lowing the compare, we note that if they are not equal (i.e. OK button not hit) then it goes off somewhere. The next instruction must be the one that executes after the user hits the OK button. So set your breakpoint at the line that reads CLR.W FDEC(A6) which is at address 5B4FA0 (this will vary) - and in fact you can see the asterisk in the listing denoting that I have done just that. Now exit, enter your name and company and serial number (keep typing anything until the OK button lights up) and hit OK. Now TMON breaks in again at the breakpoint. Now we can begin the crack.

Determining how to implement the crack.

Before you continue, think about what the program must do at this point if it wants to validate your serial number (here it helps to have read Inside Mac on dialogs). First the program must obtain a pointer to the dialog item #5 (the serial number field) and then it must obtain a pointer to the text contained in that item. Knowing this, you can just scan down until you see a GetDItem trap followed closely by a GetIText trap. After this last trap, the program can do its validation. Here is that piece of code:

```
MOVE.L  A3,-(A7)
_GetIText
MOVE.L  A3,-(A7)
JSR
  ^$005B5670

TST.L
  D0
ADDQ.L  #4,A7
BMI
  ^$005B50C8

CMPI.L  #$00000005,D0
BGT
  ^$005B50C8

ADD.L
  D0,D0
MOVE.W  ^$005B500A(D0.L),D0

JMP
  ^$005B5008(D0.W)
```

We can note that A3 is the pointer that will point to the text after the trap. Once A3 has the text, a subroutine is called and D0 is tested. At this point, we cannot be sure whether the branch executes if the serial passed or failed, so we had better take a quick look at the code at address 5B50C8. I am not going to show it here, but that code does some crap then calles ParamText and then a Dialog call so it is probably safe to guess that the branch above jumps to the error code.

With this assumption in mind, what can we do about it? An initial guess would be to just make that BMI either not execute or even better, make the BMI branch down to the ADD.L D0,D0. Unfortunately, if you look at the last two lines, you can see that D0 not only determines whether the code branches to

the error routine, but is then used for a JMP instruction so we had better take care of D0. Let's take a quick look at that JSR up a few lines that sets D0 in the first place and remember, we are trying to figure out what D0 should be set to. Also remember that the branch is a BMI meaning that the error occurs if the high bit of D0 is set.

```
004B1508:'CODE'®$0003ƒ$04C8+$096C LINK.W  A6,#$FF00
004B150C:'CODE'®$0003ƒ$04C8+$0970 MOVEM.L A3/A4,-(A7)
004B1510:'CODE'®$0003ƒ$04C8+$0974 LEA
  `FF00(A6),A4
004B1514:'CODE'®$0003ƒ$04C8+$0978 MOVEA.L $0008(A6),A3
004B1518:'CODE'®$0003ƒ$04C8+$097C MOVEQ
  #$00,D0
004B151A:'CODE'®$0003ƒ$04C8+$097E MOVE.B  (A3),D0
004B151C:'CODE'®$0003ƒ$04C8+$0980 MOVEQ
  #$06,D1
004B151E:'CODE'®$0003ƒ$04C8+$0982 CMP.L
  D0,D1
004B1520:'CODE'®$0003ƒ$04C8+$0984 BLE.S
  ^$004B1526
      ;'CODE'®$0003ƒ$04C8+$98A
004B1522:'CODE'®$0003ƒ$04C8+$0986 MOVEQ
  #`FF,D0
004B1524:'CODE'®$0003ƒ$04C8+$0988 BRA.S
  ^$004B1592
      ;'CODE'®$0003ƒ$04C8+$9F6
004B1526:'CODE'®$0003ƒ$04C8+$098A MOVEQ
  #$00,D0
004B1528:'CODE'®$0003ƒ$04C8+$098C MOVE.B  (A3),D0
004B152A:'CODE'®$0003ƒ$04C8+$098E MOVEQ
  #$28,D1
     ;'('
004B152C:'CODE'®$0003ƒ$04C8+$0990 CMP.L
  D0,D1
004B152E:'CODE'®$0003ƒ$04C8+$0992 BGE.S
  ^$004B1534
      ;'CODE'®$0003ƒ$04C8+$998
004B1530:'CODE'®$0003ƒ$04C8+$0994 MOVEQ
  #`FF,D0
004B1532:'CODE'®$0003ƒ$04C8+$0996 BRA.S
  ^$004B1592
      ;'CODE'®$0003ƒ$04C8+$9F6
004B1534:'CODE'®$0003ƒ$04C8+$0998 MOVE.L  A4,-(A7)
004B1536:'CODE'®$0003ƒ$04C8+$099A PEA
  $3802
         ;$000037D8+$2A
004B153A:'CODE'®$0003ƒ$04C8+$099E JSR
  $1702(A5)
004B153E:'CODE'®$0003ƒ$04C8+$09A2 MOVE.L  A4,-(A7)
004B1540:'CODE'®$0003ƒ$04C8+$09A4 JSR
  $0532(A5)
004B1544:'CODE'®$0003ƒ$04C8+$09A8 MOVE.L  A3,-(A7)
004B1546:'CODE'®$0003ƒ$04C8+$09AA MOVE.L  A4,-(A7)
004B1548:'CODE'®$0003ƒ$04C8+$09AC JSR
  $0392(A5)
004B154C:'CODE'®$0003ƒ$04C8+$09B0 TST.L
```

```
  D0
004B154E:'CODE'®$0003ƒ$04C8+$09B2 LEA
  $0014(A7),A7
004B1552:'CODE'®$0003ƒ$04C8+$09B6 BEQ.S
  ^$004B1558
      ;'CODE'®$0003ƒ$04C8+$9BC
004B1554:'CODE'®$0003ƒ$04C8+$09B8 MOVEQ
  #$05,D0
004B1556:'CODE'®$0003ƒ$04C8+$09BA BRA.S
  ^$004B1592
      ;'CODE'®$0003ƒ$04C8+$9F6
004B1558:'CODE'®$0003ƒ$04C8+$09BC MOVE.B  $0001(A3),D0
004B155C:'CODE'®$0003ƒ$04C8+$09C0 SUBI.B  #$30,D0
004B1560:'CODE'®$0003ƒ$04C8+$09C4 BCS.S
  ^$004B1590
      ;'CODE'®$0003ƒ$04C8+$9F4
004B1562:'CODE'®$0003ƒ$04C8+$09C6 CMPI.B  #$02,D0
004B1566:'CODE'®$0003ƒ$04C8+$09CA BHI.S
  ^$004B1590
      ;'CODE'®$0003ƒ$04C8+$9F4
004B1568:'CODE'®$0003ƒ$04C8+$09CC MOVEQ
  #$00,D1
004B156A:'CODE'®$0003ƒ$04C8+$09CE MOVE.B  D0,D1
004B156C:'CODE'®$0003ƒ$04C8+$09D0 ADD.W
  D1,D1
004B156E:'CODE'®$0003ƒ$04C8+$09D2 MOVE.W  ^$004B1576(D1.W),D1
     ;'CODE'®$0003ƒ$04C8+$9DA
004B1572:'CODE'®$0003ƒ$04C8+$09D6 JMP
  ^$004B1574(D1.W)
     ;'CODE'®$0003ƒ$04C8+$9D8
```

There are no traps here to quickly tell us what is happening, but we can quickly look at the lines that affect D0. Basically, there are a bunch of interspersed MOVEQ instructions putting various values into D0. One of the values is $FF which (since the high bit of $FF is set - in fact, all the bits of $FF are set) must trigger the error in the previous procedure. Other values include 5 and 0. Right now, that is enough information to proceed with the previous procedure - if we need more in depth info, we can always come back. So we have the following code again:

```
MOVE.L  A3,-(A7)
JSR
  ^$005B5670

TST.L
  D0
ADDQ.L  #4,A7
BMI
  ^$005B50C8

CMPI.L  #$00000005,D0
BGT
  ^$005B50C8

ADD.L
  D0,D0
```

```
MOVE.W  ^$005B500A(D0.L),D0

JMP
  ^$005B5008(D0.W)
```

Once again, we have an initial BMI which tells us that $FF won't work for D0. We also have BGT after comparing D0 with 5 which branches to the error - so D0 must be between 0 and 5 (the other values we noted from the subroutine above). At this point, I would (and did) simply try inserting values into D0. I started with 5 and the program went into Demo mode - strike one. Next I tried 1 and some other error occured. Finally, I tried 0 and the program continued flawlessly.

So you are asking, how exactly might you go about inserting these values into D0? Consider: once D0 is set to the proper value, the two branches become meaningless since they would not execute anyways (they only execute if there is an error). This little tidbit tells us that we can safely overwrite these instructions with anything we like. So we have several free bytes to put our own code into (don't panic yet - this is pretty straightforward) and all our code has to do is set D0 to 0 then proceed. One quick note: Never Never Ever modify code that affects the stack. If you do, you can easily cause system errors later on down the road. In the above code, this translates into not changing the ADDQ.L #4,A7 (A7 is the stack pointer, remember?). So what is the easiest way to put 0 into D0? Use a MOVEQ instruction. This is particularly nice because you probably do not know the machine hex code for instructions (like me). But that subroutine we looked at before is chalk full of MOVEQ instructions. If you look, a MOVEQ 0 #0,D0 translates into 70 00. So far so good except that the stupid BMI is one of those 4 byte branches. So we still have two bytes left that will be garbage since we just changed the first two. This is an excellent candidate for a NOP instruction - a two byte instruction that does absolutely nothing. The code for this (from the Cracker's Guide Part 1) is 4E 71.

So, open a dump window to the PC and find the BMI (I think it is 68 00 00 D4 or something like that). Change the four values to 70 00 4E 71 and now the program loads D0 with the correct value and proceeds as if nothing had happened. Now you have the crack, but you want to make a cracked / un-serialized copy right? So, unstuff a fresh copy of the application, open it in Resedit, and open the proper CODE resource. To find the ID #, look back at the TMON listing. It says CODE 0003 plus some benutia about the File reference number and then +nnnn where nnnn is the offset from the beginning of the Code resource. There is all you need. Open CODE ID 3 and jump down to line 2E8 (since 2EE is our byte) and change the 68 00 00 D4 to 70 00 4E 71. Now run it and enter anything you like for the serial number.

QuickFormat 7.01

[due to burn-out, the final sections have not been written up]

```
     33E:                                QUAL    CHECKFOR ; b# =508  s#3
=proc196

                                ;-refs -  3/INITPROG
```

```
     33E: 4E56 FFE4      'NV..'  CHECKFOR LINK    A6,#-$1C
     342: 48E7 0108      'H...'           MOVEM.L D7/A4,-(A7)
     346: 594F           'YO'             SUBQ    #4,A7
     348: 2F3C 6465 6D6F '/<demo'         PUSH.L  #'demo'
     34E: 3F3C 0080      '?<..'           PUSH    #128
     352: A81F           '..'             _Get1Resource ; (theType:ResType;
ID:INTEGER):Handle
     354: 285F           '(_'             POP.L   A4
     356: 200C           ' .'             MOVE.L  A4,D0
     358: 6656           30003B0          BNE.S   lih_2
     35A: 594F           'YO'             SUBQ    #4,A7
     35C: 7004           'p.'             MOVEQ   #4,D0
     35E: 2F00           '/.'             PUSH.L  D0
     360: 4EAD 0082      10005EA          JSR     NewHandle(A5)
     364: 285F           '(_'             POP.L   A4
     366: 2F0C           '/.'             PUSH.L  A4
     368: 4EAD 0092      1000614          JSR     HLock(A5)
     36C: 2054           ' T'             MOVEA.L (A4),A0
     36E: 20BC 000F 423F ' ...B?'         MOVE.L  #$F423F,(A0)
     374: 2F0C           '/.'             PUSH.L  A4
     376: 2F3C 6465 6D6F '/<demo'         PUSH.L  #'demo'
     37C: 3F3C 0080      '?<..'           PUSH    #128
     380: 487A 007C      30003FE          PEA     data209     ; len= 2
     384: A9AB           '..'           _AddResource ; (theResource:Handle;
theType:ResType; theID:INTEGER; name:Str255)
     386: 554F           'UO'             SUBQ    #2,A7
     388: A9AF           '..'             _ResError ; :OSErr
     38A: 4A5F           'J_'             TST     (A7)+
     38C: 6714           30003A2          BEQ.S   lih_1
     38E: 3F3C 008B      '?<..'           PUSH    #139
     392: 1F3C 0001      '.<..'           PUSH.B  #1
     396: 4EAD 0462      2000B7C          JSR     DOSTANDA(A5)
     39A: 554F           'UO'             SUBQ    #2,A7
     39C: A9AF           '..'             _ResError ; :OSErr
     39E: 4EAD 0452      20009FE          JSR     DOERROR(A5)
     3A2: 2F0C           '/.'    lih_1    PUSH.L  A4
     3A4: A9AA           '..'             _ChangedResource ;
(theResource:Handle)
     3A6: 2F0C           '/.'             PUSH.L  A4
     3A8: A9B0           '..'             _WriteResource ;
(theResource:Handle)
     3AA: 2F0C           '/.'             PUSH.L  A4
     3AC: 4EAD 009A      100061E          JSR     HUnLock(A5)
     3B0: 2F0C           '/.'    lih_2    PUSH.L  A4
     3B2: 4EAD 0092      1000614          JSR     HLock(A5)
     3B6: 2E3C 176F 7C4E '.<.o|N'         MOVE.L  #$176F7C4E,D7
     3BC: 2054           ' T'             MOVEA.L (A4),A0
     3BE: BE90           '..'             CMP.L   (A0),D7
     3C0: 6606           30003C8          BNE.S   lih_3
     3C2: 422D FDE2        -$21E          CLR.B   glob73(A5)
     3C6: 6020           30003E8          BRA.S   lih_4
     3C8: 554F           'UO'    lih_3    SUBQ    #2,A7
     3CA: 2F07           '/.'             PUSH.L  D7
     3CC: 4EBA FE68      3000236          JSR     DODEMODI
     3D0: 1B5F FDE2        -$21E          POP.B   glob73(A5)
```

```
     3D4: 102D FDE2        -$21E          MOVE.B  glob73(A5),D0
     3D8: 5300            'S.'            SUBQ.B  #1,D0
     3DA: 670C            30003E8         BEQ.S   lih_4
     3DC: 2054            ' T'            MOVEA.L (A4),A0
     3DE: 2087            ' .'            MOVE.L  D7,(A0)
     3E0: 2F0C            '/.'            PUSH.L  A4
     3E2: A9AA            '..'            _ChangedResource ;
(theResource:Handle)
     3E4: 2F0C            '/.'            PUSH.L  A4
     3E6: A9B0            '..'            _WriteResource ;
(theResource:Handle)
     3E8: 2F0C            '/.'    lih_4   PUSH.L  A4
     3EA: 4EAD 009A       100061E         JSR     HUnLock(A5)
     3EE: 4CDF 1080       'L...'          MOVEM.L (A7)+,D7/A4
     3F2: 4E5E            'N^'            UNLK    A6
     3F4: 4E75            'Nu'            RTS

Finder 7 Menus


     458:                                 QUAL    GETPASSW ; b# =31  s#1
=proc14

                              vap_1     VEQU  -288
                              vap_2     VEQU  -280
                              vap_3     VEQU  -276
                              vap_4     VEQU  -274
                              vap_5     VEQU  -272
     458:                               VEND

                            ;-refs - DOCOMMAN

     458: 4E56 FED8       'NV..'  GETPASSW LINK    A6,#-$128
     45C: 48E7 0018       'H...'          MOVEM.L A3-A4,-(A7)
     460: 4A2D FEFE        -$102          TST.B   glob59(A5)
     464: 670C            1000472         BEQ.S   lap_1
     466: 487A 01B4       100061C         PEA     data23      ; 'Password has
already
     46A: 4EBA 139E       100180A         JSR     OUTPUTTE
     46E: 6000 00A2       1000512         BRA     lap_5
     472: 3F2D FEBA        -$146 lap_1    PUSH    glob28(A5)
     476: A998            '..'            _UseResFile ; (frefNum:RefNum)
     478: 594F            'YO'            SUBQ    #4,A7
     47A: 3F3C 0101       '?<..'          PUSH    #257
     47E: 42A7            'B.'            CLR.L   -(A7)
     480: 70FF            'p.'            MOVEQ   #-1,D0
     482: 2F00            '/.'            PUSH.L  D0
     484: A97C            '.|'            _GetNewDialog ; (DlgID:INTEGER;
wStorage:Ptr; behind:WindowPtr):DialogPtr
     486: 285F            '(_'            POP.L   A4
     488: 2F0C            '/.'            PUSH.L  A4
     48A: 3F3C 0002       '?<..'          PUSH    #2
     48E: 486E FEEC       200FEEC         PEA     vap_3(A6)
     492: 486E FEE8       200FEE8         PEA     vap_2(A6)
     496: 486E FEE0       200FEE0         PEA     vap_1(A6)
```

```
     49A: A98D           '..'           GetDItem ; (dlg:DialogPtr;
itemNo:INTEGER; VAR kind:INTEGER; VAR item:Handle; VAR box:Rect)
     49C: 42A7           'B.'    lap_2    CLR.L   -(A7)
     49E: 486E FEEE      200FEEE          PEA     vap_4(A6)
     4A2: A991           '..'             _ModalDialog ; (filterProc:ProcPtr;
VAR itemHit:INTEGER)
     4A4: 0C6E 0001 FEEE 200FEEE          CMPI    #1,vap_4(A6)
     4AA: 66F0           100049C          BNE     lap_2
     4AC: 2F2E FEE8      200FEE8          PUSH.L  vap_2(A6)
     4B0: 486E FEF0      200FEF0          PEA     vap_5(A6)
     4B4: A990           '..'             _GetIText ; (item:Handle; VAR
text:Str255)
     4B6: 487A 0152      100060A          PEA     data22      ;
'cc5187efH28b911af'
     4BA: 486E FEF0      200FEF0          PEA     vap_5(A6)
     4BE: 4EBA FC4A      100010A          JSR     proc6
     4C2: 6642           1000506          BNE.S   lap_3
     4C4: 594F           'YO'             SUBQ    #4,A7
     4C6: 486E FEF0      200FEF0          PEA     vap_5(A6)
     4CA: A906           '..'             _NewString ;
(theString:Str255):StringHandle
     4CC: 265F           '&_'             POP.L   A3
     4CE: 2F0B           '/.'             PUSH.L  A3
     4D0: 2F3C 5354 5220 '/<STR '         PUSH.L  #'STR '
     4D6: 3F3C 0080      '?<..'           PUSH    #128
     4DA: 487A 012C      1000608          PEA     data21      ; len= 2
     4DE: A9AB           '..'             _AddResource ; (theResource:Handle;
theType:ResType; theID:INTEGER; name:Str255)
     4E0: 3F2D FEBA        -$146          PUSH    glob28(A5)
     4E4: A999           '..'             _UpdateResFile ; (frefNum:RefNum)
     4E6: 1B7C 0001 FEFE   -$102          MOVE.B  #1,glob59(A5)
     4EC: 487A 00C0      10005AE          PEA     data20      ; 'Thanks for
registeri
     4F0: 4EBA 1318      100180A          JSR     OUTPUTTE
     4F4: 4A2D FEFE        -$102          TST.B   glob59(A5)
     4F8: 6714           100050E          BEQ.S   lap_4
     4FA: 2F2D FEB0        -$150          PUSH.L  glob25(A5)
     4FE: 487A 0086      1000586          PEA     data19      ; ''Thank you
for payin
     502: A91A           '..'             _SetWTitle ; (theWindow:WindowPtr;
title:Str255)
     504: 6008           100050E          BRA.S   lap_4
     506: 487A 001A      1000522 lap_3    PEA     data18      ; 'For only $10,
you ca
     50A: 4EBA 12FE      100180A          JSR     OUTPUTTE
     50E: 2F0C           '/.'    lap_4    PUSH.L  A4
     510: A982           '..'             _CloseDialog ; (dlg:DialogPtr)
     512: 4CDF 1800      'L...'  lap_5    MOVEM.L (A7)+,A3-A4
     516: 4E5E           'N^'             UNLK    A6
     518: 4E75           'Nu'             RTS
```