z/OS 3.2 IBM Education Assistant

Solution Name: Prevent Security Check Bypass for VSAM OPEN

Solution Element(s): DFSMSdfp Catalog, VSAM RLS

July 2025

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

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

Objectives

- Discuss and present:
  - Pre-3.2 default behavior of automatically bypassing security check for OPENs of VSAM data sets when user is running code that is in supervisor state or protection key 0
  - Behavior change in 3.2 for these VSAM OPENs that formerly bypassed the security check, to now do the security check
  - 2 new FACILITY class resources:
    - 2.5, 3.1 and 3.2 FACILITY class resource STGADMIN.IGG.AUTO.BYPASS.LOG
    - 3.2 FACILITY class resource STGADMIN.IGG.AUTO.BYPASS.ALLOW
  - Upgrade actions

Overview (1)

- Who (Audience)
  - Programmers of utility programs
    - who open VSAM data sets while running in supervisor state or protection key 0
  - Users of those utility programs
  - Security Administrators and Analysts
    - who would like to **have logging** of security check bypass during OPENs of VSAM data sets
    - who would like to **prevent** security check bypass during OPENs of VSAM data sets
- The problem this initiative addresses:
  - Currently, the *z/OS DFSMS Using Data Sets* manual states “VSAM OPEN routines bypass RACF security checking if the program issuing OPEN is in supervisor state or protection key 0”.
  - The above statement includes
    - OPENs done with base VSAM or VSAM RLS or TVS.
    - OPEN for input or for output.

Overview (2)

- What (Solution)
  - Change default behavior in 3.2 such that VSAM OPENs that are done in supervisor state or key zero will not automatically bypass security check
- This is a 2 Part Solution:
  - 2.5 and 3.1
    - APARs OA66738 & OA67032 enabled logging.
    - Users can discover the places in which System Authorization Facility (SAF) & RACF check is being bypassed.
    - No default behavior change in 2.5 or 3.1.
  - 3.2
    - In 3.2 the default will change such that VSAM OPENs that are done in supervisor state or key zero will not automatically bypass security check (ACBBYPSS or similar can still be used.)
    - 3.2 will provide more security on VSAM OPENs!
- Wow (Benefit / Value, Need Addressed)
  - This change will provide improved security for clients who run applications which run in supervisor state or key 0, and wish to enforce authorization checking for VSAM data sets being opened.

Usage & Invocation

- In 3.2 the default behavior will be to no longer bypass the security check for VSAM OPENs. No user action is needed to enable the security checks.
- Situations in which the security check will or may still be bypassed:
  - If ACBBYPSS is set on the ACB of the data set, the security check will be bypassed.
  - JSCBPASS allows security check bypass, however setting JSCBPASS to circumvent the planned 3.2 VSAM OPEN behavior change is not recommended. JSCBPASS should never be set in an unauthorized address space.
  - VSAM OPENs done in started tasks and procedures that have attributes TRUSTED or PRIVILEGED set may bypass the security check. See https://www.ibm.com/docs/en/zos/3.1.0?topic=tailoring-assigning-racf-trusted-attribute

Usage & Invocation – ALLOW

- FACILITY class resource STGADMIN.IGG.AUTO.BYPASS.**ALLOW** (3.2 only)
  - If a user has at least READ access authority to STGADMIN.IGG.AUTO.BYPASS.ALLOW in 3.2, VSAM OPEN will behave just as it does now in 3.1, in terms of SAF bypass.
  - In other words, if a user has at least READ access authority to this FACILITY class resource, then VSAM OPENs in which the user is running in key 0 or supervisor state will continue to automatically bypass the security check, as they do now in 3.1.

Usage & Invocation – LOG

- FACILITY class resource STGADMIN.IGG.AUTO.BYPASS.**LOG** (2.5, 3.1, 3.2) allows logging of the automatic security check bypasses during VSAM OPEN
  - On 2.5 and 3.1 with PTFs for OA66738 & OA67032:
    - if a user automatically bypasses SAF check during a VSAM OPEN because the user is running in supervisor state or key 0
    - AND ACBBYPSS or JSCBPASS is not set on
    - AND if that user also has at least READ access to STGADMIN.IGG.AUTO.BYPASS.LOG,
    - AND logging options are specified on the profile covering this resource,
    - An SMF 80 record will be written during OPENs of VSAM data sets. The records will contain information about the SAF bypass
  - On 3.2:
    - if a user has at least READ access to STGADMIN.IGG.AUTO.BYPASS.ALLOW (which allows the bypass if the user is running in supervisor state or key 0)
    - AND the user is running in supervisor state or key 0
    - AND ACBBYPSS or JSCBPASS is not set on
    - AND if that user also has at least READ access to STGADMIN.IGG.AUTO.BYPASS.LOG,
    - AND logging options are specified on the profile covering this resource,
    - An SMF 80 record will be written during OPENs of VSAM data sets. The records will contain information about the SAF bypass.

  Lines in red show difference how this resource works in 2.5/ 3.1 vs in 3.2.

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level: No
- There are no toleration/coexistence APARs/PTFs.
  - 2.5 and 3.1 APARs OA66738 and OA67032 provide support for z/OS 3.2 but are not toleration or coexistence APARs.
- List anything that doesn't work the same anymore:
  - Default 3.2 is that VSAM OPENs done in supervisor state or key 0 that used to automatically bypass security check will no longer.

Upgrade Actions:

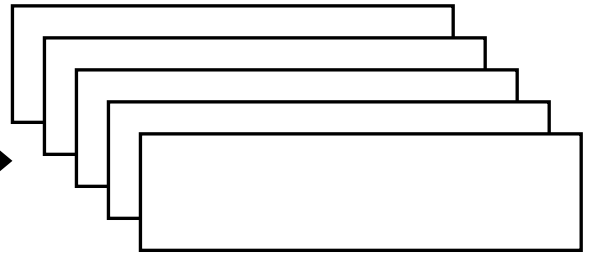
- In 2.5 and/or 3.1
  - Discovery of automatic bypasses of SAF check:
    - Determine which jobs or programs are automatically bypassing security by defining a profile for STGADMIN.IGG.AUTO.BYPASS.LOG and give users at least READ access authority. Run jobs and use the resulting SMF 80 records to determine what jobs are opening data sets and bypassing the authorization checks.
  - For each SMF80 custom record (i.e. each automatic bypass discovered):
    - Analyze the program called in the job to see if it should be bypassing authorization on the data set.
      - If it **should be**, then the ACBBYPSS bit can be set to ensure this behavior continues for future releases.
      - If the program **should not be** bypassing authorization (if you want the SAF check) in 2.5 and 3.1 consider modifying the program to be in problem state and user key at the time the OPEN is issued to ensure the proper authorization checks are done. Prepare for SAF check to occur by default in 3.2

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (1)

1. Setup

1. Apply PTFs for OA66738 & OA67032
2. Define a profile for STGADMIN.IGG.AUTO.BYPASS.LOG
3. Give READ authority to users.
4. Run job that sets up SMF80 collection

2. Run Jobs



3. Collect and Analyze SMF 80

Collect and analyze SMF 80 records

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (2)

Setup Step

- Run a job that does the setup for collecting SMF type 80 records
- Run a job that creates a RACF profile for STGADMIN.IGG.AUTO.BYPASS.LOG and gives user READ access to profile.

```
//STEPLOG   EXEC PGM=IKJEFT01,COND=EVEN
//SYSUDUMP DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *

   RDELETE FACILITY STGADMIN.IGG.AUTO.BYPASS.LOG
   RDEFINE FACILITY STGADMIN.IGG.AUTO.BYPASS.LOG      +
         UACC(READ) AUDIT(SUCCESS)   OWNER(IBMUSER)
```

Or

```
   RDEFINE FACILITY STGADMIN.IGG.AUTO.BYPASS.LOG      +
         UACC(NONE) AUDIT(SUCCESS)   OWNER(IBMUSER)
    PERMIT  STGADMIN.IGG.AUTO.BYPASS.LOG               +
         CLASS(FACILITY) ID(*)  ACCESS(READ)
```

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (3)

Run Jobs

- Now any number of jobs can be run.
- An example that recreates a SAF bypass is
  - JobA which is running under User1, defines a data set and denies access to User2.

```
//STEP02  EXEC  PGM=IKJEFT01,DYNAMNBR=20
 //SYSUDUMP DD    SYSOUT=A
 //SYSTSPRT DD    SYSOUT=A
 //SYSTSIN  DD    *
      SETROPTS GENERIC(DATASET) REFRESH
      DELDSD   'USER1.*'
      ADDSD 'USER1.*' UACC(NONE) AUDIT(ALL)
      PERMIT 'USER1.*' ID(USER2) ACCESS(NONE)
      SETROPTS GENERIC(DATASET) REFRESH
```

  - JobB which is running under User2, tries to open the data set while in supervisor state or key 0. ACBBYPSS and JSCBPASS are not on.
  - Even though User2 has been denied access to this data set, OPEN is successful
  - This SAF bypass will be logged.

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (4)

Collect and Analyze SMF 80

- After running the jobs, collect the SMF 80 records
- Search SMF output for the message text “SAF check” to find the records written by STGADMIN.IGG.AUTO.BYPASS.LOG. This is what the custom message in the SMF 80 looks like :

```
SAF CHECK BYPASSED FOR VSAM OPEN. PROGRAM NAME=
PROGRAMA, JOB STEP=STEP03   , DSN=USER1.KSDS01
```

- The SMF 80 record will also include the Jobname
- Formatters such as ICETOOL or zSecure can be used to format the output.

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (5)

Collect and Analyze SMF 80

111/13/24 20:10:58 RACF report of Bypass events - 1 -

Event TYPE	Event QUAL	Userid	Date	Time	Resource name
ACCESS	SUCCESS	USER2	2024-11-13	19:54:01	STGADMIN.IGG.AUTO.BYPASS.LOG

Jobname	Custom Message
PROAJOB	SAF CHECK BYPASSED FOR VSAM OPEN. PROGRAM NAME=PROGRAMA, JOB STEP=STEP03 ,

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (6)

Collect and Analyze SMF 80

- Use the resulting SMF 80 records to determine what jobs/programs are opening data sets and automatically bypassing the authorization checks.
- For each SMF 80 record generated:
  - Analyze the job or program called in the job to see if it should be bypassing authorization on the data set.
    - If bypass is desired
      - ACBBYPSS bit can be set to ensure this bypass continues in 3.2
    - If no bypass desired
      - then owners of applications should consider running in non-key 0 and problem state when doing VSAM OPENs in 2.5 and 3.1.
      - Prepare for SAF check to occur by default in 3.2

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (7)

Collect and Analyze SMF 80 - FAQ

Will I get SMF 80 record with custom message if I have ACBBYPSS coded on the ACB of the data set or if I have JSCBPASS on job?

No.
SMF 80 records will not be generated if the user asked for the bypass, or if the user is running an application in which a bypass was asked for.

Upgrade Action (Discovery of SAF bypass in 2.5 or 3.1) (8)

Collect and Analyze SMF 80 - FAQ

Does each the SMF 80 records with the custom message such as

```
SAF CHECK BYPASSED FOR VSAM OPEN. PROGRAM NAME=
PROGRAMA, JOB STEP=STEP03  , DSN=USER1.KSDS01
```

represent a VSAM OPEN that will fail in 3.2?

No.
A SMF 80 record with the custom message only means that a SAF bypass happened on the OPEN. If the data set is not covered by a data set profile, or the user does have proper authority to open the data set, that OPEN will not be impacted in 3.2.

Installation & Configuration

- Since the new behavior in 3.2 will be default, there will be no APARs or PTFs needed for enablement of the new behavior, nor any special configurations or other procedures.

Summary

- Current pre-3.2 behavior is that users who run in supervisor state or key 0 are automatically bypassing SAF check during OPENs of VSAM data sets.
- Prevent Security Check Bypass for VSAM OPEN will provide improved security for clients who run applications which run in supervisor state or key 0 and wish to enforce authorization checking for VSAM data sets being opened.

Appendix (1)

- Publications
  - z/OS DFSMS Managing Catalogs
  - z/OS DFSMSdfp Storage Administration
  - z/OS DFSMS Using Data Sets
  - ICN 2060

- Upgrade Workflows
  - 3.1: “DFSMSdfp: Prepare for the change to SAF checking during VSAM OPEN of data sets”
  - 3.2: “DFSMSdfp: Accommodate the change to SAF checking during VSAM OPEN of data sets”

Appendix – Tables (1)

VSAM OPEN in 3.2

Is SAF security check bypassed?
(Chart below assumes the VSAM OPEN is called in supervisor state or key 0)

	ACBBYPSS or JSCBPASS is ON	ACBBYPSS & JSCBPASS are OFF	
User has authority to "ALLOW" FACILITY class profile	Yes	Yes	<--This is the old behavior
No authority to "ALLOW" FACILITY class profile (or no profile exists)	Yes	No**	<-new default behavior for 3.2

In 3.2 if you want the same behavior as today, use the "ALLOW" FACILITY class resource.*

* "ALLOW" FACILITY class resource = STGADMIN.IGG.AUTO.BYPASS.ALLOW
** Assuming no other considerations that allow bypass such as mentioned on page 11.

Appendix – Tables (2)

VSAM OPEN in 3.2

In other words, will the OPEN do a SAF (security) check?
(Chart below assumes the VSAM OPEN is called in supervisor state or key 0)

	ACBBYPSS or JSCBPASS is ON	ACBBYPSS & JSCBPASS are OFF	
User has authority to "ALLOW" FACILITY class profile	No	No	<--This is the old behavior
No authority to "ALLOW" FACILITY class profile (or no profile exists)	No	Yes	<-new default behavior for 3.2

In 3.2 if you want the same behavior as today, use the "ALLOW" FACILITY class resource.*

* "ALLOW" FACILITY class resource = STGADMIN.IGG.AUTO.BYPASS.ALLOW

By default, in 3.2, this OPEN will perform SAF check, unless bypass bit is on. (Assuming no other considerations that allow bypass such as mentioned on page 11.)

Appendix – Tables (3)

VSAM OPEN in 3.2

Should the OPEN of the data set succeed or fail?
(Chart below assumes the VSAM OPEN is called in supervisor state or key 0)

	Data set has no RACF (or equivalent) profile	Data set is protected under a RACF (or equivalent) profile		
		User doing the OPEN has authority to the data set	User doing the OPEN does not have necessary authority to the data set.	
User has authority to "ALLOW" FACILITY class profile	succeeds*	succeeds	succeeds	<--This is the old behavior
No authority to "ALLOW" FACILITY class profile (or no profile exists)	succeeds*	succeeds	fails**	<-new default behavior for 3.2

Default 3.2 without authority to the "ALLOW”*** FACILITY class profile means no special treatment for VSAM OPENs called from programs running in key 0 or supervisor state.

* Assuming that PROTECTALL is not enabled.
** Assuming no other considerations that allow bypass such as mentioned on page 11.
*** "ALLOW" FACILITY class resource = STGADMIN.IGG.AUTO.BYPASS.ALLOW

Appendix – Tables (4)

Logging of bypass in 2.5 and 3.1

Will it write SMF 80 record with custom message showing the bypass?

(Chart below assumes the VSAM OPEN is called in supervisor state or key 0)

	ACBBYPSS or JSCBPASS is on	ACBBYPSS & JSCBPASS are OFF
User has authority to "LOG" FACILITY class profile	NO	YES
No authority to "LOG" FACILITY class profile (or no profile exists)	NO	NO

<-no auth to "LOG" means no SMF 80 record with custom message

Logging of bypass in 3.2

Will it write SMF 80 record with custom message showing the bypass?

(Chart below assumes the VSAM OPEN is called in supervisor state or key 0)

	ACBBYPSS or JSCBPASS is on	ACBBYPSS & JSCBPASS are OFF	
		User has authority to "ALLOW" FACILITY class profile	User has NO authority to "ALLOW" FACILITY class profile (or no profile exists)
User has authority to "LOG" FACILITY class profile	NO	YES	NO
No authority to "LOG" FACILITY class profile (or no profile exists)	NO	NO	NO

<-no auth to "LOG" means no SMF 80 record with custom message