GUIDE SHARE EUROPE UK REGION

GSE UK Conference 2018

Better, stronger, faster; The Mainframe….. the Machine!

z/OS® Security and RACF® Update

Session FA
Mark Nelson, CISSP®, CSSLP®
RACF Design and Development, IBM®

markan@us.ibm.com

Roadmap

- **Pervasive Encryption**
  - Overview
  - z/OS Data Set Encryption
- **RACF**
  - RACF Parmlib
  - FLAC Enhancements – Field Level Access Control Enhancements
  - Enhanced Generic Owner
  - E-Mail Address – New E-Mail field
  - UID(0) Consistent Reporting
  - RACF Installed Certificate Authority Certificates
  - Security Deployment Descriptor
  - Common Criteria
- **PKI Services**

PERVASIVE ENCRYPTION

Data Protection and Compliance are Business Imperatives

> *"It's no longer a matter of if, but when …"*

26%

Likelihood of an organization having a data breach in the next 24 months [1]

European Union General Data Protection Regulation (GDPR

GDPR

DSS COMPLIANT

Payment Card Industry Data Security Standard (PCI-DSS)

$4M

Average cost of a data breach in 2016 [2]

Of the **9 Billion** breached since 2013 only **4%** were encrypted

Health Insurance Portability and Accountability Act (HIPAA)

HIPAA

1, 2 Source: 2016 Ponemon Cost of Data Breach Study: Global Analysis -- http://www.ibm.com/security/data-breach/

3 Source: Breach Level Index -- http://breachlevelindex.com/

Data Protection through Encryption

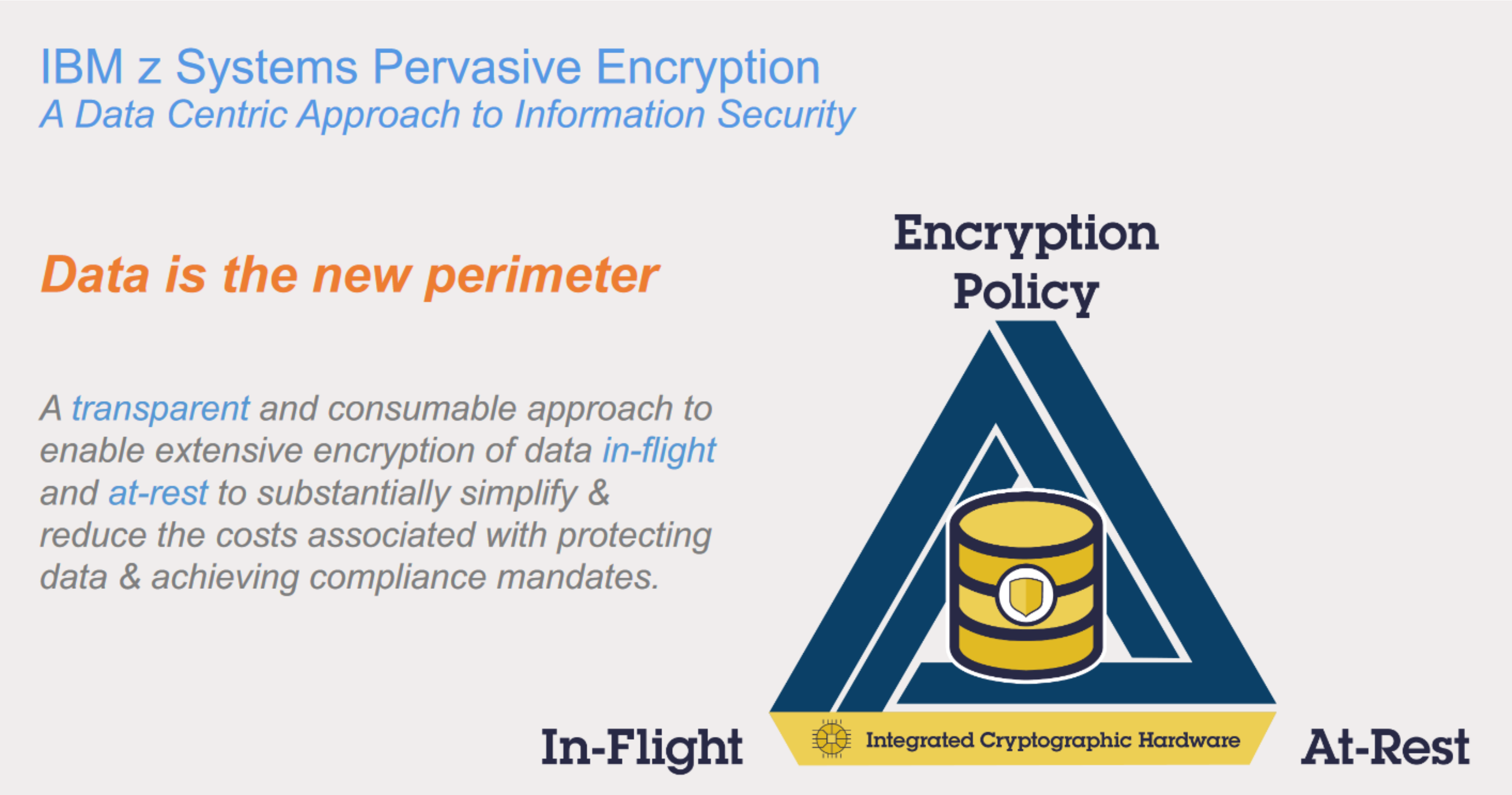
Extensive use of encryption is one of the most impactful ways to help reduce the risks and financial losses of a data breach and help meet complex compliance mandates.

However, implementing encryption can be a complex process …

1. **What data should be encrypted?**
2. **Where should encryption occur?**
3. **Who is responsible for encryption?**

Comprehensive data protection requires a huge investment to deploy point solutions and/or enable encryption directly in the applications

Data is the New Perimeter



IBM z Systems Pervasive Encryption

A Data Centric Approach to Information Security

Data is the new perimeter

A transparent and consumable approach to enable extensive encryption of data in-flight and at-rest to substantially simplify & reduce the costs associated with protecting data & achieving compliance mandates.

Encryption Policy
In-Flight
Integrated Cryptographic Hardware
At-Rest

IBM has been Providing Security & Encryption Solutions for over 40 years

- **IBM submits the Lucifer cipher to become the Data Encryption Standard (DES): 1974 – 1976**
- **Resource Access Control Facility (RACF), 1976**
- **Hardware Cryptography using IBM 3845 Channel Attached DES/TDES: 1977 - 1979**
- **IBM 4753 Channel Attached CCA Unit with smart cards and signature dynamics pen: 1989**
- **Key management built into operating system (ICSF): 1991**
- **Distributed Key Management System (DKMS) (1990's)**
- **Trusted Key Entry (TKE) Workstation: ~1997**
- **Encryption Facility for z/OS: 2005**
- **TS1120 Encrypting Tape Drive: 2006**
- **LTO4 Encrypting Tape Drive: 2007**
- **Tivoli Encryption Key Lifecycle Manager: 2009**
- **Self-Encrypting Disk Drives, DS8000: 2009**
- **System z10 CPACF Protected Key Support: 2009**
- **Crypto Express3 Crypto Coprocessor: 2009**
- **z Systems z196 with additional CPACF encryption modes: 2010**
- **Crypto Express4S Crypto Coprocessor: 2012**
- **z Systems zEC12 with Public Key Cryptography Standards – PKCS#11: 2012**
- **Crypto Express5S Crypto Coprocessor: 2015**
- **z Systems z13 with Visa Format Preserving Encryption: 2015**
- **Z Systems Secure Service Container**
- **Pervasive Encryption (z14, z/OS, z/VM, z/TPF, Linux on z): 2017**

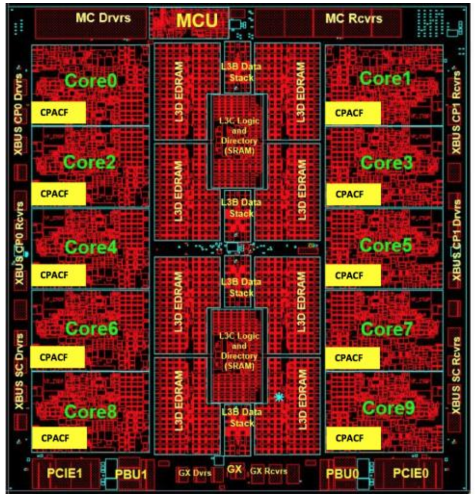
z14 Integrated Cryptographic Hardware

CP Assist for Cryptographic Functions (CPACF)

- Hardware accelerated encryption on every microprocessor core
- Performance improvements of up to 7x for selective encryption modes

Crypto Express6S

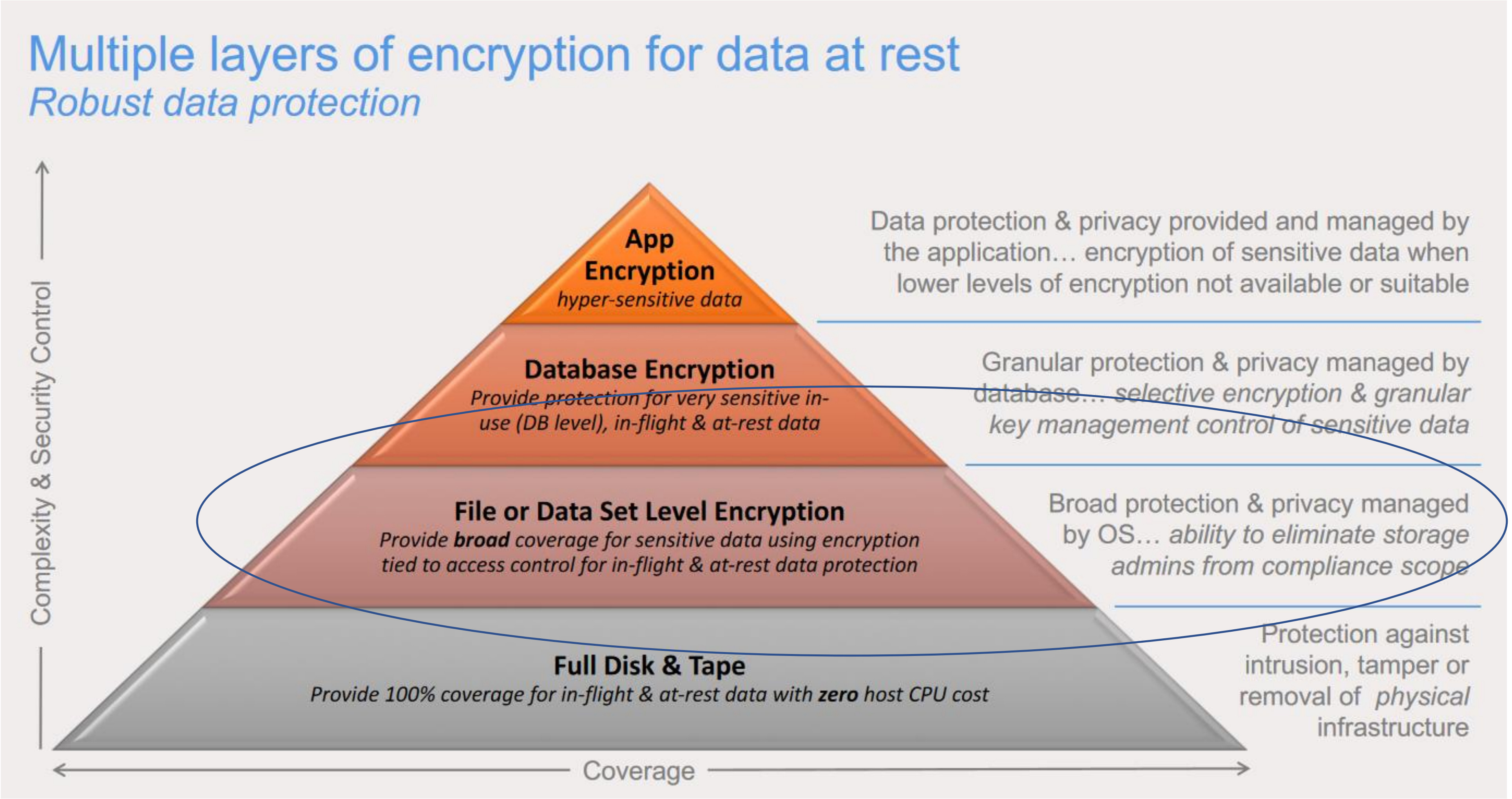
- Next generation PCIe Hardware Security Module (HSM)
- Performance improvements up to 2x
- Industry leading FIPS 140-2 Level 4 Certification Design



Why is it valuable:

- More performance = lower latency + less CPU overhead for encryption operations
- Highest level of protection available for encryption keys
- Industry exclusive “protected key” encryption

Layers of Encryption



Multiple layers of encryption for data at rest

Robust data protection

App Encryption
hyper-sensitive data

Data protection & privacy provided and managed by the application… encryption of sensitive data when lower levels of encryption not available or suitable

Database Encryption
Provide protection for very sensitive in-use (DB level), in-flight & at-rest data

Granular protection & privacy managed by database… *selective encryption & granular key management control of sensitive data*

File or Data Set Level Encryption
*Provide **broad** coverage for sensitive data using encryption tied to access control for in-flight & at-rest data protection*

Broad protection & privacy managed by OS… *ability to eliminate storage admins from compliance scope*

Full Disk & Tape
*Provide 100% coverage for in-flight & at-rest data with **zero** host CPU cost*

Protection against intrusion, tamper or removal of *physical* infrastructure

Pervasive Encryption July, 2017 Announcement

IBM US Software Announcement 217-246, 17 July, 2017

- **z/OS V2.3** is designed to provide **policy-enabled** enhanced data protection, including:
  - **z/OS data set encryption without requiring changes to application programs.**
  - **zFS** encryption of individual files (file content), access control lists, security information, and symbolic link contents.
  - **Encryption of Coupling Facility data**, including list and cache structures, under the control of the Coupling Facility Resource Management (CFRM) policy. **The data is encrypted as it travels on the CF link and remains encrypted while resident in the CF.**
  - **z/OS Communications Server Encryption Readiness Technology (zERT)** to help z/OS administrators to determine which TCP and Enterprise Extender (EE) traffic patterns to and from their z/OS systems meet approved encryption criteria and which do not.

Pervasive Encryption July, 2017 Announcement…

IBM US Software Announcement 217-246, 17 July, 2017

- **Additional Support for Pervasive Encryption**
  - **IMS V14** supports z/OS data set encryption for select data sets.
  - **DB2 for z/OS, V11 and V12** support for z/OS data encryption.

PERVASIVE ENCRYPTION: DATA SETS

Pervasive Encryption for z/OS Data Sets

Encrypted data is transparent to the application when the application uses standard access method APIs*. No application changes are needed.*

Supported access methods

- BSAM/QSAM sequential data sets, extended format only (Version 2)
- VSAM and VSAM/RLS (KSDS, ESDS, RRDS, VRRDS, LDS), extended format only
- New option to allow access to data in the encrypted form

Restrictions

- System data sets (such as catalogs, HSM data sets, RACF data sets) must not be encrypted unless otherwise specified
- Encrypted data sets are supported only on 3390 device types
- Sequential (non-compressed) data sets with a BLKSIZE of less than 16 bytes cannot be encrypted as they cannot be extended format
- Data sets used during IPL must not be encrypted

Pervasive Encryption for z/OS Data Sets…

- **Data set encryption is available *now* for all in-service z/OS releases**
- **All systems sharing the data must be at a minimum level of z/OS V2.1 or higher**
  - **z/OS V2.3**
  - **z/OS V2.2**
    - New function PTFs (main APAR OA50569)
  - **z/OS V2.1**
    - Coexistence PTFs (APAR OA50569)
      - Supports read and write access to existing encrypted data sets.
  - **ICSF**
    - HCR77C0 or HCR77A0 through HCR77B1 with APAR OA50450
- **Requires:**
  - Feature 3863, CP Assist for Cryptographic Functions (CPACF)
  - The minimum processor hardware is z196 or higher with CEX3 or later

Pervasive Encryption for z/OS Data Sets…

- **One of the basic strengths of the z System architecture is the support of different levels of protections for keys**
  - **Secure key:** The key material is exposed only when in use in the hardware security module (HSM), which for z Systems is the Crypto Express card
    - Tamper evident/resistant
    - Internal high performance for cryptographic operations
    - **Advantage:** Highly secure as the key is *never* in the clear outside of the HSM
    - **Disadvantage:** Data must be marshalled to and from the card
  - **Clear key:** The key, while it can be protected by normal z Architecture means (storage key, address space isolation, etc.), is located in main memory for at least the time of its use, such as by the CPACF.

Pervasive Encryption for z/OS Data Sets…

- **DFSMS is designed to use protected keys which allow the use of a key that never appears in the clear in storage and is not available to the operating system (like z/OS) or any application, but which can still be used by CPACF**
  - Protected keys usually (and should!) start as secret keys, but are marked as eligible for use as protected keys with **SYMCPACFWRAP(YES)** in the ICSF segment for the CSFKEYS profile which defines the key to RACF
  - **SYMCPACFRET(YES)** allows the *wrapped* protected key to be returned to the caller
  - **Both SYMCPACFWRAP(YES) and SYMCPACFRET(YES) are needed for encrypting data sets**
  - **Advantage:** Substantially higher performance than secure key operations as the data does not have to be marshalled/demarshalled to/from the crypto card

Pervasive Encryption for z/OS Data Sets…

- **Data owners who must access content will need authority to the data set as well as authority to the encryption key label**
- **System/Storage administrators who only manage the data sets need authority to the data set but not access to the encryption key label, thus protecting access to the content**
  - Prevent administrators from accessing the content
- **Different keys can be used to protect different data sets – ideal for multiple tenants or data set specific policies.**
- **Many utilities can process data preserving encrypted form**
  - COPY, DUMP and RESTORE
  - Migrate/Recall, Backup/Recover, Dump/Data Set Restore
  - PPRC, XRC, FlashCopy®, Concurrent Copy, etc.

Pervasive Encryption for z/OS Data Sets…

- **With this support, DFSMSdfp calls ICSF to get key information. Users creating, updating, or accessing encrypted data will need access to certain ICSF services**
  - – For example, CSFKRR2 (CKDS Key Record Read2 service) in the CSFSERV class
- **The key to be used for the encryption is specified by its ICSF key-label. The user must be authorized to this key-label in the CSFKEYS class**
  - – You can permit users and groups to the CSFKEYS resources for all purposes:

```
PERMIT key-label CLASS(CSFKEYS) ID(…) ACCESS(READ)
```

  - – Alternatively, you can allow the use of the key, but only when using it for data set encryption/decryption using the CRITERIA specification on PERMIT:

```
PERMIT key-label CLASS(CSFKEYS) ID(…) ACCESS(READ)
WHEN(CRITERIA(SMS(DSENCRYPTION)))
```

Pervasive Encryption for z/OS Data Sets…

- **A sequential or VSAM data set would be defined as ‘encrypted’ when a key label is supplied on allocation of a new sequential or VSAM data set. The key label can be specified (in order of precedence):**

  – RACF data set profile

  ```
  ALTDSD ‘PROJECT.DATA.*’ UACC(NONE) DFP(DATAKEY(key-label))
  ```

  – JCL, dynamic allocation, TSO allocate, IDCAMS

  ```
  //DD1    DD DSN=A.B.C,DISP=(NEW,CATLG),DATACLA=DSN1DATA,
  //          MGMTCLASS=DSN1MGMT,STORCLAS=DSN1STOR,
  //          DSKEYLBL=’LABEL.FOR.DSN1’
  ```

  – SMS Data Class

- **The derived key label is stored in the catalog when the data set is allocated**

Pervasive Encryption for z/OS Data Sets…

- **Users who want to create encrypted data sets with the key-label coming from *anywhere other than the DFP segment* must have READ authority to the resource STGADMIN.SMS.ALLOW.DATASET.ENCRYPT in the FACILITY class**

- **If an encryption key label is specified for a DASD data set that is not extended format, the key label is ignored and the data set is successfully created as non-encrypted non-extended format data set**

  If you would rather have the system fail the allocation, then give the user READ authority to the STGADMIN.SMS.FAIL.INVALID.DSNTYPE.ENC resource in the FACILITY class

Pervasive Encryption with IBM z Systems

Enabled through full-stack platform integration

Integrated Crypto Hardware

Hardware accelerated encryption on every core – CPACF performance improvements of up to 7x

Next Gen Crypto Express6S – up to 2x faster than prior generation

Data at Rest

Broadly protect Linux file systems and z/OS data sets[1] using policy controlled encryption that is transparent to applications and databases

Clustering

Protect z/OS Coupling Facility[2] data end-to-end, using encryption that's transparent to applications

Network

Protect network traffic using standards based encryption from end to end, including encryption readiness technology[2] to ensure that z/OS systems meet approved encryption criteria

Secure Service Container

Secure deployment of software appliances including tamper protection during installation and runtime, restricted administrator access, and encryption of data and code in-flight and at-rest

Key Management

The IBM Enterprise Key Management Foundation (EKMF) provides real-time, centralized secure management of keys and certificates with a variety of cryptographic devices and key stores.

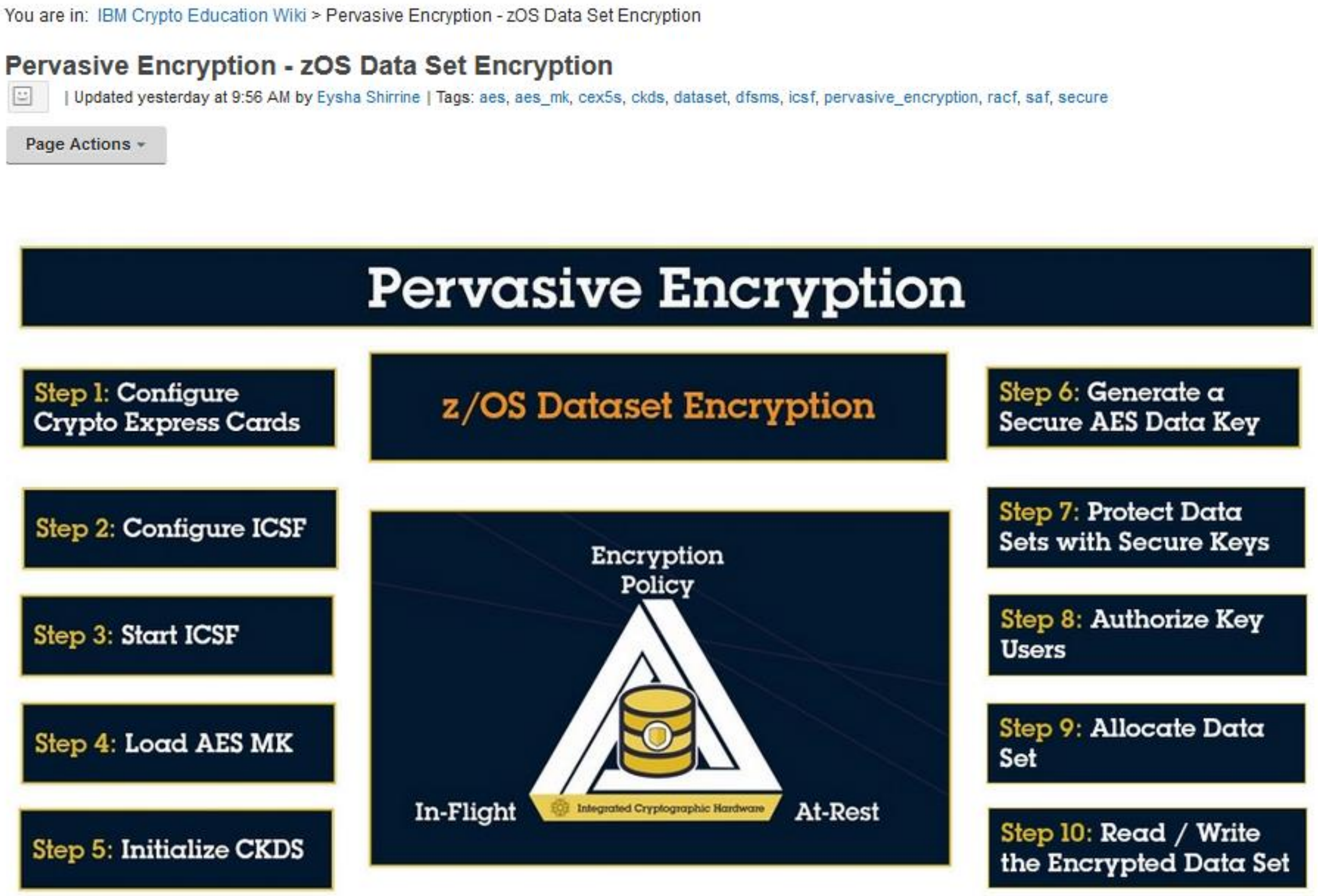
And we're just getting started …

[1] Statement of Direction* in the z/OS Announcement Letter (10/4/2016) - http://ibm.co/2ldwKoC

[2] IBM z/OS Version 2 Release 3 Preview Announcement Letter (2/21/2017) - http://ibm.co/2l43ctN

It’s all about defense in depth!

Shameless Plug #1: The Essential Wiki for Pervasive Data Set Encryption
http://ibm.biz/zos-pervasive-encryption-wiki



You are in: IBM Crypto Education Wiki > Pervasive Encryption - zOS Data Set Encryption

Pervasive Encryption - zOS Data Set Encryption

| Updated yesterday at 9:56 AM by Eysha Shirrine | Tags: aes, aes_mk, cex5s, ckds, dataset, dfsms, icsf, pervasive_encryption, racf, saf, secure

Page Actions ▾

Pervasive Encryption

z/OS Dataset Encryption

Step 1: Configure Crypto Express Cards

Step 2: Configure ICSF

Step 3: Start ICSF

Step 4: Load AES MK

Step 5: Initialize CKDS

Step 6: Generate a Secure AES Data Key

Step 7: Protect Data Sets with Secure Keys

Step 8: Authorize Key Users

Step 9: Allocate Data Set

Step 10: Read / Write the Encrypted Data Set

RACF PARMLIB

RACF Parmlib: The RACF Data Set Name Table

Data set name table (ICHRDSNT) specifies data set names and configuration options for the RACF database:

- **Data Set Name(s) of the primary and backup RACF DB(s)**
- **Backup database options**
- **Number of resident data blocks**
- **RACF sysplex communications options**

```
ICHRDSNT CSECT
         DC    AL1(3)                  Number of entries
         DC    CL44'RACFTEST.RDB.PRIM1'  First RACF primary data set
         DC    CL44'RACFTEST.RDB.BACK1'  First RACF back-up data set
         DC    AL1(255)                Number of index/data blocks
         DC    B'10000000'             Flags: All Updates execpt stats
*
*
         DC    CL44'RACFTEST.RDB.PRIM2'  Second RACF primary data set
         DC    CL44'RACFTEST.RDB.BACK2'  Second RACF back-up data set
         DC    AL1(255)                Number of index/data blocks
         DC    B'10000000'             Flags: All Updates execpt stats
*
*
         DC    CL44'RACFTEST.RDB.PRIM3'  Third  RACF primary data set
         DC    CL44'RACFTEST.RDB.BACK3'  Third  RACF back-up data set
         DC    AL1(255)                Number of index/data blocks
         DC    B'10000000'             Flags: All Updates execpt stats
         END
```

RACF Parmlib: The RACF Range Table

- **Range table (ICHRRNG) determines in which data set of the RACF database RACF places each profile:**
  - Number of Ranges
  - For each range - Profiles range start and RACF data set number

```
ICHRRNG  CSECT
         DC   F'3'                         Count of ranges
*
RANGE1   DC   XL44'0'
         DC   AL1(1)                       Data set to put profiles
*
RANGE2   DC   XL44'0'                      Prime range with nulls
         ORG  RANGE2
         DC   C'I'                         Range start value
         ORG
         DC   AL1(2)                       Data set to put profiles
*
RANGE3   DC   XL44'0'                      Prime range with nulls
         ORG  RANGE3
         DC   C'Q'                         Range start value
         ORG
         DC   AL1(3)                       Data set to put profiles
         END
```

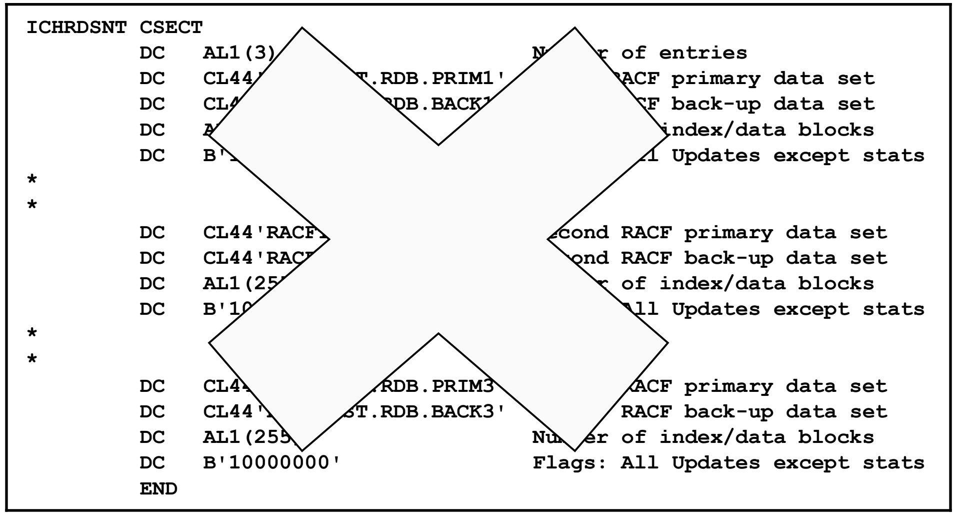
- **Requires assembler skills to create and maintain**
- **Do you still have your source?**
- **Wouldn’t it be nice if there was a better way?**

RACF Parmlib: IRRPRMxx **New**

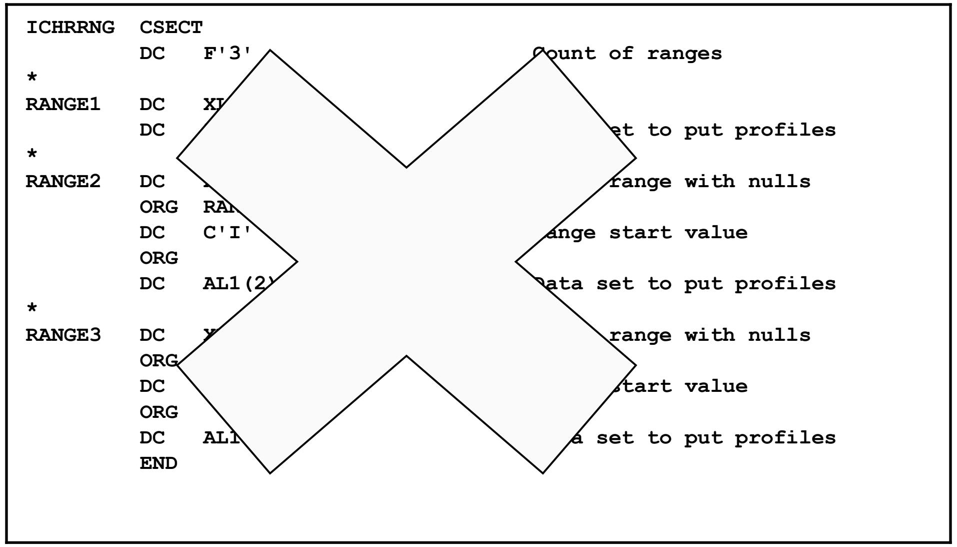
- **Support RACF configuration options with a SYS1.PARMLIB member**
- **New IEASYSxx keyword: RACF=xx**
- **New RACF PARMLIB member: IRRPRMxx**
- **Content:**

```
DATABASE_OPTIONS
SYSPLEX(NOCOMMUNICATIONS | COMMUNICATIONS |  DATASHARING)
DATASETNAMETABLE
     ENTRY
         PRIMARYDSN(data-set-name)
         BACKUPDSN(data-set-name)
         UPDATEBACKUP(ALL | NONE | NOSTATS)
         BUFFERS(num-buffers)
     RANGETABLE
         START(start-value [CHAR | HEX])  ENTRYNUMBER(entry-sequence-number)
```

RACF Parmlib: IRRPRMXX **New** …



```
ICHRDSNT CSECT
         DC    AL1(3)                   N    r of entries
         DC    CL44'      T.RDB.PRIM1'         ACF primary data set
         DC    CL4        DB.BACK1           F back-up data set
         DC    A                            index/data blocks
         DC    B'1                           ll Updates except stats
*
*
         DC    CL44'RACF                 cond RACF primary data set
         DC    CL44'RAC                   ond RACF back-up data set
         DC    AL1(25                     r of index/data blocks
         DC    B'10                        All Updates except stats
*
*
         DC    CL4         RDB.PRIM3         ACF primary data set
         DC    CL44'      ST.RDB.BACK3'      RACF back-up data set
         DC    AL1(255                  Nu   er of index/data blocks
         DC    B'10000000'              Flags: All Updates except stats
         END
```



```
ICHRRNG  CSECT
         DC    F'3'                     Count of ranges
*
RANGE1   DC    XL
         DC                                  et to put profiles
*
RANGE2   DC                                  range with nulls
         ORG   RA
         DC    C'I'                       ange start value
         ORG
         DC    AL1(2                    Data set to put profiles
*
RANGE3   DC    X                             range with nulls
         ORG
         DC                                  start value
         ORG
         DC    AL1                        a set to put profiles
         END
```

```
DATABASE_OPTIONS
    SYSPLEX(NOCOMMUNICATIONS)
    DATASETNAMETABLE
      ENTRY
        PRIMARYDSN('RACFTEST.RDB.PRIM1')
        BACKUPDSN('RACFTEST.RDB.BACK1')
        UPDATEBACKUP(NOSTATS)
        BUFFERS(255)
      ENTRY
        PRIMARYDSN('RACFTEST.RDB.PRIM2')
        BACKUPDSN('RACFTEST.RDB.BACK2')
        UPDATEBACKUP(NOSTATS)
        BUFFERS(255)
      ENTRY
        PRIMARYDSN('RACFTEST.RDB.PRIM3')
        BACKUPDSN('RACFTEST.RDB.BACK3')
        UPDATEBACKUP(NOSTATS)
        BUFFERS(255)
    RANGETABLE
       START('00' HEX) ENTRYNUM(1)
       START('I' CHAR) ENTRYNUM(2)
       START('Q' CHAR) ENTRYNUM(3)
```

RACF Parmlib: RACPRMCK Command **New**

RACPRMCK – New RACF Command which validates the syntax of one or more IRRPRMxx PARMLIB members.

- **RACPRMCK MEMBER**(*member-name1, member-name2, member-name3*)
  - Up to 3 PARMLIB members can be specified
  - Member can have any name, but it must be in a dataset that is part of the PARMLIB concatenation
- **Example:**
  - `RACPRMCK MEMBER(IRRPRM01)`
- **Issues messages when errors are detected:**

```
IRRY107E RACF DETECTED AN ERROR WHILE PARSING PARMLIB.
         ERROR FOUND IN MEMBER member-name
         LINE xxxx: some-keyword KEYWORD VALUE IS NOT VALID.
```

RACF Parmlib: IRRUT400 Enhancements **New**

- **RACF database split / merge / extend utility program allows splitting of the database using the specification of an output range**
- **IRRUT400 EXEC Parameters:**
  - NOTABLE – (default) All profiles copied into a single output data set
  - TABLE(table-name) – Specify the name of an ICHRRNG style module
  - **RANGE(member)** – *NEW* - Specify the name of a PARMLIB-style member
- **Example:**

```
// EXEC PGM=IRRUT400,PARM=’NOLOCKINPUT,RANGE(IRRPRM02)’
```

- **A new DD statement RANGEDD, can be specified to indicate the fully qualified data set name which contains the range member**
- **Example:**

```
// EXEC PGM=IRRUT400,PARM=’NOLOCKINPUT,RANGE(IRRPRM02)’
...
//RANGEDD DD DSN=INSTALL.WORK.PARMLIB,DISP=SHR
```

Note: If RANGEDD is not specified, the dataset is selected from the PARMLIB concatenation

RACF Parmlib: The DSN2PRM Utility **New**

- **Convert the ICHRDSNT and/or ICHRRNG load modules to generate a matching PARMLIB member**

- **DSN2PRM is an "as-is" REXX utility developed by the RACF component test team to convert the ICHRDSNT and/or ICHRRNG load modules into a RACF PARMLIB member**
  - Supported only through the RACF-L discussion list

- **Available from**:
  - **Utility**: ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/dsnt2prm/dsnt2prm.txt
  - **Documentation:** ftp://public.dhe.ibm.com/eserver/zseries/zos/racf/dsnt2prm/dsnt2prm.pdf

FLAC ENHANCEMENTS

FLAC – Field Level Access Checking (“FLAC”)

- **Information in the non-base segment of a RACF profile can be accessed only by**
  - Users with the SPECIAL attribute (read, update, or delete)
  - Users with the AUDITOR or ROAUDIT attribute (read)
  - Users with authority to the resources in the FIELD class
    - READ authority allows viewing, UPDATE authority allows modification
    - The resource name is <profile>.<segment>.<field>
    - Generic profiles are supported

- **For example, to give the ID TSOADM the ability to manipulate and list the ACCTNUM field in *all* user profiles:**

```
RDEFINE FIELD    USER.TSO.TACCNT UACC(NONE)
PERMIT  USER.TSO.TACCNT CLASS(FIELD) ID(TSOADM) ACCESS(UPDATE)
```

- **But, *this is an “all or nothing” authority* in the sense that you can’t limit the authority to a set a profiles. *Any RACF group-SPECIAL authority is not considered.***

FLAC – Field Level Access Checking **New**

- **Starting with z/OS V2.3, you can "scope" the FLAC authority**
  - FLAC can now be scoped so it only applies to profiles which the user can also modify the base segment (with authority such as GROUP-SPECIAL)
  - Controlled by a new profile in the FIELD class: FLAC.SKIP.BASECHECK
  - Authority to the FLAC.SKIP.BASECHECK resource in the FIELD class controls the scoping behavior:
    - **Profile not defined or READ access:** Old behavior – FLAC authority can be used to modify fields for ALL users.
    - **Profile exists and NONE access:** New behavior – FLAC authority can be used to modify fields only for profiles that the administrator also has the ability to modify the base segment.

FLAC – Field Level Access Checking **New**…

- **Note that with the New FLAC processing:**
  - Users with access of NONE to FLAC.SKIP.BASECHECK are still able to alter the fields in their own user profiles that have UPDATE permission granted to &RACUID
  - The FLAC.SKIP.BASECHECK profile does not apply to RACF SPECIAL users
    - RACF SPECIAL users have never been subject to FIELD class checking, and this is unchanged..

ENHANCED GENERIC OWNER

Enhanced Generic Owner

- **GENERIC OWNER** is a feature which can allow an installation to restrict an administrators ability to create profiles in general resource classes.

- **Example to scope administrators ability to create profiles:**
  - Enable GENERICOWNER with SETROPTS:

    ```
    SETR GENERICOWNER
    ```

  - Define a generic profile ** with a special user as owner.
    - This generic profile prevents other non-special users from creating profiles in the class
  - Define a top profile for each user, covering the subset of resources in the class which the user is allowed to create. Each user should be the owner of this top profile.
  - You have created an environment where the user can create only profiles that are more specific than the user's top profile.

Enhanced Generic Owner…

- **Grouping Classes:**
  - Grouping classes are similar to generic profiles:
    - Group together sets of resources and protect them with one access list.
  - But with grouping classes the resource names can be dissimilar:
    - Controlled by adding the resource name to the profile with the **ADDMEM()** keyword.
- **Generic Owner does not work with Grouping classes:**
  - GENERICOWNER does not prevent the creation of a more specific profile if the more specific profile is created in the grouping class and is specified on the ADDMEM operand.

Enhanced Generic Owner **New**

- **Enhanced Generic Owner:**

  ```
  SETR ENHANCEDGENERICOWNER
  ```

  - Grouping classes work as you would expect:
    - Resources added to a grouping class with **ADDMEM()** are checked to ensure they pass the GENERICOWNER test.

UID(0) CONSISTENT REPORTING

UID(0) Consistent Reporting

- **Multiple users can share the same IUD, such as UID(0).** For example, SUSER1, SUSER2, SUSER3 are all super users created in that order. Their UIDs are all 0.
- **V2R2 and older:**
  - OMVS / RACF shows the oldest UID(0) user ID exists in the system
  - If SUSER1 was created first, then the listing of the file owned by SUSER2 or SUSER3 still shows SUSER1 as the owner

```
-rwxr-xr-x   3 SUSER1   STCGROUP    8192 Jun 13 02:26 suser3file
```

  - If SUSER1 was deleted, then the listing of the same shows SUSER2 as the owner

```
-rwxr-xr-x   3 SUSER2   STCGROUP    8192 Jun 13 02:26 suser3file
```

UID(0) Consistent Reporting… **New**

- **V2R3:**
  - OMVS / RACF shows the value from the PARMLIB **BPXPRMxx** SUPERUSER(user_name) keyword.
  - **BPXPRMxx - SUPERUSER(user_name):**
    - Normally the super user ID BPXROOT should exist in the system. If SUPERUSER is not specified in BPXPRMxx, BPXROOT will be used
    - If BPXROOT is not defined or if the specified user_name is not defined, “0” will be used
    - Value of user_name can by dynamically changed using the SETOMVS or SET OMVS command
  - getUMAP(IRRSUM00): Get UID-to-User-ID mapping – Updated to use SUPERUSER value for UID(0)
  - Now it should consistently show

```
-rwxr-xr-x   3 BPXROOT   STCGROUP    8192 Jun 13 02:26 suser3file
```

E-Mail Address in WORKATTR Segment

E-mail Address in WORKATTR Segment

New e-mail field in the WORKATTR segment : WAEMAIL

- **Managed with RACF commands:**

```
ADDUSER JOES WORKATTR(WAEMAIL(‘JOESMITH@us.ibm.com’))
ALTUSER JOES WORKATTR(NOWAEMAIL)
```

- **Serve as an ALIAS to the RACF user id, using the exact case matching**

- **RACF AIM Stage 3 is required**

- **Valid length is 3-246, must contain ‘@’ sign**

- **Two new functions added for R_UserMap (IRRRIM00):**
  - Return the e-mail address associated with the supplied RACF user ID
  - Return the RACF user ID associated with the supplied e-mail address

- **The e-mail value cannot be assigned to more than one user**

```
IRR52165I The value for the segment_name segment operand_name operand must
be unique.  Command processing ends.
```

E-mail Address in WORKATTR Segment…

- **Allows applications to map between UserID and e-mail field**
- **For example, JES2 uses this feature for email notification and authentication using email address instead of a User ID**
- **With the RACF WAEMAIL support, you can specify in the job card:**

  //LABEL NOTIFY USER=JOES,TYPE=EMAIL

  JES2 can use the RACF user ID, JOES, to retrieve the email address

  //LABEL NOTIFY EMAIL='JOESMITH@us.ibm.com',TYPE=MSG

  JES2 can use the email address, JOESMITH@us.ibm.com, to retrieve the RACF ID

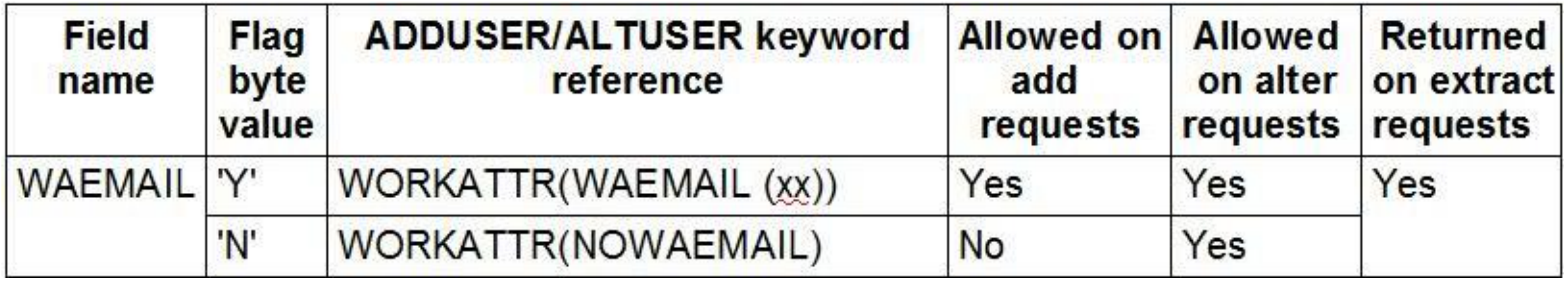
  Note that the email specified must match the string specified in the WAEMAIL field in case sensitive manner

  *JES2 also supports email notification using this form, which does not involve RACF
  //LABEL NOTIFY EMAIL='joesmith@us.ibm.com'
- **RACF support rolled back to V2R2:**
  - **RACF APAR:** OA50735 & **SAF APAR:** OA50736

E-mail Address in WORKATTR Segment…

Exploiters can obtain the e-mail address programmatically:

- RACROUTE REQUEST=EXTRACT,TYPE=EXTRACT for the standard method of getting fields from the profile (no updates)
- RACROUTE REQUEST=EXTRACT,TYPE=EXTRACT,BRANCH=YES to obtain WORKATTR fields from the ACEE (**IRRPRXTW** maps the WORKATTR fields which are returned; updated to add the e-mail field)
- **R_Admin** callable service (IRRSEQ00) to obtain user profile information (**IRRREQTB** is updated); **R_admin** can also be used for Add and Alter



Field name	Flag byte value	ADDUSER/ALTUSER keyword reference	Allowed on add requests	Allowed on alter requests	Returned on extract requests
WAEMAIL	'Y'	WORKATTR(WAEMAIL (xx))	Yes	Yes	Yes
	'N'	WORKATTR(NOWAEMAIL)	No	Yes	

RACF-INSTALLED CERTIFICATE AUTHORITIES

RACF-Installed Certificate Authority Certificates

- **Prior to z/OS V2.3, RACF installed 26 certificate authority certificates from these certificate authorities:**
  - VeriSign (13)
  - Thawte (5)
  - Integration (1)
  - Entrust (1)
  - Equifax (1)
  - ICP-Brasil (1)

- **Starting with z/OS V2.3, only these three certificate authorities are installed at IPL time:**
  1. STG Code Signing Certificate Authority
     (for RACF program signature verification for release z/OS V2R1 and earlier)
  2. STG Code Signing Certificate Authority - G2
     (for RACF program signature verification for release z/OS V2R2 and later)
  3. GeoTrust Global Certificate

- ***RACF does not delete any certificates which have already been defined***

SECURITY DEPLOYMENT DESCRIPTOR

Security Deployment Descriptor

- **Security Configuration:**
    - Post-install security configuration can be a challenge.
    - The presence of multiple ESMs makes it even more challenging:
        - Vendors end up writing multiple sets of documentation for the configuration of RACF or Top Secret or ACF2
        - IBM products often only includes documentation to configure RACF, leaving it to customers to translate from RACF to Top Secret or ACF2 when needed.
- **There's got to be a better way!**
    - Wouldn't it be nice to have the ability to specify a generic specification for the security requirement of a given product?

Security Deployment Descriptor **New**

- **NEW SAF Callable Service:**
  - **R_SecMgtOper**(IRRSMO00): Security Management Operations API
  - SAF service to allow add / alter / delete of users, groups, resources, permissions, etc.

- **Input:**
  - ***An XML document*** containing the security definitions to be defined on the target system.
  - A security definition is a user, group, dataset, or general resource profile. It can also be a group-connection, permission (permit command) or SETROPTS operation. Additionally, certain high level functions are also defined, such as the creation of a started task, Kerberos realm and others.
- **Processing and Output:**
  - The contents of the XML input are translated into RACF command text which is returned to the caller and optionally executed on the system.
  - The resulting command text and optional command execution results are returned to the caller in another XML document.

Security Deployment Descriptor **New**…

- **Deployment Descriptor SAMPLIB members:**
  - **IRRSMOEX**

    Sample REXX program to read XML input file from dataset and call IRRSMO00 to execute it. Results are saved into specified output dataset.
  - **IRRSMOMN**

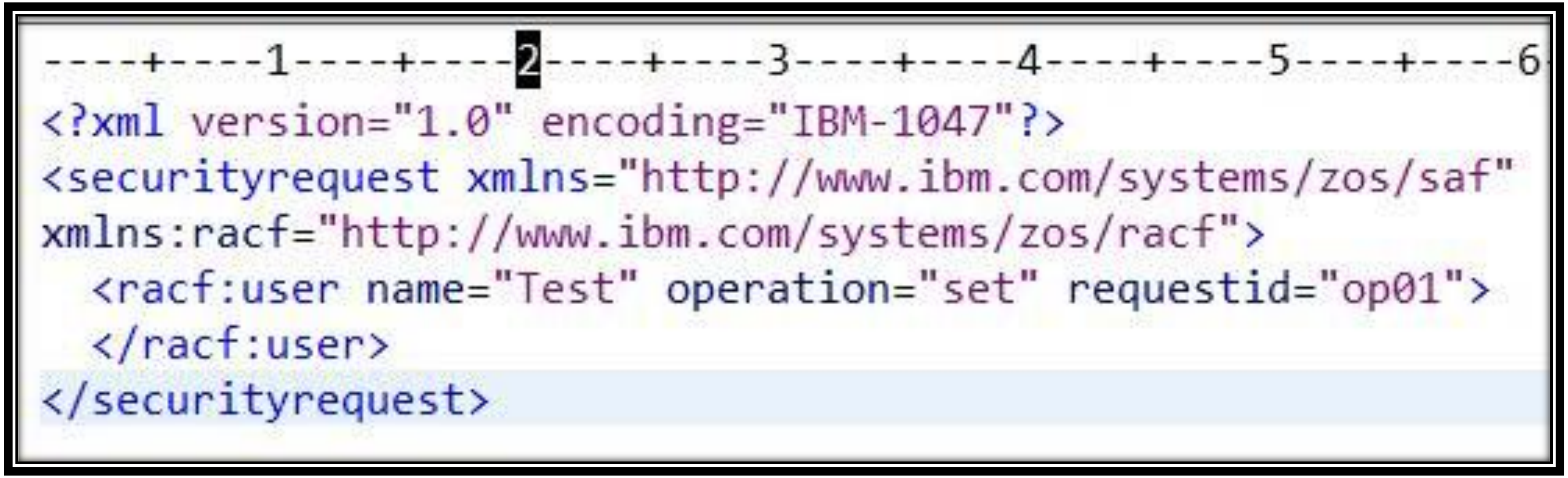
    Sample REXX program to demonstrate how to call IRRSMO00 from REXX.
  - **IRRSMOOP**

    Sample REXX program which can be used to process XML output from IRRSMO00. (formats dataset to xml readable format)

- **Java IRRSMO00 Driver:**
  - /usr/include/java_classes/IRRSMO00.jar - Sample Java program to call IRRSMO00

Security Deployment Descriptor **New** …

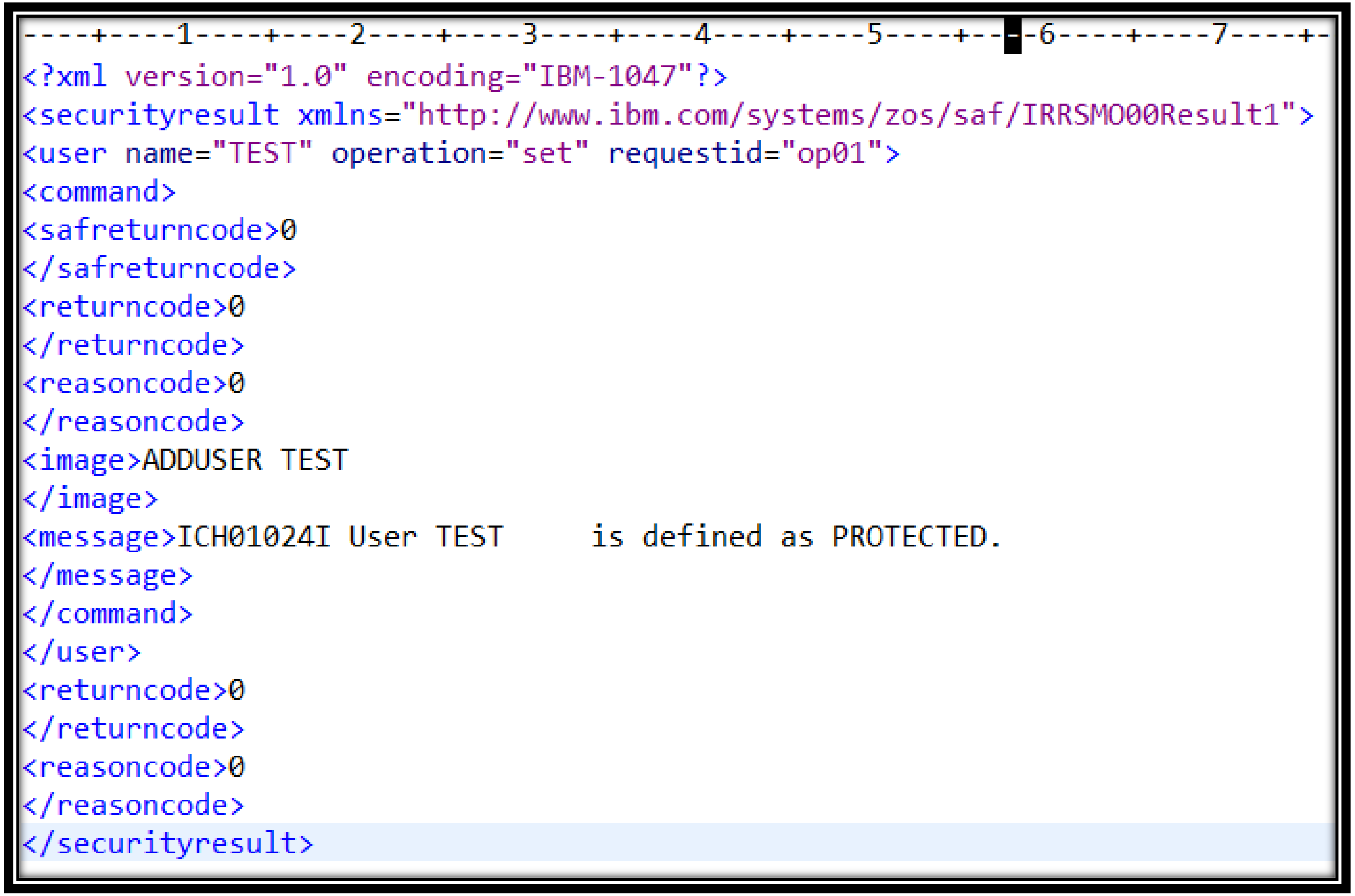
Sample Input:



```
----+----1----+----2----+----3----+----4----+----5----+----6
<?xml version="1.0" encoding="IBM-1047"?>
<securityrequest xmlns="http://www.ibm.com/systems/zos/saf"
xmlns:racf="http://www.ibm.com/systems/zos/racf">
  <racf:user name="Test" operation="set" requestid="op01">
  </racf:user>
</securityrequest>
```

Security Deployment Descriptor **New** …

Sample Output:



```
----+----1----+----2----+----3----+----4----+----5----+---6----+----7----+-
<?xml version="1.0" encoding="IBM-1047"?>
<securityresult xmlns="http://www.ibm.com/systems/zos/saf/IRRSMO00Result1">
<user name="TEST" operation="set" requestid="op01">
<command>
<safreturncode>0
</safreturncode>
<returncode>0
</returncode>
<reasoncode>0
</reasoncode>
<image>ADDUSER TEST
</image>
<message>ICH01024I User TEST     is defined as PROTECTED.
</message>
</command>
</user>
<returncode>0
</returncode>
<reasoncode>0
</reasoncode>
</securityresult>
```

COMMON CRITERIA

Common Criteria Update

- **Recent Common Criteria Evaluations of interest:**
  - z/OS V2R2/RACF, EAL5+, 25 August, 2017
  - z/OS V2R2, EAL4+, 10 July, 2017

  - z/OS V2R1/RACF, EAL5+, 14 April, 2015
  - z/OS V2R1, EAL4+, 2 September 2014

- **V2R3 z/OS (EAL4) and RACF (EAL5) are in evaluation**

PKI Services

PKI Services V2R3 – DB2 Enhancement

- **PKI Services maintains two databases:**
    - Object store (OST) – Certificate Requests
    - Issued certificate list (ICL) - Issued Certificates
- The databases can be implemented using the VSAM datasets or DB2 tables.
- **DB2 Availability:**
    - When PKI Services starts, it will check if DB2 is available. It will stop initialization if DB2 is not available. But after the initialization PKI Services is not aware of the unavailability of DB2 and keeps on processing requests even when DB2 fails.
- **Solution:**
    - Enable PKI Services to shut down when DB2 is not available or to wait for DB2 to come back to resume its functions.
    - Enables PKI to report DB2 issues ASAP and resume operation automatically once the DB2 issue is solved.
    - New option in pkiserv.conf file: DBWAITTIME

PKI Services V2R3 – Liberty Support

- **With z/OS V2R3, PKI Services can use a lightweight version of the Websphere Application Server (WAS) call “Liberty Profile” for JavaServer pages (JSPs)**

- **Benefits of the Liberty Profile WebSphere Application Server:**
  - **Lightweight** - loading of functions is optimized to achieve an smaller footprint
  - **Fast** – server starts faster and application runs faster
  - **Multiple CA Domains -** Enable PKI Services to run multiple instances of CA domains with different sets of Java Server pages (JSP)
  - **Good fit for PKI Services** - since it does not need the full traditional WAS capabilities and Liberty provides all the needed functions
  - Shipped with z/OS base in V2R3

Shameless Plug #2: Podcasts

- **IBM Developer Works: Mainframe, Performance, Topics**
  - Hosts: Marna Walle, Martin Packer
  - https://developer.ibm.com/tv/category/mpt/

- **Terminal Talk**
  - Hosts: Frank DeGillio, Jeff Bisti
  - http://terminaltalk.net/

Shameless Plug #3: The Enterprise Knights of IBM Z

- **The Enterprise Knights of IBM Z have produced a series of short videos that provide educational insights to the security and integrity of the IBM Z platform**

- **Videos is short (less than ten minutes each) and cover a range of topics:**
  - IBM Security Portal
  - CVSS Scores
  - AT-TLS
  - Authorized QNAMES
  - ETDEF
  - SSL/TLS Cipher Lists
  - Buffer Overflows
  - Untrusted Indirect Parameters
  - zERT
  - Untrusted Registers for PCs/SVCs
  - Asymmetric Encryption
  - The RACF_SENSITIVE_RESOURCES Health Check

Enterprise Knights of IBM Z

Providing educational insights to the security & integrity of our platform.

Start exploring the **Enterprise Knights of IBM Z** video library.

➜ Browse Videos

Find out about the **knights behind the videos**.

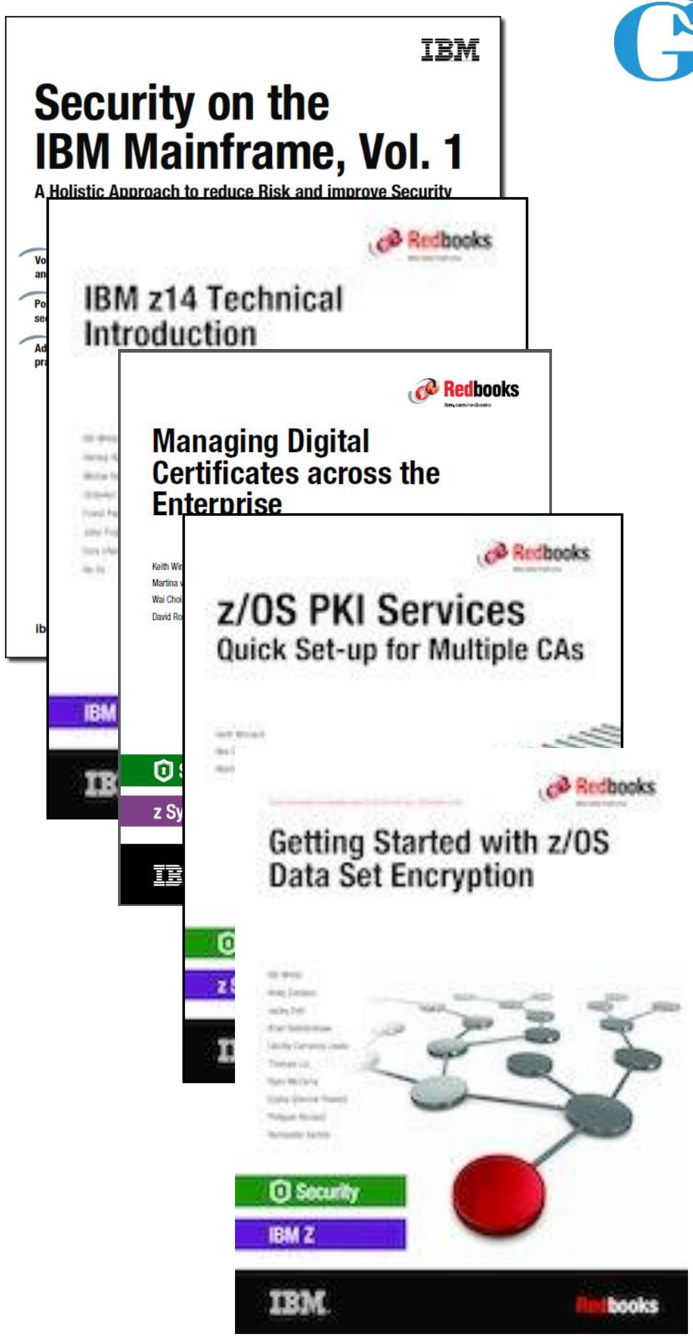
➜ View the knights

Thanks to **developerWorks TV** for hosting our video curriculum.
See a variety of other cool topics here: https://developer.ibm.com/tv/topics/

- **Available at** www.ibm.biz/ek-ibmz

Shameless Plug #4: Redbooks

- **Security on the IBM Mainframe**
- **IBM z14 Technical Introduction**
- **Managing Digital Certificates across the Enterprise**
- **z/OS PKI Services: Quick Set-up for Multiple CAs**
- **Getting Started with z/OS Data Set Encryption**



IBM

Security on the IBM Mainframe, Vol. 1

A Holistic Approach to reduce Risk and improve Security

Redbooks

IBM z14 Technical Introduction

Redbooks

Managing Digital Certificates across the Enterprise

Redbooks

z/OS PKI Services

Quick Set-up for Multiple CAs

Redbooks

Getting Started with z/OS Data Set Encryption

Security

IBM Z

IBM

Redbooks

Shameless Plug #5: zPet Test Community and Blog

- **IBM Z Platform Evaluation Test Community and Blog**
  - Real-world experiences configuring and operating the latest IBM Z technologies
  - http://ibm.biz/zPETBlog

LCST/e System z Platform Evaluation Test

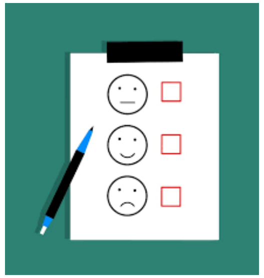
The Final Verification

z/OS | CICS | IMS | DB2 | WebSphere MQ | WebSphere Application Server | Tivoli | InfoSphere

We are a team of system programmers and testers that run a Parallel Sysplex on which we perform the final verification of a z/OS release and System z hardware and System Storage before they become generally available to clients. We gather our experiences and recommendations and document them here in our blog.

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/FA
- Paper feedback forms are also available from the Chair person
- This session is FA



z/OS® Security and RACF® Update

Session FA
Mark Nelson, CISSP®, CSSLP®
RACF Design and Development, IBM®

markan@us.ibm.com