```
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL                 000000000          GGGGGGGGGGGG     IIIIIIIII     NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNN           NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNNNNN        NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG                    III      NNN    NNN    NNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG    GGGGGGGGG       III      NNN       NNNNNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLL             000           000   GGG          GGG       III      NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
LLLLLLLLLLLLLLL     000000000          GGGGGGGGG        IIIIIIIII     NNN           NNN
```

FILEID**BREAKIN

```
BBBBBBBB    RRRRRRRR    EEEEEEEEEE    AAAAAA    KK      KK    IIIIII    NN      NN
BBBBBBBB    RRRRRRRR    EEEEEEEEEE    AAAAAA    KK      KK    IIIIII    NN      NN
BB      BB  RR      RR  EE           AA      AA  KK      KK      II      NN      NN
BB      BB  RR      RR  EE           AA      AA  KK      KK      II      NN      NN
BB      BB  RR      RR  EE           AA      AA  KK    KK        II      NNNN    NN
BB      BB  RR      RR  EE           AA      AA  KK    KK        II      NNNN    NN
BBBBBBBB    RRRRRRRR    EEEEEEEE     AA      AA  KKKKKK          II      NN  NN  NN
BBBBBBBB    RRRRRRRR    EEEEEEEE     AA      AA  KKKKKK          II      NN  NN  NN
BB      BB  RR  RR      EE           AAAAAAAAAA  KK    KK        II      NN    NNNN
BB      BB  RR  RR      EE           AAAAAAAAAA  KK    KK        II      NN    NNNN
BB      BB  RR    RR    EE           AA      AA  KK      KK      II      NN      NN      ....
BB      BB  RR    RR    EE           AA      AA  KK      KK      II      NN      NN      ....
BBBBBBBB    RR      RR  EEEEEEEEEE   AA      AA  KK      KK    IIIIII    NN      NN      ....
BBBBBBBB    RR      RR  EEEEEEEEEE   AA      AA  KK      KK    IIIIII    NN      NN      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE breakin (IDENT = 'V04-000',
;     2          0002  0                  ADDRESSING_MODE(EXTERNAL = GENERAL)) =
;     3          0003  1 BEGIN
;     4          0004  1
;     5          0005  1 !
;     6          0006  1 !*****************************************************************************
;     7          0007  1 !*                                                                           *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                     *
;    11          0011  1 !*                                                                           *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                             *
;    18          0018  1 !*                                                                           *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                             *
;    22          0022  1 !*                                                                           *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*****************************************************************************
;    28          0028  1
;    29          0029  1 !++
;    30          0030  1 ! FACILITY:  Login
;    31          0031  1 !
;    32          0032  1 ! ABSTRACT:
;    33          0033  1 !
;    34          0034  1 !       This module contains all the routines to scan and manipulate the
;    35          0035  1 !       Compound Intrusion Analysis blocks, add new entries to the CIA
;    36          0036  1 !       queues, locate intruders, and remove suspects.
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:
;    39          0039  1 !
;    40          0040  1 !       VAX/VMS operating system.
;    41          0041  1 !
;    42          0042  1 ! AUTHOR:                                   Gerry Smith             12-July-1983
;    43          0043  1 !
;    44          0044  1 ! Modified by:
;    45          0045  1 !
;    46          0046  1 !       V03-006 ACG0436         Andrew C. Goldstein,    23-Jul-1984  17:25
;    47          0047  1 !               Add support for LGI_BRK_TERM, put result in global cell
;    48          0048  1 !
;    49          0049  1 !       V03-005 MHB0131         Mark Bramhall            5-Apr-1984
;    50          0050  1 !               Change to new TERMINAL_DEVICE flag.
;    51          0051  1 !
;    52          0052  1 !       V03-004 ACG0390         Andrew C. Goldstein,    18-Jan-1984  11:33
;    53          0053  1 !               Fix arg list mismatches (remove unused username)
;    54          0054  1 !
;    55          0055  1 !       V03-003 ACG0376         Andrew C. Goldstein,    22-Nov-1983  11:02
;    56          0056  1 !               Redesign match algorithm to reduce service denial
;    57          0057  1 !               problems, fix expiration of old entries.
```

```
;    58          0058  1 !
;    59          0059  1 !       V03-002 GAS0189                 Gerry Smith             22-Sep-1983
;    60          0060  1 !               If a terminal is specified, see if there's another device,
;    61          0061  1 !               the actual physical device, associated with the terminal.
;    62          0062  1 !
;    63          0063  1 !       V03-001 GAS0185                 Gerry Smith             16-Sep-1983
;    64          0064  1 !               Add randomness to the hide time, so that the actual hide time
;    65          0065  1 !               is between 50% and 150% of that specified by the SYSGEN
;    66          0066  1 !               parameter.
;    67          0067  1 !
;    68          0068  1 !--
;    69          0069  1
;    70          0070  1 !
;    71          0071  1 ! Include files
;    72          0072  1 !
;    73          0073  1 LIBRARY 'SYS$LIBRARY:LIB';              ! VAX/VMS system definitions
;    74          0074  1
;    75          0075  1 !
;    76          0076  1 ! Declare the linkages to allocate and deallocate nonpaged pool, as well
;    77          0077  1 ! as grab the CIA mutex.
;    78          0078  1 !
;    79          0079  1 LINKAGE
;    80          0080  1         CVTDEV = JSB (REGISTER = 0,             ! Length of output buffer,
;    81          0081  1                       REGISTER = 1,             ! Address of output buffer
;    82          0082  1                       REGISTER = 4,             ! Format of device name
;    83          0083  1                       REGISTER = 5;             ! Address of UCB
;    84          0084  1                       REGISTER = 1),            ! Length of final name
;    85          0085  1         ALLO   = JSB (REGISTER = 1;             ! R1 = size (on input)
;    86          0086  1                       REGISTER = 1,             ! R1 = size of block
;    87          0087  1                       REGISTER = 2):            ! R2 = address of block
;    88          0088  1                       NOPRESERVE (3,4,5),       ! R3, R4, R5 destroyed
;    89          0089  1         DEALLO = JSB (REGISTER = 0):            ! R0 = address of block
;    90          0090  1                       NOPRESERVE (1,2,3,4,5),   ! R1-R5 destroyed
;    91          0091  1         LOCK   = JSB (REGISTER = 0,             ! R0 = address of mutex
;    92          0092  1                       REGISTER = 4);            ! R4 = PCB address
;    93          0093  1
;    94          0094  1
;    95          0095  1 !
;    96          0096  1 ! Macro to set the processor interrupt priority level register
;    97          0097  1 !
;    98          0098  1 BUILTIN
;    99          0099  1         MTPR;
;   100          0100  1
;   101          0101  1 MACRO                                           ! set processor IPL
;   102          0102  1     SET_IPL (LEVEL) = MTPR (%REF (LEVEL), PR$_IPL)%;
;   103          0103  1
```

```
;  105          0104  1 !
;  106          0105  1 ! Table of contents
;  107          0106  1 !
;  108          0107  1 FORWARD ROUTINE
;  109          0108  1     cia_scan,                       ! Set up for elevated-IPL scans
;  110          0109  1     check_intruder,                 ! Check intruder block
;  111          0110  1     check_suspect,                  ! Check suspect block
;  112          0111  1     add_suspect;                    ! Add a suspect block
;  113          0112  1
;  114          0113  1 !
;  115          0114  1 ! External routines
;  116          0115  1 !
;  117          0116  1 EXTERNAL ROUTINE
;  118          0117  1     ioc$cvt_devnam : CVTDEV,        ! Construct device name from UCB
;  119          0118  1     sch$lockw : LOCK,               ! Lock CIA mutex
;  120          0119  1     sch$unlock : LOCK,              ! Unlock CIA mutex
;  121          0120  1     exe$alononpaged : ALLO,         ! Allocate non-paged pool
;  122          0121  1     exe$deanonpaged : DEALLO;       ! Deallocate non-paged pool
;  123          0122  1
;  124          0123  1 !
;  125          0124  1 ! Declare system areas
;  126          0125  1 !
;  127          0126  1 EXTERNAL
;  128          0127  1     break_attempt : BYTE,
;  129          0128  1     terminal_device : BYTE,
;  130          0129  1     phy_term_name : VECTOR,
;  131          0130  1     uaf_record : REF $BBLOCK,
;  132          0131  1     org_username,
;  133          0132  1     ctl$gl_pcb : REF $BBLOCK,
;  134          0133  1     ctl$t_nodename : VECTOR[,BYTE],
;  135          0134  1     ctl$t_remoteid : VECTOR[,BYTE],
;  136          0135  1     cia$gl_mutex,
;  137          0136  1     cia$gq_intruder : $BBLOCK,
;  138          0137  1     sys$gb_brk_lim : BYTE,
;  139          0138  1     sys$gl_brk_tmo,
;  140          0139  1     sys$gl_hid_tim,
;  141          0140  1     exe$gl_dynamic_flags : BITVECTOR;
;  142          0141  1
;  143          0142  1 EXTERNAL LITERAL
;  144          0143  1     exe$v_brk_term : UNSIGNED (6);
```

```
;  146          0144  1 GLOBAL ROUTINE cia_scan (intruder) =
;  147          0145  2 BEGIN
;  148          0146  2
;  149          0147  2 !+++
;  150          0148  2 !
;  151          0149  2 ! This is the routine that is called to start either a suspect
;  152          0150  2 ! or intruder scan.
;  153          0151  2 !
;  154          0152  2 ! Inputs:
;  155          0153  2 !       intruder - indicator for which type of scan to perform
;  156          0154  2 !                  1 ==> intruder list
;  157          0155  2 !                  0 ==> suspect list
;  158          0156  2 !
;  159          0157  2 ! Outputs:
;  160          0158  2 !       None.
;  161          0159  2 !
;  162          0160  2 ! Return status: (low bit)
;  163          0161  2 !       0 - intruder (either new, or in evasive action)
;  164          0162  2 !       1 - just another suspect, or an expired intruder
;  165          0163  2 !
;  166          0164  2 !---
;  167          0165  2
;  168          0166  2 BUILTIN
;  169          0167  2     emul,
;  170          0168  2     addm,
;  171          0169  2     cmpm,
;  172          0170  2     remque;
;  173          0171  2
;  174          0172  2 LABEL
;  175          0173  2     cia_search,
;  176          0174  2     cia_check;
;  177          0175  2
;  178          0176  2 LOCAL
;  179          0177  2     type : BYTE,
;  180          0178  2     data : VECTOR [cia$s_data, BYTE],
;  181          0179  2     time : VECTOR[2],
;  182          0180  2     term_length,
;  183          0181  2     delta,
;  184          0182  2     dummy,
;  185          0183  2     status,
;  186          0184  2     next,
;  187          0185  2     cia : REF $BBLOCK;
;  188          0186  2
;  189          0187  2 !
;  190          0188  2 ! Get the current time.
;  191          0189  2 !
;  192          0190  2 $gettim (timadr = time);
;  193          0191  2
;  194          0192  2 !
;  195          0193  2 ! Get the data string to match on, which is meant to correspond to the
;  196          0194  2 ! source of the login. It is, in order of preference:
;  197          0195  2 !       node name and remote ID
;  198          0196  2 !       terminal if a real terminal plus username if real user
;  199          0197  2 !       original username (presumably creator of this process)
;  200          0198  2 !
;  201          0199  2 IF .ctl$t_nodename[0] NEQ 0                    ! If node present,
;  202          0200  2 OR .ctl$t_remoteid[0] NEQ 0
```

```
;  203          0201  2 THEN                                        ! get the remote node ID,
;  204          0202  3     BEGIN
;  205          0203  3     type = cia$k_network;
;  206          0204  3     CH$COPY(.ctl$t_nodename[0],             ! Copy the nodename first,
;  207          0205  3             ctl$t_nodename[1],
;  208          0206  3             .ctl$t_remoteid[0],             ! then the remote id
;  209          0207  3             ctl$t_remoteid[1],
;  210          0208  3             ,
;  211          0209  3             cia$s_data,
;  212          0210  3             data);
;  213          0211  3     END
;  214          0212  3
;  215          0213  2 ELSE IF .terminal_device
;  216          0214  2 THEN
;  217          0215  3     BEGIN
;  218          0216  3     IF .exe$gl_dynamic_flags[exe$v_brk_term]
;  219          0217  3     THEN term_length = .phy_term_name[0]
;  220          0218  3     ELSE term_length = 0;
;  221          0219  3
;  222          0220  3     IF .uaf_record NEQ 0
;  223          0221  3     THEN
;  224          0222  4         BEGIN
;  225          0223  4         type = cia$k_term_user;
;  226          0224  4         CH$COPY(.term_length,
;  227          0225  4                 .phy_term_name[1],
;  228          0226  4                 uaf$s_username,
;  229          0227  4                 uaf_record[uaf$t_username],
;  230          0228  4                 ,
;  231          0229  4                 cia$s_data,
;  232          0230  4                 data);
;  233          0231  4         END
;  234          0232  3     ELSE
;  235          0233  4         BEGIN
;  236          0234  4         type = cia$k_terminal;
;  237          0235  4         CH$COPY(.term_length,
;  238          0236  4                 ,phy_term_name[1],
;  239          0237  4                 ,
;  240          0238  4                 cia$s_data,
;  241          0239  4                 data);
;  242          0240  3         END;
;  243          0241  3     END
;  244          0242  3
;  245          0243  2 ELSE
;  246          0244  3     BEGIN
;  247          0245  3     type = cia$k_username;
;  248          0246  3     CH$COPY(uaf$s_username,
;  249          0247  3             org_username,
;  250          0248  3             ,
;  251          0249  3             cia$s_data,
;  252          0250  3             data);
;  253          0251  2     END;
;  254          0252  2
;  255          0253  2 !
;  256          0254  2 ! Set up pointers to the address of the data.  Also set STATUS = 1, to show
;  257          0255  2 ! that nothing so far.
;  258          0256  2 !
;  259          0257  2 status = 1;
```

```
;  260          0258  2
;  261          0259  2 !
;  262          0260  2 ! Scan the lists, if there are any blocks.
;  263          0261  2 !
;  264          0262  3 cia_search: BEGIN
;  265          0263  3 sch$lockw(cia$gl_mutex, .ctl$gl_pcb);
;  266          0264  3 cia = .cia$gq_intruder;
;  267          0265  3
;  268          0266  3 UNTIL .cia EQL cia$gq_intruder
;  269          0267  3 DO
;  270          0268  4     BEGIN
;  271          0269  5     cia_check: BEGIN
;  272          0270  5     next = .cia[cia$l_flink];
;  273          0271  5 !
;  274          0272  5 ! Check expiration on each entry encountered. Expiration is done with
;  275          0273  5 ! a simple compare of the entry's time against current time.
;  276          0274  5 !
;  277          0275  5
;  278          0276  5     IF cmpm (2, cia[cia$q_time], time) LEQ 0
;  279          0277  5     THEN
;  280          0278  6         BEGIN
;  281          0279  6
;  282          0280  6 !
;  283          0281  6 ! An expired intruder is turned back into a suspect with the
;  284          0282  6 ! count set to one short of the limit. The time is computed
;  285          0283  6 ! from the current time plus the count times the break timeout.
;  286          0284  6 !
;  287          0285  6
;  288          0286  6         IF .cia[cia$v_intruder]
;  289          0287  6         THEN
;  290          0288  7             BEGIN
;  291          0289  7             cia[cia$v_intruder] = 0;
;  292          0290  7             cia[cia$w_count] = .sys$gb_brk_lim;
;  293          0291  7             delta = .sys$gb_brk_lim * .sys$gl_brk_tmo;
;  294          0292  7             emul(%REF (10000000),       ! Convert seconds into delta time
;  295          0293  7                  delta,
;  296          0294  7                  %REF(0),
;  297          0295  7                  cia[cia$q_time]);
;  298          0296  7             addm(2, time, cia[cia$q_time], cia[cia$q_time]);
;  299          0297  7             END
;  300          0298  7
;  301          0299  7 !
;  302          0300  7 ! An expired suspect entry is removed from the list.
;  303          0301  7 !
;  304          0302  7
;  305          0303  6         ELSE
;  306          0304  7             BEGIN
;  307          0305  7             remque(.cia, dummy);                ! Remove the block
;  308          0306  7             exe$deanonpaged(.cia);              ! And return to pool
;  309          0307  7             LEAVE cia_check;                    ! No entry to check
;  310          0308  6             END;
;  311          0309  5         END;
;  312          0310  5
;  313          0311  5 !
;  314          0312  5 ! Now check for a match on the type we are looking for.
;  315          0313  5 !
;  316          0314  5
```

```
;  317          0315  5     IF .type EQL .cia[cia$b_subtype]
;  318          0316  5     AND CH$EQL (cia$s_data, data, cia$s_data, cia[cia$t_data])
;  319          0317  5 !
;  320          0318  5 ! If we have a matching block, make the appropriate checks.
;  321          0319  5 !
;  322          0320  5     THEN
;  323          0321  6         BEGIN
;  324          0322  6         IF .intruder
;  325          0323  6         THEN status = check_intruder(.cia, time)
;  326          0324  6         ELSE status = check_suspect(.cia, time);
;  327          0325  6         LEAVE cia_search;
;  328          0326  5         END;
;  329          0327  4     END;                                    ! end of block CIA_CHECK
;  330          0328  4     cia = .next;
;  331          0329  3     END;
;  332          0330  3
;  333          0331  3 !
;  334          0332  3 ! If this is a failed login and no list entry was found, create a
;  335          0333  3 ! suspect entry.
;  336          0334  3 !
;  337          0335  3 IF NOT .intruder
;  338          0336  3 THEN add_suspect(.type, data, time);
;  339          0337  2 END;                                        ! end of block CIA_SEARCH
;  340          0338  2
;  341          0339  2 !
;  342          0340  2 ! Unlock the mutex and lower IPL.
;  343          0341  2 !
;  344          0342  2 sch$unlock(cia$gl_mutex, .ctl$gl_pcb);
;  345          0343  2 SET_IPL(0);
;  346          0344  2
;  347          0345  2 break_attempt = NOT .status;
;  348          0346  2 RETURN .status;
;  349          0347  1 END;


                                                .TITLE  BREAKIN
                                                .IDENT  \V04-000\

                                                .EXTRN  IOC$CVT_DEVNAM, SCH$LOCKW
                                                .EXTRN  SCH$UNLOCK, EXE$ALONONPAGED
                                                .EXTRN  EXE$DEANONPAGED
                                                .EXTRN  BREAK_ATTEMPT, TERMINAL_DEVICE
                                                .EXTRN  PHY_TERM_NAME, UAF_RECORD
                                                .EXTRN  ORG_USERNAME, CTL$GL_PCB
                                                .EXTRN  CTL$T_NODENAME, CTL$T_REMOTEID
                                                .EXTRN  CIA$GL_MUTEX, CIA$GQ_INTRUDER
                                                .EXTRN  SYS$GB_BRK_LIM, SYS$GL_BRK_TMO
                                                .EXTRN  SYS$GL_HID_TIM, EXE$GL_DYNAMIC_FLAGS
                                                .EXTRN  EXE$V_BRK_TERM, SYS$GETTIM

                                                .PSECT  $CODE$,NOWRT,2

                                    0FFC 00000  .ENTRY  CIA_SCAN, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 0144
                                                R11                                                 ;
                          5E    C0  AE  9E 00002  MOVAB  -64(SP), SP                                  ;
                                    5E  DD 00006  PUSHL  SP                                         ; 0190
                00000000G   00      01  FB 00008  CALLS  #1, SYS$GETTIM                            ;
```

```
                                58 00000000G   00   9A 0000F          MOVZBL  CTL$T_NODENAME, R8                                    ; 0199
                                               08   12 00016          BNEQ    1$                                                  ;
                                   00000000G   00   95 00018          TSTB    CTL$T_REMOTEID                                      ; 0200
                                               2F   13 0001E          BEQL    2$                                                  ;
                                5A             03   90 00020 1$:      MOVB    #3, TYPE                                            ; 0203
                                59 00000000G   00   9A 00023          MOVZBL  CTL$T_REMOTEID, R9                                  ; 0206
                                57             38   D0 0002A          MOVL    #56, R7                                             ; 0207
                                56       08    AE   9E 0002D          MOVAB   DATA, R6                                            ;
        57      20 00000000G    00             58   2C 00031          MOVC5   R8, CTL$T_NODENAME+1, #32, R7, (R6)                 ;
                                               66        0003A          
                                               77   18 0003B          BGEQ    7$                                                  ;
                                56             58   C0 0003D          ADDL2   R8, R6                                              ;
                                57             58   C2 00040          SUBL2   R8, R7                                              ;
        57      20 00000000G    00             59   2C 00043          MOVC5   R9, CTL$T_REMOTEID+1, #32, R7, (R6)                 ;
                                               56        0004C          
                                               65   11 0004D          BRB     7$                                                  ; 0199
                                50 00000000G   00   E9 0004F 2$:      BLBC    TERMINAL_DEVICE, 6$                                 ; 0213
                09 00000000G    00             00G  E1 00056          BBC     S^EXE$V_BRK_TERM, EXE$GL_DYNAMIC_FLAGS, 3$          ; 0216
                                59 00000000G   00   D0 0005E          MOVL    PHY_TERM_NAME, TERM_LENGTH                          ; 0217
                                               02   11 00065          BRB     4$                                                  ;
                                               59   D4 00067 3$:      CLRL    TERM_LENGTH                                         ; 0218
                                50 00000000G   00   D0 00069 4$:      MOVL    PHY_TERM_NAME+4, R0                                 ; 0225
                                56 00000000G   00   D0 00070          MOVL    UAF_RECORD, R6                                      ; 0220
                                               21   13 00077          BEQL    5$                                                  ;
                                5A             02   90 00079          MOVB    #2, TYPE                                            ; 0223
                                58             38   D0 0007C          MOVL    #56, R8                                             ; 0227
                                57       08    AE   9E 0007F          MOVAB   DATA, R7                                            ;
        58      20              60             59   2C 00083          MOVC5   TERM_LENGTH, (R0), #32, R8, (R7)                    ;
                                               67        00088          
                                               29   18 00089          BGEQ    7$                                                  ;
                                57             59   C0 0008B          ADDL2   TERM_LENGTH, R7                                     ;
                                58             59   C2 0008E          SUBL2   TERM_LENGTH, R8                                     ;
        58      20        04    A6             20   2C 00091          MOVC5   #32, 4(R6), #32, R8, (R7)                           ;
                                               67        00097          
                                               1A   11 00098          BRB     7$                                                  ; 0220
                                5A             01   90 0009A 5$:      MOVB    #1, TYPE                                            ; 0234
        38      20              60             59   2C 0009D          MOVC5   TERM_LENGTH, (R0), #32, #56, DATA                   ; 0235
                                         08    AE        000A2          
                                               0E   11 000A4          BRB     7$                                                  ; 0213
                                5A             04   90 000A6 6$:      MOVB    #4, TYPE                                            ; 0245
        38      20 00000000G    00             20   2C 000A9          MOVC5   #32, ORG_USERNAME, #32, #56, DATA                   ; 0246
                                         08    AE        000B2          
                                59             01   D0 000B4 7$:      MOVL    #1, STATUS                                          ; 0257
                                50 00000000G   00   9E 000B7          MOVAB   CIA$GL_MUTEX, R0                                    ; 0263
                                54 00000000G   00   D0 000BE          MOVL    CTL$GL_PCB, R4                                      ;
                                   00000000G   00   16 000C5          JSB     SCH$LOCKW                                           ;
                                57 00000000G   00   D0 000CB          MOVL    CIA$GQ_INTRUDER, CIA                                ; 0264
                                50 00000000G   00   9E 000D2 8$:      MOVAB   CIA$GQ_INTRUDER, R0                                 ; 0266
                                50             57   D1 000D9          CMPL    CIA, R0                                             ;
                                               03   12 000DC          BNEQ    9$                                                  ;
                                           008E 31   000DE          BRW     18$                                                 ;
                                56             67   D0 000E1 9$:      MOVL    (CIA), NEXT                                         ; 0270
                                50             01   CE 000E4          MNEGL   #1, R0                                              ; 0276
                          04    AE       14    A7   D1 000E7          CMPL    20(CIA), TIME                                       ;
                                               0E   19 000EC          BLSS    12$                                                 ;
                                               08   14 000EE          BGTR    10$                                                 ;
                                6E       10    A7   D1 000F0          CMPL    16(CIA), TIME                                       ;
```

```
                                                04  13 000F4          BEQL    11$                                     ;
                                                04  1F 000F6          BLSSU   12$                                     ;
                                                50  D6 000F8 10$:     INCL    R0                                      ;
                                                50  D6 000FA 11$:     INCL    R0                                      ;
                                                50  D5 000FC 12$:     TSTL    R0                                      ;
                                                3E  14 000FE          BGTR    14$                                     ;
                                    2C      0C  A7  E9 00100          BLBC    12(CIA), 13$                            ; 0286
                                    0C      A7  01  8A 00104          BICB2   #1, 12(CIA)                             ; 0289
                                            50  00000000G 00  9A 00108 MOVZBL  SY$$GB_BRK_LIM, R0                      ; 0290
                                    0E      A7  50  B0 0010F          MOVW    R0, 14(CIA)                             ;
                                58          50  00000000G 00  C5 00113 MULL3   SY$$GL_BRK_TMO, R0, DELTA               ; 0291
    10  A7                      00          58  00989680  8F  7A 0011B EMUL    #10000000, DELTA, #0, 16(CIA)           ; 0295
                                    10      A7  6E  C0 00125          ADDL2   TIME, 16(CIA)                           ; 0296
                                    14      A7      04  AE  D8 00129  ADWC    TIME, 20(CIA)                           ;
                                                0E  11 0012E          BRB     14$                                     ; 0286
                                            5B      67  0F 00130 13$:     REMQUE  (CIA), DUMMY                        ; 0305
                                            50      57  D0 00133          MOVL    CIA, R0                             ; 0306
                                                00000000G 00  16 00136    JSB     EXE$DEANONPAGED                         ;
                                                2B  11 0013C          BRB     17$                                     ; 0307
                                    0B      A7      5A  91 0013E 14$:     CMPB    TYPE, 11(CIA)                       ; 0315
                                                25  12 00142          BNEQ    17$                                     ;
                            18  A7      08  AE      38  29 00144      CMPC3   #56, DATA, 24(CIA)                      ; 0316
                                                1D  12 0014A          BNEQ    17$                                     ;
                                            0B      04  AC  E9 0014C  BLBC    INTRUDER, 15$                           ; 0322
                                                4080    8F  BB 00150  PUSHR   #^M<R7,SP>                             ; 0323
                                    0000V   CF          02  FB 00154  CALLS   #2, CHECK_INTRUDER                      ;
                                                09  11 00159          BRB     16$                                     ;
                                                4080    8F  BB 0015B 15$:     PUSHR   #^M<R7,SP>                      ; 0324
                                    0000V   CF          02  FB 0015F  CALLS   #2, CHECK_SUSPECT                       ;
                                            59      50  D0 00164 16$:     MOVL    R0, STATUS                          ;
                                                17  11 00167          BRB     19$                                     ; 0325
                                            57      56  D0 00169 17$:     MOVL    NEXT, CIA                           ; 0328
                                                FF63    31  00016C    BRW     8$                                      ; 0266
                                            0D      04  AC  E8 0016F 18$:     BLBS    INTRUDER, 19$                   ; 0335
                                                5E  DD 00173          PUSHL   SP                                      ; 0336
                                                0C  AE  9F 00175      PUSHAB  DATA                                    ;
                                            7E      5A  9A 00178      MOVZBL  TYPE, -(SP)                             ;
                                    0000V   CF          03  FB 0017B  CALLS   #3, ADD_SUSPECT                         ;
                                            50  00000000G 00  9E 00180 19$:     MOVAB   CIA$GL_MUTEX, R0                ; 0342
                                            54  00000000G 00  D0 00187      MOVL    CTL$GL_PCB, R4                      ;
                                                00000000G 00  16 0018E    JSB     SCH$UNLOCK                              ;
                                            12      00  DA 00194          MTPR    #0, #18                             ; 0343
                                00000000G   00      59  92 00197      MCOMB   STATUS, BREAK_ATTEMPT                   ; 0345
                                            50      59  D0 0019E          MOVL    STATUS, R0                          ; 0346
                                                    04  001A1          RET                                             ; 0347

; Routine Size:  418 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  351          0348  1 ROUTINE check_suspect (cia, time) =
;  352          0349  2 BEGIN
;  353          0350  2
;  354          0351  2 !+++
;  355          0352  2 !
;  356          0353  2 ! Check if enough login failures have occurred to change the suspect
;  357          0354  2 ! to an intruder.
;  358          0355  2 !
;  359          0356  2 ! Inputs:
;  360          0357  2 !       cia - address of the Compound Intrusion Analysis block
;  361          0358  2 !       time - address of current time of login failure
;  362          0359  2 !
;  363          0360  2 ! Outputs:
;  364          0361  2 !       None.
;  365          0362  2 !
;  366          0363  2 ! Return status:
;  367          0364  2 !       0 - this is an intruder
;  368          0365  2 !       1 - just another suspect
;  369          0366  2 !
;  370          0367  2 !---
;  371          0368  2
;  372          0369  2 BUILTIN
;  373          0370  2     emul,
;  374          0371  2     addm;
;  375          0372  2
;  376          0373  2 MAP
;  377          0374  2     time : REF VECTOR[,WORD],
;  378          0375  2     cia : REF $BBLOCK;
;  379          0376  2
;  380          0377  2 LOCAL
;  381          0378  2     delta : VECTOR [2];
;  382          0379  2
;  383          0380  2 !
;  384          0381  2 ! Bump the count of login (perhaps breakin) attempts. Also bump the
;  385          0382  2 ! expiration time of the entry.
;  386          0383  2 !
;  387          0384  2 cia[cia$w_count] = .cia[cia$w_count] + 1;
;  388          0385  2 emul (sys$gl_brk_tmo,                    ! Convert timeout into
;  389          0386  2       %REF (10000000),                   ! delta time value
;  390          0387  2       %REF (0),
;  391          0388  2       delta);
;  392          0389  2 addm (2, delta, cia[cia$q_time], cia[cia$q_time]);
;  393          0390  2
;  394          0391  2 !
;  395          0392  2 ! If the number of attempts is greater than the number of retries allowed,
;  396          0393  2 ! then remove the CIA block from the suspect queue and put it on the
;  397          0394  2 ! intruder queue.
;  398          0395  2 !
;  399          0396  2 IF .cia[cia$w_count] GTR .sys$gb_brk_lim
;  400          0397  2 THEN
;  401          0398  3     BEGIN
;  402          0399  3     LOCAL
;  403          0400  3         semi_second;
;  404          0401  3
;  405          0402  3 !
;  406          0403  3 ! Obtain a number between 10000000 and 15000000, which represents a unit of
;  407          0404  3 ! time between 1.0 and 1.5 seconds.
```

```
;  408          0405  3 !
;  409          0406  3 ! This is done by taking the system time, and using the longword in the
;  410          0407  3 ! "middle", swapping the words of that longword, scaling that  quantity
;  411          0408  3 ! down to be in the range of 5000000 or less, and adding 10000000.
;  412          0409  3 !
;  413          0410  4     BEGIN
;  414          0411  4     MAP semi_second : VECTOR[2,WORD];
;  415          0412  4     semi_second[0] = .time[2];                  ! Get middle longword,
;  416          0413  4     semi_second[1] = .time[1];                  ! swapping words on the way.
;  417          0414  4     END;
;  418          0415        semi_second = .semi_second / ((1^31-1)/5000000) + 10000000;
;  419          0416
;  420          0417        emul(sys$gl_hid_tim,                        ! Multiply the hide time
;  421          0418             semi_second,                           ! to get number of seconds
;  422          0419             %REF(0),
;  423          0420             cia[cia$q_time]);                      ! and put it in the block
;  424          0421        addm(2,                                     ! then add
;  425          0422             .time,                                 ! the current time
;  426          0423             cia[cia$q_time],                       !
;  427          0424             cia[cia$q_time]);                      ! and put it in the block
;  428          0425        cia[cia$v_intruder] = 1;                    ! Show it's an intruder
;  429          0426  3     RETURN 0;
;  430          0427  2     END;
;  431          0428  2
;  432          0429  2 RETURN 1;
;  433          0430  1 END;


                                        000C 00000 CHECK_SUSPECT:
                                                   .WORD   Save R2,R3                                          ; 0348
                                    5E             0C  C2 00002         SUBL2   #12, SP
                                    50       04    AC  D0 00005         MOVL    CIA, R0                             ; 0384
                                             0E    A0  B6 00009         INCW    14(R0)
04  AE          00 00989680       8F 00000000G   00  7A 0000C         EMUL    SYS$GL_BRK_TMO, #10000000, #0, DELTA ; 0385
                                    53       10    A0  9E 0001A         MOVAB   16(R0), R3                          ; 0389
                                    63       04    AE  C0 0001E         ADDL2   DELTA, (R3)
                                             04    A3  D8 00022         ADWC    DELTA, 4(R3)
                                    51 00000000G   00  9A 00027         MOVZBL  SYS$GB_BRK_LIM, R1                  ; 0396
                                             0E    A0  51 B1 0002E      CMPW    R1, 14(R0)
                                                   33  1E 00032         BGEQU   1$
                                    51       08    AC  D0 00034         MOVL    TIME, R1                            ; 0412
                                    6E       04    A1  B0 00038         MOVW    4(R1), SEMI_SECOND
                                             02    AE  02 A1 B0 0003C   MOVW    2(R1), SEMI_SECOND+2                ; 0413
                                52                 6E 000001AD    8F C7 00041   DIVL3   #429, SEMI_SECOND, R2       ; 0415
                                                   6E 00989680    E2 9E 00049   MOVAB   10000000(R2), SEMI_SECOND
                                63          00     6E 00000000G   00 7A 00050   EMUL    SYS$GL_HID_TIM, SEMI_SECOND, #0, (R3) ; 0420
                                                   63             61 C0 00059   ADDL2   (R1), (R3)                  ; 0424
                                             04    A3       04    A1 D8 0005C   ADWC    4(R1), 4(R3)
                                             0C    A0             01 88 00061   BISB2   #1, 12(R0)                 ; 0425
                                                                  04 11 00065   BRB     2$                          ; 0426
                                    50                 01  D0 00067 1$:      MOVL    #1, R0                      ; 0429
                                                       04  0006A             RET
                                                   50  D4 0006B 2$:         CLRL    R0                          ; 0430
                                                       04  0006D             RET
```

; Routine Size: 110 bytes,   Routine Base: $CODE$ + 01A2

```
;  435          0431  1 ROUTINE check_intruder (cia, time) =
;  436          0432  2 BEGIN
;  437          0433  2
;  438          0434  2 !+++
;  439          0435  2 !
;  440          0436  2 ! Check if list entry is an intruder (for otherwise successful
;  441          0437  2 ! validation).
;  442          0438  2 !
;  443          0439  2 ! Inputs:
;  444          0440  2 !       cia - address of intruder block
;  445          0441  2 !       time - current time
;  446          0442  2 !
;  447          0443  2 ! Outputs:
;  448          0444  2 !       none.
;  449          0445  2 !
;  450          0446  2 ! Return status:
;  451          0447  2 !       0 - this is an intruder, perform evasive action
;  452          0448  2 !       1 - this is only suspect, no action
;  453          0449  2 !
;  454          0450  2 !---
;  455          0451  2
;  456          0452  2 MAP
;  457          0453  2     cia : REF $BBLOCK;
;  458          0454  2
;  459          0455  2 LOCAL
;  460          0456  2     dummy;
;  461          0457  2
;  462          0458  2 !
;  463          0459  2 ! If this is not an intruder entry, take no action.
;  464          0460  2 !
;  465          0461  2 IF NOT .cia[cia$v_intruder]
;  466          0462  2 THEN RETURN 1;
;  467          0463  2
;  468          0464  2 cia[cia$w_count] = .cia[cia$w_count] + 1;
;  469          0465  2 RETURN 0;
;  470          0466  1 END;


                                        0000 00000 CHECK_INTRUDER:
                                                   .WORD   Save nothing                    ; 0431
                                50      04   AC  D0 00002          MOVL    CIA, R0         ; 0461
                                04      0C   A0  E8 00006          BLBS    12(R0), 1$      ;
                                50           01  D0 0000A          MOVL    #1, R0          ; 0462
                                                 04 0000D          RET                     ;
                                             0E  A0  B6 0000E 1$:  INCW    14(R0)          ; 0464
                                                 50  D4 00011      CLRL    R0              ; 0465
                                                     04 00013      RET                     ; 0466

; Routine Size:  20 bytes,    Routine Base:  $CODE$ + 0210
```

```
;  472     0467  1 ROUTINE add_suspect (type, data,  time) =
;  473     0468  2 BEGIN
;  474     0469  2
;  475     0470  2 !+++
;  476     0471  2 !
;  477     0472  2 ! Add a suspect block
;  478     0473  2 !
;  479     0474  2 ! Inputs:
;  480     0475  2 !       type - type code of data
;  481     0476  2 !       data - data string to match on
;  482     0477  2 !       time - time of login attempt
;  483     0478  2 !
;  484     0479  2 ! Outputs:
;  485     0480  2 !       None.
;  486     0481  2 !
;  487     0482  2 !---
;  488     0483  2
;  489     0484  2 BUILTIN
;  490     0485  2     emul,
;  491     0486  2     addm,
;  492     0487  2     insque;
;  493     0488  2
;  494     0489  2 LOCAL
;  495     0490  2     size,
;  496     0491  2     cia : REF $BBLOCK;
;  497     0492  2
;  498     0493  2 !
;  499     0494  2 ! Get a chunk of non-paged pool.
;  500     0495  2 !
;  501     0496  2 IF NOT exe$alononpaged(cia$c_length; size, cia)
;  502     0497  2 THEN RETURN 1;
;  503     0498  2
;  504     0499  2 !
;  505     0500  2 ! Fill in the block.
;  506     0501  2 !
;  507     0502  2 cia[cia$w_size] = .size;                            ! Put in size
;  508     0503  2 cia[cia$b_type] = dyn$c_cia;                        ! Type
;  509     0504  2 cia[cia$b_subtype] = .type;                         ! Data type
;  510     0505  2 cia[cia$w_flags] = 0;                               ! This is a suspect
;  511     0506  2 cia[cia$w_count] = 1;                               ! Tried one time
;  512     0507  2 emul (sys$gl_brk_tmo,                               ! Convert timeout into
;  513     0508  2       %REF (10000000),                              ! delta time
;  514     0509  2       %REF (0),                                     ! and add to current time
;  515     0510  2       cia[cia$q_time]);
;  516     0511  2 addm (2, .time, cia[cia$q_time], cia[cia$q_time]);
;  517     0512  2 CH$MOVE(cia$s_data,                                 ! And this is the
;  518     0513  2         .data,                                      ! match data
;  519     0514  2         cia[cia$t_data]);
;  520     0515  2
;  521     0516  2 insque(.cia, .cia$gq_intruder[cia$l_blink]);        ! Put at end of intruder queue
;  522     0517  2
;  523     0518  2 RETURN 1;
;  524     0519  1 END;
```

```
                                                 0FFC 00000 ADD_SUSPECT:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 0467
                                      51          50    8F    9A 00002          MOVZBL  #80, R1                         ; 0496
                                           00000000G    00    16 00006          JSB     EXE$ALONONPAGED                 ;
                                      56                52    D0 0000C          MOVL    R2, R6                          ;
                                      42                50    E9 0000F          BLBC    R0, 1$                          ;
                                   08 A6                51    B0 00012          MOVW    SIZE, 8(CIA)                    ; 0502
                                   0A A6          45    8F    90 00016          MOVB    #69, 10(CIA)                    ; 0503
                                   0B A6          04    AC    90 0001B          MOVB    TYPE, 11(CIA)                   ; 0504
                                   0C A6 00010000       8F    D0 00020          MOVL    #65536, 12(CIA)                 ; 0505
      10   A6          00 00989680    8F 00000000G    00    7A 00028          EMUL    SYS$GL_BRK_TMO, #10000000, #0, 16(CIA)  ; 0510
                                      50          0C    BC    9E 00036          MOVAB   @TIME, R0                       ; 0511
                                   10 A6                60    C0 0003A          ADDL2   (R0), 16(CIA)                   ;
                                   14 A6          04    A0    D8 0003E          ADWC    4(R0), 20(CIA)                  ;
                             18    A6          08    BC    38    28 00043          MOVC3   #56, @DATA, 24(CIA)             ; 0514
                                      50 00000000G       00    9E 00049          MOVAB   CIA$GQ_INTRUDER+4, R0           ; 0516
                                   00 B0                66    0E 00050          INSQUE  (CIA), @0(R0)                   ;
                                      50                01    D0 00054 1$:      MOVL    #1, R0                          ; 0518
                                                              04 00057          RET                                     ; 0519

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 0224
```

```
;  526          0520  1 END
;  527          0521  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                        636  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                                 Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1            18619        24         0        1000        00:01.4
```

```
;                                        COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:BREAKIN/OBJ=OBJ$:BREAKIN MSRC$:BREAKIN/UPDATE=(ENH$:BREAKIN)

; Size:         636 code + 0 data bytes
; Run Time:         00:08.6
; Elapsed Time:     00:35.8
; Lines/CPU Min:     3626
; Lexemes/CPU-Min:  19595
; Memory Used:  132 pages
; Compilation Complete
```

PPDDEF
MDL

FILEIO
LIS

LOGIN

DETACHED
LIS

LOGINOUT
MAP

UTILDEF
REQ

LOGINCMD
CLD

HPWD
LIS

LGIDEF
MDL

AUDIT
LIS

BREAKIN
LIS