z/OS System Integrity: Authorized Software without Security Holes

Karl Schmitz, IBM Corporation, Poughkeepsie NY
kdsch@us.ibm.com
February 25, 2003
SHARE Session Number: 2889

z/OS System Integrity

- Definition
- Guidelines
- Examples

z/OS System Integrity: Definition

- Property of a system that prevents unauthorized users from circumventing or disabling protection mechanisms.
- In z/OS, unauthorized programs may not:
  - Bypass store or fetch protection
  - Bypass password/RACF protection
  - Obtain control in an authorized state

z/OS System Integrity: Guidelines

- Creating Predictable Interfaces
- Dealing with User Supplied Storage
- Dealing with User Supplied Values
- Dealing with User Supplied System Control Blocks
- Protecting Data
- Bypassing Authorization Requirements
- Serializing Resources

z/OS System Integrity: Predictable Interfaces

- Interfaces between unauthorized and authorized programs must behave predictably.
- This includes intended and unintended interfaces.

z/OS System Integrity: User Supplied Storage

- Only read or write caller supplied storage in the key of the caller.
  - MVCK
  - MVCSK/MVCDK
  - MODESET to caller's key
  - MVCP/MVCS
- When possible, stay out of PSW key 0.

z/OS System Integrity: User Supplied Values

- Don't trust user supplied values.
- Verify that values are legitimate.
- Check for 0, negative, too small, too large, uneven multiple.
- Beware of changing values. When possible copy to protected storage.
- Ensure consistency between values.

z/OS System Integrity: User Supplied Control Blocks

- Verify system control blocks through trusted pointers in system key storage.
- Serialize.

z/OS System Integrity: Protecting Data

- Authorized programs must protect data from unauthorized programs that run concurrently.
- User data must be protected from other users.

z/OS System Integrity: Bypassing Authorization Checks

- Services that bypass security checks must be restricted to authorized callers.
- Callers allowed to bypass normal security checks must provide equivalent controls.
- Don't create "authorization" SVC or PC routines.

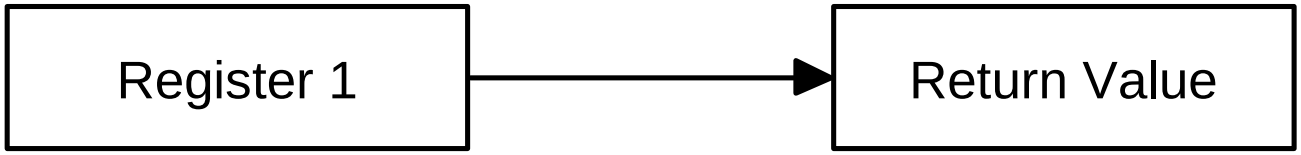
z/OS System Integrity: Serializing Resources

- Access to shared resources must be controlled through serialization.
- Serialization technique must be restricted to authorized programs.

z/OS System Integrity: Exposure Examples

- SVC routines
- Supervisor state
- PSW key 0
- Register 0: Set by caller
- Register 1: Set by caller
- All examples contain exposures
- Do not use this code!

z/OS System Integrity: Exposure Example 1



z/OS System Integrity: Exposure Example 1

```
.
          MODESET EXTKEY=RBT234  to caller key
          OC      0(4,R1),0(R1)  test access
          MODESET EXTKEY=ZERO    back to key zero
.
<calculate value and save in R5>
.
          ST        R5,0(R1)     return result
.
```

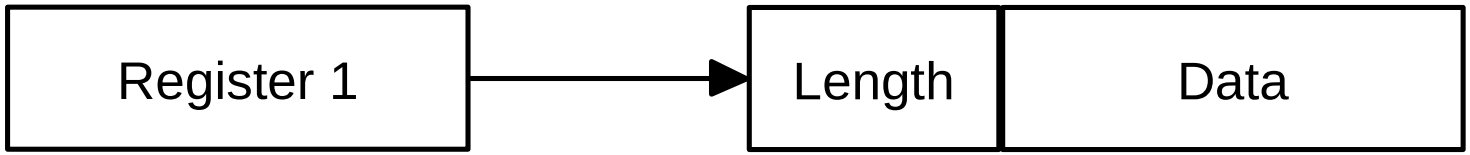
z/OS System Integrity: Exposure Example 2

```
 * Verify that TCB address in R1
            IVSK      R5,R1        Get key of TCB
            LTR       R5,R5
            BNZ       ERROR        Key must be zero
            USING     TCB,R1
            CLC       =C'TCB ',TCBTCBID
            BNE       ERROR        No eyecatcher?
* Update Imaginary User Field of TCB
            ST        R0,TCBXXXX
```

z/OS System Integrity: Exposure Example 3

```
IGC00ATH     CSECT                          Bad Auth SVC
             BALR      12,0
             USING     *,12
             L         2,28(5)              Caller's RB
* Resume address < Beginning of PLPA
             CLC       21(3,2),361(3)
             BL        RETURN
             L         2,180(4)             JSCB
* R0 != 1 request auth off
             BCT       0,AUTHOFF
AUTHON       OI        236(2),X'01'         Set JSCBAUTH
             B         RETURN
AUTHOFF      NI        236(2),X'FE'         Clear JSCBAUTH
RETURN       BR        14
             END       IGC00ATH
```

z/OS System Integrity: Exposure Example 4



z/OS System Integrity: Exposure Example 4

```
LR         R4,R1              Save addr of parms
L          R5,0(R4)           Get length of parms
GETMAIN    RU,LV=(R5)         For copy of parms
LR         R3,R1              Save address in R3
MODESET    EXTKEY=RBT234      Switch to caller key
MODESET    EXTKEY=ZERO,SAVEKEY=(2) R2 = caller key
LR         R1,R2              R1 = caller key
BCTR       R0,0               Subtract 1 from length
MVCSK      0(R3),0(R4)        R0=Length-1, R1=key
<Process and update the copied parms>
L          R0,0(R3)           Obtain length of parms
LR         R5,R0              Length for FREEMAIN
BCTR       R0,0               Subtract 1 from length
MVCDK      0(R4),0(R3)        R0=Length-1, R1=key
FREEMAIN   RU,LV=(R5),A=(R3) Free copy of parms
```

z/OS System Integrity: Example Answers

- Example 1
  - Caller supplied storage should only be referenced in the key of the caller. Key of storage may have changed between check and use.
- Example 2
  - Control blocks should only be trusted if they can be validated using trusted pointers. A counterfeit TCB could reside in key 0 storage and contain the data required to pass these tests.
  - Should not reference a control block until after it has been verified.
- Example 3
  - Don't write services that make unauthorized programs become authorized. This service may be used by a malicious user to obtain authorization.

z/OS System Integrity: Example Answers

- Example 4
  - The length to use in copying the caller's data is retrieved from the caller's storage while running in key 0. All references to caller supplied storage should be in the key of the caller.
  - Value of length should be validity checked.
  - The length could change between the time of the first "L" and the "MVCSK". This could result in more storage being copied or freed than should be.

z/OS System Integrity: References

- z/OS MVS Programming: Authorized Assembler Services Guide (SA22-7608-02)
  - Chapter 21: Protecting the System
- MVS Planning: Security (GC28-1439-00)
  - Chapter 5: MVS System Integrity
  - http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/iea5f400/5.0?DT=19940303073219
- z/Architecture Principles of Operation (SA22-7832-01)