SQL WorkloadExpert for Db2 z/OS

Compliance with compliments!

Audit Component
AGENDA

1. Audit needs and musts
2. Solution overview and their Pros/Cons
3. The viable way – let Db2 do the magic!
4. Customer results from the banking industry
Audit – do you need it, do you care?!

- 61% of companies state that cybercrime and data theft are their most challenging threats
- 92% of respondents of a Lloyds Bank survey suffered a data breach in the past five years
- The number of attacks is growing each year
- Last year about ½ Billion records stolen
- Reputation is significantly affected by breaches
- The average cost of a data breach is $5.4M+
- Fines, penalties and losses are $105 - $359 per data record → calculate your companies costs
- Securing environments is still a catch-up task:
  - Technology gets better, but attackers still find their way – sometimes from inside!
- 95% of the attacks start with a human error (website, or Email attachment)
Audit – do you need it, do you care?!

- Attackers use...
  - SQL injection
  - DDoS
  - Third-party software
  - XSS
  - Malware
  - Phishing
  - Watering holes/Honey pots
  - Physical access

... with the ultimate goal of gaining access to your crown jewels ...
Audit – do you need it, do you care?!

... the enterprises database servers are the target in 96% of the security breaches!
Because they contain your/your client’s most valuable information...
  ▪ Customer personal information (PI, such as SSN)
  ▪ Detailed personal information
  ▪ Personal financial data (PFI, also credit reporting)
  ▪ Bank account/credit card information
  ▪ Health information
... and once they’re in, there are high volumes of easy-to-access, structured data.

→ Companies (and governments) love Big Data – attackers love companies’/governments’ databases!
Audit – do you need it, do you care?!

But the mainframe is safe – isn’t it?!
- 50% of the concerns are about privileged insiders
- 29% of the concerns are with web-enabled z/OS apps
- 21% of the concerns are with advanced persistent threats

“As mainframes become a major component in SOA, they are increasingly exposed to malware. Web services on the mainframe have significantly impacted security”

President, Mittal Technologies Inc.
Audit needs and musts

However, protecting and auditing is a major cost factor nowadays, thus the authorities had to force companies to pay attention:

- SOX – Sarbanes Oxley Act
- FIEL – Financial Instruments and Exchange Law
- PCI DSS – Payment Card Industry Data Security Standards
- HIPAA – Health Insurance Portability and Accountability Act
- CMS ARS – Center for Medicare/Medicaid Services Acceptable Risk Safeguards
- GLBA – Gramm-Leach-Bliley Act (Financial Services Modernization)
- ISO 17799 (Basel II), ISO 27001 (Basel III)
- NERC – North American Electric Reliability Corporation
- NIST 800-53 (FISMA) - National Institute of Standards and Technology (Federal Information Security Management Act)
Audit needs and musts

- Chose your *favorite* one and/or use a reliable resource for guidance:
  - Center of Internet Security (CIS)
  - Department of Defense (DoD)
  - Security Technical Implementation Guide (STIG)
  - Common Vulnerability Exposure (CVE)
  - Bundesamt für Sicherheit in der Informationstechnik (BSI)
# Audit needs and musts

- Make sure you meet your business needs (e.g. PCI DSS):

<table>
<thead>
<tr>
<th>PCI DSS Requirement</th>
<th>Authentication, Authorization or Audit?</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1 Establish a process for linking all access to system components to each individual user...</td>
<td>Authentication, authorization AND audit</td>
</tr>
<tr>
<td>10.2.7 Log the creation and deletion of system level objects</td>
<td>Audit</td>
</tr>
<tr>
<td>10.3 Record audit trail entries for all system components for each event...</td>
<td>Audit</td>
</tr>
<tr>
<td>10.5 Secure audit trails so they cannot be altered.</td>
<td>Audit</td>
</tr>
<tr>
<td>10.6 Review logs for all system components related to security functions at least daily</td>
<td>Audit</td>
</tr>
<tr>
<td>10.7 Retain audit trail history for at least one year; at least three months of history must be immediately available for analysis</td>
<td>Audit</td>
</tr>
</tbody>
</table>

© 2018 SEGUS & SOFTWARE ENGINEERING GMBH 9
Audit needs and musts

- Critical activities that enterprises should be auditing
  - Privileged Users
    - Access/changes/deletion to critical data
    - Access using inappropriate channels
    - Schema modifications
    - Unauthorized addition of user accounts
  - End Users
    - Unusual access to excessive amounts of data
    - Access to data outside standard working hours
    - Access to data through inappropriate channels
  - Developers, Analysts and System Administrators
    - Access to live production systems
  - IT Operations
    - Inappropriate changes to DB/DB applications
Audit needs and musts

- Focusing on the major areas of concern – the database server:

<table>
<thead>
<tr>
<th>Audit Logging Requirements</th>
<th>Cobit (SOX) FIEL</th>
<th>PCI DSS</th>
<th>HIPAA</th>
<th>CMS ARS</th>
<th>GLBA</th>
<th>ISO 17799 27001</th>
<th>NERC</th>
<th>NIST 800-53 FISMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECTs against sensitive data</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Insert, Update, Delete</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access violations</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Schema Changes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Grants/Revokes</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Audit needs and musts

- ... or in other words:
  Collect as much data as you can, because you probably don’t know today what you’ll need tomorrow → breach patterns do change!!!

- Make sure you include:
  - SELECTs (against sensitive data)
  - DDL
  - DML
  - DCL
  - Utilities (online + offline)
  - Commands
  - Assignment, or modification of a user ID/authorization – especially privileged users
Audit needs and musts

- Be careful what happens outside of a table:
  - Consider clones
  - Consider backups
  - Consider extended statistics in catalog tables, like SYSCOLDIST + SYSKEYTGTDIST
  - Consider utility output (REORG, RUNSTATs)
  - Consider UNLOADs
  - Consider Replication
  - Consider access to the underlying VSAM cluster

- Also consider your INSTALL SYSADM/SYSOPR
  → Sorry DBAs, but Auditing requires a separation of duties
Audit needs and musts

- Most Home-Grown Solutions are based on the Db2 Audit Trace
  - Class 1, 2, 7, 8 have very little overhead
    - Access violations
    - GRANTs/REVOKEs
    - Assignment, or modification of a user ID/authorization
    - Start of a Db2 online utility
  - Class 3 (IFCID 142) has very little overhead
    - DDL (only for TB having the AUDIT ALL attribute)
  - Class 4, 5 (IFCID 143, 144) has 15% - 20% overhead
    - 1st SELECT, INSERT/UPDATE/DELETE of a UOR
  - IFCID 90, 91 have very little overhead
    - Db2 Commands
Solution overview and their Pros/Cons

- There are a variety of existing resources Db2 already provides/comes with:
  - Db2 Log
  - Db2 Trace
  - Db2 Memory (DSC/EDM)
  - Db2 Exits

- And of course additional products
Solution overview and their Pros/Cons

Db2 Log:

- **Pros:**
  - Comes with Db2 and supports all versions
  - No additional overhead
  - No additional costs (except you want to keep logs for a longer period of time than currently and, of course, your analysis)
  - Most companies have Log analysis tools they’re already familiar with

- **Cons:**
  - Not all required data is logged
    - SELECTs are especially lacking
Solution overview and their Pros/Cons

Db2 Trace:

- **Pros:**
  - Comes with Db2 and supports all versions
  - No additional costs (except for storing and processing the collected data)
  - Most companies have trace data analysis tools they’re already familiar with

- **Cons:**
  - Depending on the scope (number of IFCIDs/classes), and the type (SMF, OPX, GTF, SRV), the overhead may be significant
  - You need to build your own repository
Solution overview and their Pros/Cons

Db2 Trace:
- What are the differences:
  - There are different types of traces:
    - Statistics, Accounting, Audit, Monitor, Performance, Global
  - There are different classes
  - There are hundreds of individual IFCIDs

→ Depending on your choice, the overhead is unmeasurable to significant

→ A key difference in cost is the trace destination!
  - SMF, OPX, GTF, SRV
Solution overview and their Pros/Cons

Db2 Trace:
  - What are the differences:
    - Processing the data requires simple to more sophisticated knowledge:
      - SMF: System Management Facility: Most commonly used, easy to process (use DSN1SMFP)
      - Opn/OPX: Buffer Destination Trace: very efficient, but Assembler needed to process (DSN1SDMP is pretty poor)
      - GTF: Generalized Trace Facility: Used for detailed monitoring
      - SRV: Serviceability Routine: We have never seen it used
Solution overview and their Pros/Cons

Db2 Memory (DSC/EDM):

- **Pros:**
  - Comes with Db2 and supports all versions
  - No additional overhead
  - No additional costs (except for storing and processing)

- **Cons:**
  - Not all required data is there
  - Usually you can’t access it yourself, unless you hook into it
  - The information is volatile and can get lost quickly
Solution overview and their Pros/Cons

Db2 Exits:

- **Pros:**
  - Partially comes with Db2 and supports all versions
  - No additional costs (except for storing and processing)

- **Cons:**
  - Not all required data is there
  - Lot’s of coding necessary to catch and process the data
  - The overhead may be significant
Solution overview and their Pros/Cons

Additional Tools:

- **Pros:**
  - There are various solutions to choose from
  - Usually easy to use and more powerful than native Db2 options
- **Cons:**
  - Vendors charge for it
  - Implementation and processing overhead may be significant
  - Additional appliances lead to more vulnerability and administration overhead
Solution overview and their Pros/Cons

Additional Tools:

• What are the differences?
  • Good solutions have efficient data collectors and share repositories for Audit, Performance Management, Accounting, Analytics ...
  • Some solutions use hooks into the Db2 address space to capture SQL activity – errors can bring down Db2, or the entire LPAR, thus they try to protect Db2 by encapsulating the “foreign” code
  • Some solutions need additional appliances (easily up to 100+ virtual appliances) → all SQL captured is sent (unencrypted!) through the network. If the connection gets lost they try to cache it. Keep in mind that attackers do DDoS attacks!
Solution overview and their Pros/Cons

Additional Tools:

- What are the differences?
  - Some solutions exploit zIIP processors
    - Optional (scope)
    - Forced usage
  - Some solutions offer reporting in real-time
  - Some solutions offer alerting
    - This requires a rule, or profile setup
      → keep in mind that they are based on known patterns
  - and of course solutions differ in
    - Setup (collector per Db2 system/LPAR)
    - Filtering
    - Dedicated support of compliance reports
Solution overview and their Pros/Cons

Additional Tools:

- What are the differences?
  - Some solutions have additional capabilities:
    - Covering a variety of databases (Db2, z/OS/LUW, IMS, Oracle, SQL Server, ...)
    - Covering applications (CICS, SAP, ...)
    - Covering dataset activity and Content Managers (VSAM, FTP, SharePoint, ...)
    - Covering Big Data (Hadoop, HANA, ...)
    - Covering vulnerability scanning of up to entire infrastructures (including network, firewall, workstations, ...)
    - Covering logons, connects

→ Depending on your choice it may become complex and expensive and you’re locked to a specific vendor!
The viable way – let Db2 do the magic

The most reliable/efficient solution is based on those reliable and robust Db2 key functions we’ve been using for ages.

Exploiting them results in the most powerful solution:
- You benefit from rock solid features, like:
  - Security
  - Compression
  - Native Db2 functions
  - Extended Client Identification Registers, sqleseti()

The only question is: What key Db2 functions are needed?
The viable way – let Db2 do the magic

Using IFCIDs along with OPX buffers delivers in-depth information without the overhead of SMF processing:

316/318  Dynamic SQL (SELECT, INSERT, UPDATE, DELETE)  
(+317 for the full SQL statement)
400/401  Static SQL (SELECT, INSERT, UPDATE, DELETE)  
(+SYSPACKSTMT for the full SQL statement)

Add the correlation headers to get detailed authentication data
The viable way – let Db2 do the magic

Using IFCIDs along with OPX buffers delivers in-depth information without the overhead of SMF processing:

23/24/25 Utility start/phase/stop (+219=Listdef+220=DSs)
55/83/87 SQLID setting
90/91 Commands and their completion status
140 Authorization failures
141 Authorization changes
62/142 DDL/DDL for tables with audit changes/all
270/271 Trusted Context and Row Permission masks

Add the correlation headers to get detailed authentication data
The viable way – let Db2 do the magic

BUT:

Make sure it’s secure!

- Set up and audit access to the repository
- Alert via WTO if someone messes with the IFCIDs you’ve chosen
- Consider automatically cancelling threads of users violating the rules
The viable way – let Db2 do the magic

- All IFCIDs listed have a much smaller footprint than AUDIT CHANGES/ALL
- This is integrated, reliable Db2 technology
- OPX is the right target for efficient capturing
- Store it in a repository and protect it using proven technology (e.g. RACF, ACF2, Top Secret)
- Using Db2 compression reduces storage requirements exploiting proven, integrated technology

→ No new vulnerabilities:
  - Black Box appliance
  - Massive sensitive data transmissions over the network
The viable way – let Db2 do the magic

Do your (automated) reporting/alerting/analytics as needed:
- SPUFI
- Batch Job
- Enterprise wide reporting system
- GUI (DRDA based queries are fully zIIP eligible)

If you don’t want to improve your Home Grown solution, find a vendor who exploits this technology
The viable way – let Db2 do the magic

DSC and EDM provide detailed workload insights, including flushed statements:
- SQL text
- Statement ID
- Date/time
- Current status
- Resource consumption
- Identification/environmental data
The viable way – let Db2 do the magic

Efficient data collector for your desired scope of Audit

Mainframe Engine

Workstation Engine

24 x 7 SQL Workload Capture

WLX
WLX Started Task or iterative job

DB2
DB2 DSNMSTR
System Service Address Space

IFCID

Iterative Workload Processing

Capture processing

Select

Explained

Insert, Update

DB2 Catalog/RTS

WLX
WLX Tables

WLX
Workload Warehouse Repository

Graphical User Interface

Type 4 Java
The viable way – let Db2 do the magic

Capture the data e.g. using a STC:
Run a started task 24x7 to catch all the IFCIDs that Db2 will be throwing and store the data.

Process the workload:
Externalize and process the data, such as every 60 min:
- customizable (e.g. 30 - 180 minutes)
- allow Ad hoc data refresh triggered via operator command for the started task (MODIFY)
- capture the SQL Text at trace time
The viable way – let Db2 do the magic

Use a GUI front end, preferably Eclipse:
Exploit and integrate into Eclipse based GUI front ends
  - GUIs can come as a Plug-in for
    - IBM Rational
    - IBM Data Studio
    - Eclipse native
  - Existing Db2 connections are used to connect to the mainframe
  - Interactive dialogs allow complex and powerful analysis
  - Export features can create PDF reports and allow MS Excel hand over
The viable way – let Db2 do the magic

GUI features – button overview

© 2018 SEGUS & SOFTWARE ENGINEERING GMBH
The viable way – let Db2 do the magic

Delivered Use Cases make using the product as easy as possible
The viable way – let Db2 do the magic

Selection, Filtering and Sorting makes the delivered Use Cases easy to customize
The viable way – let Db2 do the magic

Drill down to the statement text to see what the suspect did
The viable way – let Db2 do the magic

Compare workload and SQL to find anomalies
Create reports using powerful graphical charts
The viable way – let Db2 do the magic

Choose how you’d like to find out who did what and when...

Access to audited tables
Audit (DML)
Authorization failures
Changed audited tables
CREATE, ALTER, DROP(DDL)
DBADM data updates
DBADM object update
DB2 commands
End of identify
End of signon in CICS
GRANTs and REVOKEs (DCL)
Object Update Dynamic
PUBLIC access to tables
Set current SQLID
Show Primary Auth Ids
System DBADM data update
System DBADM object upd.
SQL INTENTs
SYSADM data updates
SYSADM object updates
The viable way – let Db2 do the magic

Choose how you’d like to find out who did what and when...
The viable way – let Db2 do the magic

Use free text search capabilities to scan your entire workload for sensitive data = in-depth audit candidates (e.g. credit card numbers, social security numbers, ...)
Customer results from the banking industry

Requirements:

- Capture DDL, DCL, DML from ‘inside’ as well as DDF
- Capture any activity in a UoR
- Capture static and dynamic SQL statements
- Show logon id as well as functional id
- Generate daily audit reports matching give filters
- Generate specific reports matching specific SQL statement classification
- Generate reports based on RACF id/group
- Generate unified reports for a data sharing group, as well as individual subsystem
- Email reports to Db2 Auditor group
- Capture Db2 online utilities
- Merge multiple systems reports
Customer results from the banking industry

Setup:

- WLX STC HA implementation
  - STC at the LPAR/Db2 DS member level to assure continuous capturing even during LPAR restart
- Workload processing once a day to generate daily audit reports
  - Automated via job scheduler
  - All Db2 systems merged into a common report
  - Objects and activity (DML, DDL, DCL) filtered
  - Reports sent via Email
- Specific reporting as needed via GUI
  - In-depth suspect analysis
  - Banking authority quarterly/annual reports
Customer results from the banking industry

Customization:
- **Capture DDL, DCL, DML from ‘inside’ as well as DDF**
- **Capture any activity in a UoR**
- **Capture static and dynamic SQL statement**
- **Capture Db2 online utilities**

PRINTCOM=YES
MONITOR=mbr
IFCIDS=(23, 24, 25, 62, 90, 91, 140, 141, 142, 143, 144, 219, 220, 316, 318, 400, 401)
BUFSIZE=16384
PERC=25
SWAP=YES
PGFIXRET=NO
PGFIXBUF=NO
PINGINT=5
WAKEUP=10
WAIT4DSN=600
INIT64=2
MAX64=64
The viable way – let Db2 do the magic

Using IFCIDs along with OPX buffers delivers in-depth information without the overhead of SMF processing:

23/24/25 Utility start/phase/stop (+219=Listdef+220=DSs)
90/91 Commands and their completion status
140 Authorization failures
141 Authorization changes
62/142 DDL/DDL for tables with audit changes/all
316/318 Dynamic SQL (SELECT, INSERT, UPDATE, DELETE) (+317 for the full SQL statement)
400/401 Static SQL (SELECT, INSERT, UPDATE, DELETE) (+SYSPACKSTMT for the full SQL statement)

Add the correlation headers to get detailed authentication data
Customer results from the banking industry

Show DDL activities:
Customer results from the banking industry

Show DCL activities:

- Audit selection
  - Choose type of audit
    - Audit
    - SQL INTEGRITY
    - Object Update Dynamic
    - Show Primary Auth IDs
    - SYSADM object updates
    - SYSADM data updates

- DCL and DDL
  - Authorization failures
  - GRANTS and REVOKEs (DCL)
  - Changed audited tables
  - CREATE, ALTER, DROP (DDL)
Customer results from the banking industry

Access violations due to insufficient authorities:
Customer results from the banking industry

DML Reporting:

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement Timestamp</td>
<td>The timestamp that this statement was written into the SSC database.</td>
</tr>
<tr>
<td>WLX DB2 SSID</td>
<td>The WorkloadExpert Group or Subsystem DB2 name for this statement.</td>
</tr>
<tr>
<td>Primary Authorization ID</td>
<td>The Primary Authorization ID used to identify the application.</td>
</tr>
<tr>
<td>Package</td>
<td>The package used by the statement.</td>
</tr>
<tr>
<td>Collection ID</td>
<td>The Collection ID used by the statement.</td>
</tr>
<tr>
<td>Primary Authorization ID</td>
<td>The Primary Authorization ID used to identify the application.</td>
</tr>
<tr>
<td>Sum of Executions</td>
<td>The total number of Executions.</td>
</tr>
<tr>
<td>Transaction name</td>
<td>A value provided by the RRS signon or resignon.</td>
</tr>
<tr>
<td>End User ID</td>
<td>A value provided by the RRS signon or resignon.</td>
</tr>
<tr>
<td>Workstation name</td>
<td>A value provided by the RRS signon or resignon.</td>
</tr>
<tr>
<td>Package CONTEXT</td>
<td>A value provided by the RRS signon or resignon.</td>
</tr>
<tr>
<td>Current SQL ID</td>
<td>For Static SQL the CONTEXT of the Package.</td>
</tr>
<tr>
<td>Qualifier</td>
<td>The Current SQL ID that is running the statement.</td>
</tr>
<tr>
<td>First referred Table Qualifer</td>
<td>The Qualifier used at Bind time for user table.</td>
</tr>
<tr>
<td>First referred Table Name</td>
<td>The first table Qualifier in the statement.</td>
</tr>
<tr>
<td>Statement text</td>
<td>The first table name in the statement.</td>
</tr>
<tr>
<td>Query no.</td>
<td>The complete text for the SQL statement.</td>
</tr>
</tbody>
</table>

| User provided id string      | User provided id string. |
| Authorization ID             | Authorization ID. |
| Job name or logon ID         | Job name or logon ID. |
| Connection name              | Connection name. |
| Plan name                    | Plan name. |
| Initial authorization ID     | Initial authorization ID. |
| Connection type              | Connection type that was used for an access. |
| Accounting                   | Accounting token. |
| Workstation user ID          | Workstation user ID. |
| Transaction or application name | Transaction or application name. |
| Workstation name             | The enduser's workstation name. |
| Context name                 | Trusted context name. |
Customer results from the banking industry

Detected anomalies: suspicious increase in SQL executions:

SQL Timeline: WLX Key
Customer results from the banking industry

Show logon id as well as functional id:

![SQL WorkloadExpert: Database activity](image-url)

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>End User ID</th>
<th>Primary Authorization ID</th>
<th>Current SQL ID</th>
<th>Qualifier</th>
<th>Package</th>
<th>Query type</th>
<th>Table creator</th>
<th>Table name</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
<tr>
<td>WLUPRIVST</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>HOPPE</td>
<td>IQADBACP</td>
<td>UPDATE</td>
<td>IQA6300</td>
<td>IQAXXX001</td>
<td>T</td>
</tr>
</tbody>
</table>

Result counter: 94
Customer results from the banking industry

Generate daily audit reports matching give filters
Customer results from the banking industry

Generate daily audit reports matching give filters
Customer results from the banking industry

Runtime & Costs:
- Capture STC < 15sec. CPU/month (3-way DS)
- 150k stmt. < 3min processing

Results:
- Fully automated report generation for authorities and internal/external auditors, provided via Email
- Exceptional workload detected and stopped within minutes
- Power User-IDs found, being used for daily work
- Access from VPN/WAN networks found
- Access violations detected
- 3rd party applications with update intent, but should actually be read
SQL WorkloadExpert for Db2 z/OS

So now you know...

- Of course it is easier with the Audit component of SQL WorkLoadExpert for Db2 z/OS
  - Data Warehouse
  - Extensible and Extendable
  - Low CPU cost
  - Fully based on official Db2 features and functions
  - Exploits Db2 security and compression
  - Is inside your protected environment
    - No new vulnerability
    - No time consuming implementation
  - Utilizes your existing investments
WLX typical use cases

Application Development:

- Application Workload Analysis: E.g. which machine load is produced by a certain Application?
- Explain Tool link (e.g. SQL Performance Expert, IBM DataStudio)
- Show same SQL on Multiple Schemas to detect "heavy-hitters"
- SQL Text Analysis for free text search (e.g.: BIF [Built-in Function] and UDF [User-Defined Functions] -usage, Java-formatted timestamps, etc.)
- View to detect "heavy-hitters" resulting from identical statements using different predicates
- Find unused (orphaned) SQL
WLX typical use cases

**Workload/Performance management:**

- Workload-Change, Problem-Detection and Trending, Comparison of CPU consumption, I/O, execution rates, current KPIs and deltas – calculated and summarized to the costs of multiple apps
- Disc Problem Detection – I/O Rates
- SQL KPIs – Background Noise and Exceptions
- SELECT Only Table Detection (READ only activity)
- Delay Detection (All queries which are delayed)
- Up and Down Scaling of SQL Workloads
- DSC Flush Analysis
- CPU Intensive Statements
- Index Maintenance Costs
WLX functional packages of use cases

**Database Administration:**
- Find never used Objects (Tables, Indexes, and Tablespaces)
- Find never executed Packages

**Audit and Security:**
- AUDIT tables being accessed
- AUDIT Db2 data being accessed
- AUDIT data manipulation (insert/update/delete)
- See where updates came from (inside or outside the local network)
- See where data is being accessed from (IP address, intra-/extranet, etc.)
- SQL Text Analysis for free text search (BIF [Built-in Function] and UDF [User-Defined Functions] - usage, Java-formatted timestamps, etc.)
Questions???

Many thanks for your attention and now....

Q & A
you have
Questions
we have
Answers