Alphabet Soup

SAF Classes in CICS

White Paper by Julie-Ann Williams

CICS security is quite a complex subject with many layers and facets. Understanding of the basic functions is essential in order to appreciate the very sophisticated way that the various SAF classes are exploited by CICS.

Many organisations today use only a very limited subset of the functions available. For example, CICS transaction level protection is found quite commonly but other classes are rarely implemented. SAF security related parameters in the CICS SIT can be used to provide a basic security shell to CICS applications whether the external security manager in use is RACF, CA ACF2 or CA Top Secret.

In this paper I hope to use my experience working in the field to explain how to provide for a better secured CICS environment suitable for our increasingly open applications to run in.

NewEra Software White Paper
One of a Series of CICS Essentials

SIT Parms & Default Classes

There are today 15 parameters in CICS SIT which enable various SAF classes for inclusion. Whilst this often increases at new releases, CICS 4.1 did not implement any more Xnnn SIT security parms. The parameters are detailed below with the default RACF classes that will be used when active are also listed:

XAPPC

The XAPPC parm enables Application Peer to Peer Communication, or APPC, partner-LU verification aka RACF LU6.2 bind-time security.
Default RACF class: *APPCLU*

XCMD

CICS command security controls the use of system programming (SP) commands such as CEMT with INQUIRE, CREATE, DISCARD, PERFORM and SET. CEMT is particularly important to control as it is used to close the CICS service under normal conditions.
Default RACF classes: *CCICSCMD VCICSCMD*

XDB2

When external security has been implemented in DB2, this parameter allows the organisation to specify the RACF class defined by the organisation which contains the security information needed.
Default RACF class: No default – must be specified if active

XDCT

The XDCT parameter allows protection of Transient Data Queues. Queues are sequential storage facilities, generally transitory in nature. Popular “in the old days” for mainframe channel attached printing, transient data queues provide general queue functions. They are now often used for offloading work for asynchronous updates.
Default RACF classes: *DCICSDCT ECICSDCT*

XEJB

Enterprise JavaBeans are extensions to Sun’s Java language and are intended to handle such common concerns as security, persistence and transactional integrity in a standard way. Implementation of EJB in a z/OS CICS environment would indicate the use of CICS Web Server and/or other “internet facing” applications.
Default RACF class: *EJBROLE*

XFCT

A file is identified to CICS by an up to 8 character file name. One or more files can be defined to CICS that refer to the same physical data set.
Default RACF classes: *FCICSFCT HCICSFCT*

XHFS

XHFS is one of the newer SIT parms. It is used to specify whether CICS is to check with RACF if the transaction user is authorised to access files in the USS file system. This offers very limited functionality presently as the

checking applies only to the ID of the Web client when CICS Web support is returning USS file data as static content as identified by a URIMAP definition.
Default RACF classes: *UID GID*

XJCT

Users who need to write journal records must have authority to write to the CICS JOURNALNAME. User transactions should not have write authority to DFHLOG as CICS uses it for the primary system log.
Default RACF classes: *JCICSJCT KCICSJCT*

XPCT

The START command enables a CICS application program to start another transaction associated with a terminal other than the one from which the start command is issued. XPCT allows an installation to control who has the authority to issue them.
Default RACF classes: *ACICSPCT BCICSPCT*

XPPT

CICS application programs can invoke more than 1 program. XPPT is the SIT parm to implement if you want to control all programs run by individual users.
Default RACF classes: *MCICSPPT NCICSPPT*

XPSB

PSBs consist of one or more program communication blocks or PCBs. PCBs describe a particular application program's interface to an IMS database. Only used with IMS.
Default RACF classes: *PCICSPSB QCICSPSB*

XRES

XRES is a fairly new parm and is used to secure a number of web service type resources.
Default RACF classes: *RCICSRES WCICSRES*

XTRAN

This is the most commonly implemented resource protection class in CICS environments across the world. It is used to control who can execute what transactions and, alone, is considered to be a minimum level of security implementation.
Default RACF classes: *TCICSTRN GCICSTRN*

XTST

Temporary storage queues are typically used for shared reading, writing, and updating by multiple transactions e.g. scratchpad for shared data. Data stored in recoverable auxiliary storage is retained after a CICS region terminates and can be recovered in a subsequent restart.
Default RACF classes: *SCICSTST UCICSTST*

XUSER

XUSER will activate surrogate user security checking, as well as AUTHTYPE checking for DB2.
Default RACF class: *SURROGAT*

Deciding What to Protect?

Some of the decisions to be taken when implementing external CICS security are easy - do I have this software? While some are much trickier - do I have complex transactions - and what is a complex transaction anyway? The first question for all sites has to be - Do I need external security with CICS? This is easy as the answer will always be yes. It is implemented using another SIT parm SEC=YES.

The second decision is whether to protect transactions. Again, this is a no-brainer. CICS transaction security is the very minimum to consider. Everything else hangs off of the transaction, i.e. if XTRAN=NO is specified, nothing can be protected. If XTRAN=YES is specified then ALL transactions will be protected and profiles must be defined to cover every single transaction. Implementation of the lower level resources is enabled using the CMDSEC and/or RESEC parameter either globally in the SIT or in the RDO for individual transactions allowing truly granular implementation. The profiles will then be loaded into CICS storage when the related Xnnn parm is selected in the SIT.

CICS Systems Programmer (SP) command protection can be achieved quite simply and so should be a yes. It is activated by specifying YES or a class name in the XCMD parameter and also using CMDSEC parameter in the SIT and/or individual RDO entry. This implementation is common across all of the other resource types and allows for more granular security than can be achieved with XTRAN, which applies to every transaction.
The product related parms are next. Do you have: DB2, IMS, LU 6.2, EJB or web services needs?

And finally, there are all of the other CICS security parameters. This is the subset that represents the hardest decision to make and the most work to implement. This is probably why so few installations use any of these classes. However, the additional control that can be applied make the effort worthwhile – just ask your Auditors what they think of the idea...

Transient Data Queue Control allows an organisation to maintain granular security over transient data some of which can be used for recovery processing.

Limiting update access can help prevent interference with recovery data.
CICS File Control can help to define correct separation of duties in complex transaction environments, e.g. if a single transaction can process multiple files with different levels of access requirements then XFCT should be implemented.

CICS Journals are used for recoverability and developers should not have write access to DFHLOG journals.

…there are all of the other CICS security parameters. This is the subset that represents the hardest decision to make and the most work to implement. This is probably why so few installations use any of these classes.

However, the additional control that can be applied makes the effort worthwhile.

Started Transactions are how CICS handles activity that is not initiated by a person. Again, complex transaction users should consider implementing XPCT to limit who can do what from one start point.

CICS Program Control is useful if your application programs LINK to other programs. A user should have specific access to all of the programs which will be executed.

Temporary Storage Control is rarely implemented as the function is designed for sharing.

Build the profiles in your chosen external security manager before changing the SIT parameters once the decision about what to protect has been made. CICS will fail to start if it does not find all expected profiles.

And make sure that you know who has been changing SIT parameters and for what reasons. Auditors will want to know.

CICS Program Control is useful if your application programs LINK to other programs. A user should have specific access to all of the programs which will be executed.

So in summary: deciding what to protect in CICS is a very complicated topic but well worth the effort in terms of increased security and auditability.

Good luck!

Making it Easier on Yourself

NewEra products have helped save many customers from errors and oversights in their maintenance of system parameters and datasets over the years. The Control Editor continues in that vein in that it enhances the security already in place by allowing sites to name the data sets it wants to protect.

In audit parlance, The Control Editor represents a compensating control. With The Control Editor in place, staff can continue to do their work the same way except that everything they do is recorded and subject to monitoring. When applied to the libraries which contain the CICS SIT definitions, The Control Editor provides a history of changes, reducing the possibility of manipulating the system.

It can also be highly useful for SAS70 and other regularly recurring audits where an auditor needs to know the changes that have taken place over a period of time.

And as an added advantage, the sysprogs have a tool that they can use when they need to recall which button they pushed last and who pushed the button.

NewEra also have a new product called DFHz Explorer which can be used to examine various criteria on a running CICS Region(s). This facility can be used to make sure that a specific CICS Region (or group of Regions) comes up exactly as expected and to give early warning of any tampering.

NewEra easily brings added value to the management of complex CICS environments. Their products can offer real time savings.

About the Sponsor – NewEra Software

NewEra Software, Inc. was founded in 1989 with the specific goal of developing, marketing and supporting innovative system management software tools and services.

Thanks to the continued support of thousands of systems professionals worldwide that have come to depend on NewEra, the company has become an industry leader and its products the industry standard for repair, recovery, data erasure and integrity of large systems.

The Control Editor is a separately licensed component of NewEra's Image Control Environment (ICE). Other ICE Applications include Image FOCUS, IODF Explorer, DFHz Explorer, Fast DASD Erase for z/OS, and the UACC Explorer (RACF).

For more information about these products, visit our website

www.newera.com

or call 1-800-421-5035.

NewEra Software

155 E. Main Ave., Suite 130

Morgan Hill, CA 95037

PHONE:

(800) 421-5035

(408) 201-7000

FAX:

(408) 201-7099

EMAIL:

support@newera.com

About the Author – Julie-Ann Williams

Julie-Ann is a passionate System z Advocate and loves to show people working on the other platforms how IT should be done.

She ran the UK GSE Large Systems Working Group for 10 years, is a regular speaker at technical events and currently works for millennia... as a Senior Technical Specialist and Mentor primarily in the EMEA region.

millennia...

55a Rowtown

Addlestone

Surrey

KT15 1EN

UK

+44 (0)1932 887489

julie@sysprog.co.uk

www.sysprog.co.uk