IBM Education Assistance for z/OS V2R3

Safe Cross-Memory (XM) Post
Element/Component: Supervisor

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- **Describe the Safe Cross-Memory Post service (Safe XMPost) IEAMSXMP**

Overview

- **Problem Statement / Need Addressed**
  - **Problem: Misuse of cross-memory post can lead to exposures if the target address space is recycled or if a new job starts (consider setting the “post bit” in a word in the wrong address space or the wrong job)**
- **Solution**
  - **Provide a new service and exploit it**
  - **Inform the ISV's of the new service**
- **Benefit / Value**
  - **Avoid exposure, because now it will either do the post successfully or know enough not to do it at all**

Usage & Invocation
Where things are now

- **POST macro supports “XM Post” to post an ECB in another address space. The ASCB is used to identify the target address space**
- **Problem: insufficient serialization to be able to know that the ASCB is the intended one (perhaps it was reused) or (for an initiator where the ASCB is the intended one) that the same job is still running**
- **Problem: for a TSO/E address space that is running an authorized request, waking up the “unauthorized side” could be a problem**

Safe XM Post Protocol

- **(User) Capture a TCB Token (TTOKEN) representing a valid task of the waiter**
- **(the Service) Schedule an SRB to the space identified by the TTOKEN (hence using the STOKEN)**
- **(the Service) Validate the TTOKEN**
- **(the Service) POST from the SRB or, when a TSO/E authorized request is in process, schedule an IRB to run on top of the target task and do the POST from there**

IEAMSXMP

- **New service (macro) IEAMSXMP being introduced by APAR OA49677 (available as of June 2016)**
- **Service available when new field ECVTSXMP is non-0. Macro does not check this. If not post-z/OS 2.2 and not req'ing OA49677, you must check this.**

IEAMSXMP

- **Requires supervisor state. Any PSW key.**
- **User provides: ECB address, post code, TTOKEN (and STOKEN in one case where the “first task” in the address space is requested), 512-byte workarea for the system target routine to use**
- **Also optionally: IRB (yes or no), storage key to use for ECB validation, purge ASID and purge TCB for termination considerations, an exit routine address to get control at a specific point in the validation (and 16-byte parameter to provide to the exit routine),**

IEAMSXMP

- **If TCB Token validation fails, that is an ECB-owner error. The POST is skipped (there is nothing to wake up). The exit routine will be given control.**
- **If the IRB cannot “do its thing”, CALLRTM TYPE=ABTERM is issued against the target task (e.g, the IRB could not obtain 16 bytes of storage that it needs). This is abend x'3C4' reason x'1C'.**

IEAMSXMP exit routine

- **Environment: key 0, supervisor state, P=H=S, AMODE 31**
- **Gets control in 4 cases (with indicator in reg 0 of which case)**
  - **TCB Token validation succeeds, IRB=NO in effect. Under the SRB before the POST. Indicates if POST is to be done or not. Local lock held (do not release).**
  - **TCB Token validation succeeds, IRB=YES in effect. Under the IRB before the POST. Indicates if POST is to be done or not. Local lock held (do not release).**
  - **TCB Token validation fails. Local lock held (do not release).**
  - **PurgeDQ. No locks held.**

IEAMSXMP example 1

```
* Code to avoid invoking IEAMSXMP if field
* ECVTSXMP is 0
 ...
* Code to put the address of the ECB into register n
 ...
* Invoke IEAMSXMP
         IEAMSXMP ECB=(n),POSTCODE=pc,                          *
               TTOKENTYPE=ANY,TTOKEN=tt,                        *
               KEY0TO15=0,                                      *
               RETCODE=LRETCODE,RSNCODE=LRSNCODE,               *
               WORKAREA=wa,MF=(E,SXMPL)
* Here you would place code to process the return and
* reason codes.
 ...
DYNAREA  DSECT
         DS    0D
wa       DS    CL512
pc       DS    F
tt       DS    CL16
LRETCODE DS    F
LRSNCODE DS    F
         IEAMSXMP MF=(L,SXMPL),PLISTVER=MAX
```

IEAMSXMP example 2

```
* Code to avoid invoking IEAMSXMP if field
* ECVTSXMP is 0
 ...
* Code to put the address of the ECB into register n
 ...
* Invoke IEAMSXMP
         IEAMSXMP ECB=(n),POSTCODE=pc,                          *
               TTOKENTYPE=FIRSTTASK,STOKEN=st,                  *
               KEY0TO15=0,                                      *
               RETCODE=LRETCODE,RSNCODE=LRSNCODE,               *
               WORKAREA=wa,MF=(E,SXMPL)
* Here you would place code to process the return and
* reason codes.
 ...
DYNAREA  DSECT
         DS    0D
wa       DS    CL512
pc       DS    F
st       DS    CL8
LRETCODE DS    F
LRSNCODE DS    F
         IEAMSXMP MF=(L,SXMPL),PLISTVER=MAX
```

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None.
- Exploiters
  - There are IBM exploiters; there likely are ISV exploiters

Migration & Coexistence Considerations

- None

Installation

- No unique considerations

Session Summary

- Safe XM Post service IEAMSXMP is available

Appendix

Publications:

- MVS Authorized Assembler Services Reference