WWW.RACF.NET.AU presents:

Win a 'free pass' for your next audit

a white-paper created by Mike Cairns

For more information, contact us at info@racf.net.au or call 0412 488 484

“A review of the various freeware utilities available to help audit and administer the IBM Security Server (RACF) environment.”

Win a 'free pass' for your next audit

first published May 2007 – Xephon RACF Update

In our last issue (see ‘Forget Free Willy, how about free RACF!’, RACF Update, issue 47, February 2007) we looked at free tools for analysis of the RACF Unload file generated by the IRRDBU00 program: RACF reporting tools from Nigel Pentland, an MS-Access database from Cory Curtis, and the IBM RACF Goodies site MS-Excel database. While these tools are great for examining the structures in your RACF database, they’re only part of the total picture.

In this issue we’re going to explore tools for interpreting System Management Facility (SMF) data, the essential forensic audit trail of the mainframe environment. As luck would have it, all three contributors from our last article offer similar free tools for SMF reporting, so we will start with these.

OBTAINING THE SMF DATA

It is worth briefly covering the processes by which most mainframe installations’ SMF data is generated. SMF contains a multitude of information about all aspects of running your mainframe. Data is available for all job/user initiation, system hardware/software conditions and status, as well as the security activity records we are interested in.

By default z/OS supplies three datasets for ‘live’ SMF; these are: SYS1.MAN1, SYS1.MAN2, and SYS1.MAN3. Most likely at your installation these names have been customized in the current system parmlib member SMFPRMxx. You can view the dataset names in use with the MVS operator command D SMF. Common variations are SYS1.lpar-or-jesnode.MANx and SMF.*.MANx. Try using SYS1.**.MAN* or SMF.** in ISPF Dslist (option 3.4) if you don’t have access to the above mentioned definitive methods.

In any case, although the ‘live’ SMF datasets above are interesting, what most audit processes are concerned with is the SMF ‘dump’ datasets. Typically, SMF live datasets are ‘dumped’, or emptied, by a started task or batch job initiated via either a system exit or automated operations. The dump datasets created by this process are usually collected into a Generation Data Group (GDG) dataset and often grouped by date or period (eg SMF.DAILY, SMF.WEEKLY, etc) to provide a history of system management data for later analysis and processing. It is also common to see post-processing of the SMF data for special interest groups, eg SMF.DAILY.CICS, SMF.DAILY.DB2, and SMF.DAILY.RACF.

As a RACF administrator or auditor, you need to understand the typical flow of SMF data described above in order to know what information is available in the selection of security-related SMF records presented to you, the end user of this automated system process. You may have to request that the current SMF processing is altered to collect additional SMF record types into the subset generated for security analysis. A recommended set of SMF record types to cover most security-related events are:

- Dataset activity records (types 14, 15, 17, 18, 62, and 64).

- JES2 spool offload records (type 24).
- Catalog activity records (types 60, 61, 63, 65, 66, 67, and 68).
- RACF processing records (types 80, 81, and 83).
- HSM function statistics records (custom record type).

For the IRRADU processing, though, we require only a few types of SMF records – types 30, 80, 81, and 83 are used. The IRRADU utility consists of user exits for IFASMFDP (the SMF dump program). The exits IRRADU00 and IRRADU86 (see the JCL example below) select and format the relevant SMF records from the dump job. This formatting is what allows the relatively easy creation of new security reports using DFSORT and ICETOOL.

```
//*-------------------------------------------------------------------
//*- CREATE NEW IRRADUØØ OUTPUT
//*-------------------------------------------------------------------
//STEPØ2Ø EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//ADUPRINT DD SYSOUT=*
//OUTDD DD DISP=(NEW,CATLG),DSN=userid.RACFICE.IRRADUØØ, <--CHANGE
// SPACE=(CYL,(1ØØØ,5ØØ),RLSE),                          <--CHANGE
// DCB=(DSORG=PS,RECFM=V)
//SMFDATA DD DISP=SHR,DSN=SYS1.SMF.DAILY.DUMP(-1)        <--CHANGE
//SMFOUT DD DUMMY
//SYSIN DD *
INDD(SMFDATA,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(ØØØ:255))
ABEND(NORETRY)
USER2(IRRADUØØ)
USER3(IRRADU86)
```

The job above can be executed against both raw SMF datasets (ie the live SMF datasets) or SMF dump datasets previously unloaded during normal SMF processing. This job produces SMF records in a readable format that are then easily parsed by other tools. See below for example output from IRRADU SMF dump exits (records are truncated):

```
ACCESS   SUCCESS  1Ø:2Ø:51 1995-11-Ø6 IM13 NO  YES NO IEESYSAS JESXC
DEFINE   SUCCESS  1Ø:47:25 1995-11-Ø6 IM13 NO  NO  NO MARKN    SYS1
DELRES   SUCCESS  1Ø:47:51 1995-11-Ø6 IM13 NO  NO  NO MARKN    SYS1
RDEFINE  SUCCESS  11:14:Ø8 1995-11-Ø6 IM13 NO  NO  NO MARKN    SYS1
```

```
PERMIT   SUCCESS  11:14:34 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
SETROPTS SUCCESS  11:14:46 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
DELDSD   SUCCESS  14:01:52 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
DELUSER  SUCCESS  14:01:54 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
DELGROUP SUCCESS  14:01:59 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
ADDUSER  SUCCESS  14:02:00 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
ADDGROUP SUCCESS  14:02:04 1995-11-06 IM13 NO   NO  NO MARKN    SYS1
JOBINIT  INVPSWD  14:02:08 1995-11-06 IM13 YES NO  NO MARK     SYS1
CONNECT  SUCCESS  14:09:59 1995-10-04 IM13 NO   NO  NO RRSFU2   SYS1
PASSWORD INSAUTH  14:10:03 1995-10-04 IM13 YES NO  NO RRSFU2   SYS1
PERMIT   SUCCESS  14:10:07 1995-10-04 IM13 NO   NO  NO RRSFU2   SYS1
DEFINE   ALRDEFD  14:10:16 1995-10-04 IM13 YES NO  NO RRSFU2   SYS1
SETROPTS INSAUTH  14:10:20 1995-10-04 IM13 YES NO  NO RRSFU2   SYS1
```

So now we have some SMF data ready for analysis with the various tools at our disposal. For this article I am using the supplied sample IRRADU output from the RACFICE tools, available from the RACFICE ftp server in TSO XMIT format as filename ‘racfice.sampadu.xmit’. Simply transfer this file to your z/OS system using binary FTP to a pre-allocated dataset with RECFM=FB and LRECL=80. The file occupied just under 140 tracks (3390) when I tested this. Unpack the file to a dataset of your choice using the TSO RECEIVE command:

```
TSO RECEIVE INDA('uploaded-file')
```

Respond to the prompt with `DA(‘userid.RACFICE.IRRADU00’)` or your preferred dataset name.

I expect that at your installation you would be using ‘real’ SMF data rather than the sample data I’ve used here, the JCL in the first example should be customized to match your SMF input data, your chosen dataset name for the IRRADU output data, and an appropriate size for the output dataset. The IRRADU dataset (OUTDD in the example) is the input data for the reporting jobs covered in the remainder of this discussion.

NIGEL PENTLAND’S RACFICE JOBS

Nigel provides three sets of SMF reports on his Web site at www.racf.co.uk. Although some overlap exists between each set and also with the IBM-supplied default RACFICE reports, these reports provide a comprehensive overview of SMF activity requiring little or no customization to get working in your z/OS environment.

The three files provided are:

- racfsmf.txt – a set of RACF action reports.
- moresmf.txt – additional RACF action reports.
- audtrpts.txt – a comprehensive set of RACF reports incorporating the previous two sets.

These files contain JCL and DFSORT/ICETOOL control cards to produce the various reports. You can download the files from Nigel’s site and copy them to any JCL dataset member on your z/OS system. You will need to adjust the jobcards as well as the IRRADU00 input DD to be that of the IRRADU output dataset we generated from raw SMF above. In audtrpts.txt is a sample IRRADU00 SMF dump job, which I removed for my testing because I already had valid IRRADU data available.

Nigel gives credit to the many contributors who assisted him in creating these reporting suites. The audtrpts.txt sample is a collection of reports from the other samples, re-written in such a manner as to eliminate multiple passes of the SMF data. After running the three sample report sets, I agree that the audtrpts.txt seems to be the most comprehensive and the most efficient.

The list of reports produced by audtrpts.txt is:

- ACCESS VIOLATIONS
- ADDSD COMMANDS
- ADDGROUP COMMANDS
- ADDUSER COMMANDS
- ALTDSD COMMANDS
- ALTGROUP COMMANDS
- ALTUSER COMMANDS
- CONNECT COMMANDS
- DELDSD COMMANDS
- DELGROUP COMMANDS
- DELUSER COMMANDS
- PASSWORD COMMANDS
- PERMIT COMMANDS
- RALTER COMMANDS
- RDEFINE COMMANDS
- RDELETE COMMANDS
- REMOVE COMMANDS
- RVARY COMMANDS
- SETROPTS COMMANDS
- USERIDS WITH EXCESSIVE INCORRECT PASSWORDS
- USERIDS WITH EXCESSIVE ACCESS VIOLATIONS
- RESOURCES WITH EXCESSIVE ACCESS VIOLATIONS
- RESOURCES WITH EXCESSIVE ACCESS RECORDS
- RESOURCES WITH EXCESSIVE ACCESS RECORDS BY USERID.

Slight differences exist between the audtrpts.txt report set and the report sets generated by the other two jobs, an example being the distinction between ADDSD and DEFINE type records in the IRRADU00 output. For most practical purposes these records contain the same data and reporting on both is redundant. There is provision for reporting on both in the audtrpts.txt job; however, the duplicate report is not executed by default in the current version.

If you don't already have some regular RACF activity reporting, this suite is comprehensive, very easy to implement, and provides a great start to monitoring security activity within your z/OS system.

CORY CURTIS IRRADU00 DATABASE

The SMF database available from Cory's Web site (http://racf.curtistree.com/) must be initialized with definitions from SYS1.SAMPLIB(IRRADULD) before its first use. This samplib member contains DB2 load statements that are read to determine the correct structure for the MS-Access database. After initialization of the database, the output of the IRRADU SMF dump job is transferred to a PC where the MS-Access database can import it for offline analysis. These instructions are covered in detail on Cory's Web site and reproduced in brief here:

1. Download and unzip the racf.zip file to a convenient directory. This contains the Access database.
2. File transfer your current SYS1.SAMPLIB(RACDBULD) dataset to C:\temp\temp\irraduld.txt.
3. Open the SMF.mdb Access database and run the 'First time run' macro.
4. File transfer IRRADU SMF dump output to C:\temp\temp\Data.txt.
5. Again in the SMF.mdb database, run the 'One Step Import' macro.

This process is identical to that used in Cory's RACF unload file MS-Access database. The 'First time run' macro must be executed against a fresh copy of the IRRADULD samplib member whenever your z/OS system is upgraded and all file transfers must be done as ASCII text. The file names for the IRRADULD and the SMF data are coded within the MS-Access macros, but can be altered if desired. For my testing I used the sample IRRADU output from the IBM RACFICE package – after transferring this to the mainframe and unpacking the XMIT file, I simply FTPed this data as ASCII text back to my PC.

Following these steps I had a working MS-Access copy of the SMF data within minutes. Once again Cory has created a very useful analysis tool for those comfortable with using MSAccess.

There are a large number of tables available that represent all kinds of security-related events, including many OMVS-related event types such as filesystem mounts/unmounts, use of Extended Access Lists, and UID/GID use within the Unix filesystem. This detailed information is not readily available in the other SMF processing tools discussed here.

Unfortunately, many of these tables are not populated by the RACFICE provided sample data. The tables that are populated, though, interpret the entire IRRADU record and include many fields that are not displayed in the reports produced by the DFSORT- and ICETOOL-based utilities.

If you need to create custom reports using potentially all the available SMF fields, this relational database utility would be very useful. Similarly, it is far easier to create a truly relational query incorporating the results from more than one table (ie ADDSD and DEFINE can be 'merged' into one comprehensive record) using MS-Access than DFSORT and ICETOOL. The availability of record types not currently processed by the other utilities reviewed here makes Cory's MS-Access database a highly-attractive free auditing tool.

IBM RACF GOODIES’ SITE RACFICE REPORTS

IBM’s RACFICE is the new RACF Report Writer. There is even a direct comparison of RACF Report Writer functions against equivalent RACFICE reports within the RACFICE documentation. IBM’s vision is to use generic report writing tools already available at all z/OS installations rather than continue to enhance the RACF Report Writer to process the regularly updated list of SMF records and sub-types – hence the SMF dump exits IRRADUxx were created.

RACFICE can be downloaded directly from the IBM RACF FTP server at: ftp://ftp.software.ibm.com/eserver/zseries/zos/racf/racfice/ – the RACFICE Web site recommends using the copy provided in SYS1.SAMPLIB(IRRICE) because this is guaranteed current with your installed version of z/OS. However, on examining the supplied SAMPLIB member and discovering that it contains IEBUPDTE statements to build the final jobstream(s), I elected to upload the racfice.xmit PDS supplied on the IBM RACF Web site instead. More years ago than I care to remember, I forgot how to use IEBUPDTE, and nowadays always have to look up this cryptic, albeit extremely useful, tool whenever it is needed. Being a person who prefers the path of least resistance, I decided it was prudent to at least check out the xmit file from the RACFICE pages first.

Transfer this dataset as usual using binary FTP to an FB(80) pre-allocated dataset, then use the TSO RECEIVE command to unpack to a destination PDS of your choice. On examining the output PDS you will find over 70 members, the most important of which is the $$CNTL$$ member. Edit this member to have a valid jobcard and ensure that the three JCL SET statements point to the IRRADU00, IRRDBU00, and the JCL library just created respectively. Similarly, point the JCLLIB statement to the just-created JCL library.

With these minor changes I was able to produce the complete set of RACFICE reports in only a few minutes – just submit the $$CNTL$$ member and browse its output. However, after examining the reports generated, I realized that the RACFICE version available on the RACF Web site is not as current as the version in my SYS1.SAMPLIB(IRRICE) member – so back we go to IEBUPDTE.

A quick reference to the DFSMSdfp Utilities guide gave me a rough IEBUPDTE job to work with. The JCL below can be used to unpack the jobstream supplied in SYS1.SAMPLIB(IRRICE) to a new JCL library of your choice:

```
//USERØ1 JOB (),' ',CLASS=A,MSGCLASS=X,
// NOTIFY=&SYSUID
//STEP1 EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSNAME=userid.RACFICE.CNTL,DISP=(,CATLG),     <--CHANGE
// SPACE=(TRK,(2Ø,1Ø,1Ø)),
// DCB=(RECFM=FB,LRECL=8Ø)
//SYSIN DD DSN=SYS1.SAMPLIB(IRRICE),DISP=SHR
```

Once this job is complete, edit the library created and apply exactly the same changes mentioned above to the $$CNTL$$ member. You can then submit the $$CNTL$$ JCL and generate your reports – there are 38 reports provided by default with the version I tested on a z/OS 1.4 system:

- ALDS – discrete dataset profiles that have IDs on the standard access list with ALTER authority.
- ASOC – users who have explicit associations defined.
- BGGR – discrete general resource profiles with generic characters in their name.

- CCON – count of user connections, flagging those with more than x connections.
- CGEN – count of general resource profiles.
- CPRO – count of profiles.
- CONN – user IDs with group privileges above use.
- GIDS – shared Unix System Services GIDs.
- IDSC – dataset conditional access lists with ID(*) of other than NONE.
- IDSS – dataset standard access lists with ID(*) of other than NONE.
- IGRC – general resource conditional access lists with ID(*) of other than NONE.
- IGRS – general resource standard access lists with ID(*) of other than NONE.
- OMVS – user IDs that have Unix System Services (OMVS) segments.
- PCAM – PROGRAM class specific profiles with MAIN or BASIC APPLDATA.
- SUPU – Unix System Services super users (UID of zero).
- UGLB – user IDs with extraordinary system-level authorities.
- UGRP – user IDs with extraordinary RACF group authorities.
- UIDS – shared Unix System Services UIDs.
- URVK – user IDs that are currently revoked.
- UADS – dataset profiles with UACCs of other than NONE.
- UAGR – general resource profiles with UACCs of other than NONE.
- WNDS – dataset profiles in WARNING mode.
- WNGR – general resource profiles in WARNING mode.
- ACD$ – users who are using automatic command direction.
- CADU – count of IRRADU00 events.
- CCMD – count of commands issued (by user).
- ECD$ – users who are directing commands explicitly.
- LOGB – users who log on with LOGON BY.
- LOGF – all users with excessive incorrect passwords.
- OPER – accesses allowed because the user has OPERATIONS authority.
- PWD$ – users who are using password synchronization.
- RACL – RACLINK audit records.
- RINC – RACF class initialization records.
- SELU – all audit records for a specific user.
- SPEC – events that succeeded because the user has SPECIAL authority.
- TRMF – excessive incorrect passwords from terminals.

- VIOL – access violations.
- WARN – accesses allowed due to WARNING mode profiles.

Remember that many of these reports use the RACF unload file (IRRDBU00) as well as the SMF data (IRRADU00), so for the best results it's important to have both sources of security information up-to-date before you run these reports.

Once you've mastered IEBUPDTE again, RACFICE is easy to use and produces a good range of reports. Comparing the reports produced by IRRICE against the list produced using Nigel's audtrpts.txt example reveals that there is only a slight overlap between these two sets of reports, mainly in the access violations and excessive passwords derived reports.

Of course, RACFICE also analyses the RACF flat file IRRDBU00 and produces many reports from this data that are not available in the alternatives reviewed here. Overall, I would be tempted to use both sets of reports for a comprehensive overview of security.

CONCLUSIONS

Similar to the utilities reviewed in the last issue, I find the table names created by utilities that rely on IBM's DB2 Load statements – Cory's SMF.MDB in this case – to be a little cryptic and difficult to understand without a good knowledge of the source data and record and field definitions. The format of records produced by the SMF unload utility IRRADU00 is thoroughly documented in the current RACF Macros and Interfaces manual, but there is a lot of information to absorb in there, and to me seems a shame that more 'intuitive' table and field names cannot easily be used.

Despite this, of the three utilities reviewed here, my personal preference has to go with Cory Curtis' MS-Access database. Although it supplies no canned reports that can be run immediately against your SMF data, it does produce the most comprehensive coverage of SMF events by virtue of using the IBM-supplied (ie current) record and field definitions in order to create its table definitions. If you need a comprehensive review of the contents of your SMF data this is a great tool.

The main advantage of the other two tools looked at here are the supplied reports, getting you off to a fine start towards a set of audit reports relevant to your installation. As stated previously, I would tend to use a combination of Nigel's supplied reports and the base IRRICE reports that IBM supplies.

In the next issue we will continue to find new sources of data to increase our awareness of security-related activity and our ability to report on this. Specifically we will take a look at methods of analysing the Unix System Services Hierarchical File System.

If you need any advice about effectively exploiting your System z investment through consulting, performance measurement and management improvement, staff training etc don't hesitate to contact us at info@racf.net.au for a confidential discussion.