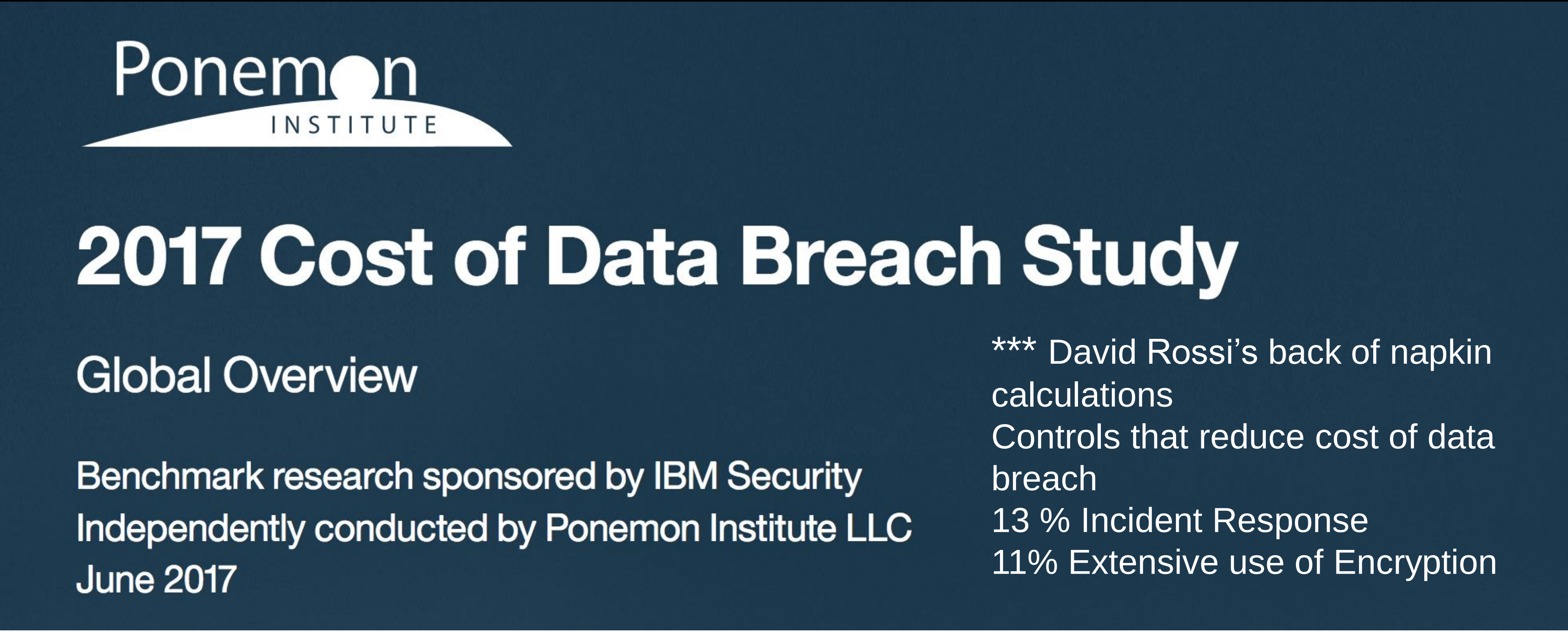
New York/Tampa/Raleigh RACF User Groups Meeting

May 9,2018

Not Your Gym Teacher's PE

Implementation of Pervasive Encryption

- □ Sue Marcotte
- □ Phil Peters
- □ Dave Rossi



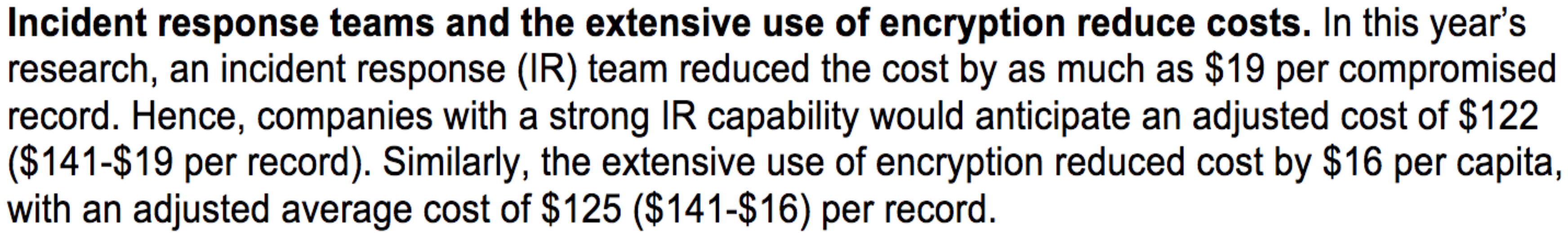
Ponemon INSTITUTE

2017 Cost of Data Breach Study

Global Overview

Benchmark research sponsored by IBM Security
Independently conducted by Ponemon Institute LLC
June 2017

*** David Rossi’s back of napkin calculations
Controls that reduce cost of data breach
13 % Incident Response
11% Extensive use of Encryption



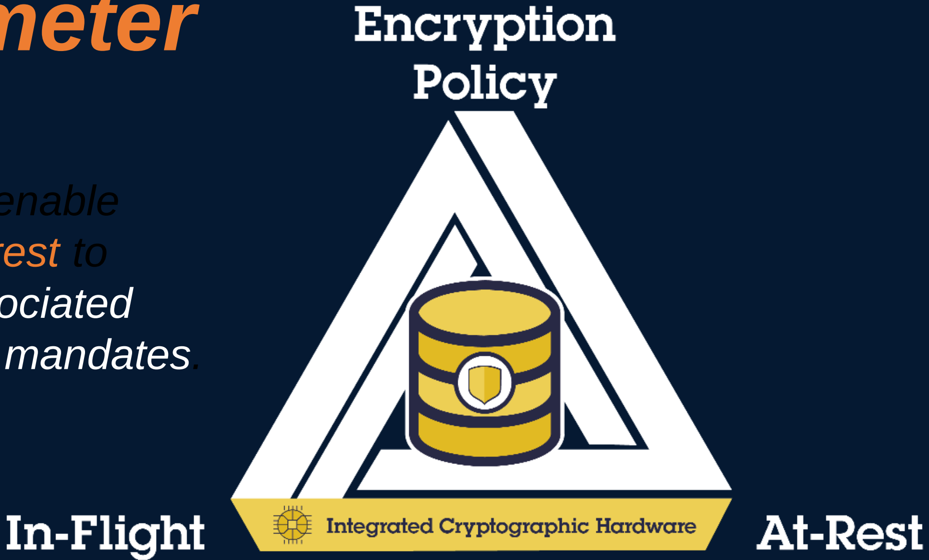
Incident response teams and the extensive use of encryption reduce costs. In this year’s research, an incident response (IR) team reduced the cost by as much as $19 per compromised record. Hence, companies with a strong IR capability would anticipate an adjusted cost of $122 ($141-$19 per record). Similarly, the extensive use of encryption reduced cost by $16 per capita, with an adjusted average cost of $125 ($141-$16) per record.

IBM Z 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.



Pervasive Encryption with IBM Z

Enabled through tight platform integration

Full Disk Encryption

Full disk encryption utilizes encrypting disk drives that protect data at rest when disk drives are retired, sent for repair or repurposed

Integrated Crypto Hardware

Hardware accelerated encryption on every core – CPACF

PCIe Hardware Security Module (HSM) & Cryptographic Coprocessor – Crypto Express5S

Network Encryption

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

Data Set & File Encryption

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

Coupling Facility

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

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

[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

* All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

zPET Environment

- Data sharing Parallel Sysplex on z Systems platform
- Latest z/OS release
- Customer like applications
- Concept: If two software products run on the same operating system platform, they should be tested together w/ a focus on their interactions.
- Ensure z/OS elements and features work seamlessly together and support true production, mission-critical work.
- Verify z/OS provides the industrial-strength z/OS advantages: reliability, availability and serviceability
- Focus on availability of applications to end users, pay attention to performance objectives.
- Look at recovery aspects and behavior of our systems from an end user’s perspective.

Net: zPET runs customer like workloads interacting w/ components across the Z software platform running on latest z Systems in data sharing Parallel Sysplexes.

Data Set Encryption: Planning

Pervasive Encryption Setup

Pervasive Encryption

Step 1: Configure Crypto Express Cards

Step 2: Configure ICSF

Step 3: Start ICSF

Step 4: Load AES MK

Step 5: Initialize CKDS

z/OS Dataset Encryption

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: Write & Print the Encrypted Data Set

Introduction to Key Management

IBM Crypto Education Community: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/W7df80301055d_495b_bb88_a0a2f84757c5/page/Pervasive%20Encryption%20-%20zOS%20Data%20Set%20Encryption

Roles

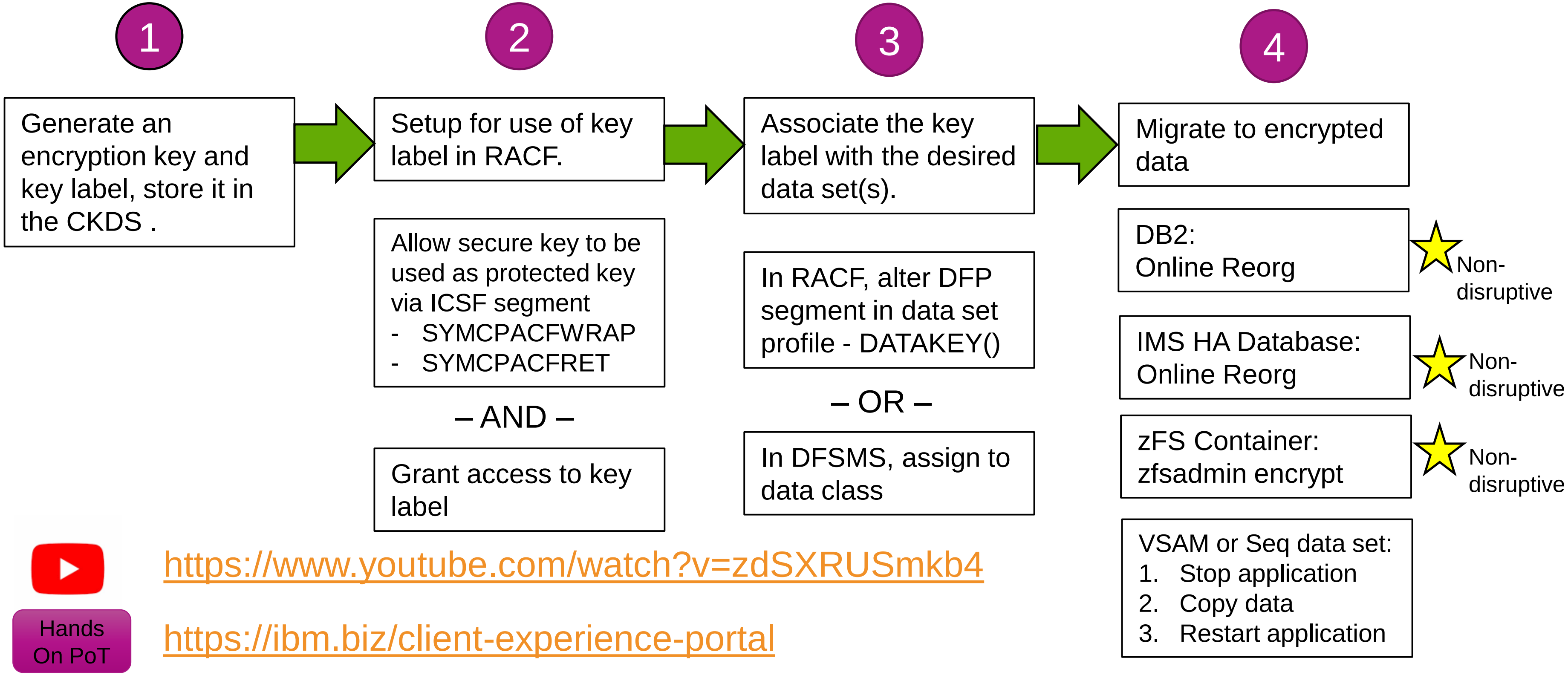
ICSF Admin
Responsible for key management (defining keys, key labels, etc), working with key mgmt system; Manages ICSF, key changes, etc;

Security Admin
Provide encryption capabilities via RACF DS profile
Responsible for creating RACF profiles, assigning access to key labels, etc

Storage Admin
Provide encryption capabilities via storage management policies (updating data classes, updating ACS routines, etc)
Manage backup, migration and replication of encrypted data sets

Data Owner/User
Runs applications, submits jobs, etc

z/OS data set encryption – High Level Steps



1

Generate an encryption key and key label, store it in the CKDS .

2

Setup for use of key label in RACF.

Allow secure key to be used as protected key via ICSF segment
- SYMCPACFWRAP
- SYMCPACFRET

– AND –

Grant access to key label

3

Associate the key label with the desired data set(s).

In RACF, alter DFP segment in data set profile - DATAKEY()

– OR –

In DFSMS, assign to data class

4

Migrate to encrypted data

DB2:
Online Reorg

Non-disruptive

IMS HA Database:
Online Reorg

Non-disruptive

zFS Container:
zfsadmin encrypt

Non-disruptive

VSAM or Seq data set:
1. Stop application
2. Copy data
3. Restart application

Hands On PoT

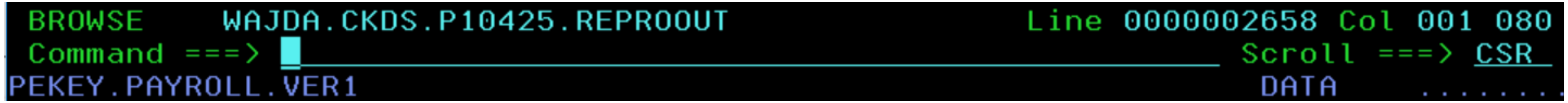
https://www.youtube.com/watch?v=zdSXRUSmkb4

https://ibm.biz/client-experience-portal

Data Set Encryption: Implementation

Generate an Encryption Key

- Use ICSF services to create the key labels and data keys. Various ways to accomplish this.
    - Exec: CSNBKGN (AES 256 bit variables) , CSNBKRC, CSNBKRW
    - Exec: CSNBKGN (AES 256 bit variables), CSNBKRC2
    - Through ICSF panel (HCR77C1 and above):
        - Option 5.5 ICSF - CKDS KEYS panel
            - Option 7 Generate AES DATA keys
- To Verify key PEKEY.PAYROLL.VER1 is indeed created and in the CKDS:
    - Issue a REPROOUT of the CKDS:

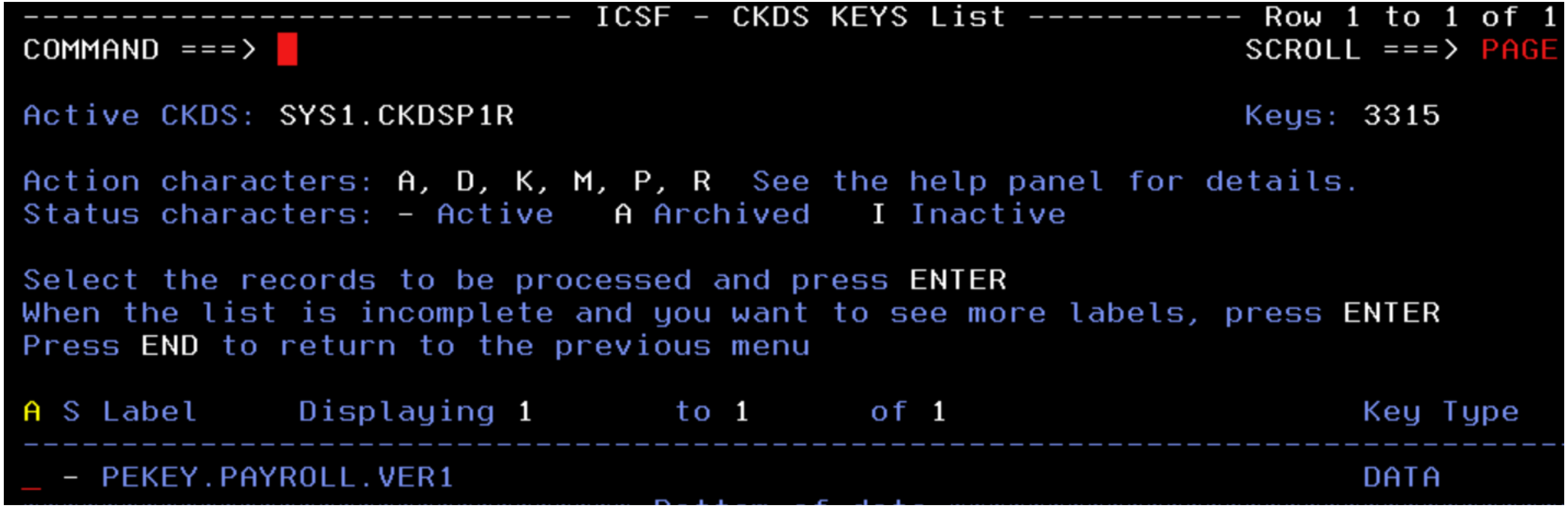


```
 BROWSE    WAJDA.CKDS.P10425.REPROOUT                    Line 0000002658 Col 001 080
 Command ===>                                                      Scroll ===> CSR
PEKEY.PAYROLL.VER1                                                  DATA     ........
```

NOTE1: ICSF HCR77C1 supplies support for a CKDS browser where keys can be displayed and created.

Generate an Encryption Key

- To Verify key PEKEY.PAYROLL.VER1 is indeed created and in the CKDS: (continued)
    - Through ICSF panel (HCR77C1 and above):
        - Option 5.5 ICSF - CKDS KEYS panel
            - Options 1, 2 or 3



```
---------------------------- ICSF - CKDS KEYS List ------------ Row 1 to 1 of 1
COMMAND ===>                                                  SCROLL ===> PAGE

Active CKDS: SYS1.CKDSP1R                                     Keys: 3315

Action characters: A, D, K, M, P, R  See the help panel for details.
Status characters: - Active   A Archived   I Inactive

Select the records to be processed and press ENTER
When the list is incomplete and you want to see more labels, press ENTER
Press END to return to the previous menu

A S Label     Displaying 1        to 1       of 1                  Key Type
-------------------------------------------------------------------------------
_ - PEKEY.PAYROLL.VER1                                             DATA
```

NOTE1: ICSF HCR77C1 supplies support for a CKDS browser where keys can be displayed and created.

Defining the Key to RACF

```
===> RDEF CSFKEYS PEKEY.PAYROLL.VER1 OWNER(SECADM)   UACC(NONE)
ICSF(SYMCPACFWRAP(YES) SYMCPACFRET(YES))
```

Permitting User to the Key

```
===> PE PEKEY.PAYROLL.VER1 CL(CSFKEYS) ID(PAYID1) ACC(READ)
WHEN(CRITERIA(SMS(DSENCRYPTION)))
```

Permitting User to CSF Service CSFKRR2

```
===> PE CSFKRR2 CL(CSFSERV) ID(PAYID1) ACC(READ)
```

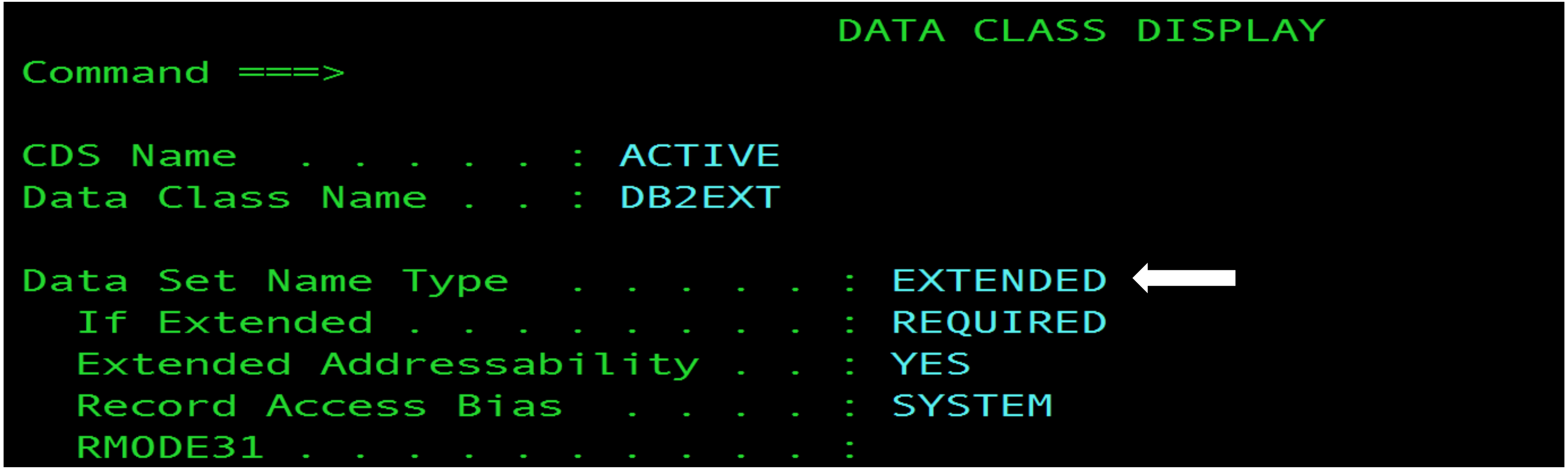
This access is needed because we have CSFSERV profile CSFKRR2 defined and have CHECKAUTH(YES) specified in SYS1.PARMLIB(CSFPARMxx)

Defining Data Set Encryption Policy

```
===> ALTDSD ‘PAYROLL.**' DFP(RESOWNER(PAYROLL)
DATAKEY(PEKEY.PAYROLL.VER1))
```

DFSMS Setup

- Storage Manager updates ACS routines to assign the hilevel qualifier to a data class.
- In order to do PE Data Set encryption, Data Set Name Type must be EXTENDED



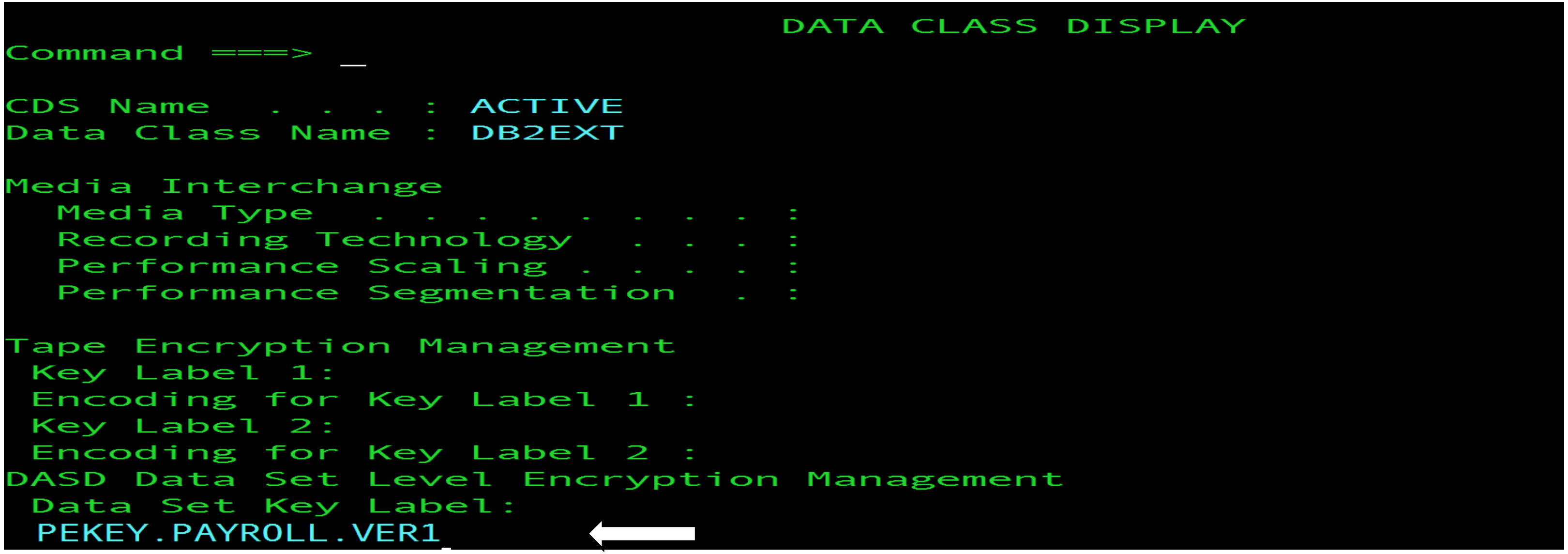
```
                              DATA CLASS DISPLAY
Command ===>

CDS Name . . . . . : ACTIVE
Data Class Name . . : DB2EXT

Data Set Name Type . . . . . . : EXTENDED
  If Extended . . . . . . . . : REQUIRED
  Extended Addressability . . : YES
  Record Access Bias . . . . . : SYSTEM
  RMODE31 . . . . . . . . . . :
```

User’s Role

The user does not have to make any changes to the JCL that creates new data sets. As long as the Data Key is specified in the DFP segment of the data set profile and the appropriate access has been granted, the new extended-format datasets will be encrypted.

Alternative – Data Key in SMS Data Class

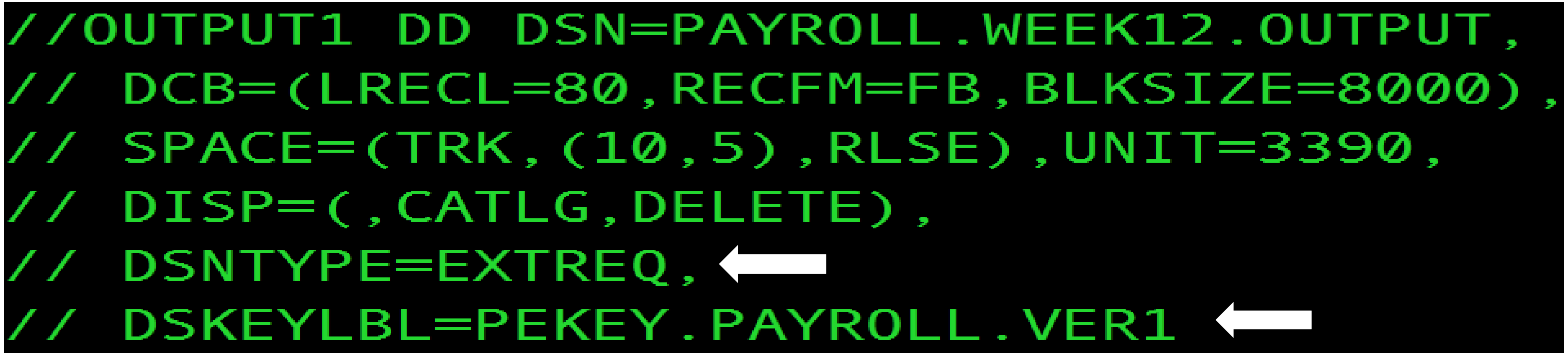
The Data Key can be specified in the SMS Data Class rather than the DFP Segment. This requires the user to have READ access to FACILITY profile STGADMIN.SMS.ALLOW.DATASET.ENCRYPT



```
                              DATA CLASS DISPLAY
Command ===> _

CDS Name    . . . : ACTIVE
Data Class Name : DB2EXT

Media Interchange
  Media Type  . . . . . . . . :
  Recording Technology  . . . :
  Performance Scaling . . . . :
  Performance Segmentation  . :

Tape Encryption Management
 Key Label 1:
 Encoding for Key Label 1 :
 Key Label 2:
 Encoding for Key Label 2 :
DASD Data Set Level Encryption Management
 Data Set Key Label:
 PEKEY.PAYROLL.VER1_
```

Another Alternative – Specify Data Key in JCL

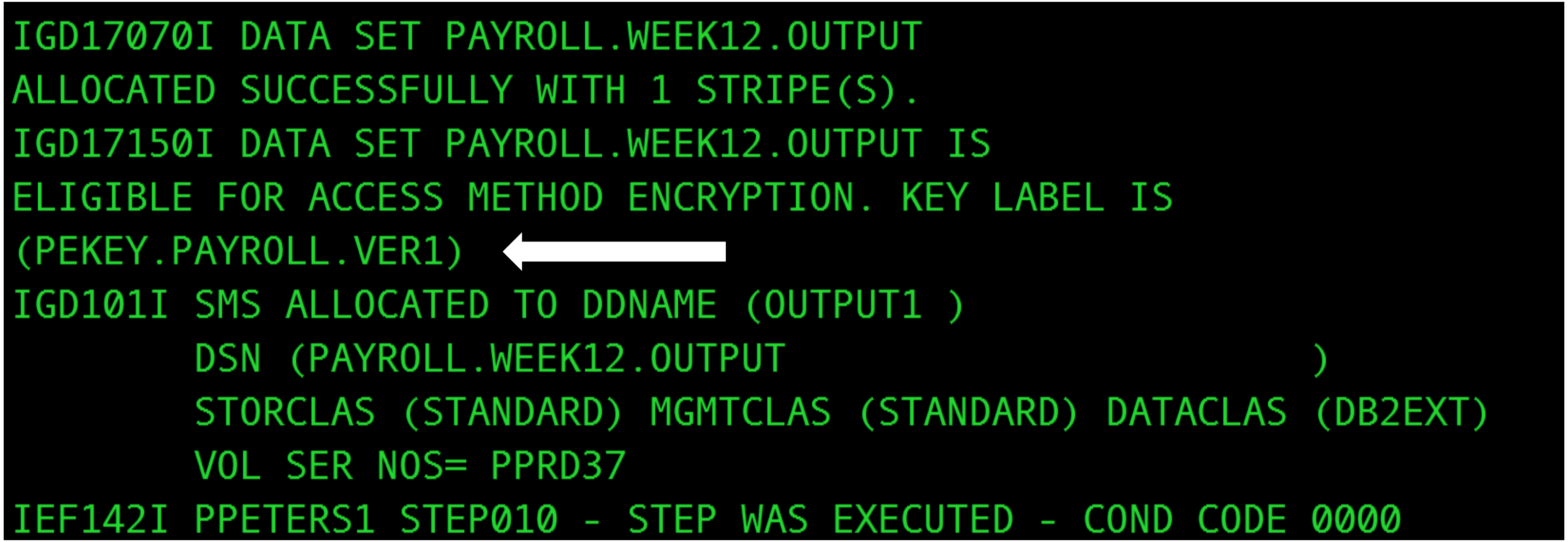
- The user can code DSKEYLBL=<key-label> for the new data set
- If their SMS data class doesn’t specify Data Set Name Type=Extended, they can code DSNTYPE=EXTREQ in their JCL



```
//OUTPUT1 DD DSN=PAYROLL.WEEK12.OUTPUT,
// DCB=(LRECL=80,RECFM=FB,BLKSIZE=8000),
// SPACE=(TRK,(10,5),RLSE),UNIT=3390,
// DISP=(,CATLG,DELETE),
// DSNTYPE=EXTREQ,
// DSKEYLBL=PEKEY.PAYROLL.VER1
```

Data Set Encryption: Indicators

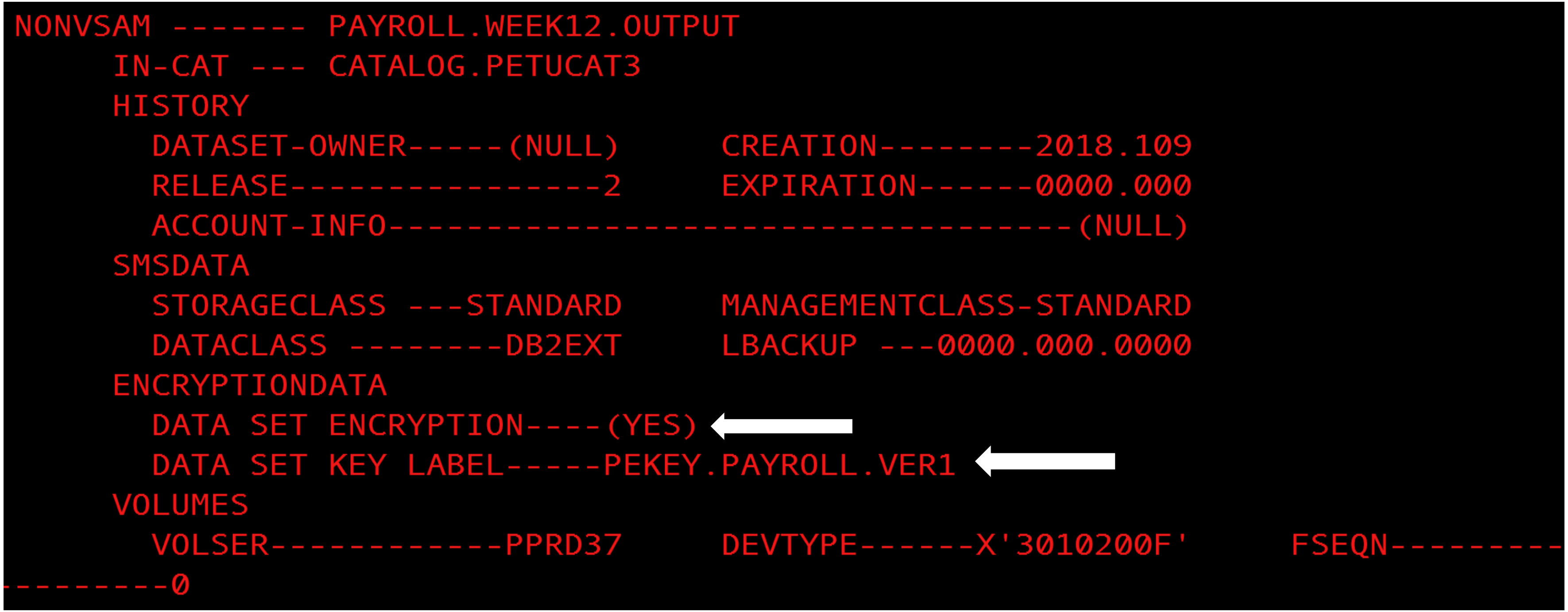
Job Output



```
IGD17070I DATA SET PAYROLL.WEEK12.OUTPUT
ALLOCATED SUCCESSFULLY WITH 1 STRIPE(S).
IGD17150I DATA SET PAYROLL.WEEK12.OUTPUT IS
ELIGIBLE FOR ACCESS METHOD ENCRYPTION. KEY LABEL IS
(PEKEY.PAYROLL.VER1)
IGD101I SMS ALLOCATED TO DDNAME (OUTPUT1 )
        DSN (PAYROLL.WEEK12.OUTPUT                       )
        STORCLAS (STANDARD) MGMTCLAS (STANDARD) DATACLAS (DB2EXT)
        VOL SER NOS= PPRD37
IEF142I PPETERS1 STEP010 - STEP WAS EXECUTED - COND CODE 0000
```

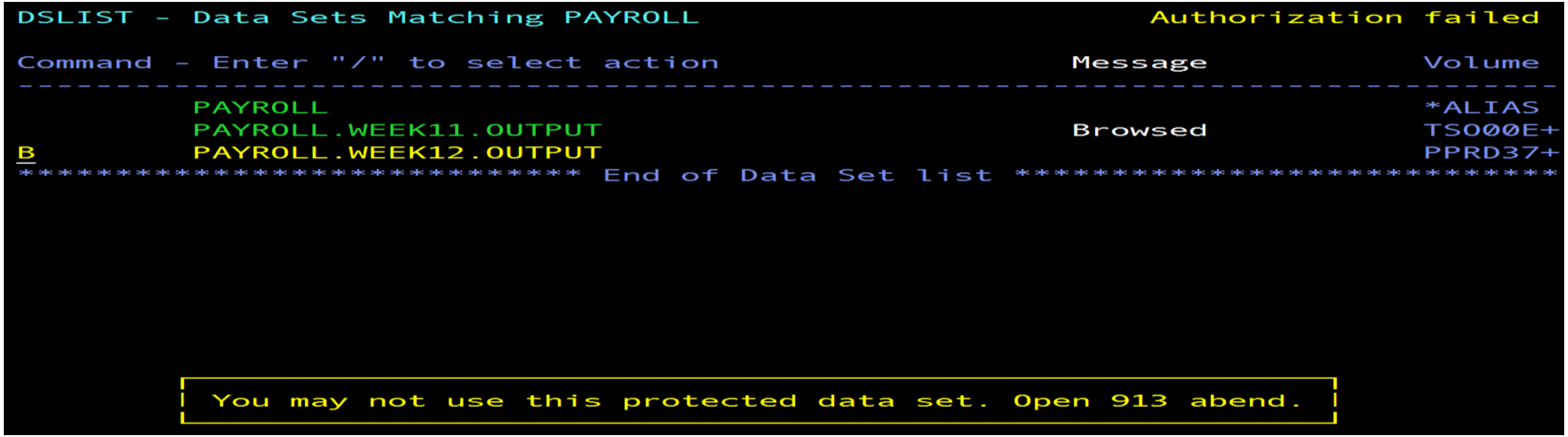
LISTCAT Command

listc ent(‘PAYROLL.WEEK12.OUTPUT’) all



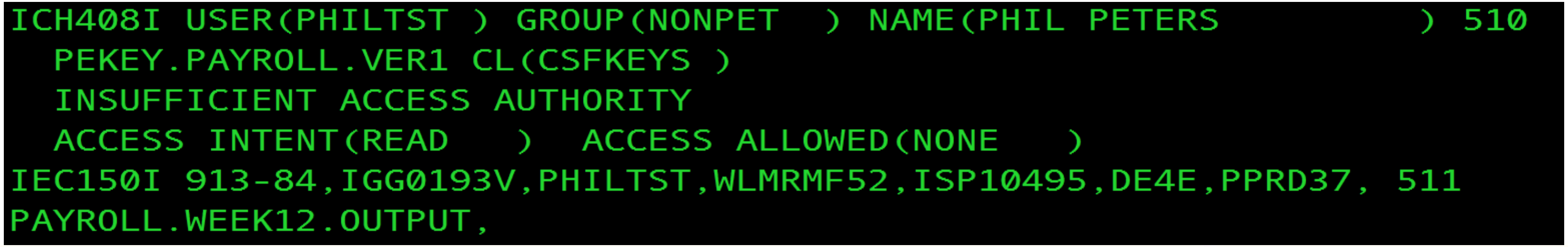
```
NONVSAM ------- PAYROLL.WEEK12.OUTPUT
     IN-CAT --- CATALOG.PETUCAT3
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2018.109
       RELEASE----------------2     EXPIRATION------0000.000
       ACCOUNT-INFO-----------------------------------(NULL)
     SMSDATA
       STORAGECLASS ---STANDARD     MANAGEMENTCLASS-STANDARD
       DATACLASS --------DB2EXT     LBACKUP ---0000.000.0000
     ENCRYPTIONDATA
       DATA SET ENCRYPTION----(YES)
       DATA SET KEY LABEL-----PEKEY.PAYROLL.VER1
     VOLUMES
       VOLSER------------PPRD37     DEVTYPE------X'3010200F'     FSEQN---------
---------0
```

Insufficient Access to Key



```
DSLIST - Data Sets Matching PAYROLL                             Authorization failed

Command - Enter "/" to select action                         Message           Volume
-------------------------------------------------------------------------------
         PAYROLL                                                               *ALIAS
         PAYROLL.WEEK11.OUTPUT                               Browsed           TS000E+
B        PAYROLL.WEEK12.OUTPUT                                                 PPRD37+
***************************** End of Data Set list ****************************

         You may not use this protected data set. Open 913 abend.
```

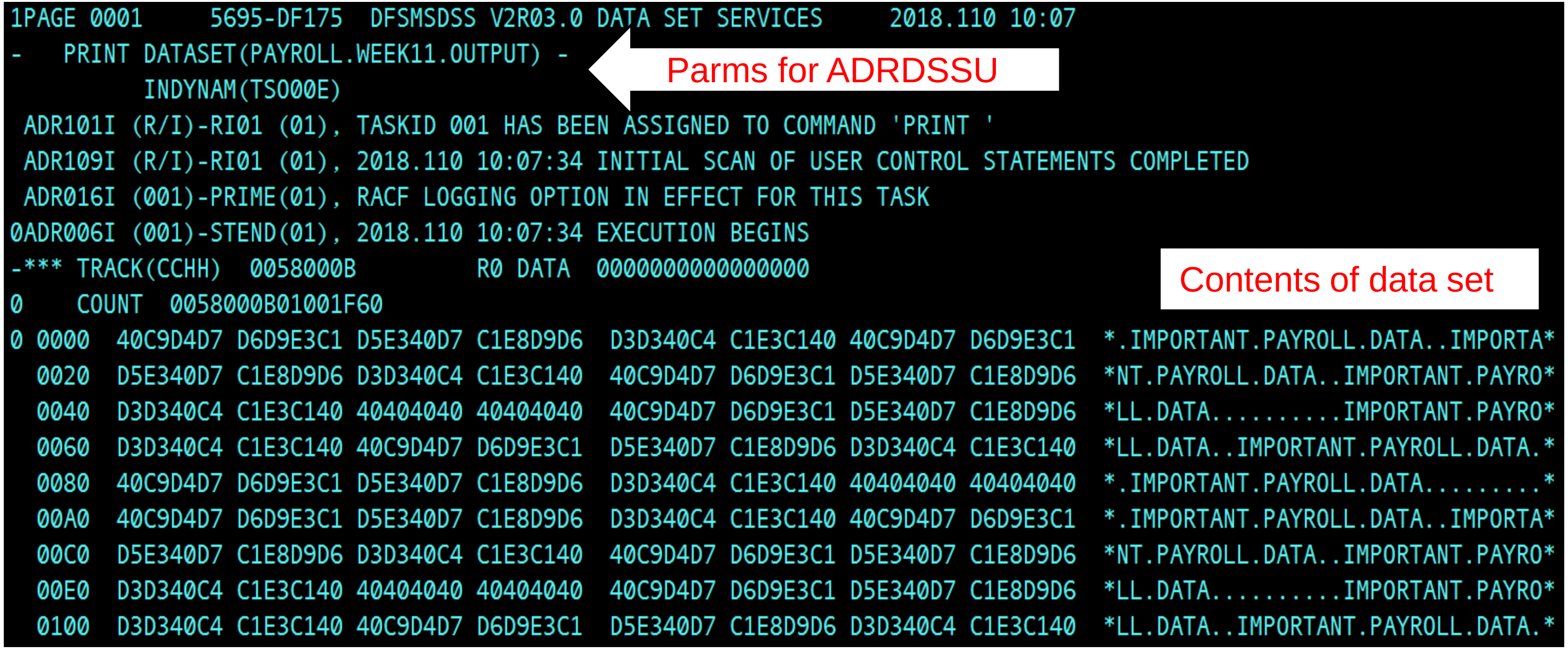
In SYSLOG:



```
ICH408I USER(PHILTST ) GROUP(NONPET   ) NAME(PHIL PETERS        ) 510
  PEKEY.PAYROLL.VER1 CL(CSFKEYS )
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
IEC150I 913-84,IGG0193V,PHILTST,WLMRMF52,ISP10495,DE4E,PPRD37, 511
PAYROLL.WEEK12.OUTPUT,
```

DSSPRINT

ADRDSSU PRINT against unencrypted data set



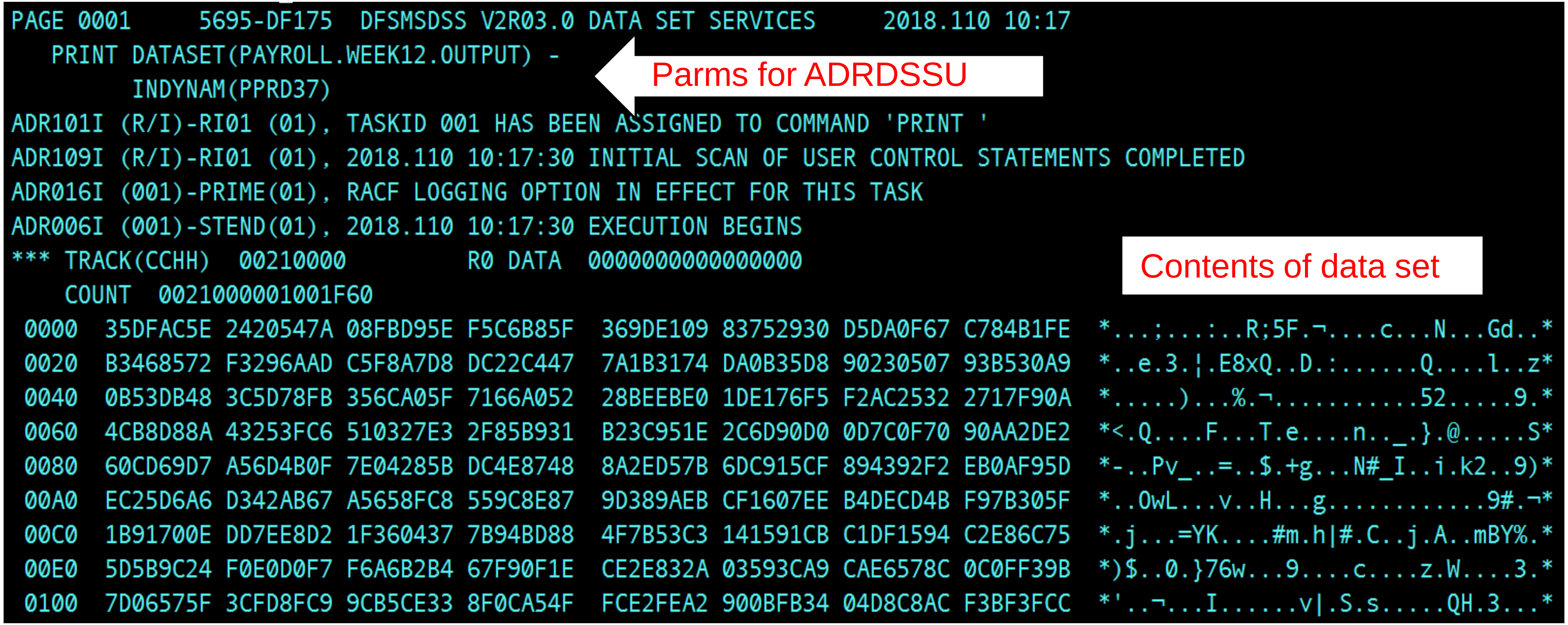
```
1PAGE 0001     5695-DF175  DFSMSDSS V2R03.0 DATA SET SERVICES     2018.110 10:07
-   PRINT DATASET(PAYROLL.WEEK11.OUTPUT) -
            INDYNAM(TS000E)
 ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PRINT '
 ADR109I (R/I)-RI01 (01), 2018.110 10:07:34 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
 ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
0ADR006I (001)-STEND(01), 2018.110 10:07:34 EXECUTION BEGINS
-*** TRACK(CCHH)  0058000B        R0 DATA  0000000000000000
0    COUNT  0058000B01001F60
0 0000  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  D3D340C4 C1E3C140 40C9D4D7 D6D9E3C1  *.IMPORTANT.PAYROLL.DATA..IMPORTA*
  0020  D5E340D7 C1E8D9D6 D3D340C4 C1E3C140  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  *NT.PAYROLL.DATA..IMPORTANT.PAYRO*
  0040  D3D340C4 C1E3C140 40404040 40404040  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  *LL.DATA..........IMPORTANT.PAYRO*
  0060  D3D340C4 C1E3C140 40C9D4D7 D6D9E3C1  D5E340D7 C1E8D9D6 D3D340C4 C1E3C140  *LL.DATA..IMPORTANT.PAYROLL.DATA.*
  0080  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  D3D340C4 C1E3C140 40404040 40404040  *.IMPORTANT.PAYROLL.DATA.........*
  00A0  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  D3D340C4 C1E3C140 40C9D4D7 D6D9E3C1  *.IMPORTANT.PAYROLL.DATA..IMPORTA*
  00C0  D5E340D7 C1E8D9D6 D3D340C4 C1E3C140  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  *NT.PAYROLL.DATA..IMPORTANT.PAYRO*
  00E0  D3D340C4 C1E3C140 40404040 40404040  40C9D4D7 D6D9E3C1 D5E340D7 C1E8D9D6  *LL.DATA..........IMPORTANT.PAYRO*
  0100  D3D340C4 C1E3C140 40C9D4D7 D6D9E3C1  D5E340D7 C1E8D9D6 D3D340C4 C1E3C140  *LL.DATA..IMPORTANT.PAYROLL.DATA.*
```

Parms for ADRDSSU

Contents of data set

DSSPRINT

ADRDSSU PRINT against encrypted data set



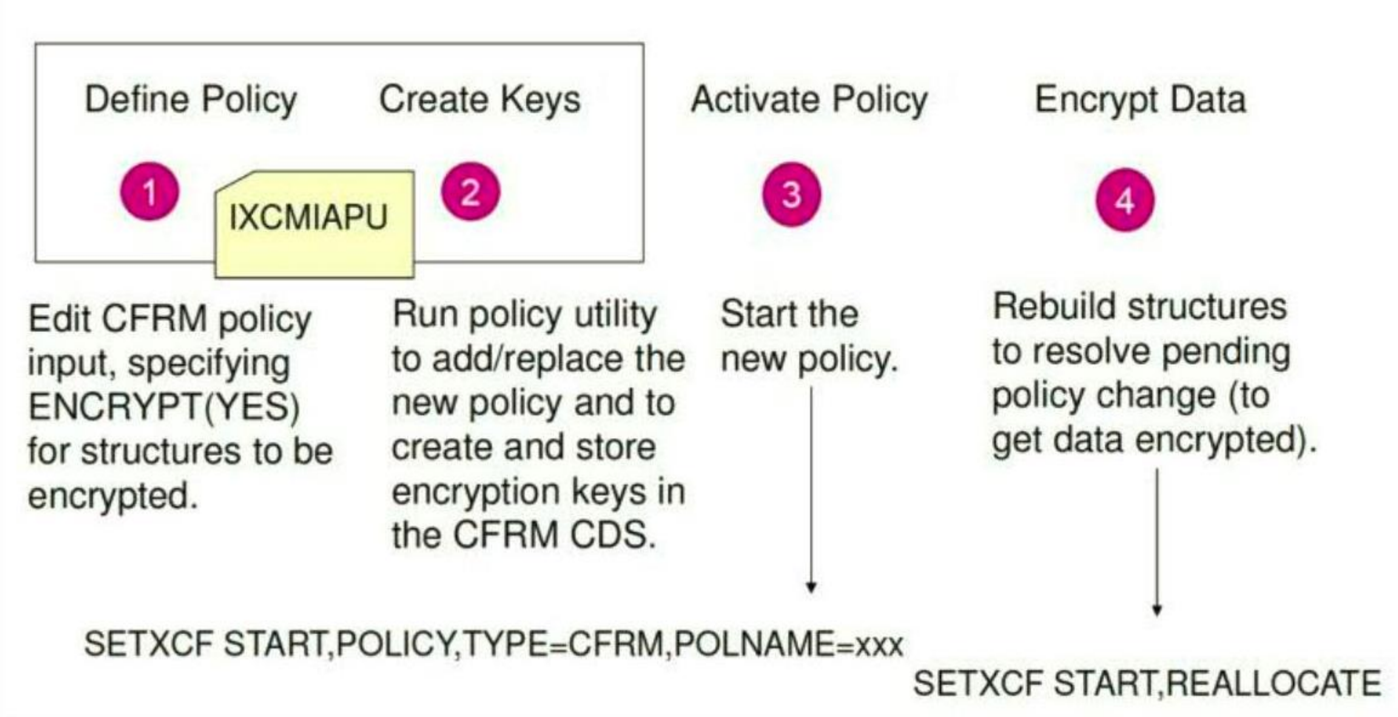
```
PAGE 0001     5695-DF175  DFSMSDSS V2R03.0 DATA SET SERVICES     2018.110 10:17
   PRINT DATASET(PAYROLL.WEEK12.OUTPUT) -
         INDYNAM(PPRD37)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'PRINT '
ADR109I (R/I)-RI01 (01), 2018.110 10:17:30 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 2018.110 10:17:30 EXECUTION BEGINS
*** TRACK(CCHH)  00210000         R0 DATA  0000000000000000
    COUNT  0021000001001F60
 0000  35DFAC5E 2420547A 08FBD95E F5C6B85F  369DE109 83752930 D5DA0F67 C784B1FE  *...;...:..R;5F.¬....c...N...Gd..*
 0020  B3468572 F3296AAD C5F8A7D8 DC22C447  7A1B3174 DA0B35D8 90230507 93B530A9  *..e.3.¦.E8xQ..D.:......Q....l..z*
 0040  0B53DB48 3C5D78FB 356CA05F 7166A052  28BEEBE0 1DE176F5 F2AC2532 2717F90A  *.....)...%.¬..........52.....9.*
 0060  4CB8D88A 43253FC6 510327E3 2F85B931  B23C951E 2C6D90D0 0D7C0F70 90AA2DE2  *<.Q....F...T.e....n.._.}.@.....S*
 0080  60CD69D7 A56D4B0F 7E04285B DC4E8748  8A2ED57B 6DC915CF 894392F2 EB0AF95D  *-..Pv_..=..$.+g...N#_I..i.k2..9)*
 00A0  EC25D6A6 D342AB67 A5658FC8 559C8E87  9D389AEB CF1607EE B4DECD4B F97B305F  *..OwL...v..H...g...........9#.¬*
 00C0  1B91700E DD7EE8D2 1F360437 7B94BD88  4F7B53C3 141591CB C1DF1594 C2E86C75  *.j...=YK....#m.h|#.C..j.A..mBY%.*
 00E0  5D5B9C24 F0E0D0F7 F6A6B2B4 67F90F1E  CE2E832A 03593CA9 CAE6578C 0C0FF39B  *)$..0.}76w...9....c....z.W....3.*
 0100  7D06575F 3CFD8FC9 9CB5CE33 8F0CA54F  FCE2FEA2 900BFB34 04D8C8AC F3BF3FCC  *'..¬...I......v|.S.s.....QH.3...*
```

Parms for ADRDSSU

Contents of data set

Coupling Facility Encryption

CF Encryption



1. Edit CFRM policy input, specifying ENCRYPT(YES) for structures to be encrypted.
2. Run policy utility to add/replace the new policy and to create and store encryption keys in the CFRM CDS.
3. Start the new policy.
4. Rebuild structures to resolve pending policy change (to get data encrypted).

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=xxx
```

```
SETXCF START,REALLOCATE
```

CF Encryption

- Encryption enabled via the new ENCRYPT structure keyword in CFRM policy definition.
- Administration Data Utility, IXCMIAPU, **creates** and assigns secure cryptographic key tokens to a structure whose CFRM policy specifies ENCRYPT(YES).
    - Edit CFRM policy specifying ENCRYPT(YES)
    - Run policy utility, IXCMIAPU, to add/replace new policy and create/store keys in CFRM CDS
    - Start new policy
    - Rebuild structures to resolve pending policy change to encrypt data.
- Structures
    - CF List and Cache structures can contain customer data and can therefore be encrypted.
    - Lock structures as well as Directory Only Cache structures do not contain customer data and therefore will not allow encryption.

zERT

Overview: z/OS Encryption Readiness Technology (zERT – 1 of 2)

- zERT positions the TCP/IP stack as a central collection point and repository for cryptographic protection attributes for:
  - **TCP** connections that are protected by **TLS, SSL, SSH, IPsec or** are **unprotected**
  - **Enterprise Extender** connections that are protected by **IPsec or** are **unprotected**
    - Each peer-to-peer UDP port is considered a separate EE connection
    - In this presentation, we'll focus on TCP examples
- Two methods for discovering the security sessions and their attributes:
  - Stream observation (for TLS, SSL and SSH) – the TCP/IP stack observes the protocol handshakes as they flow over the TCP connection
  - Advice of the cryptographic protocol provider (System SSL, OpenSSH, TCP/IP's IPsec support)
- Reported through new SMF 119 records via:
  - SMF and/or
  - New real-time NMI services

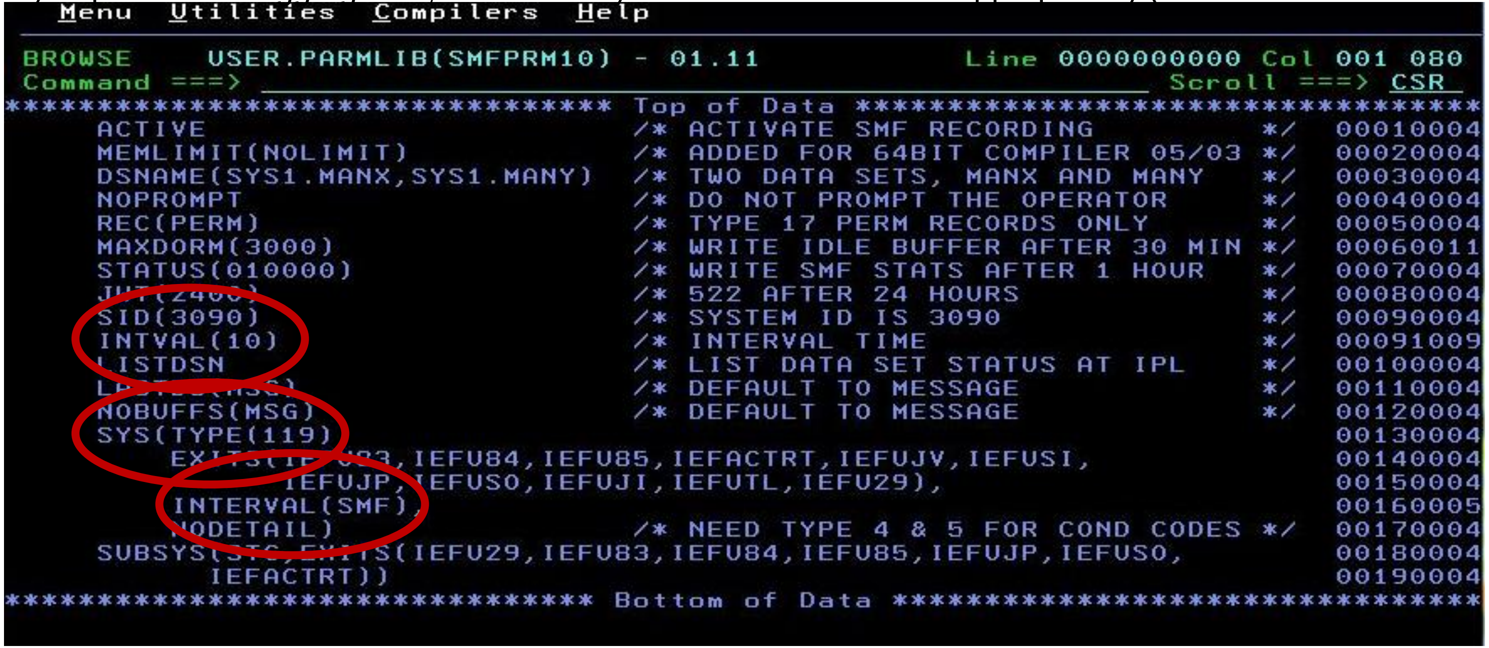
Overview: z/OS Encryption Readiness Technology (zERT – 2 of 2)

- zERT **Discovery – available in V2R3**
  - Attributes are collected and recorded at the connection level
  - SMF 119 subtype 11 “zERT Connection Detail” records
  - These records **describe the cryptographic protection history of each TCP and EE connection**
  - Measures are in place to minimize the number of subtype 11 records, but they could still be very voluminous
- zERT **Aggregation – available via V2R3 new function APAR PI83362**
  - Attributes collected by zERT discovery are aggregated by security session
  - SMF 119 subtype 12 “zERT Summary” records
  - These records **describe the repeated use of security sessions over time**
  - Aggregation can greatly reduce the volume of SMF records while maintaining the fidelity of the information – well suited for reporting applications

Configuring: 1. Enable SMF 119 records in SMF (PARMLIB)

In your PARMLIB(SMFPRMxx):

- Ensure that SMF 119 records are enabled (SYS(TYPE(119)…)
- If you plan to use Aggregation, ensure that your SMF interval is set appropriately (INTVAL and INTERVAL(SMF))



```
BROWSE    USER.PARMLIB(SMFPRM10) - 01.11            Line 0000000000 Col 001 080
Command ===> ________________________________________________ Scroll ===> CSR
********************************* Top of Data **********************************
     ACTIVE                          /* ACTIVATE SMF RECORDING           */   00010004
     MEMLIMIT(NOLIMIT)               /* ADDED FOR 64BIT COMPILER 05/03   */   00020004
     DSNAME(SYS1.MANX,SYS1.MANY)     /* TWO DATA SETS, MANX AND MANY     */   00030004
     NOPROMPT                        /* DO NOT PROMPT THE OPERATOR       */   00040004
     REC(PERM)                       /* TYPE 17 PERM RECORDS ONLY        */   00050004
     MAXDORM(3000)                   /* WRITE IDLE BUFFER AFTER 30 MIN   */   00060011
     STATUS(010000)                  /* WRITE SMF STATS AFTER 1 HOUR     */   00070004
     JWT(2400)                       /* 522 AFTER 24 HOURS               */   00080004
     SID(3090)                       /* SYSTEM ID IS 3090                */   00090004
     INTVAL(10)                      /* INTERVAL TIME                    */   00091009
     LISTDSN                         /* LIST DATA SET STATUS AT IPL      */   00100004
     LASTDS(MSG)                     /* DEFAULT TO MESSAGE               */   00110004
     NOBUFFS(MSG)                    /* DEFAULT TO MESSAGE               */   00120004
     SYS(TYPE(119)                                                            00130004
         EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT,IEFUJV,IEFUSI,                   00140004
               IEFUJP,IEFUSO,IEFUJI,IEFUTL,IEFU29),                           00150004
         INTERVAL(SMF),                                                       00160005
         NODETAIL)                   /* NEED TYPE 4 & 5 FOR COND CODES   */   00170004
     SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFU85,IEFUJP,IEFUSO,              00180004
           IEFACTRT))                                                         00190004
******************************** Bottom of Data ********************************
```

Configuring: 2. Enable zERT monitoring (TCPIP profile)

In your TCPIP profile data set:

- GLOBALCONFIG ZERT controls zERT **in-memory** monitoring (default is NOZERT)
  - `GLOBALCONFIG ZERT [AGGRegation] | NOZERT`
  - `AGGRegation` subparameter enables aggregation function
- Note that the discovery and aggregation in-memory functions are enabled independently of the destinations to which records are written.
- Can be dynamically enabled or disabled
- Can be configured by hand or through the z/OSMF Configuration Assistant for z/OS Communications Server

Configuring: 3. Specify recording destinations (TCPIP profile)

In your TCPIP profile data set:

- SMFCONFIG controls writing of zERT records to System Management Facility
  - `SMFCONFIG ZERTDetail | NOZERTDetail`
  - `SMFCONFIG ZERTSUMmary | NOZERTSUMmary`
  - Defaults are NOZERTDetail and NOZERTSUMmary
- NETMONITOR controls writing of zERT records to new real-time network monitoring services
  - `NETMONITOR ZERTService | NOZERTService`
  - `NETMONITOR ZERTSUMmary | NOZERTSUMmary`
  - Defaults are NOZERTService and NOZERTSUMmary
- Note that the discovery and aggregation in-memory functions are enabled independently of the destinations to which records are written.
- Can be dynamically enabled or disabled
- Can be configured by hand or through the z/OSMF Configuration Assistant for z/OS Communications Server

zPET Experiences

zPET Experiences

- Initial start
  - Data Set Encryption on V2.2
  - ICSF HCR77C0 , setup w/ an AES Master Key, access to Crypto Express cards and CPACF.

- ICSF setup
  - If running with HCR77C0 and above, you can dynamically update the ICSF CHECKAUTH setting using the SETICSF command
  - We run workloads across multiple images with different CKDS' setup w/ the same AES Master Key. We copied the key from the one CKDS using CSNBKRR to the other CKDS using CSNBKRR2.

zPET Experiences

- CF Encryption
  - Does not require manual setup of the key. The administrative data utility creates/assigns keys to structure definitions in the CDS.
  - Enabled on a structure by structure basis using a new ENCRYPT structure keyword
  - zPET adopted a staged approach to encrypting structures: individual structures by type and exploiter, to encrypt structures for an entire data sharing group and finally encrypting all structures for all applications.
  - Encrypted structures for:
    - IBM IMS V14
    - IBM Db2 at V11 & V12
    - IBM MQ
    - IBM CICS
    - z/OS infrastructure support structures such as XCF signaling, Operlog and JES2 checkpoint
  - No issues managing and switching multiple policies containing structures w/ encrypted data, no differences in switching in or out Couple Data Sets that contain those policies, no issues transparently changing secure key tokens for structures with the new SETXCF MODIFY,STRNM=strname,ENCRYPTKEY

zPET Experiences

- IBM MQ
  - IBM MQ V8, V9 and V9.0x.
  - Along w/ CF structures, encrypted new BSDS and archive logs.

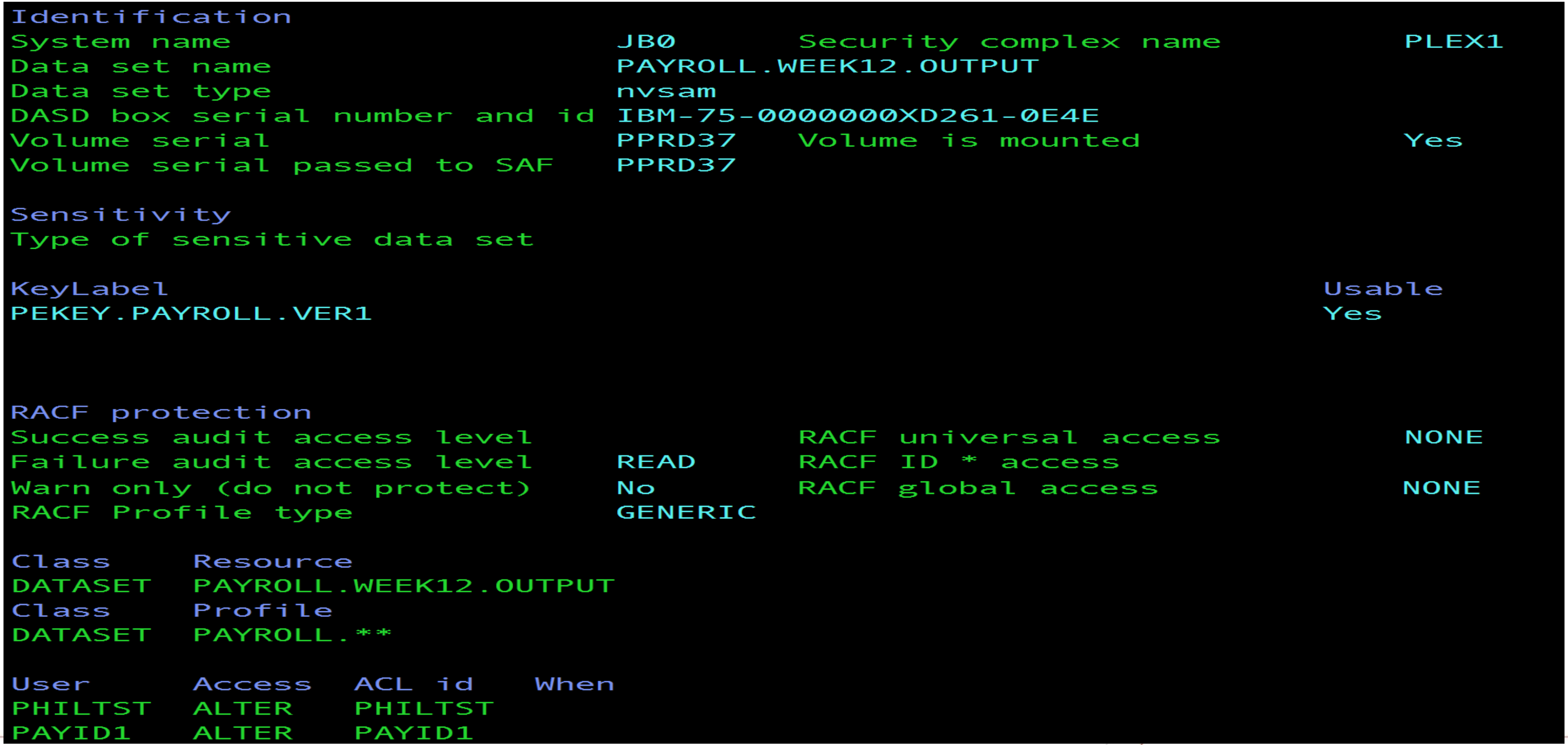
- IBM IMS
  - IMS V14
  - Along w/ CF structures, encrypted:
    - VSAM non-HALDB and HALDB databases
    - IMS online log data sets (OLDS)
    - IMS system log data sets (SLDS)
    - IMS image copy data sets
    - CQS structure recovery data sets (SRDS)
- IBM CICS
  - IBM CICS TS 5.3
  - Along w/ CF structures, encrypted:
    - VSAM RLS data sets
    - VSAM non-RLS data sets

zPET Experiences

- IBM zBNA
  - Capacity planning tool that provides both capacity planning function and the ability to evaluate a Z Server’s data sets and CF structures.
  - zPET used zBNA to identify encryption candidates on z/OS V2R3.
  - Downloadable from IBM PartnerWorld, http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS5132

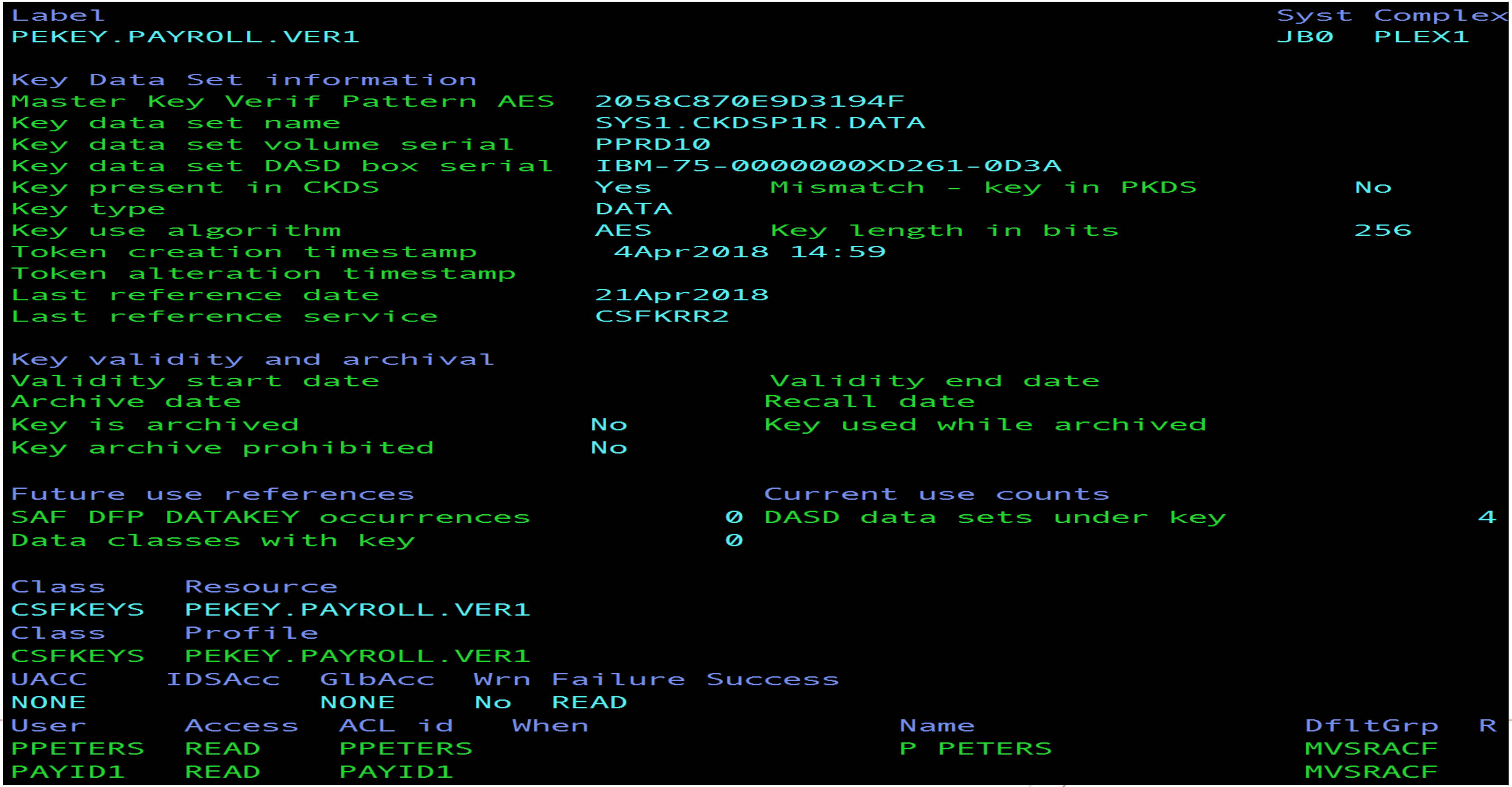
Verification Reports

zSecure Report – RE.K.DA Data sets under encryption policy or encrypted



```
Identification
System name                  JB0       Security complex name        PLEX1
Data set name                PAYROLL.WEEK12.OUTPUT
Data set type                nvsam
DASD box serial number and id IBM-75-0000000XD261-0E4E
Volume serial                PPRD37    Volume is mounted            Yes
Volume serial passed to SAF  PPRD37

Sensitivity
Type of sensitive data set

KeyLabel                                                        Usable
PEKEY.PAYROLL.VER1                                              Yes



RACF protection
Success audit access level             RACF universal access        NONE
Failure audit access level   READ      RACF ID * access
Warn only (do not protect)   No        RACF global access           NONE
RACF Profile type            GENERIC

Class     Resource
DATASET   PAYROLL.WEEK12.OUTPUT
Class     Profile
DATASET   PAYROLL.**

User      Access   ACL id   When
PHILTST   ALTER    PHILTST
PAYID1    ALTER    PAYID1
```

zSecure Report – RE.K.S Symmetric Keys



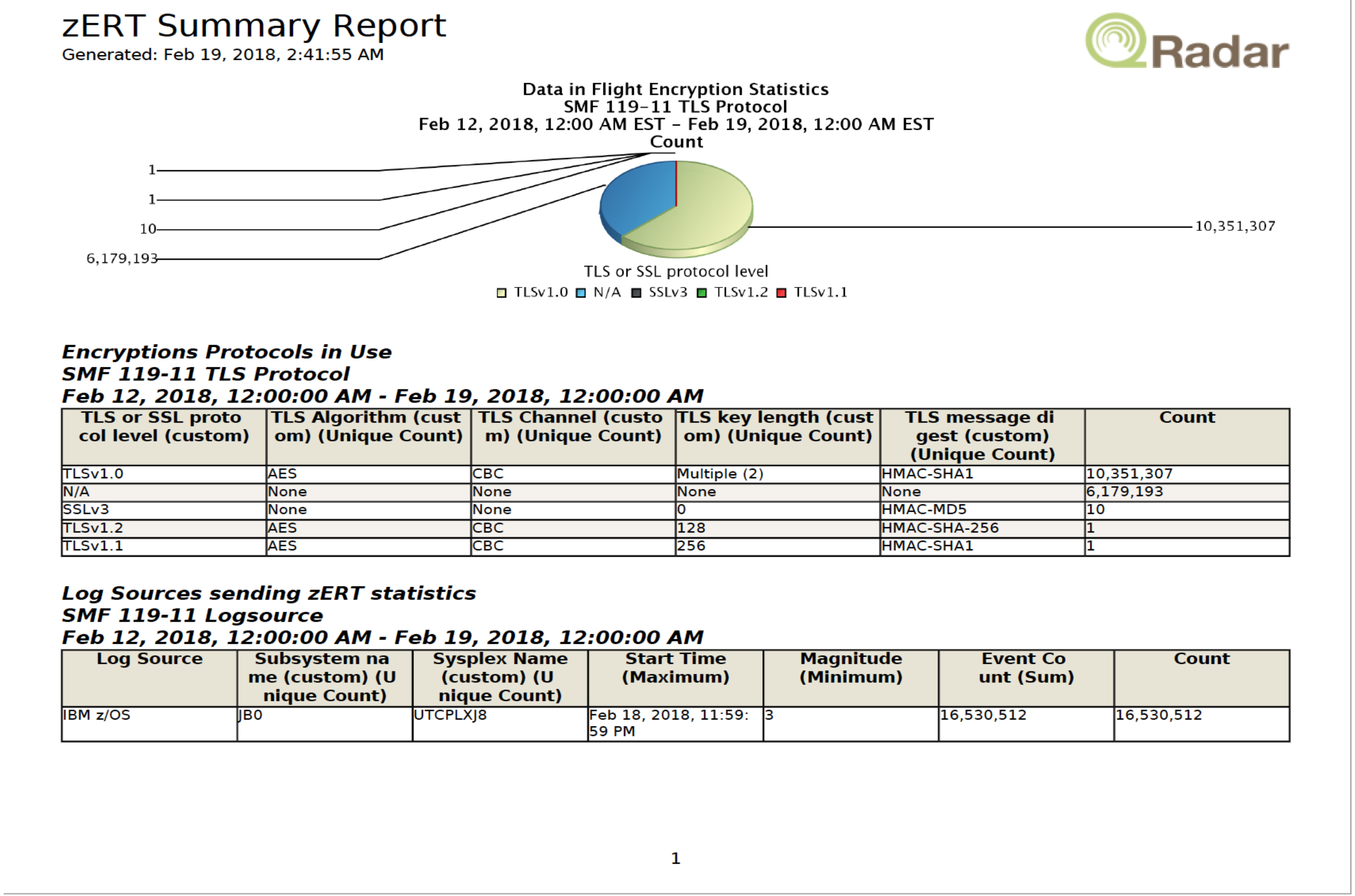
```
Label                                                            Syst Complex
PEKEY.PAYROLL.VER1                                               JB0  PLEX1

Key Data Set information
Master Key Verif Pattern AES  2058C870E9D3194F
Key data set name             SYS1.CKDSP1R.DATA
Key data set volume serial    PPRD10
Key data set DASD box serial  IBM-75-0000000XD261-0D3A
Key present in CKDS           Yes      Mismatch - key in PKDS         No
Key type                      DATA
Key use algorithm             AES      Key length in bits             256
Token creation timestamp       4Apr2018 14:59
Token alteration timestamp
Last reference date           21Apr2018
Last reference service        CSFKRR2

Key validity and archival
Validity start date                    Validity end date
Archive date                           Recall date
Key is archived               No       Key used while archived
Key archive prohibited        No

Future use references                  Current use counts
SAF DFP DATAKEY occurrences          0 DASD data sets under key             4
Data classes with key                0

Class    Resource
CSFKEYS  PEKEY.PAYROLL.VER1
Class    Profile
CSFKEYS  PEKEY.PAYROLL.VER1
UACC    IDSAcc  GlbAcc  Wrn Failure Success
NONE            NONE    No  READ
User     Access  ACL id   When                Name                 DfltGrp  R
PPETERS  READ    PPETERS                      P PETERS             MVSRACF
PAYID1   READ    PAYID1                                            MVSRACF
```

Sample SMF Records to collect.

Data Set Encryption

SMF Type	Sub-type		Required	Recommended
Record Type 14	---	INPUT or RDBACK Data Set Activity	yes	
Record Type 15	---	OUTPUT, UPDAT, INOUT, or OUTIN Data Set Activity	yes	
Record Type 30	---	Common address space work		
	1	Job start or start of other work unit	yes	
	2	Activity since previous interval ended		yes
	3	Activity for the last interval before step termination		yes
	4	Step total		yes
	5	Job termination or termination of other work unit	yes	
	6	System address space, which did not go through full function start.		yes
Record Type 42	---	DFSMS statistics and configuration		
	6	records DASD data set level I/O statistics		yes
Record Type 60	---	VSAM Volume Data Set Updated	yes	
Record Type 61	---	ICF Define Activity	yes	
Record Type 62	---	VSAM Component or Cluster Opened	yes	
Record Type 64	---	VSAM Component or Cluster Status		yes
Record Type 65	---	ICF Delete Activity	yes	
Record Type 66	---	ICF Alter Activity	yes	
Record Type 80	---	Security Product Processing	yes	
Record Type 81	---	RACF Initialization	yes for RACF	
Record type 92	---	File system activity		
	1	file system is mounted.	yes	
	2	file system is quiesced (or suspended).		yes
	4	file system is unquiesced (or resumed).		yes
	5	file system is unmounted.	yes	
	6	file system is remounted.		yes
	7	file system is moved.		yes
	10	file is opened.	yes	
	11	file is closed.	yes	
	12	MMAP subtype information.		yes
	13	MUNMAP subtype information.		yes
	14	file or file directory is deleted or renamed.	yes	
	15	file's security attributes for APF authorized, program control, or shared library are changed.	yes	
	16	socket, character special file, pipe, or fifo is closed.	yes	
	17	how many times a file is accessed throughout the life of an open and is written on the SMF global recording interval.		yes
Record Type 119	---	TCP/IP Statistics		
	1	TCP connection initiation record (subtype 1)	yes	
	2	TCP connection termination record (subtype 2)	yes	
	3	FTP client transfer completion record (subtype 3)	yes	
	4	TCP/IP profile event record (subtype 4)		yes
	5	TCP/IP statistics record (subtype 5)		yes
	6	Interface statistics record (subtype 6)		yes
	7	Server port statistics record (subtype 7)		yes
	8	TCP/IP stack start/stop record (subtype 8)	yes	
	10	UDP socket close record (subtype 10)	yes	
	11	zERT connection detail record		yes
	20	TN3270E Telnet server SNA session initiation record (subtype 20)	yes	
	21	TN3270E Telnet server SNA session termination record (subtype 21)	yes	
	22	TSO Telnet client connection initiation record (subtype 22)	yes	
	23	TSO Telnet client connection termination record (subtype 23)	yes	
	24	Telnet profile configuration		yes
	32	DVIPA status change record (subtype 32)		yes
	33	DVIPA removed record (subtype 33)		yes
	34	DVIPA target added record (subtype 34)		yes
	35	DVIPA target removed record (subtype 35)		yes
	36	DVIPA target server started record (subtype 36)		yes
	37	DVIPA target server ended record (subtype 37)		yes
	41	SMC-R link group statistics record (subtype 41)		yes
	42	SMC-R link state start record (subtype 42)		yes
	43	SMC-R link state end record (subtype 43)		yes
	44	RDMA network interface card (RNIC) interface statistics record (subtype 44)		yes
	48	CSSMTP configuration record (CONFIG subtype 48)		yes
	49	CSSMTP connection record (CONNECT subtype 49)	yes	
	50	CSSMTP mail record (MAIL subtype 50)	yes	
	51	CSSMTP spool file record (SPOOL subtype 51)	yes	
	52	CSSMTP statistical record (STATS subtype 52)		yes
	70	FTP server transfer completion record (subtype 70)	yes	
	71	FTP daemon configuration record (subtype 71)		yes
	72	FTP server logon failure record (subtype 72)	yes	
	73	IPSec IKE tunnel activation and refresh record (subtype 73)		yes
	74	IPSec IKE tunnel deactivation and expire record (subtype 74)		yes
	75	IPSec dynamic tunnel activation and refresh record (subtype 75)		yes
	76	IPSec dynamic tunnel deactivation record (subtype 76)		yes
	77	IPSec dynamic tunnel added record (subtype 77)		yes
	78	IPSec dynamic tunnel removed record (subtype 78)		yes
	79	IPSec manual tunnel activation record (subtype 79)		yes
	80	IPSec manual tunnel deactivation record (subtype 80)		yes
	94	OpenSSH Client Connection Started	yes	
	95	OpenSSH Server Connection Started	yes	
	96	OpenSSH Server Transfer Completion	yes	
	97	OpenSSH Client Transfer Completion	yes	
	98	OpenSSH Login Failure	yes	

zERT



zERT Summary Report

Generated: Feb 19, 2018, 2:41:55 AM

Data in Flight Encryption Statistics
SMF 119-11 TLS Protocol
Feb 12, 2018, 12:00 AM EST - Feb 19, 2018, 12:00 AM EST
Count

1
1
10
6,179,193
10,351,307
TLS or SSL protocol level
🟨 TLSv1.0 🟦 N/A ⬛ SSLv3 🟩 TLSv1.2 🟥 TLSv1.1

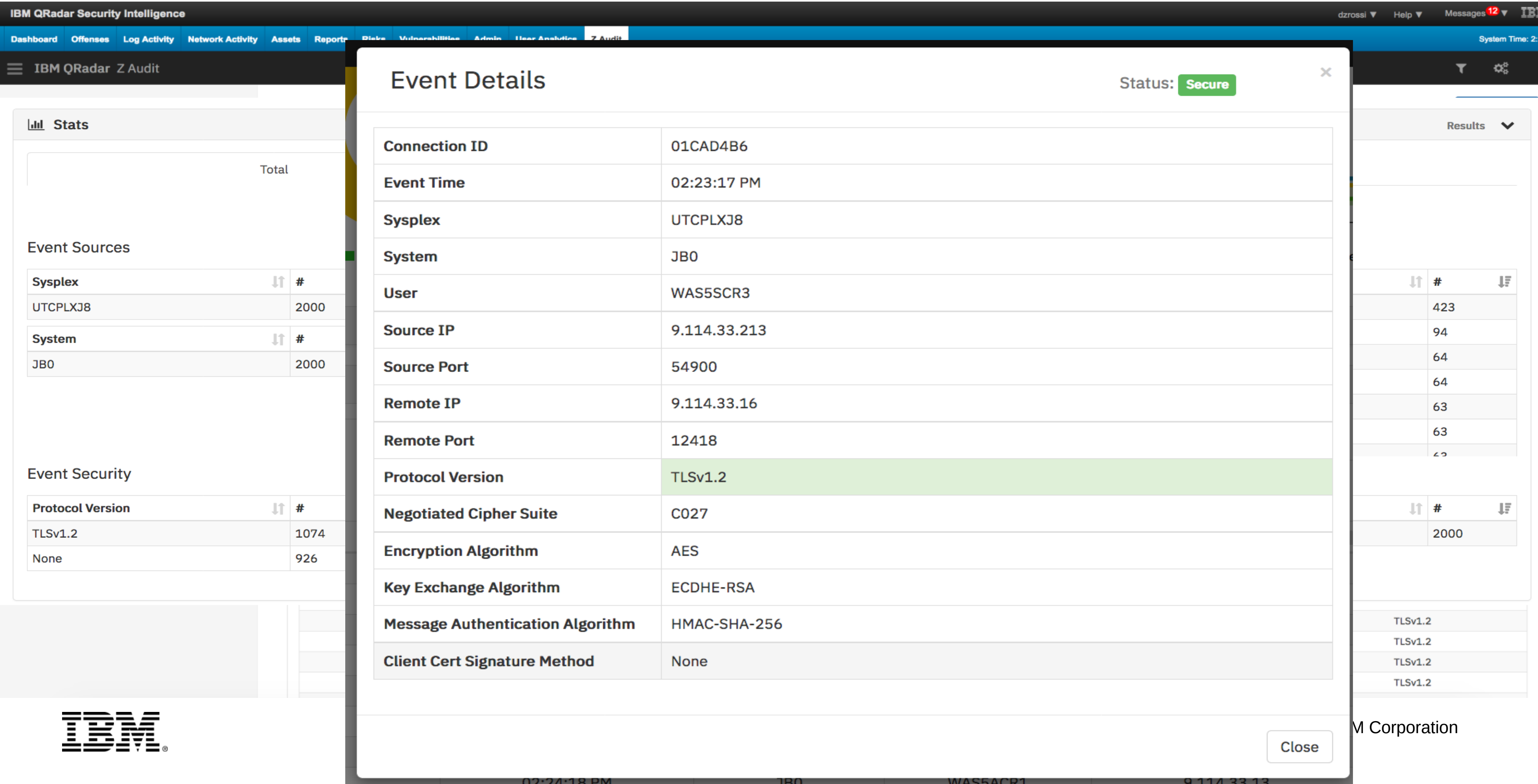
Encryptions Protocols in Use
SMF 119-11 TLS Protocol
Feb 12, 2018, 12:00:00 AM - Feb 19, 2018, 12:00:00 AM

TLS or SSL protocol level (custom)	TLS Algorithm (custom) (Unique Count)	TLS Channel (custom) (Unique Count)	TLS key length (custom) (Unique Count)	TLS message digest (custom) (Unique Count)	Count
TLSv1.0	AES	CBC	Multiple (2)	HMAC-SHA1	10,351,307
N/A	None	None	None	None	6,179,193
SSLv3	None	None	0	HMAC-MD5	10
TLSv1.2	AES	CBC	128	HMAC-SHA-256	1
TLSv1.1	AES	CBC	256	HMAC-SHA1	1

Log Sources sending zERT statistics
SMF 119-11 Logsource
Feb 12, 2018, 12:00:00 AM - Feb 19, 2018, 12:00:00 AM

Log Source	Subsystem name (custom) (Unique Count)	Sysplex Name (custom) (Unique Count)	Start Time (Maximum)	Magnitude (Minimum)	Event Count (Sum)	Count
IBM z/OS	JB0	UTCPLXJ8	Feb 18, 2018, 11:59:59 PM	3	16,530,512	16,530,512

Pervasive Encryption Dashboard



Event Sources

Sysplex	#
UTCPLXJ8	2000

System	#
JB0	2000

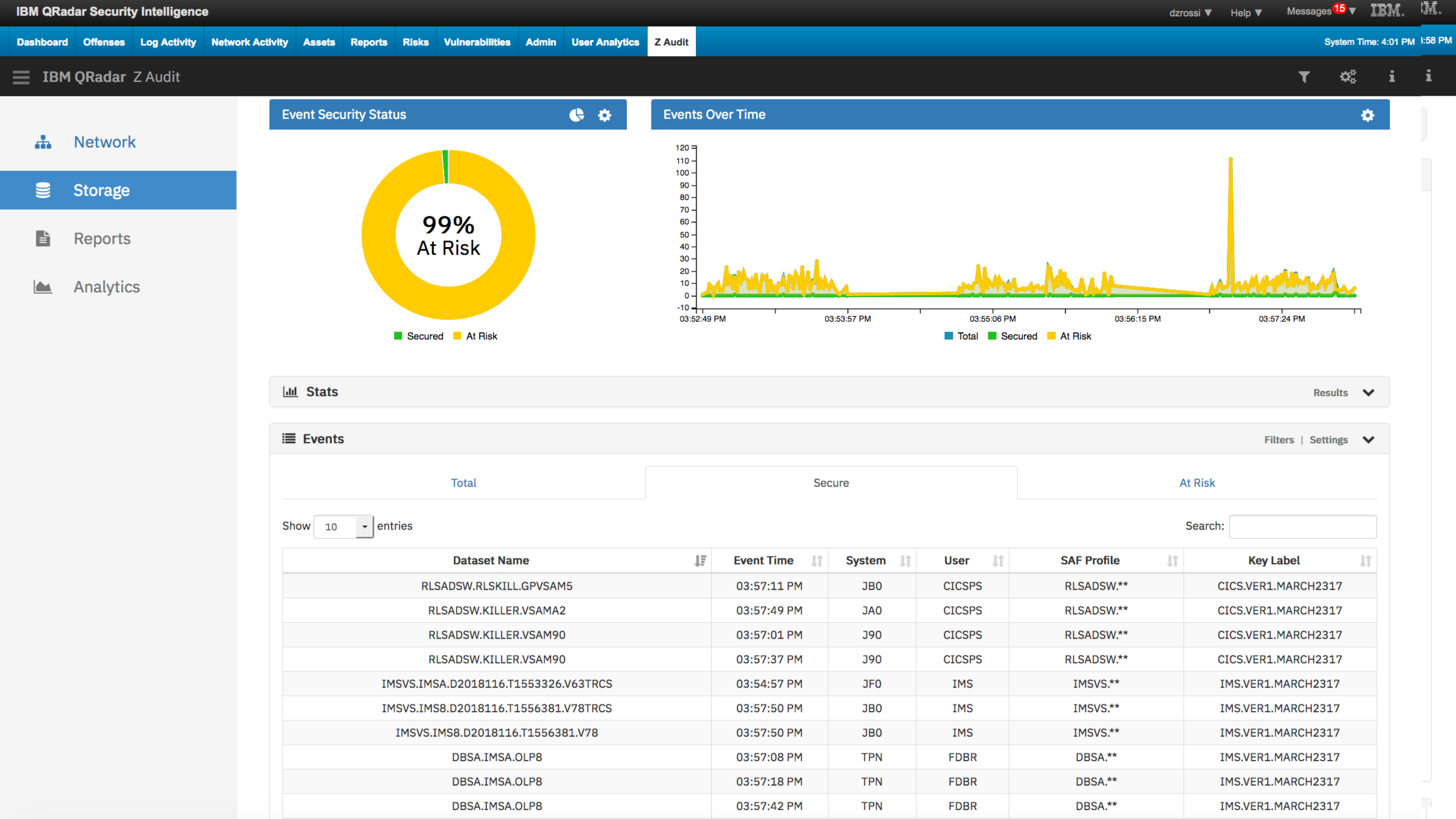
Event Security

Protocol Version	#
TLSv1.2	1074
None	926

Event Details

Status: Secure

Connection ID	01CAD4B6
Event Time	02:23:17 PM
Sysplex	UTCPLXJ8
System	JB0
User	WAS5SCR3
Source IP	9.114.33.213
Source Port	54900
Remote IP	9.114.33.16
Remote Port	12418
Protocol Version	TLSv1.2
Negotiated Cipher Suite	C027
Encryption Algorithm	AES
Key Exchange Algorithm	ECDHE-RSA
Message Authentication Algorithm	HMAC-SHA-256
Client Cert Signature Method	None



Dataset Name	Event Time	System	User	SAF Profile	Key Label
RLSADSW.RLSKILL.GPVSAM5	03:57:11 PM	JB0	CICSPS	RLSADSW.**	CICS.VER1.MARCH2317
RLSADSW.KILLER.VSAMA2	03:57:49 PM	JA0	CICSPS	RLSADSW.**	CICS.VER1.MARCH2317
RLSADSW.KILLER.VSAM90	03:57:01 PM	J90	CICSPS	RLSADSW.**	CICS.VER1.MARCH2317
RLSADSW.KILLER.VSAM90	03:57:37 PM	J90	CICSPS	RLSADSW.**	CICS.VER1.MARCH2317
IMSVS.IMSA.D2018116.T1553326.V63TRCS	03:54:57 PM	JF0	IMS	IMSVS.**	IMS.VER1.MARCH2317
IMSVS.IMS8.D2018116.T1556381.V78TRCS	03:57:50 PM	JB0	IMS	IMSVS.**	IMS.VER1.MARCH2317
IMSVS.IMS8.D2018116.T1556381.V78	03:57:50 PM	JB0	IMS	IMSVS.**	IMS.VER1.MARCH2317
DBSA.IMSA.OLP8	03:57:08 PM	TPN	FDBR	DBSA.**	IMS.VER1.MARCH2317
DBSA.IMSA.OLP8	03:57:18 PM	TPN	FDBR	DBSA.**	IMS.VER1.MARCH2317
DBSA.IMSA.OLP8	03:57:42 PM	TPN	FDBR	DBSA.**	IMS.VER1.MARCH2317

Additional Information

❑ Pervasive Encryption: IBM Z Platform Evaluation Test Experiences:
https://www.ibm.com/developerworks/community/blogs/43ea8e78-acbe-49f5-9290-379e4f4569cb/entry/Pervasive_Encryption_IBM_Z_Platform_Evaluation_Test_Experiences?lang=en

❑ How to Implement IBM Pervasive Encryption Data Set Encryption on z/OS (YouTube video):
https://www.ibm.com/developerworks/community/blogs/43ea8e78-acbe-49f5-9290-379e4f4569cb/entry/How_to_Implement_Pervasive_Dataset_Encryption_on_IBM_z_OS?lang=en

❑ IBM Crypto Education Community – Pervasive Encryption
https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/W7df80301055d_495b_bb88_a0a2f84757c5/page/Pervasive%20Encryption%20-%20zOS%20Data%20Set%20Encryption

❑ Data Set Encryption for IBM® z/OS® V2.2 Frequently Asked Questions:
https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FQ131494

❑ Documentation Updates for APAR OA50569 z/OS Data Set Encryption z/OS V2R2:
http://publibz.boulder.ibm.com/zoslib/pdf/OA50569.pdf

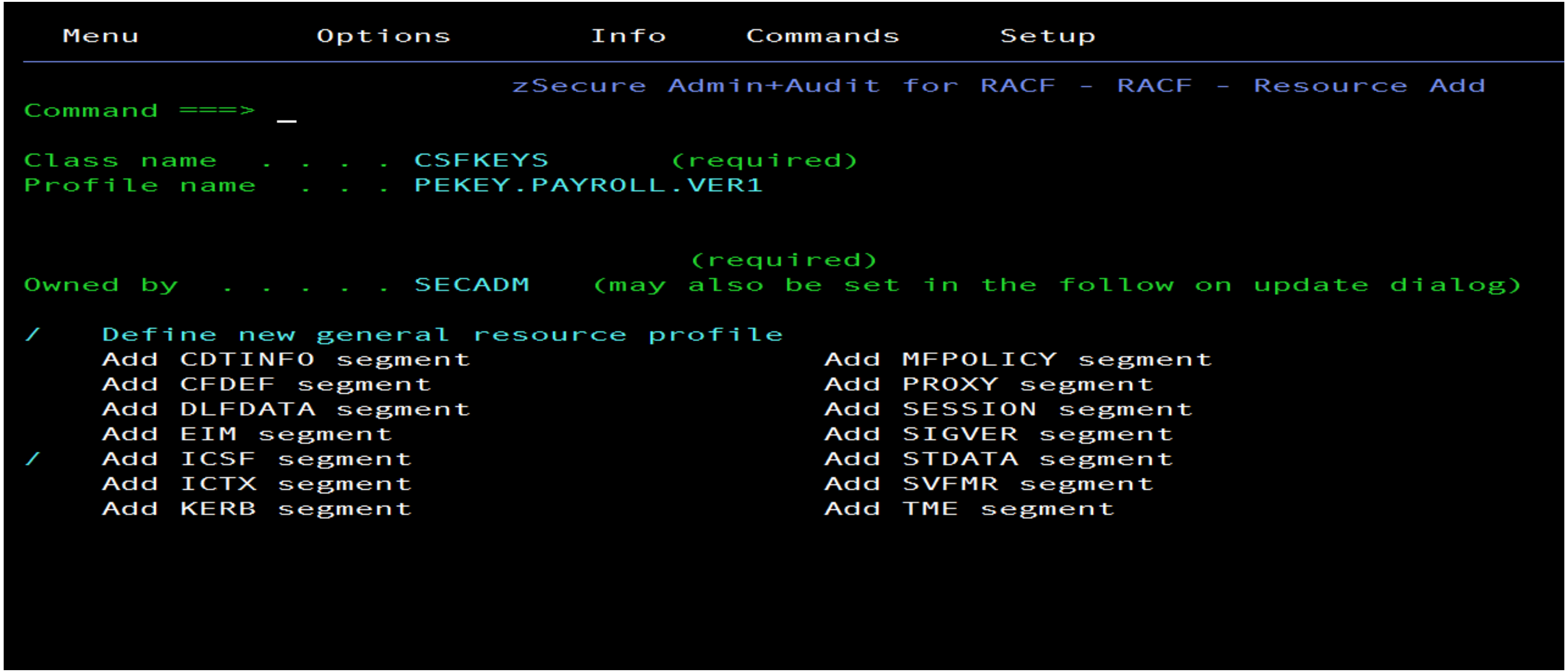
❑ IBM KnowledgeCenter Pervasive Encryption for V2R3:
https://www-304.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r3izsp100/$file/izsp100_v2r3.pdf

❑ z/OS DFSMS Using the New Functions z/OS V2R3:
https://www-304.ibm.com/servers/resourcelink/svc00100.nsf/pages/zOSV2R3sc236857/$file/idak100_v2r3.pdf

Backup

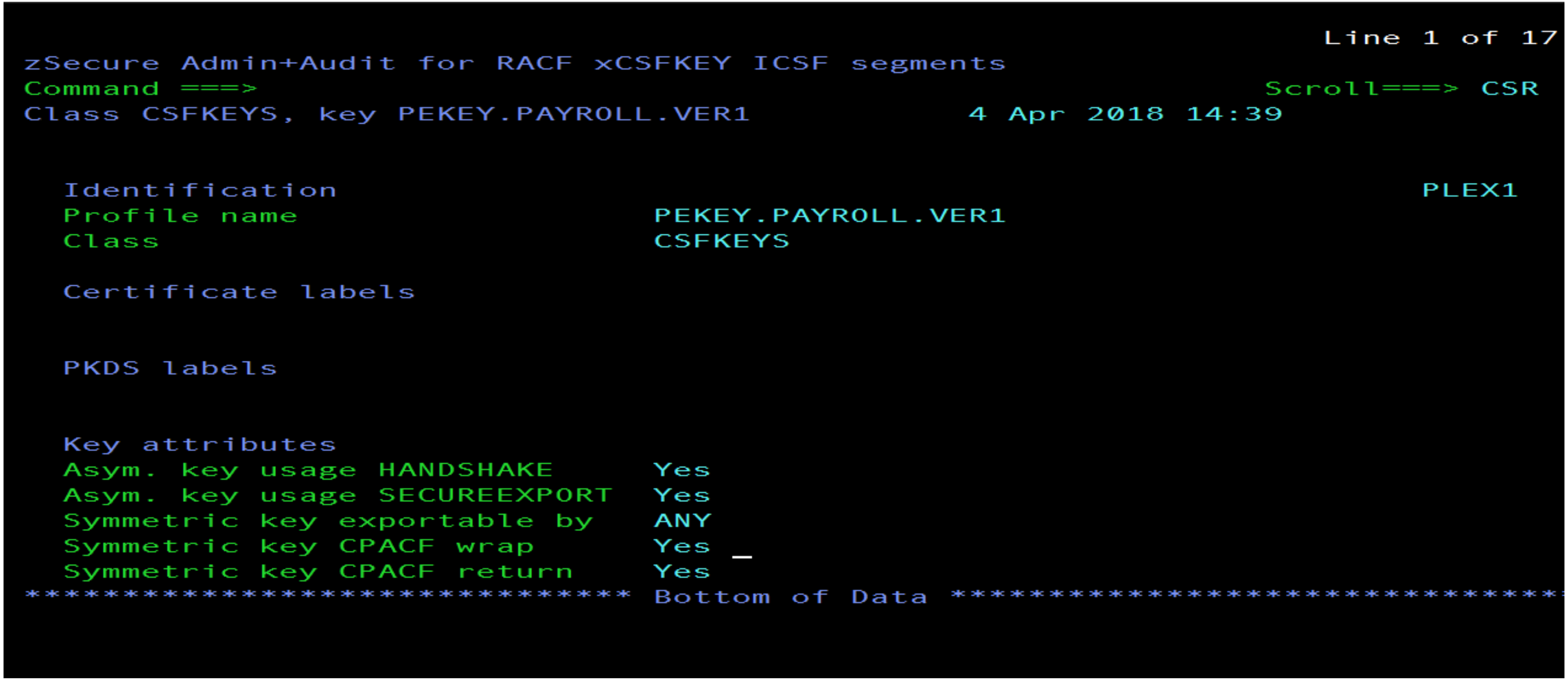
Using zSecure to Implement

Defining the Key via zSecure



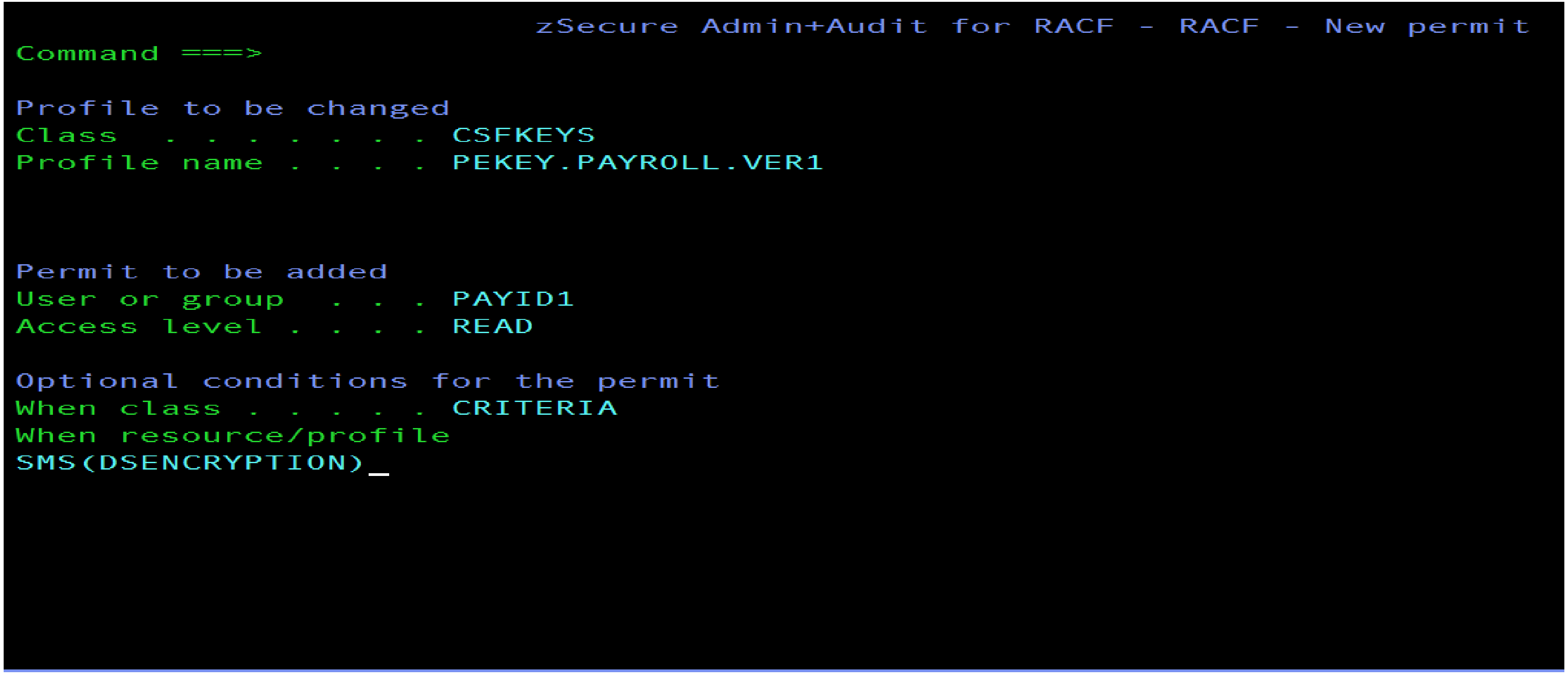
```
 Menu         Options       Info    Commands     Setup
-----------------------------------------------------------------------------
                         zSecure Admin+Audit for RACF - RACF - Resource Add
Command ===> _

Class name . . . . CSFKEYS      (required)
Profile name   . . . PEKEY.PAYROLL.VER1


                                    (required)
Owned by  . . . . . . SECADM   (may also be set in the follow on update dialog)

/   Define new general resource profile
    Add CDTINFO segment                         Add MFPOLICY segment
    Add CFDEF segment                           Add PROXY segment
    Add DLFDATA segment                         Add SESSION segment
    Add EIM segment                             Add SIGVER segment
/   Add ICSF segment                            Add STDATA segment
    Add ICTX segment                            Add SVFMR segment
    Add KERB segment                            Add TME segment
```

Adding ICSF Segment to Key via zSecure



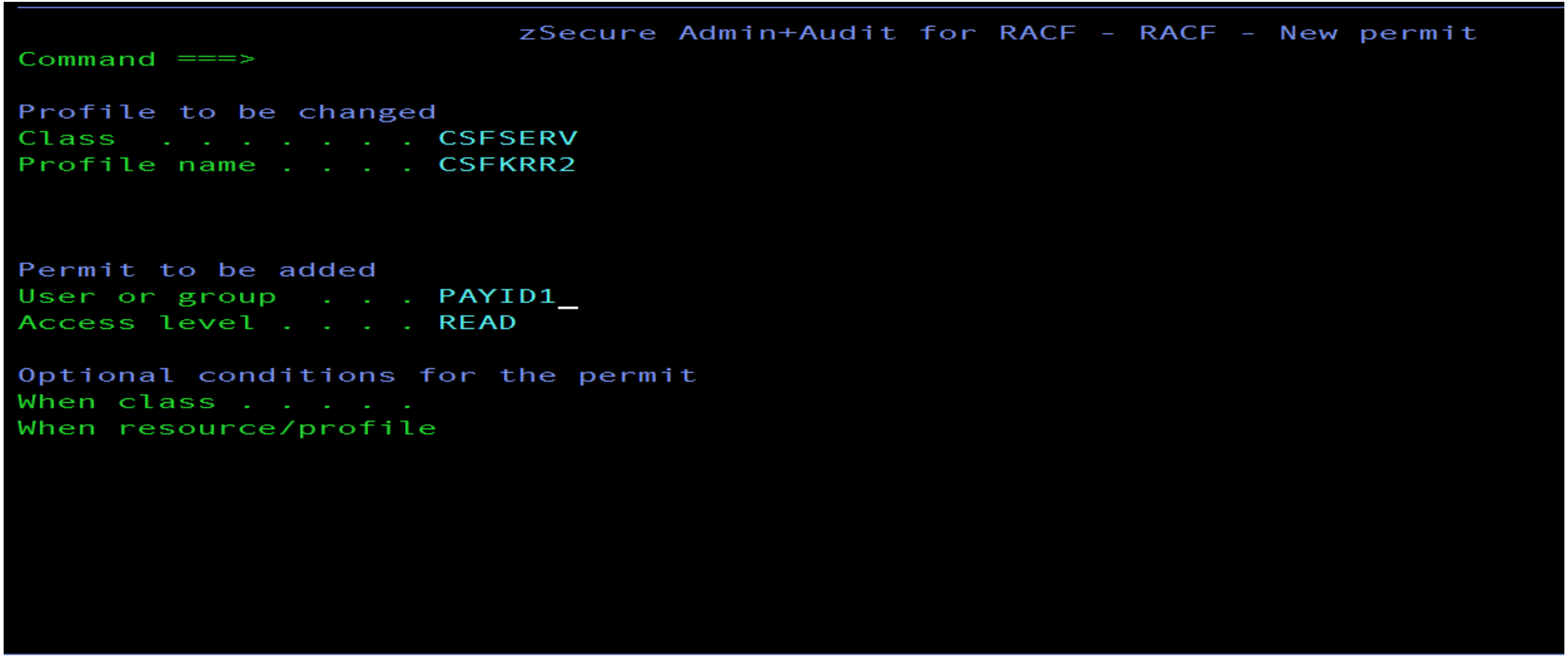
```
                                                                      Line 1 of 17
zSecure Admin+Audit for RACF xCSFKEY ICSF segments
Command ===>                                                       Scroll===> CSR
Class CSFKEYS, key PEKEY.PAYROLL.VER1                4 Apr 2018 14:39


  Identification                                                          PLEX1
  Profile name                    PEKEY.PAYROLL.VER1
  Class                           CSFKEYS

  Certificate labels


  PKDS labels


  Key attributes
  Asym. key usage HANDSHAKE       Yes
  Asym. key usage SECUREEXPORT    Yes
  Symmetric key exportable by     ANY
  Symmetric key CPACF wrap        Yes _
  Symmetric key CPACF return      Yes
******************************* Bottom of Data ********************************
```

Permitting User to the Key via zSecure



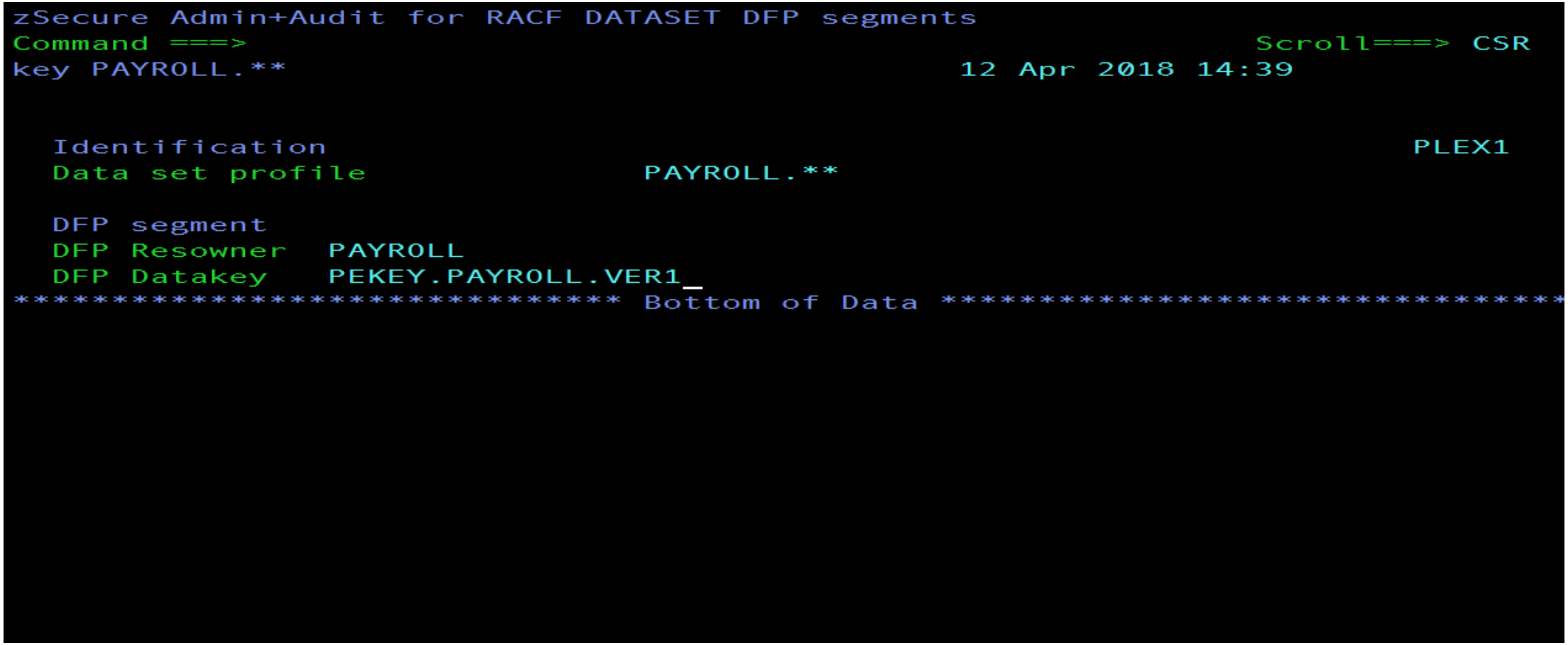
```
                         zSecure Admin+Audit for RACF - RACF - New permit
Command ===>

Profile to be changed
Class  . . . . . . .  CSFKEYS
Profile name . . . .  PEKEY.PAYROLL.VER1



Permit to be added
User or group  . . .  PAYID1
Access level . . . .  READ

Optional conditions for the permit
When class . . . . .  CRITERIA
When resource/profile
SMS(DSENCRYPTION)_
```

Permitting User to CSFKRR2 via zSecure



```
                         zSecure Admin+Audit for RACF - RACF - New permit
Command ===>

Profile to be changed
Class  . . . . . . .  CSFSERV
Profile name . . . .  CSFKRR2



Permit to be added
User or group  . . .  PAYID1_
Access level . . . .  READ

Optional conditions for the permit
When class . . . . .
When resource/profile
```

Defining Data Set Encryption Policy via zSecure



```
zSecure Admin+Audit for RACF DATASET DFP segments
Command ===>                                                Scroll===> CSR
key PAYROLL.**                                   12 Apr 2018 14:39


  Identification                                                  PLEX1
  Data set profile              PAYROLL.**

  DFP segment
  DFP Resowner   PAYROLL
  DFP Datakey    PEKEY.PAYROLL.VER1_
******************************* Bottom of Data ********************************
```