Sample Security Exits for Broker Security

Sample security exits are a user-written security solution for use only in exceptional processing situations. Example: If your organization wants to access its own user-written security system when operating EntireX Broker.

This chapter covers the following topics:

- Sample Security Exits as Alternative Security Solution
- Major Advantages of EntireX Security
- Lightweight USRSEC
- Implementation of Sample Security Exits
- Definition of Terms

Note:
See *Using Sample Security Exits for Broker Security*, which describes implementation issues and how to use sample security exits on the operating where Broker executes.

See also *Security Solutions in EntireX*.

Sample Security Exits as Alternative Security Solution

Software AG intends security supplied by EntireX Broker to be only an alternative to EntireX Security, which is Software AG's standard security solution and shipped with EntireX. See *Overview of EntireX Security*. Do not mix these two security solutions: do not use a stub secured with a sample exit against a kernel secured with EntireX Security or vice versa.

Most organizations that use Software AG's EntireX choose EntireX Security instead of sample security exits for EntireX Broker security. If your organization is deploying distributed computer systems encompassing mainframe, UNIX and Windows environments, you will use EntireX Security instead of sample security exits for EntireX Broker security.

Major Advantages of EntireX Security

Comprehensive Security

EntireX Security provides comprehensive security for EntireX Broker:

- user authentication
- user authorization
- application-data encryption

- supplied in object code only

Protection of Application Systems

EntireX Security protects client and server and publish and subscribe application systems, and, in most installations, EntireX Security operates without altering runtime applications.

One User=One Definition

EntireX Security allows your organization to control the use of all applications, including distributed components, from a central point, enabling flexible control with a "one user = one definition" approach.

No User Exits to Write/Debug

There are no user exits to write and debug when using EntireX Security. Compare *Sample Security Exits for Broker Security*.

Standard Security Definitions

EntireX Security enables security definitions, based on class/name/service (client and server) or topic (publish and subscribe), to be validated within your SAF Security system. All definitions are managed using existing security procedures and software.

Protected Investment in SAF-based Security Repositories

Your investment in SAF-based security repositories is protected. This includes not only the security systems RACF, CA ACF2 and CA Top Secret, but also the infrastructure to administer security profiles.

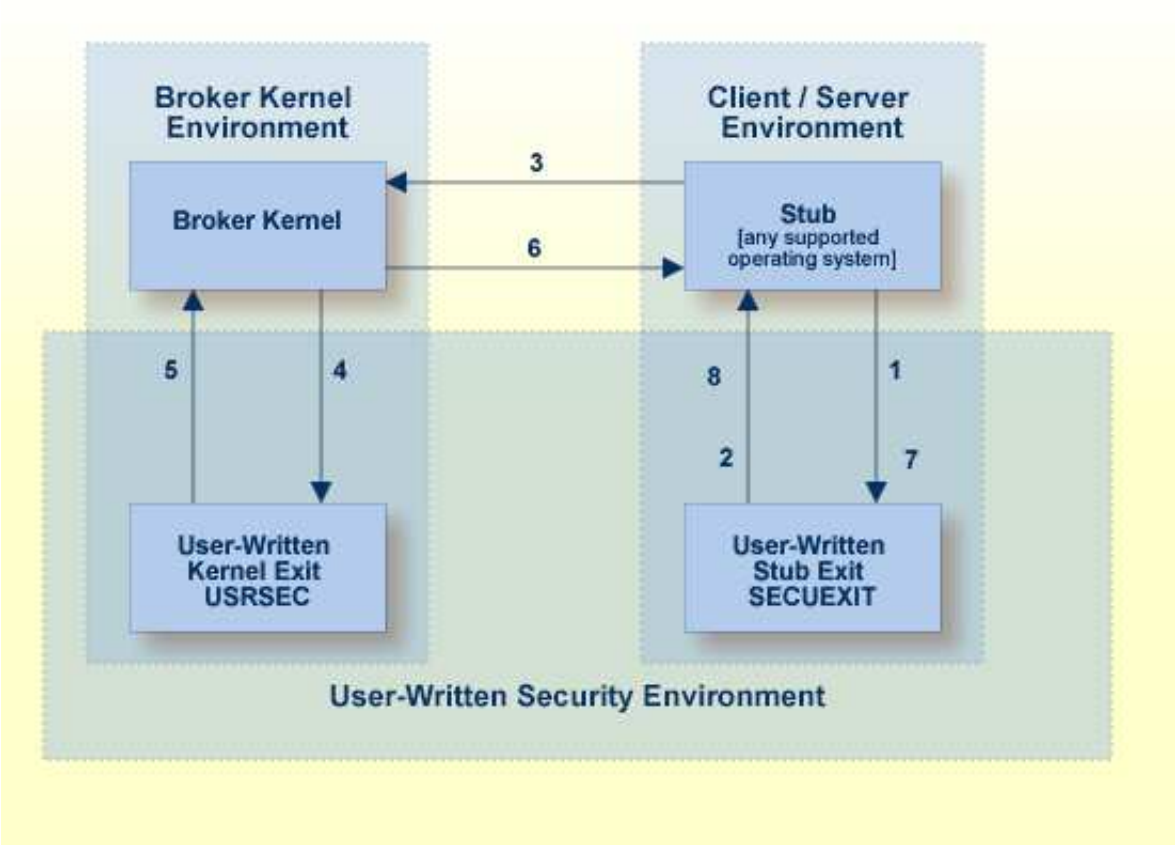
Lightweight USRSEC

For compatibility with previous versions (API level 3 and below), a "lightweight" security exit is supplied in load module USRSEC in library EXX970.LOAD for Broker under z/OS. This "lightweight" version of USRSEC performs authentication only against RACF, CA ACF2 and CA Top Secret. It does not include the full functionality of the standard EntireX Security installation of USRSEC (e.g. resource authorization, etc.). The "lightweight" version of USRSEC does not require any security components, i.e. SECUEXIT, to be installed into the application (stub) environment. If you are using ACI version 1 to 7 and you intend to use the "lightweight" version of USRSEC, please ensure you do not have any security components installed into the application (stub) environment.

Note:
You cannot use the SNMP support provided by System Management Hub in conjunction with the "lightweight" version of USRSEC.

Implementation of Sample Security Exits

Sample security exits are a user-written security solution for use only in exceptional processing situations. The diagram below depicts the data flow which users can implement in their own user exits for Broker security. In this example, the Broker kernel is located on z/OS.



Note:
To activate your user-written security exits, specify `SECURITY=YES` in the broker attribute file.

Description of Steps in Data Flow

1. Broker stub calls security exit *`SECUEXIT`*, if present.
2. Security exit *`SECUEXIT`* encrypts the password and optionally the application data. See *Encryption / Decryption*. `SECUEXIT` accesses the ACI control block and the `SEND/RECEIVE` buffers. `SECUEXIT` returns call to the broker stub.
3. Broker stub communicates the call to the broker kernel.
4. Broker kernel calls security exit *`USRSEC`* for each specific event type:
   - Create security context for user; authentication is usually performed in this event. See *Authentication*.
   - Destroy security context for user.
   - Perform authorization for server to register a service. See *Authorization*.
   - Perform authorization for client to send request.
   - Perform encryption of application data.

- Perform decryption of application data.
- Perform optional processing if a user acquires a new physical user ID. Re-authentication can also be performed.
- Perform optional processing if the value of a user's ACI security token changes. Re-authentication can also be performed.

5. Security exit *USRSEC* passes call to broker kernel.
6. Broker kernel communicates the call to the broker stub of the partner application.
7. The broker stub calls SECUEXIT. SECUEXIT determines whether decryption is to be performed, if correspondingly coded by user.
8. Security exit SECUEXIT returns call to broker stub.

Definition of Terms

- Authentication
- Authorization
- Broker and Kernel
- Broker Stub
- Encryption / Decryption
- Exits

Authentication

Authentication verifies whether the identity specified by the user ID in the ACI control block is the actual identity. Authentication is usually performed by checking the user's ID and password against a security system. The details of this check are specific to the specific operating system and security system.

Authentication is not needed with every call. It is required when the user's security context is created within the Broker kernel; it is also required, optionally, if the user's physical user ID or ACI security token changes.

Authorization

Authorization can be performed when:

- a client issues a request to a service in the case of the first SEND command in a conversation, or of each SEND command if CONV-ID=NONE;
- a server registers a service to the Broker;

Broker and Kernel

It is the location of the Broker kernel that determines the point at which the authentication and authorization checks can be performed. *Authentication* and *Authorization* can be performed in the kernel exit USRSEC. Encryption/decryption can be performed in the kernel exit USRSEC (as well as in the stub exit SECUEXIT).

See *List of Components per Platform* for where Broker kernel is supported.

Broker Stub

In EntireX Broker, a module that implements the ACI (Advanced Communication Interface) is commonly referred to as broker stub or stub. Stubs are installed on the client and the server side.

See *Platform Coverage* for where Broker stubs are supported.

Encryption / Decryption

Encryption is the process by which the information or data being sent back and forth between two computers (including the password submitted when logging on) is encoded, shielding it from view by unauthorized persons. With EntireX, the algorithms for encryption/decryption must be present in both the Broker stubs and in the Broker kernel.

In the case of user-written security exits, encryption/decryption must be implemented in:

- the stub security exits (SECUEXIT or ETBUPRE / ETBUEVA);
- the kernel security exit (USRSEC).

See Encryption of Application Data.

Exits

- **Kernel Exit USRSEC**
  USRSEC is the name of the security exit which is invoked if SECURITY=YES is specified in the attribute file.

  In the case of user-written security exits, this exit will include functionality for authentication, authorization and encryption/decryption.

  See *Platform Coverage* for where Broker kernel is supported.

- **Stub Exit SECUEXIT**
  SECUEXIT is the stub security exit for use with the Broker C-based stub. This module is executed during a Broker command if SECUEXIT is present in the path of execution.

  In the case of user-written security exits, this exit will include functionality for encryption/decryption.

- **Stub exit ETBUPRE / ETBUEVA**
  ETBUPRE / ETBUEVA are the stub security exits for use with the Broker Assembler stub. These modules are executed during a Broker command if they are linked to the Assembler stub.