Installation

This section describes how to install the SAF Security Kernel.

This chapter covers the following topics:

- Prerequisites
- Preparing for Installation
- Authorization
- Modes of Operation
- Installation Datasets
- Installation Procedure
- Embedded SAF Security Kernel
- Installing the SAF Security Daemon
- Daemon Configuration

Prerequisites

The following are prerequisites:

- z/OS
- SAF-compliant security system
- For Adabas SAF Security (AAF) installations, the SAF Security Kernel supplied in the shared Adabas 8.2 SP4 library (WAL 8.2 SP4 patch level 2, or WAL824P002) requires Adabas SAF Security 8.2 SP2 or later; it is not compatible with previous versions of Adabas SAF Security.

Preparing for Installation

Before installing the SAF Security Kernel, review all possible configuration options for the kernel itself and for the product(s) it will secure.

If the kernel will execute as a daemon, in its own address space, allocate a unique node number to it.

Authorization

The kernel load library and any other step libraries in the kernel’s loading environment must be APF authorized.

Modes of Operation

The kernel may be embedded with a product (that is, it may run in the same address space). This is the case for Adabas and Entire Net-Work. To implement this mode of operation, you simply need to add the kernel load library (and any load libraries used as the target of installation assembly and link jobs) to the step library concatenation, ensuring that they are APF authorized.

For products other than Adabas and Entire Net-Work, the kernel operates under a daemon, in its own address space as a target in the Software AG network. This mode of operation is described in more detail below.

For both modes of operation, the SAF Security Kernel must run under a defined user ID. This user ID must have sufficient authority to invoke the AUTH, VERIFY, and EXTRACT functions of RACROUTE and to issue third-party checks on behalf of all users.

Installation Datasets

The SAF Security Kernel is supplied as a component of the Adabas Limited Libraries

WAL*vrs*.LOAD

WAL*vrs*.LOAD is a standard load library containing modules needed to operate the SAF Security Kernel.

This library must be APF-authorized and available on the loading environment of any job that uses the SAF Security Kernel. Jobs that include the SAF Security Kernel are:

- The SAF Security Daemon, used by Natural SAF Security and EntireX Security
- Adabas nuclei protected by Adabas SAF Security
- Entire Net-Work nodes protected by Entire Net-Work SAF Security

The WAL*vrs*.LOAD modules for SAF Security all have names beginning with SAF.

WAL*vrs*.SRCE

WAL*vrs*.SRCE is a standard source library containing Assembler macros (names beginning NA2M) and source books (SAFCFG, SAFPOS and SAFPSEC) which must be assembled as part of the SAF Security Kernel installation. There are also several example members:

SAFAEXT	CA-ACF2 extract for Natural RPC and program protection
SAFRCLSN	RACF class definitions for Natural SAF Security
SAFRCLSX	RACF class definitions for EntireX Security
SAFTEXT	CA-Top Secret extract for Natural RPC and program protection
SAFDDCAR	Daemon DDCARD input
SAFPARMS	Sample SAFCFG

WAL*vrs*.JOBS

WAL*vrs*.JOBS is a standard source library containing example jobs for installing the SAF Security Kernel. These examples have names beginning SAF.

Installation Procedure

This section describes how to install the SAF Security Kernel.

Step 1 Assemble the Configuration Mode

The configuration module defines the required installation options. Only general options are described here. For information about product-specific options, see the relevant product documentation. A sample job is provided in SAFI010 in the jobs library.

The configuration module is created by assembling a source member similar to the SAFPARMS member supplied on the source library. This source member invokes the SAFCFG macro, (also supplied on the source library), specifying your site-specific options and requirements. The SAF Security Kernel uses the settings in SAFCFG to determine:

- Which resources are protected for which products
- Security classes to be used for resource checking
- How resource profile names are constructed
- Caching requirements

The resulting load module, SAFCFG, must be available to any job that includes the SAF Security Kernel and, in the case of EntireX, to the jobs being secured. You may decide to maintain different SAFCFG modules for different secured products. However, it is critical that the daemon use exactly the same configuration module as EntireX jobs secured by that daemon.

Set the following parameters to the appropriate values:

GWDBID=nnnnn	Node ID of SAF server
GWSIZE=nnnnn	Buffer size in K (approximately 512 bytes per user)
GWMSGL={0, 1 ,2,3}	Message level
GWSTYP={ 1 ,2,3}	Security repository type
SAFPRINT={ N ,Y}	Write trace messages to DDPRINT (N) or SAFPRINT (Y)

Message level indicates which diagnostic messages will be written to DDPRINT or SAFPRINT:

1 (the default)	only security violations are traced
2	only successful checks are traced
3	all checks are traced
0	tracing is suppressed

Security repository type identifies the SAF security system in use:

1 (the default)	RACF
2	CA-Top Secret
3	CA-ACF2

SAFPRINT specifies where security check trace messages should be written:

N (the default)	DDPRINT
Y	SAFPRINT

If you specify Y, but do not provide a SAFPRINT dataset, the trace messages will be written to DDPRINT. The SAFPRINT dataset must be defined in the JCL and may refer to a SYSOUT dataset or to a file defined with `RECFM=F` (or FB) and `LRECL=121`.

Step 2 Assemble the RACROUTE Macros

The SAF Security Kernel requires the same version of the RACROUTE macros as used at the customer site. Sample job SAFI020 is provided to assemble the module containing these macros.

Before running SAFI020, set the parameter `STY` to RACF, TSS or ACF2 as appropriate. The `REL` parameter specifies the RACROUTE macro `RELEASE` parameter used by SAFPSEC. Unless advised otherwise, specify `REL=2.1` (the default).

The resulting load module, SAFPSEC, must be available to any job that includes the SAF Security Kernel.

Step 3 Assemble the Operating System Services Module

Sample job SAFI021 is provided to assemble the operating system services module, SAFPOS. The resulting load module, SAFPMAC, must be available to any job that includes the SAF Security Kernel.

Embedded SAF Security Kernel

For those products (Adabas and Entire Net-Work) that use an embedded SAF Security Kernel, you need only add the load library containing the kernel (SAFKRN) and the three load modules created above to the step library concatenation.

Installing the SAF Security Daemon

For those products (Natural and EntireX) that need a SAF Security Kernel running in a separate, authorized address space, you must install a SAF Security Daemon.

The SAF Security Daemon runs in its own address space, using Adabas modules to establish inter-process communication. It signs on to the Adabas SVC as a target and is therefore accessible in the same way as an Adabas database. Consequently, the SAF Security Daemon (and its Kernel) can be accessed remotely, via Entire Net-Work.

Software AG recommends that you run the SAF Security Daemon as a started task, although it may be run as a batch job. The SAF Security Daemon must run APF-authorized, therefore all step libraries must be APF-authorized.

Additionally, the SAF Security Daemon must run under a userid with sufficient authority to invoke the RACROUTE AUTH, EXTRACT and VERIFY functions and to make third-party checks on behalf of other users.

Sample JCL to execute the daemon is provided in SAFI024 in the jobs library.

Daemon Configuration

The daemon is configured by parameter input. The parameters are read from the DDCARD dataset at startup. An example dataset is provided in SAFDDCAR in the source library. Following is a description of valid parameters, with default value and meaning.

Parameter	Default	Meaning
`NODE`	None	Identifies this SAF Security Daemon. Must be a number between 1 and 65535 and must be unique among all targets.
`PRODUCT`	None	Defines which products are available in this server. Specify SAF.
`FORCE`	NO	Defines whether or not an existing ID table entry for the same node should be overwritten. Valid values are YES and NO. Specify YES only when advised to by Software AG.
`LOCAL`	NO	Defines whether or not this server is to be accessible from remote users, via Entire Net-Work. Valid values are YES (the server is not accessible) and NO (the server is accessible).
`NC`	20	Defines the maximum number of concurrent requests that can be processed by the server. Specify a number between 1 and 32767. If a request to the server fails with response code 151 (ADARSP151), increase NC.
`NABS`	16	Defines the number of 4K storage blocks to be used for transmitting information between clients and the server. Specify a number between 1 and 32767. If a request to the server fails with response code 255 (ADARSP255), increase `NABS`.
`LU`	65535	Defines the maximum total length of data for a request to the server. Do not change this parameter value unless advised to by Software AG.

Parameter	Default	Meaning
`TIMER`	10	Defines how often the server is to wake up and look for work (note that the server wakes up anyway whenever it receives a request or operator command). Specify a value in seconds.
`CT`	60	Defines how many seconds the server will allow for a client to accept a completed request. If the client fails to acknowledge receipt of the request within this time, the server issues an ADAM93 USER GONE message and the client receives response code 254 (ADARSP254). If you get response code 254 (ADARSP254) frequently, increase the value of CT (the maximum is 32767) and also of `NC` and `NABS`.
`SVC`	0	Defines which SVC number is to be used. Specify your Adabas SVC.
`MPMWTO`	NO	Defines whether the server should send informational messages to the operator console or not. You should specify YES until you are satisfied that the server is operating correctly.
`DEFAULT`	None	Defines the default product to which requests will be passed. Specify SAF.
`SAF PARM`	SAFCFG	If you need to change the name of the configuration module (for example, you have different configuration modules with different settings), you can specify the name of the configuration module the daemon is to use. For example: `SAF PARM=CFGDAEM` .