CICS Essentials

CICS Best Practices

NewEra Software

Special Interest Article

Table of Contents

CICS and External Security Manager - Best Practices

While CICS has its own internal security built in it does not provide an acceptable level, not covering many internal policies and legal compliance requirements for storing data.

At its most basic level users should have USERIDs defined to the security package. And the user should have to sign into CICS prior to access to any transactions ensuring that the correct level of access is granted to resources and transactions required for their role.

The use of external security has to be enabled within the CICS SIT and care should be taken that required external security SIT options are also enabled, and that any associated RACF resources are defined.

Two default resource class names are provided with RACF to define protection for transactions processing, the member class TCICSTRN and the group class GCICSTRN. Individual transactions can be added to these groups to define levels of access by users and groups.

Your external security manager should be used to ensure that the use of CICS commands is sufficiently protected, preventing their potential misuse as this could have an adverse effect on your installation. It is important that use of these commands is logged.

Security settings used to grant access to CICS resources and the use of CICS commands should be controlled by your security team.

It should be remembered that not all CICS regions need the same levels of security and that security policies can also be dependent on whether the region in question is Production, Development or Test.

CICS and VTAM Best Practices

CICS and VTAM Best Practices – Changes to VTAM config settings can cause errors in the network. This in turn can lead to catastrophic connectivity issues for CICS - both to user signon and ISC.

A unique VTAM APPLID is required for each CICS region across your plex and this is defined using an APPL statement in SYS1.VTAMLST. Users must be granted access to this resource in RACF to allow them to logon, and the CICS region must be given READ access to the VTAM APPLID defined in VTAMAPPL class. The profile must match the APPLID defined in SIT and include SECPREFIX if active for the region.

VTAM terminal defs can be defined in the CSD or by using the CICS autoinstall facility. TYPETERM and TERMINAL definitions must exactly match the VTAM resource defs.

A terminal definition can refer to any device that connects to CICS, such as VDU’s, printers and System Consoles.

A time limit for how long VTAM terminals used by CICS will remain connected should be defined. This is because a hung terminal can prevent clean shutdown of the CICS region.

Great care should be taken over VTAM resources defined to CICS. Older, unnecessary definitions can remain as newer, replacements are added or when new CICS regions are created as clones, leaving potential back doors to the CICS region thus exposing a security risk. Any obsolete entries should be removed, and backups of all definitions should be taken on a regular basis.

The CICS System Definition Dataset – Best Practices

The CICS System Definition Dataset (DFHCSD) is one of the most important files in your CICS installation. It contains all of the resource definitions to be loaded by CICS during cold start. It also houses dynamically changed resource defs (by using CEDA or DFHCSDUP).

The recommended way of adding new resource defs to the CICS CSD is via Resource Definition Online (RDO). The 3 RDO transactions are:

CEDA – allows changes to the CSD in active CICS regions enabling 'on the fly' changes to resource defs

CEDB – gives READ to the CSD and modify access to all resource defs except INSTALL

CEDC – allows read only command access to the CSD

Changes can also be made using the batch CSD update utility DFHCSDUP.

As the CSD is accessed at a regions cold start, changes made to these resource defs could pass unnoticed until the CICS region is recycled. This is why it is essential that access to this dataset is tightly controlled (and robust backup & recovery strategies are in place).

A CSD dataset can be shared between multiple CICS regions so care should be exercised before allowing changes to ensure that they do not have an adverse affect on any region that uses it.

The resource defs that are to be loaded are defined in the GRPLIST SIT parameter. The addition of resource groups using the CEDA or DFHCSDUP LOCK command should be controlled using CMDSEC=YES in SIT with general resource profiles in RACF classes CCICSCMD & VCICSCMD.

CICS Development Environment - Best Practices

The driving service ethos for a Development CICS Environment is Flexibility. The CICS infrastructure needs to be biased towards what the developers need to create new code and functionality but without creating a “free for all” culture. Developers should have access to clear guidelines on what they can, can’t, should or shouldn’t do.

The security requirements for the environment may be less restrictive but the integrity of the entire CICS service must not be compromised. Enhanced access always generates enhanced risk and it is important to remember that to its users the Development CICS service IS a “production service”, the better way is to streamline the change process to minimize red tape delays whilst leaving overall configuration control with the sysprogs who understand it.

The Development CICS configuration should replicate that of Production in all areas, as much as is possible, to ensure that the underlying CICS configuration envelopes match. Ensure that both the addition of new functionality and the removal of obsolete functionality are correctly replicated between the two environments.

Lastly ensure adequate backups are taken of both the CICS configuration and the application data. A Development CICS service that won’t start is of no use to anybody.

CICS Jobs and JCL – Best Practices

Job Control Language (JCL) defines how CICS will run in your System. Special care should be taken when setting up or changing the JCL for your CICS regions as parameters and settings have a major affect on the security and functionality of the region.

JCL to start a CICS region can be stored in any location meaning that when you submit a task for execution you should ensure that you are submitting the correct task, and that a region with the same STC or JOB name is not already running. The best way to view the right JCL for a specific CICS region that is already up is to look in JESJCL.

The CICS start up creates a z/OS address space which has all of the required resources available to it. Remember that any changes to CICS JCL will not come into use until the region in question is recycled. This can lead to sleeper issues where a change can be made quite some time before the effect is seen.

Pre CICS TS v4.1 //STEPLIB and //STEPCAT statements should be checked as both can change the location of datasets used at CICS startup.

The order of //DFHRPL (Relocatable Program Libraries) concatenation must be verified as it defines the location(s) of all the programs that may be executed by a particular CICS region.

The PARM parameter of the EXEC PGM=DFHSIP (as well as SYSIN dataset(s) in the JCL) can be used to control which System Initialisation Table is used at region start. You should ensure that the expected SIT is used as this can affect both the shape and the security of CICS.

CICS Production Environment - Best Practices

The driving service ethos for a Production CICS Environment is Availability. The Production CICS infrastructure must be protected as much as, if not more than, the Production application data. However care must be taken not prevent the staff responsible for the service from being able to do their jobs, particularly activities that require real-time response / resolution.

The choices taken at the Development and Test levels will either help or hinder the control overhead a Production environment generates. Correctly defined processes at these levels should result in far fewer, if any, issues when changes reach the Production environment. Often separate teams are responsible for each CICS level, making clear communication between them a vital part of reducing the complex task of managing the entire CICS infrastructure.

Any Production CICS service should have a tried and tested recovery plan that also caters for loss / recovery of the Development or Test CICS environments.

All changes to the production CICS infrastructure must have an audit trail that not only covers both the ‘what’ was changed and the ‘why’ but also includes checks to ensure that the change matches what was requested.

Establish procedures to ensure that obsolete / redundant definitions are deleted when no longer required.

Region USERID - Best Practices

Region USERID Best Practices - every CICS region runs under a base USERID which determines which RACF (or other ESM) protected resources it is allowed to access. Referred to as the CICS Region USERID, this does not affect the level of access given to individual users.

The CICS Region USERID should never be the same as the HLQ of any of the datasets (or the PROC name) for its associated CICS region. ALTER access will be granted to the Region USERID when processing these datasets if there is a match.

The CICS Region USERID can be associated with the task in 3 ways:

1 in ICHRIN03 – the RACF started procedures table

2 the USER parameter of the STDATA segment in a STARTED class profile

3 the USER parameter on the JOB statement if CICS is started as a job

The TRUSTED and/or PRIVILEGED attributes should never be allocated to the CICS Region USERID.

When the SIT parameter SECPRFX is set to YES then the CICS Region USERID is used as the prefix to RACF resource names.

Each separate CICS region should run with its own unique USERID to ensure separation of any granted authority levels.

If you are using Inter System Communication (ISC) then it's vitally important that unique USERIDs are used. Using common USERIDs in conjunction with ISC can lead to data being passed between regions, or even to unauthorised transactions being started remotely with increased levels of access granted.

The System Initialization Table – Best Practices

The System Initialization Table (SIT) controls the shape and security of your CICS region. Access to and changes within this dataset should be highly controlled with anything higher than READ being granted only in special cases.

It is a complicated file with around 300 parameters available. Some of these parameters enable security on your CICS regions and the correct use of these parameters and their settings should be ensured.

From a security point of view the most important SIT parameter is SEC. It is used to activate the implementation of external security (RACF, ACF2 or TSS). External security won’t be active unless SEC=YES.

A minimum level of security in CICS can be achieved using XTRAN=YES to enable user authorization to transactions checking.

CMDSEC parameter controls sub commands available to the CICS master terminal (CEMT) and the CICS command interpreter (CECI) transactions. Both of these transactions are very powerful including the shut down command for CICS. This is why command security should be implemented.

CICS can perform some program processing prior to security being fully initialised for the region e.g. during startup processing. PLTPIUSR and PLTPISEC parameters in SIT define the USERID to be used and scope for this processing.

Descriptions of all other SIT parameters, together with recommended settings for these can be found in NewEra’s CICS Essentials - Auditing CICS - a Beginners Guide book available from www.newera.com/CICS

CICS Test Environment - Best Practices

The driving service ethos for a Test CICS Environment is Repeatability. The Test CICS infrastructure can consist of multiple discrete services offering parallel or multi-stage testing cycles often combined with some form of baseline process designed to simplify the “costs” of testing code. Test environment users should have access to clear guidelines on how the environment has been configured and what testing procedures should be followed.

The testing cycle should also include the migration processes used for the complete change “package” to help ensure that the change is correctly migrated to production first time.

The security requirements for the environment will be tighter than in Development but may still be less restrictive than for Production.

The Test CICS configuration should replicate that of Production in all areas, as much as is possible, to ensure that the underlying CICS configuration envelopes match. The closer the Test environment replicates Production the better the testing results tend to be. Ensure that both the addition of new functionality and the removal of obsolete functionality are correctly replicated between the two environments.

Common Security - Best Practices

Common Transaction and Command Security - Best Practices : These are the minimum principals which should be applied to securing any CICS system.

Transaction Security - GCICSTRN & TCICSTRN

TCICSTRN is the default RACF class which is used to define member resource profiles for individual CICS transactions. GCICSTRN is the default member grouping class in RACF.

Grouping class profiles should be used in preference to member resource ones. Alternative class names can be used by specifying the suffix (the first character remains either T or G) in the SIT.

Enabling the use of transaction security in SIT is effectively a binary option - it is on or off. There is no way to apply transaction security to just some transactions in a particular CICS region.

Command Security - VCICSCMD & CCICSCMD

CICS command security controls the use of system programming (SP) commands such as CEMT with INQUIRE, CREATE, DISCARD, PERFORM and SET.

Varying levels of RACF authorization are needed to each command depending on the required action. Regardless of this, the user must also have authority to run the CEMT transaction.

In general, RACF READ-level access to a CICS command implies the capability to view but not change the information managed by the command - UPDATE access is required to change CICS definitions via the commands.

ESM System Dataset - Best Practices

There are 3 available External Security Managers (ESM): CA ACF2, CA Top Secret and RACF. This panel discusses only RACF system datasets.

The RACF database contains profile records. The format of the records is controlled using IBM supplied templates. Whenever a change is made to a template your RACF database must be updated using IRRMIN00 with PARM=UPDATE. This must be followed by an IPL to pick up the changes.

You can maintain all of your RACF profiles in one data set or divide your RACF database between multiple data sets. A RACF database can be made up of as many as 90 data sets. This is a performance option.

You should always define a backup RACF database in ICHRDSNT which you can switch to without an IPL in case of problems. RACF maintains this backup automatically based on the requirements you define in ICHRDSNT.

If possible make your RACF database dataset(s) unmovable (PSU). If an active database is moved from where RACF thinks it is results can be extremely unpredictable. If the database is SMS managed you must use PS instead and ensure it is excluded from any DEFRAG type operations.

UACC(NONE) should be used for all RACF database dataset profiles. Many sites allow READ but this can lead to the ability to copy the data. General users do not need *any* direct access to the information.

Backup and recovery processes for the RACF database based on IRRUT200 should be both in place and regularly tested.

You should contact NewEra Technical Support for more info if you are one of the Customers who use CA ACF2 or CA Top Secret.

High Level Security - Best Practices

High Level Security - Best Practices : These represent additional areas of security which can be applied to a CICS region.

Resource security can be applied selectively using resource defs in CSD to control whether to call RACF for access requests.

It is particularly appropriate to use these extra levels of security when you have a complex CICS environment. For example, if you have a single transaction that performs multiple functions each requiring differing security lock down.

ACICSPCT & BCICSPCT - Program Control Table

Controls who can START a CICS *transaction* from within a program.

DCICSDCT & ECICSDCT - Transient Data Queues

FCICSFCT & HCICSFCT - File Control Table

Control from CICS resource level rather than z/OS dataset level.

JCICSJCT & KCICSJCT - Journal Control Table

MCICSPPT & NCICSPPT - Program Properties Table

Controls who can START *programs* from within other programs.

PCICSPSB & QCICSPSB - IMS

SCICSTST & UCICSTST - Temporary Storage Table

RCICSRES & WCICSRES - Web Resources

Additional classes can be added to RACF if these default classes are not appropriate and they can be managed from within ICE.

For additional detail on these resource types please obtain a copy of CICS Essentials - Auditing CICS - a Beginners Guide available from www.newera.com/CICS