

APFCHECK

APFCHECK

APF Library Privilege Checker

Goldis Consulting Services

1 Trowbridge Terrace
Cambridge, MA 02138

Phone: 617-492-4364

Fax: 617-492-1710

www.goldisconsulting.com



APFCHECK

Introduction

The Authorized Program Facility (APF) is used to extend the Z/OS (and its predecessors) operating system with user-written or third party products. Programs residing in APF libraries have the ability to run in a system storage protection key (keys 0-7) or Supervisor State. They can also bypass all RACF rules checking or logging, and any other mainframe-based control, if coded to do so. MVS integrity must be assured in order to rely on security software such as RACF, and the most important single factor governing Z/OS integrity is the control over access to APF libraries.

Each installation defines which libraries will be APF libraries by naming them in a special list (the PROGxx member of PARMLIB). In addition, it is also possible to give APF status to LINKLIST libraries.

It goes without saying that UPDATE access to any APF library should be limited to only those individuals with a demonstrable need based on job description. Some Z/OS installations go as far as removing all "standing" UPDATE access to all APF libraries from every user, even technical support personnel. Even READ access is potentially sensitive, since programs that bypass security may already exist in an APF library, and READ access is all that is required to execute them. Access control software like RACF is used to restrict access.

When doing an MVS integrity audit, the first question that should be answered is "Who can update an APF library?" To get a general feel for whether an installation is well controlled, you might ask "What is the total number of userids that have more than READ access to at least one APF library?" In a way, this is the same as asking "How many users have the power to bypass RACF and any other MVS control?"

Unfortunately, answering either of these questions in a RACF environment has always been a rather labor-intensive task. You had to figure out who had access based on special privileges like the OPERATIONS attribute, group membership, Global Access Checking specification, ID(*) specifications, profiles in WARNING mode, libraries with no RACF profile, and so on. The picture can be further complicated when a userid is designated as "RESTRICTED", meaning that the userid (or group containing the userid) must be specifically designated in an access list.

In short, getting answers to the most important and direct questions about Z/OS integrity has always been way too difficult. APFCHECK solves this problem.

APFCHECK

Installation

The APFCHECK program is provided on the distribution CD as an object deck which can be uploaded to the mainframe and then link-edited (with the AC=1 attribute) into an APF library. The CD also contains the same JCL for link-editing, sample execution JCL as follows:

- APFCHECK.OBJ an MVS object deck. Upload in binary format.
- APFCHECK.RUN sample JCL for executing the program.
- APFCHECK.LKE sample JCL to link-edit the object deck.

The JCL files are ASCII text files. They must be converted from ASCII to EBCDIC when they are uploaded. You can view or edit them on your PC before uploading them. The object deck should be uploaded in binary format (no ASCII-EBCDIC translation).

If you are comfortable uploading files and working with simple JCL and the linkage editor, you will have no problem understanding the jobs.

If you are not completely comfortable in this area, you should ask your system programmer to install the product.

You will need a license code to use APFCHECK. This is a single 80-character record which is delivered at the time of purchase.

Since APFCHECK requires RACF SPECIAL or AUDITOR for operation, there is no real harm in placing the load module in a generally-available library. However, we suggest you place it in a private authorized library to prevent unwanted copying of the program. You can further limit the execution of the program through RACF protection over the load library and the program itself.

You can then test APFCHECK. You will need to modify the JOB statement and the STEPLIB statement. The PRIVLIC statement points to a data set containing your license code.

```
//JOBNAME JOB 1,acct info,MSGCLASS=X,NOTIFY=userid
//          EXEC PGM=APFCHECK
//STEPLIB  DD DSN=hlq.apf.library,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//OUTD1   DD DSN=hlq.REPORT1,DISP=SHR
//OUTD2   DD DSN=hlq.REPORT2,DISP=SHR
//OUTD3   DD DSN=hlq.REPORT3,DISP=SHR
//PRIVLIC  DD *
your license code goes here
//PRIVPARM DD *
RUNLIMIT=(0000000,9999999)
USERMASK=(***** )
DFTGROUP=(***** )
RUNTYPE=(QUICK)
```

APFCHECK

Notes on the APFCHECK JCL sample:

- a.** You may or may not need a STEPLIB, depending on whether the load library you select is in the LNKLIST. (If it is, remember to issue an 'F LLA,REFRESH' command.) Using a STEPLIB avoids any complication.
- b.** A license code is needed to use APFCHECK. The PRIVLIC DD statement must point to a data set (or member) containing the license code. The license code is a single 80-character record, of which only the first 72 characters are used. You could use an in-stream data set (//PRIVLIC DD *) or point to a data set.
- c.** Note that you must have RACF SPECIAL or AUDITOR privileges for the userid that submits this job. If you use RACF program protection, you also need the necessary 'permit' in class PROGRAM.

REPORTS

Report 1 Detail

The following figure illustrates the primary, detailed output of APFCHECK. This is the information that will appear in the dataset referenced by the OUTD1 DD card.

As illustrated here, included in detailed reports is environmental information, including the active RACF data set name, license information, time and date the program was executed, and parameters in effect for this run.

In this example, `RUNLIMIT=(0000000,0000500)` was specified, indicating that only the first 500 userids on the RACF database should be analyzed.

APFCHECK

APFCHECK VERSION 2.7.1 - EXPIRES 12/31/2008
COPYRIGHT GOLDIS CONSULTING SERVICES. 2008
THIS PROGRAM IS LICENSED TO: XYZ CORPORATION EVALUATION
LICENSE CODE: A260T21

THIS PROGRAM WAS RUN AT 12:04 ON 02/02/2008
THE SYSID OF THE SYSTEM WE ARE RUNNING ON IS: CPOA
WE ARE RUNNING ON CPU ID 166124, MODEL 3090
THE NAME OF THE RACF DATASET IS: SYS1.RACFDS
INPUT PARAMETERS FOR THIS RUN ARE:
RUNLIMIT=(0000000,0000500)
USERMASK=(*****)
DFTGROUP=(*****)

```
ACCESS--  USERID--  VOLUME DATASET NAME-----  
  
WARNMODE  ADCADM    CSNX1A  CICS.DEVR212.LOADLIB1  
WARNMODE  ADMPRINT  CSNX1A  CICS.DEVR212.LOADLIB1  
ALTER     ADMPRINT  MVSX6S  IMS.IMSP02.RESLIB  
ALTER     ADMPRINT  MVSX6S  IMS.IMSP02.MATRIXA  
ALTER     ADMPRINT  MVSX6S  IMS.IMSP02.MATRIXB  
ALTER     ADMPRINT  MVSX6S  IMS.IMSP02.MODBLKSA  
ALTER     ADMPRINT  SMPF01  IMS.IMSP02.MODBLKSA  
ALTER     ADMPRINT  *SMS*   SYS2.CANDLE.V500.ESYS.RKANMODL  
ALTER     ADMPRINT  *SMS*   SYS2.CANDLE.V500.ESYS.RKANMOD  
UPDATE    ADMPRINT  MVS009  SYSL.DBDC.LINKLIB  
UPDATE    ADMPRINT  MVS009  SYSL.DBDC.LPALIB  
WARNMODE  AEXUUCP    CSNX1A  CICS.SYSR212.LOADLIB1  
WARNMODE  ALPHA     CSNX1A  CICS.DEVR212.LOADLIB1  
WARNMODE  APPC     CSNX1A  CICS.DEVR212.LOADLIB1  
ALTER     APPC     MVSX6S  IMS.IMSP02.RESLIB  
ALTER     APPC     MVSX6S  IMS.IMSP02.MATRIXA  
ALTER     APPC     MVSX6S  IMS.IMSP02.MATRIXB  
ALTER     APPC     MVSX6S  IMS.IMSP02.MODBLKSA  
ALTER     APPC     SMPF01  IMS.IMSP02.MODBLKSA  
ALTER     APPC     MVSX4S  IMS.IMSP02.MODBLKSB  
ALTER     APPC     MVSX6S  IMS.IMST02.RESLIB  
ALTER     APPC     MVSX6S  IMS.IMST02.MATRIXA  
ALTER     APPC     MVSX6S  IMS.IMST02.MATRIXB  
ALTER     APPC     MVSX6S  IMS.IMST02.MODBLKSA  
ALTER     APPC     MVSX6S  IMS.IMST02.MODBLKSB  
ALTER     APPC     *SMS*   SYS2.CANDLE.V500.ESYS.RKANMODL  
ALTER     APPC     *SMS*   SYS2.CANDLE.V500.ESYS.RKANMOD  
UPDATE    APPC     MVS009  SYSL.DBDC.LINKLIB  
UPDATE    APPC     MVS009  SYSL.DBDC.LPALIB  
ALTER     APPC     PPGT6S  SYS2.CANDLE.TLOADLIB  
WARNMODE  APPC     MVS006  CICS.CICS410.TNAT97A.SDFHAUTH  
WARNMODE  BEFIAH1  CSNX1A  CICS.DEVR212.LOADLIB1
```

.
. .
. .

NUMBER OF DIFFERENT RACF IDS WITH APF FOR THIS RUN: 0000448

APFCHECK

Report 2 Detail

This report shows each userid tested during the run, how many APF libraries the USERID can update, whether the userid has any of certain attributes (SPECIAL, OPERATIONS, AUDIT,)

```
APFCHECK VERSION 2.7.1 - EXPIRES 12/31/2008
COPYRIGHT GOLDIS CONSULTING SERVICES. 2008
THIS PROGRAM IS LICENSED TO: XYZ CORPORATION EVALUATION
LICENSE CODE: A260T21
```

```
THIS PROGRAM WAS RUN AT 12:04 ON 02/02/2008
THE SYSID OF THE SYSTEM WE ARE RUNNING ON IS: CPOA
WE ARE RUNNING ON CPU ID 166124, MODEL 3090
THE NAME OF THE RACF DATASET IS: SYS1.RACFDS
```

```
INPUT PARAMETERS FOR THIS RUN ARE:
RUNLIMIT=(0000000,0000500)
USERMASK=(***** )
DFTGROUP=(***** )
```

-USERID-	ACCESSES	S	O	A	USER NAME-----
ADCADM	11				ADMIN ID FOR ITA
ADMPRINT	27				STC - ADMPRINT
AEXUUCP	11				THOMAS, ELLEN
ALPHA	11				APPLICATION 1
AOFARCAT	228	O			STC - AOFARCAT ID
AOIBMP1	11				DB2 RES
APPC	27				STC - APPC (FOR IMS)
ASBATCH	11				MINTON, CAROL
ASCH	27				STC - APPC SCHEDULER
ASCHINT	27				STC - ASCHINT (APPC)
ASBR001	11	S			MARTIN, JOHN E.
ASCZ004	11				LEONARDO, ANTHONY
ASFI001	13				BLOCK, ROBERT
ASFM002	17				BUCUVALAS, DAN
ASU9110	91				PERLMAN, KARL
AUC0184	21		A		CAULFIELD, HOLDEN
.					
.					
.					

NUMBER OF DIFFERENT RACF IDS WITH APF FOR THIS RUN: 0000448

APFCHECK

Report 3 Detail

APFCHECK VERSION 2.7.1 - EXPIRES 12/31/2008
COPYRIGHT GOLDIS CONSULTING SERVICES. 2008
THIS PROGRAM IS LICENSED TO: XYZ CORPORATION EVALUATION
LICENSE CODE: A260T21

THIS PROGRAM WAS RUN AT 12:04 ON 02/02/2008
THE SYSID OF THE SYSTEM WE ARE RUNNING ON IS: CPUA
WE ARE RUNNING ON CPU ID 166124, MODEL 3090
THE NAME OF THE RACF DATASET IS: SYS1.RACFDS

INPUT PARAMETERS FOR THIS RUN ARE:
RUNLIMIT=(0000000,0000010)
USERMASK=(*****)
DFTGROUP=(*****)
RUNTYPE=(QUICK)

ACTIVE APF LIBRARIES AT THE TIME OF THIS RUN WERE:

VOLUME	DATASET NAME-----	-UACC--	WARN	PROFILE
OS3R8A	SYS1.LINKLIB	READ	FAIL	EXISTS
OS39R8	SYS1.SVCLIB	READ	FAIL	EXISTS
WORK01	PGOLDIS.A.LOAD	READ	WARN	EXISTS
OS39R8	CPAC.LINKLIB	READ	FAIL	EXISTS
OS3R8A	SYS1.SERBLINK	READ	FAIL	EXISTS
OS39R8	SYS1.CSSLIB	READ	FAIL	EXISTS
OS39R8	TME10GEM.V1R1M0.SIHSMOD1	READ	FAIL	EXISTS
OS39R8	IGY.V1R2M0.SIGYCOMP	READ	FAIL	EXISTS
OS39R8	REXX.V1R3M0.SEAGALT	READ	FAIL	EXISTS
OS39R8	DIT.V1R3M0.SDITMOD1	UPDATE	FAIL	EXISTS
OS39R8	MQM.SCSQLINK	READ	FAIL	EXISTS
OS39R8	MQM.SCSQAUTH	READ	FAIL	EXISTS
OS39R8	MQM.SCSQSNLE	READ	FAIL	EXISTS
OS39R8	COK.SCOKLINK	READ	FAIL	EXISTS
OS3R8A	SYS1.V2R8M0.SHASLINK	READ	FAIL	EXISTS
OS3R8A	SYS1.V2R8M0.SHASMIG	READ	FAIL	EXISTS
OS39R8	CSF.SCSFMOD0	READ	FAIL	NO PROF
OS39R8	SYS1.SBDTCMD	READ	FAIL	EXISTS
OS39R8	SYS1.SBDTLIB	READ	FAIL	EXISTS
OS39R8	FFST.V120ESA.SEPWMOD2	READ	FAIL	EXISTS
OS39R8	FFST.V120ESA.SEPWMOD1	READ	FAIL	EXISTS
FOLLOWING LIBRARY NOT ON VOLUME IN RACF PROFILE:				
OS3R8A	SYS1.NFSLIB	READ	FAIL	EXISTS
OS39R8	SYS1.ISAMLPA	READ	FAIL	EXISTS
OS39R8	TCPIP.SEZATCP	READ	FAIL	EXISTS
OS39R8	TCPIP.SEZALNK2	READ	FAIL	EXISTS
OS39R8	TCPIP.SEZAMIG	READ	FAIL	EXISTS
OS39R8	TCPIP.SEZADSIL	READ	FAIL	EXISTS
OS39R8	SYS1.SEZALPA	READ	FAIL	EXISTS
OS39R8	SCRIPT.R40.DCFLOAD	READ	FAIL	EXISTS

Report 3 is an overview report that shows the UACC of each APF library, whether any profile governing an APF library is in “WARNING” mode, and whether any APF library lacks a RACF profile. *Libraries that appear as exceptions in this report are not shown in Report1 or Report2.* This report will also indicate when a library in the APF list has a RACF profile, but the VOLUME doesn’t match the one in the APF list.

APFCHECK

Data Sets Used by APFCHECK

The OUTD1, OUTD2, and OUTD3 DD statements refer to datasets that will contain the three reports issued by APFCHECK as described above. These can be sequential datasets or members in a partitioned dataset. The program expects them to have DCB=LRECL=80, RECFM=FB.

PRIVLIC

The PRIVLIC DD statement points to a data set with your license code. APFCHECK will read only the first record from this data set.

PRIVPARM

The PRIVPARM DD statement points to a data set containing various statements that control the operation of the program.. These parameter records all start in column 1 and have a rigid format. Only one record of each of parameter type may be present. The first three parameter types are RUNLIMIT, USERMASK, and DFTGROUP. A typical set of parameters might be

```
RUNLIMIT=(0000000,9999999)
USERMASK=(OPER****)
DFTGROUP=(TEST****)
```

Each of the parameters serves to limit processing in some way. In an installation with a large number of userids and a large number of APF libraries defined, the APFCHECK program could run for quite a long time and potentially produce voluminous output. The control parameters provide a variety of ways to process only a subset of your total userids during a given run of APFCHECK.

Within the RACF database, userid definitions can be considered as an ordered set of records, starting with record zero. If, for example, you have 60,000 userids defined. You could process the first third with:

```
RUNLIMIT=(0000000,0020000)
```

and sometime later you could process the second third with:

```
RUNLIMIT=(0020001,0040000)
```

and so forth. The two parameters for RUNLIMIT must be exactly seven digits, with the equals sign, parenthesis, and comma exactly as shown. An incorrect specification will result in an ABEND (user code 201).

APFCHECK

The USERMASK parameter must have exactly eight characters. If an asterisk is replaced with any other character, then only userids that match the pattern character(s) are tested. For example:

```
USERMASK=( *Y**AB** )
```

would report on only those userids that have a Y as the second character and AB as the fifth and sixth characters. Again, the equals sign and parenthesis must be exactly as shown.

The DFTGROUP provides a similar restriction for the default group defined for each userid. For example:

```
DFTGROUP=( SYS1**** )
```

would test only those userids that have a default group name beginning with SYS1. There is an AND relation between the three parameters. For example,

```
RUNLIMIT=( 0000101, 0000200 )  
USERMASK=( ****PAY* )  
DFTGROUP=( FIN***** )
```

would test the APF access of users that are numbers 101 - 200 in the RACF data base, AND have userids containing PAY in the indicated positions, AND have a default group name beginning with FIN.

The numeric position of a userid in the RACF data base is not normally of interest, and cannot be readily displayed with RACF commands. The purpose of the RUNLIMIT parameter is to somewhat arbitrarily divide a large user population into smaller, more manageable processing runs. We suggest you do not spend time trying to determine the "number" of specific userids.

The parameters:

```
RUNLIMIT=( 0000000, 9999999 )  
USERMASK=( ***** )  
DFTGROUP=( ***** )
```

will process all userids in your RACF data base, since there are no restrictions imposed.

APFCHECK

Other Options

In addition to the three parameter cards described above, a fourth parameter can be specified. The RUNTYPE card can be specified as either

```
RUNTYPE=(OVIEW) OR  
RUNTYPE=(QUICK)
```

When RUNTYPE=(OVIEW) is specified, only Report 3, the Overview report is produced. This allows you to see whether any APF libraries might be specified in a way that allows all users to access them, and also whether an APF library is on a volume different from the one specified in the RACF profile.

We recommend that you first run APFCHECK using this option, since if an APF library has been specified with UACC(UPDATE), for example, there may not be any need for the detailed analysis that the other reports offer. In other words, if you can see at a glance that all users can update a particular APF library, you already have the answer to the question “How many userids can update even one APF library?”

After any obvious mistakes in protection have been addressed, you might run APFCHECK with RUNTYPE=(QUICK). This option reveals all the userids with UPDATE access, but only reports the first library accessible to each user. Using this option can greatly decrease processing time and output volume while revealing any userids that you know shouldn't have access.

When running with RUNTYPE=(QUICK), APFCHECK will also honor any restrictions specified in the other three parameter cards.

APFCHECK

Protection

Protecting the APFCHECK program and its data sets is important. If not properly protected they may provide sensitive security information to unauthorized users. By “protection”, we mean that the data sets should have RACF profiles with UACC(NONE) and as few permitted users as possible. To exclude OPERATIONS users, consider defining their access as NONE.

The APFCHECK object data set, which you uploaded from diskette, is needed only during installation. It should be protected during installation and deleted as soon as installation is complete.

APFCHECK will work only for users with system-level RACF SPECIAL or AUDITOR attributes. RACF *program control* is not needed or verified by the program, but may be used to further reduce the number of legitimate users.

APFCHECK input and output files are potential security exposures and should be protected by UACC(NONE) and a very limited access list. Unless you have stringent spool controls in place, we recommend using normal (protected) data sets instead of SYSOUT data sets. These should have the erase-on-scratch attribute. Remember that APFCHECK produces information you should not share with operators and other users who have unlimited access to SYSOUT spooled data.