

INVENTORY OF METHODS TO STRENGTHEN THE SECURITY AND AUDITABILITY OF IBM i DB2 DATA



Steve Bradshaw
steve.bradshaw@rowtonit.com



Guy Marmorat
gmarmorat@resiliane.com

- 
- Db2 Data Access Methods
 - Technologies Available to Improve Auditability and Protection

Associated protocols:

ODBC
JDBC
.Net
OLEDB
DRDA
SSH Db2
CLI/QShell Db2
Remote SQL, XDA
VS Code – Db2 for i
RSE
Maepire
...

SQL Remote

SELECT DROP
UPDATE CREATE
INSERT ALTER
DELETE GRANT
MERGE TRUNCATE
...

SSH
SCP/SFTP
Put
Get ...

SSH
PASE
cp
mn
rm
chmod ...

User Commands
(CPP)
DBU ...

User Programs
*PGM
*PGMSRV
SQL, RLA

**Inventory of all
methods
accessing Db2
Files/Tables**

Triggers
ADDPFTRG
CREATE TRIGGER

QUERY/400
RUNQRY
WRKQRY
QQQQQRY

System Commands
UPDDTA **EDTF** } INTER
DSPPFM DSPF }
SAVxxx RSTxxx } BATCH
CPYxxx DMPxxx }
SNDSMTPEMM ...

**Remote
Commands:**
FTP Server - Quote Rcmd
FTP Client - Syscmd
REXEC - RUNRMTCMD
DDM - SBMRMTCMD
IBM i Access for Windows - RMTCMD
ODBC/DRDA - QCMDXEC proc or scalar function
SSH – System ...

IBM i Services
get_clob_from_file
QSYS2.IFS_READ
QSYS2.IFS_WRITE
SYSTOOLS.IFS_RENAME
...

LIBRARY



**Accessing JOURNAL receiver
contents:**
QSYS2.DISPLAY_JOURNAL
DSPJRN, RCVJRNE
QjoRetrieveJournalEntries
DATA_JOURNAL_READER
...

FTP Server
FTP Client
Put
Get
Delete
Rename ...

File Server
NetServer/QSYS.LIB
Open
Rename
Delete ...

ObjectConnect
SAVRSTxxx

Commands & Pgms
SQL Execution
RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

DDM File
Commands (CPYF...)
SQL, RLA

AUDIT

SQL Remote

SELECT DROP
UPDATE CREATE
INSERT ALTER
DELETE GRANT
MERGE TRUNCATE
...

SSH
SCP/SFTP
Put
Get ...

SSH
PASE
cp
mn
rm
chmod ...

User Commands
(CPP)
DBU ...

User Programs
*PGM
*PGMSRV
SQL, RLA

Triggers

ADDPFTRG
CREATE TRIGGER

QUERY/400

RUNQRY
WRKQRY
QQQQQRY

System Commands

UPDDTA	EDTF	} INTER
DSPPFM	DSPF	
SAVxxx	RSTxxx	} BATCH
CPYxxx	DMPxxx	
SNDSMTPEMM	...	

DDM File

Commands (CPYE...)
SQL, RLA

Commands & Pgms

SQL Execution

RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

ObjectConnect

SAVRSTxxx

File Server

NetServer/QSYS.LIB

Open
Rename
Delete ...

FTP Server FTP Client

Put
Get
Delete
Rename ...

LIBRARY

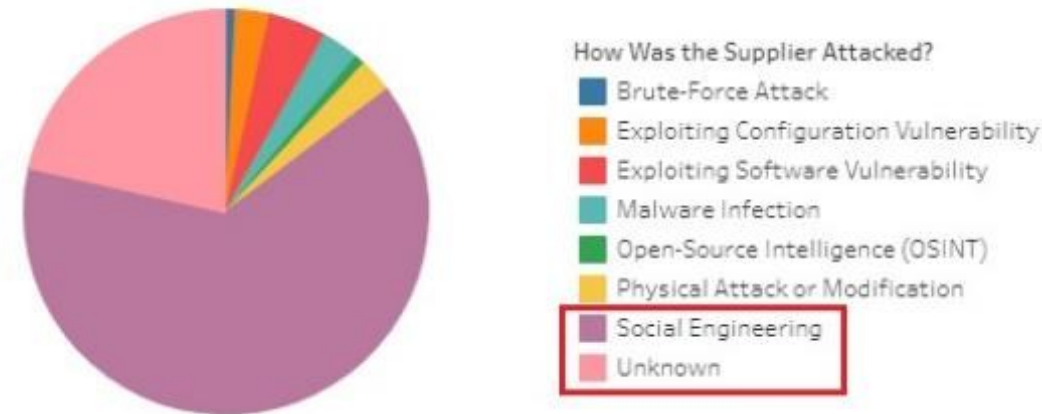
D 2
File
Tabl

Why an Audit Trail is so important?

- Always understand what happened
- Identify security attempts, breaches and human errors
- Prove the positive effects of your remediations
- Identify the side effects of your remediations
- Identify the negative effects of your remediations



Supplier Attacks-Type



<https://www.comparitech.com/software-supply-chain-attacks/>

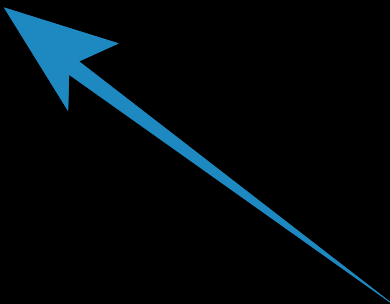
The events to be audited and their associated audit trails

Events to audit	Journal	Journal code	Journal Entry types	Other Audit Trails
File openings (read, modify)	QAUDJRN	T	ZC – ZR (access type = 30 ...)	Authority Collection, Exit Point ‘Open Database File’
	Database	F	OP	
Actions at the Object Level (creation, deletion, rename, backup, restoration, authorities, owner, etc.)	QAUDJRN	T	CO - DO - OM - ZR (access type = 46 47 48) - OR - CA – OW	Command & SQL Exit points, Database Monitor, Plan Cache
	Database	D	CT - DT - FN - DH - DZ - GT – GO	
Actions on certain file attributes (triggers, constraints, RCAC functions, logging, etc.)	Database	D	TC TD TG - AC DC GC - M1 M2 M3 P1 P2 P3 - DJ EF JF	Command & SQL Exit points, Database Monitor, Plan Cache
	QAUDJRN	T	AX	
Actions at the Member Level (creation, deletion, rename, backup, restoration, logging, clear, reorganize, ...)	Database	F	MC - MD - MN - MF MS - MR - JC EJ JM - CR - RG RM	Command & SQL Exit points, Database Monitor, Plan Cache
Actions at the Record Level (add, modify, delete)	Database	R	PT PX - UB UP - DL	Triggers (system & SQL), SQL Exit points, Database Monitor, Plan Cache
Attempts to access files	QAUDJRN	T	AF	Authority Collection
Changes to audit values	QAUDJRN	T	AD	Command Exit point
Actions at record level (read)				Read-Triggers (system) 

How to find the description of Audit Journal entry types?

```
Object . . . . . :
Member . . . . . :
Incomplete data . . : No      Minimized entry data : *NONE
Sequence . . . . . : 87515767
Code . . . . . : T - Audit trail entry
Type . . . . . : C0 - Create object
```

```
Column      Entry specific data
*...+....1...+....2...+....3...+....4...+....5
00001      'NGLFPGM011 XXERPPGM *PGM   RPGLE
00051      :
00101      :
00151      :
00201      :
00251      :
00301      :
```



<https://www.ibm.com/docs/en/i/7.6?topic=security-reference>
Security Reference Guide, Appendix "Layout of audit journal entries" (only for Journal Code = T)

Model files in QSYS (example : QASYCPJ5 for entry type CP)
(only for Journal Code = T)

Fields you may encounter in many entries for Journal Code T
XXETYP TYPE OF ENTRY A 1
XXONAM Object Name A 10
XXOLIB Library name A 10
XXOTYP Object type A 8
XXPNM Path name A 5000

https://www.ibm.com/docs/en/i/7.5.0?topic=ssw_ibm_i_75/rzaru/rzarufinder.html
Journal entry information finder

How to read Audit Journal entries?

Structure of the entry image is interpreted

```
Object . . . . . : Library . . . . . :
Member . . . . . :
Incomplete data . . . : No Minimized entry data : *NONE
Sequence . . . . . : 87515767
Code . . . . . : T - Audit trail entry
Type . . . . . : C0 - Create object
```

```
Column      Entry specific data
00001      *...+....1....+....2....+....3....+....4....+....5
00051      'NGLFPGM011 XXERPPGM *PGM   RPGLE
00101      :
00151      :
00201      :
00251      :
00301      :
```

OBJLIB	OBJNAME	OBJTYPE	OBJATTRIBUTE	OBJTEXT
QSYS	CPYAUDJRNE	*CMD		Copy Audit Journal Entries
QSYS	DSPAUDJRNE	*CMD		Display Audit Journal Entries
QSYS	DSPJRN	*CMD		Display Journal
QSYS	RCVJRNE	*CMD		Receive Journal Entry
QSYS	RTVJRNE	*CMD		Retrieve Journal Entry

Program Library	Program Name	Object Type	Symbol Name	Symbol Usage
QSYS	QJOURNAL	*SRVPGM	QjoRetrieveJournalEntries	*PROCEXP

ROUTINE_SCHEMA	ROUTINE_NAME	ROUTINE_TYPE	ROUTINE_BODY	EXTERNAL_NAME	EXTERNAL_LANGUAGE
QSYS2	DISPLAY_JOURNAL	FUNCTION	EXTERNAL	QSYS/QDBSSUDF2 (QJOJRNTF)	C
QSYS2	DISPLAY_JOURNAL_ENTRY_INFO	PROCEDURE	EXTERNAL	QSYS/QDBSSUDF2 (DSPJEI)	C

QSYS2	AUDIT_JOURNAL_DATA_MART_INFO	FUNCTION	EXTERNAL	QSYS/QDBSSUDF2 (QSAJDMART)	C
QSYS2	MANAGE AUDIT JOURNAL DATA MART	PROCEDURE	SQL	QSYS2/MANAGE AJ	-

ROUTINE_SCHEMA	ROUTINE_NAME	ROUTINE_TYPE	ROUTINE_BODY	EXTERNAL_NAME	EXTERNAL_LANGUAGE
SYSTOOLS	AUDIT_JOURNAL_AD	FUNCTION	SQL	SYSTOOLS/AUDIT_AD (AUDIT_JOURNAL_AD_1)	-
SYSTOOLS	AUDIT_JOURNAL_AF	FUNCTION	SQL	SYSTOOLS/AUDIT_AF (AUDIT_JOURNAL_AF_1)	-
SYSTOOLS	AUDIT_JOURNAL_AP	FUNCTION	SQL	SYSTOOLS/AUDIT_AP (AUDIT_JOURNAL_AP_1)	-
SYSTOOLS	AUDIT_JOURNAL_AU	FUNCTION	SQL	SYSTOOLS/AUDIT_AU (AUDIT_JOURNAL_AU_1)	-
SYSTOOLS	AUDIT_JOURNAL_AX	FUNCTION	SQL	SYSTOOLS/AUDIT_AX (AUDIT_JOURNAL_AX_1)	-
SYSTOOLS	AUDIT_JOURNAL_CA	FUNCTION	SQL	SYSTOOLS/AUDIT_CA (AUDIT_JOURNAL_CA_1)	-
SYSTOOLS	AUDIT_JOURNAL_CD	FUNCTION	SQL	SYSTOOLS/AUDIT_CD (AUDIT_JOURNAL_CD_1)	-
SYSTOOLS	AUDIT_JOURNAL_CO	FUNCTION	SQL	SYSTOOLS/AUDIT_CO (AUDIT_JOURNAL_CO_1)	-

Audit Journal entries and their SYSTOOLS functions

new entries brought by version 7.5				
C3 - Advanced Analysis Command Configuration (TLSCONFIG				
FT - FTP Client Operations				
M0 - Db2 Mirror setup tools				
M6 - Db2 Mirror communication services				
M7 - Db2 Mirror replication services				
M8 - Db2 Mirror product services				
M9 - Db2 Mirror replication state				

QAUDJRN journal entry types, including those with SYSTOOLS function				
7.3 TR12	79	14		
7.4 TR9	86	40		
7.4 TR10	86	40		
7.5 TR5	87	41		
7.6 TR1	87	45		
7.6 TR2	87	47	(JD & RU)	

7.6 TR2			7.5 TR5			7.4 TR10			7.4 TR9			7.3 TR12		
JOI	ROU	CREATED	JOI	ROU	CREATED	JOI	ROU	CREATED	JOI	ROU	CREATED	JOI	ROU	CREATED
AD	AUDIT	2024-10-22	AD	AUDIT	11/17/2023	AD	AUDIT	2/8/2024	AD	AUDIT	3/31/2024	AD		
AF	AUDIT	2024-10-22	AF	AUDIT	11/17/2023	AF	AUDIT	2/8/2024	AF	AUDIT	3/31/2024	AF	AUDIT	9/27/2022
AP	AUDIT	2024-10-22	AP	AUDIT	11/17/2023	AP	AUDIT	2/8/2024	AP	AUDIT	3/31/2024	AP		
AU	AUDIT	2024-10-22	AU	AUDIT	5/3/2024	AU	AUDIT	7/4/2024	AU	AUDIT	9/8/2024	AU		
AX	AUDIT	2024-10-22	AX	AUDIT	5/3/2024	AX	AUDIT	7/4/2024	AX	AUDIT	9/8/2024	AX		
CA	AUDIT	2024-10-22	CA	AUDIT	11/17/2023	CA	AUDIT	2/8/2024	CA	AUDIT	3/31/2024	CA	AUDIT	9/27/2022
CD	AUDIT	2024-10-22	CD	AUDIT	11/17/2023	CD	AUDIT	2/8/2024	CD	AUDIT	3/31/2024	CD	AUDIT	9/27/2022
CO	AUDIT	2024-10-22	CO	AUDIT	11/17/2023	CO	AUDIT	2/8/2024	CO	AUDIT	3/31/2024	CO	AUDIT	9/27/2022
CP	AUDIT	2024-10-22	CP	AUDIT	5/3/2024	CP	AUDIT	7/4/2024	CP	AUDIT	9/8/2024	CP	AUDIT	9/27/2022
DO	AUDIT	2024-10-22	DO	AUDIT	11/17/2023	DO	AUDIT	2/8/2024	DO	AUDIT	3/31/2024	DO	AUDIT	9/27/2022
DS	AUDIT	2024-10-22	DS	AUDIT	11/17/2023	DS	AUDIT	2/8/2024	DS	AUDIT	3/31/2024	DS		
EV	AUDIT	2024-10-22	EV	AUDIT	5/3/2024	EV	AUDIT	7/4/2024	EV	AUDIT	9/8/2024	EV	AUDIT	9/27/2022
GR	AUDIT	2024-10-22	GR	AUDIT	5/3/2024	GR	AUDIT	7/4/2024	GR	AUDIT	9/8/2024	GR	AUDIT	9/27/2022
IM	AUDIT	2024-10-22	IM	AUDIT	11/17/2023	IM	AUDIT	2/8/2024	IM	AUDIT	3/31/2024	IM		
JD	AUDIT	2026-07-14	JD			JD			JD			JD		
JS	AUDIT	2024-10-22	JS	AUDIT	5/3/2024	JS	AUDIT	7/4/2024	JS	AUDIT	9/8/2024	JS	AUDIT	9/27/2022
LD	AUDIT	2024-10-22	LD	AUDIT	11/17/2023	LD	AUDIT	2/8/2024	LD	AUDIT	3/31/2024	LD		
M0	AUDIT	2024-10-22	M0	AUDIT	11/17/2023	M0	AUDIT	2/8/2024	M0	AUDIT	3/31/2024			
M6	AUDIT	2024-10-22	M6	AUDIT	11/17/2023	M6	AUDIT	2/8/2024	M6	AUDIT	3/31/2024			
M7	AUDIT	2024-10-22	M7	AUDIT	11/17/2023	M7	AUDIT	2/8/2024	M7	AUDIT	3/31/2024			
M8	AUDIT	2024-10-22	M8	AUDIT	11/17/2023	M8	AUDIT	2/8/2024	M8	AUDIT	3/31/2024			
M9	AUDIT	2024-10-22	M9	AUDIT	11/17/2023	M9	AUDIT	2/8/2024	M9	AUDIT	3/31/2024			
NA	AUDIT	2024-10-22	NA	AUDIT	2024-10-22	NA	AUDIT	7/4/2024	NA	AUDIT	9/8/2024	NA		
OM	AUDIT	2024-10-22	OM	AUDIT	5/3/2024	OM	AUDIT	7/4/2024	OM	AUDIT	9/8/2024	OM	AUDIT	9/27/2022
OR	AUDIT	2024-10-22	OR	AUDIT	11/17/2023	OR	AUDIT	2/8/2024	OR	AUDIT	3/31/2024	OR		
OW	AUDIT	2024-10-22	OW	AUDIT	11/17/2023	OW	AUDIT	2/8/2024	OW	AUDIT	3/31/2024	OW	AUDIT	9/27/2022
PA	AUDIT	2024-10-22	PA	AUDIT	11/17/2023	PA	AUDIT	2/8/2024	PA	AUDIT	3/31/2024	PA		
PF	AUDIT	2024-10-22	PF	AUDIT	11/17/2023	PF	AUDIT	2/8/2024	PF	AUDIT	3/31/2024	PF		
PG	AUDIT	2024-10-22	PG	AUDIT	11/17/2023	PG	AUDIT	2/8/2024	PG	AUDIT	3/31/2024	PG		
PO	AUDIT	2025-05-15	PO			PO			PO			PO		
PS	AUDIT	2024-10-22	PS	AUDIT	2024-10-22	PS	AUDIT	7/4/2024	PS	AUDIT	9/8/2024	PS		
PU	AUDIT	2024-10-22	PU	AUDIT	11/17/2023	PU	AUDIT	2/8/2024	PU	AUDIT	3/31/2024	PU		
PW	AUDIT	2024-10-22	PW	AUDIT	11/17/2023	PW	AUDIT	2/8/2024	PW	AUDIT	3/31/2024	PW	AUDIT	9/27/2022
RA	AUDIT	2024-10-22	RA	AUDIT	11/17/2023	RA	AUDIT	2/8/2024	RA	AUDIT	3/31/2024	RA		
RO	AUDIT	2024-10-22	RO	AUDIT	11/17/2023	RO	AUDIT	2/8/2024	RO	AUDIT	3/31/2024	RO		
RP	AUDIT	2025-11-21	RP			RP			RP			RP		
RU	AUDIT	2026-07-14	RU			RU			RU			RU		
RZ	AUDIT	2024-10-22	RZ	AUDIT	11/17/2023	RZ	AUDIT	2/8/2024	RZ	AUDIT	3/31/2024	RZ		
SF	AUDIT	2025-05-15	SF			SF			SF			SF		
SK	AUDIT	2024-10-22	SK	AUDIT	11/17/2023	SK	AUDIT	2/8/2024	SK	AUDIT	3/31/2024	SK		
SM	AUDIT	2024-10-22	SM	AUDIT	11/17/2023	SM	AUDIT	2/8/2024	SM	AUDIT	3/31/2024	SM		
ST	AUDIT	2024-10-22	ST	AUDIT	11/17/2023	ST	AUDIT	2/8/2024	ST	AUDIT	3/31/2024	ST	AUDIT	9/27/2022
SV	AUDIT	2024-10-22	SV	AUDIT	11/17/2023	SV	AUDIT	2/8/2024	SV	AUDIT	3/31/2024	SV	AUDIT	9/27/2022
VO	AUDIT	2025-11-21	VO			VO			VO			VO		
VP	AUDIT	2025-01-22	VP	AUDIT	1/22/2025	VP			VP			VP		
ZC	AUDIT	2024-10-22	ZC	AUDIT	5/3/2024	ZC	AUDIT	7/4/2024	ZC	AUDIT	9/8/2024	ZC		
ZR	AUDIT	2024-10-22	ZR	AUDIT	11/17/2023	ZR	AUDIT	2/8/2024	ZR	AUDIT	3/31/2024	ZR		

Audit Journal Entries and their SYSTOOLS Functions

Source SYSTOOLS.AUDIT_JOURNAL_AX

```

244 select * from qsys2.QSQAJMRI
245 where msg_key like 'AX_&'
246 for update ;

```

MSG_KEY	MSG_TEXT
AX_ET_M	Column mask
AX_ET_P	Row permission
AX_ET_T	Table

```

BEGIN
DECLARE RESULT_SQL_STATEMENT_TEXT VARGRAPHIC ( 5000 ) CCSID 1200 ;
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
FOR AUDIT_CURSOR CURSOR FOR
SELECT
J . ENTRY_TIMESTAMP ,
J . SEQUENCE_NUMBER ,
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
B . * ,
CHAR ( SUBSTR ( J . ENTRY_DATA , 1 , 1 ) , 1 ) AS A_ENTRY_TYPE ,
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
FROM TABLE ( QSYS2 . DISPLAY_JOURNAL (
'QSYS' ,
'QAUDJRN' ,
JOURNAL_CODES => 'T' ,
JOURNAL_ENTRY_TYPES => 'AX' ,
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
) ) AS J
LEFT OUTER JOIN QSQAJMRI AS B
ON CONCAT ( 'AX_ET_' , CHAR ( SUBSTR ( J . ENTRY_DATA , 1 , 1 ) , 1 ) ) = B . MSG_KEY
DO
IF A_SQL_STATEMENT_LENGTH > 0 THEN
BEGIN
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CASE WHEN A_ENTRY_TYPE = 'T' AND A_OPERATION_TYPE = 'A'
THEN CASE A_ENABLED_OR_ROW_ACCESS_CONTROL
WHEN 'A' THEN 'ACTIVATE'
WHEN 'D' THEN 'DEACTIVATE'
WHEN '' THEN NULL
ELSE A_ENABLED_OR_ROW_ACCESS_CONTROL
END
ELSE NULL
END ,
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
) ;
END FOR ;
RETURN ;
END

```

AUDIT_JOURNAL_DATA_MART

7.4 TR10/11 7.5 TR4/5

Uses
SYSTOOLS.AUDIT_JOURNAL_PW
under the cover

```
-- initialize with option CREATE or REPLACE, add the FILTER at the beginning (not possible after)
CALL QSYS2.MANAGE_AUDIT_JOURNAL_DATA_MART(JOURNAL_ENTRY_TYPE => 'PW',
                                           DATA_MART_LIBRARY => 'RSLAUD',
                                           STARTING_TIMESTAMP => current date - 6 months,
                                           ENDING_TIMESTAMP => current date - 2 days,
                                           DATA_MART_ACTION => 'REPLACE',
                                           DATA_MART_FILTER => 'VIOLATION_TYPE in (''P'', ''Q'', ''U'')'
                                           );

-- keep the FILTER for ADD
CALL QSYS2.MANAGE_AUDIT_JOURNAL_DATA_MART(JOURNAL_ENTRY_TYPE => 'PW',
                                           DATA_MART_LIBRARY => 'RSLAUD',
                                           STARTING_TIMESTAMP => '*CONTINUE',
                                           ENDING_TIMESTAMP => current date - 1 days,
                                           DATA_MART_ACTION => 'ADD'
                                           );
```

```
-- track the build process
select data_mart_library, data_mart_table, data_mart_filter, audit_journal_starting_timestamp, audit_journal_ending_timestamp,
build_end, second(build_end) - second(build_start) build_duration_seconds
FROM QSYS2.AUDIT_JOURNAL_DATA_MART_INFO
WHERE JOURNAL_ENTRY_TYPE = 'PW' AND DATA_MART_LIBRARY = 'RSLAUD' ;

-- query the result
select entry_timestamp, user_name, job_user, program_library, program_name, remote_address,
VIOLATION_TYPE_DETAIL, AUDIT_USER_NAME from rslaud.audit_journal_pw ;

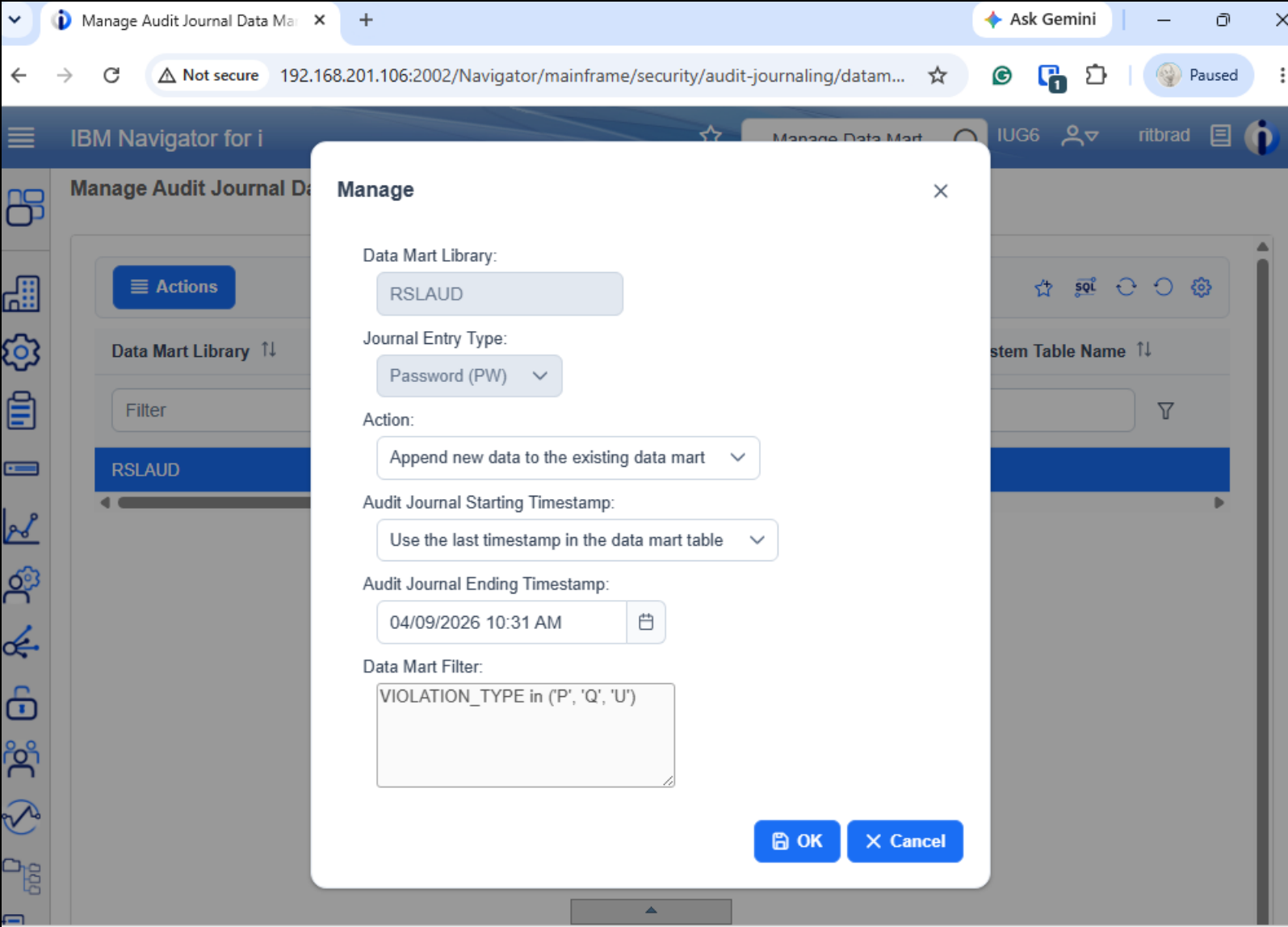
-- keep retention up to date
delete rslaud.audit_journal_pw where entry_timestamp < current timestamp - 1 month ;
```

DATA_MART_LIBRARY	DATA_MART_TABLE	DATA_MART_FILTER	AUDIT_JOURNAL_STARTING_TIMESTAMP	AUDIT_JOURNAL_ENDING_TIMESTAMP	BUILD_END	BUILD_DURATION_SECONDS
Data Mart Library	Data Mart Table	Data Mart Filter	Audit Journal Starting Timestamp	Audit Journal Ending Timestamp	Build End	BUILD_DURATION_SECONDS
RSLAUD	AUDIT_JOURNAL_PW	VIOLATION_TYPE in ('P', 'Q', 'U')	2026-06-24 16:36:07.6...	2026-06-24 16:45:07...	2026-06-24 16:45:07...	0

ENTRY_TIMESTAMP	USER_NAME	JOB_NAME	PROGRAM_LIBRARY	PROGRAM_NAME	REMOTE_ADDRESS	VIOLATION_TYPE_DETAIL	AUDIT_USER_NAME
2026-06-24 16:27...	QUSER_NC	QZSOSIGN	QSYS	QZSOSIGN	10.242.2.131	Password not valid	GUY
2026-06-24 16:28...	QSYS	QINTER	QSYS	QLESPI	10.242.2.131	User name not valid	GMXXX
2026-06-24 16:32...	QSECOFR_NC	QPOZSPWT	QSYS	QPOZPCP2	10.242.2.131	Attempted signon (user authentication) failed because user profile is disabled	GUY

AUDIT_JOURNAL_DATA_MART

7.4 TR10/11 7.5 TR4/5



File Opening Auditing - Management

```
-- maintain repository of sensitive tables to be audited
-- keep the object audit value ni case of rollback
create or replace table rslaud.tables_to_audit as
(select system_table_schema, system_table_name, object_audit initial_object_audit from qsys2.sysfiles
join table(QSYS2.OBJECT_STATISTICS(system_table_schema, '*FILE')) st
on st.objlib = system_table_schema and st.objname = system_table_name and st.objtype = '*FILE'
where system_table_schema = 'ERPFILE' and native_type = 'PHYSICAL' and system_table_name like 'GLF%'
and right(system_table_name, 1) <> '2' )
with data on replace delete rows ;
insert into rslaud.tables_to_audit values('XXERPDATA', 'GLFCLIEN', ' ' ) ;
merge into rslaud.tables_to_audit
using table(QSYS2.OBJECT_STATISTICS(system_table_schema, '*FILE')) st
on st.objlib = system_table_schema and st.objname = system_table_name and st.objtype = '*FILE' and initial_object_audit = ' '
WHEN MATCHED THEN UPDATE SET initial_object_audit = object_audit ;

select * from rslaud.tables_to_audit ;
```

System Table Schema	System Table Name	INITIAL_OBJECT_AUDIT
ERPFILE	GLFCLIEN	*ALL
ERPFILE	GLFCUENTA	*ALL
ERPFILE	GLFTRANS	*NONE
XXERPDATA	GLFCLIEN	*CHANGE

```
-- start auditing *ALL
select 'CHGOBJAUD OBJ(' concat system_table_schema concat '/' concat system_table_name concat ') OBJTYPE(*FILE) OBJAUD(*ALL)' command,
qcmdexc('CHGOBJAUD OBJ(' concat system_table_schema concat '/' concat system_table_name concat ') OBJTYPE(*FILE) OBJAUD(*ALL)') command_result
from rslaud.tables_to_audit
join table(QSYS2.OBJECT_STATISTICS(system_table_schema, '*FILE')) st
on st.objlib = system_table_schema and st.objname = system_table_name and st.objtype = '*FILE'
where initial_OBJECT_AUDIT <> '*ALL' ;
```

COMMAND	COMMAND_RESULT
CHGOBJAUD OBJ(ERPFILE/GLFTRANS) OBJTYPE(*FILE) OBJAUD(*ALL)	1
CHGOBJAUD OBJ(XXERPDATA/GLFCLIEN) OBJTYPE(*FILE) OBJAUD(*ALL)	1

File Opening Auditing - Analysis

ZC- use object with change intention
ZR- use object for read intention

```
-- using QAUDJRN, check for Db2 file usage (starting with 7.5 TR1 7.4 TR7)
create or replace table rslaud.audit_zc as
(SELECT entry_timestamp, jrn.user_name, job_name, program_library, program_name, remote_address,
entry_type_detail, access_type_detail, library_name, object_name
FROM TABLE (SYSTOOLS.AUDIT_JOURNAL_ZC (STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current timestamp - 15 minutes)) jrn where object_type = '*FILE')
with data on replace delete rows;
insert into rslaud.audit_zc
(SELECT entry_timestamp, jrn.user_name, job_name, program_library, program_name, remote_address,
entry_type_detail, access_type_detail, library_name, object_name
FROM TABLE (SYSTOOLS.AUDIT_JOURNAL_ZR (STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current timestamp - 15 minutes)) jrn where object_type = '*FILE');
```

```
-- check accesses outside the application
select * from rslaud.audit_zc
where program_library not in ('ERPPGM', 'ERPTF') and not (job_name = 'QZDASSINIT' and remote_address = '1.2.3.4' and user_name = 'ERPSRVUSER')
order by entry_timestamp;
```

ENTRY_TIMESTAMP	USER_NAME	JOB_NAME	PROGRAM_LIBRARY	PROGRAM_NAME	REMOTE_ADDRESS	ENTRY_TYPE_DETAIL	ACCESS_TYPE_DETAIL	LIBRARY_NAME	OBJECT_NAME
2026-06-26 07:43:07.591568	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Read of an object	Dump	ERPPFILE	GLFCLIE
2026-06-26 07:43:08.672656	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:43:08.807152	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Change of an object	Reorganize	ERPPFILE	GLFCLIE
2026-06-26 07:43:32.492848	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Read of an object	Save	ERPPFILE	GLFCLIE
2026-06-26 07:43:34.072448	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Read of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:43:34.089680	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Read of an object	Read	ERPPFILE	GLFCLIE
2026-06-26 07:43:37.879888	GM	QZDASOINIT	QSYS2	QCMD	10.242.2.130	Read of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:43:41.723200	GM	QZDASOINIT	QSYS	QZDASOINIT	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:44:13.419744	GM	QPADEV0003	QTEMP	QZTD00001	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:44:13.419872	GM	QPADEV0003	QTEMP	QZTD00001	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:54:01.737584	GM	QPADEV0003	QSYS	QCMD	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:54:36.384096	GM	QPADEV0003	QSYS	QCMD	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:55:02.800064	GM_BASIC	QPADEV0003	QSYS	QCMD	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:55:16.779056	GM_BASIC	QPADEV0003	QSYS	QCMD	10.242.2.130	Change of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:55:43.261744	GM_BASIC	QPADEV0003	QSYS	QCMD	10.242.2.130	Read of an object	Open	ERPPFILE	GLFCLIE
2026-06-26 07:56:04.220432	GM_BASIC	QPADEV0003	QSYS	QCMD	10.242.2.130	Read of an object	Open	ERPPFILE	GLFCLIE

Db2 Changes Auditing at Record Level

```
-- using database journal, check for all changes
select entry_timestamp, journal_entry_type type, jrn.current_user, job_name, program_name, remote_address,
cast(cast(SUBSTR(entry_data ,1, 200) as char(200) for bit data) as char(200) ccsid 1141) record_image
from table( qsys2.Display_Journal ('ERPFIL', 'ERPJRN', Journal_Codes => 'R', STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn ;

-- using database journal, check for changes made by DFU
select entry_timestamp, journal_entry_type type, jrn.current_user, job_name, program_name, remote_address,
cast(cast(SUBSTR(entry_data ,1, 200) as char(200) for bit data) as char(200) ccsid 1141) record_image
from table( qsys2.Display_Journal ('ERPFIL', 'ERPJRN', Journal_Codes => 'R', STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn
where program_library not in ('ERPPGM', 'ERPPTF') and not (job_name = 'QZDASSINIT' and remote_address = '123.456.789.1' and user_name = 'ERPSRVUSER') ;
```

ENTRY_TIMESTAMP	TYPE	CURRENT_USER	JOB_NAME	PROGRAM_NAME	REMOTE_ADDRESS	RECORD_IMAGE
2026-06-29 08:28:29...	UB	GM	QZDASOINIT	QZDASOINIT	10.242.2.130	00159150000000000001John Ford (Jr.)
2026-06-29 08:28:29...	UP	GM	QZDASOINIT	QZDASOINIT	10.242.2.130	00159150000000000001John Ford (Jr.)
2026-06-29 08:28:29...	UB	GM	QZDASOINIT	QZDASOINIT	10.242.2.130	00249150000000000001Petros CHRISTOFIDES
2026-06-29 08:28:29...	UP	GM	QZDASOINIT	QZDASOINIT	10.242.2.130	00249150000000000001Petros CHRISTOFIDES
2026-06-29 08:28:52...	UB	GM	QPADEV0002	QDZTD00001	10.242.2.130	00149150000000000002Mr Jaime GONZALES
2026-06-29 08:28:52...	UP	GM	QPADEV0002	QDZTD00001	10.242.2.130	00149150000000000002Mr Jaime GONZALES
2026-06-29 08:29:11...	UB	GM	QPADEV0002	GLFPGM001E	10.242.2.130	00149150000000000002Mr Jaime GONZALES
2026-06-29 08:29:11...	UP	GM	QPADEV0002	GLFPGM001E	10.242.2.130	00149150000000000002Mr Jaime GONZALES

Db2 Changes Auditing at Record Level (Function added by 7.6 TR2 & 7.5 TR8)

```
369 -- create data-journal reader table function for all the registered sensitive files
370 select tb.*,
371 QSYS2.CREATE_DATA_JOURNAL_READER(LIBRARY_NAME => system_table_schema, FILE_NAME => system_table_name, OUTPUT_LIBRARY => 'RSLAUD') data_journal_reader_created
372 from rslaud.tables_to_audit tb ;
373
```

SYSTEM_TABLE_SCHEMA	SYSTEM_TABLE_NAME	INITIAL_OBJECT_AUDIT	DATA_JOURNAL_READER_CREATED
System	System		
Table	Table		
Schema	Name		
ERPPFILE	GLFCLIE	*ALL	1
ERPPFILE	GLFCUENTA	*ALL	1
ERPPFILE	GLFTRANS	*NONE	1
XXERPPDATA	GLFCLIE	*CHANGE	-1

```
374 -- using data-journal reader, check for changes made by a program outside of the ERP application to a sensitive file
375 SELECT entry_timestamp, user_name, operation, clicomp, clinumero, clinombr, clitipo, clistat, job_name, program_library, program_name
376 FROM TABLE (RSLAUD.DISPLAY_JOURNAL_ERPPFILE_GLFCLIE(
377 STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current_timestamp - 150 minutes))
378 where program_library not in ('ERPPGM', 'ERPPTF')
379 and not (job_name = 'QZDASSINIT' and remote_address = '123.456.789' and user_name = 'ERPSRVUSER');
380
```

ENTRY_TIMESTAMP	USER_NAME	OPERATION	CLICOMP	CLINUMERO	CLINOMBR	CLITIPO	CLISTAT	JOB_NAME	PROGRAM_LIBRARY	PROGRAM_NAME
2026-08-02 15:10:1...	GM	UPDATE (before)	001	5915000000000001	John Ford (Jr.)	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (after)	001	5915000000000001	John Ford (Jr.)	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (before)	002	4915000000000001	Petros CHRISTOFIDES	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (after)	002	4915000000000001	Petros CHRISTOFIDES	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (before)	001	5915000000000001	John Ford (Jr.)	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (after)	001	5915000000000001	John Ford (Jr.)	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (before)	002	4915000000000001	Petros CHRISTOFIDES	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10:1...	GM	UPDATE (after)	002	4915000000000001	Petros CHRISTOFIDES	EEEE	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:11:1...	GM	UPDATE (before)	001	4915000000000002	Mr Jaime GONZALES	DDDD	Y	QPADEV0005	QTEMP	QDZTD00001
2026-08-02 15:11:1...	GM	UPDATE (after)	001	4915000000000002	Mr Jaime GONZALES	DDDD	Y	QPADEV0005	QTEMP	QDZTD00001

```
381 -- using data-journal reader, check for updates made by a program outside of the ERP application to a sensitive file and a sensitive column
382 SELECT ub.entry_timestamp, ub.user_name, 'UPDATE' operation, ub.clicomp, ub.clinumero, ub.clinombr,
383 ub.clitipo CLITIPO_before, up.clitipo CLITIPO_after, ub.clistat CLISTAT_before, up.clistat CLISTAT_after,
384 ub.job_name, ub.program_library, ub.program_name
385 FROM TABLE (RSLAUD.DISPLAY_JOURNAL_ERPPFILE_GLFCLIE(JOURNAL_ENTRY_TYPES => 'UB',
386 STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current_timestamp - 150 minutes)) ub
387 join TABLE (RSLAUD.DISPLAY_JOURNAL_ERPPFILE_GLFCLIE(JOURNAL_ENTRY_TYPES => 'UP',
388 STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current_timestamp - 150 minutes)) up
389 on ub.sequence_number+1 = up.sequence_number
390 where ub.program_library not in ('ERPPGM', 'ERPPTF')
391 and not (ub.job_name = 'QZDASSINIT' and ub.remote_address = '123.456.789' and ub.user_name = 'ERPSRVUSER')
392 and (ub.clitipo <> up.clitipo or ub.clistat <> up.clistat);
393
```

ENTRY_TIMESTAMP	USER_NAME	OPERATION	CLICOMP	CLINUMERO	CLINOMBR	CLITIPO_BEFORE	CLITIPO_AFTER	CLISTAT_BEFORE	CLISTAT_AFTER	JOB_NAME	PROGRAM_LIBRARY	PROGRAM_NAME
2026-08-02 15:10...	GM	UPDATE	001	5915000000000001	John Ford (Jr.)	EEEE	EEEE	Y	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10...	GM	UPDATE	002	4915000000000001	Petros CHRISTOFIDES	EEEE	EEEE	Y	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10...	GM	UPDATE	001	5915000000000001	John Ford (Jr.)	EEEE	EEEE	Y	Y	QZDASOINIT	QSYS	QZDASOINIT
2026-08-02 15:10...	GM	UPDATE	002	4915000000000001	Petros CHRISTOFIDES	EEEE	EEEE	Y	Y	QZDASOINIT	QSYS	QZDASOINIT

Powerful at filtering and presenting!!!

Filtering at database file field level

Filtering at journal context field level



Don't forget to :

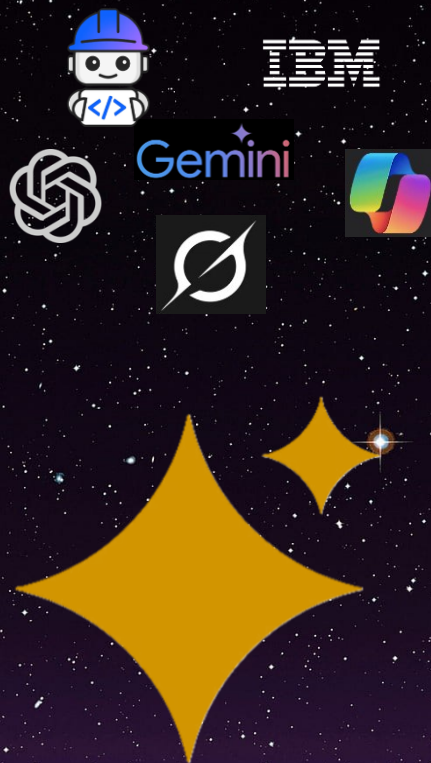
Manage the retention

Manage the change in the file structure

Don't miss any period of time when scheduled (no *CONTINUE value)

Ask AI Questions...

What does the Special Authority *AUDIT allow on IBM i?



Source	Reliability (0=>5)	Consistency (0=>5)	What's wrong?	What's missing?
Bob				
ChatGPT				
Copilot				
Gemini				
Grok				
Security Reference Guide				
Online help				

DISCLAIMER : Answers collected from AI have been done specific days during July 2026 and may be different on another day

What does the Special Authority *AUDIT allow on IBM i?



Bob's answer



What's wrong?	What's missing?
	CHGAUD command
	CHGDLOAUD command
	QSYS2.SECURITY_INFO service from 7.6

🔒 Security Architecture View

In IBM i's security model, special authorities are system-wide privileges that go beyond object-level authority. *AUDIT specifically controls:

Capability	Details
Change user auditing	Modify the AUDLVL (audit level) parameter on user profiles via CHGUSRPRF
Change object auditing	Modify the OBJAUD (object auditing) attribute on any object via CHGOBJAUD
Change system auditing values	Change system values like QAUDLVL, QAUDLVL2, QAUDCTL, QAUDJRN
Control the audit journal	Manage the security audit journal (QAUDJRN in QSYS)

⚙️ Commands Gated by *AUDIT

These commands **require** *AUDIT special authority to succeed:

Command	Description
CHGUSRPRF AUDLVL(...)	Change what actions are audited for a specific user
CHGOBJAUD	Change auditing on a specific object
CHGSYSVAL SYSVAL(QAUD*)	Change audit-related system values
CHGUSRAUD	Change user auditing settings

What does the Special Authority *AUDIT allow on IBM i?

On IBM i, the special authority `*AUDIT` allows a user to **control security auditing functions**. It is intended for security officers and auditors who manage the system's audit journal.

With `*AUDIT`, a user can:

- Start or end system auditing.
- Change auditing-related system values (such as `QAUDCTL`, `QAUDLVL`, and related audit configuration values).
- Configure what events are written to the audit journal.
- Manage aspects of the audit journal used to record security-related events.

What it does *not* allow

`*AUDIT` does **not** automatically grant:

- General security administration (`*SECADM` does that).
- Full system control (`*ALLOBJ` or `*SERVICE` provide broader capabilities).
- Authority to access all objects on the system.

Typical use case

An organization might grant `*AUDIT` to:

- A security administrator responsible for configuring auditing.
- An internal or external auditor who needs to manage audit settings without broader administrative privileges.

What's wrong?	What's missing?

What does the Special Authority *AUDIT allow on IBM i?

On IBM i, the special authority ***AUDIT** allows a user to **view and change auditing characteristics** for the system, objects, and user profiles. [\[ibm.com\]](#), [\[ibm.com\]](#)

A user with ***AUDIT** can:

- Change and display system values that control auditing. [\[ibm.com\]](#)
- Use commands such as **CHGAUD**, **CHGOBJAUT**, and **CHGDLOAUD** to modify object auditing settings. [\[ibm.com\]](#)
- Use **CHGUSRAUD** to modify user auditing settings. [\[ibm.com\]](#)
- Display an object's auditing configuration. [\[ibm.com\]](#)
- Display a user profile's auditing configuration. [\[ibm.com\]](#)
- Run certain security tools, such as **PRTADPOBJ**. [\[ibm.com\]](#)

Security Considerations

***AUDIT** does **not** grant access to all objects or security administration functions, but it is still a sensitive authority because a user can:

- Start or stop auditing.
- Reduce or disable auditing for specific objects or users.
- Potentially hide security-relevant activities from audit trails. [\[ibm.com\]](#), [\[nicklitten.com\]](#)

Because of this, IBM recommends carefully controlling who receives ***AUDIT** authority. [\[ibm.com\]](#)



Copilot's answer



What's wrong?	What's missing?
CHGOBJAUT/CHGOBJAUD	QSYS2.SECURITY_INFO service from 7.6

What does the Special Authority *AUDIT allow on IBM i?

The *AUDIT special authority on IBM i provides users with the ability to view and change the system's auditing characteristics.

Specifically, users with this authority can perform the following tasks:

- **Modify Auditing Configurations:** Change and display the system values that control auditing.
- **Manage Object Auditing:** Use commands such as CHGOBJAUT, CHGDLOAUD, and CHGAUD to configure auditing for specific objects.
- **Manage User Auditing:** Use the CHGUSRAUD command to change auditing settings for individual user profiles.
- **View Auditing Data:** Display auditing values for both user profiles and system objects.
- **Run Security Tools:** Execute many of the security tool commands, such as those found in GO SECTOOLS.

Security Note: Because users with *AUDIT authority can start or stop auditing and potentially prevent specific actions from being recorded, it is critical to carefully control and monitor who is granted this authority to prevent the concealment of security-relevant activities.

What's wrong?	What's missing?
CHGOBJAUT/CHGOBJAUD	QSYS2.SECURITY_INFO service from 7.6

What does the Special Authority *AUDIT allow on IBM i?

1ST TO 3RD TIME



Grok's answer



What does the special authority *AUDIT allow on IBM i?

The *AUDIT special authority is a powerful but focused system right. It allows a user to:

- Read and work with entries in the Security Audit Journal (QAUDJRN).
- Use audit-related commands and APIs, such as:
 - DSPJRN on QAUDJRN
 - DISPLAY_JOURNAL () SQL table function
 - RTVJRNE , RCVJRNE
 - CPYAUDJRNE , PRTTRC , etc.
- Access security audit data even without having *ALLOBJ or broad object authorities.

It is the minimum authority needed to review the system's security logs without giving full administrative power.

What's wrong?	What's missing?
*ALL	*ALL

What does the Special Authority *AUDIT allow on IBM i?

4th TIME

***AUDIT** special authority on IBM i allows a user to manage and control the system auditing functions.

What *AUDIT specifically allows:

- Change system values related to auditing:
 - QAUDCTL
 - QAUDLVL
 - QAUDLVL2
 - QAUDFRCLVL
 - etc.
- Start/stop auditing on the system.
- Enable or disable object auditing (CHGOBJAUD) on any object.
- Enable or disable user auditing (CHGUSRAUD) on any user profile.
- Work with the audit journal (QAUDJRN).
- Use commands like:
 - CHGOBJAUD
 - CHGUSRAUD
 - DSPAUDJRNE (in some contexts)
 - STRSECAUD / ENDSECAUD (indirectly)



Grok's answer



What's wrong?	What's missing?
Work with the audit journal (QAUDJRN)	CHGAUD command
STRSECAUD/CHGSECAUD	CHGDLOAUD command
ENDSECAUD does not exist	QSYS2.SECURITY_INFO service from 7.6

What does the Special Authority *AUDIT allow on IBM i?



Security Reference Guide's answer



Setting up security auditing

With security auditing, you can collect information about security events in the QAUDJRN journal.

Overview:

Purpose:

Set up the system to collect security events in the QAUDJRN journal.

How To:

Required authorities

- CRTJRNRCV *ADD to journal receiver library
- CRTJRN QSYS/QAUDJRN *ADD to QSYS library (*PUBLIC = *USE)
- WRKSYSVAL *SEC *AUDIT special authority (for audit values only)
- CHGOBJAUD *AUDIT special authority
- CHGDLOAUD *AUDIT special authority
- CHGUSRAUD *AUDIT special authority

Authority:

- *ADD authority to QSYS and to journal receiver library
- *AUDIT special authority

Journal Entry:

- CO (create object)
- SV (system value change)
- AD (object and user audit changes)

Note:

QSYS/QAUDJRN must exist before QAUDCTL can be changed, otherwise the system auditing function doesn't know the journal name and won't find it.

What's wrong?	What's missing?
CHGOBJAUT/CHGOBJAUD	QSYS2.SECURITY_INFO service from 7.6

What does the Special Authority *AUDIT allow on IBM i?



Security Reference Guide's answer



Auditing security tools commands

These commands enable you to work with security auditing, the entries from the security audit journal and the system values that control security auditing.

For more information about the security tools, see [Appendix G, "Commands and menus for security commands,"](#) on page 911.

Table 150. Auditing security tools commands		
Command name	Descriptive name	Function
CHGSECAUD	Change Security Auditing	Set up security auditing and to change the system values that control security auditing.
CPYAUDJRNE	Copy Audit Journal Entries	Copy entries from the security audit journal to output files that you can query. You can select specific entry types, specific users, and a time period.
DSPAUDJRNE ¹	Display Audit Journal Entries	Display or print information about entries in the security audit journal. You can select specific entry types, specific users, and a time period.
DSPSECAUD	Display Security Auditing Values	Display information about the security audit journal and the system values that control security auditing.
¹ IBM has stopped providing enhancements for the DSPAUDJRNE command. The command does not support all security audit record types, and the command does not list all the fields for the records it does support.		

What's wrong?	What's missing?
CHGOBJAUT/CHGOBJAUD	QSYS2.SECURITY_INFO service from 7.6

CPYAUDJRNE ⁸	Output file already exists	*OBJOPR, *OBJMGT, *ADD, *DLT	*EXECUTE
	Output file does not exist		*EXECUTE, *ADD
CRTJRN	Journal		*READ, *ADD
	Journal receiver	*OBJOPR, *OBJMGT, *READ	*EXECUTE
DLTJRN	Journal	*OBJOPR, *OBJEXIST	*EXECUTE
DSPAUDJRNE ⁸			

⁸ You must have *AUDIT special authority to use this command.
--

Only Grok was able to find them!!

What does the Special Authority *AUDIT allow on IBM i?



Security Reference Guide's answer



*AUDIT special authority

Audit (*AUDIT) special authority gives the user the ability to view and change auditing characteristics.

A user can perform the following tasks with the *AUDIT special authority:

- Change and display the system values that control auditing.
- Use the CHGOBJAUT, CHGDLOAUD, and CHGAUD commands to change auditing for objects.
- Use the CHGUSRAUD command to change auditing for a user.
- Display an object's auditing values.
- Display a user profile's auditing values.
- Run some of the security tool commands, such as PRTADPOBJ.

Risks: A user with *AUDIT special authority can stop and start auditing on the system or prevent auditing of particular actions. If having an audit record of security-relevant events is important for your system, carefully control and monitor the use of *AUDIT special authority.

To prevent general users from viewing auditing information, restrict general users' access to the following information:

- The security audit journal (QAUDJRN)
- Other journals that contain auditing data
- Save files, outfiles, spool files, and printed output that contain auditing information

Note: Only a user with *ALLOBJ, *SECADM, and *AUDIT special authorities can give another user *AUDIT special authority.

What's wrong?	What's missing?
CHGOBJAUT/CHGOBJAUD	QSYS2.SECURITY_INFO service from 7.6

The Display Security Attributes (DSPSECA), command, the Retrieve Security Attributes (QSYRTVSA) API, and QSYS2.SECURITY_INFO now require *SECADM or *AUDIT special authority to use on IBM i 7.6.

What does the Special Authority *AUDIT allow on IBM i?



LESSON: Multiply your sources!

1. Surprisingly, Grok helped uncover missing authorities hidden behind the special authority *AUDIT requirement.
2. Only Bob and Grok managed to fix the mistake in the Security Reference Guide (CHGOBJAUT versus CHGOBJAUD)

Source	Reliability (0=>5)	Consistency (0=>5)	What's wrong?	What's missing?
Bob 	4	4	0	3
ChatGPT	2	1	0	0
Copilot 	4	4	1	1
Gemini	3	3	1	1
Grok (1st to 3rd questions)	0	0	99	99
Grok (4th question)	3	3	3	3
 Security Reference Guide	5	5	1	1
 Online help	5	5		

Authorities required to query a database journal using QSYS2.DISPLAY_JOURNAL

```
select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) record_changed
from table( qsys2.Display_Journal ('GMJRN','JRN', Journal_Codes => 'R', STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days, object_library => 'GMFILE', object_name => 'GLFCLIEN', object_objtype => '*FILE', object_member => '*FIRST')) jrn ;
```

AUTHORIZATION _NAME	SYSTEM_OBJECT _SCHEMA	SYSTEM_OBJECT _NAME	SYSTEM_OBJECT _TYPE	AGGREG_REQUIRED_AUTHORITY
GM_ALLOBJ	GMFILE	GLFCLIEN	*FILE	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMJRN	JRN	*JRN	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMJRNRCV	RCV0001	*JRNRCV	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	QSYS	GMFILE	*LIB	*EXECUTE
GM_ALLOBJ	QSYS	GMJRN	*LIB	*EXECUTE

```
select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) record_changed
from table( qsys2.Display_Journal ('GMJRN','JRN', Journal_Codes => 'R', STARTING_RECEIVER_LIBRARY => 'GMJRNRCV', STARTING_RECEIVER_NAME => 'RCV0001',
STARTING_TIMESTAMP => current date - 1 days, object_library => 'GMFILE', object_name => 'GLFCLIEN', object_objtype => '*FILE', object_member => '*FIRST')) jrn ;
```

AUTHORIZATION _NAME	SYSTEM_OBJECT _SCHEMA	SYSTEM_OBJECT _NAME	SYSTEM_OBJECT _TYPE	AGGREG_REQUIRED_AUTHORITY
GM_ALLOBJ	GMFILE	GLFCLIEN	*FILE	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMJRN	JRN	*JRN	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMJRNRCV	RCV0001	*JRNRCV	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	QSYS	GMFILE	*LIB	*EXECUTE
GM_ALLOBJ	QSYS	GMJRN	*LIB	*EXECUTE
GM_ALLOBJ	QSYS	GMJRNRCV	*LIB	*EXECUTE

```
select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) record_changed
from table( qsys2.Display_Journal ('GMJRN','JRN', Journal_Codes => 'R', STARTING_RECEIVER_LIBRARY => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn ;
```

AUTHORIZATION _NAME	SYSTEM_OBJECT _SCHEMA	SYSTEM_OBJECT _NAME	SYSTEM_OBJECT _TYPE	AGGREG_REQUIRED_AUTHORITY
GM_ALLOBJ	GMFILE	GLFCLIEN	*FILE	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMFILE	GLFTRANS	*FILE	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	GMJRN	JRN	*JRN	*EXECUTE *OBJEXIST *OBJOPR *READ
GM_ALLOBJ	GMJRNRCV	RCV0001	*JRNRCV	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	QSYS	GMJRN	*LIB	*EXECUTE

Authorities required to query a database journal using DSPJRN

Display Journal (DSPJRN)

- o *USE is required for all objects that exist on the system in the QSYS file system that are specified on the FILE and OBJ parameters.

DISPLAY_JOURNAL table function

Last Updated: 2025-10-16

The DISPLAY_JOURNAL table function returns information about journal entries. It returns information similar to what is returned by the Display Journal (DSPJRN) CL command and the Retrieve Journal Entries (QjoRetrieveJournalEntries) API.

Authorization:

- The caller must have *USE authority to the journal and *EXECUTE authority to the library containing the journal.
- The caller must have *USE authority to every requested journal receiver and *EXECUTE authority to the library containing the journal receiver.
- *OBJEXIST authority is required to the journal if *object-name* is omitted or if *object-name* specifies an object that no longer exists.
- If *object-name* is *ALL, the caller must be authorized to every object associated with a journal entry.
 - The caller must have:
 - *USE authority to the object, and
 - *EXECUTE authority to the library containing the object.

LESSON

Another good reason to use DISPLAY_JOURNAL

DISPLAY_JOURNAL requires *USE authority for each file for which journal entries have been retrieved (whether selected or not).

DSPJRN requires *USE only on the selected files.

DSPJRN does not require *USE on the files when not selected !

Authorities required to query the system audit journal QAUDJRN

```
select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) image
from table( qsys2.Display_Journal ('QSYS','QAUDJRN', Journal_Codes => 'T', STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn
where journal_entry_type = 'PW' ;
```

```
select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) image
from table( qsys2.Display_Journal ('QSYS','QAUDJRN', Journal_Codes => 'T', STARTING_RECEIVER_LIBRARY => 'QGPL', STARTING_RECEIVER_NAME => 'AUDRCV0004',
STARTING_TIMESTAMP => current date - 1 days)) jrn
where journal_entry_type = 'PW' ;
```

```
SELECT entry_timestamp, jrn.user_name, job_name, job_user, program_library, program_name, remote_address,
VIOLATION_TYPE,VIOLATION_TYPE_DETAIL, AUDIT_USER_NAME
FROM TABLE (SYSTOOLS.AUDIT_JOURNAL_PW (STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn ;
```

```
c1: DSPJRN JRN(QAUDJRN) RCVRNG(*CURCHAIN) ENTTPW(PW) OUTPUT(*PRINT) ;
```

AUTHORIZATION _NAME	SYSTEM_OBJECT _SCHEMA	SYSTEM_OBJECT _NAME	SYSTEM_OBJECT _TYPE	AGGREG_REQUIRED_AUTHORITY
GM_ALLOBJ	QGPL	AUDRCV0004	*JRNRCV	*EXECUTE *OBJOPR *READ
GM_ALLOBJ	QSYS	QAUDJRN	*JRN	*EXECUTE *OBJEXIST *OBJOPR *READ

```
31 c1: DSPAUDJRNE ENTTPW(PW) JRNRCV(*CURCHAIN) OUTPUT(*PRINT) ;

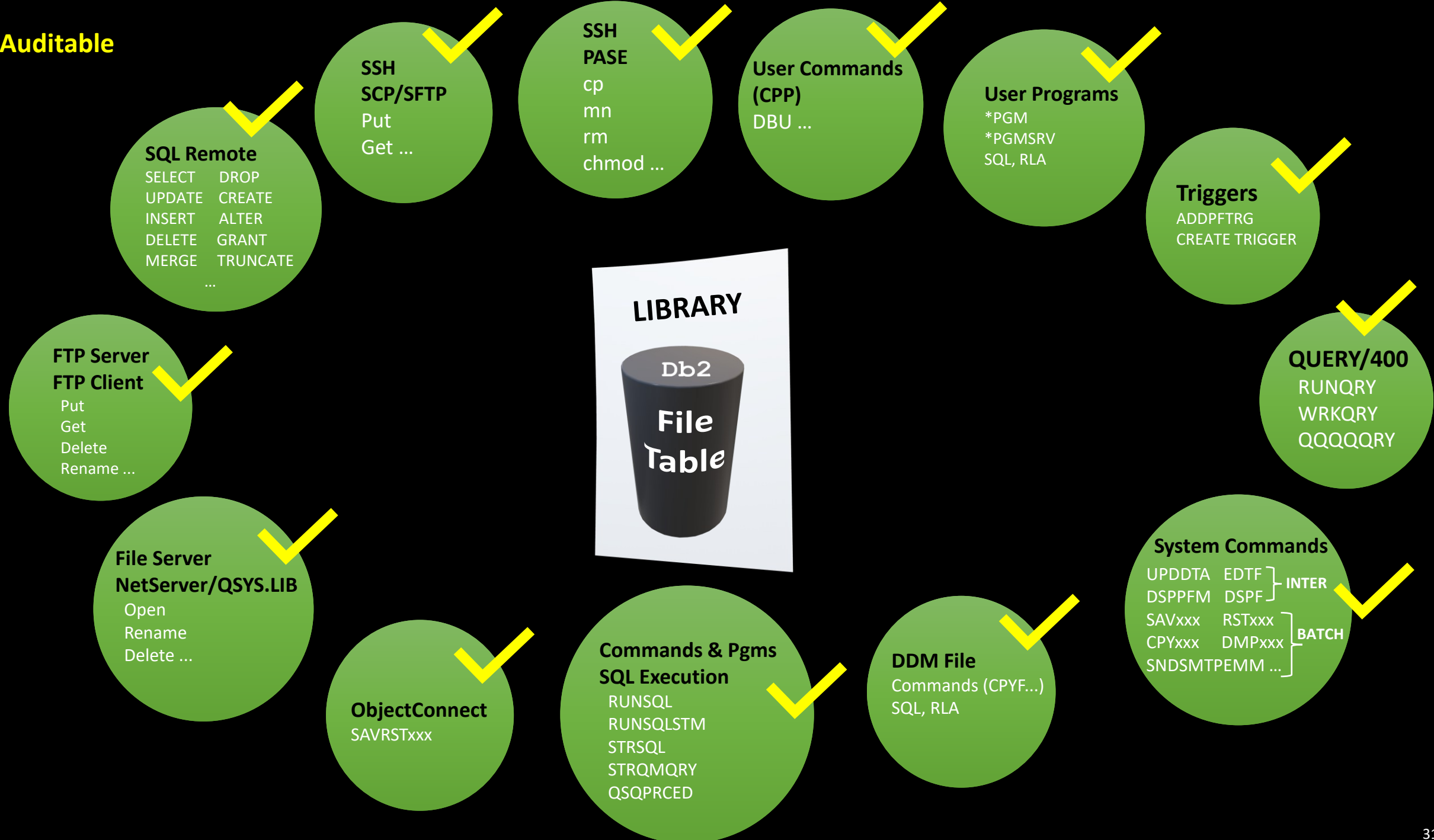
[ 07/14/2026, 05:27:35 PM ] Run Selected...
DSPAUDJRNE ENTTPW(PW) JRNRCV(*CURCHAIN) OUTPUT(*PRINT)

CPFB304: User does not have required special authorities.
```

AUTHORIZATION _NAME	SYSTEM_OBJECT _SCHEMA	SYSTEM_OBJECT _NAME	SYSTEM_OBJECT _TYPE	AGGREG_REQUIRED_AUTHORITY

```
c1: CPYAUDJRNE ENTTPW(PW) JRNRCV(*CURCHAIN) ;
```

Once the user has been granted *AUDIT, commands DSPAUDJRNE & CPYAUDJRNE bypass the required authorities on QAUDJRN and its journal receivers



PROTECTION

LIBRARY

Db2
File
Table

SQL Remote

SELECT DROP
UPDATE CREATE
INSERT ALTER
DELETE GRANT
MERGE TRUNCATE
...

SSH
SCP/SFTP
Put
Get ...

SSH
PASE
cp
mn
rm
chmod ...

User Commands
(CPP)
DBU ...

User Programs
*PGM
*PGMSRV
SQL, RLA

Triggers

ADDPFTRG
CREATE TRIGGER

QUERY/400

RUNQRY
WRKQRY
QQQQQRY

System Commands

UPDDTA	EDTF	} INTER
DSPPFM	DSPF	
SAVxxx	RSTxxx	} BATCH
CPYxxx	DMPxxx	
SNDSMTPPEMM	...	

DDM File

Commands (CPYF...)
SQL, RLA

Commands & Pgms

SQL Execution

RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

ObjectConnect

SAVRSTxxx

File Server

NetServer/QSYS.LIB

Open
Rename
Delete ...

FTP Server

FTP Client

Put
Get
Delete
Rename ...

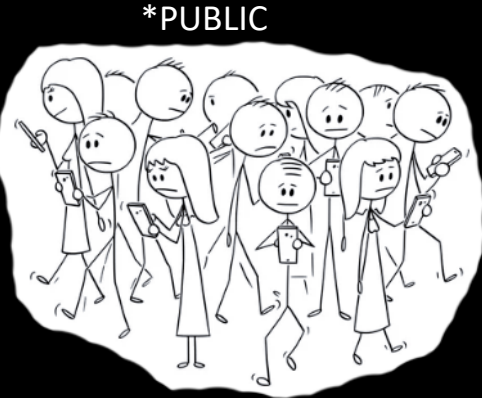
Object Level Security - Static Authorities Foundation



Objet
Level



USER PROFILE



*PUBLIC

Authority Checking
Process

AUTHORIZATION LIST
GROUP PROFILE(S)
SPECIAL AUTHORITY *ALLOBJ
PRIMARY GROUP

PGM
Adopted Authority



✓ **Protectable**
Object Level
Security

SQL Remote

SELECT	DROP
UPDATE	CREATE
INSERT	ALTER
DELETE	GRANT
MERGE	TRUNCATE
...	

**SSH
SCP/SFTP**

Put
Get ...

**SSH
PASE**

cp
mn
rm
chmod ...

**User Commands
(CPP)**

DBU ...

User Programs

*PGM
*PGMSRV
SQL, RLA

Triggers

ADDPFTRG
CREATE TRIGGER

QUERY/400

RUNQRY
WRKQRY
QQQQQRY

System Commands

UPDDTA	EDTF	} INTER
DSPPFM	DSPF	
SAVxxx	RSTxxx	} BATCH
CPYxxx	DMPxxx	
SNDSMTPEMM ...		

DDM File

Commands (CPYF...)
SQL, RLA

**Commands & Pgms
SQL Execution**

RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

ObjectConnect

SAVRSTxxx

**File Server
NetServer/QSYS.LIB**

Open
Rename
Delete ...

**FTP Server
FTP Client**

Put
Get
Delete
Rename ...



Function Usage – Basic contextual Access Authorities

Protocol
Level

Protocol - Function
Navigator for i
FTP
Database Server
DDM - DRDA
...

FUNCTION_PRODUCT_ID	NUMBER
QIBM_ACS	2
QIBM_BASE_OPERATING_SYSTEM	22
QIBM_NAV	16
QIBM_QINAV_NAVIGATOR_WEB	3
QIBM_QSY_DIGITAL_CERT_MGR	1
QIBM_QTM_TCPIP	18
QIBM_QTMS_TCPIP	1
QIBM_QYCM_CIMOM	11
QIBM_QYPS_MGTCTRL	1
QIBM_XD1_OPNAV	75

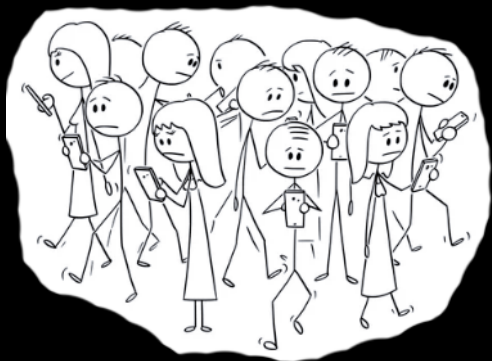
Removed in 7.6:

QIBM_QYCM_CIMOM	11
QIBM_QYPS_MGTCTRL	1
QIBM_XD1_OPNAV	75

USER PROFILE



*PUBLIC



By default: *ALLOWED or *DENIED
Access specified for *ALLOBJ profiles

QIBM_NAV_ALL_FUNCTION	New Nav Access	*DENIED
QIBM_NAV_*	New Nav functions	*ALLOWED
QIBM_DB_ZDA	ODBC	*ALLOWED
QIBM_DB_DDMDRDA	DDM & DRDA	*DENIED
QIBM_QTMF*	FTP	*ALLOWED

Function ID	Default Usage	Allobj Indicator	Function Name Message Text	New in 7.6
QIBM_IOSYSCFG_DDM	DENIED	NOT USED	DDM (CRT/CHG/DLT/DSP/WRK) command access	
QIBM_IOSYSCFG_VIEW	DENIED	NOT USED	View Input/Output System Configuration	
QIBM_QZLS_NETSVR_SHARE	DENIED	NOT USED	Allow object owner shares	
QIBM_RUN_UNDER_USER_NO_AUTH	ALLOWED	NOT USED	Run under a user without authentication	

Function Usage in Action

User Name	Usage	User Type	Default Usage	Allobj Indicator	Function ID	Function Name Message Text
EPIADM	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
GM	ALLOWED	GROUP	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
GM_BASICLM	DENIED	USER	ALLOWED	USED	QIBM_DB_ZDA	Toolbox Application Server Access
GRP_EXPLT	ALLOWED	GROUP	DENIED	USED	QIBM_ACCESS_ALLOBJ_JOBLOG	Access job log of *ALLOBJ job
JPLTOOLS	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SQLADM	Database Administrator
JPLTOOLS	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
QDIRSRV	ALLOWED	USER	DENIED	USED	QIBM_QSY_SYSTEM_CERT_STORE	*SYSTEM certificate store
QOBJC	ALLOWED	USER	DENIED	USED	QIBM_QSY_SYSTEM_CERT_STORE	*SYSTEM certificate store
QTCP	ALLOWED	USER	DENIED	USED	QIBM_QSY_SYSTEM_CERT_STORE	*SYSTEM certificate store
QYPSJSVR	ALLOWED	USER	DENIED	USED	QIBM_QSY_SYSTEM_CERT_STORE	*SYSTEM certificate store

```
-- check if the effective user is allowed by the function
values(qsys2.SQL_CHECK_FUNCTION_USAGE('QIBM_DB_ZDA')) ;
```

01

1

```
-- list functions default values and usages
select us.user_name, us.usage, user_type, default_usage, allobj_indicator, us.function_id, FUNCTION_NAME_MESSAGE_TEXT
FROM QSYS2.FUNCTION_USAGE us join QSYS2.FUNCTION_INFO fu on us.function_id = fu.function_id
order by us.user_name;

-- add ADMIN group to QIBM_Q1A* (BRMS) if default = DENIED
SELECT fu.function_id, default_usage, allobj_indicator,
'CHGFCNUSG USER(GRP_ADMIN) USAGE(*ALLOWED) FCNID(' concat fu.function_id concat ')' command,
qcmdexc('CHGFCNUSG USER(GRP_ADMIN) USAGE(*ALLOWED) FCNID(' concat fu.function_id concat ')' ) cmd_result
FROM QSYS2.FUNCTION_INFO fu
where fu.function_id like 'QIBM_Q1A%' and default_usage = 'DENIED'
ORDER BY fu.function_id ;

-- grant GRP_EXPLT to QIBM_ACCESS_ALLOBJ_JOBLOG
SELECT authorization_name,
'CHGFCNUSG USER(' concat authorization_name concat ')' USAGE(*ALLOWED) FCNID(QIBM_ACCESS_ALLOBJ_JOBLOG)' command,
qcmdexc('CHGFCNUSG USER(' concat authorization_name concat ')' USAGE(*ALLOWED) FCNID(QIBM_ACCESS_ALLOBJ_JOBLOG)' ) cmd_result
FROM qsys2.user_info up where authorization_name in ('GRP_EXPLT') ;
```

Authorization Name	COMMAND	CMD_RESULT
GRP_EXPLT	CHGFCNUSG USER(GRP_EXPLT) USAGE(*ALLOWED) FCNID(QIBM_ACCESS_ALLOBJ_JOBLOG)	1

Function Usage in Action

```
-- monitor USAGE FAILURES
select jrn.ENTRY_TIMESTAMP, function_registration_operation, qualified_job_name, user_name, program_name,
user_profile_name, function_name
from table(SYSTOOLS.AUDIT_JOURNAL_GR (STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current date - 1 days)) jrn
where entry_type = 'F' and function_registration_operation = 'USAGE FAILURE' ;
```

ENTRY_TIMESTAMP	FUNCTION_REGISTRATION_OPERATION	QUALIFIED_JOB_NAME	USER_NAME	PROGRAM_NAME	USER_PROFILE_NAME	FUNCTION_NAME
2026-07-23 15:25...	USAGE FAILURE	816199/QTCP/QTFTP00022	QTCP	QTMFSRVR	GM_BASIC	QIBM_QTMF_SERVER_REQ_0
2026-07-23 15:25...	USAGE FAILURE	816199/QTCP/QTFTP00022	QTCP	QTMFSRVR	GM_BASIC	QIBM_QTMF_SERVER_REQ_10
2026-07-23 15:26...	USAGE FAILURE	819823/QUSER/QZRCRSVS	GM_BASIC	QZRCRSVS	GM_BASIC	QIBM_NAV_ALL_FUNCTION
2026-07-23 15:31...	USAGE FAILURE	819615/QUSER/QZDASOINIT	GM_BASICLM	QZDASOINIT	GM_BASICLM	QIBM_DB_ZDA
2026-07-23 15:38...	USAGE FAILURE	819830/QSECOFR/QPADEV000B	QSECOFR	QCMD	QSECOFR	QIBM_DB_SECADM

```
-- monitor JOBLLOG ACCESS FAILURES
SELECT jrn.ENTRY_TIMESTAMP, qualified_job_name, program_name, user_name,
jrn.VIOLATION_TYPE_DETAIL, jrn.USER_PROFILE_NAME, jrn.OBJECT_NAME, jrn.OBJECT_TYPE
FROM TABLE (SYSTOOLS.AUDIT_JOURNAL_AF (STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 1 days)) jrn
where jrn.VIOLATION_TYPE = 'K' and object_name = 'DSPJOBLOG';
```

ENTRY_TIMESTAMP	QUALIFIED_JOB_NAME	PROGRAM_NAME	USER_NAME	VIOLATION_TYPE_DETAIL	USER_PROFILE_NAME	OBJECT_NAME	OBJECT_TYPE
2026-07-23 15:23:...	819783/GM_BASIC/QPADEV000B	QCMD	... GM_BASIC	Special authority violation	GM_BASIC	DSPJOBLOG	*CMD

Parameters or command
==> dspjoblog 815928/EPI/MONDTAQ
F3=Exit F5=Refresh F7=Find
F11=Display elapsed data F12=Canc
Not authorized to display job log.

```
C:\Users\GUY>ftp 123.456.6.78
Connecté à 123.456.6.78.
220-QTCP at MOKELE.MBE.LOCAL.
220 Connection will close if idle more than 5 minutes.
501 OPTS unsuccessful; specified subcommand not recognized.
Utilisateur (123.456.6.78:(none)) : gm_basic
331 Enter password.
Mot de passe :

530 Log on attempt by user GM_BASIC rejected.
Échec de l'identification.
ftp>
```



```
✖ Etat SQL : 08S01
Code fournisseur : -99999
Message : Communication link failure. (Connection was dropped unexpectedly.)
```

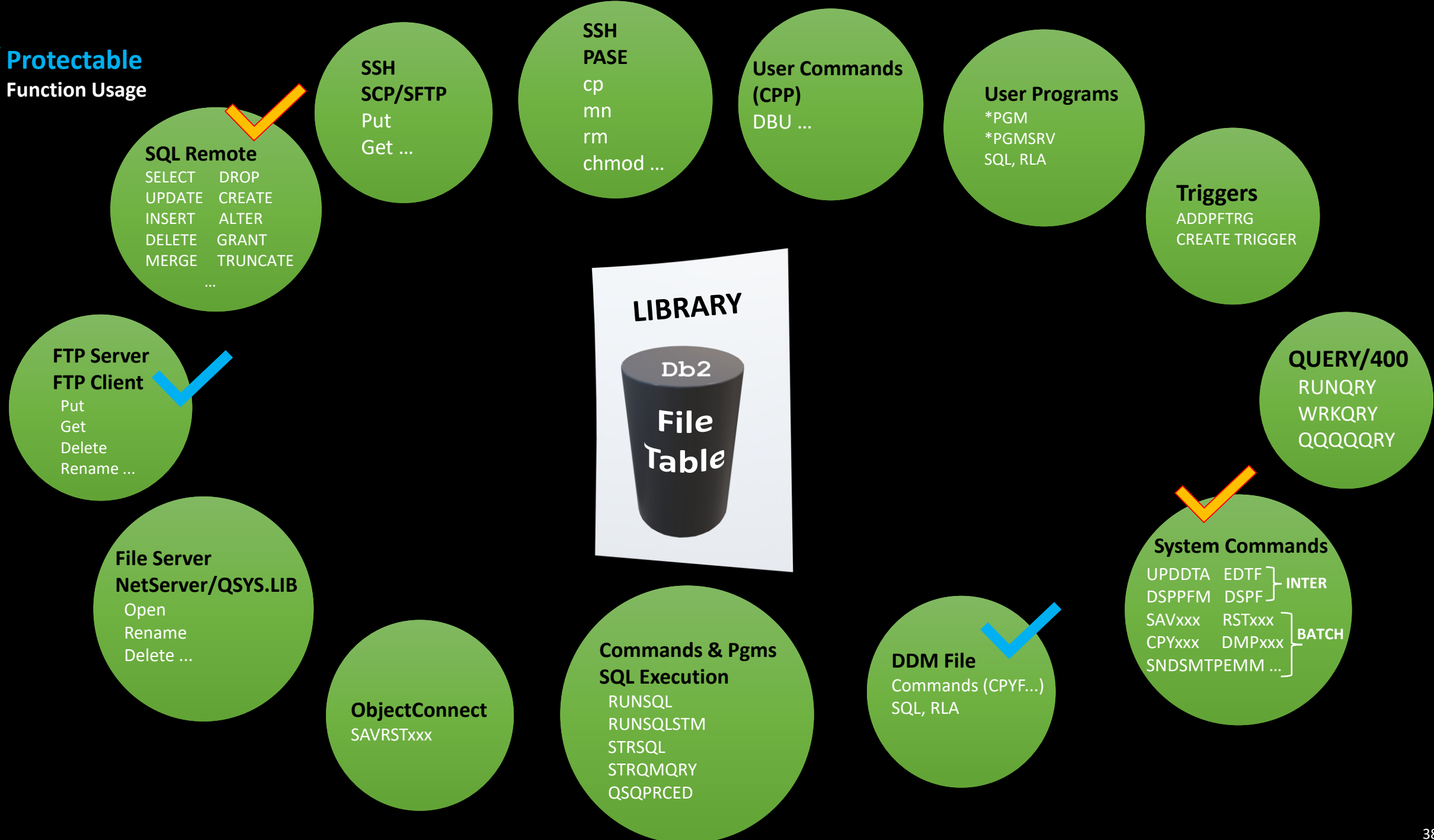
Enter SQL Statements

Type SQL statement, press Enter.
>

```
create permission gm.gm_authorized_only on gm.testsec for rows
where user = 'GM'
enforced for all access enable
Not authorized to CREATE PERMISSION.
```

==>

✓ **Protectable**
Function Usage



RCAC/Row – Access Authorities by Row

```
-- setup RCAC permissions
create permission erpfile.company_protection on erpfile.glfclien for rows
where (clicomp = '001' and verify_group_for_user(current_user, 'WGRP001') = 1) or
(clicomp = '002' and verify_group_for_user(current_user, 'WGRP002') = 1) or
(clicomp = '003' and verify_group_for_user(current_user, 'WGRP003') = 1) or
(verify_group_for_user(current_user, 'WGRPALL') = 1)
enforced for all access enable ;

alter table erpfile.glfclien activate row access control ;
```

COMPANY			Client	Client				
ID	Client ID	Client Name	Type	Status	Taxe Id	Adress 1	Adress 2	Adress 3
001	4915000000000002	Mr Jaime GONZALES	... CCCCC	N	272-227-2234	... NORTE 15	... test 4	... EGYPT
001	4915310000000011	Christopher Wang	... CCCCC	N	4884554448	... Ocean Drive	... SOUTH AFRICA	... Pretoria
001	5915000000000001	John Ford (Sr.)	... CCCCC	Y	913-073-4574	... Lambert Walk	... Saint Louis	... MISSOURI
002	4915000000000001	Petros CHRISTOFIDES	... EEEEE	Y	275-073-6190	... Avenue Petros	... LEFKOSIA	... CYPRUS
002	4915310000000002	Basel HASSAN	... 4RSBB	X	88888220000-2	... Avenue Jacques Cartie...	DUBAI	... EMIRATES
003	101245894317825	JOHN Watson	... BBBB	X	VAT-118	... Downing Street	... PRETORIA	... SOUTH AFRICA
003	1234567890000001	Pascal Tarloucellowitch	... CCCCC	Y	172-227-2234	... NORTE 15	... Saint-Etienne	... FRANCE
003	1234567890123456	Rastapopoulos	... CCCCC	N	233-073-6290	... RICHELIEU	... Maputo	... MOZAMBIQUE
003	5900100010101011	Jean DURAND	... GOOD	Y	12345678	... Avenue ONTARIO	... Perth	... AUSTRALIA
003	6078787878787878	Ekaterina RAWITZ	... BBBB	Y	BBBB4	... Mining Avenue	... KRAKOW	... POLAND

"Silent" mechanism and independent of interfaces (implicit WHERE clause)

Rejected by default
(condition 0=1)

Covers any row-level operation (read, update, delete, insert)

Triggered after object-level security

Allows to present an "empty" file to an *ALLOBJ user

Exclusive lock required for *alter table*

RCAC/Column - Hiding a Column

```
-- setup RCAC masks
create mask erpfile.taxid_masked on erpfile.glfclien for column clitaxid return
case when verify_group_for_user(current_user, 'WGRPALL') = 1 or
clidir2 = 'Saint-Etienne' then clitaxid
else '*****'
end
enable ;

alter table erpfile.glfclien activate column access control ;
```

COMPANY ID	Client ID	Client Name	Client Type	Client Status	Taxe Id	Adress 1	Adress 2	Adress 3
003	101245894317825	JOHN Watson	... BBBB	X	*****	... Downing Street	... PRETORIA	... SOUTH AFRICA
003	1234567890000001	Pascal Tarloucellowitch	... CCCCC	Y	172-227-2234	... NORTE 15	... Saint-Etienne	... FRANCE
003	1234567890123456	Rastapopoulos	... CCCCC	N	*****	... RICHELIEU	... Maputo	... MOZAMBIQUE
003	5900100010101011	Jean DURAND	... GOOD	Y	*****	... Avenue ONTARIO	... Perth	... AUSTRALIA
003	6078787878787878	Ekaterina RAWITZ	... BBBB	Y	*****	... Mining Avenue	... KRAKOW	... POLAND

FREE!

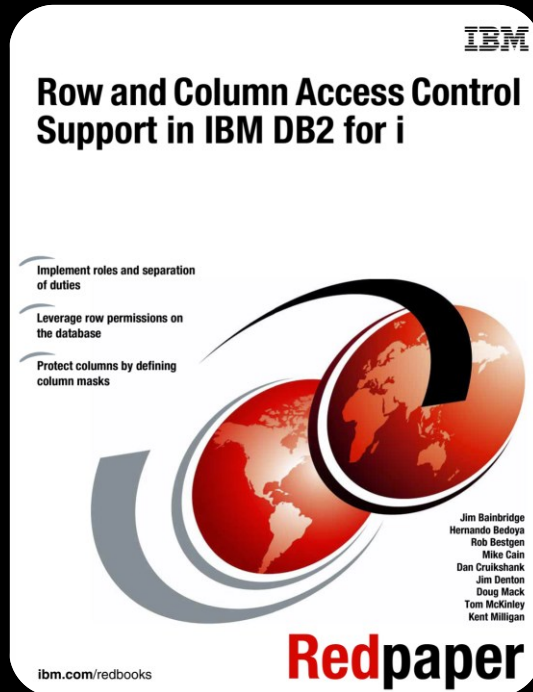
Fairly easy to implement on selected tables

"Silent" mechanism and independent of interfaces

(implicit WHERE clause)

Exclusive lock required for alter table

RCAC – Ressources



Session iUG 2021

Technical feedback - Eddie Chaffin

RCAC Implementation in multi-company M3 environment

(in collaboration with Infor and Kent Milligan

IBM Lab Services)



VIGILANCE
Copy, BackUp, Restore
Data: Updates, aggregation
Performance Impact

Presentation OmniTech 2017

Scott Forstie



RCAC – Peripheral Security

Permission, Mask, Alter Table

- Registration in QIBM_DB_SECADM
- No authorities required on the Table and its Library
- Even QSECOFR must be registered!

Code fournisseur : -552
Message : [SQL0552] Not authorized to CREATE PERMISSION. Cause : The operation cannot be performed without the required authority. -- CREATE TABLE requires *USE authority to the CRTPF command. -- CREATE VIEW or CREATE INDEX requires *USE authority to the CRTLF command. -- CREATE ALIAS requires *USE authority to the CRTDDMF command. -- CREATE SCHEMA requires *USE authority to the CRTLIB command. -- ALTER TABLE requires *USE authority to the ADDPFCST command in order to add constraints, and *USE authority to the PMVPCST command in order to drop constraints. -- ALTER TRIGGER requires *USE authority to the CHGPFTRG command. -- CREATE PROCEDURE or CREATE FUNCTION requires *OBJOPR and *ADD authority to the catalog table SYSROUTINES in QSYS2. -- DROP PROCEDURE or DROP FUNCTION requires *OBJOPR and *DLT authority to the catalog table SYSPARMS in QSYS2. -- CREATE TYPE requires *OBJOPR and *ADD authority to the catalog table SYSTYPES in QSYS2. -- DROP TYPE requires *OBJOPR and *DLT authority to the catalog table SYSTYPES in QSYS2. -- CREATE TRIGGER requires *USE authority to the ADDPFTRG command. -- DROP TRIGGER requires *USE authority to the PMVPCST command. -- CREATE SEQUENCE requires *USE authority to the CRTDTAARA command. -- DROP SEQUENCE requires *USE authority to the DLTDTAARA command. -- ALTER SEQUENCE requires *USE authority to the RTVDTAARA and CRTDTAARA commands. -- The COMMENT ON statements for procedures, functions, types, triggers and sequences require *OBJOPR, *READ, and *UPD authority to the catalog table associated with the object. -- SET SESSION AUTHORIZATION requires that the authorization ID associated with the statement has *ALLOBJ special authority. -- SET CURRENT DEGREE and the QSYS2/RESET_ENVIRONMENTAL_LIMITS procedure require that the authorization ID associated with the statement has *JOBCTL special authority or be authorized to the QIBM_DB_SQLADM function. -- CREATE, ALTER, DROP, LABEL ON and COMMENT ON of a MASK or PERMISSION requires that the authorization ID associated with the statement be authorized to the QIBM_DB_SECADM function. The same authorization is required to RENAME, DROP or delete an object that is referenced by a mask or permission and to CREATE, DROP, or ALTER a secure TRIGGER or FUNCTION. Recovery : Obtain authority from the security officer and try the operation again. Authorization to the QIBM_DB_SQLADM and QIBM_DB_SECADM functions can be handled by Application Administration in System i Navigator. The Change Function Usage (CHGFCNUSG) command can also be used to allow or deny use of the function. For example: CHGFCNUSG FCNID(QIBM_DB_SQLADM) USER(yyyyyy) USAGE(*ALLOWED).

Managing QIBM_DB_SECADM

- *SECADM required – So, *ALLOBJ is sufficient if Privilege Elevations are possible
- Default authority = *DENIED (not changeable)

Display Function Usage					
Function ID :		QIBM_DB_SECADM			
Function name :		Database Security Administrator			
Description :		Database Security Administrator Functions			
Product :		QIBM_BASE_OPERATING_SYSTEM			
Group :		QIBM_DB			
Default authority :		*DENIED			
*ALLOBJ special authority :		*NOTUSED			
User	Type	Usage	User	Type	Usage
EPIADM	User	*ALLOWED			
GM	Group	*ALLOWED			
GM_ALLOBJ	User	*ALLOWED			
GM_BASIC	User	*ALLOWED			
JPLTOOLS	User	*ALLOWED			

QIBM_DB_SECADM allows to :

- Managing authority collection
- Managing RCAC
- Managing QSYS.LIB Authorities (private, public, autl, owner) - Except private Authority for itself

VIGILANCE

Listing RCAC configuration

- No authorities required !!!!!

Still true
in 7.6

```
-- list existing RCAC controls & dependancies
select * from qsys2.syscontrols ;
select * from qsys2.syscontrolsdep ;
```

RCAC – Auditing QIBM_DB_SECADM

-- who is authorized to manipulate RCAC?

```
SELECT fu.user_name, fu.usage, fu.user_type, fi.default_usage, fi.allobj_indicator, fi.function_id, fi.function_name_message_text
FROM QSYS2.FUNCTION_INFO fi left join QSYS2.FUNCTION_USAGE fu on fu.function_id = fi.function_id
where fu.function_id = 'QIBM_DB_SECADM'
order by user_name ;
```

USER_NAME	USAGE	USER_TYPE	DEFAULT_USAGE	ALLOBJ_INDICATOR	FUNCTION_ID	FUNCTION_NAME_MESSAGE_TEXT
User Name	Usage	User Type	Default Usage	Allobj Indicator	Function ID	Function Name Message Text
GM	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
RITBRAD	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
RITSTEFAN	ALLOWED	USER	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator
WGRPALL	ALLOWED	GROUP	DENIED	NOT USED	QIBM_DB_SECADM	Database Security Administrator

-- monitor changes in function QIBM_DB_SECADM

```
select jrn.ENTRY_TIMESTAMP, job_name, user_name, program_name,
remote_address, function_name, function_registration_operation, user_profile_name, usage_setting, previous_usage
from table(SYSTOOLS.AUDIT_JOURNAL_GR (STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current date)) jrn
where entry_type = 'F' and function_name = 'QIBM_DB_SECADM'
and function_registration_operation = 'CHANGE USAGE';
```

ENTRY_TIMESTAMP	JOB_NAME	USER_NAME	PROGRAM_NAME	REMOTE_ADDRESS	FUNCTION_NAME	FUNCTION_REGISTRATION_OPERATION	USER_PROFILE_NAME	USAGE_SETTING	PREVIOUS_USAGE
2026-07-03 08:00:05...	QPADEV0004	GM	QCMD	10.24...	QIBM_DB_SECADM	CHANGE USAGE	RITSTEFAN	ALLOWED	UNKNOWN
2026-07-03 08:00:21...	QPADEV0004	GM	QCMD	10.24...	QIBM_DB_SECADM	CHANGE USAGE	RITBRAD	ALLOWED	UNKNOWN
2026-07-03 08:00:30...	QPADEV0004	GM	QCMD	10.24...	QIBM_DB_SECADM	CHANGE USAGE	WGRPALL	ALLOWED	UNKNOWN
2026-07-03 08:12:14...	QPADEV0004	GM	QCMD	10.24...	QIBM_DB_SECADM	CHANGE USAGE	WGRPALL	DENIED	ALLOWED
2026-07-03 08:12:25...	QPADEV0004	GM	QCMD	10.24...	QIBM_DB_SECADM	CHANGE USAGE	WGRPALL	REMOVED	DENIED

RCAC – Auditing RCAC Functions

```
-- monitor changes in RCAC (QAUDJRN)
select jrn.entry_timestamp, jrn.user_name, jrn.job_name, jrn.program_name, jrn.remote_address, jrn.entry_type, jrn.entry_type_detail,
jrn.operation_type, jrn.library_name, jrn.file_name, jrn.mask_name, jrn.column_name, jrn.permission_name, jrn.enabled, jrn.row_access_control, jrn.column_access_control,
jrn.prev_enabled, jrn.prev_row_access_control, jrn.prev_column_access_control, jrn.sql_statement_text, jrn.statement_truncated
from table(SYSTOOLS.AUDIT_JOURNAL_AX (STARTING_RECEIVER_NAME => '*CURCHAIN', STARTING_TIMESTAMP => current date - 0 days)) jrn ;
```

ENTRY_TIMESTAMP	USER_NAME	JOB_NAME	PROGRAM_NAME	REMOTE_ADDRESS	ENTRY_TYPE	ENTRY_TYPE_DETAIL	OPERATION_TYPE	LIBRARY_NAME	FILE_NAME	MASK_NAME	COLUMN_NAME	PERMISSION_NAME	ENABLED
2026-07-03 08:17:21...	GM	QZDASOINIT	QZDASOINIT	10.24...	P	Row permission	DROP	ERPFILE	GLFCLIEN	-	-	COMPANY_PROTECTION	-
2026-07-03 08:18:09...	GM	QZDASOINIT	QZDASOINIT	10.24...	T	Table	ALTER	ERPFILE	GLFCLIEN	-	-	-	-
2026-07-03 08:18:09...	GM	QZDASOINIT	QZDASOINIT	10.24...	P	Row permission	DROP	ERPFILE	GLFCLIEN	-	-	QIBM_DEFAULT_GLFCLIEN_ERPFILE	-
2026-07-03 08:18:24...	GM	QZDASOINIT	QZDASOINIT	10.24...	T	Table	ALTER	ERPFILE	GLFCLIEN	-	-	-	-
2026-07-03 08:18:39...	GM	QZDASOINIT	QZDASOINIT	10.24...	T	Table	ALTER	ERPFILE	GLFCLIEN	-	-	-	-
2026-07-03 08:23:52...	GM	QZDASOINIT	QZDASOINIT	10.24...	P	Row permission	CREATE	ERPFILE	GLFCLIEN	-	-	COMPANY_PROTECTION	YES
2026-07-03 08:23:54...	GM	QZDASOINIT	QZDASOINIT	10.24...	T	Table	ALTER	ERPFILE	GLFCLIEN	-	-	-	-
2026-07-03 08:23:54...	GM	QZDASOINIT	QZDASOINIT	10.24...	P	Row permission	CREATE	ERPFILE	GLFCLIEN	-	-	QIBM_DEFAULT_GLFCLIEN_ERPFILE	YES
2026-07-03 08:24:02...	GM	QZDASOINIT	QZDASOINIT	10.24...	M	Column mask	DROP	ERPFILE	GLFCLIEN	TAXID_MASKED	-	-	-
2026-07-03 08:24:02...	GM	QZDASOINIT	QZDASOINIT	10.24...	M	Column mask	CREATE	ERPFILE	GLFCLIEN	TAXID_MASKED	CLITAXID	-	YES
2026-07-03 08:24:05...	GM	QZDASOINIT	QZDASOINIT	10.24...	T	Table	ALTER	ERPFILE	GLFCLIEN	-	-	-	-

```
-- monitor changes in RCAC (Db2 journal)
select entry_timestamp, journal_entry_type type, jrn.current_user, job_name, program_name, remote_address,
interpret(SUBSTR(entry_data ,1, 20) as char(20)) qualified file
from table( qsys2.Display_Journal ('ERPFILE','ERPJRN', Journal Codes => 'D', STARTING_TIMESTAMP => current date - 0 days)) jrn
where journal_entry_type in ('P1', 'P2', 'P3', 'M1', 'M2', 'M3', 'CG') ;
```

Journal Cod	Entry type	Description
D	CG	Change file
D	M1	Create Mask
D	M2	Drop Mask
D	M3	Alter Mask
D	P1	Create Permission
D	P2	Drop Permission
D	P3	Alter Permission

ENTRY_TIMESTAMP	TYPE	CURRENT_USER	JOB_NAME	PROGRAM_NAME	REMOTE_ADDRESS	QUALIFIED_FILE
2026-07-03 08:17:21...	P2	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:18:09...	P2	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:18:09...	CG	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:18:24...	CG	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:18:39...	CG	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:23:52...	P1	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:23:54...	P1	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:23:54...	CG	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:24:02...	M2	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:24:02...	M1	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE
2026-07-03 08:24:05...	CG	GM ...	QZDASOINIT	QZDASOINIT	10.24...	GLFCLIEN ERPFILE

RCAC – Database Journal!



```
-- who can view journal entries on a file protected by RCAC?
set session authorization gmrccac001 ;

select entry_timestamp, journal_entry_type , jrn.current_user, job_name, program_name,
interpret(SUBSTR(entry_data ,1, 200) as char(200)) record_changed
from table( qsys2.Display_Journal ('ERPFILE','ERPJRN', Journal_Codes => 'R', STARTING_RECEIVER_NAME => '*CURCHAIN',
STARTING_TIMESTAMP => current date - 0 days)) jrn where journal_entry_type in ('UP', 'PT', 'DL');
```

ENTRY_TIMESTAMP	JOURNAL_ENTRY_TYPE	CURRENT_USER	JOB_NAME	PROGRAM_NAME	RECORD_CHANGED
2026-07-03 14:04:05...	UP	GMRCACALL	QPADEV0003	GLFPGM001E	00149150000000000002Mr Jaime GONZALES
2026-07-03 14:04:16...	UP	GMRCACALL	QPADEV0003	QDZTD00001	NOT AUTHORIZED
2026-07-03 14:05:51...	UP	GMRCACALL	QPADEV0003	QCMD	00159150000000000001John Ford (Jr.)
2026-07-03 14:05:51...	UP	GMRCACALL	QPADEV0003	QCMD	NOT AUTHORIZED
2026-07-03 14:05:59...	UP	GMRCACALL	QPADEV0003	QCMD	00159150000000000001John Ford (Jr.)
2026-07-03 14:05:59...	UP	GMRCACALL	QPADEV0003	QCMD	NOT AUTHORIZED

~~DSPJRN JRN(ERPFILE/ERPJRN) RCVRNG(*CURCHAIN) ENTYP(UP)~~

Display Journal Entry

Object : GLFCLIE

Member : GLFCLIE

Incomplete data . . . : No

Sequence : 80

Code : R - Operation on specific record

Type : UP - Update, after-image

Library : ERPFILE

Minimized entry data : *NONE

Entry specific data

Column	*...+...1...+...2...+...3...+...4...+...5
00001	'003101245894317825 JOHN Watson
00051	'BBBBBZVAT-118
00101	'Downing Street
00151	'SOUTH AFRICA
00201	'6 1234567890
00251	'YEUR'

Display_journal (SQL Service)

masks on columns: not applied

Permissions on rows: applied

DSPJRN (Command)

masks on columns: not applied

Permissions on rows: not applied

Important: RCAC is applied to the table or physical file access. It is not applied to the journal receiver access. Any and all database transactions are represented in the journal regardless of RCAC row permissions and column masks. This makes it essential that IBM i security is used to ensure that only authorized personnel have access to the journaled data.



RCAC – Triggers & Functions!

FLEXIBILITY

EFFICIENCY



TRIGGERS associated with a file under RCAC control

- “SECURED” Option required (manipulations authorised for users registered in QIBM_DB_SECADM)
- Accesses hidden columns

FUNCTIONS used in RCAC permission

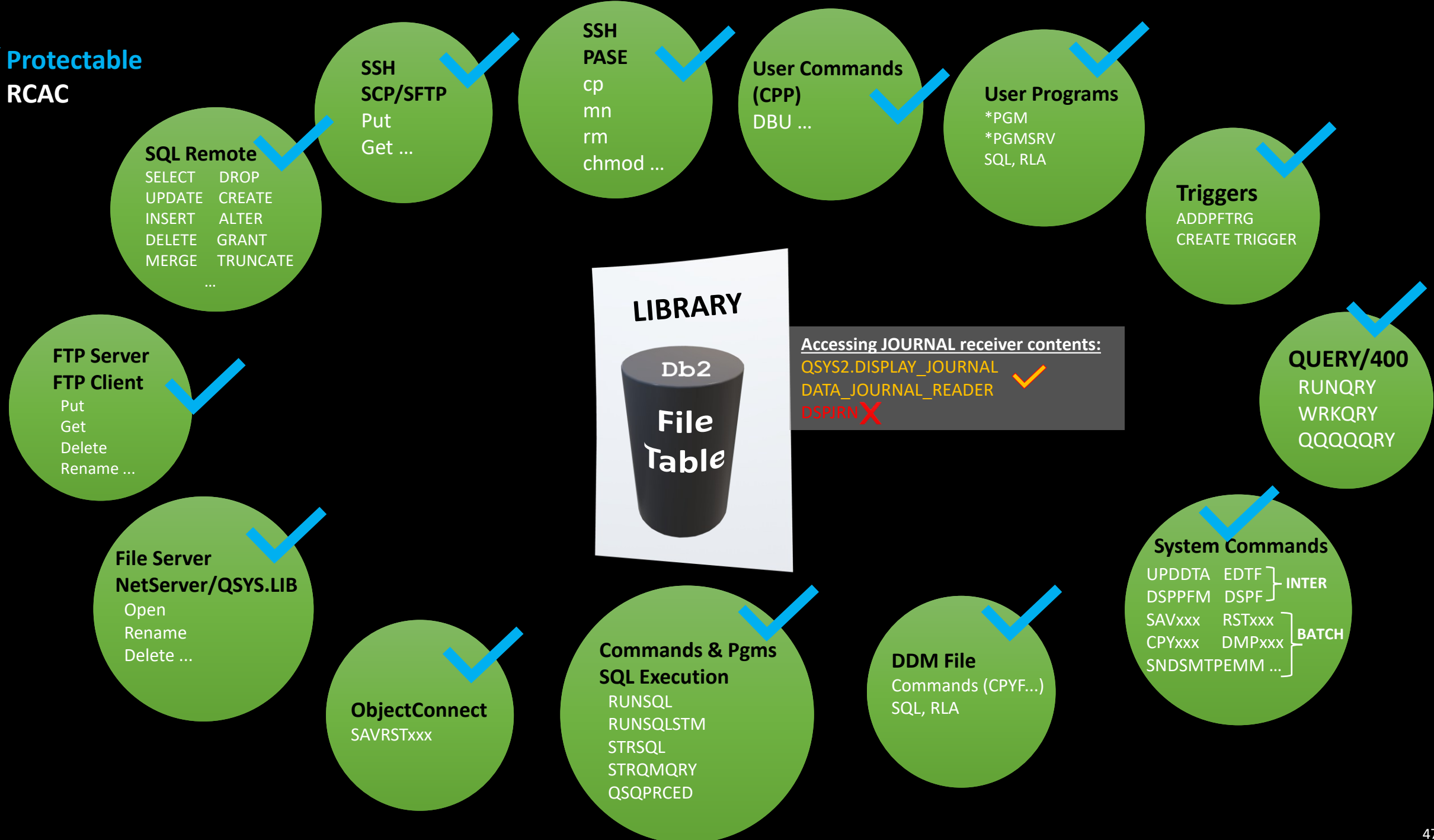
- “SECURED” Option required (manipulations authorised for users registered in QIBM_DB_SECADM)
- “NO EXTERNAL ACTION” Option required (actions outside of Db2 prohibited)
- Accesses all rows and columns

VIGILANCE

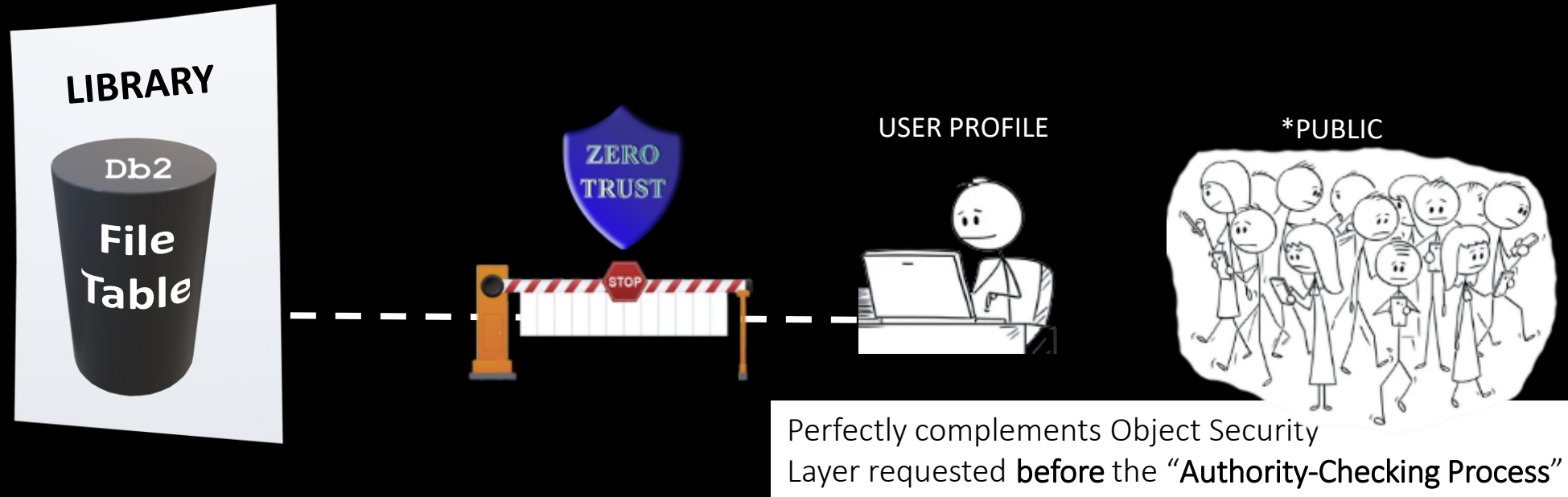
Must be subject to drastic controls (code review, deployment in prod, manipulations, etc.)

```
-- review secured functions & triggers
select * from qsys2.sysroutines where secure = 'Y'
select * from qsys2.systriggers where secure = 'Y'
```

✓ **Protectable**
RCAC



Exit points - Full contextual Access Authorities



Multiple Levels:

Protocol
Job
TimeStamp
IP
Client Register
SQL statement
...

What to do with an Exit Program:

- Reject certain connections and/or transactions
- Log rejected attempts
- Log certain sensitive connections and/or transactions (admin user, critical table, unlisted IP, non-application call stack, etc.)
- Trigger actions (sending to a SIEM, alert, automatic remediation, MFA challenge, interact with a SOAR, etc.)

Most advanced « contextual » Access Control

Exit Points - Full Contextual Access Authorities

Security-Related Exit Point Categories

Those who are:

- attached to authentication (FTP Server, REXEC, ODBC, TCP Logon)
- attached to transactions, commands, functions, ... (FTP client, FTP Server, REXEC, ODBC, NetServer, Remote Commands, DDM)
- attached to commands (before, after)
- attached to Sockets (Low-level communication - IP & Port)
- attached to SQL engine (Query Governor, Query Supervisor)
- more exotic (job_notify, virus scanning, profile, password, data queues, ...)

Multiple Levels:

Protocol
Job
TimeStamp
IP
Client Register
SQL statement
...

New in 7.6

Exit Point Name	Exit Point Format	Exit Program Library	Exit Program
QIBM_QSY_AUTH	AUTH0100	RSLAUD	LOG_AUTH

Most advanced « contextual » Access Control

✓ **Protectable**
Exit Points
Protocols

SQL Remote ✓

SELECT	DROP
UPDATE	CREATE
INSERT	ALTER
DELETE	GRANT
MERGE	TRUNCATE
...	

SSH ✗

SCP/SFTP

Put
Get ...

SSH ✗

PASE

cp
mn
rm
chmod ...

User Commands (CPP) ✓

DBU ...

User Programs ✗

*PGM
*PGMSRV
SQL, RLA

Triggers ✗

ADDPFTRG
CREATE TRIGGER

QUERY/400 ✗

RUNQRY
WRKQRY
QQQQQRY

System Commands ✓

UPDDTA	EDTF	} INTER
DSPPFM	DSPF	
SAVxxx	RSTxxx	} BATCH
CPYxxx	DMPxxx	
SNDSMTPEMM ...		

DDM File ✓

Commands (CPYF...)
SQL, RLA

Commands & Pgms ✓

SQL Execution

RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

ObjectConnect ✓

SAVRSTxxx

File Server ✓

NetServer/QSYS.LIB

Open
Rename
Delete ...

FTP Server ✓

FTP Client

Put
Get
Delete
Rename ...



Exit points - Full Contextual Access Authorities

Security-Related Exit Point Categories

Those who are:

Multiple Levels:

Protocol
Job
TimeStamp
IP
Client Register
SQL statement
...

- attached to authentication (FTP Server, REXEC, ODBC, TCP Logon)
- attached to transactions, commands, functions, ... (FTP client, FTP Server, REXEC, ODBC, NetServer, Remote Commands, DDM)
- attached to commands (before, after)
- attached to Sockets (Low-level communication - IP & Port)
- attached to SQL engine SQL (Query Governor, Query Supervisor)
- more exotic (job_notify, virus scanning, profile, password, data queues, ...)
- attached to Db2 file openings (audit values *CHANGE, *ALL) & IFS stmf (attributes *CRTRUNEXIT & *RUNEXIT)

Most advanced « contextual » Access Control

✓ **Protectable
Exit Points
Open Files**

SQL Remote
SELECT DROP
UPDATE CREATE
INSERT ALTER
DELETE GRANT
MERGE TRUNCATE
...

**SSH
SCP/SFTP**
Put
Get ...

**SSH
PASE**
cp
mn
rm
chmod ...

**User Commands
(CPP)**
DBU ...

User Programs
*PGM
*PGMSRV
SQL, RLA

Triggers
ADDPFTRG
CREATE TRIGGER

QUERY/400
RUNQRY
WRKQRY
QQQQQRY

System Commands
UPDDTA EDTF } INTER
DSPPFM DSPF }
SAVxxx RSTxxx } BATCH
CPYxxx DMPxxx }
SNDSMTPEMM ... }

DDM File
Commands (CPYF...)
SQL, RLA

**Commands & Pgms
SQL Execution**
RUNSQL
RUNSQLSTM
STRSQL
STRQMQR
QSQPRCED

ObjectConnect
SAVRSTxxx

**File Server
NetServer/QSYS.LIB**
Open
Rename
Delete ...

**FTP Server
FTP Client**
Put
Get
Delete
Rename ...



Exit Points

Full Contextual Access Authorities

BEFORE

« A call for collective action on cyber defense »

« Food » for thought:

- Numerous vulnerabilities discovered since 2022 that have always existed in our favorite OS...
- Software vendors are targeted by hackers and/or had critical breaches (Software Supply Chain Attacks). What about our habits with trusted third parties?!
- SQL and open-source capabilities are becoming more numerous and complex.
- A user with no authorities and limited capabilities still has significant system discovery possibilities
- Defining read access? A SELECT associated with an ACS download is not equivalent to the same SELECT in a Java application... (data export for one)



Exit Points

Full Contextual Access Authorities

AFTER

« A call for collective action on cyber defense »

« Food » for thought:

- Stay current (as soon as you can)
- Protect heavily authentications
- Reject discovery behaviors
- Implement zero trust as much as possible
- Keep your audit trail clean, long and correlated
- Threat can come from everywhere, including the inside....



Other measures increasing Db2 Data Protection

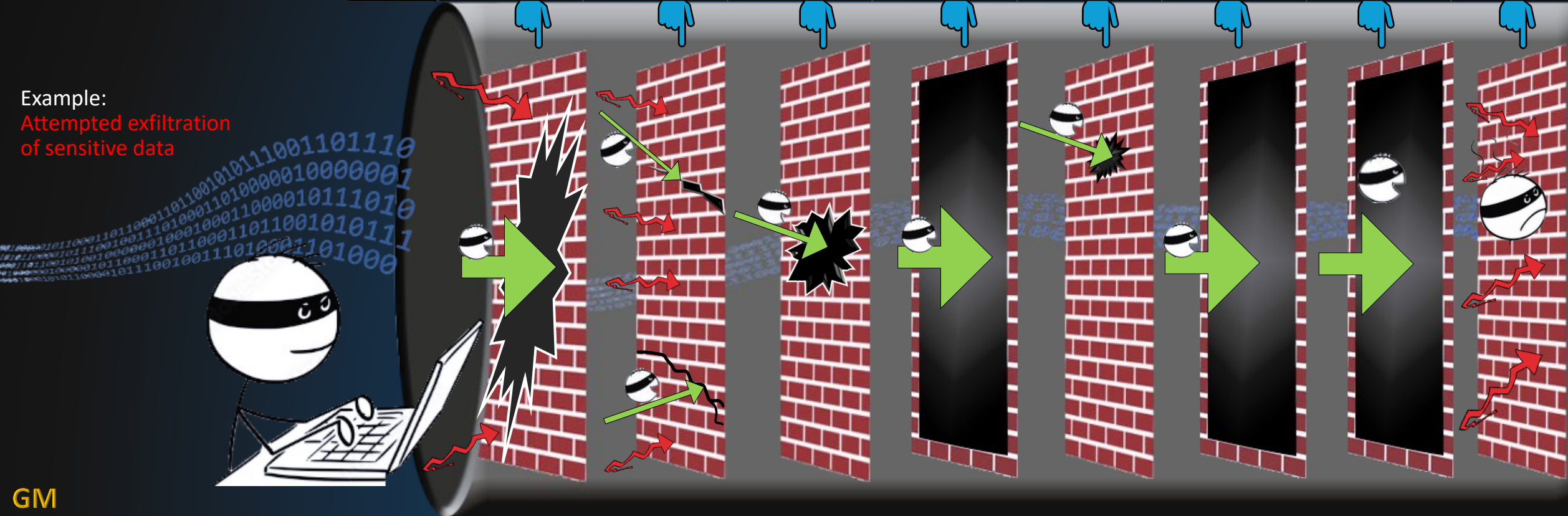
- Data encryption using Field procedures
- Data anonymization
- Combinations of various measures to strengthen access to sensitive data and/or the use of direct interfaces (MFA, privilege escalation, ticketing system integrated into exit points)



THE EFFECTIVENESS OF SECURITY THROUGH INDEPENDENT LAYERS

LEVEL	Workstation	Workstation	IBM i Partition	IBM i Partition	IBM i Partition	IBM i Partition	IBM i Partition	IBM i Partition
Protective measure	Prohibited Admin Authorities	Packaged Software deployment according to the Role	Public authorities *EXCLUDE on data *CHANGE authorities via the Application Group	Function usage	Exit Point SQL & FTP (connection & détail)	Exit Point SQL (Client Register)	Exit Point on ultra-sensitive tables	RCAC
Status of the measure	In place	In place	In place	Not activated because redundant with exit points	In place in blocking mode	Not activated because not supported by the exit program	Not activated because exit program is not available or not performing	In place
Method(s) of violation	Phishing or dishonest user	FTP & DBeaver accessible	SQL, FTP	SQL, FTP	FTP bloqued, SQL accessible	SQL accessible	SQL accessible	SQL accessible , but no data returned
Explanations	Takeover of the workstation via a website proxy and installation of an executable; Theft of ID and password for connecting to the servers	This workstation belonged to a user with a different role; the station was not cleaned up properly before redeployment	The profile used being a member of the application group, its access to the data is legitimate	Function usage: Not activated	Members of the application group are legitimate to use a Java application manipulating IBM i data including the targeted sensitive table	No control in place over the client used	No additional control over the targeted sensitive table	SQL permission on sensitive files with analysis of working environment criteria via an SQL function: Reading a sensitive table outside the application gives an empty result

Example:
Attempted exfiltration of sensitive data



THANK YOU
FOR
YOUR ATTENTION!



Steve Bradshaw
steve.bradshaw@rowtonit.com



Guy Marmorat
gmarmorat@resiliane.com