```
30     CALL DES(0.IDBIN,IDBIN)
C      NOW PACK UP THE RESULTS OF THE ENCRYPTION INTO PASSWD. CALL PACK( 8,8, IDBIN,
       PASSWD)
C      ENCRYPTED  PASSWORD  NAY  NOW  BE  STORED  (PRINTED  FOR  DEMONSTRATION).  WRITE(6,300)
       (PASSWD(M),M-1,8)
C      Z FORMAT PRINTS HEXADECIMAL CHARACTERS.
300    FORMAT(' THE ENCRYPTED PASSWORD IS: ,8Z2.2)
       GO TO 1
       END



       SUBROUTINE TRANSF(LEN, STRING)
C      TRANSFORM A LONG STRING INTO A STRING OF 8 CHARACTERS INTEGER LEN,STRING(*),KEYBIN(64)
       ,TEMP(128), IV(64)

C      LOAD THE FIRST 8 BYTES AS KEY AFTER SHIFTING EACH BYTE LEFT,
C      UNPACKING AND COMPUTING ODD PARITY.
      DO 10 1-1,8
10     TEMP(I)-2*STRING(I)
       CALL UNPBIN( 8 , 8 , TEMP , KEYBIN)
       CALL PARITY(KEYBIN)
C      CALL PACK(8,8,KEYBIN,TEMP)
C      WRITE(6,500) (TEMP(M),M=1,8)
500    FORMAT(' THE KEY FOR THE VIRTUAL TRANSFORMATION IS: ',8Z2.2) CALL SETKEY(KEYBIN)

C      CBC ENCRYPT THE REST OF THE STRING
       DO 20 1=1,64
20     IV(I)=O
       CALL CBC(IV,LEN-8,STRING(9) STRING)

C      DETERMINE THE VIRTUAL ENTITY.  ACCEPT ONLY THE 95 LEGAL CHARACTERS C (32-126)
       CALL UNPBIN( 8 , 8, STRING, TEMP) J=1
       DO 50 1=1,8
C      GET ADDITIONAL BITS IF NECESSARY
30     IF(J.GE.59) CALL DES(0,TEMP,TEMP(65))
C      EXTRACT 7 ASCII BITS AND CHECK FOR VALIDITY
       CALL PACK(1,7,TEMP(J),STRING(I))
       IF((STRING(I).EQ.127).OR.(STRING(I).LT.32)) THEN
C     SHIFT TO NEXT 7-BIT SET IF ILLEGAL
       J=J+1
       GO TO 30
       ENDIF
C      OTHERWISE, SHIFT TO NEXT 7-BIT SET
50     J=J+7
       RETURN
       END


       SUBROUTINE CBC(IV,LEN, PLAIN,MAC)
C      COMPUTES A 64-BIT MAC ON PACKED PLAINTEXT (LENGTH-LEN)
C      LEN MUST BE A MULTIPLE OF 8 AND PLAIN MUST BE PADDED.
       INTEGER IV(64).LEN,PLAIN(*),MAC(8).INP(64).OUTP(64)
C      USE INITIAL VECTOR

  10       DO 10 1-1,64
           DO 30 I-1,LEN,8
           CALL UNPBIN(8,8,PLAIN(I) .INP)
           DO 20 J-1,64
  20       INP(J)-MOD(OUTP(J)+INP(J),2)
  30       CALL DES(0,INP,OUTP)
           CALL PACK(8,8,OUTP,MAC)
           RETURN
           END
```

```
          SUBROUTINE HEXKEY(PWLEN, STRING,HEXFLG)
C         CHECK WHETHER THE ENTERED STRING IS HEXADECIMAL WITH CORRECT PARITY
          TO SERVE AS A KEY
          INTEGER PWLEN, STRING( *) ,HEXFLG,TEMP(8) ,TEMPI(64)

C         TEST FOR HEXADECIMAL. RETURN IF NOT, ELSE CONVERT
          HEXFLG=0
          DO 5 1=1,8
 5        TEMP(I)=0
          DO 10 1=1,16

          TEMP(J)=16*TEMP(J)+STRING(I)
          IF((STRING(I).GE.48).AND.(STRING(I).LE.57)) THEN
                         TEMP(J)=TEMP(J)-48
                         ELSE IF((STRING(I).GE.65).AND.(STRING(I).LE.70)) THEN
                         TEMP(J)=TEMP(J)-55
                         ELSE IF((STRING(I).GE.97).AND.(STRING(I).LE.102)) THEN
                         TEMP(J)=TEMP(J)-'87
                         ELSE
                         RETURN
                         ENDIF
10        CONTINUE

          HEXFLG= 1
          PWLEN=8
          DO 30 I=1,8
 30       STRING(I)=TEMP(I)
          RETURN
          END


          SUBROUTINE TSTVAL( LEN, BUFFER, STATUS)
C         MASK CHARACTERS, CHECK FOR VALIDITY AND DETERMINE BUFFER LENGTH INTEGER
  LEN, BUFFER(*) , STATUS
          STATUS=0
          DO 10 I=LEN,1,-1
          BUFFER(I)=BUFFER(I)-(BUFFER(I)/128) *128
          IF((BUFFER(I).EQ.127).OR.(BUFFER(I).LT.32)) THEN
                         WRITE(6, 100) BUFFER(I)
100                      FORMAT(' UNACCEPTABLE ASCII CHARACTER: ,Z2.2)
                         DO 5 J=1,LEN
5                        BUFFER(J)='
                         STATUS=- 1
                         RETURN
                         ENDIF
          IF((STATUS.EQ.O).AND.(BUFFER(I).NE.32)) STATUS=I
10        CONTINUE
          RETURN
          END

          SUBROUTINE PACK(LEN, BPW, BINARY, PACKED)
C         SUBROUTINE TO PACK FROM BINARY VECTOR TO PACKED
C         VECTOR OF LENGTH LEN FILLING THE LEAST SIGNIFICANT BPW BITS. INTEGER
          BINARY( *) , PACKED( *)
          INTEGER LEN,BPW
          K=0
          DO 20 I=1,LEN
          ITEM=0
          DO 10 J=1,BPW
          K=K+1
10        ITEM=ITEM*2+MOD(BINARY(K) ,2)
20        PACKED(I)=ITEM
          END
```

```
                    SUBROUTINE UNPBIN (LEN, BPW,PACKED, BINARY)
O         C        SUBROUTINE TO UNPACK INTO BINARY VECTOR
O         C        FROM PACKED VECTOR OF LENGTH N CONTAINING BPW BITS IN EACH WORD. INTEGER
                   PACKED(*), BINARY(*)
                    INTEGER LEN,BPW
                    K=0
                    DO 100 I=1,LEN
                    ITEM=MOD(PACKED(I) ,2**BPW)
                    DO 100 J-1,BPW
                    IEX=2**(BPW-J)
                    IBIT=ITEM/ IEX
                    K-K+1
                    BINARY(K)=IBIT
          100       ITEM=ITEM-IBIT*IEX
                    END


                    SUBROUTINE PARITY (VECTOR)
          C         COMPUTE ODD PARITY ON A 64-BIT UNPACKED VECTOR INTEGER VECTOR( 64)
                    DO 10 1=8,64,8
                    VECTOR(I)=1
                    DO 10 J=1,7
          10        VECTOR(I)=MOD(VECTOR(I)+VECTOR(I+J-8) ,2) RETURN
                    END


          C         PROGRAMS PRODUCED BY THE NATIONAL BUREAU OF STANDARDS
          C         NOT SUPPORTED BY NBS AND NOT SUBJECT TO COPYRIGHT
          C         SUBROUTINE TO PERFORM THE DATA ENCRYPTION STANDARD ALGORITHM SUBROUTINE
                    DES ( SWITCH, INPUT , OUTPUT)
          C         SWITCH = O CAUSES ENCRYPTION; SWITCH = 1 CAUSES DECRYPTION.
          C         INPUT IS PLAINTEXT FOR ENCRYPTION, CIPHERTEXT FOR DECRYPTION.
          C         OUTPUT IS CIPHERTEXT AFTER ENCRYPTION. PLAINTEXT AFTER DECRYPTION.

          C         CALL DES(0,PT,CT) IS AN EXAMPLE OF CALL TO ENCRYPT PT INTO CT
          C
          C         CALL DES(1,CT,PT) IS AN EXAMPLE OF CALL TO DECRYPT CT INTO PT
          C
          C         THIS PROGRAM SHOULD NOT BE EXPORTED FROM THE UNITED STATES. IMPLICIT INTEGER (A-
                    z)
                    INTEGER INPUT(64) ,OUTPUT(64)
                    INTEGER KS(48,16)
          O         KS IS AN ARRAY TO HOLD THE 16 SUBKEYS FOR 16 ROUNDS OF DES
          C         KS IS COMPUTED BY SETKEY SUBROUTINE WHICH MUST BE CALLED BEFORE
          C         THE DES SUBROUTINE IS CALLED.  SETKEY IS CALLED ONCE TO SET THE
          C         KEY AND THEN DOES NOT HAVE TO BE CALLED AGAIN UNTIL THE KEY IS
          C         CHANGED.
          C
                    COMMON KS
                    INTEGER LR(64),L(32),R(32)
                    EQUIVALENCE (LR(1),L(1)), (LR(33),R(1))
                    INTEGER TEMPL(32),EX(48),F(32)
          C         IP IS THE INITIAL PERMUTATION FOR THE DATA INTEGER IP(64)/
                   1 58,50,42,34,26,18,10,02.
                   1 60,52,44.36.28.20.12,04,
                   1 62,54,46,38,30,22,14,06,
                   1 64,56,48,40,32,24,16,08,
                   1 57,49,41,33,25,17,09,01,
                   1 59,51,43,35,27,19,11,03,
                   1 61,53,45,37,29,21,13,05,
                   1 63,55,47,39,31,23,15,07/
```

```
C  IPINV IS THE INVERSE OF THE INITIAL PERMUTATION
C  INTEGER IPINV(64)/
C  1 40,08,48,16,56,24,64,32,
C  1 39,07,47,15,55,23,63,31,
C  1 38,06,46,14,54,22,62.30,
C  1 37,05,45,13,53,21,61,29,
C  1 36,04,44,12,52,20,60,28,
C  1 35,03,43,11,51,19,59,27,
C  1 34,02,42,10,50,18,58,26,
C  1 33,01,41,09,49,17,57,25/
C  NOTE: THE REVERSED 1P INVERSE TABLE -REVIPI- IS USED BECAUSE
C  IT SAVES THE TIME OF REVERSING THE L AND R REGISTERS.
    INTEGER REVIPI (64)/
C  1 08,40,16,48,24,56,32,64,
    1 07,39,15,47,23,55,31,63,
   1 06,38,14,46,22,54,30,62,
   1 05,37,13,45,21,53,29,61,
   1 04,36,12,44,20,52,28,60,
   1 03,35,11,43,19,51,27,59,
   1 02,34,10,42,18,50,26,58,
   1 01,33,09,41,17,49,25,57/
C  E IS THE EXPANSION OPERATION WHICH EXPANDS 32 BITS TO 48 BITS
    INTEGER E(48)/
   1 32,01,02,03,04,05,
   1 04,05,06,07,08,09,
   1 08,09,10,11,12,13,
   1 12,13,14,15,16,17,
   1 16,17,18,19,20,21,
   1 20,21,22,23,24,25,
   1 24,25,26,27,28,29,
   1 28,29,30,31,32,01/
C  P IS THE PERMUTATION USED IN THE MAIN DES FUNCTION
    INTEGER P(32)/
   1 16,07,20,21,
   1 29,12,28,17,
   1 01,15,23,26,
   1 05,18,31,10,
   1 02,08,24,14,
   1 32,27,03,09,
   1 19,13,30,06,
   1 22,11,04,25/
C  THE EIGHT SUBSTITUTION TABLES ARE USED IN THE DES TO SUBSTITUTE 4 BITS
C   FOR SIX BITS.  ONE S TABLE IS USED FOR EACH 6 BIT TO 4 BIT TRANSFORMATION. INTEGER
    5(64,8)
    EQUIVALENCE (S(1,1),S1(1)), (S(1,2),S2(1))
    EQUIVALENCE (S(1,3),S3(1)), (S(1,4),S4(1))
    EQUIVALENCE (S(1,5),S5(1)), (S(1,6),S6(1))
    EQUIVALENCE (S(1,7),S7(1)), (S(1,8),S8(1))
    INTEGER SI(64)/
   1 14,04,13,01,02,15,11,08,03,10,06,12,05,09,00,07,
   1 00,15,07,04,14,02,13,01,10,06,12,11,09,05,03,08,
   1 04,01,14,08,13,06,02,11,15,12,09,07,03,10,05,00,
   1 15,12,08,02,04,09,01,07,05,11,03,14,1O,00,06,13/
    INTEGER S2(64)/
   1 15,01,08,14,06,11,03,04,09,07,02,13,12,00,05,10,
   1 03,13,04,07,15,02,08,14,12,00,01,10,06,09,11,05,
   1 00,14,07,11,10,04,13,01,05,08,12,06,09,03,02,15,
   1 13,08,10,01,03,15,04,02,11,06,07,12,00,05,14,09/
    INTEGER S3(64)/
   1 10,00,09,14,06,03,15,05,01,13,12,07,11,04,02,06,
   1 13,07,00,09,03,04,06,10,02,08,05,14,12,11,15,01,
   1 13,06,04,09,08,15,03,00,11,01,02,12,05,10,14,07,
   1 01,10,13,00,06,09,08,07,04,15,14,03,11,05,02,12/
```

```
          INTEGER S4(64)/
         1 07,13,14,03,00,06,09,10,01,02,08,05,11,12,04,15,
         1 13,08,11,05,06,15,00,03,04,07,02,12,01,10,14,09,
         1 10,06,09,00,12,11,07,13,15,01,03,14,05,02,08,04,
         1 03,15,00,06,10,01,13,08,09,04,05,11,12,07,02,14/
          INTEGER SS(64)/
         1 02,12,04,01,07,10,11,06,08,05,03,15,13,00,14,09.
         1 14,11,02,12,04,07,13,01,05,00,15,10,03,09,08,06.
         1 04,02,01,11,10,13,07,08,15,09,12,05,06,03,00,14,
         1 11,08,12.07,01,14,02,13,06,15,00,09,10,04,05,03/
          INTEGER S6(64)/
         1 12,01,10,15,09,02.06,08,00, 13,03,04,14,07,05,11,
         1 10,15,04,02,07.12,09,05,06,01, 13,14,00,11,03,08,
         1 09,14,15,05,02,08,12,03.07,00,04,10,01,13,11,06,
         1 04,03,02,12,09,05,15,10,11,14,01,07,06,00,08,13/
          INTEGER S7(64)/
         1 04,11,02, 14,15,00,08,13,03,12,09,07,05,10,08,01,
         1 13,00,11,07,04,09,01,10, 14,03,05,12,02,15,08,06.
         1 01,04,11,13,12,03,07,14,10, 15,06,08,00,05,09.02.
         1 06. 11,13,08,01,04,10.07,09,05,00,15,14,02,03,12/
          INTEGER S8(64)/
         1 13,02,08,04,06,15,11,01,10,09,03,14,05,00,12,07,
         1 01,15,13,08,10,03,07,04,12.05,06.11,00, 14,09,02,
         1 07,11,04,01,09,12,14,02,00,08,10,13,15,03,05,08,
         1 02,01.14.07,04.10,08,13,15,12,09.00,03,05.06,11/


C
C*** SWITCH=0 IS ENCRYPTION; SWITCH=1 IS DECRYPTION MODE
C
         N1=1
         N2=16
         N3=1
          IF (SWITCH .EQ. 0)GO TO 20
         N1=16
         N2=1
         N3=-1
C         LOOP WHICH DOES THE INITIAL PERMUTATION OF THE INPUT
20        DO 50 1=1,64
50        LR(I)=INPUT(IP(I))
C         MAIN LOOP WHICH ENCRYPTS OR DECRYPTS FOR 16 ROUNDS
          DO 500 N=N1,N2,N3
C         LOOP WHICH SAVES THE R REGISTER DO 75 1=1,32
75        TEMPL(I)=R(I)
C         LOOP WHICH EXPANDS THE R REGISTER USING THE E FUNCTION
C         AND DOES THE XOR OF THE KEY SCHEDULE SUBKEY AND THE EXPANDED R. DO 100
          1=1,48
          EX(I)=1
100       IF(R(E(I)) .EQ. KS(I.N)) EX(I)=0
C         LOOP WHICH DOES THE SUBSTITUTIONS USING THE 8 S TABLES.
         DO 200 J=0,7
          K=J*6+1
          IN=EX(K) *32+EX(K+5) * 16+EX(K+1 ) *8+EX(K+2) *4+EX(K+3) *2+EX(K+4)
          SUB=S(IN+1 .J+1)
          K=J*4
          F(K+1) = SUB/8
          F(K+2) = MOD(SUB,8)/4
          F(K+3) = MOD(SUB,4)/2
```

```
     200 F(K+4) - MOD(SUB,2)
C         LOOP WHICH DOES THE P PERMUTATION OPERATION TO THE 32-BIT RESULT
C         AND ALSO DOES THE XOR OF THE OLD L AND THE NEW RESULT OF THE F FUNCTION.
          DO 300 J=1,32
         R(J)-1
300       IF (L(J) .EQ. F(P(J))) R(J)-0
```

```
C         SETS THE NEW L TO BE THE OLD R THAT HAD BEEN SAVED.
          DO 400 J=1,32
400       L(J)=TEMPL(J)
500       CONTINUE
C         DOES THE INVERSE PERMUTATION AFTER THE 16 ROUNDS TO COMPLETE THE DES. DO
          600 J=1,64
600       OUTPUT(J)=LR(REVIPI(J))
          END
          SUBROUTINE SETKEY (KEY)
C         SUBROUTINE TO PERFORM DES KEY SCHEDULE
C         PRODUCED BY THE NATIONAL BUREAU OF STANDARDS
C         NOT SUBJECT TO COPYRIGHT - NOT SUPPORTED BY NBS
          IMPLICIT INTEGER (A-Z)
          INTEGER KEY(64)
C         KS IS THE COMMON AREA WITH THE DES SUBROUTINE.  SETKEY MUST SET KS
C         BEFORE THE DES IS CALLED OR GARBAGE WILL RESULT FROM THE DES. INTEGER
          KS(48.16).CD(56).C(28).D(28)
          COMMON KS
          EQUIVALENCE (CD(1), C(1)), (CD(29),D(1))
C         PCI IS THE PERMUTED CHOICE 1 DEFINED IN THE DES
          INTEGER PCI(56)/
         1 57,49,41,33,25,17,09,
         1 01,58,50,42,34,26,18,
         1 10,02,59,51,43,35,27,
         1 19,11,03,60.52,44,36,
         1 63,55,47,39,31,23,15,
         1 07,62,54,46,38,30,22,
         1 14,06,61,53,45,37,29,
         1 21,13,05,28,20,12,04/
C         PC2 IS THE PERMUTED CHOICE 2 DEFINED IN THE DES.
          INTEGER PC2(48)/
          1 14,17,11,24,01,05,
         1 03,28,15,06,21,10,
         1 23,19,12,04,26,08,
         1 16,07,27,20,13,02,
         1 41,52,31,37,47,55,
         1 30,40,51,45,33,48,
         1 44,49,39,56,34,53,
         1 46,42,50,36,29,32/
C         KSFT IS THE KEY SHIFT VECTOR DEFINED IN THE DES.
         INTEGER KSFT(16)/1.1,2,2,2,2,2.2,1,2,2,2,2,2,2,1/
C         LOOP WHICH PERFORMS THE FIRST PERMUTATION ON THE KEY, REMOVING THE
C         PARITY BITS AND SETTING UP THE C AND D REGISTERS.
         DO 100 1=1,56
100       OD(I)=KEY(PCI(I))
C         LOOP WHICH COMPUTES THE KS ARRAY FOR THE DES.  THIS TECHNIQUE USES
C         LOTS OF MEMORY BUT RUNS FAST.  THE KS NEED ONLY BE COMPUTED ONCE
C         FOR ALL THE ENCRYPT AND DECRYPT OPERATIONS WHICH USE THIS KEY. DO 500
          1=1,16
          DO 300 J=1,KSFT(I)
          CT=C(1)
          DT=D(1)
```

```
       DO 200 K=1,27
        C(K)=C(K+1)
200     D(K)=D(K+1)
       C(28)=DT
300    D(28)=DT
       DO 400 J=1,48
400     KS(J,I)=CD(PC2(J))
500     CONTINUE
END
```

APPENDIX E

PASSWORD MANAGEMENT GUIDELINE

This appendix contains the complete Department of Defense Password Management Guideline issued by the DoD Computer Security Center. It is included as a part of the Password Usage Standard as additional information and guidance that may be used when implementing a password security system. This guideline was not available for coordination with the Password Usage Standard but it, like the other appendices, is not a required part of the Standard.

This guideline provides a set of good practices related to the use of password-based user authentication mechanisms in automatic data processing systems. While it was originally issued for systems processing classified and national security related information, it is also useful for application in systems processing sensitive, fragile or critical information (i.e., information that must be protected from unauthorized disclosure, modification or destruction). Comments on this guideline should be directed to the Office of Standards and Products, DoD Computer Security Center, Fort George G. Meade, Maryland 20755.

This guideline is the result of the work of numerous individuals. Sheila L. Brand, DoD Computer Security Center (DoDCSC) and Jeffrey D. Makey, formerly DoDCSC, are recognized as principal authors. Additional contributions were made by: Daniel J. Edwards, Mary B. Flaherty, Steven J. Padilla, all of the DoDCSC; John J. Stasak III, Gregory Wessel and Bernard Peters, all of the DoD; Roger R. Schell, formerly DoDCSC; and James P. Anderson. These people contributed to the formulation and review of the guideline.

1. Introduction

In August 1983, the DoD Computer Security Center published CSC-STD-001-83, Department of Defense Trusted Computer System Evaluation Criteria. That 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:

> "Individual accountability is the key to securing and controlling any system that processes information on behalf of individuals or groups of individuals. A number of requirements must be met in order to satisfy this objective."
>
> "The first requirement is for individual user identification. Second, there is a need for authentication. Without authentication, user identification has no credibility. Without a credible identity (no) ... security policies can be properly

invoked because there is no assurance that proper authorizations can be made." [2]

This guideline has been developed to assist in providing that much needed credibility of user identity by presenting a set of good practices related to the design, implementation and use of password-based user authentication mechanisms. It is intended that features and practices described in this guideline be incorporated into DoD automatic data processing (ADP) systems used for processing classified or other sensitive information.

2. Scope

The security provided by a password system depends on the passwords being kept secret at all times. Thus, a password is vulnerable to compromise whenever it is used, stored, or even known. In a password-based authentication mechanism implemented on an ADP system, passwords are vulnerable to compromise due to five essential aspects of the password system: 1) a password must be initially assigned to a user when enrolled on the ADP system; 2) a user's password must be changed periodically; 3) the ADP system must maintain a "password database"; 4) users must remember their passwords; and 5) users must enter their passwords into the

ADP system at authentication time. This guideline prescribes steps to be taken to minimize the vulnerability of passwords in each of these circumstances.

Specific areas addressed in this guideline include the responsibilities of the system security officer and of users, the functionality of the authentication mechanism, and password generation. The major features advocated in this guideline are:

- Users should be able to change their own passwords
- Passwords should be machine-generated rather than user-created
- Certain audit reports (e.g., date and time of last login) should be provided by the system directly to the user

For certain sensitive applications such as Command and Control Systems, pertinent DoD directives should be referenced in order to assess the need for additional identification and authentication features.

3. Control Objectives

The CSC-STD-001-83 gives the following as the Accountability Control Objective:

> "Systems that are used to process or handle classified or other sensitive information must assure individual accountability whenever either a mandatory or discretionary security policy is invoked. Furthermore, to assure accountability, the capability must exist for an authorized and competent agent to access and evaluate accountability information by a secure means, within a reasonable amount of time, and without undue difficulty." [2]

In order to attain the individual accountability required, it is necessary for the ADP system to be able to uniquely identify each person who uses it. In many cases, a password scheme will be used to achieve this. The Accountability Control Objective, applied to password systems, leads to the following control objectives for password systems.

Personal Identification. Password systems used to control access to ADP systems that process or handle classified or other sensitive information must assure the capability to uniquely identify each individual user of the system.

Authentication. Password systems used to control access to ADP systems that process or handle classified or other sensitive information must assure unequivocal authentication of the user's claimed identity.

Password Privacy. Password systems must assure, to the extent possible, protection of the password database consistent with the protection afforded the classified or other sensitive information processed or handled by the ADP system in which the password systems operate.

Auditing. Password systems used to control access to ADP systems that process or handle classified or other sensitive information must be able to assist in the detection of password compromise.

4. Definitions

Access Port-A logical or physical identifier that a computer uses to distinguish different terminal input/ output data streams.

Expired Password-A password that must be changed by the user before login may be completed.

Password-A character string used to authenticate an identity. Knowledge of the password that is associated with an ID is considered proof of authorization to use the capabilities associated with that ID.

Password System -A part of an ADP system that is used to authenticate a user's identity. Assurance of unequivocal identification is based on the user's ability to enter a private password that no one else should know.

System Security Officer (550)-The person responsible for the security of an ADP system. The 550 is authorized to act in the "security administrator" role defined in CSC-STD-001-83. Functions that the 550 is expected to perform include auditing and changing security characteristics of a user.

Trusted Identification Forwarding-An identification method used in networks where the sending host can verify that an authorized user on its system is attempting a connection to another host The sending host transmits the required user authentication information to the receiving host. The receiving host can then verify that the user is validated for access to its system. This operation may be transparent to the user.

User ID-A unique symbol or character string that is used by an ADP system to uniquely identify a user. The security provided by a password system should not rely on secrecy of the user's ID.

5. Guidelines

In the remainder of this document, guidelines for good practice are presented in bold print, while amplifications, examples, and rationale are presented in normal print. The guidelines are given with two degrees of emphasis. Those that are most important to the security of a password system are presented with such wording as "The 550 should ..." (the word "should" is the key), while less critical functions are presented with such wording as "It is recommended that ..." ("recommended" is the key). Because it is anticipated that diverse user communities will adopt this guideline, all recommendations are presented in general rather than specific terminology, divorced from vendor-specific hardware or system software. Where features require the setting of a specific value (e.g., password maximum lifetime), it is suggested that these be designed as parametric settings leaving the determination of exact values to local security management who understand the particular security requirements of their user environment.

It is recommended that, whenever possible, the mechanisms discussed in this guide he automated. Automation will result in a minimal burden on the system administration and on the users, and thus in greater effectiveness of the mechanisms by eliminating situations where passwords might be exposed to people.

5.1 SSO Responsibilities

5.1.1 Initial System Passwords

Many ADP systems come from the vendor with a few standard user

IDs (e.g., SYSTEM, TEST, MASTER, etc.) already enrolled in the system. The System Security Officer (550) should change the passwords for all standard user IDs before allowing the general user population to access the system. This can be easily assured if the standard user IDs are initially identified by the system as having "expired" passwords. (See sec. 3.2.2. 1 for discussion of expired passwords.)

5.1.2 Initial Password Assignment

The 550 is responsible for generating and assigning the initial password for each user ID. The user must then be informed of this password. In some areas, it may be necessary to prevent exposure of the password to the 550. In other cases, the user can easily nullify this exposure.

5.1.2.1 Preventing Exposure

There are methods that can be implemented to prevent exposure of a password to the 550 after it has been generated.

One technique is to print the user's password on a sealed multipart form in such a way that it is not visible on the top page of the form. The 550 would then protect the sealed password appropriately until it could be delivered to the user. In this case, the password is generated randomly by the ADP system and is not known by the 550. The password should he sealed so it is not visible and cannot he made visible without breaking the seal. Delivery of the password in this manner could require several days.

Another method of preventing exposure is to have the user present at password generation. The 550 must initiate the procedure and the user must shield the generated password and then remove or erase it from the display. This method cannot be used when user terminals are at remote locations.

It is recommended that a technique comparable to one of the above he used to prevent exposing a user's initial password to the 550.

Whatever method is used to distribute passwords, the 550 must receive an acknowledgment of receipt of the password within a specified time period.

5.1.2.2 Nullifying Exposure

When a user's initial password must be exposed to the 550, this exposure may be nullified by having the user immediately change the password by the normal procedure. (Presumably, this change procedure does not expose the new password to the 550).

When a user's initial password is not protected from exposure to the 550, the user ID should be identified by the system as having an "expired password" which will require the user to change the password by the usual procedure (see sec. 5.2.2.3) before receiving authorization to access the system.

5.1.2.3 Classification Assignment

Where the password must be classified, the initial classification assignment should be entered by the 550 to designate the highest security level that may be associated with each user's initial password and its successors.

5.1.3 Password Change Authorization

Occasionally, a user will forget the password or the 550 may determine that a user's password may have been compromised. To be able to correct these problems, it is recommended that the 550 be permitted to change the password of any user by generating a new one. The 550 should not have to know the user's password in order to do this, but should follow the same rules for distributing the new password that apply to initial password assignment (see sec. 5. 1.2). Positive identification of the user by the 550 is required when a forgotten password must be replaced.

5.1.4 Group IDs

Throughout the lifetime of an ADP system, each user ID should be assigned to only one person. In other words, no two people may ever have the same user ID at the same time, or even at different times. It should be considered a security violation when two or more people know the password for a user ID (except in the case when the 550 is the other person and the user ID is identified by the system as having an "expired password"). Note that there is no intention of prohibiting alternate forms of user identification (e.g., group IDs, functional titles) for non-authentication purposes (e.g., data access control, mail). If alternate IDs are used, they must be based on user IDs.

5.1.5 User ID Revalidation

The 550 should be responsible for the development of a procedure whereby prompt notification is given to the 550 when a user ID and password must be removed from the ADP system (e.g., when an employee leaves the sponsoring organization). In addition, all user IDs should be revalidated periodically, and information such as sponsor and means of offline contact (e.g., phone number, mailing address) updated as necessary. It Is recommended that this revalidation be done at least once per year.

5.2 User Responsibilities

5.2.1 Security Awareness

Users should understand their responsibility to keep passwords private and to report changes in their user status, suspected security violations, etc. To assure security awareness among the user population, it is recommended that each user be required to sign a statement to acknowledge understanding these responsibilities.

5.2.2 Changing Passwords

The simplest way to recover from the compromise of a password is to change it. Therefore, passwords should be changed on a periodic basis to counter the possibility of undetected password compromise. They should

be changed often enough so that there is an acceptably low probability of compromise during a password's lifetime. To avoid needless exposure of users' passwords to the 550, users should be able to change their passwords without Intervention by the 550.

5.2.2.1 Password Lifetime

The most obvious threat to the security provided by a password system is from the compromise of passwords. The greater the length of time during which a password is used for authentication purposes, the more opportunities there are for exposing it. In a useful password system, the probability of compromise of a password increases during its lifetime. For a period of time, this probability could be considered acceptably low while after a longer period of time, it would be considered unacceptably high. At this latter point, use of the password should be considered suspect rather than a reliable proof of identity. By appropriately limiting the length of time (called the password lifetime) during which a password can be used, the vulnerability of the password can remain acceptable.

There should be a maximum lifetime for all passwords. To protect against unknown threats, it is recommended that the maximum lifetime of a password be no greater than 1 year. The presence of known threats may indicate a need for a shorter maximum lifetime. Also, depending on the size of the password space and on how fast a penetrator can execute a login attempt, it may be necessary to change passwords even more frequently. See Appendix B.3 for a discussion of the relationship between password lifetime, password space, and the guess rate.

A password should be invalidated at the end of its maximum lifetime. It is recommended that, at a predetermined period of time prior to the expiration of a password's lifetime, the user ID it is associated with be notified by the system as having an "expired" password. A user who logs in with an ID having an expired password should be required to change the password for that user ID before further access to the system is permitted. If a password is not changed before the end of its maximum lifetime, it is recommended that the user ID it is associated with be identified by the system as "locked." No login should be permitted to a locked user ID, but the 550 should be able to unlock the user ID by changing the password for that user ID, following the same rules that apply to Initial password entry (see sec. 5.1.2). After a password has been changed, the lifetime period for the password should be reset to the maximum value established by the system.

5.2.2.2 Change Authorization

To be consistent with the Password Privacy control objective, users (other than the 550) should be permitted to change only their own passwords. To ensure this, it is recommended that the user enter the old password and the user ID/password combination be validated as part of the password changing procedure.

5.2.2.3 Change Procedure

Changing a password in a secure manner involves several steps. The following procedure is recommended:

> The procedure should be Invoked at the user's request or when a user logs in with an expired password. If the change is necessary due to an expired password, the user should be so informed. The user should be presented with a brief summary of the major steps in changing a password, including a caution that the user should ensure that no one else is watching what the user is doing. Except when the change procedure is part of the login procedure (e.g., logging in with an expired password), the user's current password should be entered to re-authenticate identity. The change procedure should display a new password for the user. The new password should be different from the old one and should be generated by an algorithm that satisfies the specifications In Appendix E.l. The user should then enter the new password twice so the procedure can verify that the user can consistently enter the password correctly. The new password should be obliterated by techniques such as overprinting or terminal screen erasing. If the two entered passwords are identical to the generated password, the password database should be updated (i.e., the old password deleted or invalidated and the new password associated with the user ID) and a message to this effect should be displayed. Failure by the user to correctly enter the current password or the generated password should result in a useful error message to the user and In the change procedure being aborted without changing the password. When the attempt to change an expired password is not successful, the password should be retained as expired and the user given the option to again change the password or logout. An audit record should be generated that indicates whether or not the change was successful.

5.2.3 Login to a Connected System

Users should be required to authenticate their identities at "login" time by supplying their password along with their user ID. It is recommended that some form of trusted identification forwarding be used between hosts when users connect to other ADP systems through a network. When trusted identification forwarding is not used, a remote host should require the user's ID and password when logging in through a network connection. Note that user IDs on different hosts for the same user may be different, and that corresponding machine-generated passwords almost certainly will be different. Note also that a password required by a remote host is vulnerable to compromise by the local host or intermediate hosts.

5.2.4 Remembering Passwords

Since users must supply their passwords to the ADP system at authentication time, it follows that they must know what their passwords are. It is recommended that users memorize their passwords and not write them on any medium. If passwords must be written, they should be protected in a manner that is consistent with the damage that could be caused by their compromise. See Appendix E.4 for guidance on the protection of passwords.

5.3 Authentication Mechanism Functionality

5.3.1 Internal Storage of Passwords

It is normally necessary for the ADP system to store internally the user ID for each authorized system user as well as some representation of the password and, when required, the clearance and authorizations that at associated with each user ID. Without some form of access control over this information, it will be possible for unauthorized users to read and/or modify the password database. Unauthorized reading and writing of the password database are a concern. Reading it could result in disclosure of passwords to unauthorized users. Being able to write it could result, for example, in user A changing user B's password so user A could login under user B's identity. Note that it is necessary for the login process to be able to read the password database and for the password changing process to be able to read and write the password database.

Stored passwords should be protected by access controls provided by the ADP system, by password encryption, or by both.

5.3.1.1 Use of Access Control Mechanisms

Access control mechanisms (e.g., mandatory or discretionary controls as discussed in CSC-STD-001-83) should be used to protect the password database from unauthorized modification and disclosure.

5.3.1.2 Use of Encryption

Encryption of stored passwords should be used whenever the access control mechanisms provided by the ADP system are not adequate to prevent exposure of the stored passwords. It is recommended that password encryption be used even when other access controls are considered adequate, as this helps protect against possible exposure when access controls are bypassed (e.g., system dumps). When encryption is used to protect stored passwords, it is recommended that the algorithm meet the specifications in Appendix E.2. It is recommended that encryption be done immediately after entry, that the memory containing the plaintext password be erased immediately after encryption, and that only the encrypted password be used in comparisons. There is no need to be able to decrypt passwords. Comparisons can be made by encrypting the password entered at login and comparing the encrypted form with the encrypted password stored in the password database.

5.3.2 Entry

Passwords should be entered after providing a user ID to the system. If the entry is correct, the system should then display the date and time of the user's last login.

It is recommended that the system not echo passwords that users type in. When the system cannot prevent a password from being echoed (e.g., in a half-duplex connection), it is recommended that a random overprint mask be printed before or after the password is entered, as appropriate, to conceal the typed password.

The complete password as entered by the user should be an exact match, character for character, with the user's current password.

5.3.3 Transmission

During transmission of a password from a user's terminal to the computer In which the authentication is done, passwords should be protected In a manner that is consistent with the damage that could be caused by their compromise. Since passwords are no more sensitive than the data they provide access to, there is generally no reason to protect them, during transmission, to any greater degree (e.g., encryption) than regular data is protected. See Appendix B.4 for guidance on the protection of passwords.

5.3.4 Login Attempt Rate

By controlling the rate at which login attempts can be made (where each attempt constitutes a guess of a password), the number of guesses a penetrator can make during a password's lifetime is limited to a known upper bound. To control attacks where a penetrator attempts many logins through a single access port, the password guess rate should be controlled on a per-access port basis. That is, each access port should be individually controlled to limit the rate at which logIn attempts can be made at each port. When a penetrator can easily switch among multiple access ports, it is recommended that the password guess rate also be controlled on a per-user ID basis.

It is recommended that maximum logIn attempt rates fall within the range of one per second to one per minute. This range provides reasonable user-friendliness without permitting so many login attempts that an extremely large password space or an extremely short password lifetime is necessary. See Appendix E.3 for a discussion of the relationship between the guess rate, password lifetime, and password space.

Note that it is not intended that login be an inherently slow procedure, for there is no reason to delay a successful login. However, in the event of an unsuccessful login attempt, it is quite reasonable to use an internal timer to enforce the desired delay before permitting the next login attempt. The user should not be able to bypass this procedure.

5.3.5 Auditing

5.3.5.1 Audit Trails

The system should be able to create an audit trail of password usage and changes. Such an audit trail should not contain actual passwords or character strings that were incorrectly given as passwords, since this could expose the password of a legitimate user who mistyped his user ID or password. Auditable events should include: successful login, unsuccessful login attempts, use of the password changing procedure, and the locking of a user ID due to its password reaching the end of its lifetime. For each recorded event, the audit record should include: date and time of the event, type of event, offered user ID for unsuccessful logins or actual user ID for other events, and origin of the event (e.g., terminal or access port ID). Audit records of password changes should also indicate whether or not the change was successful.

5.3.5.2 Real-time Notification to System Personnel

It is recommended that each accumulation of 5 consecutive unsuccessful login attempts from a single access port or against a single user ID results in immediate notification of the event to the ADP system operator or the 550. While there is no requirement for the 550 or operator to take any action upon receiving the notification, frequent notifications may indicate that a penetration attempt is in progress and may warrant investigation and possible corrective action.

5.3.5.3 Notification to the User

Upon successful logIn, the user should be notified of:

- The date and time of user's last login;
- The location of the user (as can best be determined) at last login; and
- Each unsuccessful login attempt to this user ID since the last successful login.

This provides a means for the user to determine if someone else is using or attempting to guess this user ID and password.

5.4 Password Protection

5.4.1 Single Guess Probability

The probability that any single attempt at guessing a password will be successful is one of the most critical factors in a password system. This probability depends on the size of the password space and the statistical distribution within that space of passwords that are actually used. Since many user-created passwords are particularly easy to guess, all passwords should be machine-generated using an algorithm that meets the specifications in Appendix E.l.

5.4.2 Password Distribution

During distribution to the user, passwords should be protected to the same degree as the information to which they provide access. Machine-generated passwords should be displayed on the user's terminal at the time of change, along with appropriate cautions to the user to protect the password. At the completion of the change procedure, It Is recommended that displayed passwords be erased or overstrike as appropriate for the terminal type. Passwords changed by the 550 should be distributed in a manner that is consistent with the damage that could be caused by their compromise. See Appendix E.4 for guidance on the protection of passwords.

APPENDIX E.1

PASSWORD GENERATION ALGORITHM

This appendix describes the requirements to be met by an acceptable password generation algorithm. The issues involved relate to the specifications for password space, random seed generation, pseudo-random number generation and "user-friendly" passwords.

1. Password Space

The size of the password space is a function of the size of the alphabet and the number of characters from that alphabet that are used to create passwords. (The maximum size of the password space can be expressed as S=AM where S is the maximum password space, A is the alphabet size and M is the password length.) To determine the minimum size of the password space needed to satisfy the security requirements for an operating environment, equation (3) in Appendix B.3 can be used. The password generation algorithm selected should be able to generate at least that number of passwords. In addition, the generated passwords should be, at a minimum, 6 characters in length.

2. Random Seeds

When a pseudo-random number generator is used in a password generation algorithm, it should accept random data as input that would provide output which has a high degree of unpredictability. This random data (seed) can be derived from a number of available parameters such as a system clock, system registers, date, time, etc. The parameters should be selected to ensure that the number of unique seeds that can be generated from these inputs should be at least equal to the minimum number of passwords that must be generated. When passwords are used to protect classified information, the seed generator should be approved by the DoD Computer Security Center.

3. Pseudo-Random Number Generator

Using a random seed as input, the pseudo-random number generator that drives a password generation algorithm should have the property that each bit in the pseudo-random number that it generates is a complex function of all the bits in the seed. The Federal Data Encryption Standard (DES), as specified in FIPS 46, is an example of a pseudo-random number generator with this property. If DES is used, it is suggested that the 64-bit Output Feedback (OFB) mode be used as specified in FIPS 81. In this case, the seed used as input could consist of:

- An initialization vector
- A cryptographic key
- Plain text

Factors that can be used as input to these parameters are:

For the initialization vector:

- System clock
- System ID
- User ID
- Date and time

For the cryptographic key:

- System interrupt registers
- System status registers
- System counters

The plain text can be an external randomly generated 64-bit value (8 characters input by the 550).

The resulting pseudo-random number that is output will be the 64 bits of cipher text generated in the 64-bit OFB mode. The password generation algorithm can either format this pseudo-random number into a password or use it as an index (or indices) into a table and use the contents from this table to form a password or a passphrase.

4. "User-Friendly" Passwords

To assist users in remembering their passwords, the password generation algorithm should generate pass-words or passphrases that are "easy" to remember. Passwords formed by randomly choosing characters are generally difficult to remember. Passwords that are pronounceable are often easy to remember, as are passphrases that are formed by concatenating real words into a phrase or sentence.

APPENDIX E.2

PASSWORD ENCRYPTION ALGORITHM

Password encryption is advocated as a password protection measure. The algorithm selected for this would be determined by the system environment. Some environments may require that a classified encryption algorithm be used, while for other environments an unclassified algorithm would be required.

1. Encryption Algorithm

A conventional or public key cryptographic algorithm which is configured as a "one-way" encryption algorithm may be used for password encryption, but whatever algorithm is used, the protection that the encryption algorithm provides should rely on its complexity. If there is a key that can be used with the algorithm to decrypt passwords, that key should not be stored In the ADP system.

2. Assurance for Unique Encrypted Passwords

If a password encryption system depends only on the password and other fixed information, there is a possibility that two different users will have identical encrypted passwords. A user who discovers another user with an identical encrypted password will then know that the same password will work for both user IDs even if they don't have identical plaintext passwords. To minimize this possibility, it is recommended that the encryption algorithm use the ADP system name (in network environments) and the user's ID as factors in the encryption. (This can be easily accomplished by concatenating the system ID, user ID and password, and then applying the encryption algorithm to the resulting string.)

APPENDIX E.3

DETERMINING PASSWORD LENGTH

The security afforded by passwords is determined by the probability that a password can be guessed during its lifetime. The smaller that probability, the greater the security provided by the password. All else being equal, the longer the password, the greater the security it provides. This appendix reviews the mathematics involved in establishing how long a password should be.

The basic parameters that affect the length of the password needed to provide a given degree of security are:

L=maximum lifetime that a password can be used to log into the system.

P = probability that a password can be guessed within its lifetime, assuming continuous guesses for this period.

R=number of guesses per unit of time that it is possible to make.

S =password space, i.e., the total number of unique passwords that the password generation algorithm can generate.

1. Relationship

Considering only the cases where S is greater than L X R and therefore P is less than 1, the relationship between these parameters is expressed by the equation:

$$P = \frac{LXR}{S}$$

A detailed explanation of the derivation of this basic equation is given in Appendix E.6.

2. Guess Rate

Several factors contribute to the rate at which attempts can be made to gain access to the data on a system when a valid password is not known. First and foremost is the protection given to the password data base itself. If the password data base is unprotected (i.e., can be read by anyone as ordinary data), then "guessing" may not be required.

If the password data base can be read, but the passwords are encrypted (see Appendix E.2), a very high guess rate may be possible by using a computer to try a dictionary of possible passwords to see if ciphertext can be generated that is the same as one in the password data base. A similar situation frequently occurs where only passwords are used to protect files.

Finally, if the password data base has effective access controls and the login procedure cannot be bypassed, the guess rate can be controlled by setting limits on the number of login or other attempts that can be made before terminating the connection or process.

3. Password Lifetime

All other things being equal, the shorter the lifetime of a password, the fewer the number of guesses that can be made and thus the greater the degree of password security. As stated in 5.2.2.1, the maximum password lifetime should not exceed 1 year.

4. Password Space

Password length and alphabet size are factors in computing the maximum password space requirements. Equation (2) expresses the relationship between S, A, and M where:

S =password space
A =number of alphabet symbols
M=password length

$$S=A^M \qquad (2)$$

To illustrate: If passwords consisting of 4 digits using an alphabet of 10 digits (e.g., 0-9) are to be generated:

$$S=10^4$$

That is, 10,000 unique 4-digit passwords could be generated. Likewise, to generate random 6-character passwords from an alphabet of 26 characters (e.g., A-Z):

$$5=26^6$$

That is, 3.089*108 unique 6-character passwords could be generated.

"User-friendly" passwords (sometimes referred to as passphrases) could be generated by using, for example, 3 symbols from an alphabet (dictionary) of 2000 symbols, where each symbol was a pronounceable word of 4, 5, or 6 characters. Using equation (2) and setting:

A =2000 symbols (words)

M=3

Then $S=2000^3$

That is, $8* \ 10^9$ unique passwords could be generated where each password was made up of 3 words taken from a dictionary of 2000 words.

5. A Procedure for Determining Password Length

What is important in using passwords is how long to make the password to resist exhaustive penetration attacks. We can do this by using the following procedure:

a. Establish an acceptable probability, P, that a password will be guessed during its lifetime. For example, when used as a login authenticator, the probability may be no more than 1 in 1,000,000. In another case, where very sensitive data is involved, the value for P may be set at 10^{-20}.

b. Solve for the size of the password space, S, with the equation derived from equation (1)

$$S = \frac{G}{P} \qquad (3)$$

where G=L x R

c. Determine the length of the password, M, from the equation

$$M = \frac{\log S}{\log \text{(number of symbols in the 'alphabet')}} \qquad (4)$$

M will generally be a real number that must be rounded up or down to the nearest whole number. Examples of calculating many of the values described above are given below.

6. Worked Examples

An example shown here is drawn from a real network case. The problem is to determine the needed password length to reduce to an acceptable level the probability that a password will be guessed during its lifetime.

The network to which this is applied supports both a 300-baud and a 1200-baud service. Experiments on the network have determined that it is possible to make about 8.5 guesses per minute on the 300-baud service and 14 guesses per minute on the 1200-baud service. (The reason that the 'guess rate' for the 1200-baud service is not 4 times that of the 300-baud service is that the system response time, which is not affected by the improved transmission speed, becomes the limiting factor in how many guesses can be accomplished in a given amount of time.)

In this example, the arbitrary value of 10^{-6} is used for the probability (P) of guessing the password in its lifetime. As we will see below, the password lifetime is not the critical factor here as long as the password is changed at least once per year.

The statement of the problem is to find a password length that will resist being guessed with a probability of 1 in 10^{-6} in 1 year of continuous guesses.

When three parameters in equation (1) are known, the fourth value can be found. To find the password space required by our examples, the following parameters are given:

L is set for 6 months and 12 months.

P is set for 1 in 1,000,000 (acceptable probability of guessing the password).

R is set at 8.5 guesses per minute (guess rate possible with 300-baud service).

At 8.5 guesses per minute, the number of guesses per day would be 12,240.

Substituting 183 days for 6 months then using equation (3),

$$S=\frac{G}{P}=\frac{183X\ 12240}{.000001}=2.23992x10^{12} \text{ passwords}$$

The 12-month value is twice that of the 6-month case.

With this data, and using equation (4), we can determine the length of the passwords as a function of the size of the alphabet from which they are drawn. We will assume two alphabet sizes: a 26-letter alphabet and a 36-letter-and-number alphabet.

$$M=\frac{\log(2.23992x10^{12})}{\log 26}=8.72 \text{ (for 6-month lifetime)}$$

$$M=\frac{\log(4.4676 \text{ x } 10^{12})}{\log 26} = 8.94 \text{ (for 12-month lifetime)}$$

$$M=\frac{\log(2.23992 \text{ x } 10^{12})}{\log 36}=793 \text{ (for 6-month lifetime)}$$

$$M=\frac{\log(4.4676X\ 10^{12})}{\log 36}=8.13 \text{ (for 12-month lifetime)}$$

Table 1 presents the results.

Table 1

Length of Password

MAXIMUM LIFETIME (months)	26-character alphabet	36character alphabet
6	9 (rounded up from 8.72)	8 (rounded up from 7.93)
12	9 (rounded up from 8,94)	8 (rounded down from 8.13)

7. Passphrases

A "passphrase" is a concatenation of words drawn from a dictionary. The dictionary is merely the collection of symbols making up the "alphabet" from which the password is generated. As an example, suppose the passphrase is made up of words drawn from a dictionary of 4-, 5- and 6-letter words. There are approximately 3,780 4-letter words, 7,500 5-letter words and 12,000 6-letter words in English. The "alphabet size" for generating passphrases is approximately 23,300.

We can compute how many words, drawn at random from the dictionary of 23,300 words, are needed to produce a passphrase that will be resistant to exhaustive attack with the probability of 1 x 10-6.

We have to solve for S as before, and from that, solve for M, the length of the password (i.e., number of alphabet symbols or words).

For L= 12 months, 5=4.4676*1012, Log S=

12.6500

For L=6 months, 5=2.2399*1012, Log 5=12.3502

Log 23300=4.3669

Using equation (4) we obtain:

12 6500
For L=12 months M= 4.3669 =3 (rounded from 2.89)

12 3502

For L=6 months 4.'3669 =3 (rounded from 2.82)

Thus, for the passphrase algorithm described, namely selection at random from a dictionary of 23,300 words, only 3 words are needed in a passphrase to obtain the desired resistance to exhaustive enumeration. In using the algorithm, each word of the phrase is drawn independently from the dictionary. This may result in a word appearing more than once in the passphrase.

APPENDIX E.4

PROTECTION BASIS FOR PASSWORDS

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 [3] 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 [3], 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 the 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 processes only to the extent that passwords are protected. For example, a user with a 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 [5].

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.

APPENDIX E.5

FEATURES FOR USE IN VERY SENSITIVE APPLICATIONS

The following features can be used to enhance the security provided by a password system. Because they are somewhat "user-unfriendly," they are recommended for environments only when there is a high threat of password compromise.

1. One-Time Passwords

One-time passwords (i.e., those that are changed after each use) are useful when the password is not adequately protected from compromise during login (e.g., the communication line is suspected of being tapped). The difficult part of using one-time passwords is in the distribution of the new passwords. If a one-time password is changed often because of frequent use, the distribution of new one-time passwords becomes a significant point of vulnerability. There are products on the market that generate such passwords through a cryptographic protocol between the destination host and a hand-held device the user can carry.

2. Failed Login Attempt Limits

In some instances, it may be desirable to count the number of unsuccessful login attempts for each user ID and to base password expiration and user ID locking on the actual number of failed attempts. (Changing a password would reset the count for that user ID to zero.) For example, the password could be identified as expired after 100 failed login attempts, and the user ID locked after 500.

APPENDIX E.6

ON THE PROBABILITY OF GUESSING A PASSWORD

Appendix B.3 discusses the techniques for finding a password length that will resist exhaustive enumeration over the lifetime of the password with a given probability. This appendix derives the probability of guessing a password during its lifetime.

As in Appendix B.3, we use the parameters:

L = password lifetime R = guess rate S = size of the password space
P=probability of guessing a password during its lifetime.

The total number of guesses, (G), that can be made during a password's lifetime is:

$$G=R \text{ X } L \qquad (I)$$

At this point, we need to consider the relation of the size of the password space, S, to G. Clearly, if S is so small that one could try all possible passwords before the lifetime of the password expires, the probability of guessing the password is 1. As a result, we consider only cases where S is greater than G.

The probability question then is, "For the case where S is greater than G, what is the probability that in G guesses the password will be guessed?". This is the same as asking the question, "What is the probability that in the lifetime of the password, it will be guessed?". The probability sought is:

How many ways one can make G guesses
(of S objects)

that include the password
How many different ways one can make
G guesses of S objects

Note that the probability that is appealed to is of the simplest form. It is derived from the definition of probability that the probability of an event is given by the number of ways the event can happen divided by the number of ways an event can happen or fail.

We first observe that the total number of ways one can make G guesses of S things is given by sCg (the combinatorial notation that means the number of combinations of "s" things taken "g" at a time). (Lower case letters are used with the combinatorial notation in order to make the expressions more readable.) This is determined by:

$$\frac{s!}{g!(s-g)!}$$

Thus, if S=A,B,C,D,E, one could make 3 guesses in 5C3 different ways, 5*4*3*2*1/3*2*1*2*1 = 10.

(Enumerating, they are ABC,ABD,ABE,ACD,ACE,ADE,BCD,BCE,BDE,CDE.)

The problem of finding the number of guesses of this total that include a specific password, e.g., an "A" is addressed by considering a reduced set without the specific password and asking how many ways one can make G guesses with the reduced set. Then, the total number of ways to make G guesses that include the specified password is the difference between the two values. This is given by:

$$sCg-(s-1)Cg \qquad (2)$$

That is, remove the designated password from the set S, compute the number of ways of making G guesses without the password, then consider the difference between the two values.

If we ask in our example how many ways to make 3 guesses that do NOT include a particular password from the set of 5 (say an "A"), this is given by:

4C3=4*3*2*1/3*2*1*1 =4

Enumerating for the specific case of an "A", they are BCD,BCE,BDE,CDE.

The number of ways to make 3 guesses that include the designated element is 10-4=6. Thus, the probability of guessing a designated password in 3 guesses is 6/10 or .6.

Simplification

It is indeed fortuitous that there is a theorem in any number of books on Probability Theory that states:

$$nCr=(n-1)C(r-1)+(n-1)Cr \qquad (3)$$

This may also be expressed as:

$$nCr-(n- 1)Cr=(n- 1)C(r-1) \qquad (4)$$

Substituting s for n and g for r we obtain the expression:

$$(s- 1)C(g-1) \qquad (5)$$

for the number of ways of making G guesses that include a specific password. Then, the probability that a given password will be guessed during the lifetime of that password is given by:

$$P-\frac{(s-1)C(g-1)}{sCg} \qquad (6)$$

Evaluating this expression gives:

$$P=\frac{\frac{(s-1)!}{(g-1)!((s-1)-(g-1))!}}{\frac{s!}{g!(s-g)!}}=\frac{\frac{(s-1)!}{(g-1)!(s-g)!}}{\frac{s!}{g!(s-g)!}}=\frac{g!(s-1)!}{(g-1)s!}=\frac{g}{s} \qquad (7)$$

This derivation of the probability of guessing a password during its lifetime, i.e.,

$$P=\frac{G}{S} \qquad (8)$$

is important in that it allows us to derive the size of the password space

$$S=\frac{G}{P} \qquad (9)$$

given an acceptable probability of not guessing the password during its lifetime.

APPENDIX E.7

REFERENCES

[1] Brown, R. L. Computer System Access Control Using Passwords, final draft, Aerospace Corporation, 16 January 1984.

[2] DoD Computer Security Center. Department of Defense Trusted Computer System Evaluation Criteria, CSC-STD-001-83, 15 August 1983.

[3] DoD Directive 5200.28, Security Requirements for Automatic Data Processing (ADP) Systems, revised April 1978.

[4] Downey, P. J. Multics Security Evaluation: Password and File Encryption Techniques, ESD-TR-74- 193, Vol. III, AD-A045279, AFSC Electronic Systems Division, Hanscom AFB, Mass., June 1977.

[5] Executive Order 12356, National Security Information, 6 April 1982.

[6] Gasser, M. A Random Word Generator for Pronounceable Passwords, MTR-3006, ESD-TR-75-97, ADA017676, MITRE Corp., Bedford, Mass., November 1975.

[7] Wood, H. M. The Use of Passwords for Controlled Access to Computer Resources, NBS Special Publication 500-9, U.S. Department of Commerce, National Bureau of Standards, May 1977.

[8] National Bureau of Standards. Federal Information Processing Standards Publication 46, Data Encryption Standard, 15 January 1977.

[9] National Bureau of Standards. Federal Information Processing Standards Publication 81, DES Modes of Operation, 2 December 1980.