

IBM Z Data Privacy for Diagnostics

Elpida Tzortzatos
IBM Fellow, z/OS CTO, Solution Architect
elpida@us.ibm.com

Purvi Patel
z/OS Service Aids Product Owner
purvi@us.ibm.com

presented by:
Andrea Conzett
IBM zSystems Client Architect, IBM Switzerland
ajc@ch.ibm.com



Clients risk accidentally sharing sensitive customer information when sending diagnostic data to vendors

\$3.9M

Global average cost of a data breach

The average total cost has increased by 10% since 2014.

80%

Breaches that contained customer PII

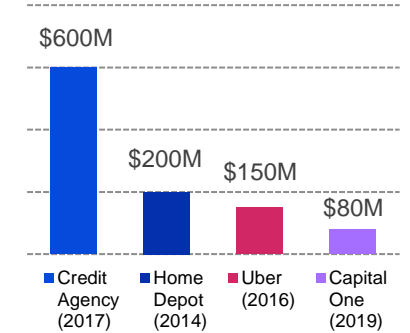
Customer PII is the most frequently compromised type of record.

\$150

Customer PII average cost per record

Customer PII is also the costliest type of record compromised.

Biggest data breach fines in millions of USD



Hacks enabled by weak security, cover-ups or avoidable mistakes have cost these companies a total of nearly **\$1.3B** and counting.

Source: CSO, The biggest data breach fines, penalties and settlements so far, 2020 - bit.ly/33Kt1rk

Market problem

- Data at-rest can be protected on premises with pervasive encryption and/or other solutions, but, if an error occurred while sensitive data was in-use, it might end up being included in the resulting dump that is available to others for root cause analysis.
- Organizations are forced to make a choice between regulatory compliance and serviceability.



Challenges

How to identify/detect sensitive data?

- Memory for sensitive data does not have unique characteristics

What if user data and meta-data were intermixed?

- Many applications intermix their sensitive data with meta-data making it difficult to take actions



Design points



Capture a complete dump as-is

- No additional impact to dump capture time
- Post-process the dumps to handle sensitive information



Maintain First Failure Data Capture (FFDC)

- Always keep the complete original dump
- Collect information about what led to a failure so that you don't need to re-create the failure



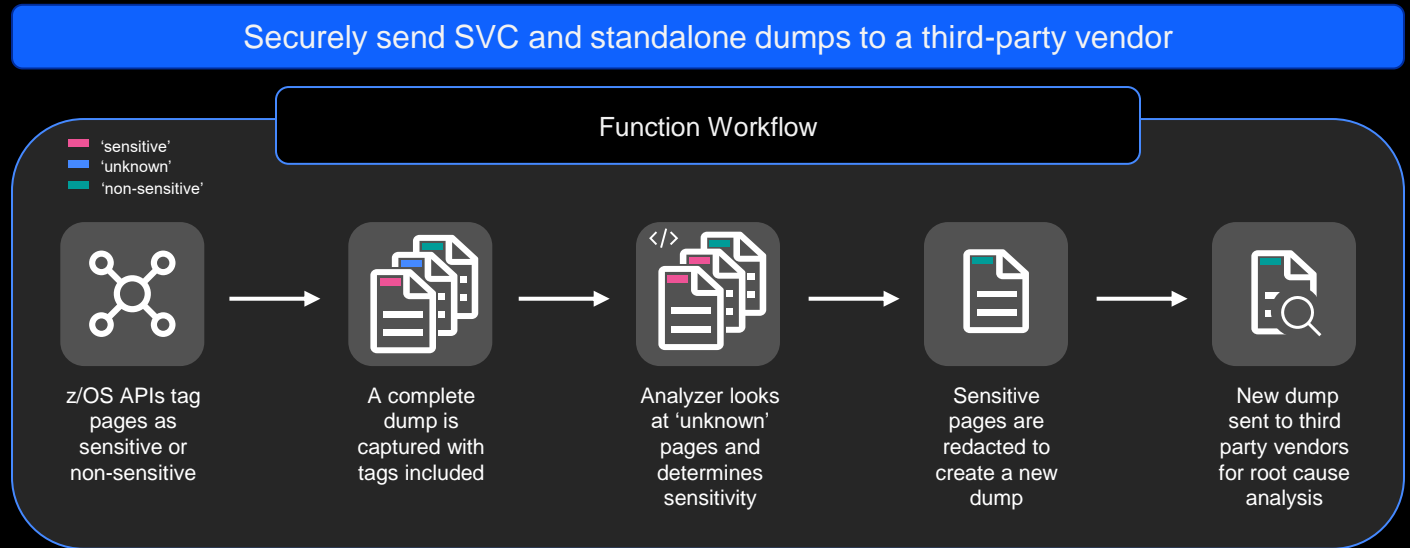
Redaction as a chosen method

- Not reversible
- Over-redaction is preferred to under-redaction

Introducing: IBM Z Data Privacy for Diagnostics

The only z/OS function that is designed to:

- Help clients **address compliance challenges** in the area of diagnostic data
- Help clients **more securely share** diagnostic data with third-parties
- Tag and redact sensitive diagnostic data **in minutes***



*Disclaimer: as measured in lab environment (to be approved)

Data Privacy for Diagnostics: solution overview

1. Provide z/OS APIs to tag user (sensitive) and system (non-sensitive) data

2. Record sensitivity tags with pages on IBM z15 system dumps

- A complete dump to be captured as-is, with sensitive tags
- No additional impact to dump capture times

3. Secure sensitive data in dumps – detect, tag, protect

- Post-process the complete dump → create a second dump
 - Leverage z/OS Diagnostics Analyzer to detect and tag or redact additional sensitive data
 - Opportunity to customize detection of sensitive data
 - Redact tagged pages
 - Post-processed dump can be sent to vendors
 - Provide redaction reports
 - List of redacted pages, etc.

Data Privacy for Diagnostics: solution overview

Designed to tag and redact sensitive user data from [SVC](#) and [stand-alone](#) diagnostic dumps captured on [IBM z15](#)

Two ways to tag data:

- z/OS APIs
- z/OS Diagnostics Analyzer



z/OS APIs for 64-bit storage

Tag known user data and metadata

- Tags user data as 'sensitive = yes'
- Tags metadata as 'sensitive = no'
- Defaults the rest as 'sensitive = unknown'



z/OS Diagnostics Analyzer

Interrogate untagged pages in the dump and tag as needed

- Uses regular expressions to form a search pattern
- Uses built-in custom, and dependent identifiers to tag sensitive data



At memory allocation time

- 64-bit memory allocation services to support new keyword: 'Sensitive(Unknown|Yes|No)'
- IARV64 Request (GetStor, GetCommon, GetShared)
- Supports all PageFrameSize types
- IARST64 Request (Get)
- IARCP64 Request (Build)



After memory allocation time

- New Request = ChangeAttributes on IARV64
- Change the sensitive specification of a set of pages
- Granularity is on 4K increments

z/OS Diagnostics Analyzer

Complements the z/OS API tagging solution

- Detects sensitive data in untagged pages
 - Pages tagged sensitive and non-sensitive are skipped by the z/OS Diagnostics Analyzer
- Includes a list of built-in identifiers and allows specification of custom identifiers
- Allows customization to identify sensitive data in one's environment
- Written in Java
 - Runs on zIIPs and exploits multi-threading to analyze dump pages

File system setup is required (See SYS1.SAMPLIB(BLSDPJIN))

- Create, initialize and mount the file system to the home directory
- Shell script provided to create necessary sub-directories
 - **Knowledgebase** – stores ingested knowledge and user feedback
 - **Configuration** – stores configuration files
 - **Reports** – stores generated reports on DPfD functions
 - A subdirectory is created for each dump
 - A subdirectory with each run-number

Provides the framework

- Must review the list of built-in identifiers
- Utilize custom identifiers to enhance accuracy of sensitive data detection
- Can be used to consider as sensitive or non-sensitive
- Allows specifying dependent identifiers
 - e.g., name with race/ethnicity

z/OS Diagnostics Analyzer

Detect sensitive data in untagged pages

- Runs as batch job (submitted from IPCS panel or JCL) to analyze dump pages
- Utilizes a filesystem to store various reports from different z/OS Diagnostics Analyzer functions



1. Analyze

Analyze input dump using built-in and custom identifiers



2. Report

Generate a user-friendly report of 'analyze' operation



3. Feedback

Provide feedback to fine tune analysis



Ingest

Ingest user provided data



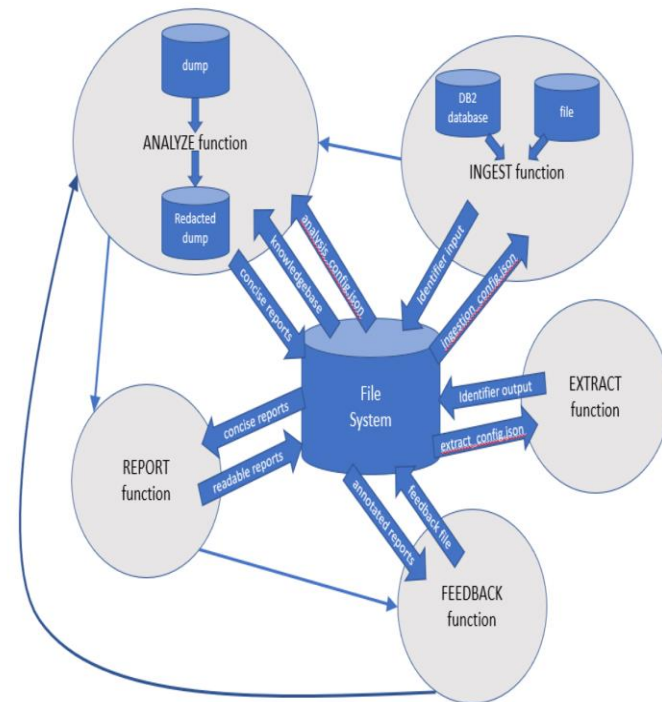
Extract

Extract/list ingested or included data

z/OS Diagnostics Analyzer

Customize and fine-tune sensitive data detection to improve accuracy

- Utilize 'ingest' to inform z/OS Diagnostics Analyzer about sensitive data patterns in your environment
- Analyze → Report → Feedback



Offering: Data Privacy for Diagnostics

1. Provide the infrastructure to tag data in z/OS

- At memory allocation & anytime after memory allocation
 - 64-bit storage allocation services, 4K granularity, tag it sensitive or non-sensitive

2. z/OS, Middleware, vendors, etc. to tag known data (sensitive or non-sensitive)

- Db2 & IMS buffer pools, various DFSMS buffers to be tagged sensitive

3. Engage applications to tag their known data (sensitive or non-sensitive)

4. z/OS to tag known system areas as non-sensitive

- Reduces post-processing times

5. Protect sensitive data in z15 dumps (SVC & stand-alone dumps)

- Leverage z/OS Diagnostics Analyzer to detect and tag additional sensitive data in a dump
- Redact 100% of tagged sensitive data → Create a second dump

IPCS – post processor panel

```
----- IPCS MVS DUMP BATCH JOB OPTION MENU -----
OPTION  ===>

 1  SADUMP      - Prepare stand alone dump for analysis
 2  SVCDUMP     - Prepare SVC dump for analysis
 3  SYSMDUMP    - Prepare SYSMDUMP for analysis
 4  SUPPLEMENT - Perform supplementary dump analysis
 5  EREP        - Process software data using EREP
 6  DPFD        - Data Privacy for Diagnostics

JOB STATEMENT INFORMATION:  (Verify before proceeding)

===> //DPFD JOB MSGCLASS=A,MSGLEVEL=(1,1),REGION=0M,
===> //      MEMLIMIT=NOLIMIT
===>
===>
===>
===>

Enter END to terminate batch job processing.
```

```
-----
*****
* USERID   - PURVIP
* DATE     - 19/09/24
* JULIAN   - 19.267
* TIME     - 01:39
* PREFIX   - PURVIP
* TERMINAL- 3278
* PF KEYS  - 24
*****
```


IPCS – DPfD main panel

----- Data Privacy for Diagnostics Request -----

COMMAND ===>

Press ENTER to edit parameters, END to terminate without job submission.

REQUESTED FUNCTION ===> ANALYZE (ANALYZE, REPORT, FEEDBACK, INGEST, EXTRACT)

ANALYZE - Analyze the input file using built-in and custom identifiers

REPORT - Generate a user friendly report of ANALYZE operation

FEEDBACK - Provide feedback to fine-tune future DPFD analysis

INGEST - Ingest the user provided data

EXTRACT - Extract identifier information

F1=HELP

F2=SPLIT

F3=END

F4=RETURN

F5=RFIND

F6=MORE

F7=UP

F8=DOWN

F9=SWAP

F10=LEFT

F11=RIGHT

F12=CURSOR

IPCS – DPfD ‘analyze’ panel

Press ENTER to submit the job, END to terminate without job submission.

```
DATA SET NAME      ===> 'D83DUMP.DYNZOS24.S58.D190911.T221308.SV00006'  
NEW DATA SET NAME ===> 'D83DUMP.DPFD.RED'  
TEMP DATA SET/PAT ===> 'D83DUMP.DPFD.TEMP'  
BYPASS DP ANALYSIS ===> N      (Y or N)  
REDACTION STRING   ===> !!!!!!!!!!!!!                               (0-32 characters)  
NUMBER OF THREADS  ===> 8      (1-8)  
ALLOW PAGE LEVEL   ===> Y      (Y or N)  
SENSITIVE REPORT   ===> Y      (Y or N)  
DPfD HOME DIR      ===> /DPfD  
JAVA HOME DIR      ===> /usr/lpp/java/java800/current_64  
JAVA OPTIONS       ===> -Xms2g -Xmx4g  
JZOS LOAD MODULE   ===> JVMLDM86  (JVMLDMxx, see JZOS Batch Launcher Toolkit)  
MIGLIB DATASET     ===> 'SYS1.MIGLIB'  
TEMP ALLOC PARMS   ===> storclas(sclarge) dataclas(compress)  
EDIT CONFIG FILE?  ===> Y
```

DPfD default analysis_config.json file

➔ /dpfd/configuration/analysis_config.json

```
{
  "built_in_identifiers_include" : [
    "Credit Card Number",
    "Credit Card Type",
    "Email",
    "IBAN",
    "FullName",
    "RaceOrEthnicity",
    "Religion",
    "US Address",
    "US Phone Number",
    "US SSN",
    "US SWIFT Code"
  ],
  "built_in_identifiers_exclude" : [
    "Age",
    "Animal",
    "ATC",
    "Continent",
    "Country",
    "County",
    "Date Time",
    "Day",
    "DEPENDENT",
    "EU NIN",
    "Gender",
    "Hospital Name",
    "ICDv10",
    "ICDv9",
    "IMEI",
    "IMSI",
    "IN Aadhaar Card Number",
    "IN Pan Card Number",
    "International Phone Number",
    "IP Address",
    "Latitude Longitude",
    "MAC Address",
    "Marital Status",
    "Medical Name",
    "Medical Record Number",
    "Month",
    "Occupation",
    "Phone Number",
    "PO Box",
    "Street Types",
    "UK NIN",
    "URL",
    "US States",
    "Vehicle Identification Number",
    "Year",
    "Zipcode",
    "FirstName",
    "LastName"
  ],
  "custom_identifiers" : [
    #{ "inputfilename" : "ingest_output_file.bin", "entitytype" : "Name that appears in reports", "description" : "Description of this identifier", "format" : "custom" }
  ],
  "dependent_identifiers" : [
    #{ "name": "NameWithEmail", "identifiers": [ "FullName", "Email" ] }
  ],
  "built_in_ns_identifiers_include" : [
    "moduleName"
  ],
  "built_in_ns_identifiers_exclude" : [],
  "custom_ns_identifiers" : [],
  "printable_characters" : ""
}
```

Who are our exploiters?

Following components exploited API option at service levels

- Db2
- IMS
- VSAM
- Various DFSMS components
- ZFS (in v2.5)

- See next page for PTF numbers

IBM z15 and z/OS V2.3 introduce IBM Z Data Privacy for Diagnostics at no additional cost

Fix Category (IBM.Function.DataPrivacyForDiagnostics), Keyword (DPFD/K)

Hardware

IBM z15 T01

IBM z15 T02

Software [z/OS v2.3 or higher](#)

- Storage Manager API Support:
APAR OA57633 (PTFs: UJ00688, UJ00699)
- Storage Manager API-2 (ChangeAttributes) Support
APAR OA58289 (PTFs: UJ03895, UJ03896)
- Storage Manager API-2 **PE**:
APAR OA60373 (PTFs: UJ04600, UJ04601)
- Service Aids Support:
APAR OA57570 (PTFs: UJ00714, UJ00715)
- z/OS Diagnostics Analyzer Support:
APAR OA58114 (PTFs: UJ04576, UJ04586)
- Basic Access Method:
APAR OA58712 (PTFs: UJ01664, UJ01665)
- Object Access Method:
APAR OA58431 (PTFs: UJ01359, UJ01360)
- Db2:
APAR PH15940 (PTF: UI65280)
- IMS:
APAR PH14059 (PTF: UI65556)
- VSAM:
APAR OA58730 (PTFs: UJ04647, UJ04649)



Additional resources

IBM

Blog, March 2020

[Link](#)

IBM Z Enterprise Security

[Link](#)

Announcement, December 2020

[Link](#)

Knowledge Center

z/OS MVS

[Link](#)

Db2 Support

[Link](#)

IMS Support

[Link](#)

Sales Enablement

Seismic

[Link](#)

Thank you!

© Copyright IBM Corporation 2021. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

Backup

Customize “analyze” options

- Number of threads (max 8)
- Redaction level
 1. Page (faster but may over-redact)
 - Token level for below-the-bar storage
 2. Token (detailed but can take longer)
- Redaction string
 - Makes it easier to spot redacted data in dump pages

New terms we've introduced with DPfD

Redactable dumps

- Dumps collected on z15 that can be post-processed

Redacted dumps

- Redactable dumps that have been post-processed

SVC and stand-alone dumps on v2.3 and v2.4 with OA57570 PTFs

- See the last slide for PTF numbers

SYSMDUMP and T-dump on v2.5

New terms we've introduced with DPfD

Sensitivity tags

- Each dump record has one byte indicator at '2B'x offset
- Sensitive=Unknown ('00'x)
- Tags set by API
 - Sensitive=Yes ('01'x)
 - Sensitive=No ('02'x)
- Tags set by the z/OS Diagnostic Analyzer
 - Sensitive=Yes ('80'x) full page
 - Sensitive=Yes ('40'x) partial page

How to find a sensitive tag in the dump record in IPCS?

- Displaying the 'block' (64+4096=4160) in IPCS in the original dump
 - DRPX is 64 bytes header for each 4K page in the dump

```
00000000  C4D9F240  C3E50000  FFFFFFFF  0000003C  | DR2 CV..... |
00000010  00000000  00000050  00800000  96513002  | .....&...o... |
00000020  00000000  08000000  00004001  00000001  | ..... |
00000030  550AE000  00000000  00000000  00000000  | ..\..... |
00000040.:01CF.--All bytes contain X'00'
000001D0  00000000  00D98195  8140E581  A4878895  | .....Rana Vaughn |
000001E0.:01EF.--All bytes contain X'40', C' '
000001F0  4040F1F4  F840D481  9385A2A4  81848140  | 148 Malesuada |
00000200  C1A58595  A4854040  40404040  40404040  | Avenue |
00000210  40404040  40404040  4040C485  93948595  | Delmen |
00000220  889699A2  A3404040  40404040  40404040  | horst |
00000230  40404040  40404040  4040F5F4  F3F0F440  | 54304 |
00000240  40404040  4040F3F6  F2F4F7F4  F6F1F4F3  | 3624746143 |
00000250  F6F6F1F5  40404040  40F5F3F9  00000000  | 6615 539.... |
00000260.:103F.--All bytes contain X'00'
```

64-byte DRPX

Sensitivity byte
Sensitive=yes('01'x)

4K data

How to find a sensitivity tag in the dump record?

- Display all BLSREDCT records in IPCS using COMPDATA(BLSREDCT) in the BROWSE panel
 - Sensitive=yes info also included in BLSREDCT

```
DSNAME ('MVSSVA.SECURDMP.SY1.D210125.S00001.REDACT') POINTERS -----  
Command ==>                                SCROLL ==> CSR  
ASID(X'003C') is the default address space  
PTR   Address      Address space      Data type  
s0001 00.          COMPDATA (BLSREDCT)  AREA  
Remarks:  
***** END OF POINTER STACK *****
```

```
COMPDATA(BLSREDCT) ADDRESS(00.) STORAGE -----  
Command ==>                                SCROLL ==> CSR  
00000000.:0FFF.--Storage not available  
00001000  D9C5C4C1  C3E34040  C3E50000  FFFFFFFF  | REDACT CV..... |  
00001010  0000003C  00000000  02000000  00000000  | ..... |  
00001020  00000050  00800000  01000000  00000800  | ...&..... |  
00001030  00000050  02000000  01000000  00000800  | ...&..... |  
00001040.:1FFF.--All bytes contain X'00'  
00002000.:01000FFF.--Storage not available  
01001000  D9C5C4C1  C3E34040  C3E50000  FFFFFFFF  | REDACT CV..... |  
01001010  00000001  00000000  ED000000  00000000  | ..... |  
01001020  00000000  008DE000  40000000  00000002  | .....|. |  
01001030  00000000  008E1000  40000000  00000001  | ..... |  
01001040  00000000  008E7000  40000000  00000001  | ..... |
```

What is tagged as non-sensitive in 31-bit storage?

- Enforce token-level redaction in 31-bit storage
- SDUMP Exit allows storage to be tagged
 - Following comp-exits have tagged their data as non-sensitive
 - Console, IOS, RSM, RTM, SLIP and WLM
- Following 'tagged' as non-sensitive when capturing the storage in dump
 - System trace buffers
 - Nucleus
 - LPA

IPCS – is this a redacted dump?

- New IPCS messages at dump initialization
 - Redactable and post-processed dump

```
IKJ56650I TIME-10:21:06 AM. CPU-00:00:01 SERVICE-154397 SESSION-01:15:51 JANUARY 15,2020
BLS18122I Initialization in progress for DSNAME('MVSSVA.SECURDMP.SADMP.DPA.AFTERE35')
BLS18124I TITLE=SADMP DPF0001
BLS18223I Dump written by z/OS 02.04.00-0 SADUMP - level same as IPCS level
BLS18557I This redactable dump has been post-processed to protect sensitive data
BLS18222I z/Architecture mode system
BLS18125I CPU(1) STATUS available
BLS18125I CPU(2) STATUS available
BLS18310I Stand alone dump required 00:03 to record to MVSSVA.SECURDMP.SADMP
BLS18255I Dump Init      Elapsed Time      CPU Time
          Input I/O    00:00:01.408455    00:00:00.023984
          DDIR         00:00:00.242923    00:00:00.124712
```

IPCS – is this a redacted dump?

- New IPCS messages at dump initialization
 - Redactable but not post-processed dump

```
IKJ56650I TIME-05:09:44 PM. CPU-00:00:04 SERVICE-394156 SESSION-02:29:07 JANUARY 15,2020
BLS18122I Initialization in progress for DSNNAME('MVSSVA.VARAN1.TWEAKSWO.NOREDACT.S00003')
BLS18124I TITLE=DPFD0011 WITH SUMLIST64
BLS18223I Dump written by z/OS 02.04.00-0 SVC dump - level same as IPCS level
BLS18558I This redactable dump has not been post-processed to protect sensitive data
BLS18222I z/Architecture mode system
BLS18255I Dump Init      Elapsed Time          CPU Time
          Input I/O      00:00:00.507451      00:00:00.010237
          DDIR           00:00:00.044555      00:00:00.021118
```


What else is new in redacted dumps?

A new component data record - BLSREDCT

- Lists pages tagged sensitive by an API or Analyzer
- SVC Dump
 - BLSREDCT for API tagged pages is created at the dump write time
 - BLSREDCT for Analyzer tagged pages is created during post-processing
- Stand-alone dump
 - BLSREDCT is created during post-processing for API and Analyzer tagged pages

BLSREDCT

- Lists pages tagged sensitive by an API or Analyzer
 - Format like the 1st reference compdata (BLSZEROP)

```
00000000.:0FFF.--Storage not available
00001000  D9C5C4C1  C3E34040  C3E50000  FFFFFFFF  | REDACT  CV..... |
00001010  0000003F  00000000  02000000  00000000  | ..... |
00001020  00000050  00800000  01000000  00000800  | ...&..... |
00001030  00000050  02000000  01000000  00000800  | ...&..... |
00001040.:1FFF.--All bytes contain X'00'
00002000  D9C5C4C1  C3E34040  C3E50000  FFFFFFFF  | REDACT  CV..... |
00002010  00000041  00000000  02000000  00000000  | ..... |
00002020  00000050  00800000  01000000  00000800  | ...&..... |
00002030  00000050  02000000  01000000  00000800  | ...&..... |
00002040.:2FFF.--All bytes contain X'00'
```

Redacted storage range starting address

Number of consecutive pages with same status

Tagged sensitive by API

BLSREDCT

- Lists pages tagged sensitive by an API or Analyzer
 - Format like the 1st reference compdata (BLSZEROP)

```
00000000.:0FFF.--Storage not available
00001000  D9C5C4C1  C3E34040  C3E50000  FFFFFFFF  | REDACT  CV..... |
00001010  00000001  00000000  FE000000  00000000  | ..... |
00001020  00000000  18676000  40000000  00000002  | ...&..... |
00001030  00000000  18679000  40000000  00000001  | ...&..... |
...
00002020  000001EF  85529000  80000000  00000001  | ...&..... |
...
```

Tagged sensitive (partial page) by the Analyzer (token level)

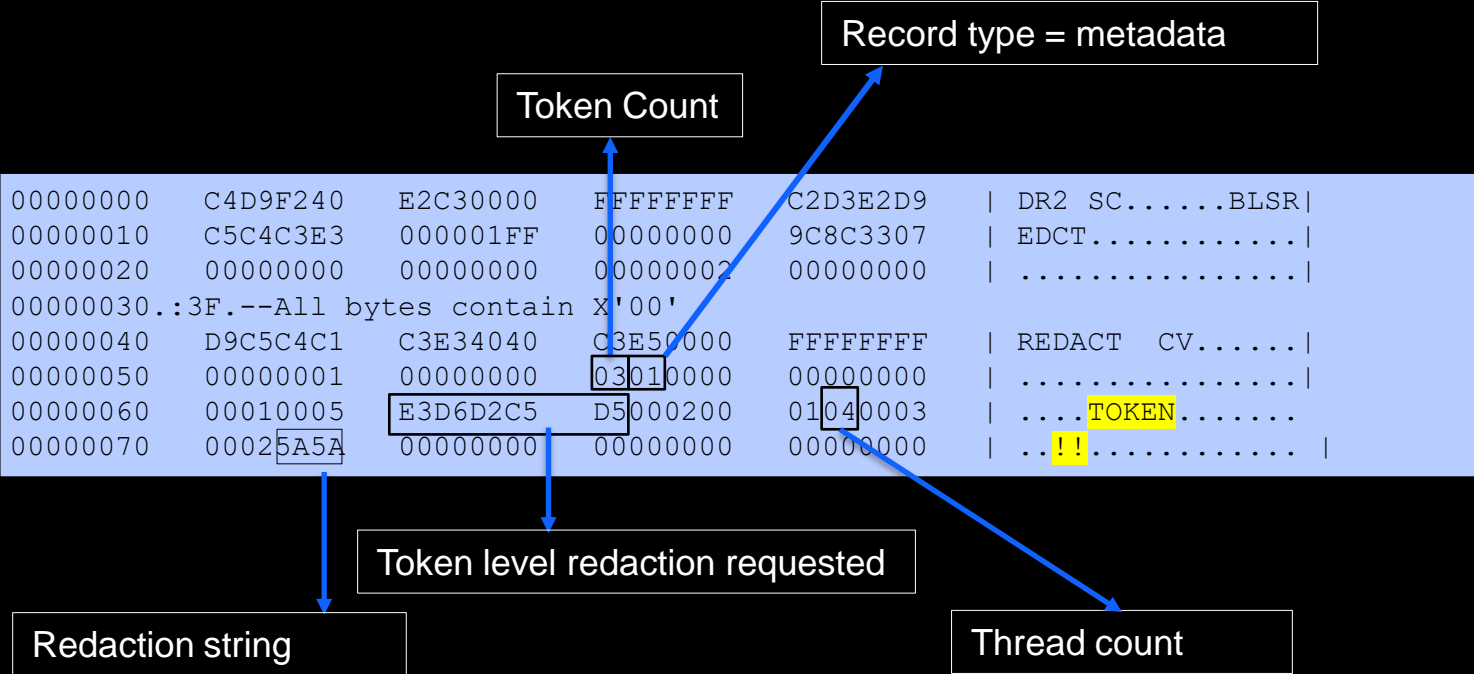
Tagged sensitive by the Analyzer (page level)

Redacted storage range starting address

Number of consecutive pages with same status

BLSREDCT – Analyze input

- Selected input from Analyze job – referred to as ‘analyze metadata record’
 - Created by BLSXDPPD



Is there an easier way to view redacted records?

BLSXREDR – REXX that reads COMPDATA(BLSREDCT)

- Lists ranges tagged sensitive
- Can be invoked from IPCS or TSO
- Input – dump name, ASID (optional)

BLSXREDR output

- In TSO: %BLSXREDR 'MVSSVA.DPFD.TESTDUMP'

Dump being analyzed: 'MVSSVA.DPFD.TESTDUMP'

```
.
00000000_7EBB0000 : 00000000_7EBB0FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_7EBBA000 : 00000000_7EBBAFFF, Redacted by Data Privacy Analysis, partial page(s)
.
Records for ASID 0007
00000050_0020A000 : 00000050_0020AFFF, Redacted by Data Privacy Analysis, whole page(s)
00000050_00212000 : 00000050_00213FFF, Redacted by Data Privacy Analysis, whole page(s)
00000050_00688000 : 00000050_00688FFF, Redacted by Data Privacy Analysis, whole page(s)
00000050_00878000 : 00000050_00878FFF, Redacted by Data Privacy Analysis, whole page(s)
.
Records for ASID 0006, DSPNAME=IXCDSMEM.
00000000_00900000 : 00000000_00901FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_01200000 : 00000000_01200FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_01D00000 : 00000000_01D00FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_02400000 : 00000000_02400FFF, Redacted by Data Privacy Analysis, partial page(s)
.
Records for COMPDATA='IARHVSHR'.
0000040C_FEFB9000 : 0000040C_FEFB9FFF, Redacted by Data Privacy Analysis, whole page(s)
0000040C_FF015000 : 0000040C_FF015FFF, Redacted by Data Privacy Analysis, whole page(s)
0000040C_FF01F000 : 0000040C_FF01FFFF, Redacted by Data Privacy Analysis, whole page(s)
0000040C_FF068000 : 0000040C_FF068FFF, Redacted by Data Privacy Analysis, whole page(s)
0000040C_FF087000 : 0000040C_FF087FFF, Redacted by Data Privacy Analysis, whole page(s)
```

BLSXREDR output

- In TSO: %BLSXREDR 'MVSSVA.DPFD.TESTDUMP' A A5

```
Dump being analyzed: 'MVSSVA.DPFD.TESTDUMP'
```

```
Records for ASID 00A5
```

```
00000000_00DB7000 : 00000000_00DB8FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_00DC0000 : 00000000_00DC0FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_00DC4000 : 00000000_00DC4FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_00DC6000 : 00000000_00DC8FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_7F266000 : 00000000_7F266FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_7F295000 : 00000000_7F295FFF, Redacted by Data Privacy Analysis, partial page(s)
00000000_7F2A6000 : 00000000_7F2A6FFF, Redacted by Data Privacy Analysis, partial page(s)
```

```
..
```

```
00000010_008FF000 : 00000010_008FFFFFF, Redacted by Data Privacy Analysis, whole page(s)
00000010_00AFC000 : 00000010_00AFCFFF, Redacted by Data Privacy Analysis, whole page(s)
0000005C_80007000 : 0000005C_80007FFF, Redacted by Data Privacy Analysis, whole page(s)
```

```
Records for COMPDATA='IARHVSHR'.
```

```
00000321_00006000 : 00000321_00006FFF, Redacted by Data Privacy Analysis, whole page(s)
00000321_F8A79000 : 00000321_F8A79FFF, Redacted by Data Privacy Analysis, whole page(s)
00000321_F8C10000 : 00000321_F8C10FFF, Redacted by Data Privacy Analysis, whole page(s)
```

Can the redacted storage range be listed in IPCS?

- Displaying sensitive ranges in a post-processed dump

```
>> ip list 50_00800000.asid(x'41') len(200)

_LIST 0050_00800000. ASID(X'0041') LENGTH(X'C8') AREA
_0800000. D9C5C4C1 C3E3C5C4 40C4C1E3 C1000000 |REDACTED DATA.....|
_0800010. 00000000 00000000 00000000 00000000 |.....|
_6B00020 LENGTH(X'A0')==>All bytes contain X'00'
_6B000C0. 00000000 00000000 |.....|
```

- Displaying sensitive ranges in the original dump

```
>> ip list 50_00800000.asid(x'41') len(4096)

LIST 50_00800000. ASID(X'0041') LENGTH(X'1000') AREA
_0800000 LENGTH(X'03B0')==>All bytes contain X'00'
_08003B0. 00D3E8C4 C9C14040 40404040 4040D2C5 |.LYDIA KE|
_08003C0. D3D3E840 40404040 40404040 4040D74B |LLY P.|
_08003D0. D64B40C2 D6E740F1 F5F96B40 F3F6F9F4 |O. BOX 159, 3694|
_08003E0. 40D4C1E4 D9C9E240 E2E34B40 40404040 | MAURIS ST. |
_08003F0. 40404040 4040E2C1 D540D1D6 C1D8E4C3 | SAN JOAQUIC|
_0800400. 60D54040 40404040 40404040 40404040 |-N |
_0800410. 40404040 4040F0F8 F4F8F340 40404040 | 08483 |
_0800420. 4040F5F1 F5F6F1F8 40F4F6F1 F7F3F640 | 515618 461736 |
_0800430. F7F4F3F1 40F2F1F9 00000000 00000000 |7431 219.....|
_0800440 LENGTH(X'0BC0')==>All bytes contain X'00'
```

Can the redacted storage range be listed in IPCS?

- Displaying a page with in-place “token” level redaction in post-processed dump

```
LIST 0B1A0320. ASID(X'0038') LENGTH(X'64') AREA
0B1A0320. 00000000 00000000 00005A5A 5A5A5A5A |.....!!!!!!|
0B1A0330 LENGTH(X'40')==>All bytes contain X'5A', EBCDIC C''
0B1A0370. 5A5A5A5A 5A5A4040 40404040 40404040 |!!!!!!|
0B1A0380. 40404040 | |
```

- Displaying a page with in-place “token” level redaction in the original dump

```
LIST 0B1A0320. ASID(X'0038') LENGTH(X'64') AREA
0B1A0320. 00000000 00000000 0000C194 81A88140 |.....Amaya |
0B1A0330. 40404040 404040E2 A3859788 8595A296 | Stephenso|
0B1A0340. 95404040 404040F2 F0F760F4 F9F9F540 |n 207-4995 |
0B1A0350. C58785A2 A381A240 E2A39985 85A34040 |Egestas Street |
0B1A0360. 40404040 40404040 40404040 404040C4 | D|
0B1A0370. 85A39996 89A34040 40404040 40404040 |etroit |
0B1A0380. 40404040 | |
```

What if DPfD redacts diagnostic data?

- Over redaction is possible and preferred by clients
- How to spot over redaction?
 - Via the redaction string
 - Analyze joblog will include redaction string if specified, default is 'xxx'
- Collaborate to determine if the redacted information can be shared
 - Support can provide IPCS command to issue against the original dump

Where are the JCLs?

- SYS1.SAMPLIB provides the sample JCLs
 - BLSDPJIN – Allocates the Data Privacy for Diagnostics home directory
 - Creates, initializes and mounts the file system at home directory & creates necessary subdirectories and JSON files for DPfD operations
 - BLSJDPFD – Redacts all API tagged pages & creates a redacted dump
 - Does not process untagged pages via the z/OS Diagnostics Analyzer
 - BLSJDPA – Analyzes the input dump & creates a redacted dump
 - Utilizes the z/OS Diagnostics Analyzer
 - BLSJDPF – Used to provide feedback after the ANALYZE/REPORT runs
 - BLSJDPI – Ingests the custom data to be used for future analysis
 - BLSJDPR – Generates user friendly reports after an ANALYZE run
 - BLSJDPX – Allows to extract either built-in or custom identifiers via INGEST

Ready to exploit Data Privacy for Diagnostics?

- Suggested steps
 1. Set up your DPfD environment – Customize & run SYS1.SAMPLIB(BLSDPJIN)
 2. Have custom data? – Ingest using IPCS->5.6->Ingest or SYS1.SAMPLIB(BLSJDPI)
 3. Ready to analyze – Analyze using IPCS->5.6->Analyze or SYS1.SAMPLIB(BLSJDPA)
 4. Create sensitive tokens report – Using IPCS->5.6->Report or SYS1.SAMPLIB(BLSJDPR)
 5. Need to adjust detection – Provide feedback using IPCS->5.6->Feedback or SYS1.SAMPLIB(BLSJDPF)
 6. Want to confirm? Re-analyze same dump and follow steps 3 & 4

DPfD configuration files

- Editable JSON files created to communicate with z/OS Diagnostics Analyzer where customization is allowed
 - Needed for ANALYZE, INGEST and EXTRACT operations

```
→ /dpfd/configuration/Ingestion_config.json
{
  "inputtype": "",
  "inputsource": "",
  "entitytype": "",
  "description": "",
  "outputfilename": "",
  "inputfilename": "",
  "inlinedata": [],

  # databaselocation is same as databasename
  # MVS Command to get database information : display ddf

  "database": "",
  "databasehost": "",
  "databaseport": "",
  "databaseusername": "",
  "databasepassword": "",
  "databaselocation": "",
  "databaseschema": "",
  "databasetablename": "",
  "databasecolumnname": ""
}
```


Updated documentation

- **OA58114 updated the following manuals**
 - <ftp://public.dhe.ibm.com/s390/zos/sva/OA58114.pdf>
- **z/OS MVS Diagnosis: Tools and Service Aids (GA32-0905)**
 - High level introduction to 'Data Privacy for Diagnostics'
- **z/OS MVS IPCS Customization(SA23-1383)**
 - New chapter 'Customizing Data Privacy for Diagnostics'
- **z/OS MVS IPCS User's Guide(SA23-1384)**
 - New chapter 'Using Data Privacy for DiagnosticsAnalyze'
- **z/OS MVS IPCS Commands(SA23-1382)**
 - Explains BLSXREDR REXX EXEC