Security in the Datacom™ World

Robert Florian, Principal Software Engineer

robert.florian@Broadcom.com

BROADCOM

Disclaimer

- Certain information in this presentation may outline CA's general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. This presentation is based on current information and resource allocations as of October 7, 2021 and is **subject to change or withdrawal by CA at any time without notice**. **The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion**.
- Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available to new licensees in the form of a regularly scheduled major product release. Such release may be made available to licensees of the product who are active subscribers to CA maintenance and support, on a when and if-available basis. The information in this presentation is not deemed to be incorporated into any contract.
-
- **THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY**. Broadcom assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. **In n**o event will Broadcom be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if Broadcom is expressly advised in advance of the possibility of such damages.

Abstract

- In this world of hyper sensitivity to security, understanding how to implement Datacom external facility security is critical. This session will provide the user with a basic setup to get started with external security.

BROADCOM

Robert Florian
Principal Software Engineer
Plano, Texas, USA

- Robert has been part of the Datacom/DB Engine Development team since ADR days in 1987
- He became involved with External Security as it replaced internal security over 30 years ago
- He is a SME in Index Processing and DB internals
- He holds Multiple Patents and loves tinkering with code
- Robert is a history buff who has spent many years restoring a large house built in 1908

Agenda

- Show the available Datacom components used to set a secured environment
- Highlight two use cases
- Not get too bogged down in details
  - But have the details in this presentation for your reference
- For those who have not yet set up Security a cookbook
- For experienced users interesting things you may not have considered
- Questions and comments welcome in the chat window

External Security Risks and Rewards

- Without External Security
  - Anyone who can get to a MUF can do anything they want!
    - Accidently or maliciously corrupt data
    - Steal or copy data
    - Accidently or maliciously harm the MUF
  - A few non external security tools exist inside MUF for protection
    - Such as Open Exit
    - Simplify Mode
- Startup message

```
DB00270W - ACCESS TO DATACOM TABLES NOT PROTECTED BY EXTERNAL SECURITY
DB00201I - MULTI-USER ENABLED, CXX=yourcxx ...
```

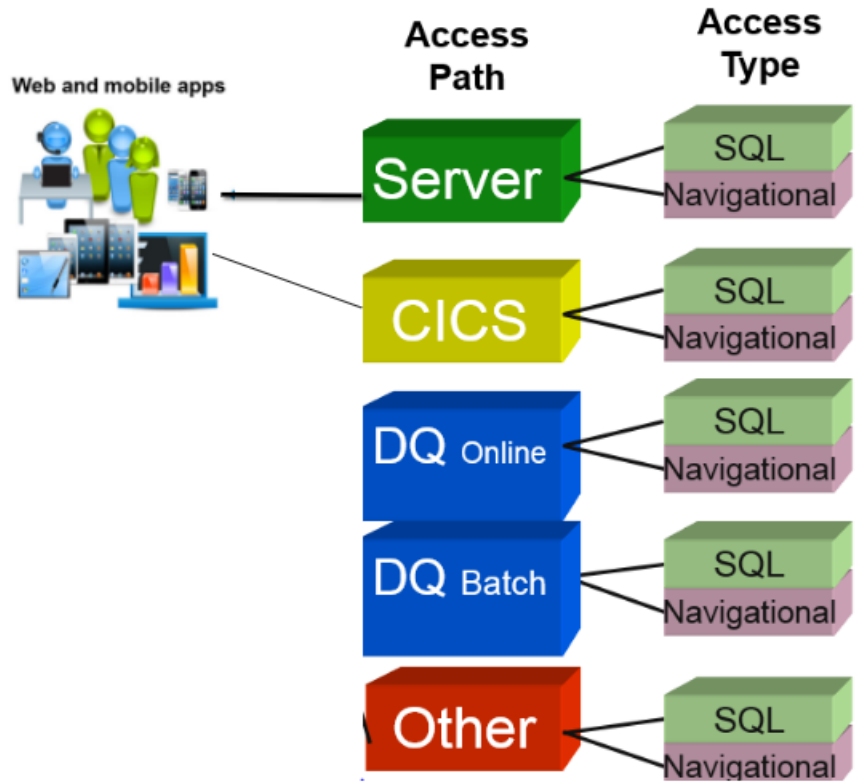
Steps for Implementing External Security

- Decide how the data needs to be secured
  - Data access Paths
- Analyze roles and departments for logical groupings
- Set up Table Classes and Rules
  - Which are checked when the table is accessed
- Set up Function Security using DTUTIL class
- Decide if other External Security Protocols needed
  - Such as XCF, Plan Security, …
  - Can be done before or after turning on
- Set up DTSYSTEM resources to turn it on

Table Access Path Security

- MUF recognizes different ways data is accessed
- Which can have independent security rules
- Including some (or all) paths which can be set up with NO external security

The ten recognized paths



BROADCOM

Two sample Use Cases Securing Table Access

- An AD CA 7 MUF
  - A single table class to protect CA 7 tables from being accessed or changed outside CA 7
- A full DB MUF
  - Datacom Server access
    - Use a different rule set and table class
  - CICS no table security
    - Because I have implemented CICS transaction security outside Datacom
  - All other table access (including Dataquery™ CICS)
    - Use a separate or different class than Server
- For these samples I will be using Top Secret
  - And assuming it has been set up as a closed system

Table Classes used by Datacom

- 10 Available classes just as there are 10 paths
  - DCTABLE DFTABLE DGTABLE DHTABLE DPTABLE
  - DQTABLE DRTABLE DSTABLE DTTABLE DXTABLE
- You pick the classes you will use
- Table Access Levels used by Datacom
  - ADD DELETE READ UPDATE
  - Autonomous in ACF2 and Top Secret
  - Hierachical in RACF
- Resource name's format
  - cxxname.DB0nnnn.ttt
    - nnnn 4 character DBID
    - ttt 3 character Datacom table name

CA 7 AD MUF

Using DCTABLE Resource Class

- Example CXXname is CA7CXX
- Allow access to CA 7 support tables for User Id associated with the CA 7 started task
- TSS PER(CA7STC) DCTABLE(CA7CXX.DB00770.) ACCESS(ALL)
- Allow read access to Dynamic System Tables containing metadata which CA 7 uses
- TSS PER(CA7STC) DCTABLE(CA7CXX.DB01000.) ACCESS(READ)
- Allow a SYSPROG maintenance access to CA 7 support tables
- TSS PER(CA7SPG) DCTABLE(CA7CXX.DB00770.) ACCESS(ALL)
- Allow user read access to CA 7 support tables
- TSS PER(userid1) DCTABLE(CA7CXX.DB00770.) ACCESS(READ)

Datacom Component Rules

DD, SQL, etc.

- For both AD and full DB
- A Datacom component has implicit rights to its support tables
  - Hence an SQL request has rights to read and update the DDD database as needed to run SQL
  - An SQL or Datadictionary catalog request (Create, etc.) has rights to update the DD Database
- Meaning you do not have to give table rights to these support databases
  - So they are still protected from corruption via direct user requests
- Datacom Server has rights to what it needs, including the Dynamic System Tables
- AD products in general *do not* piggyback on this for their own supports tables
  - Allows AD products can flexibly change their code in an agile way

Full DB System Sample

- I will use DTTABLE to secure Datacom Server table access
- And DCTABLE to secure non CICS non SERVER access
- Rule format the same, just a different CXXname
  - For my Defaults:
  - *Close down Server:* TSS PER(ALL) DTTABLE(TESTCXX.) ACCESS(NONE)
  - Open up others: TSS PER(ALL) DCTABLE(TESTCXX.) ACCESS(ALL)
- Then through analysis add special rules for some DBIDs and perhaps tables within DBIDs
  - TESTCXX.DB0nnnn.ttt
  - In DCTABLE for Server and DTTABLE non Server non CICS
  - With Access Levels of NONE, READ, ADD, UPDATE, DELETE, or ALL
    - TSS PER(usera) DTTABLE(TESTCXX.DB00111.ORD) ACCESS(READ)

Function Security the DTUTIL Resource Class

- Most function security is defined in the DTUTIL resource class
- For DBUTLTY
  - Resource name is cxxname.DBUTLTY.function.subfunction
  - DBUTLTY secured if any non SQL (i.e. RAAT) path is secured
  - If it is, MUF must be enabled when DBUTLTY runs
    - Security layer is in MUF
    - Similar to Simplify mode
- Other components also use DTUTIL for functional security such as
  - Datadictionary™
    - cxxname.DD….
  - SQL
    - cxxname.SQ…

DBUTLTY Security, DTUTIL

- Need rights to run the function
  - And if a table is involved, appropriate rights for each affected tables
- DBUTLTY functions broad categories
  - Reporting with no data exposure
    - Likely needed by app developers and DBAs
    - Example TSS PER(appuser) DTUTIL(CXXTEST.DBUTLTY.REPORT.CXX)
    - With underlying table DISPLAY rights
    - Example in TSS PER(appuser) DTUTIL(CXXTEST.DB00001.PAY,DISPLAY)
  - Note from above table rights of
    - BACKUP, DISPLAY, CATALOG, LOAD and OPR are defined in DTUTIL
  - DBA functions which could affect the health of the MUF and are needed by DBAs and operators
    - Examples

      TSS PER(opuser) DTUTIL(CXXTEST.DBUTLTY.COMM.CLOSE)
      TSS PER(opuser) DTUTIL(CXXTEST.DBUTLTY.DEFRAG)
      Neither of which has table rights

DBUTLTY Data Exposure

- But some DBUTLTYs create data exposure and should be treated with care
  - Example 1 EXTRACT which creates a flat file with table's data rows
    - TSS PER(userx) DTUTIL(CXXTEST.DBUTLTY.EXTRACT)
    - With appropriate table READ privileges in the RAAT table class
  - Example 2 BACKUP data which creates a Datacom backup file
    - Which can be compressed, etc. so maybe less exposure
    - TSS PER(userx) DTUTIL(CXXTEST.BACKUP.DATA)
    - TSS PER(userx) DTUTIL(CXXTEST.DB00001.PAY,BACKUP)
- Some affect the contents of data and index areas and system areas
  - Examples INIT, LOAD
- You could use special security user ids for running defined or scheduled DBUTLTYs
- Likely start with relatively open or closes system and work in special cases

Cataloging SQL Create Table

- SQL Create Table may be needed at either AD or DB sites
  - In DTUTIL
  - cxxname.DB0nnnn.999.CATALOG
  - Example allow SQL Create into the CA 7 database
    - TSS PER(userid) DTUTIL(CA7CXX.DB00770.999.CATALOG)
  - No Dictionary access needed
    - Since being done on SQL's behalf

Console Like Commands an Important Aside

- Since these can be issued in a host of ways be cognizant of who can issue them and how
- Example EOJ a MUF
  - Through actual console
    - z/OS console privileges
    - No additional Datacom external security
  - DBUTLTY COMM OPTION=EOJ (older protocol)
    - DTUTIL resource cxxname.DBUTLTY.COMM.EOJ
  - Through DBUTLTY “console-like” facility
    - DBUTLTY COMM OPTION=CONSOLE,OPTION2='EOJ ‘
    - DTUTIL resource cxxname.DBUTLTY.COMM.CONSOLE
  - Through SQL Insert into the Dynamic System Tables
    - Table right to DnTABLE (SQL other path)
    - cxxname.DB01000.SQX ACCESS(ADD)

LEVEL concept and DTSYSTEM

- Datacom asks a pair or questions to prove it is not just reacting to defaults
- A Level is an architectural level
  - We have been at level 5 for awhile. If just setting up, choose LEVL05
- Must have access denied in DBSYSTEM resource class to
  - ACTIVATE.LEVEL05.FAIL
- And access allowed to
  - ACTIVATE.LEVEL05.PASS
- For USERID associated with MUF job or started task
- Note no CXXname as part of this resource name!
- Set up levels after at least some minimal Datacom rule set has been defined

Turn on Security with Access Denied in DTSYSTEM

- Counterintuitive: access denied = security on
- Resource name is cxxname.DBccppp
  - cc is the resource class
    - For instance DC is DCTABLE
  - ppp is the access path,
- Since for CA 7 sample we want all 10 to use the same class
  - Sample rule set
    - TSS PER(ALL) DTSYSTEM(CA7CXX.DBDC*) ACCESS(NONE)
    - TSS PER(ALL) DTSYSTEM(CA7CXX.*) ACCESS(ALL)
  - This forces MUF to use DCTABLE class for all path

Turn on Security with Access Denied Full DB Sample

- In first line below
  - DT is DTTABLE class
  - SSR is access path SQL Server (paths codes listed in the Security Doc)
- TSS PER(ALL) DTSYSTEM(DBCXX.DBDTSSR) ACCESS(NONE)
- TSS PER(ALL) DTSYSTEM(DBCXX.DBDTRSR) ACCESS(NONE)
- TSS PER(ALL) DTSYSTEM(DBCXX.DBNOSCI) ACCESS(NONE)
- TSS PER(ALL) DTSYSTEM(DBCXX.DBNORCI) ACCESS(NONE)
  - NO means no security for this path (SCI SQL CICS, RCI RAAT CICS)
- TSS PER(ALL) DTSYSTEM(DBCXX.DBDC*) ACCESS(NONE)
  - All others paths use DCTABLE
- TSS PER(ALL) DTSYSTEM(DBCCXX.*) ACCESS(ALL)

Enabling External Security

- Turn on the level
  - TSS PER(ALL) DTSYSTEM(ACTIVATE.LEVEL05.FAIL) ACCESS(NONE)
  - TSS PER(CA7STC) DTSYSTEM(ACTIVATE.LEVEL05.PASS) ACCESS(ALL)
- The SECURITY MUF Startup Option
  - If coded and no choice, must match the DTSYSTETM rule set
    - For full DB
    - SECURITY DBNOSCI,DBNORCI,DBDTSSR,DBDTRSR
    - SECURITY DBDCRAQ,DBDCSQQ,DBDCSCQ,DBDCRCQ,DBDCRAT,DBDCSQL
  - But is optional if no choice

For Test Systems can allow a choice

- TSS PER(ALL) DTSYSTEM(TESTCXX.DB*) ACCESS(NONE)
  - All path class combinations access denied
  - So must code the SECURITY Startup Option
  - And what you code is what will be used (including no)
  - SECURITY DBwwSCI,DBwwRCI,DBwwSSR,DBwwRSR
  - SECURITY DBwwRAQ,DBwwSQQ,DBwwSCQ,DBwwRCQ,DBwwRAT,DBwwSQL
  - Where ww is whatever two character class name of the 10 you choose
- I use this on my personal test system to turn on and off security at will for testing purposes
  - Since I am not a security administrator
  - And it is bothersome to constantly ask for a rule change
- Note other values exist for cxxname. In DTSYSTEM such as cxxname.DD for Datadictionary functional security, hence my rule is cxxname.DB* not cxxname.*

Some Messages

- From my Full DB Example at MUF Startup

```
DB00231I - EXTERNAL SECURITY LEVEL 05 ACTIVE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON SQL OTHER DQ WITH DCTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON RAT OTHER DQ WITH DCTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON SQL CICS DQ WITH DCTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON RAT CICS DQ WITH DCTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON SQL SERVER WITH DTTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON RAT SERVER WITH DTTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON SQL OTHER WITH DCTABLE
DB00220I - EXTERNAL SECURITY ACTIVE FOR TESTCXX ON RAT OTHER WITH DCTABLE
```

Cataloging using DDUPDATE with BTG

- Rights to catalog tables and DBIDs needed at both AD or DB sites
- Datadictionary function security controlled through an independent DTSYSTEM resource
  - Simple name cxxname.DD
    - With Access denied to turn it on
    - DB00220I - EXTERNAL SECURITY ACTIVE FOR PRODCXX ON DATADICTIONARY
- DTADMIN class
  - Access allowed for cxxname.DD
    - A Dictionary Administrator
  - (DTADMIN also has a DB node for things like SQL DROP table)
- Many needed rights in DTUTIL
  - Sample rule TSS PER(CA7DBA) DTTUTIL(CA7CXX.DB00002.) ACCESS(ALL)

Datacom Server Considerations

- LOGON=YES (DEFAULT)
  - SVDBSPR must reside in an APF authorized library
  - USERID is defined in the external security package
  - USERID is passed to the associated Datacom MUF to determine data access privileges
  - If LOGON=NO, ACEE is not accessed and no user info is passed to MUF!
- Datacom Server does NOT perform FACILITY checking. Rather, it simply validates the userid/password with external security and passes it to the MUF where additional access privileges are checked
- CONEXIT and SECEXIT can be used
  - CONEXIT is called BEFORE the call to the external security interface.
  - SECEXIT is called AFTER the call to external security

Useful SQL External Security Tools

- Up to now we have said security is at the table access level
- And my sample suggested since a CICS transaction has a predefined set of resources transaction security may be good enough
- SQL also has Plan and View External Security
  - Somewhat analogous to the CICS transaction security
- With Plan Security
  - You can inherited the binders table rights
    - Via check who, binder or executor and
    - Check when, bind or execute
- A view is a collection or subset of tables, so checking using View instead of table means a user only see the data the view represents
  - Chosen using the VIEWSEC Preprocessor plan option, see doc

Security Product Differences

- Resource Class Names are different
  - ACF2 Class Names are three characters
    - DTSYSTEM is DTS
    - DTUTIL is DTU
    - Table classes are also first three characters
  - RACF Class Names 3rd character is an “@”
    - DTSYSTEM is DT@YSTEM
    - DTUTIL is DT@TIL
    - Table class follow this pattern
  - With Top Secret MUF must be defined as a Facility

“CA Database User Experience: Calling all Customers!”

- Customer feedback and direction is invaluable to our overall strategic direction. We are interested in hearing your voice and providing us with insight based on real-life experiences and requirements.
- Please join us for an interactive user experience (UX) session where we look to you – our customers – to assist in shaping the future of our products.
- Be sure to join session DM19-L Thursday, October 7 @ 12:00 – 12:50pm ET

Request a Design Thinking Workshop

mainframe.broadcom.com/resources/design-thinking-workshop

- Influence our product direction
- Collaborate on solutions to your problems
- Build a strategic partnership

- We listen
- We take action
- We come back to you with results

Thank You

Now, please join us for a live Question and Answer discussion

- **Click the meeting link at the bottom of the Session Description to join us.**
- This is your opportunity to connect with the presenter(s) and your peers, ask questions, and share information related to this topic.