Security Definitions

SAF Security is implemented by defining resource classes and profiles and permitting users the necessary access to those profiles. Specific requirements for class and profile definitions and access levels are described in the individual product documentation.

This section describes in general how to define resources to RACF, CA-Top Secret and CA-ACF2.

- Defining Resources to RACF
- Defining Resources to CA-TOP SECRET
- Defining Resources to ACF2

Defining Resources to RACF

This section describes how the resources are defined to RACF. For exact details of the procedures to be followed for the installed RACF version, consult the relevant IBM manuals.

Overview of tasks

- Add classes to Class Descriptor Table
- Update z/OS Router Table
- Activate new classes
- Assign user ID for the SAF Security Started Task
- Permit user access to resource profiles

To add classes to Class Descriptor Table

1. Add the resource classes to the RACF Class descriptor table. Refer to the *IBM SPL RACF* manual. For an example, see IBM SYS1.SAMPLIB, member RACINSTL.
2. For flexibility, allocate maximum length for the classes (80).
3. Define the classes to enable discrete and generic profile use.
4. Check further attributes controlling the level of RACF messages generated when performing RACROUTE calls, as well as the required level of SMF recording. Sample definitions are provided in source members SAFRCLSN and SAFRCLSX.

To update the z/OS Router Table

- Update the z/OS router table as described in the *IBM SPL RACF* manual. For an example, see the IBM SYS1.SAMPLIB, member RACINSTL, section RFTABLE.

To activate new classes

- Activate new resource classes with SETROPTS (see *IBM RACF Command Language Reference* manual). For an example, activate class NBKSAG:

```
SETROPTS CLASSACT(NBKSAG)
SETROPTS GENCMD(NBKSAG)
SETROPTS GENERIC(NBKSAG)
```

To assign user ID for the SAF Security Started Task

- The SAF Security Kernel runs either in its own Started Task or in an Adabas or Entire Net-Work started task. Assign a user ID to these jobs with the relevant RACF authorizations, including the ability to perform `RACROUTE, TYPE=EXTRACT, TYPE=AUTH` and `TYPE=VERIFY` calls on profiles belonging to the defined classes.

To permit user access to resource profiles

- After adding profiles to protect the different resources, permit users the required level of access, using the relevant RACF Commands. The following example adds resource profile ETB.POLICY.QUOTE1 and grants READ access to user USER2 and CONTROL access to USER3. USER2 represents a client and requires READ access to execute while USER3 represents a server component that needs CONTROL access to register:

```
RDEFINE NBKSAG ETB.POLICY.QUOTE1 UACC(NONE)
PERMIT ETB.POLICY.QUOTE1 CLASS(NBKSAG) ACCESS(READ) ID(USER2)
PERMIT ETB.POLICY.QUOTE1 CLASS(NBKSAG) ACCESS(CONTROL) ID(USER3)
```

Defining Resources to CA-TOP SECRET

This section describes how the resources are defined to TOP SECRET. For exact details of the procedures to be followed for the installed version of TOP SECRET, consult the relevant CA-TOP SECRET manual.

Overview of tasks

- Add CA-TOP SECRET Facility
- Assign user ID for the SAF Security Started Task
- Add procedure name for the Started Task
- Add resource type to Resource Definition Table
- Assign ownership of resources
- Permit defined resources to Users

To add CA-TOP SECRET facility

- CA-TOP SECRET enables a set of authorization checks to be made against a certain facility. For example, this can be used to secure the development environment SAGDEV separately from the production environment SAGPROD. Alternatively, a default facility of batch can be used.

When adding additional facilities, use the following attributes:

```
AUTHINIT,MULTIUSER,NONPWR,PGM=ADA,NOABEND
```

To assign a user ID for the SAF Security Started Task

- Add one user ID for each instance of the SAF Security Started Task.

  If required, different facilities can be assigned to development and production tasks.

  The designated facility is assigned to the Started Task user ID:

  ```
  TSS CRE(user-id) DEPT(dept) MASTFAC(fac)
  ```

To add a procedure name for the SAF Security Started Task

- The procedure name under which the SAF Security Started Task executes must be defined to CA-Top Secret. Different procedure names are suggested when securing different environments separately with the use of non default CA-Top Secret facilities:

  ```
  TSS ADD(STC) PROC(proc) USER(user-id)
  ```

To add resource types to Resource Definition Table

- Add the resource types to the CA-TOP SECRET Resource Definition Table (RDT). Below is an example for resource type NBKSAG. Refer to the CA-TOP SECRET Reference Guide for a detailed explanation of the following commands and arguments:

  ```
  TSS ADD(RDT) RESCLASS(NBKSAG)
  RESCODE(HEXCODE)
  ATTR(LONG)
  ACLST(NONE,READ,CONTROL)
  DEFACC(NONE)
  ```

To assign ownership of resources

- Assign ownership to a particular resource as shown in the following example. This must be done before permitting access to defined resource profiles:

  ```
  TSS ADD(user1) NBKSAG(ETB.POLICY.QUOTE1)
  ```

  This makes user user1 the owner of the Broker service etb.policy.quote1.

To permit defined resource to users

- Permit access to a resource profile as in the following example. In the example, user user2 is permitted READ access to the Broker service etb.policy.quote1. This enables the user to execute as a client and issue requests to this Broker service:

  ```
  TSS PER(user2) NBKSAG(ETB.POLICY.QUOTE1) FAC(fac) ACCESS(READ)
  ```

Defining Resources to ACF2

This section describes the definition of resources to ACF2 versions 5 and 6. For details of the procedures required for the current software version, please consult the relevant ACF2 manual.

Note:
ACF2 provides insufficient return codes to determine whether a resource profile does not exist or simply the user does not have access to it. Therefore, if access is denied by ACF2, the SAF Security Kernel will always report "Access denied resource not allowed" in the error message.

Consequently the SAF Security configuration options such as BKUNI=Y to allow access to undefined resources are not applicable where ACF2 is used.

To define resources to ACF2 version 5

1. The SAF Security Kernel executes as a normal started task in z/OS. Define the user ID of the server task to ACF2 with the following attributes:

   ```
   MUSASS,NON-CNCL,STC
   ```

   To avoid the NON-CNCL attribute, APAR TW95626 must be applied.

2. Activate the SAF Interface using the command:

   ```
   GSO OPTS -  SAF
   ```

3. Switch off all SAF checks by inserting the SAFSAVE record as follows:

   ```
   SAFSAVE CLASSES(-) CNTLPTS(-) SUBSYS(-)
   ```

4. Switch on the SAF security checks for the SAF Security Kernel by inserting the SAFPROT record as follows:

   ```
   CLASSES(-) CNTLPTS(-) SUBSYS(ADARUN)
   ```

5. For the general resource class name used by SAF Security product options, define a 3-character ACF2 resource type code by inserting a SAFMAPS record as follows:

   ```
   SAFMAPS MAPS(NBK/NBKSAG)
   ```

6. Define the required resource profiles to ACF2 using the new type code.

   The following example shows the addition of a Broker service etb.policy.quote1, allowing READ access for USER2:

   ```
   $KEY(ETB.POLICY.QUOTE1) TYPE(NBK) UID(user2) ALLOW SERVICE(READ)
   ```

To define resources to ACF2 version 6 and above

1. The SAF Security Kernel executes as a normal started task in z/OS. Define the user ID of the server task to ACF2 with the following attributes:

```
MUSASS,STC
```

ACF2 version 6.1 and 6.2 no longer require TW95626,as these versions are more SAF-compliant.

2. Insert SAFDEF records as follows:

```
SAFDEF.EXS1
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=VERIFY SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)

SAFDEF.EXS2
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=AUTH SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)

SAFDEF.EXS3
FUNCRET(4) FUNCRSN(0) ID(ENTIREX) MODE(GLOBAL)
RACROUTE(REQUEST=EXTRACT SUBSYS=ADARUN REQSTOR=-)
RETCODE(4)
```

3. For the general resource class names used by SAF Security product options, define a 3-character ACF2 resource type code by inserting a CLASMAP record as follows:

```
CLASMAP
ENTITYLN(0) MUSID() RESOURCE(NBKSAG) RSRCTYPE(NBK)
```

4. Define the required security profiles to ACF2 using the new type code. The following example shows the addition of a Broker service etb.policy.quote1, allowing READ access only for user ID user2:

```
$KEY(ETB) TYPE(NBK)
POLICY.QUOTE1 UID(user2) SERVICE(READ)   ALLOW
POLICY.QUOTE1 UID(-)                     PREVENT
```