior patterns make a credit-card type token especially attractive.

A token can be lent, stolen or otherwise fall into another's hand. If it alone establishes identity, that identity is readily transferred, as a car key transfers the driver's identity as far as the car is concerned. One widely adopted method of tying the token logically to its legitimate user is to have it smart enough to require a user to enter a memorized "wake up" PIN. PINs are familiar to ATM users. Very sophisticated authentication devices may also have clocks and multiple self-aging PINs, which change regularly. Tokens may contain multiple "virtual" identities. (Spender, 1987)

Disadvantages of using token systems include the financial burden of providing each employee with a hand-held token, not to mention that workers likely would leave the devices in their desk drawers, once again breaching security. If a token is stolen, broken or disabled it may require more

time then is acceptable before a user can be granted access to the system (Spender, 1987; Highland, 1990).

Many of today's tokens are vast improvements over earlier models. To consider their use for all everyday business operations is as foolhardy as using the same encryption algorithm for all data. Token are a selective tool. Each organization has special files and/or systems that require additional protection; tokens are an effective way to solve this (Highland, 1990).

C. BIOMETRICS/PERSONAL CHARACTERISTICS

Experimental personal recognition systems have been built around lip prints, blood-vessel patterns in the retina of the eye, voice recognition, signature verification, and electroencephalogram traces (Carroll, 1987; Wilson, 1987).

A person's biometric data tends to be a wholly fixed password or a way of giving a user a lifetime password. Problems arise if this device is compromised. For example, if a user is using a signature verification device and a user's signature is forged, there is no way for a user to regain access to his or her signature. It is lost as an authentication mechanism.

Biometric characteristics are complex, implying large data transfers between user and host. Protecting these data between reading device and host is correspondingly more difficult. The comparisons are automated but statistical,

opening the system to problems with Type I errors (admitting the wrong user) and Type II errors (excluding the right user). The complexity and variability of biometric data also creates a new type of problem, a user who cannot produce a satisfactory template for the system to compare against.

Getting computers to recognize people can logically be approached two ways; make computers more like people, equipping them with biometric readers or make people more like computers, equipping them with personal computerized authentication devices. The latter strategy seems less expensive, more secure and more readily implemented at the present time. (Spender, 1987)

Because of this vendors of biometric systems have focused very heavily on errors in reading and recognizing, on the cost/performance ratio of their readers and on miniaturization (Wilson, 1987).

Human characteristics, although measurable, do change unpredictably. For example, a thumb may be dirty, or have a cut on it; a user can be hoarse from speaking, or suffer from laryngitis. Perhaps his eyes are red and his hands are shaky, so neither can be read. Most biometric devices have a rejection rate in the 4-6 percent range. In a commercial environment, where you're trying to use this device to identify customers, that may be unacceptably high.

Overall, this kind of security, provides a reasonable basis for end-user authentication and the foundation for the

future implementation of biometrics in open networks. But the cost for biometric devices can be very high, which is clearly unacceptable for most applications. (Wilson, 1987)

Biometric devices which have been successfully applied in commercially available products include:

1. Retina Scanner

The retina scanner bounces an infrared beam off the retinas of a subject's eye and traces the pattern of distinct blood vessels. No two individuals have the same pattern, so this provides identification as precise as a fingerprint (Kanner, 1990; Parks, 1991).

The capillaries within the eye reflect less infrared light than the surrounding tissue. What the scanner measures is the intensity of the reflection at 320 points along the beam path. A number between 0 and 4,095 is assigned to each point's intensity.

These numbers are then translated into an 80-byte computer code to form an "eye-signature." The small amount of data that his code uses gives the retina scanner an advantage over other forms of biometric devices, such as those used for voice prints or finger prints.

By matching the retina pattern code to those stored in a database, the system can positively identify a person in less than three seconds. Another plus is that, once users enroll their eye signatures, subsequent updates are

unnecessary. Retina patterns don't change as people's voices and signatures do.

The U.S. Defense and Energy departments have been the largest users of retina scanners for the last seven years. Primarily, they're used as stationary physical-access control systems to weapons facilities and computer rooms. Corporate America is slowly finding use for them among financial data centers.

Retina scanners present a tremendous opportunity for database security applications. This technology offers the highest level of security. If it is compared with card-key systems, there isn't anything that can be lost or stolen. Users carry their ID with them. Also, if an employee leaves the company, the locks do not have to be rekeyed.

A key criterion that any security device must live up to is an extremely low false-acceptance rate. A user does not have to identify himself/herself to the system beforehand. A retina scanner can determine by itself if a user is enrolled in the system. Every other machine needs a PIN, or code, to know what template by which to compare a user.

The technology could be valuable for tracking insider computer threats from criminal perpetrators. Retina scanners can tell which person has accessed a file at a certain time; if damage is done, the perpetrator cannot deny it a week later.

Retina scanners provide an audit trail and accountability that lets the system point out an individual who may have done something harmful to the system (Kanner, 1990). Retinal patterns have proven to be very effective in detected attempted impersonation (Holmes et al., 1990). It's like leaving fingerprints at the scene of the crime (Kanner, 1990). In terms of physical-access control, retina scanners are outstanding (Kanner, 1990).

2. Voice Verification Device

Advances in speech processing technology now offer an attractive and unobtrusive supplement to current security methods. The wide distribution of microphones in installed telephones has stimulated the development of user verification devices exploiting the variation in voice quality from person to person (Parks, 1990; Penzias, 1990).

With a voice lock, a speaker's own vocal chords act as the key. By speaking, instead of merely typing, authorized users allow the voice lock to confirm their identities by matching the attributes of their voices against the speech samples stored under their names. (Penzias, 1990)

Imagine a hacker, trying to gain unauthorized access to a computer system via a dial-up telephone line. Until now, it has been relatively easy to get a list of phone numbers, log-ins and commonly used passwords from underground publications, electronic bulletin boards and similar sources.

Bad security habits make the gathering of such information a commonplace fact of life. No matter how hard system administrators try to increase security, some users will thwart this effort by using easy-to-remember passwords--the electronic equivalent of setting a safe's combination to "0,0,0."

A hacker begins an assault by dialing the first number; then a personal computer's programmed attack plan can take over and automatically redial the phone number over and over as it tries each possible character combination.

But instead of the familiar modem tone, a hacker hears a voice message: "You have reached port number 6. Please identify yourself by speaking your name."

Such a response complicates an assault because a hacker must read aloud from the list on each try or record all candidate names in advance. Furthermore, even if a hacker happens to hit upon a valid name (and one whose owner happens to have a similar accent, age, and gender), the odds against getting through are about 100-to-1. (Penzias, 1990)

Even if a hacker has somehow obtained a tape recording of a user's spoken name by eavesdropping on an earlier session, an assault would then move to the next barrier.

"Please verify your identity by speaking the words, 'Good morning America how are you.'" (A randomly selected sentence is stored in the system's memory.)

Suppose the voice lock contains its own tape recorder. The hacker runs the risk of hearing those words played to a jury someday.

These formidable obstacles do not place any additional demands upon a legitimate users: he/she must remember his or her name and use his or her own vocal tracts. Voice locks offer similar security enhancements at the desktop end as well.

On the hardware side, a voice lock calls for the same digital signal processor chip platform employed by other speech processing applications. Such a platform's open architecture would permit users to tailor applications to suit their individual needs, or to buy them from software vendors.

Imagine a stand-alone PC equipped with a voice board, dialer and telephone connection. Potential users could either dial the PC or access the system directly in whatever manner they normally use. In the dial-up situation, users can get a series of voice prompts (when the modem shares a line with the user's telephone) or begin with a typed request for a log-in.

In the latter case, the logged-in user will be asked to type the number of whatever telephone happens to be closest to the user's terminal. The PC then dials that number, which the user must answer and reply to in a voice-verification sequence. With the user's identity established, the PC hangs up the second line and transfers the original line to what used to be a regular dial-in port before the voice lock was

added--using the normal transfer features of the local telephone switch. In addition, that same telephone switch would presumably limit direct access to the PC's dial-in port.

For systems in which the user's terminal must access the main system directly from the start, the log-in sequence triggers a request to the PC (via a hard-wire connection) for speaker verification. In response, the PC dials the phone number requested from the user during the log-in procedure and proceeds to engage the user in a voice-verification dialogue during the telephone conversation.

Once the speaker's identity has been established, the PC sends to the main system an okay which allows the typed sequence to proceed. Because the PC is engaged only during the verification transaction, a single machine can accommodate multiple users one after another. Furthermore, such a PC's capacity can be further enhanced by the addition of multiple voice boards and dialers, thereby sharing the cost of common equipment among users. Also, hardware costs might be reduced even further by incorporating the above capabilities within an existing system and utilizing whatever components it already contains.

What happens when the voice lock does not recognize a legitimate user? A typical system can be expected to reject a legitimate user about once in every hundred attempts. That makes such false negatives about as common as misdialed seven-

digit telephone numbers. As with telephone dialing (and typed passwords), a simple retry usually solves the problem.

Adding a voice lock must not cause a burdensome increase in the number of randomly generated retries. The need to type unfamiliar words introduces a higher probability of random errors than does a typical voice lock system.

Non-random problems are handled the same as with any other system. For example, if a user has a sore throat, it is no worse than forgetting your password or leaving your token at home. In such cases, a user might keep the instructions for an emergency access procedure locked in a safe place, or get a colleague to vouch for him/her.

Users whose permanent speech impediments preclude the use of spoken passwords could train the system with sequences of tones instead, such as ones produced by a telephone keyboard. (Penzias, 1990)

3. **Finger-Prints**

Fingerprint devices are based on measuring the distance between features in a user fingerprint and storing this information in a template of some 400-1000 bytes. Low cost products using this method are available from several manufacturers, Fingermatrix, Inc., Identix, Inc., and Thumbscan, Inc. Their prices range from $2000 to $4000 per unit. Another method is to use line patterns on the palm of the hand to authenticate a user. (Parks, 1990)

The fingerprint devices are small and compatible with desk and portable terminals and have found their best applications in control of electronic channels (Parks, 1990).

4. Hand Geometry

A system based on the silhouette of the hand was the first biometric device commercially offered. Several manufacturers have resurrected the idea and one system is also using the vertical profile of the hand in addition to the silhouette.

Template sizes vary between 9 and 1000 bytes. Price per unit range between $3000 and $5000. (Parks, 1990)

5. Signature Dynamics

The use of the written signature is so familiar in commercial dealing for authenticating documents and for closing transactions that their use is generally preferred for automation of user verification in banking transactions (Parks, 1990).

Signature dynamic devices all use instrumentation which measures geometric and/or dynamic properties of the action of writing a signature in real time and, therefore, at the point and time of the transaction. Different instrumentation requires the use of a special stylus connected to the unit or allow the user to use any convenient stylus.

Devices are currently available from IBM Corp., Analytical Instruments Ltd., Digital Signatures, Inc., De La

Rue Systems Ltd., British Technology Group, Communication Intelligence Corp., Rolls Royce Business Ventures Ltd., T.I.T.N., Xenetek Corp and others. Templates used in characterizing signatures range from 40 bytes to 4 kilo-bytes according to the method used. Unit prices range from $600-$1200. (Parks, 1990)

6. **Typing Rhythms**

The timing between pairs of keystrokes in a typewritten stream of characters has been found to vary significantly between typists, even those of modest facility. This approach is unique in being potentially both covert and continuous, as it can operate on user keystroking in general use. (Parks, 1990)

This method is software based, possibly with a plug-in card for PCs and the cost ranges from $500 upwards per terminal. Uses are clearly for computing and communications system protection. Commercial sources for typing rhythm systems include Electronic Signature Lock Corp., and International Bioaccess Systems, Inc. (Parks, 1990).

Several other biometric techniques that are not yet available commercially but are in the development stages include bloodvessels in the hand, facial images, and iris patterns. Other aspects of the human being which have been

advanced speculatively as potentially usable for personal identification have included gait, ear shape, heart and brain waves. (Parks, 1990)

VIII. AVAILABLE TOOLS AND PRODUCTS

This chapter will provide the reader a description of how encrypted password files operate. It will also describe five different commercial access control software packages used on IBM operating systems. Other security enhancement software packages discussed include password salting and password monitors.

A. PASSWORD ENCRYPTION

To validate passwords, a system must have a way of comparing entries with actual passwords. Rather than trying to guess a user's password, an attacker may instead target the system password file.

Encryption of password tables is relied on in many instances to preclude unauthorized access to a particular password. The encryption process employed often is not very sophisticated. For instance, it may involve nothing more than modification of each password character by the addition or subtraction of a binary or hexadecimal constant. (Menkus, 1988)

A safe way to avoid the compromise of a password file is to encrypt the file. Systems have two methods of using encryption to protect their password information: two-way

encryption and one-way encryption. With two-way encryption, the entire password table is encrypted, or perhaps just the password column, with a secret key when it is stored. Then when a user enters a password to log-in, the password file information is decrypted with the secret key, and compared with the password that was entered. There is still a slight exposure with this method. For an instant a user's password is available in plain text in main memory. It is available to anyone who could obtain access to all memory.

A safer approach uses one-way encryption--an encryption function for which encryption is relatively easy and decryption is relatively difficult. The password in the password table is stored in encrypted form. When a user enters a password, it too is encrypted, and the encrypted forms are compared. If the two forms are equal, the authentication succeeds.

With one-way encryption the password file can be stored in plain view; in fact, the password table for Unix operating system can be read by any user, unless special access controls have been installed. Backup copies of the password tables are also not a problem. (Jobusch and Oldehoeft, 1989)

One-way encryption process prevents any form of password recovery. However, strictly speaking, there is no such thing as one-way encryption. Use of the term, typically implies that it is impossible to derive material encrypted by such a process through known cryptanalytical processes. That is not

correct. Rather, the encryption process used simply has raised the cryptanalysis work factor to a very high level, making it unrealistic, in most instances, to attempt to derive the password from an attack on the table in which it has been stored in encrypted form. (Menkus, 1988)

Storing a password file in a disguised form relieves much of the pressure to secure it. Access may still be limited to those processes that have a legitimate need for access. However, securing the contents of the table as well as access to the table provides a second layer of security. Someone who successfully penetrates the outer security layer does not get access to useful information. (Kochanski, 1989)

Because of today's open networks, controls that were implemented in the past are no longer adequate. They are still necessary but not sufficient. The old network controls used point-to-point encryption and the Data Encryption Standard (DES). These are being replaced by end-to-end encryption, message authentication, and even new encryption algorithms. (Wilson, 1987)

B. SOFTWARE/COMMERCIAL PRODUCTS

All access control software packages discussed deal with three elements: a user, a resource, and an attempt to access. When the user (which could be a person at a terminal, a program or a batch job) attempts to access a resource (a dataset, a transaction, a CICS region, A VM minidisk, or

almost any other definable entity), it is the function of the access control software to determine whether the access is authorized and should be permitted. (Henderson, 1987)

The five packages described, CA-ACF2, Omniguard, RACF, CA-Top Secret, and VMSECURE all run in any IBM or IBM capable environment.

1. Access Control Software

a. CA-ACF2

In a CA-ACF2 controlled system, passwords are used for system entry validation. The user's password is stored in the CA-ACF2 Logonid database in a one-way encrypted format. When a user logs on and enters his/her password, it is immediately encrypted and compared to the stored password. If they match, access is allowed. An installation can specify that passwords meet certain requirements:

- Number of invalid tries in a session before the session will be canceled.
- Number of invalid tries in a day before Logonid will be suspended.
- Minimum character length of the password.
- Whether the user is allowed to change the password.
- The minimum number of days which must pass before a user can change the password.
- The maximum number of days which can pass before a user much change the password.
- The number of days prior to expiration that CA-ACF2 will warn the user that the password must be changed. (Henderson, 1988)

CA-ACF2 prompts for password entry in display inhibited screen areas. In a CA-ACF2 system, passwords are encrypted with a one-way encryption algorithm, using an extension to DES.

If a TSO user does not specify a specific Log-on ID in the Job Control Language (JCL), CA-ACF2 provides for the automatic inheritance of the Log-on ID of the TSO user. Access decisions in the batch job will be based upon the authority of the submitter. No additional password entry is required, nor must the password be retained in the system or put in the JCL for these submissions.

Users may change passwords without jeopardizing previously submitted jobs. Any job submitted before a password change will be unaffected as CA-ACF2 provides for automatic ID inheritance without password revalidation.

There are techniques available for administrators who prefer not to use passwords on JOB cards. An installation can specify Logonids and passwords in a batch job with special CA-ACF2 control cards in the JCL called //*LOGONID and //*PASSWORD. A password submitted in this fashion will be suppressed at validation, so that it will never appear in a listing. Also, Log-on IDs can be authorized to run without the password requirement. This feature provides for ease of use in a production environment.

When an on-line user first enters a new password it must be entered twice to confirm that the first entry doesn't

have any typos. This ensures the user does not make a typo when changing passwords. This is particularly useful, as all passwords are entered in non-display fields, and the user cannot visually inspect what has been entered. (Henderson, 1988)

b. OMNIGUARD

OMNIGUARD encrypts the users' passwords and then uses a hashing algorithm. Encryption of all passwords is via DES. The users' passwords are not made available to anyone, not even a top level administrator. If the password is hard-coded in the scheme, the OMNIGUARD compiler will return the output with the password commented out.

Password controls for OMNIGUARD include: the password must not match the last four passwords, that they be at least four characters in length, and that the password contains at least three different characters. The length and number of different characters may be changed by the security administrator. It is also possible to force a user to sign on with two additional passwords that may be up to 256 characters each in length. Assignment of passwords can be as follows: user selects his own password, the system administrator assigns the password or OMNIGUARD can randomly assign a

password. Additional editing criteria applied to a users password are:

- Cannot be the same as the user ID.
- Maximum and minimum character length can be required.
- No more than three characters can be the same.
- A time expiration for passwords may be set.
- Number of invalid password attempts before a user session is canceled by either deactivating the terminal or user identification. (Henderson, 1988)

OMNIGUARD provides the default values of three unsuccessful attempts in five minutes. If this occurs, the system will take the terminal out of service, drop the port, or deactivate the user's ID (Henderson, 1988).

c. RACF

RACF uses the DES algorithm for encrypting passwords when stored on its database. RACF also enforces the installation's defined password standards which include:

- Number of consecutive invalid password attempts.
- Password value.
- Frequency with which passwords must be changed.
- Limitations on re-use of old passwords.
- User-definition of new passwords. (Henderson, 1988)

There are special controls over passwords that include length and character composition. Like the Omniguard

software it is possible to force a user to sign on with two additional passwords that may be up to 256 characters in length. Assignment of passwords can be as follows: user selects his own password, the system administrator assigns the password or RACF can randomly assign a password.

Once a user's password is entered into the system, RACF insures that the password will not be compromised. Because the password is one-way encrypted via DES, no means is provided to read a password from a user's profile. With TSO/E Release 3, the password is not kept in memory for TSO. Passwords are not displayed on terminals when entered and are print suppressed to JES output.

When jobs are submitted from TSO a user may supply a user ID/password on a JOB if desired. If no password is supplied, the JOB will automatically run under the user ID of the validated user. Once jobs are submitted, a user can change his/her password without jeopardizing those previously submitted jobs.

Passwords are not required to verify that a batch job properly represents the user it seems to. Validated user ID propagation by JES2/3 supports entry of batch jobs into the system without userid/password parameters. In addition, RACF supports the surrogate user function (via the FACILITY resource class). This allows a designated user to submit a job on behalf of another user.

A common user ID can be used for different systems and different passwords can be used for each subsystem. If the RACF dataset is shared there will be one password. If there are separate systems, each with RACF, the user IDs could be common but with different passwords. (Henderson, 1988)

d. CA-TOP SECRET

CA-TOP SECRET provides extensive password controls to minimize password compromise or guessing. Once entered into the system user passwords are encrypted on the CA-TOP SECRET security file. Special controls for user- selected passwords associated with CA-TOP SECRET include:

- A user may not use any of his/her last three passwords.
- A user may not use a password similar to the last password used.
- A user may not change a password more than once per day (Henderson, 1988).

The installation may optionally specify the following controls:

- The minimum length of a password
- The minimum number of days during which a user will not be allowed to change a password after it has been changed
- That the user may not use a password equal to his/her access control ID name or prefixed with information found in the user's name field
- That only number may be used
- That letters may not be repeated
- That vowels may not be used

- That the user may not change his/her own password
- That random password generation is required
- That a password mask or pattern of consonants, vowels, and numerics must be followed when changing or randomly generating a password
- That a password must contain more than one word making it more difficult to guess
- That the password may not be prefixed with any entries in a CA-supplied restricted password list (this list may be modified to reflect installation standards)
- The interval during which warning messages are issued before a password expires. (Henderson, 1988)

The number of invalid attempts to enter the system is variable from 1 to 255 occurrences. This feature may also be deactivated if desired by the installation. Once the threshold is reached, the user is suspended and can only be reactivated by an authorized administrator.

There are several options available for deriving and validating access control IDs (ACID) without requiring JCL changes or passwords on job cards. If a user submits a job through any facility, including batch, which uses the internal reader, CA-TOP SECRET propagates the ACID of the user who submits the job to the job card without revealing the password in plain text. If the user submits a job to run under another user's ACID, CA-TOP SECRET will verify at submit time that the user has the authority to do so. The installation may also choose to propagate an ACID that is equal to the job name or a portion of the job name. A default ACID may be specified

for each facility. This ACID will be used if an invalid ACID or no ACID appears on the job card. The installation can additionally choose to derive an ACID from information on the job card without requiring the coding of a password. The ACID can be derived from a specific portion of accounting information, programmer name, user keyword, job name, or reader name allowing special ACIDs to be derived for card readers and RJE/NJE readers. (Henderson, 1988)

e. VMSECURE

VMSECURE requires the use of a password for log-on authentication. With full rule-based access, no passwords are required for minidisks. To prevent compromised passwords, passwords can be masked so that users and system administrators cannot see their passwords. If optional password encryption is implemented, the clear text passwords cannot be seen even by the system administrator.

Password controls include reuse limiting (can't reuse any of the last 8 passwords), automatic expiration. password encryption, number of consecutive invalid attempts allowed to enter the system, and user exit so a site can specify additional controls it needs.

When an on-line user first enters a new password it must be entered twice to confirm that the first entry doesn't have any typos. This ensures the user does not make a typo when changing passwords. This is particularly useful, as all

passwords are entered in non-display fields, and the user cannot visually inspect what has been entered. (Henderson, 1988)

2. Password Salting

The Unix operating system incorporates an encryption defense mechanism called "password salting". When a user's new password is first entered, the password program obtains a 12-bit random number and appends it to the password. The linked string is then encrypted and both the 12-bit random number (or salt), and the results of the encryption are stored in the password file. When a user subsequently logs onto the system, the 12-bit number is taken from the password file and appended to the typed password. The encrypted result must match the encrypted string in the password file. This modification significantly complicates the work of testing a given character string, using key search, against a large collection of encrypted passwords. Each password now has 4,096 possible encrypted versions. (Gish, 1985)

While the key search method of attack has been slowed down by the use of DES and the "salt", this technique still works on most Unix systems. Since the password-based ciphertext, "salt", and the encryption algorithm are not secret, key search techniques are limited only by available computer time to do the encryption and dictionaries used to provide the guesses. (Jobusch and Oldehoeft, 1989)

3. Upass

The Navy is now testing a new Unix password management and control product that will give systems administrators better security controls (Schwartz, 1990).

Unitech Software Inc., produces Upass, a Unix security package that allows a systems security officer to administer user account control and maintain a secure Unix environment without being a Unix programmer. The Navy Military Personnel Command (NMPC) is testing Upass. (Schwartz, 1990)

Upass gives a system greater security through password control and automatic reporting procedures that are transparent to the end users.

Using Upass, system administrators can make existing log-in procedures secure enough to meet DOD requirements for systems that handle classified or sensitive materials. Upass stores passwords as one-way encrypted data, making it impossible for anyone to view them. Forgotten passwords cannot be re-created and user numbers cannot be reassigned, other than to the original name.

Administrators can receive notice of possible attempts to penetrate the system in real time. The system can notify administrators of repeated unsuccessful attempts to log in or attempts to log-in from a port not authorized to a given user name.

Upass also provides administrators with a comprehensive security profile for each user, showing user

name, number and all security options in effect. It allows for password changes and log-in history and current password change status.

The package lets security officers administer user account control without having access to root passwords (Schwartz, 1990).

4. Password Monitors

A password monitor is a program that grades a user's choice for a password based on how likely it is that the password could be guesses. Such programs are usually incorporated into the password changing program, so that when users try to select a poor password, the system will reject it.

a. The Password Predictor

"The Password Predictor" is a password monitor program designed to augment the existing password mechanism for 4.3 Berkeley Software Distribution (4.3BSD) version of Unix, by giving the system administrator an automatic mechanism to monitor the use of trivial passwords. When the program is executed, it carries out a selective key search on the password file. The password predictor guesses trivial passwords and then leaves a message in the user's area. This compromise of the user's password should encourage him/her to enter a more complex password since it demonstrates that a trivial password is easy to guess. The message tells the

user, "Your password is NOT secure". The program uses several lists of guesses that include:

- frequently used words from the system dictionary
- common names and nicknames
- a "large sampling of the most frequently spoken and written six to eight character English words"
- strings from the comment field of the system password file
- miscellaneous names, including streets, music groups, and cities
- "personalized guesses"; "trivial passwords a user is known to have used in the past". (Jobusch and Oldehoeft, 1989)

Use of the password predictor will heighten password security awareness and result in passwords being composed from a richer character set. It does not inconvenience the user by forcing him/her to choose an obscure password; it just demonstrates the importance of having one.

The 4.3BSD system has a simple password monitor that can be avoided. No password guessing program is provided with the standard password mechanism software (Carroll et al., 1988; Jobusch and Oldehoeft, 1989).

b. Password Coach

Another example of a commercially available password monitor is Password Coach. It is a completely transparent unless users choose a weak password. Users continue to choose their own passwords, so their passwords will continue to be easy to remember. It screens these

passwords to make sure that they are not in the dictionary; common first names; biographical names; geographical names; technical, medical ,or legal terms; keyboard scales (for example asdfghjkl); account names; or other easily guessed character strings. If a user chosen password is weak, Password Coach provides the user immediate feedback on specific reasons why it is weak and then asks the user to enter another. Users quickly learn how to construct strong passwords because the program provides explicit reasons why passwords are weak. Password Coach comes with a dictionary of over 140,000 American English words. Each word constitutes a weak password. Optional dictionaries include several other languages. The software also allows organizations to define their own forbidden weak passwords. For example, user names, job title, social security numbers, telephone numbers addresses, and other "words" defined as weak. (Wood, 1990)

Password monitoring programs have the same effect on users as password generators. If the monitor programs accept only random characters as passwords, users will not be able or willing to commit the password to memory, and will instead write them down. Allowing these programs to accept rememberable passwords, while discarding obvious ones, is the key to a successful monitoring program. (Jobusch and Oldehoeft, 1989)

IX. PASSWORD USE IN THE MILITARY

The military has always taken for granted the overriding importance of security. They are particularly concerned about preventing leakage of information, and have tended to see computer security largely in terms of the control of access to classified documents (Wilkes, 1990).

In August 1983, the Department of Defense Computer Security Center published CSC-STD-001-83, Department of Defense Trusted Computer System Evaluation Criteria. This publication defines and describes feature and assurance requirements for six hierarchical classes of enhanced security protection for computer systems that are to be used for processing classified or other sensitive information. A major requirement common to all six classes is accountability. (DoD Password Management Guide, 1985)

The trusted computer system evaluation criteria described in the Appendix applies primarily to trusted, commercially available automatic data processing systems. They are also applicable to the evaluation of existing systems and to specification of security requirements for ADP system acquisition. Included are two distinct sets of requirements: 1) specific security feature requirements; and 2) assurance requirements. The specific feature requirements encompass the

capabilities typically found in information processing systems employing general-purpose operating systems that are distinct from the applications programs being supported. However, specific security feature requirements may also apply to specific systems with their own functional requirements applications or special environments (e.g., communications processors, process control computers, and embedded systems in general). The assurance requirements, on the other hand, apply to systems that cover the full range of computing environments from dedicated controllers to full range multilevel secure resource sharing systems. (DoD Trusted Computer system Evaluation Criteria, 1985)

A. MILITARY ENVIRONMENT PASSWORD USAGE

Passwords are used to prevent people who have physical access to an ADP system from gaining access to data belonging to another user. Thus, a password should be protected in a manner that is consistent with the damage that might be caused by its exposure to someone who has the opportunity to use it (i.e., has physical access to the ADP system terminals). Exposure of a password to someone who is physically prevented from attempting to use it is not a threat.

1. Systems Containing Only Unclassified Information

Although an ADP system may process only unclassified information, it still may require that the data be protected from unauthorized use. Although the password is unclassified,

the obligation remains that the user protect this password so that only those with a need-to-know can access the data.

2. Systems Containing Classified Information

Passwords that are used in ADP systems that operate in the dedicated or system high security modes should not be classified, but should be protected to the same degree as For Official Use Only information. In this case, there is no need to classify passwords since access to the area in which the system resides is restricted to those with a clearance as high as the highest classification level of the information processed. A person who obtained a password for a system running in dedicated or system high security mode but who did not possess the proper security clearance would be unable to gain physical access to the system and use the password.

For systems operating in the multilevel security mode, passwords may or may not have to be classified.

When the ability to access classified information is based on the physical protection of the terminal rather than on the identity of the user (i.e., when all terminals are single-level devices), passwords should not be classified, but should be protected to the same degree as For Official Use Only information. There is no need to classify passwords that can only be used on single-level terminals, since physical access to single-level terminals is controlled to the level associated with the terminal. When the ability to access

classified information is based on the user's identity and is not restricted by the level of the terminal (i.e., multilevel terminals), each password must be classified to the highest level of information to which it provides access.

When multilevel terminals are used, the system determines the user's access authorizations to classified material based on his identity, and authenticates the identity by requiring a password. Thus, the ADP system can protect the information it processed only to the extent that passwords are protected. For example, a user with Secret clearance can access Secret information. Compromise of that user's password could result in the compromise of Secret information; therefore, the password would be classified Secret. In the case of a system with multilevel terminals, disclosure of a Top Secret user's password to a Secret user would allow the Secret user to login as the Top Secret user and thus gain access to Top Secret information. Disclosure of Top Secret information to someone with only a Secret clearance can cause exceptionally grave damage to the national security. Since disclosure of the Top Secret user's password could lead to this, the password must be classified Top Secret.

Note that classified passwords must not be used on terminals that are not authorized for data at the level of the password (e.g., a Top Secret password must not be used on a Secret terminal). The presence of both single-level and multilevel terminals on a system may indicate the need for

passwords at each security level. At a minimum, an unclassified password should be available for use on terminals that are only authorized for unclassified data. (DoD Password Management Guideline, 1985)

3. Major Features of DoD Guidelines

Specific areas addressed in the DoD Password Management Guideline include the responsibility of the system security officer and of users, the functionality of the authentication mechanism, and password generation. The major features recommended in this guideline are:

- Users should be able to change their own passwords.
- Passwords should be machine-generated rather than user-generated.
- Certain audit reports (e.g., date and time of last log-in) should be provided by the system directly to the user. (DoD Password Management Guide, 1985)

B. SIMILARITIES WITH PRIVATE SECTOR USE

Authentication mechanisms are used for the same reasons in the military environment as they are in the private sector. They are used to protect some type of privileged information or data from unauthorized users.

C. DIFFERENCES WITH PRIVATE SECTOR USE

While similarities of password use between the military and private sector parallel each other the differences are many.

The military has certain requirements that must be met. In the private sector, if an organization chooses not to use an authentication mechanism to protect its information then it's a risk that they choose to make. A military organization does not have that option, if the system meets the criteria for an authentication mechanism then one must be used in accordance with current government directives.

X. CONCLUSIONS AND RECOMMENDATIONS

User authentication is an integral part of any IS security mechanism.. While absolute security seems to be unattainable (Kochanski, 1989), high degrees of security are commercially available. But they can be inappropriate. When evaluating various user authentication approaches, a user should consider how much security the system really needs. In many cases a traditional password scheme is sufficient. If that does not provide adequate security, then a combination of a password and one of the alternative authentication mechanisms might better suit the organization.

A. TRADITIONAL PASSWORD MECHANISMS

While traditional passwords are the most frequently used authentication mechanisms (Menkus, 1988), there are many problems associated with there use: hard to remember, easy to guess, user resistance, written down, low level of security.

Despite the fact that alternatives to traditional password mechanisms exist, it seems that most organizations will stick to traditional passwords. This occurs because traditional password mechanisms are an integral part of most operating systems, are readily available, and are inexpensive to install. Thus, careful attention should be given to their

selection (use a broad character set, force change after a period of time) and proper use.

B. ADVANCED PASSWORD MECHANISMS

If an organization desires to improve its present user authentication method while not advancing beyond passwords, then an advanced password scheme should be considered.

While passphrases and question-and-answer mechanisms seem to provide for both ease of memorability and difficulty of guessing there are still problems with them. Each requires some type of query system be developed for the computing system. Also, users may resist having to respond to several questions at each log-on attempt. However, user authentication by advanced password schemes provide better security than traditional password mechanisms. (Smith, 1987; Zviran and Haga, 1990b; Jobusch et al., 1980)

C. ALTERNATIVE AUTHENTICATION MECHANISMS

When advancing beyond a password security mechanism, a security manager has new options. The sophistication of the advanced scheme varies and depends on the level of security required. Another issue that plays a role in selecting an alternative authentication mechanism are the costs associated with its implementation.

Combining what the user knows (i.e. a password) with what the user possesses (i.e. a token, smart card, biometer device, etc.) may provide the level of security required.

D. RECOMMENDATIONS

While organizations have many options available in the area of access control, user acceptance for any particular authentication mechanism is needed. The level of access control to implement is strictly determined within a particular organization and may be unique to that organization.

APPENDIX

The six Trusted Computer System Evaluation Criteria Divisions and Classes are:

- Division D: Minimal Protection - This division is reserved for those systems that have been evaluated but fail to meet all of the requirements for a higher evaluation division.

- Division C: Discretionary Protection - Classes in this division provide for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate.

- Class C1: Discretionary Security Protection - The Trusted Computing Base (TCB) of a C1 system nominally satisfies the discretionary access security requirements by providing separation of users and data. It incorporates some form of credible controls capable of enforcing access limitations on an individual basis, i.e, ostensibly suitable for allowing users to be able to protect project or private information and to keep other users from accidentally reading or destroying their data. The class C1 environment is expected to be one of cooperating users processing data at the same level(s) of security.

- Class C2: Controlled Access Protection - Systems in this class enforce a more finely grained discretionary access control than C1 systems, making users individually accountable for their actions through login procedures, auditing of security-relevant events, and resource isolation.

- Division B: Mandatory Protection - The notion of a TCB that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules is a major requirement in this division. Systems in this division must carry the sensitivity labels with major data structures in the system. The system developer also provides the security policy model on which the TCB is based and furnishes a specification of the TCB. Evidence must be provided to demonstrate that the reference monitor concept has been implemented.

- Class B1: Labeled Security Protection - Class B1 systems require all the features requires for a class C2. In addition, an informal statement of the security policy model, data labeling, and mandatory access control over named subjects and objects must be present. That capability must exist for accurately labeling exported information. Any flaws identified by testing must be removed.

- Class B2: Structured Protection - In class B2 systems, the TCB is based on a clearly defined and documented formal security model that requires the discretionary and mandatory access control enforcement found in class B1 systems be extended to all subjects and objects in the ADP system. In addition, covert channels are addressed. The TCB must be carefully structured into protection-critical and non-protection-critical elements. The TCB interface is well defined and the TCB design and implementation enable it to be subjected to more thorough testing and more complete review. Authentication mechanisms are strengthened, trusted facility management is provided in the form of support for system administrator and operator functions, and stringent configuration management controls are imposed. The system is relatively resistent to penetration.

- Class B3: Security Domains - The class B3 TCB must satisfy the reference monitor requirements that it mediate all accesses of subjects to objects, be tamper-proof, and be small enough to be subject to analysis and tests. To this end, the TCB is structured to exclude code not essential to security policy enforcement, with significant system engineering during the TCB design and implementation directed toward minimizing its complexity. A security administrator is supported, audit mechanisms are expanded to signal security-relevant events, and system recovery procedures are required. The system is highly resistent to penetration.

- Division A: Verified Protection - This division is characterized by the use of formal security verification methods to assure that the mandatory and discretionary security controls employed in the system can effectively protect classified or other sensitive information stored or processed by the system. Extensive documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development and implementation.

- Class A1: Verified Design - Systems in A1 are functionally equivalent to those in class B3 in that no additional architectural features or policy requirements are added. The distinguishing feature of systems in this class is the analysis derived from formal design specification and verification techniques and the resulting high degree of assurance that the TCB is correctly implemented. This assurance is developmental in nature, starting with a formal model of the security policy and a formal top-level specification (FTLS) of the design. In keeping with the extensive design and development analysis of the TCB required of systems in class A1, more stringent configuration management is required and procedures are established for securely distributing the systems to sites. A system security administrator is supported. (IS Security Products and Services Catalogue, 1989)

LIST OF REFERENCES

Ahituv, N., Lapid, Y. and Neumann, S., "Verifying the Authentication of an Information System User," Computers and Security, Vol. 6, No. 2, April, 1987, pp. 152-157.

Avarne, S., "How to Find Out a Password," DP&CS, Vol. 12, No. 2, Spring, 1988, pp. 16-17.

Beedenbender, M. G., "A Comparison of Password Techniques," Master's Thesis, Naval Postgraduate School, Monterey, California, March 1990.

Betts, M., "NBS Releases Standards for Managing Password Security," Computerworld, Vol. 19, No. 28, July 15, 1985, pp. 19.

Bishop, M., "A Proactive Password Checker," Proceedings of the Seventh International Conference on Information Security, Brighton, England, May 1991, pp. 152-163.

Bosen, R. J., "Dynamic Password Security Systems Provide Complete Protection," Hardcopy, Vol. 6, No. 10, October, 1985, pp. 187-190.

Brancheau, J. C. and Wetherbe, J. C., "Key Issues in Information Systems Management," MIS Quarterly, Vol. 11, No. 1, March 1987, pp. 23-36.

Candia, T., "How VMS keeps Out Intruders," Computers and Security, Vol. 9, No. 6, 1990, pp. 499-502.

Carroll, J. M., Computer Security, 2d ed, Buttsworth Publishers, Stoneham, MA., 1987.

Carroll, J. M., Mowat, R. B., Robbins, L. E., and Wiseman, D., "The Password Predictor-A Training Aid for Raising Security Awareness," Computers and Security, Vol. 7, No. 5, 1988, pp. 475-481.

Courtney, L., "Achieving Mainframe Security in the HP Mini Environment," ISPNews, Vol. 2, No. 4, July/August 1991, pp. 38-40.

Denning, D. E., and Denning, P. J., "Security," Computing Surveys, Vol. 11, No. 3, September, 1979, pp. 227-247.

"Department of Defense Trusted Computer System Evaluation Criteria," DoD Computer Security Center, Fort George C. Meade, MD., CSC-STD-001-083, 1985.

"Department of Defense Password Management Guideline," DoD Computer Security Center, Fort George C. Meade, MD., CSC-STD-002-85, 1985.

Dickson,G. W., Leitheiser, R. L., Nechis, M., and Wetherbe, J. C., "Key Information System Issues for the 1980's" MIS Quarterly, Vol. 8, No. 3, September, 1984, pp. 135-148.

Gish, J., "Salting the Password," Infosystems, Vol. 32, No. 4, April 1985, pp. 88-89.

Haga, W. J., Hulsey, J. D. and Zviran, M., "Cognitive Passwords: From Theory to Practice," Working Paper No. 89-06, Naval Postgraduate School, Monterey, CA., 1989, pp. 1-16.

Henderson. S. C., "A Comparison of Data Access Control Packages: Part I," Computer Security Journal, Vol. 4, No. 2, 1987, pp. 75-111.

Henderson. S. C., "A Comparison of Data Access Control Packages: Part II," Computer Security Journal, Vol. 5, No. 1, 1988, pp. 67-104.

Highland, H. J., "Premature Demise of Passwords," Computers and Security, Vol. 9, No. 2, 1990, pp. 102-104.

Highland, H. J., "Demise of Passwords...Part II," Computers and Security, Vol. 9, No. 3, 1990, pp. 196-200.

Highland, H. J., "With Tokens, It's a New Password Every Time," Computerworld, Vol. 24, No. 24, June 11, 1990, pp. 88-89.

Highland, H. J., "If the Password's 'Anything Goes,' It's Your Loss," Government Computer News, Vol. 9, No. 23, October 29, 1990, pp. 61-66.

Holmes, J. P., Maxwell, R. L., Wright, L. J., A Performance Evaluation of Biometric Devices, Sandia National Laboratories Report, July 1990.

Hutt, A. E., Bosworth, S. and Hoyt, D. B., Computer Security Handbook, MacMillan Publishing Company, New York, NY., 1988.

Jobusch, D. L. and Oldehoeft, A. E., "A Survey of Password Mechanisms: Weaknesses and Potential Improvements. Part 1," Computers and Security, November 1989, Vol. 8, No. 7, pp. 587-603.

Jobusch, D. L. and Oldehoeft, A. E., "A Survey of Password Mechanisms: Weaknesses and Potential Improvements. Part 2," Computers and Security, December 1989, Vol. 8, No. 8, pp. 675-689.

Kanner, N., "Retina Scanners: Eye-secure Identification," MIS Week, Vol. 11, No. 13, March 26, 1990, pp. 34.

Kilgallen, L., "VMS: The Security Product You Already Have," ISPNews, Vol. 2, No. 4, July/August 1991, pp. 34-36.

Kochanski, M., "How Safe Is It?," Byte, Vol. 14, No. 6, June, 1989, pp. 257-264.

Lauchlan, S., "Security Threats Go Unchecked," Computing, June 27, 1991, pp. 2.

Lewis, P. H., "When the Password is a Passkey," The New York Times, Vol. 137, No. 47,275, September 27, 1987, pp. 12F.

Lonsford, E. H., "Unix Security: Fact or Fiction," Datamation Vol 36, No.4, February 15, 1990, pp. 44-48.

McFarland, Jr., C. E., Frey, T. J., and Rhodes, D. D., "Retrieval of Internally versus Externally Generated Words in Episodic Memory," Journal of Verbal Learning and Verbal Behavior, Vol. 19, 1980, pp. 210-225.

Mclellan, V., "Password Security--Crypto in Your VAX," Digital Review, Vol. 3, No. 17, October 13, 1986, pp. 86.

Menkus, B., "Understanding the Use of Passwords," Computers and Security, Vol. 7, No. 2, April, 1988, pp. 132-136.

Miller, J. E. and Cox, J. L., "The Good Old Days Versus Today: The Changed Hardware and Software Security Requirements with Automation," Computers Ind. Engineering, Vol. 17, No. 1-4, 1989, pp. 404-409.

Morris, R. and Thompson, K., "Password Security: A Case History," Communications of the ACM, Vol. 22, No. 11, November, 1979, pp. 594-597.

Murray, W. H., "Good Computer Security Practices for Two Areas of Current Concern: Personal Computer and Dial-up Systems," Computers and Security, Vol. 2, No. 2, Fall-Winter 1983, pp. 77-88.

Information Systems Security Products and Services Catalogue, U.S. Government Printing Office, Washington, DC, October, 1989.

Paans, R., and Bonnes, A. H. J., "Surreptitious Security Violations in MVS Systems," Computer and Security, Vol. 2 No. 2 1983, pp. 144-152.

Paans, R., "With MVS/ESA Security Labels Towards B1," Computers and Security, Vol. 10, No. 4, June 1991, pp. 309-324.

Parks, J. R., "Personal Identification - Biometrics," Proceedings of the Seventh International Conference on Information Security, Brighton, England, May 1991, pp. 173-183.

Penzias, A., "'Voice Lock' Key to Future Security," MIS Week, Vol. 11, No. 11, March 12, 1990, pp. 34.

Pfleeger, C. P., Security in Computing, Prentice-Hall, Englewood Cliffs, N.J., 1989.

Porter, S. N., "A Password Extension for Improved Human Factors," Computers and Security, Vol. 1, No. 1, January, 1982, pp. 54-56.

Riddle, B. L., Miron, M. S. and Semo, J. A., "Passwords in Use in a University Timesharing Environment," Computers and Security, Vol. 8, No. 7, November 1989, pp. 569-578.

Rogers, T. B., Kuiper, N. A., and Kirker, W. C., "Self-Reference and Encoding of Personal Information," Journal of Personality and Social Psychology, Vol. 35, 1977, pp. 677-688.

Schwartz, K. D., "UNIX Security Product Gives Navy Tighter Control," Government Computer News, Vol. 9, No. 22, October 15, 1990, pp. 58.

Slamecka, N. J., and Graf, P., "The Generation Effect: Delineation of a Phenomenon," Journal of Experimental Psychology: Human Learning and Memory, Vol. 4, 1978, pp. 592-604.

Smith, S. L., "Authenticating Users by Word Association," Computers and Security, Vol 6, No. 6, December 1987, pp. 464-470.

Spender, J. C., "Identifying Computer Users with Authentication Devices (Tokens)," Computers and Security, Vol. 6, No. 5, 1987, pp. 385-395.

Weiss, K. P., "When a Password is Not a Password," Access, First Quarter, 1990, pp. 10-37.

Wilkes, M. V., "Computer Security in the Business World," Communications of the ACM, Vol. 33, No. 4, April, 1990, pp. 399-401.

Wilson, D. R., "Trends in Information Security," Computer Security Journal, Vol. 4, No. 2, 1987, pp. 29-38.

Wilson, J. L., Turban, E., and Zviran, M., "Information System Security: A Managerial Perspective," Working Paper, Naval Postgraduate School, December, 1990, pp. 1-28.

Wood, C. C., "Effective Information System Security with Password Controls," Computers and Security, Vol. 2, No. 1, 1983, pp. 5-10.

Wood, C. C., "The Human Immune System as an Information Systems Security Reference Model," Computers and Security, Vol. 6, No. 6, 1987, pp. 511-516.

Wood, C. C., "To Guess or Not to Guess," ISPNews, Vol. 1, No. 3, September/October, 1990, pp. 31-32.

Zviran, M. and Haga, W. J., "Password Security: an Exploratory Study," Technical Report, Naval Postgraduate School, Monterey, CA., May 1990a, pp. 1-26.

Zviran, M. and Haga, W. J., "Cognitive Passwords: The Key for Easy Access Control," Computers and Security, Vol. 9, No. 8, December, 1990b, pp 723-736.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Moshe Zviran, Code AS/ZV Naval Postgraduate School Monterey, CA 93943-5000	1
4.	William J. Haga, Code AS/HA Naval Postgraduate School Monterey, CA 93943-5000	1
5.	LCDR John A. Coley Navy and Marine Corps Reserve Center BLDG 47, Dickman Ave Des Moines, IA 50315	1

Thesis
C53473 Coley
c.1 User authentication.