25

RACF

August 2001

In this issue

update

RACF Update

Managing RACF OMVS UIDs and GIDs

Access to Unix Systems Services (USS) for OS/390s is managed through the existence of user and group OMVS segments in RACF. Access to the file and directory structures maintained in the hierarchical file system is controlled by numeric identifiers known as UIDs and GIDs. UID and GID values are assigned to the OMVS segment for given userids and group names when those userids or groups are defined to RACF.

All userids that share a common UID have, by default, the same user access privileges for file and directory access. Similarly, all group names that share a common GID would have the same group access privileges for file and directory access.

The administration of UID and GID values is an age-old issue with Unix systems. RACF does little to help manage the potential problem that exists for UID and GID assignment. The ability to assign the same UID to multiple userids and to assign the same GID to multiple group names presents an interesting management and security challenge. This is entirely appropriate if userids JOHNS and SMITHJ represent the same user, and you may in fact want to assign the same UID to those different userids. However, if those two userids represent different users and system access levels should be different, you would not want them to have the same UID assignment.

There is no easy solution to this problem. Many sites maintain a directory of UID and GID values, and unique UID or GID values are assigned when a new userid or group name is created, based on the current values in use. This is fine if you know what the starting point is. Most OS/390 installations began their trek into USS rather blindly. This led to arbitrary UID and GID assignment, and overlap in UID and GID usage. What's more, they have no easy way of determining the current situation or the scope of the problem.

The program contained in this article examines the entire RACF database. It produces a report indicating all existing userids and the corresponding UID (if one exists) and default group name and GID (if one exists) for that userid. It also produces a report that indicates all

the defined UID values as well as all the userids that have been assigned to each used UID. This report is valuable in determining the current overlap of userids to UID values. A similar report is produced for GID values. This report can be used to determine whether multiple group names have been defined to the same GID.

The program can also be used to create two VSAM key sequenced datasets. The datasets are keyed on either the UID or GID (one dataset for each), and contained in each record in the dataset are the userids or group names that are currently associated with that specific UID or GID.

These files can be used as the starting point directory for managing the assignment of UID and GID values. Once you know the current state of your RACF environment, the VSAM files can be used to provide information about which UID or GID values are available for assignment. You can use this information to select only available values in your environment. New records would be added when a UID or GID value has been selected for use.

This tool is particularly useful for determining the current state of a RACF environment with respect to the RACF OMVS segments and how UID and GID values are being used.

CREATING THE LOAD MODULE

The UIDGIDST program needs to be assembled with your Assembler H or High Level Assembler. The resulting object module should be linkedited using the following sample job:

```
//jobname   JOB ...
//IEWL      EXEC  PGM=HEWLHØ96,PARM='XREF,LIST,MAP'
//SYSPRINT DD    SYSOUT=*
//SYSUT1    DD    UNIT=SYSDA,SPACE=(CYL,(2,1))
//OBJECT    DD    DSN=your.object.pds,DISP=SHR
//SYSLMOD   DD    DSN=an.auth.dataset,DISP=SHR
//SYSLIN    DD    *
   INCLUDE OBJECT(UIDGIDST)
   ENTRY   UIDGIDST
   SETCODE AC(1)
   NAME    UIDGIDST(R)
```

Note the requirement for APF authorization.

CREATING TARGET VSAM DATASETS

If you'll be using UIDGIDST to create the UID and GID VSAM state datasets, you'll need to pre-allocate the VSAM clusters. You may want to run UIDGIDST the first time with the BLDVSAM parameter not enabled. You can then examine the resulting output datasets to determine whether any UID or GID value has more than 1023 userids or group names associated with it. If so, you'll need to increase the maximum record size value in the DEFINE CLUSTER RECSZ parameter. You'll need to increase the maximum record size value by eight for each userid or group name over 1023.

For example, if you specify RECSZ(12 12288), you could accommodate up to 1535 userids or group names per record.

The following sample job can be used to create your VSAM state datasets:

```
//jobname  JOB  ...
//STEP1    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//FILEDD   DD   UNIT=339Ø,VOL=SER=volser,DISP=OLD
//SYSIN    DD   *
 DELETE hlq.OMVSUID.KSDS
 DEFINE CLUSTER -
      (NAME(hlq.OMVSUID.KSDS) -
       INDEXED                   -
       TRACKS(75 15)             -
       SHAREOPTIONS(2 3)         -
       RECSZ(12 8192)            -
       CISZ(8192)              -
       KEYS(4 Ø)                -
       VOL(volser)              -
       FILE(FILEDD)             -
       FREESPACE(3Ø 3Ø))          -
   DATA (NAME(hlq.OMVSUID.KSDS.DATA)) -
  INDEX (NAME(hlq.OMVSUID.KSDS.INDEX))
 DELETE hlq.OMVSGID.KSDS
 DEFINE CLUSTER -
      (NAME(hlq.OMVSGID.KSDS) -
       INDEXED                   -
       TRACKS(75 15)             -
       SHAREOPTIONS(2 3)         -
       RECSZ(12 8192)            -
       CISZ(8192)              -
       KEYS(4 Ø)                -
       VOL(volser)              -
```

```
      FILE(FILEDD)              -
      FREESPACE(3Ø 3Ø))          -
  DATA (NAME(hlq.OMVSGID.KSDS.DATA)) -
  INDEX (NAME(hlq.OMVSGID.KSDS.INDEX))
```

RUNNING THE UIDGIDST UTILITY

Once you've assembled and linked the UIDGIDST code and created
the VSAM state datasets, you're ready to examine your RACF
database. The UIDGIDST utility can be run with the BLDVSAM
parameter enabled or disabled. If the utility is run without the
BLDVSAM parameter, the UID and GID VSAM datasets will not be
created. If the BLDVSAM parameter is enabled, the UID and GID
VSAM datasets will be created. If the BLDVSAM parameter is
specified, the VSAM datasets specified in the UIDFILE and GIDFILE
DD statements should be newly-created VSAM datasets.

The following sample job can be used to create the userid report and
the UID/GID cross-reference reports without producing the VSAM
state datasets:

```
 USERID    NAME                    OMVS UID   GROUP    OMVS GID

 irrcerta  CERTAUTH Anchor
 irrsitec  SITE Anchor
 APPC      ####################             SYS1     ØØØØØØØØØØ
 ASCH      ####################             SYS1     ØØØØØØØØØØ
 BLSJPRMI  ####################             SYS1     ØØØØØØØØØØ
 BPXOINIT  BPXOINIT                ØØØØØØØØØØ SYS1     ØØØØØØØØØØ
 BPXROOT   #################### ØØØØØØØØØØ OMVSGRP  ØØØØØØØØØ1
 CICSA     CICSA                            SYS1     ØØØØØØØØØØ
 CICSTART  ####################             SYS1     ØØØØØØØØØØ
 CUSER1    C TEST DEVELOPER        ØØØØØØ1ØØØ SYS1     ØØØØØØØØØØ
 DB2IRLM   DB2IRLM                          SYS1     ØØØØØØØØØØ
 DCEADM    DCEADM                  ØØØØØØØØØØ SYS1     ØØØØØØØØØØ
 DCEKERN   #################### ØØØØØØØØØØ DCEGRP   ØØØØØØØØØ2
 DFS       #################### ØØØØØØØØØØ DFSGRP   ØØØØØØØØØ2
 DFSCM     #################### ØØØØØØØØØØ DFSGRP   ØØØØØØØØØ2
 DSN1DBM1  DSN1DBM1                         SYS1     ØØØØØØØØØØ
```

Figure 1: Sample report

```
//jobname  JOB  ...
//UIDGIDST EXEC PGM=UIDGIDST
//STEPLIB  DD   DSN=an.auth.dataset,DISP=SHR
//OUTPUT   DD   SYSOUT=*
//UGIDXREF DD   SYSOUT=*
```

The following sample job can be used to create the userid report, the UID/GID cross-reference reports, and the VSAM state datasets:

```
UID CROSS-REFERENCE AND CONFLICT REPORT

UID 0000000000 USED BY FOLLOWING USERIDS
     BPXOINIT
     BPXROOT
     DCEADM
     DCEKERN
     DFS
     DFSCM
     DSN1DIST
     EZAFTPAP
     FTPD
     HODSERVR
     IBMUSER
     IMWEBSRV
     NFSC
     NFSS
     OMVS
     OMVSKERN
     OPEN1
     OPEN2
     OPEN3
     PORTMAP
     TCPIP
     TCPIPOE
     WEBSRV

 UID 0000000206 USED BY FOLLOWING USERIDS
     WEBADM

 UID 0000000416 USED BY FOLLOWING USERIDS
     PRIVATE

 UID 0000000445 USED BY FOLLOWING USERIDS
     TCPIPMVS
```

Figure 2: Sample UID cross-reference report

```
//jobname  JOB  ...
//UIDGIDST EXEC PGM=UIDGIDST
//STEPLIB  DD   DSN=an.auth.dataset,DISP=SHR
//OUTPUT   DD   SYSOUT=*
//UGIDXREF DD   SYSOUT=*
//UIDFILE  DD   DSN=hlq.OMVSUID.KSDS,DISP=SHR
//GIDFILE  DD   DSN=hlq.OMVSGID.KSDS,DISP=SHR
```

SAMPLE OUTPUT

When the UIDGIDST utility is run, it will produce three reports. The first report is contained in the OUTPUT DD and will look similar to the sample excerpt shown in Figure 1.

From the above report:

- USERID is the userid as defined to RACF.

- NAME is the arbitrary user name value that is assigned to the userid when it's created.

- OMVS UID is the UID value that has been assigned to the userid (if the user has an OMVS segment).

- GROUP is the default group that has been assigned to this userid.

```
GID CROSS-REFERENCE AND CONFLICT REPORT

GID 0000000000 USED BY FOLLOWING GROUPS
    SYS1

GID 0000000001 USED BY FOLLOWING GROUPS
    OMVSGRP

GID 0000000002 USED BY FOLLOWING GROUPS
    DCEGRP
    DFSGRP

GID 0000000205 USED BY FOLLOWING GROUPS
    IMWEB

GID 0000000255 USED BY FOLLOWING GROUPS
    SPECIAL
```

Figure 3: Sample GID cross-reference report

- OMVS GID is the GID value that has been assigned to this user's default group (if the group has an OMVS segment).

The UGIDXREF DD is used for both the UID cross-reference report and the GID cross-reference report. These reports list the UID or GID values defined to the RACF database and all userids or group names associated with each UID or GID value. Sample UID and GID cross-reference report excerpts are shown in Figures 2 and 3.

SOURCE

```
UIDGIDST CSECT
UIDGIDST AMODE 31
UIDGIDST RMODE 24
***********************************************************************
*                                                                     *
*   THE UIDGIDST UTILITY IS USEFUL FOR DETERMINING THE CURRENT OMVS   *
*   SEGMENT STATUS FOR A RACF DATABASE. THE UTILITY PRODUCES A        *
*   USERID LIST REPORT THAT INCLUDES THE OMVS UID (IF ONE EXISTS),    *
*   THE DEFAULT GROUP FOR THE USERID, AND THE OMVS GID FOR THAT       *
*   GROUP (IF ONE EXISTS).                                            *
*                                                                     *
*   OPTIONALLY, UIDGIDST CAN BE USED TO CREATE VSAM KEY SEQUENCED     *
*   FILES THAT CONSIST OF RECORDS THAT CONTAIN THE UID OR GID AS      *
*   THE KEY FOR THE RECORD AND ALL THE USERIDS OR GROUP NAMES THAT    *
*   HAVE BEEN ASSOCIATED WITH THAT UID OR GID. THE UID RECORDS ARE    *
*   MAINTAINED IN A SEPARATE VSAM DATASET FROM THE GID RECORDS AS     *
*   DEFINED BY THE UIDFILE AND GIDFILE DD STATEMENTS. YOU REQUEST     *
*   THE CREATION OF THE VSAM DATASETS BY SPECIFYING PARM='BLDVSAM'    *
*   WHEN THE UIDGIDST PROGRAM IS EXECUTED. TO ENSURE PROPER           *
*   PROGRAM EXECUTION, THE VSAM DATASETS SHOULD BE NEWLY DEFINED      *
*   CLUSTERS WITH NO RECORD ENTRIES.                                  *
*                                                                     *
*   UIDGIDST SHOULD BE LINKEDITED INTO AN AUTHORIZED LIBRARY.         *
*                                                                     *
***********************************************************************
         STM   R14,R12,12(R13)          SAVE INCOMING REGISTERS
         LR    R12,R15                  GET MODULE ADDRESS
         LA    R11,4095(,R12)           SET SECOND BASE ...
         LA    R11,1(,R11)               REGISTER ADDRESS
         USING UIDGIDST,R12,R11         SET MODULE ADDRESSABILITY
         LR    R8,R1                    SAVE PARM ADDRESS
         LR    R10,R13                  SAVE SAVEAREA ADDRESS
         STORAGE OBTAIN,LENGTH=WORKLEN,LOC=BELOW
         LR    R0,R1                    COPY STORAGE ADDRESS
         LR    R2,R1                    AGAIN
         LR    R14,R1                   AND AGAIN
```

```
        L     R1,=A(WORKLEN)          GET STORAGE LENGTH
        XR    R15,R15                 SET FILL BYTE VALUE
        MVCL  RØ,R14                  CLEAR THE STORAGE
        ST    R1Ø,4(,R2)              SAVE OLD SAVEAREA ADDRESS
        LR    R13,R2                  LOAD NEW SAVEAREA ADDRESS
        USING WORKAREA,R13            SET ADDRESSABILITY
*********************************************************************
*   DETERMINE IF THE BLDVSAM PARAMETER HAS BEEN SPECIFIED.       *
*********************************************************************
        MVI   FLAG,X'ØØ'              SET FLAG
        L     R9,Ø(,R8)               GET ADDRESS OF PARM
        CLC   Ø(2,R9),=H'Ø'           A PARM?
        BE    NOPARM                  NO - NOTHING TO DO
        CLC   Ø(2,R9),=H'7'           CORRECT PARM?
        BNE   WRNGPARM                NO - ISSUE MESSAGE
        CLC   2(7,R9),=C'BLDVSAM'     CORRECT PARM?
        BNE   WRNGPARM                NO - ISSUE MESSAGE
        OI    FLAG,BLDVSAM            SET FLAG
        B     NOPARM                  KEEP GOING
WRNGPARM EQU  *
        WTO   'UIDGIDST - INVALID PARM SPECIFIED. PARM IGNORED.',   X
              ROUTCDE=(1),DESC=(6)
NOPARM   EQU  *
*********************************************************************
        MVC   OUTDCBWK(OUTDCBL),OUTPUT COPY THE DCB
        LA    R8,OUTDCBWK            GET DCB ADDRESS
        OI    OPENLST,X'8Ø'          SET PARM BIT ON
        OPEN  ((R8),OUTPUT),MODE=31,MF=(E,OPENLST) OPEN THE DCB
        PUT   (R8),OUTHDR            OUTPUT THE HEADER
        PUT   (R8),BLANKS            OUTPUT THE HEADER
*********************************************************************
*   OBTAIN STORAGE TO MAINTAIN AN INTERNAL REPRESENTATION OF USERIDS  *
*   TO UIDS AND GROUP NAMES TO GIDS.                              *
*********************************************************************
        STORAGE OBTAIN,LENGTH=4Ø96,LOC=ANY
        ST    R1,MAINUIDT            SAVE ADDRESS OF UID TABLE
        L     R15,=F'512'            SET LOOP COUNT
UIDTINIT EQU  *
        MVC   Ø(8,R1),=X'FFFFFFFFØØØØØØØØ'
        LA    R1,8(,R1)              POINT TO NEXT ENTRY
        BCT   R15,UIDTINIT           PROCESS NEXT ENTRY
        STORAGE OBTAIN,LENGTH=4Ø96,LOC=ANY
        ST    R1,MAINGIDT            SAVE ADDRESS OF GID TABLE
        LR    RØ,R1                  COPY STORAGE ADDRESS
        L     R14,MAINUIDT           GET SOURCE DATA ADDRESS
        L     R1,=F'4Ø96'            GET STORAGE LENGTH
        LR    R15,R1                 GET SOURCE LENGTH
        MVCL  RØ,R14                 INITIALIZE
        MVC   UIDTLEN(4),=F'4Ø96'    SET INITIAL LENGTH
        MVC   GIDTLEN(4),=F'4Ø96'    SET INITIAL LENGTH
```

```
        ****************************************************************
        XC    XUID(4),XUID             CLEAR LENGTH AREA
        MVC   XUID(2),=H'8'            SET LENGTH
        MVC   XUID+4(8),=8C' '         SET STARTING USER ID VALUE
        MODESET MODE=SUP,KEY=ZERO
        ****************************************************************
        *    LOOP THROUGH THE DEFINED USERIDS                          *
        ****************************************************************
USERLOOP EQU   *
        MVC   ROUTWRK1(ROUTLEN1),RACROUT1 MOVE IN RACROUTE MODEL
        RACROUTE REQUEST=EXTRACT,                                     X
              TYPE=EXTRACTN,                                          X
              ENTITYX=XUID,                                           X
              FIELDS=FLDLIST1,                                        X
              RELEASE=1.9.2,                                          X
              WORKA=RACWORK,MF=(E,ROUTWRK1)
        ST    R15,RETCODE              SAVE THE RETURN CODE
        LTR   R15,R15                  DATA EXTRACTED?
        BNZ   CHKGRPTB                 NO - MAKE A PASS THRU GRP TBL
        MVC   OUTBUF(8Ø),=8ØC' '       CLEAR THE BUFFER
        MVC   OUTBUF(8),XUID+4         MOVE IN THE USERID
        USING EXTWKEA,R1               EXTRACT WORKAREA ADDRESSABILITY
        XR    R15,R15                  CLEAR R15
        ICM   R15,B'ØØ11',EXTWOFF      GET OFFSET OF DATA AREA
        AR    R15,R1                   POINT TO PROGRAMMER NAME
        XC    NAMEBUF(8Ø),NAMEBUF      CLEAR THE AREA
        XR    R7,R7                    CLEAR R7
        ICM   R7,B'1111',Ø(R15)        GET PROGRAMMER NAME LENGTH
        BCTR  R7,Ø                     REDUCE BY ONE FOR EX
        EX    R7,NAMEMVC               MOVE THE NAME INTO THE BUF
        MVC   OUTBUF+9(2Ø),NAMEBUF     MOVE IN THE PROGRAMMER NAME
        LA    R7,1(,R7)                SET LENGTH BACK
        LA    R15,4(R7,R15)            POINT TO NEXT FIELD
        CLC   Ø(4,R15),=F'Ø'           ANY FIELD DATA?
        BE    NOGRP                    NO - NO GROUP DATA
        ICM   R7,B'1111',Ø(R15)        GET GROUP LENGTH
        BCTR  R7,Ø                     REDUCE BY ONE FOR EX
        EX    R7,GRPMVC                MOVE THE GROUP INTO A BUFFER
        MVC   OUTBUF+41(8),GRPBUF      MOVE IN THE GROUP NAME
NOGRP   EQU   *
        XR    R7,R7                    CLEAR R7
        ICM   R7,B'Ø111',EXTWLN        LOAD STORAGE LENGTH
        XR    R8,R8                    CLEAR R8
        ICM   R8,B'ØØØ1',EXTWSP        LOAD THE SUBPOOL
        STORAGE RELEASE,LENGTH=(R7),ADDR=(R1),SP=(R8)
        ****************************************************************
        *    FOR EACH USERID, ATTEMPT TO OBTAIN THE UID INFORMATION FROM THE   *
        *    OMVS SEGMENT.                                             *
        ****************************************************************
        MVC   ROUTWRK2(ROUTLEN2),RACROUT2 MOVE IN RACROUTE MODEL
```

```
         RACROUTE REQUEST=EXTRACT,                                  X
               TYPE=EXTRACT,                                        X
               ENTITY=XUID+4,                                       X
               FIELDS=FLDLIST2,                                     X
               RELEASE=1.9.2,                                       X
               WORKA=RACWORK,MF=(E,ROUTWRK2)
         ST    R15,RETCODE            SAVE THE RETURN CODE
         LTR   R15,R15                DATA EXTRACTED?
         BNZ   NOUIDINF               NO - NOTHING TO DO
         USING EXTWKEA,R1             EXTRACT WORKAREA ADDRESSABILITY
         ST    R1,TEMPEXTR            SAVE EXTRACT AREA ADDRESS
         XR    R15,R15                CLEAR R15
         ICM   R15,B'0011',EXTWOFF    GET OFFSET OF DATA AREA
         AR    R15,R1                 POINT TO UID
         XR    R7,R7                  CLEAR R7
         ICM   R7,B'1111',0(R15)      GET UID LENGTH
         LTR   R7,R7                  ANY DATA?
         BNZ   UIDOK                  YES - UID IS OK
         MVC   WTOWRK1(WTOLEN1),WTOLST1 MOVE IN THE WTO MODEL
         MVC   WTOWRK1+4(38),WTO1     MOVE IN THE WTO TEXT
         MVC   WTOWRK1+34(8),XUID+4   MOVE IN THE USERID
         WTO   MF=(E,WTOWRK1)         ISSUE THE WTO
         B     NOUID                  BYPASS UID PROCESSING
UIDOK    EQU   *
         BCTR  R7,0                   REDUCE BY ONE FOR EX
         EX    R7,UIDMVC              MOVE THE OMVS UID INTO THE BUF
         MVC   DBL2(4),UIDBUF         SAVE THE UID
         L     R8,UIDBUF              COPY THE UID
         UNPK  DBL1(9),DBL2(5)        UNPACK THE UID
         NC    DBL1(8),=8X'0F'        ZERO FIRST NIBBLES
         TR    DBL1(8),=C'0123456789ABCDEF' MAKE IT READABLE
         CVD   R8,DBL2                CONVERT TO DECIMAL
         UNPK  DBL1(16),DBL2(8)       UNPACK
         OC    DBL1(16),=16X'F0'      MAKE IT READABLE
         MVC   OUTBUF+30(10),DBL1+6   MOVE IN THE DECIMAL NUMBER
         CLC   UIDBUF(4),=X'FFFFFFFF' VALID UID?
         BNE   UIDVALID               YES - KEEP GOING
         MVC   OUTBUF+30(10),=C'      NONE' MOVE IN UID
         B     NOUID                  JUST BYPASS
UIDVALID EQU   *
         L     R8,MAINUIDT            GET TABLE ADDRESS
*******************************************************************
*    CHECK TO SEE IF UID ALREADY IN TABLE                        *
*******************************************************************
         L     R15,UIDTLEN            GET TABLE LENGTH
         LA    R15,0(R15,R8)          SET END OF TABLE ADDRESS
UIDTLP1  EQU   *
         CLC   0(4,R8),UIDBUF         UID ALREADY IN USE?
         BE    UIDMATCH               YES - USE CONFLICT TABLE
         CLC   0(4,R8),=X'FFFFFFFF'   A BLANK ENTRY?
```

```
              BE     UIDNEW                 YES - WE HAVE A NEW ENTRY
              LA     R8,8(,R8)              SET TO NEXT ENTRY
              CR     R8,R15                 END OF TABLE?
              BL     UIDTLP1                NO - KEEP GOING
       *********************************************************************
       *    THE CURRENT UID TABLE IS FULL. EXTEND THE TABLE BY 4096 BYTES,   *
       *    COPY THE EXISTING TABLE TO THE NEW TABLE, INITIALIZE THE         *
       *    ADDITIONAL 4096 BYTES, AND RELEASE THE OLD TABLE STORAGE.        *
       *********************************************************************
              L      R7,UIDTLEN             GET UID TABLE LENGTH
              A      R7,=F'4096'            ADD 4096
              MODESET MODE=PROB,KEY=NZERO
              STORAGE OBTAIN,LENGTH=(R7)    GET A NEW BLOCK
              ST     R1,TEMP                SAVE THE ADDRESS
              LR     R0,R1                  COPY THE ADDRESS
              L      R14,MAINUIDT           GET OLD TABLE ADDRESS
              L      R1,UIDTLEN             GET TABLE LENGTH
              LR     R15,R1                 COPY THE LENGTH
              MVCL   R0,R14                 COPY THE OLD TABLE
              L      R1,TEMP                GET NEW TABLE ADDRESS
              L      R15,UIDTLEN            GET OLD TABLE LENGTH
              LA     R1,0(R15,R1)           POINT PAST CURRENT DATA
              LR     R8,R1                  SAVE THIS ADDRESS
              L      R15,=F'512'            SET A LOOP COUNT
       UIDINIT2 EQU  *
              MVC    0(8,R1),=X'FFFFFFFF00000000' INITIALIZE THE ENTRY
              LA     R1,8(,R1)              POINT TO NEXT ENTRY
              BCT    R15,UIDINIT2           BRANCH IF MORE
              L      R9,UIDTLEN             GET OLD TABLE LENGTH
              L      R1,MAINUIDT            GET OLD TABLE ADDRESS
              STORAGE RELEASE,LENGTH=(R9),ADDR=(R1)
              MODESET MODE=SUP,KEY=ZERO
              MVC    MAINUIDT(4),TEMP       COPY IN NEW TABLE ADDRESS
              L      R9,UIDTLEN             GET OLD TABLE LENGTH
              A      R9,=F'4096'            ADD 4096
              ST     R9,UIDTLEN             SAVE NEW LENGTH
              B      UIDNEW                 GET USERID TABLE FOR NEW UID
       UIDMATCH EQU  *
       *********************************************************************
       *    THE UID IS ALREADY IN USE BY OTHER USERIDS. ADD THIS USERID     *
       *    TO THE LIST.                                                    *
       *********************************************************************
              L      R15,4(,R8)             GET CONFLICT TABLE ADDRESS
              L      R1,4(,R15)             GET CURRENT NUMBER OF ENTRIES
              L      R2,0(,R15)             GET TABLE LENGTH
              SRL    R2,3                   DIVIDE BY 8
              BCTR   R2,0                   SUBTRACT 1
              CR     R1,R2                  MAX ENTRIES?
              BL     UIDMAXOK               NO - WILL FIT IN CURRENT TABLE
       *********************************************************************
```

13

```
*    THE CURRENT USERID CONFLICT TABLE FOR THIS UID IS FULL. EXTEND    *
*    THE TABLE BY 4Ø96 BYTES, COPY THE EXISTING TABLE TO THE NEW       *
*    TABLE, RELEASE THE OLD TABLE STORAGE, AND ADD THE NEW USERID      *
*    INTO THE EXTENDED TABLE.                                          *
*********************************************************************
         LA    R2,1(,R2)             ADD ONE
         SLL   R2,3                  MULTIPLY BY 8
         A     R2,=F'4Ø96'           ADD 4Ø96
         MODESET MODE=PROB,KEY=NZERO
         STORAGE OBTAIN,LENGTH=(R2)
         ST    R1,TEMP               SAVE THE ADDRESS
         LR    RØ,R1                 COPY IT
         L     R14,4(,R8)            GET CURRENT TABLE ADDRESS
         L     R1,Ø(,R14)            GET CURRENT TABLE LENGTH
         LR    R15,R1                COPY IT
         MVCL  RØ,R14                COPY THE CURRENT TABLE
         L     R1,4(,R8)             GET CURRENT TABLE ADDRESS
         L     R2,Ø(,R1)             GET CURRENT TABLE LENGTH
         STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
         MODESET MODE=SUP,KEY=ZERO
         MVC   4(4,R8),TEMP          MOVE NEW TABLE ADDRESS
         L     R15,TEMP              GET NEW TABLE ADDRESS
         L     R1,4(,R15)            GET CURRENT # OF ENTRIES
UIDMAXOK EQU   *
         SLL   R1,3                  MULTIPLY BY 8
         LA    R1,8(R1,R15)          ADD IN HEADER LENGTH
         MVC   Ø(8,R1),XUID+4        MOVE IN NEW USERID
         L     R1,4(,R15)            GET CURRENT NUMBER OF ENTRIES
         LA    R1,1(,R1)             ADD ONE
         ST    R1,4(,R15)            SAVE NEW COUNT
         B     NOUID                 GO ON
UIDNEW   EQU   *
*********************************************************************
*    A NEW UID HAS BEEN ENCOUNTERED. ACQUIRE STORAGE FOR A TABLE      *
*    THAT WILL CONTAIN THE USERIDS THAT HAVE THIS ASSIGNED UID.       *
*********************************************************************
         MODESET MODE=PROB,KEY=NZERO
         STORAGE OBTAIN,LENGTH=4Ø96,LOC=ANY
         MVC   Ø(4,R8),UIDBUF        SAVE THE UID
         ST    R1,4(,R8)             SAVE THE TABLE ADDRESS
         LR    RØ,R1                 COPY THE TABLE ADDRESS
         LR    R14,R1                AGAIN
         L     R1,=F'4Ø96'           SET THE LENGTH
         XR    R15,R15               SET FILL BYTE
         MVCL  RØ,R14                INITIALIZE
         L     R1,4(,R8)             GET TABLE ADDRESS
         MVC   Ø(4,R1),=F'4Ø96'      SET TABLE LENGTH
         MVC   4(4,R1),=F'1'         SET USERID COUNT
         MVC   8(8,R1),XUID+4        MOVE IN USERID
         MODESET MODE=SUP,KEY=ZERO
```

```
        B     NOUID                  GO ON
NOUID   EQU   *
        L     R1,TEMPEXTR            GET EXTRACT AREA ADDRESS
        XR    R7,R7                  CLEAR R7
        ICM   R7,B'Ø111',EXTWLN      LOAD STORAGE LENGTH
        XR    R8,R8                  CLEAR R8
        ICM   R8,B'ØØØ1',EXTWSP      LOAD THE SUBPOOL
        STORAGE RELEASE,LENGTH=(R7),ADDR=(R1),SP=(R8)
NOUIDINF EQU  *
**********************************************************************
*    ATTEMPT TO EXTRACT THE OMVS INFORMATION FOR THE DEFAULT GROUP   *
*    FOR THIS USERID.                                                *
**********************************************************************
        MVC   ROUTWRK3(ROUTLEN3),RACROUT3 MOVE IN RACROUTE MODEL
        RACROUTE REQUEST=EXTRACT,                                   X
              TYPE=EXTRACT,                                         X
              ENTITY=GRPBUF,                                        X
              FIELDS=FLDLIST3,                                      X
              RELEASE=1.9.2,                                        X
              WORKA=RACWORK,MF=(E,ROUTWRK3)
        ST    R15,RETCODE            SAVE THE RETURN CODE
        LTR   R15,R15                DATA EXTRACTED?
        BNZ   NOGIDINF               NO - NOTHING TO DO
        USING EXTWKEA,R1             EXTRACT WORKAREA ADDRESSABILITY
        ST    R1,TEMPEXTR            SAVE EXTRACT AREA ADDRESS
        XR    R15,R15                CLEAR R15
        ICM   R15,B'ØØ11',EXTWOFF    GET OFFSET OF DATA AREA
        AR    R15,R1                 POINT TO GID
        XR    R7,R7                  CLEAR R7
        ICM   R7,B'1111',Ø(R15)      GET GID LENGTH
        LTR   R7,R7                  ANY DATA?
        BNZ   GIDOK                  YES - GID IS OK
        MVC   WTOWRK1(WTOLEN1),WTOLST1 MOVE IN THE WTO MODEL
        MVC   WTOWRK1+4(38),WTO2     MOVE IN THE WTO TEXT
        MVC   WTOWRK1+34(8),GRPBUF   MOVE IN THE GROUP NAME
        WTO   MF=(E,WTOWRK1)         ISSUE THE WTO
        B     NOGID                  BYPASS GID PROCESSING
GIDOK   EQU   *
        BCTR  R7,Ø                   REDUCE BY ONE FOR EX
        EX    R7,GIDMVC              MOVE THE OMVS GID INTO THE BUF
        MVC   DBL2(4),GIDBUF         SAVE THE GID
        L     R8,GIDBUF              COPY THE GID
        UNPK  DBL1(9),DBL2(5)        UNPACK THE GID
        NC    DBL1(8),=8X'ØF'        ZERO FIRST NIBBLES
        TR    DBL1(8),=C'Ø123456789ABCDEF' MAKE IT READABLE
        CVD   R8,DBL2                CONVERT TO DECIMAL
        UNPK  DBL1(16),DBL2(8)       UNPACK
        OC    DBL1(16),=16X'FØ'      MAKE IT READABLE
        MVC   OUTBUF+5Ø(1Ø),DBL1+6   MOVE IN THE DECIMAL NUMBER
        L     R8,MAINGIDT            GET TABLE ADDRESS
```

15

```
        ***********************************************************************
        *    CHECK TO SEE IF GID ALREADY IN TABLE                             *
        ***********************************************************************
                L       R15,GIDTLEN             GET TABLE LENGTH
                LA      R15,Ø(R15,R8)           SET END OF TABLE ADDRESS
        GIDTLP1 EQU     *
                CLC     Ø(4,R8),GIDBUF          GID ALREADY IN USE?
                BE      GIDMATCH                YES - USE CONFLICT TABLE
                CLC     Ø(4,R8),=X'FFFFFFFF'    A BLANK ENTRY?
                BE      GIDNEW                  YES - WE HAVE A NEW ENTRY
                LA      R8,8(,R8)               SET TO NEXT ENTRY
                CR      R8,R15                  END OF TABLE?
                BL      GIDTLP1                 NO - KEEP GOING
        ***********************************************************************
        *    THE CURRENT GID TABLE IS FULL. EXTEND THE TABLE BY 4Ø96 BYTES,   *
        *    COPY THE EXISTING TABLE TO THE NEW TABLE, INITIALIZE THE         *
        *    ADDITIONAL 4Ø96 BYTES, AND RELEASE THE OLD TABLE STORAGE.        *
        ***********************************************************************
                L       R7,GIDTLEN              GET GID TABLE LENGTH
                A       R7,=F'4Ø96'             ADD 4096
                MODESET MODE=PROB,KEY=NZERO
                STORAGE OBTAIN,LENGTH=(R7)      GET A NEW BLOCK
                ST      R1,TEMP                 SAVE THE ADDRESS
                LR      RØ,R1                   COPY THE ADDRESS
                L       R14,MAINGIDT            GET OLD TABLE ADDRESS
                L       R1,GIDTLEN              GET TABLE LENGTH
                LR      R15,R1                  COPY THE LENGTH
                MVCL    RØ,R14                  COPY THE OLD TABLE
                L       R1,TEMP                 GET NEW TABLE ADDRESS
                L       R15,GIDTLEN             GET OLD TABLE LENGTH
                LA      R1,Ø(R15,R1)            POINT PAST CURRENT DATA
                LR      R8,R1                   SAVE THIS ADDRESS
                L       R15,=F'512'             SET A LOOP COUNT
        GIDINIT2 EQU    *
                MVC     Ø(8,R1),=X'FFFFFFFFØØØØØØØØ' INITIALIZE THE ENTRY
                LA      R1,8(,R1)               POINT TO NEXT ENTRY
                BCT     R15,GIDINIT2            BRANCH IF MORE
                L       R9,GIDTLEN              GET OLD TABLE LENGTH
                L       R1,MAINGIDT             GET OLD TABLE ADDRESS
                STORAGE RELEASE,LENGTH=(R9),ADDR=(R1)
                MODESET MODE=SUP,KEY=ZERO
                MVC     MAINGIDT(4),TEMP        COPY IN NEW TABLE ADDRESS
                L       R9,GIDTLEN              GET OLD TABLE LENGTH
                A       R9,=F'4Ø96'             ADD 4096
                ST      R9,GIDTLEN              SAVE NEW LENGTH
                B       GIDNEW                  GET GROUP TABLE FOR NEW GID
        GIDMATCH EQU    *
        ***********************************************************************
        *    THE GID IS ALREADY IN USE BY OTHER GROUP NAMES. ADD THIS GROUP   *
        *    NAME TO THE LIST.                                                *
        ***********************************************************************
```

```
        L     R15,4(,R8)              GET CONFLICT TABLE ADDRESS
        L     R1,4(,R15)              GET CURRENT NUMBER OF ENTRIES
        L     R2,0(,R15)              GET TABLE LENGTH
        SRL   R2,3                    DIVIDE BY 8
        BCTR  R2,0                    SUBTRACT 1
        CR    R1,R2                   MAX ENTRIES?
        BL    GIDMAXOK                NO - WILL FIT IN CURRENT TABLE
*********************************************************************
*   THE CURRENT GROUP NAME CONFLICT TABLE FOR THIS GID IS FULL.    *
*   EXTEND THE TABLE BY 4096 BYTES, COPY THE EXISTING TABLE TO THE *
*   NEW TABLE, RELEASE THE OLD TABLE STORAGE, AND ADD THE NEW GROUP*
*   NAME INTO THE EXTENDED TABLE.                                  *
*********************************************************************
        LA    R2,1(,R2)               ADD ONE
        SLL   R2,3                    MULTIPLY BY 8
        A     R2,=F'4096'             ADD 4096
        MODESET MODE=PROB,KEY=NZERO
        STORAGE OBTAIN,LENGTH=(R2)
        ST    R1,TEMP                 SAVE THE ADDRESS
        LR    R0,R1                   COPY IT
        L     R14,4(,R8)              GET CURRENT TABLE ADDRESS
        L     R1,0(,R14)              GET CURRENT TABLE LENGTH
        LR    R15,R1                  COPY IT
        MVCL  R0,R14                  COPY THE CURRENT TABLE
        L     R1,4(,R8)               GET CURRENT TABLE ADDRESS
        L     R2,0(,R1)               GET CURRENT TABLE LENGTH
        STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
        MODESET MODE=SUP,KEY=ZERO
        MVC   4(4,R8),TEMP            MOVE NEW TABLE ADDRESS
        L     R15,TEMP                GET NEW TABLE ADDRESS
        L     R1,4(,R15)              GET CURRENT # OF ENTRIES
GIDMAXOK EQU  *
        SLL   R1,3                    MULTIPLY BY 8
        LA    R1,8(R1,R15)            ADD IN HEADER LENGTH
        LA    R14,8(,R15)             POINT TO FIRST GROUP NAME
GRPNMCHK EQU  *
        CR    R14,R1                  END OF LIST?
        BNL   GIDADD                  YES - GO ADD
        CLC   0(8,R14),GRPBUF         GROUP NAME ALREADY HERE?
        BE    NOGID                   GO ON
        LA    R14,8(,R14)             POINT TO NEXT GROUP NAME
        B     GRPNMCHK                GO CHECK IT OUT
GIDADD   EQU  *
        MVC   0(8,R1),GRPBUF          MOVE IN NEW GROUP NAME
        L     R1,4(,R15)              GET CURRENT NUMBER OF ENTRIES
        LA    R1,1(,R1)               ADD ONE
        ST    R1,4(,R15)              SAVE NEW COUNT
        B     NOGID                   GO ON
GIDNEW   EQU  *
```

17

```
****************************************************************
*    A NEW GID HAS BEEN ENCOUNTERED. ACQUIRE STORAGE FOR A TABLE      *
*    THAT WILL CONTAIN THE GROUP NAMES THAT HAVE THIS ASSIGNED GID.    *
****************************************************************
         MODESET MODE=PROB,KEY=NZERO
         STORAGE OBTAIN,LENGTH=4096,LOC=ANY
         MVC   0(4,R8),GIDBUF            SAVE THE GID
         ST    R1,4(,R8)                 SAVE THE TABLE ADDRESS
         LR    R0,R1                     COPY THE TABLE ADDRESS
         LR    R14,R1                    AGAIN
         L     R1,=F'4096'               SET THE LENGTH
         XR    R15,R15                   SET FILL BYTE
         MVCL  R0,R14                    INITIALIZE
         L     R1,4(,R8)                 GET TABLE ADDRESS
         MVC   0(4,R1),=F'4096'          SET TABLE LENGTH
         MVC   4(4,R1),=F'1'             SET USERID COUNT
         MVC   8(8,R1),GRPBUF            MOVE IN GROUP NAME
         MODESET MODE=SUP,KEY=ZERO
         B     NOGID                     GO ON
NOGID    EQU   *
         L     R1,TEMPEXTR               GET EXTRACT AREA ADDRESS
         XR    R7,R7                     CLEAR R7
         ICM   R7,B'0111',EXTWLN         LOAD STORAGE LENGTH
         XR    R8,R8                     CLEAR R8
         ICM   R8,B'0001',EXTWSP         LOAD THE SUBPOOL
         STORAGE RELEASE,LENGTH=(R7),ADDR=(R1),SP=(R8)
NOGIDINF EQU   *
         LA    R8,OUTDCBWK               GET DCB ADDRESS
         PUT   (R8),OUTBUF               WRITE THE RECORD
         B     USERLOOP                  CHECK NEXT USER ID
****************************************************************
CHKGRPTB EQU   *
****************************************************************
*    MAKE A PASS THROUGH THE GROUP NAMES TO SEE IF ANY GROUPS HAVE     *
*    BEEN MISSED.                                                      *
****************************************************************
         XC    XUID(4),XUID              CLEAR LENGTH AREA
         MVC   XUID(2),=H'8'             SET LENGTH
         MVC   XUID+4(8),=8C' '          SET STARTING USER ID VALUE
GRPLOOP  EQU   *
         MVC   ROUTWRK4(ROUTLEN4),RACROUT4 MOVE IN RACROUTE MODEL
         RACROUTE REQUEST=EXTRACT,                                  X
               TYPE=EXTRACTN,                                       X
               ENTITYX=XUID,                                        X
               FIELDS=FLDLIST4,                                     X
               RELEASE=1.9.2,                                       X
               WORKA=RACWORK,MF=(E,ROUTWRK4)
         ST    R15,RETCODE               SAVE THE RETURN CODE
         LTR   R15,R15                   DATA EXTRACTED?
         BNZ   DONETBLS                  NO - WE'RE DONE BUILDING TBLS
```

```
              USING EXTWKEA,R1                 EXTRACT WORKAREA ADDRESSABILITY
              XR    R7,R7                       CLEAR R7
              ICM   R7,B'Ø111',EXTWLN           LOAD STORAGE LENGTH
              XR    R8,R8                       CLEAR R8
              ICM   R8,B'ØØØ1',EXTWSP           LOAD THE SUBPOOL
              STORAGE RELEASE,LENGTH=(R7),ADDR=(R1),SP=(R8)
              LA    R15,XUID                    POINT TO GROUP NAME
              XR    R7,R7                       CLEAR R7
              ICM   R7,B'ØØ11',XUID+2           GET GROUP NAME LENGTH
              MVC   GRPBUF(8),=8C' '            CLEAR THE BUFFER
              LTR   R7,R7                       ANY DATA?
              BZ    GRPLOOP                     NO - GET NEXT GROUP
              BCTR  R7,Ø                        REDUCE BY ONE FOR EX
              EX    R7,GRPMVC                   MOVE THE GROUP NAME
*********************************************************************
*    ATTEMPT TO EXTRACT THE OMVS GID INFORMATION FOR THE CURRENT    *
*    GROUP NAME.                                                    *
*********************************************************************
              MVC   ROUTWRK3(ROUTLEN3),RACROUT3 MOVE IN RACROUTE MODEL
              RACROUTE REQUEST=EXTRACT,                                X
                    TYPE=EXTRACT,                                      X
                    ENTITY=GRPBUF,                                     X
                    FIELDS=FLDLIST3,                                   X
                    RELEASE=1.9.2,                                     X
                    WORKA=RACWORK,MF=(E,ROUTWRK3)
              ST    R15,RETCODE                 SAVE THE RETURN CODE
              LTR   R15,R15                     DATA EXTRACTED?
              BNZ   GRPLOOP                     NO - GET NEXT GROUP
              USING EXTWKEA,R1                  EXTRACT WORKAREA ADDRESSABILITY
              ST    R1,TEMPEXTR                 SAVE EXTRACT AREA ADDRESS
              XR    R15,R15                     CLEAR R15
              ICM   R15,B'ØØ11',EXTWOFF         GET OFFSET OF DATA AREA
              AR    R15,R1                      POINT TO GID
              XR    R7,R7                       CLEAR R7
              ICM   R7,B'1111',Ø(R15)           GET GID LENGTH
              LTR   R7,R7                       ANY DATA?
              BNZ   GIDOK2                      YES - GID IS OK
              MVC   WTOWRK1(WTOLEN1),WTOLST1 MOVE IN THE WTO MODEL
              MVC   WTOWRK1+4(38),WTO2          MOVE IN THE WTO TEXT
              MVC   WTOWRK1+34(8),GRPBUF        MOVE IN THE GROUP NAME
              B     NOGID2                      BYPASS GID PROCESSING
GIDOK2        EQU   *
              BCTR  R7,Ø                        REDUCE BY ONE FOR EX
              EX    R7,GIDMVC                   MOVE THE OMVS GID INTO THE BUF
              L     R8,MAINGIDT                 GET TABLE ADDRESS
*********************************************************************
*    CHECK TO SEE IF GID ALREADY IN TABLE                           *
*********************************************************************
              L     R15,GIDTLEN                 GET TABLE LENGTH
              LA    R15,Ø(R15,R8)               SET END OF TABLE ADDRESS
```

```
GIDTLP2  EQU   *
         CLC   Ø(4,R8),GIDBUF        GID ALREADY IN USE?
         BE    GIDMTCH2             YES - USE CONFLICT TABLE
         CLC   Ø(4,R8),=X'FFFFFFFF'  A BLANK ENTRY?
         BE    GIDNEW2              YES - WE HAVE A NEW ENTRY
         LA    R8,8(,R8)            SET TO NEXT ENTRY
         CR    R8,R15               END OF TABLE?
         BL    GIDTLP2              NO - KEEP GOING
**********************************************************************
*    THE CURRENT GID TABLE IS FULL. EXTEND THE TABLE BY 4Ø96 BYTES,   *
*    COPY THE EXISTING TABLE TO THE NEW TABLE, INITIALIZE THE         *
*    ADDITIONAL 4Ø96 BYTES, AND RELEASE THE OLD TABLE STORAGE.        *
**********************************************************************
         L     R7,GIDTLEN           GET GID TABLE LENGTH
         A     R7,=F'4Ø96'          ADD 4096
         MODESET MODE=PROB,KEY=NZERO
         STORAGE OBTAIN,LENGTH=(R7)  GET A NEW BLOCK
         ST    R1,TEMP              SAVE THE ADDRESS
         LR    RØ,R1                COPY THE ADDRESS
         L     R14,MAINGIDT         GET OLD TABLE ADDRESS
         L     R1,GIDTLEN           GET TABLE LENGTH
         LR    R15,R1               COPY THE LENGTH
         MVCL  RØ,R14               COPY THE OLD TABLE
         L     R1,TEMP              GET NEW TABLE ADDRESS
         L     R15,GIDTLEN          GET OLD TABLE LENGTH
         LA    R1,Ø(R15,R1)         POINT PAST CURRENT DATA
         LR    R8,R1                SAVE THIS ADDRESS
         L     R15,=F'512'          SET A LOOP COUNT
GIDINIT3 EQU   *
         MVC   Ø(8,R1),=X'FFFFFFFFØØØØØØØØ' INITIALIZE THE ENTRY
         LA    R1,8(,R1)            POINT TO NEXT ENTRY
         BCT   R15,GIDINIT3         BRANCH IF MORE
         L     R9,GIDTLEN           GET OLD TABLE LENGTH
         L     R1,MAINGIDT          GET OLD TABLE ADDRESS
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1)
         MODESET MODE=SUP,KEY=ZERO
         MVC   MAINGIDT(4),TEMP     COPY IN NEW TABLE ADDRESS
         L     R9,GIDTLEN           GET OLD TABLE LENGTH
         A     R9,=F'4Ø96'          ADD 4096
         ST    R9,GIDTLEN           SAVE NEW LENGTH
         B     GIDNEW2              GET GROUP TABLE FOR NEW GID
GIDMTCH2 EQU   *
         L     R15,4(,R8)           GET CONFLICT TABLE ADDRESS
         L     R1,4(,R15)           GET CURRENT NUMBER OF ENTRIES
         L     R2,Ø(,R15)           GET TABLE LENGTH
         SRL   R2,3                 DIVIDE BY 8
         BCTR  R2,Ø                 SUBTRACT 1
         CR    R1,R2                MAX ENTRIES?
         BL    GIDMAXOK2            NO - WILL FIT IN CURRENT TABLE
```

```
****************************************************************
*    THE CURRENT GROUP NAME CONFLICT TABLE FOR THIS GID IS FULL.      *
*    EXTEND THE TABLE BY 4096 BYTES, COPY THE EXISTING TABLE TO THE    *
*    NEW TABLE, RELEASE THE OLD TABLE STORAGE, AND ADD THE NEW GROUP   *
*    NAME INTO THE EXTENDED TABLE.                                     *
****************************************************************
          LA    R2,1(,R2)                ADD ONE
          SLL   R2,3                     MULTIPLY BY 8
          A     R2,=F'4096'              ADD 4096
          MODESET MODE=PROB,KEY=NZERO
          STORAGE OBTAIN,LENGTH=(R2)
          ST    R1,TEMP                  SAVE THE ADDRESS
          LR    R0,R1                    COPY IT
          L     R14,4(,R8)               GET CURRENT TABLE ADDRESS
          L     R1,0(,R14)               GET CURRENT TABLE LENGTH
          LR    R15,R1                   COPY IT
          MVCL  R0,R14                   COPY THE CURRENT TABLE
          L     R1,4(,R8)                GET CURRENT TABLE ADDRESS
          L     R2,0(,R1)                GET CURRENT TABLE LENGTH
          STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
          MODESET MODE=SUP,KEY=ZERO
          MVC   4(4,R8),TEMP             MOVE NEW TABLE ADDRESS
          L     R15,TEMP                 GET NEW TABLE ADDRESS
          L     R1,4(,R15)               GET CURRENT # OF ENTRIES
GIDMAXOK2 EQU   *
          SLL   R1,3                     MULTIPLY BY 8
          LA    R1,8(R1,R15)             ADD IN HEADER LENGTH
          LA    R14,8(,R15)              POINT TO FIRST GROUP NAME
GRPNMCH2  EQU   *
          CR    R14,R1                   END OF LIST?
          BNL   GIDADD2                  YES - GO ADD
          CLC   0(8,R14),GRPBUF          GROUP NAME ALREADY HERE?
          BE    NOGID2                   GO ON
          LA    R14,8(,R14)              POINT TO NEXT GROUP NAME
          B     GRPNMCH2                 GO CHECK IT OUT
GIDADD2   EQU   *
          MVC   0(8,R1),GRPBUF           MOVE IN NEW GROUP NAME
          L     R1,4(,R15)               GET CURRENT NUMBER OF ENTRIES
          LA    R1,1(,R1)                ADD ONE
          ST    R1,4(,R15)               SAVE NEW COUNT
          B     NOGID2                   GO ON
GIDNEW2   EQU   *
****************************************************************
*    A NEW GID HAS BEEN ENCOUNTERED. ACQUIRE STORAGE FOR A TABLE       *
*    THAT WILL CONTAIN THE GROUP NAMES THAT HAVE THIS ASSIGNED GID.    *
****************************************************************
          MODESET MODE=PROB,KEY=NZERO
          STORAGE OBTAIN,LENGTH=4096,LOC=ANY
          MVC   0(4,R8),GIDBUF           SAVE THE GID
          ST    R1,4(,R8)                SAVE THE TABLE ADDRESS
```

```
               LR    RØ,R1                     COPY THE TABLE ADDRESS
               LR    R14,R1                    AGAIN
               L     R1,=F'4Ø96'               SET THE LENGTH
               XR    R15,R15                   SET FILL BYTE
               MVCL  RØ,R14                    INITIALIZE
               L     R1,4(,R8)                 GET TABLE ADDRESS
               MVC   Ø(4,R1),=F'4Ø96'          SET TABLE LENGTH
               MVC   4(4,R1),=F'1'             SET USERID COUNT
               MVC   8(8,R1),GRPBUF            MOVE IN GROUP NAME
               MODESET MODE=SUP,KEY=ZERO
               B     NOGID2                    GO ON
NOGID2         EQU   *
               L     R1,TEMPEXTR               GET EXTRACT AREA ADDRESS
               XR    R7,R7                     CLEAR R7
               ICM   R7,B'Ø111',EXTWLN         LOAD STORAGE LENGTH
               XR    R8,R8                     CLEAR R8
               ICM   R8,B'ØØØ1',EXTWSP         LOAD THE SUBPOOL
               STORAGE RELEASE,LENGTH=(R7),ADDR=(R1),SP=(R8)
               B     GRPLOOP                   CHECK NEXT GROUP
*********************************************************************
DONETBLS       EQU   *
*********************************************************************
               MODESET MODE=PROB,KEY=NZERO
*********************************************************************
               MVC   XREFDCBW(XREFDCBL),UGIDXREF COPY THE DCB
               LA    R9,XREFDCBW              GET DCB ADDRESS
               OI    OPENLST,X'8Ø'            SET PARM BIT ON
               OPEN  ((R9),OUTPUT),MODE=31,MF=(E,OPENLST) OPEN THE DCB
               PUT   (R9),OUTHDR1            OUTPUT THE HEADER
               PUT   (R9),BLANKS            OUTPUT THE HEADER
*********************************************************************
*    CLEAN UP CONFLICT TABLES                                      *
*********************************************************************
               L     R8,MAINUIDT             GET TABLE ADDRESS
               L     R7,UIDTLEN              GET TABLE LENGTH
               SRL   R7,3                    DIVIDE BY 8 TO GET # OF ENTRIES
*********************************************************************
*    SORT UIDS IN MAIN TABLE                                       *
*********************************************************************
               LR    R2,R8                   GET TABLE START
               L     R3,UIDTLEN             GET TABLE LENGTH
               LA    R3,Ø(R3,R2)            GET TABLE END
UIDSORT1       EQU   *
               CR    R2,R3                   END OF TABLE?
               BNL   UIDFDUMP               YES - GO PROCESS TABLE ENTRIES
               CLC   Ø(4,R2),=X'FFFFFFFF'    END OF TABLE?
               BE    UIDFDUMP               YES - GO PROCESS TABLE ENTRIES
               LA    R4,8(,R2)              POINT TO COMPARE ENTRY
               CR    R4,R3                   END OF TABLE?
               BNL   UIDFDUMP               YES - GO PROCESS TABLE ENTRIES
```

```
UIDSORT2 EQU   *
         CLC   Ø(4,R2),Ø(R4)          UID GREATER?
         BL    NXTUIDIN               NO - GET NEXT INNER ENTRY
         MVC   DBL2(8),Ø(R2)          SAVE CURRENT ENTRY
         MVC   Ø(8,R2),Ø(R4)          MOVE IN NEW LOW
         MVC   Ø(8,R4),DBL2           COPY OLD CURRENT
NXTUIDIN EQU   *
         LA    R4,8(,R4)              POINT TO NEXT CHECK ENTRY
         CR    R4,R3                  END OF TABLE?
         BL    UIDSORT2               NO - DO NEW INNER CHECK
         LA    R2,8(,R2)              POINT TO NEW LOW ENTRY
         B     UIDSORT1               START AGAIN
UIDFDUMP EQU   *
*********************************************************************
         TM    FLAG,BLDVSAM           BUILD VSAM FILE?
         BZ    UIDTFREE               NO - FREE THE TABLES
*********************************************************************
*   OPEN THE UIDFILE VSAM DATASET AND WRITE EACH UID RECORD.       *
*********************************************************************
         MVC   ACBWRK1(ACBLEN1),UIDFILE MOVE IN THE ACB MODEL
         LA    R9,ACBWRK1             GET THE ACB ADDRESS
         OI    OPENLST,X'8Ø'          SET PARM BIT ON
         OPEN  ((R9)),MODE=31,MF=(E,OPENLST)
         MVC   RPLWRK1(RPLLEN1),RPL1  GET RPL MODEL
         ST    R9,RPLWRK1+X'18'       SAVE ACB ADDRESS
         L     R8,MAINUIDT            GET TABLE ADDRESS
         L     R7,UIDTLEN             GET TABLE LENGTH
         SRL   R7,3                   DIVIDE BY 8 TO GET # OF ENTRIES
         LR    R2,R8                  GET STARTING ADDRESS
UIDFILLP EQU   *
         CLC   Ø(4,R2),=X'FFFFFFFF'   END OF TABLE?
         BE    CLUIDFIL               YES - CLOSE UID FILE
         L     R3,4(,R2)              GET UID ENTRY TABLE ADDRESS
         L     R4,4(,R3)              GET USERID COUNT
         LR    R5,R4                  SAVE IT
         L     R6,Ø(,R3)              GET LENGTH OF AREA
         S     R6,=F'4'               SUBTRACT 4
         ST    R6,RPLWRK1+X'34'       SAVE AREA LENGTH IN RPL
         LA    R6,4(,R3)              GET RECORD ADDRESS
         ST    R6,RPLWRK1+X'2Ø'       SAVE RECORD ADDRESS IN RPL
         SLL   R5,3                   GET RECORD ...
         LA    R5,4(,R5)                  LENGTH
         ST    R5,RPLWRK1+X'3Ø'       SAVE RECORD LENGTH IN RPL
         MVC   4(4,R3),Ø(R2)          MOVE IN THE UID
         PUT   RPL=RPLWRK1            WRITE THE RECORD
         LTR   R15,R15                WRITE OK?
         BZ    PUTOK                  YES - GO ON
         ABEND 98                     ABEND
PUTOK    EQU   *
         ST    R4,4(,R3)              REPLACE THE USERID COUNT
```

23

```
        LA    R2,8(,R2)              POINT TO NEXT TABLE ENTRY
        BCT   R7,UIDFILLP            IF MORE, GO PROCESS
CLUIDFIL EQU   *
        LA    R9,ACBWRK1             GET THE ACB ADDRESS
        OI    CLOSELST,X'8Ø'         SET PARM BIT ON
        CLOSE ((R9)),MODE=31,MF=(E,CLOSELST)
        LA    R9,XREFDCBW            GET DCB ADDRESS
**********************************************************************
UIDTFREE EQU   *
**********************************************************************
*   PRODUCE THE USERID CONFLICT LIST FOR THIS UID AND THEN RELEASE   *
*   THE CORRESPONDING USERID TABLE STORAGE.                          *
**********************************************************************
        CLC   Ø(4,R8),=X'FFFFFFFF'   END OF VALID ENTRIES?
        BE    RLSEUIDT               YES - FREE MAIN UID TABLE
        L     R1,Ø(,R8)              GET UID VALUE
        CVD   R1,DBL2                CONVERT TO DECIMAL
        UNPK  DBL1(16),DBL2(8)       UNPACK
        OC    DBL1(16),=16X'FØ'      MAKE IT READABLE
        MVC   OUTBUF(8Ø),BLANKS      INITIALIZE
        MVC   OUTBUF(4Ø),=C'UID XXXXXXXXX USED BY FOLLOWING USERIDS'
        MVC   OUTBUF+4(1Ø),DBL1+6    MOVE IN THE DECIMAL NUMBER
        PUT   (R9),OUTBUF            WRITE THE RECORD
        L     R5,4(,R8)              GET TABLE ADDRESS
        L     R4,4(,R5)              GET NUMBER OF ENTRIES
        LA    R5,8(,R5)              POINT TO FIRST UID
        MVC   OUTBUF(8Ø),BLANKS      INITIALIZE BUFFER
UIDOUTLP EQU   *
        MVC   OUTBUF+4(8),Ø(R5)      MOVE IN THE USERID
        PUT   (R9),OUTBUF            WRITE THE RECORD
        LA    R5,8(,R5)              POINT TO NEXT ENTRY
        BCT   R4,UIDOUTLP            BRANCH FOR MORE
        PUT   (R9),BLANKS            WRITE A BLANK RECORD
        L     R1,4(,R8)              GET TABLE ADDRESS
        L     R2,Ø(,R1)              GET TABLE LENGTH
        STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
        LA    R8,8(,R8)              POINT TO NEXT UID ENTRY
        BCT   R7,UIDTFREE            PROCESS IF MORE
RLSEUIDT EQU   *
**********************************************************************
*   RELEASE THE STORAGE FOR THE UID TABLE.                           *
**********************************************************************
        L     R1,MAINUIDT            GET TABLE ADDRESS
        L     R2,UIDTLEN             GET TABLE LENGTH
        STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
**********************************************************************
*   PROCESS GID TABLE                                                *
**********************************************************************
        PUT   (R9),BLANKS            WRITE A BLANK RECORD
        PUT   (R9),BLANKS            WRITE A BLANK RECORD
```

```
        PUT   (R9),OUTHDR2              OUTPUT THE HEADER
        PUT   (R9),BLANKS               OUTPUT THE HEADER
        L     R8,MAINGIDT               GET TABLE ADDRESS
        L     R7,GIDTLEN                GET TABLE LENGTH
        SRL   R7,3                      DIVIDE BY 8 TO GET # OF ENTRIES
**********************************************************************
*     SORT GIDS IN MAIN TABLE                                        *
**********************************************************************
        LR    R2,R8                     GET TABLE START
        L     R3,GIDTLEN                GET TABLE LENGTH
        LA    R3,Ø(R3,R2)               GET TABLE END
GIDSORT1 EQU  *
        CR    R2,R3                     END OF TABLE?
        BNL   GIDFDUMP                  YES - GO PROCESS TABLE ENTRIES
        CLC   Ø(4,R2),=X'FFFFFFFF'      END OF TABLE?
        BE    GIDFDUMP                  YES - GO PROCESS TABLE ENTRIES
        LA    R4,8(,R2)                 POINT TO COMPARE ENTRY
        CR    R4,R3                     END OF TABLE?
        BNL   GIDFDUMP                  YES - GO PROCESS TABLE ENTRIES
GIDSORT2 EQU  *
        CLC   Ø(4,R2),Ø(R4)             GID GREATER?
        BL    NXTGIDIN                  NO - GET NEXT INNER ENTRY
        MVC   DBL2(8),Ø(R2)             SAVE CURRENT ENTRY
        MVC   Ø(8,R2),Ø(R4)             MOVE IN NEW LOW
        MVC   Ø(8,R4),DBL2              COPY OLD CURRENT
NXTGIDIN EQU  *
        LA    R4,8(,R4)                 POINT TO NEXT CHECK ENTRY
        CR    R4,R3                     END OF TABLE?
        BL    GIDSORT2                  NO - DO NEW INNER CHECK
        LA    R2,8(,R2)                 POINT TO NEW LOW ENTRY
        B     GIDSORT1                  START AGAIN
GIDFDUMP EQU  *
**********************************************************************
        TM    FLAG,BLDVSAM              BUILD VSAM FILE?
        BZ    GIDTFREE                  NO - FREE THE TABLES
**********************************************************************
*     OPEN THE UIDFILE VSAM DATASET AND WRITE EACH UID RECORD.       *
**********************************************************************
        MVC   ACBWRK2(ACBLEN2),GIDFILE MOVE IN THE ACB MODEL
        LA    R9,ACBWRK2                GET THE ACB ADDRESS
        OI    OPENLST,X'8Ø'             SET PARM BIT ON
        OPEN  ((R9)),MODE=31,MF=(E,OPENLST)
        MVC   RPLWRK2(RPLLEN2),RPL2     GET RPL MODEL
        ST    R9,RPLWRK2+X'18'          SAVE ACB ADDRESS
        L     R8,MAINGIDT               GET TABLE ADDRESS
        L     R7,GIDTLEN                GET TABLE LENGTH
        SRL   R7,3                      DIVIDE BY 8 TO GET # OF ENTRIES
        LR    R2,R8                     GET STARTING ADDRESS
GIDFILLP EQU  *
        CLC   Ø(4,R2),=X'FFFFFFFF'      END OF TABLE?
```

25

```
           BE      CLGIDFIL                YES - CLOSE GID FILE
           L       R3,4(,R2)               GET GID ENTRY TABLE ADDRESS
           L       R4,4(,R3)               GET GROUP NAME COUNT
           LR      R5,R4                   SAVE IT
           L       R6,Ø(,R3)               GET LENGTH OF AREA
           S       R6,=F'4'                SUBTRACT 4
           ST      R6,RPLWRK2+X'34'        SAVE AREA LENGTH IN RPL
           LA      R6,4(,R3)               GET RECORD ADDRESS
           ST      R6,RPLWRK2+X'2Ø'        SAVE RECORD ADDRESS IN RPL
           SLL     R5,3                    GET RECORD ...
           LA      R5,4(,R5)                  LENGTH
           ST      R5,RPLWRK2+X'3Ø'        SAVE RECORD LENGTH IN RPL
           MVC     4(4,R3),Ø(R2)           MOVE IN THE GID
           PUT     RPL=RPLWRK2             WRITE THE RECORD
           LTR     R15,R15                 WRITE OK?
           BZ      PUTOK2                  YES - GO ON
           ABEND 99                        ABEND
PUTOK2     EQU     *
           ST      R4,4(,R3)               REPLACE THE GROUP COUNT
           LA      R2,8(,R2)               POINT TO NEXT TABLE ENTRY
           BCT     R7,GIDFILLP             IF MORE, GO PROCESS
CLGIDFIL   EQU     *
           LA      R9,ACBWRK2              GET THE ACB ADDRESS
           OI      CLOSELST,X'8Ø'          SET PARM BIT ON
           CLOSE ((R9)),MODE=31,MF=(E,CLOSELST)
           LA      R9,XREFDCBW             GET DCB ADDRESS
********************************************************************
GIDTFREE   EQU     *
********************************************************************
*    PRODUCE THE GROUP NAME CONFLICT LIST FOR THIS GID AND THEN    *
*    RELEASE THE CORRESPONDING GROUP NAME TABLE STORAGE.           *
********************************************************************
           CLC     Ø(4,R8),=X'FFFFFFFF'    END OF VALID ENTRIES?
           BE      RLSEGIDT                YES - FREE MAIN GID TABLE
           L       R1,Ø(,R8)               GET UID VALUE
           CVD     R1,DBL2                 CONVERT TO DECIMAL
           UNPK    DBL1(16),DBL2(8)        UNPACK
           OC      DBL1(16),=16X'FØ'       MAKE IT READABLE
           MVC     OUTBUF(8Ø),BLANKS       INITIALIZE
           MVC     OUTBUF(4Ø),=C'GID XXXXXXXXX USED BY FOLLOWING GROUPS '
           MVC     OUTBUF+4(1Ø),DBL1+6     MOVE IN THE DECIMAL NUMBER
           PUT     (R9),OUTBUF             WRITE THE RECORD
           L       R5,4(,R8)               GET TABLE ADDRESS
           L       R4,4(,R5)               GET NUMBER OF ENTRIES
           LA      R5,8(,R5)               POINT TO FIRST UID
           MVC     OUTBUF(8Ø),BLANKS       INITIALIZE BUFFER
GIDOUTLP   EQU     *
           MVC     OUTBUF+4(8),Ø(R5)       MOVE IN THE USERID
           PUT     (R9),OUTBUF             WRITE THE RECORD
           LA      R5,8(,R5)               POINT TO NEXT ENTRY
```

```
              BCT    R4,GIDOUTLP            BRANCH FOR MORE
              PUT    (R9),BLANKS            WRITE A BLANK RECORD
              L      R1,4(,R8)              GET TABLE ADDRESS
              L      R2,0(,R1)              GET TABLE LENGTH
              STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
              LA     R8,8(,R8)              POINT TO NEXT UID ENTRY
              BCT    R7,GIDTFREE            PROCESS IF MORE
RLSEGIDT EQU  *
**********************************************************************
*    RELEASE THE STORAGE FOR THE GID TABLE.                         *
**********************************************************************
              L      R1,MAINGIDT            GET TABLE ADDRESS
              L      R2,GIDTLEN             GET TABLE LENGTH
              STORAGE RELEASE,LENGTH=(R2),ADDR=(R1)
              LA     R9,XREFDCBW            GET DCB ADDRESS
              OI     CLOSELST,X'80'         SET PARM BIT ON
              CLOSE  ((R9),FREE),MODE=31,MF=(E,CLOSELST) OPEN THE DCB
**********************************************************************
RETURN   EQU  *
**********************************************************************
              LA     R8,OUTDCBWK            GET DCB ADDRESS
              OI     CLOSELST,X'80'         SET PARM BIT ON
              CLOSE  ((R8),FREE),MODE=31,MF=(E,CLOSELST) OPEN THE DCB
              L      R2,4(,R13)             GET OLD SAVEAREA ADDRESS
              LR     R1,R13                 GET NEW SAVEAREA ADDRESS
              STORAGE RELEASE,LENGTH=WORKLEN,ADDR=(R1)
              LR     R13,R2                 COPY OLD SAVEAREA ADDRESS
              LM     R14,R12,12(R13)        RELOAD THE REGISTERS
              XR     R15,R15                SET RETURN CODE
              BR     R14                    RETURN
**********************************************************************
*                                                                  *
*    EXECUTED INSTRUCTIONS                                          *
*                                                                  *
**********************************************************************
NAMEMVC  MVC  NAMEBUF(*-*),4(R15)         SAVE PROGRAMMER NAME
GRPMVC   MVC  GRPBUF(*-*),4(R15)          SAVE GROUP NAME
UIDMVC   MVC  UIDBUF(*-*),4(R15)          SAVE UID
GIDMVC   MVC  GIDBUF(*-*),4(R15)          SAVE GID
**********************************************************************
*                                                                  *
*    CONSTANTS                                                      *
*                                                                  *
**********************************************************************
OUTPUT   DCB  MACRF=(PM),DDNAME=OUTPUT,DSORG=PS,LRECL=80
OUTDCBL  EQU  *-OUTPUT
OUTHDR   DC   CL40'USERID   NAME                        OMVS UID '
         DC   CL40' GROUP    OMVS GID                            '
BLANKS   DC   80C' '
OUTHDR1  DC   CL80'UID CROSS-REFERENCE AND CONFLICT REPORT '
```

```
OUTHDR2  DC    CL80'GID CROSS-REFERENCE AND CONFLICT REPORT '
UGIDXREF DCB   MACRF=(PM),DDNAME=UGIDXREF,DSORG=PS,LRECL=80
XREFDCBL EQU   *-UGIDXREF
**********************************************************************
FLDLIST1 DC    F'2'
         DC    CL8'PGMRNAME'
         DC    CL8'DFLTGRP '
**********************************************************************
FLDLIST2 DC    F'1'
         DC    CL8'UID     '
**********************************************************************
FLDLIST3 DC    F'1'
         DC    CL8'GID     '
**********************************************************************
FLDLIST4 DC    F'1'
         DC    CL8'AUTHOR  '
**********************************************************************
RACROUT1 RACROUTE REQUEST=EXTRACT,                                   X
              TYPE=EXTRACTN,                                         X
              CLASS='USER',                                          X
              RELEASE=1.9.2,                                         X
              MF=L
ROUTLEN1 EQU   *-RACROUT1
**********************************************************************
RACROUT2 RACROUTE REQUEST=EXTRACT,                                   X
              TYPE=EXTRACT,                                          X
              CLASS='USER',                                          X
              SEGMENT='OMVS',                                        X
              RELEASE=1.9.2,                                         X
              MF=L
ROUTLEN2 EQU   *-RACROUT2
**********************************************************************
RACROUT3 RACROUTE REQUEST=EXTRACT,                                   X
              TYPE=EXTRACT,                                          X
              CLASS='GROUP',                                         X
              SEGMENT='OMVS',                                        X
              RELEASE=1.9.2,                                         X
              MF=L
ROUTLEN3 EQU   *-RACROUT3
**********************************************************************
RACROUT4 RACROUTE REQUEST=EXTRACT,                                   X
              TYPE=EXTRACTN,                                         X
              CLASS='GROUP',                                         X
              RELEASE=1.9.2,                                         X
              MF=L
ROUTLEN4 EQU   *-RACROUT4
**********************************************************************
UIDFILE  ACB   DDNAME=UIDFILE,                                       X
              MACRF=(KEY,OUT)
ACBLEN1  EQU   *-UIDFILE
**********************************************************************
```

```
GIDFILE  ACB   DDNAME=GIDFILE,                                          X
               MACRF=(KEY,OUT)
ACBLEN2  EQU   *-GIDFILE
*********************************************************************
RPL1     RPL   ACB=UIDFILE,                                            X
               AREA=*-*,                                               X
               AREALEN=1,                                              X
               OPTCD=(KEY,NUP,MVE),                                    X
               RECLEN=1
RPLLEN1  EQU   *-RPL1
*********************************************************************
RPL2     RPL   ACB=GIDFILE,                                            X
               AREA=*-*,                                               X
               AREALEN=1,                                              X
               OPTCD=(KEY,NUP,MVE),                                    X
               RECLEN=1
RPLLEN2  EQU   *-RPL2
*********************************************************************
WTOLST1  WTO   '                                                       X
                                                                     ',X
               ROUTCDE=(1),DESC=(6),MF=L
WTOLEN1  EQU   *-WTOLST1
*********************************************************************
WTO1     DC    C'UIDGIDST - NO UID LOCATED FOR XXXXXXXX'
WTO2     DC    C'UIDGIDST - NO GID LOCATED FOR XXXXXXXX'
*********************************************************************
         LTORG
*********************************************************************
*                                                                   *
*    WORKING STORAGE DSECT                                          *
*                                                                   *
*********************************************************************
WORKAREA DSECT
SAVEAREA DS    18F
RETCODE  DS    F
MAINUIDT DS    F
UIDTLEN  DS    F
MAINGIDT DS    F
GIDTLEN  DS    F
TEMP     DS    F
TEMPEXTR DS    F
OPENLST  OPEN  (,),MODE=31,MF=L
CLOSELST CLOSE (,),MODE=31,MF=L
OUTDCBWK DS    CL(OUTDCBL)
XREFDCBW DS    CL(XREFDCBL)
RACWORK  DS    ØD,CL(512)
XUID     DS    3F
NAMEBUF  DS    CL8Ø
OUTBUF   DS    CL8Ø
GRPBUF   DS    CL8
UIDBUF   DS    F
```

```
GIDBUF   DS    F
ROUTWRK1 DS    ØD,CL(ROUTLEN1)
ROUTWRK2 DS    ØD,CL(ROUTLEN2)
ROUTWRK3 DS    ØD,CL(ROUTLEN3)
ROUTWRK4 DS    ØD,CL(ROUTLEN4)
ACBWRK1  DS    ØD,CL(ACBLEN1)
RPLWRK1  DS    ØD,CL(RPLLEN1)
ACBWRK2  DS    ØD,CL(ACBLEN2)
RPLWRK2  DS    ØD,CL(RPLLEN2)
WTOWRK1  DS    ØD,CL(WTOLEN1)
DBL1     DS    2D
DBL2     DS    2D
FLAG     DS    CL1
BLDVSAM  EQU   X'8Ø'
WORKLEN  EQU   *-WORKAREA
         $REQU
         IRRPRXTW
         END
```

Remote security revisited

In my first 'Remote Security' article (*RACF Update* 22, November 2000, pp 28-32), I discussed the danger of relying purely on VPN and RACF to secure remote workstations connecting to mainframes. I pointed out that if the connection uses commercial high-speed Internet services, whether they be xDSL or cable modem, the security of the Internet connection itself must be investigated. Otherwise, hackers may have easy access to the workstation any time they wish to gain control.

Although the scenario I described, hackers accessing mainframes, has not made the news, Distributed Denial of Service (DDoS) attacks certainly have. And it's now becoming clear that remote workstations running Windows 2000 are especially appealing targets to hackers because of Win 2000's complete implementation of Unix Sockets. This contrasts with the Win 3.1/NT/9x/Me implementations which restrict most DDoS attacks to those that can be dealt with using an ISP's packet filters.

DISTRIBUTED DENIAL OF SERVICE (DDoS)

For any of this to make sense, a general understanding is required of how a DDoS attack works. From the hacker's perspective, the basic concept is to personally monopolize the full bandwidth from the Internet to a given Web site, effectively shutting it down by keeping everybody else off the site. The traffic is generally ping requests or IP packets with errors in them, since these can be hacker software-generated universally for any Web site, without anyone sitting down to figure out a valid application function for each Web site.

There are two problems that hackers have to overcome to make this work:

• The hacker's bandwidth to the Internet has to exceed the bandwidth of the Web site to the Internet.

• The traffic must not be easily traceable to the hacker.

One solution solves both problems: secretly take over control of other people's computers and have them all simultaneously flood the same Web site with large volumes of traffic.

LEGAL LIABILITY

When you first attempt to address this problem, be prepared for the following response: "How can I justify spending the organization's money to prevent a hacker from using one of our remote workstations to help cripple someone else's Web site?" In other words, what's the bottom line benefit?

The easiest financial exposure to explain to management is Liability. Although a DDoS attack is not easily traceable to the hacker who controls it, it is easily traceable to any of your organization's workstations that are used to perpetrate it. Even an average lawyer could easily win a liability case against your organization for damages done to the owner of the Web site being attacked. Especially since it has been more than a year since DDoS attacks made headlines.

If it ever went to court, it would be hard to argue that your security people had not had enough time to prevent such an attack. Excuses are likely to fall on deaf ears, even when they are the truth: all attention was focused on preparing for an attack on your Web site, rather than preventing your organization's workstation from being used in an attack on someone else.

SECURITY SITE UNDER ATTACK

My previous 'Remote Security' article discussed a Gibson Research Web site that allows you to test your Internet connection for a few of the most common security holes. That site also explains how to plug any holes it finds. Unfortunately, the site has found itself the target of DDoS attacks, and has had to take its free Internet security test service down until it can be resolved.

At the beginning of June, it had been under siege for more than a month, with no end in sight. A 13 year old hacker got the software tools and a list of vulnerable workstations from more experienced hackers,

and was able to assemble and remotely control a worldwide collection of 474 high-speed Internet-attached workstations to conduct the attacks. If the site is up by the time you are reading this, a 28-page document details the events as they unfolded. It is available as either one long Web page or an Adobe Acrobat .pdf file at

http://grc.com/dos/grcdos.htm

http://media.grc.com:8080/files/grcdos.pdf

HIBERNATE, NOT STANDBY

A simpler approach, and one that should be implemented as well, is to disconnect from the Internet when the remote workstation(s) is/are not being used, especially for long periods, like weekends and overnight. This can be as simple as having staff turn off the modem at night when they go home, but a more automatic approach uses the Hibernate function of Windows 2000, Me, and XP. Once enabled in power management:

- From the Start button, click Settings from the pop-up menu.

- Click Control Panel.

- Double click the Power Options icon.

- Click the Hibernate tab.

- Click to check the Enable hibernation support box.

- Push the Apply button.

Set it to occur automatically after a defined period of non-use of the workstation:

- Click the Power Schemes tab.

- Select an appropriate amount of time, such as After 15 minutes, from the drop-down selection in the System hibernates field.

- Click the Save As button.

- Click the OK button in the Save Scheme dialogue box.

- Click the OK button in the Power Options Properties dialogue box.

Note that these instructions are specific to Windows 2000, but Windows Me and XP should be similar.

The Hibernate option now appears as an option in the Shut Down Windows dialogue box that you get when you select Shut Down... from the Start button menu. Users will find that both starting up after hibernation, as well as the hibernation itself, are much faster than a normal shut down and start up.

If this sounds like the Standby mode introduced with Windows 98, you'll be pleasantly surprised at the improvements. Hibernation actually powers the machine completely off, after storing the entire contents of real memory (RAM) to the hard drive. Hibernation also has been free of the 'failure to wake up' problems that plague Standby in Windows 2000.

ON ALL THE TIME

Of course, none of this is possible if the organization requires that all workstations be left up and running all night and over the weekend to allow the distribution of software and other updates.

The vulnerability to hacker access may well be argument enough to make remote workstations an exception to the 'always on' rule. But another, better solution, is to have a firewall, which allows remote workstations to be left constantly connected to the Internet. The first firewalls were large expensive stand-alone computer systems, often bigger than any of the network of systems they were protecting. Today, firewalls need not be large, expensive, or even hardware-based solutions, although there are convincing arguments in favour of hardware versus software solutions. Firewalls are now being actively marketed for SOHO (Small Office, Home Office) and other small clusters of workstations such as the remote situation we've been discussing.

If you find yourself saying, "I thought my ISP protected me behind its firewall," you're not alone in that misunderstanding. Most ISPs are

behind firewalls, but their customers are not. The firewalls are used only to protect the ISP's internal systems, not the customers'. All customer Internet connections are provided outside the firewall.

Many ISPs, of course, also offer managed services. That scenario often involves running shared and stand-alone Web servers for their customers, typically behind a firewall. But it still never hurts to ask your ISP if it will offer you its normal high-speed Internet connections, but protected behind its firewall.

SMALL FIREWALLS

The best known firewalls for small networks or single workstation protection are often known as personal firewalls. McAfee even uses the name Personal Firewall for a component of popular Clinic on-line applications. They are software-based and inexpensive. In fact, one often cited as very secure is free for download from Zone Labs at:

http://www.zonelabs.com/za_download_1.htm

At a recent firewall seminar, a technical representative from WatchGuard stated that the problem with software-only firewalls is that they can be compromised by directly attacking the operating system they're being run on, effectively circumventing the firewall completely. As you might guess, WatchGuard manufacturers and markets hardware firewalls, which are really just tamper-proof sealed stand-alone computers running their own software firewall on a security-hardened operating system. In WatchGuard's case, the operating system it has hardened, mainly in the area of the kernel, is Linux.

Elsewhere in this issue, we take a look at what it takes to get a WatchGuard SOHO firewall up and running, and what it's like to live with on a day-to-day basis. As its name implies, the SOHO is WatchGuard's smallest and least expensive firewall, intended for a small number of workstations (or even just one), directly connected to the Internet.

When asked, the WatchGuard technical representative was quick to point out that none of this firewall protection reduces the need for a

proven, regularly and frequently updated anti-virus software product on every workstation. Although I have not done an in-depth analysis, anti-virus software appears to protect against all of the methods used by hackers to gain control of workstations for DDoS attacks. Each method that I've read about includes at least one step involving traditional virus infections, such as e-mail attachments and trojan horses: programs containing other programs that do the actual damage.

Jon Pearkins
(Canada)   © Xephon 2001

Installing a WatchGuard SOHO firewall

Along with VPN, RACF, and anti-virus software, firewalls promise to add the final piece of the puzzle that will address the need for security on remotely-attached workstations – whether those workstations are in an employee's home, a satellite office, a very small branch, a customer's location, or any other off-site location. This article describes the process of installing one such firewall.

The stated goal of the WatchGuard SOHO is to be simple yet effective – installation without configuration, and no knowledge of IP port numbers and protocols required.

My initial experience indicates that WatchGuard is well on its way to meeting this goal. There really is no configuration necessary. And updates are very simple to install, especially for anyone who has used a Windows software installation wizard. Installation and registration problems can be attributed primarily to faulty documentation, but the newly-published *Quick Start and User's Guide* could resolve those issues so long as they completely replace past materials. Care is still required, however – even a small oversight in following the instructions can lead to failure, and having to restart from scratch.

Finally, the concept of the firewall presenting Web pages for monitoring and control is brilliant, in terms of ease of use. I was also impressed with the seamless integration between the Web pages generated internally within the firewall and WatchGuard's own Web site. Unless you watch the URLs, you never know where you are, as the Web pages have a very similar look and feel.

Access to the firewall is with a single numeric IP. Type 192.168.111.1 into your Web browser, and you get the Configuration menu. Even the Event Log is displayed as one long Web page, with colour coding to highlight events of interest.

PRODUCT OVERVIEW

WatchGuard SOHO is a hardware firewall – a stand-alone device somewhat resembling a 15-year-old external modem for a PC. SOHO is the recently coined acronym for Small Office/Home Office.

There are two models of the SOHO, one with IPSec Virtual Private Network (VPN) and one without. Both support four workstations directly with a built-in Ethernet hub, a total of 10 with a hub of your choice, with licence upgrades available to 25 and 50 workstations. List prices are $450/£320, $600/£425 for the SOHO|tc with VPN. An upgrade to 25 users costs $200/£140, and to 50 $450/£320.

All company literature emphasizes LiveSecurity Service as a major differentiator with the competitors. Each firewall includes one year of this combination of firmware updates, 24/7 technical support, and information, including virus alerts, security white papers, and background on hackers. Additional years for the SOHO products are $95/£67 each or $135/£96 for two years.

FIRST IMPRESSIONS

The first things you notice when you open the WatchGuard SOHO box are that the Installation Guide is one side of a 17-inch long piece of heavy glossy paper, and the firewall is very light (10 oz/284 grams).

Don't let the size and weight fool you. All WatchGuard products use security-hardened Linux-based operating systems. The SOHO runs it on a Toshiba TMRP3907 processor, with VLSI VMS113 encryption accelerator, 4MB of SDRAM, and 1MB of flash memory, with fairly impressive results, including 5Mbps WAN to LAN throughput, 9Mbps packet filter throughput, and 1.3Mbps 3DES encryption throughput.

ETHERNET CABLES

Having just been through a major hassle with Ethernet cables, I wanted to see how the installation instructions dealt with cabling to a high-speed Internet modem. In fact, the whole issue is neatly avoided by instructing you to use your existing cable from the modem to NIC as the connection from modem to the WAN port of the firewall. The CAT5 Ethernet Network cable in the box is intended to connect the first computer to the firewall.

INSTALLATION GUIDE

When, as instructed in the Installation Guide, I went to http://

bisd.watchguard.com/soho/install, it told me that I was running Windows NT 4.0 and Microsoft Internet Explorer (IE) Version 4.0. In fact, I was running Windows 2000 and IE 5.5. The same thing happened when I clicked on 'automatic detection' at the second URL, http://bisd.watchguard.com/soho/helpinstall.htm (for computers with different operating systems and/or browsers). This second URL also allows you to choose the correct operating system and browser without using automatic detection. Despite the fact that the box that the firewall comes in promises support for Windows 2000, the closest heading and entry on the Web page was: Microsoft Windows NT – Internet Explorer 5. I clicked on this, to find the following display: Installation Procedure – Windows NT 4.0 and Internet Explorer 5.0.

THE NEXT STEPS

After opening the first Web page, the next step in the installation process is to register. But it's not at all clear how you actually do this. Step 9, at the very bottom of this long Web page, is entitled 'Registering for LiveSecurity' – but it's not clear whether this is the same thing.

Just below the Installation Process on the Installation Guide is a section entitled Help. Step one lists two URLs. This time, the first URL actually worked, without requiring a user name and password. Unfortunately, the Troubleshooting link didn't address the problem, and the Installation link just took me back to the first install URL.

A FEW MORE PROBLEMS

With all this conflicting information, it was hard to know what to do, but I chose to follow the on-line Installation Procedure for Windows NT 4.0 and Internet Explorer 5.0.

Despite a high level of detail in the "Install the firewall" step, they missed the obvious fact that the firewall must be connected to an external power source (wall outlet). Some of the other Install steps were difficult to follow because of changes between NT 4.0 and Windows 2000. The most obvious example is NT 4.0's Network icon in Control Panel, which has been replaced by a folder in Windows 2000 and is more easily accessed from its own Start button menu item:

Start-Settings-Network and Dial-up Connections. But, more important, the contents of each dialogue box tab have changed significantly.

During installation, it became obvious that the CAT5 cable in the box didn't fit properly in the firewall port because the boot (cover for the RJ45 connector on each end of the cable) will not retract. It can be forced into place, clicking to indicate it's properly connected, but then it's impossible to remove without a small screwdriver.

Since this was the first time I'd networked the two workstations together, I didn't expect the Windows-System Error dialogue box with the message:

> A Duplicate Name exists on the network.

The first time was just before workstation log-on during start-up of the second workstation, the second when doing the ipconfig/renew on the second workstation. As you might guess, I am logged on as Administrator on both workstations. Nothing bad happened as a result.

So, I now had a working network, despite the fact that "registration is required for activation of your unit..." – or maybe I just had a very expensive Ethernet hub and no firewall.

THE RIGHT DOCUMENTATION

At this point, I ran out of time, and shelved the project for several weeks, unplugging the power from the firewall. During this time, two new installation-helpful documents arrived:

- SOHO Quick Start Guide – a poster-sized sheet with 13 installation steps on the front and 'Troubleshooting Tips' on the back. In fact, these tips also include Installation details for less common situations. And lots of page references to (2) below. Part # 97-200000-01.

- Version 2.3 of the SOHO User Guide – be sure to get the whole thing (not just Chapter 1 – Installation), and be sure to get Version 2.3 or the Quick Start Guide page references will be wrong. At the time of my review, despite being labelled as "the latest", only Version 2.2 was available for download as an Adobe Acrobat .pdf file at http://bisd.watchguard.com/soho

I would strongly recommend that you get both before attempting installation, and ignore all the other sources mentioned above – they will only confuse you, as they did me. Taken together, they address virtually all of the previously-detailed difficulties during Installation. Obviously, if you're reading this months or years after it's published, there may be more current versions available. The point is: be sure both are for the same version, or the page references will be wrong.

CONFUSION

One area of the new documentation that is still suspect is the procedure for disabling HTTP Proxy in IE 5.*x*. It focuses entirely on clearing all the check boxes under HTTP 1.1 Settings in Tools-Internet options-Advanced. It says nothing of the Use a proxy server check box in the IE menu bar item Tools-Internet options-Connections tab-Lan Settings button, which was the sole focus of the previous documentation.

Even when read by itself, the new *Quick Start Guide* is still confusing in the area of registration. It lists three methods:

- The last installation step (number 13) on the front suggests that you go to http://191.168.111.1/login.htm, then click the Register link.

- "How do I register my SOHO?" in Troubleshooting Tips-General suggests http://191.168.111.1 then System Administration-System Password-Click here to register your SOHO, enter information and then Save Profile.

- "How do I register for Live Security?" in Troubleshooting Tips-General suggests http://191.168.111.1 then click Live Security Home.

I chose the first option listed above. The login.htm screen, labelled 'System Password', provides three options:

- Click here to register your SOHO.

- Click here to update your information.

- Enable Password check box with fields to enter Name and Password (twice) just below, and Submit and Reset buttons.

The first option is the Register link referred to in Step 13. It attempted to go to

http://bisd.watchguard.com/SOHO/NewUser.asp?MAC=00907F-111597

but failed on a 404 Cannot find server or DNS Error. Once I'd ensured that the workstation had access to the Internet by bringing up a popular Web site in another browser window, I pushed the browser's Refresh button, producing a screen to register for LiveSecurity Service (1 year) that asked for contact information and ISP information, having already filled in my firewall's serial number.

When I pushed the Save Profile button, a "We're sorry" Web page appeared, announcing that:

> Access to this Web site is limited to registered WatchGuard SOHO customers with current LiveSecurity Service licenses. You must install your WatchGuard SOHO before you can register it. To register your WatchGuard SOHO, first go to SOHO's User Information page (typically at http://192.168.111.1/login), then click on LiveSecurity in the top banner.

This page gave me yet another way to register, and I was still confused about the difference, if any, between registering the firewall and registering for LiveSecurity.

Given my experience so far with WatchGuard's Web pages being out of date and/or wrong, I decided to follow the directions in the Troubleshooting Tips on the back of the new Quick Start Guide, under "How do I register my SOHO?", namely:

1   Go to http://192.168.111.1

2   Click System Administration

3   Click System Password

4   Click here to register your SOHO

5   Enter information

6   Click Save Profile.

But when I got past Step 2, there was no System Password link to click.

My second attempt was with the instructions on the "We're sorry" Web page shown above. To keep the instructions visible, I right clicked on the http://192.168.111.1/login link and selected Open in New Window from the pop-up menu. When the new browser window appeared, I clicked on LiveSecurity in the menu bar.

WELCOME

A Welcome screen appeared, prompting for user name and password, but with a message on the left:

> Are you a new User? You will need to register a user account. Your personal account information will be kept private.

When I clicked on the word 'register', the same "We're sorry" Web page reappeared. Since the sorry page began "You must install...before you can register", three things occurred to me:

- My problem could well be that I had powered off the firewall between initial installation and my current attempts at registration.

- I had not powered off my ADSL high-speed Internet modem when I most recently plugged it into the powered off firewall.

- The new Quick Start Guide did not include the step to renew TCP/IP settings for those, like me, who run DHCP.

I therefore used my browser's Back button to get to the Welcome Web page, then powered off and powered on my ADSL modem. "We're sorry" reappeared when I clicked the Register link. So I went back to the original Web page with the instructions for NT 4 and IE5.0 at

http://bisd.watchguard.com/soho/install/WinNTIE5.htm

to find the ipconfig sequence to renew the TCP/IP settings. At a Command Prompt, the current settings with ipconfig/all looked right. And they didn't change after ipconfig/release, ipconfig/renew, or ipconfig/all.

When I returned to the Welcome Web page, "We're sorry" appeared as soon as I clicked the Register link. Since it stated that Support staff didn't work at weekends, my only immediate alternative was to try installation from scratch – this time using the new *Quick Start Guide*.

PAST SORRY

This time, the last step, number 13 registration, worked, not giving the "We're sorry" page, but "Choose your LiveSecurity Login Name". I was asked to select an easy-to-remember user name and password for login. I chose a six letter ID and five letter password and both were accepted. Four fields were shown and I was asked to click the Register button:

- Serial number – matches the label on the bottom of the firewall.

- Hardware address – the MAC in the previous URL, presumably the unique address associated with the workstation's NIC.

- Customer number – the first time I have seen these five digits.

- Customer name – my first and last name.

This whole process had taken quite a while, so I took a half hour break just before hitting the Register button. Once I did hit the button, an "Invalid device ID – cannot register device" Web page was displayed:

Probable causes include:

– The device is not a WatchGuard broadband firewall

– The device is not in the WatchGuard hardware database

– Web server error

Please send e-mail to BroadbandSuport@watchguard.com with the following information:

– SOHO Serial Number

– SOHO MAC address displayed below

– Web browser type and version used to register

MAC Address = unknown

When I repeated Step 13 (registration), my selected User Name failed:

Please Try Again

The Login Name you have chosen is already in use

SUCCESSFUL REGISTRATION

The third time, I tried a different Login Name, which was accepted. When I hit the Register button, there was a long pause before the same Welcome screen appears, only this time it had my ID and the right number of asterisks for my password. The Remember my Password check box was checked off, which I left, and hit the Login button.

"Welcome" followed by my first and last name appeared, with a picture of my firewall with "Version 2.3 Available Now" just above it. The left sidebar contained four entries:

- Registration

- LiveSecurity archives

- Firewall software updates

- SOHO Version 2.2 user guide now available.

I wasn't sure if I was fully registered, so I clicked on Registration. It took me back to my registration form, allowing me to update my contact information, but changing certain fields back to their defaults:

- Send me information on software updates (checked off)

- Send me general network security and virus information (checked off)

- Primary product usage

- Connection type

- Service provider.

I made the necessary changes and pushed the Save button. An initial 404 Cannot find server or DNS Error page disappeared when I pressed the browser's Refresh button, and the Welcome-my-name page reappeared, with no indication of whether the Save actually worked. Clicking the Registration link gave another 404 page. Pushing the browser's Back button, re-clicking the Registration link, remaking the changes, and hitting the Save button was followed by a long wait, but this time the Welcome-my-name page did appear, still with no indication as to whether or not the changes had successfully been made.

UPDATING FIRMWARE

When I clicked on Firewall Software Updates I got another 404 page. I went back to Welcome, refreshed that page, and again clicked on the Firewall Software Updates link; this time I got the Firewall Software Updates page. It stated that "the current SOHO, SOHO|tc, Telecommuter Software Release is: 2.3.16", then listed four options, each a link:

- Click here to download the latest release

- Click here to read the latest Release Notes

- Click here for help installing this update

- Click here to download the latest SOHO User Guide.

From http://192.168.111.1, clicking System Information then Features and Version Information lists Firmware Version 2.2.3 dated 15 December 2000. The unit was shipped to me on 23 April 2001.

To actually do the firmware upgrades, from http://192.168.111.1, I clicked on LiveSecurity from the menu bar and was taken to http://bisd.watchguard.com/SOHO/Login.asp, with my user name and password pre-filled. All I had to do was hit the Login button. I was back to the Welcome-my-name page and I clicked on Firewall Software Updates from the left sidebar, then the link titled Click here to download the latest release. I chose to "Run this program from its current location". It was 1.18 MB and VeriSign verified the publisher's authenticity. Then the WatchGuard LiveSecurity Update Wizard appeared with the usual steps you expect from a Windows software wizard, including a Ready to Install dialogue box indicating:

> We now have all necessary information and data to update your WatchGuard SOHO. This process takes up to 3 minutes to complete.

> It is very important that you DO NOT UNPLUG your firewall during the update process. Doing so may corrupt your software, requiring re-installation.

When I pushed the Install button, the wizard showed its progress by ungreying each step, then checking off a check box beside the step to

indicate it was complete. At the end of the process, it rebooted, leaving just the Firewall Software Updates page in the browser window.

EVENT LOG AND FIRMWARE LEVELS

From http://192.168.111.1, clicking System Information then Features and Version Information listed Firmware Version 2.3.16 dated 22 March 2001. A check of the Event Log:

*   Go to http://192.168.111.1

*   A Configuration Web page is shown

*   Click on System Information link

*   Click on Event Log

showed that the Event Log had been erased before the reboot. Events are colour coded, Green for regular events like a workstation connecting and the reboot itself, and Yellow for areas of possible concern like unsolicited IP packets, perhaps hackers checking you out. Red is for critical situations, but I have yet to see any of those. When you're viewing the Event Log, the Web page is automatically updated in real-time without you having to hit the Refresh button on your browser's toolbar.

Less than 12 hours later, Firmware Version 2.3.21 was available. It was 774KB, but created a Security Warning dialogue box because it lacked an Authenticode signature. Like the Event Log, the Network Statistics also start from zero after a reboot. They can be seen from the same System Information page, by clicking Network Statistics instead of Event Log.

SOME SURPRISES

Surprisingly, this second Firewall Software Update was done using my original User Name, the one that failed to do the registration. And somewhere along the way, without shutting anything down, I plugged the second workstation into the firewall and it just started to work without any difficulties. Although the *Quick Start Guide* does not discuss connecting more than one workstation to the firewall, the

Installation chapter of the User's Guide suggests connecting them all at once, during initial installation.

All ports have equal status, and the second workstation can access the administrative Web pages on the firewall just as easily as the first. As mentioned previously, by default, there's no password required to access the firewall from workstations connected to it. But a password can easily be set from these same Administration Web pages on the firewall itself.

REMOTE ADMINISTRATION

By default, remote administration of the firewall is turned off, but it can be turned on simply by removing a check mark from a box. On the same Web page, four passwords ('phrases') can be set:

- Remote Configuration read only phrase

- Retype read only phrase

- Remote Configuration read write phrase

- Retype read write phrase.

Finally, some quick tests showed that normal activities were unimpeded by the presence of the firewall, including the following:

- Browsing Web pages.

- Downloads via Web page links.

- tn3270e access across the Internet to RACF on a mainframe running OS/390 TSO/E.

- ftp access to the same RACF mainframe, viewing a directory, and uploading, downloading, and deleting PDS members.

- ftp access across the Internet to a Windows 2000 server, viewing a directory, browsing all files on the site to calculate total disk space usage, and uploading, downloading, and deleting a file on the server.

Jon Pearkins
(Canada)   © Xephon 2001

The top sources of RACF- and security-related information

Over recent years, our *Information points – reviews* have looked at many sources of RACF-related information. This article summarizes the 11 most valuable.

IBM MANUALS ON THE INTERNET

IBM offers thousands of its manuals on the Internet, including many RACF-related publications, at no cost for Web viewing and/or downloading. Originally only in a browser-based simulation of BookManager, many are now (also or exclusively) being offered in Adobe Acrobat (.pdf) format.

VM – http://www.vm.ibm.com/library

Under the heading *z/VM, VM/ESA, and VM-related product documentation*, click on *Product publications*. For the most complete collection of VM manuals, click on System/390 bookserver under the *VM Library in BookManager format* heading. Look for, or use your browser's Find function to find, the RACF bookshelf and click on 'List books'.

z/OS – http://www.ibm.com/servers/eserver/zseries/zos/bkserv

Under the heading *z/OS elements and features publications*, click on the relevant version/release. Under the *z/OS elements and features bookshelf* heading, click on 'List – All bookshelves'. Look for, or use your browser's Find function to find, the SecureWay Security Server bookshelf, and click on 'List books'.

OS/390 – http://www.ibm.com/servers/s390/os390/bkserv

Under the heading *OS/390 elements and features publications*, click on the relevant version/release. Under the *OS/390 elements and features bookshelf* heading, click on 'List – All bookshelves'. Look for, or use your browser's Find function to find, the Security Server bookshelf and click on 'List books'.

MVS/ESA – http://www.ibm.com/servers/s390/os390/bkserv/mvs.html

Under the heading *MVS/ESA System Control 1 (Disc 2) bookshelf*, click on 'List – All bookshelves'. Find the relevant release of RACF Version 2, and click on 'List books'.

RACF HOME PAGE AND REDBOOKS – http://www.ibm.com/servers/eserver/zseries/zos/racf

The right sidebar provides access to the areas of greatest interest:

- Overview – includes two Shockwave segments

- What's new – new features and recently-published articles

- VM

- Downloads – free utilities, tools, and sample applications

- User groups – in North America

- On the road – sessions at upcoming conferences

- Migrating – including version upgrades

- Library – handy links, including RACF Redbooks

- Related links

- FAQs – several dozen technical questions and answers

- Education – recommended IBM courses

- Presentations – slides from past conference sessions.

Redbooks have always provided an outlet for less formal material than manuals, much of it of extreme value. Many of them are essentially detailed diaries of IBM projects, at both customer and IBM sites, with some effort made to remove installation-specific information in the sample code and the text. Others are based on technical presentations at conferences and other venues. A more recent trend, exemplified by the five-volume *ABCs of System Programming*, is to provide the type of textbooks you would expect to find in a computer bookstore.

Redpieces are redbooks in progress. Redpapers do not qualify as redbooks for one reason or another, and are available only on-line. Residencies list the future openings to be a member of the team that creates a specific redbook.

RACF-L – http://www.listserv.uga.edu/archives/racf-l.html

List Servers were originally developed to require only e-mail access, but have been webized in recent years to simplify use. RACF-L is the world's largest RACF technical Q&A session, with 1,500 subscribers and 200 questions a month. Each question is a thread with as many as 20 responses.

To join the list:

- Click on the Join or leave the list (or change settings) link.

- Fill in your e-mail address and name.

- Unless you want to receive every message as it is posted to RACF-L, check the 'Mail delivery disabled temporarily' box near the bottom.

- Hit the Join the list button at the bottom of the page.

- A 'confirmation request is being sent under separate cover' message will be displayed.

- Check your e-mail.

- Click on the confirmation link in the e-mail message (or copy and paste the URL into a Web browser).

- A Command confirmation page is displayed.

- A confirmation e-mail is also sent.

Registration is also required to search or browse the archives:

- Click on Register in the menu bar near the top of the page.

- Enter your e-mail address and a password of your choice (twice).

- Push the Register Password button.

- A Confirmation e-mailed page explains that an e-mail has been sent to you explaining how to activate your password.

- The e-mail contains a link that, once clicked, activates your password and displays a Command confirmation page.

Once you've registered, joining, leaving, and changing settings for a list can be done more easily, without the e-mail confirmation, so you may want to register first, before joining.

From the URL above, click on 'Search the archives or browse' by clicking on one of the listed months. Messages appear almost instantly in the archives under the current month. Each month's messages are sorted by subject, with upper-case letters preceding lower-case (which means Z is before a).

The first time you search or browse, you will have to log in:

- A Login required page will appear.

- Fill in your (registered) e-mail address and password.

- Push the 'Login and save my password in a cookie' button.

- A Password saved page will appear.

- Click on follow this link to do the search or browse you requested.

STU HENDERSON – http://www.stuhenderson.com

Best known for *RACF User News*, published three times a year, the HG Information Security Home Page also includes a few of Stu's security articles, contact information for RACF user groups, and his favourite links to:

- Computer security

- Mainframe

- RACF

- Windows NT security

- Vendor-specific

- User groups.

INFOSYSSEC – http://www.infosyssec.net

The Security Portal for Information System Security Professionals attempts to provide a link to every Web page and news group of interest on the Internet. Despite a long home page and small type, many links lead you to yet another Infosyssec list of links. A few of the links are to original material created specifically for the site.

Each link opens in a new browser window. Many external links are broken, typically leading to a 404 Page not found message, though at least one leads to an exploding set of new browser windows opening, advertising everything imaginable. One-step links are provided to a broad range of categorized search engines, allowing immediate entry of your search term (to one search engine at a time). There is also a news ticker and, of course, the advertising that keeps the site alive.

The only thing missing is a site search. The Find button of the InfoSysSec Web Search in the InfoSysSec Open Source Search Directory section looks promising, but it is an Internet-wide search engine.

SANS – http://www.sans.org

The SANS (System Administration, Networking, and Security) Institute has 100,000 members and is heavily involved in certification. Scroll down past its training events list to gain access to the real treasures of this site. The Security Reading Room section has more than 40 subject links, each to a page of links to technical papers written by SANS members to fulfill part of their GIAC (Global Incident Analysis Centre) certification requirements. The Security Digests section offers a number of weekly and monthly SANS publications, all available free. None of which takes away from the fact that the Training Events listed includes worldwide conferences and certification training in both classroom and Web-based settings.

CERT/CC – http://www.cert.org

The CERT Coordination Centre (CERT/CC) is synonymous with Internet security expertise. Click on FAQ in the menu bar at the top of

the page for more information on CERT/CC. Just below the menu bar, and repeated in the left sidebar, four major options are provided:

- Incidents, quick fixes, and vulnerabilities

- Security practices and evaluations

- Survivability research and analysis

- Training and education.

Next in the left sidebar is the 'Related' section, with links to publications and presentations by CERT staff.

AUDITNET – http://www.auditnet.org

By creating this site, auditor Jim Kaplan set an example of sharing that has inspired others – such as Pamela Jerskey of Boston College, who offers her RACF audit plan. There's a lot here, but the AuditNet Library and Search AuditNet, both on the left sidebar, are the best bets. The search uses Google; be sure to change the default Search WWW to Search www.auditnet.org.

THIERRY FALISSARD – http://os390-mvs.hypermart.net

The OS/390-MVS Cyber Mall is dedicated to Thierry Falissard's passions, including security, cryptography, and MVS. In fact, click on one of these words on the home page and you'll get to a page of links dedicated to the subject. Considerable care has been taken to serve the English-only visitor, but a knowledge of French will increase the value of this site to you.

XEPHON IS LIBRARY – http://www.xephon.com/is_library.html

Click the 'IS Library' button in the left sidebar for free access to 200 articles extracted from Xephon Reports, including some on RACF and other security topics. Members of Xephon's Corporate License scheme have access, without additional charge, to all 400 articles. They are divided into five categories:

- Business and IT strategies

- New technologies and applications

- Data centre management

- System/390

- Other systems.

Click on an article title to have an Adobe Acrobat (.pdf) version e-mailed to you; free registration is required the first time. To locate titles of interest on this page, select text anywhere on the right side of the page, then use your browser's Find function to search the titles. Alternatively, you can use the Search Site function on the left sidebar to list both articles and Update journal pages.

VIRUS DESCRIPTIONS AND HOAXES – http://www.mcafee.com/anti-virus

In the centre of the page, click on either 'Virus Information Library' or 'Virus Hoaxes'. The site offers very detailed information on 50,000 real viruses and 70 hoaxes.

Jon Pearkins
(Canada)
© Xephon 2001

Looking for a specific article?

If you keep hoping for an article on a particular topic, but we never publish one, please let us know what the subject is. If it's likely to be of interest to other subscribers too, we'll commission it and publish it in *RACF Update*.

Visit the *RACF Update* Web site

http://www.xephon.com/racfupdate.html

and follow the link to *Opportunities for RACF specialists*.

August 1995 – August 2001 index

Items below are references to articles that have appeared in *RACF Update* since August 1995. References show the issue number followed by the page number(s). Back issues of *RACF Update* can be ordered from Xephon. See page 2 for details.

Contributing to *RACF Update*

In addition to *RACF Update*, the Xephon family of *Update* publications now includes *CICS Update*, *MVS Update*, *TCP/SNA Update*, *VSAM Update*, *DB2 Update*, *AIX Update*, *Domino Update*, *MQ Update*, *NT Update*, *Oracle Update*, and *TSO/ISPF Update*. Although the articles published are of a very high standard, the vast majority are not written by professional writers, and we rely heavily on our readers themselves taking the time and trouble to share their experiences with others. Many have discovered that writing an article is not the daunting task that it might appear to be at first glance.

They have found that the effort needed to pass on valuable information to others is more than offset by our generous terms and conditions and the recognition they gain from their fellow professionals. Often, just a few hundred words are sufficient to describe a problem and the steps taken to solve it.

If you have ever experienced any difficulties with RACF, or made an interesting discovery, you could receive a cash payment, a free subscription to any of our *Updates*, or a credit against any of Xephon's wide range of products and services, simply by telling us all about it. For a copy of our *Notes for Contributors*, which explains the terms and conditions under which we publish articles, please visit our Web site at www.xephon.com/nfc. Articles can be sent to the editor, Fiona Hewitt, at any of the addresses shown on page 2, or e-mail her at fionah@xephon.com

RACF news

e3Sciences has developed two toolkits that create one-time passwords and store the key securely; they can also encrypt passwords for systems that do not support one-time passwords. PassGen for Windows runs on Windows 9*x*, NT, and 2000. PassGen for Palm OS requires at least Palm OS 2.0 and 46KB of systems memory. Both provide one-time passwords to hosts running at least RACF 1.9, CA-ACF2 6.1, CA-Top Secret 5.1, or any S/KEY (RFC 1760) compliant system, typically Unix or a firewall.

For more information, contact:
e3Sciences, Kingston House, Kings Stanley, Gloucestershire GL10 3JF, UK.
Tel: (44) 1280 707445.
URL: http://www.e3sciences.com

* * *

Version 3.7 of Tivoli SecureWay Global Sign-On (GSO) includes support for Sun Solaris 7 and 8 servers and Windows 2000 clients (with integrated login and generic target password initialization). It also offers improvements to the Software Developer's Kit (SDK).

For more information, contact:
Tivoli Systems, 9442 Capital of Texas Highway North, Arboretum Plaza One, Austin, Texas 78759, USA.
Tel: (512) 436 8000.
Sefton Park, Bells Hill, Stoke Poges, Bucks SL2 4JS, UK.
Tel: (44) 1753 780 000.
URL: http://www.tivoli.com

* * *

IBM z/VM 4.1 is now available. Only Version 1.10 of RACF is supported, with APAR VM62958 required for 64-bit mode operation.

For further information, contact your local IBM representative, or visit the Web site at http://www.ibm.com

* * *

Link/Manage's new Secure/FTP uses exits for the TCP/IP stacks and the SAF interface to let RACF, CA-ACF2, and CA-Top Secret control all ftp access to OS/390 and z/OS. And not just by user ID and password, but also by IP address, and ftp and site commands. Minimum supported versions of the required software are IBM TCP/IP 3.4 and CA's NetworkIT TCPAccess 5.2.

For further information, contact:
ASPG, 3185 Horseshoe Drive South, Naples, FL 34104, USA.
Tel: (800) 662 6090.
URL: http://www.aspg.com
Software Europe, Nibley House, Low Moor Road, Doddington Road, Lincoln LN6 3JY, England.
Tel: (44) 1522 881300.
URL: http://www.software-europe.co.uk
France: Etic, tel: (33) 01 46 48 69 69.
URL: http://www.eticsoftware.com
Germany: Net'Q, tel: (49) 89 699693 0.
URL: http://www.net-q.com
Italy: Integra, tel: (39) 02 89 30 71.
URL: http://www.integra-group.it
URL: http://www.linkmanage.com

* * *

xephon