IBM InfoSphere Guardium

Database Activity Monitoring in a High Volume DB2 on z/OS Environment - Some Considerations
Contents

2 Introduction and Executive Summary
3 IBM InfoSphere Guardium – A Brief Overview
3 Database Access Concepts
4 Access controls on DB2 for z/OS
4 Privileged Users
5 Trusted Applications
6 Activity Monitoring Solution Characteristics and Attributes
7 A typical z/OS implementation
8 Performance – Technical Details Supplement
9 Key performance considerations
10 Conclusion

Introduction and Executive Summary
This document discusses various database activity monitoring concepts and topics, and also includes a supplemental technical discussion regarding the performance characteristics of the IBM® InfoSphere® Guardium® solution.

To summarize the information contained below and our conclusions
In our experience, in particular with large database environments such as those seen on z/OS®, an effective auditing approach needs to be based on profiling through the use of “black lists” and “white lists”. “White lists” are defined as a group of “well trusted” connections/users/plans, and, by extension, “black lists” are connections/users/plans which are “suspicious”. Administrators can tune the contents of these lists based on observed access behavior as necessary.

In any active database environment, it is neither cost effective nor practical to monitor everything. For very large z/OS customers, SQL activity can number into the billions of statements per day. Collecting each of these SQL statements can result in some fairly obvious challenges such as:

- Collection of this amount of data will trigger operational issues surrounding the audit environment such as, increased storage costs, inability to effectively store data for long term retention, increased costs and complexity.
- Billions of SQL Statements being collected, transmitted over the network to the appliance, stored in the audit repository, and all associated system and network overhead associated with that level of collection.
- Audit reports will result in possibly 1000’s of pages of output, which few auditors would be able/willing to read, making any meaningful analysis of the data difficult, and without the ability to gain any real security insight from the SQL Statements being collected.
For these reasons, we currently find we have very few z/OS customers who are implementing an approach based on the “collect everything” method. Rather, we see most z/OS customers using some form of the “white/black list” approach. This approach generally follows the following guidelines:

- For most z/OS customers only database accesses by privileged users, such as Database or System Administrators are of audit data collection interest.
- OLTP and Production application access to DB2® data is typically viewed as well protected by RACF® or the equivalent 3rd party security product, and as such are of little or no audit interest on z/OS. These SQL events are excluded from collection.
- Using a combination of monitoring based on the “white/black list” approach, customers find that an approach that gives visibility for ALL activity of privileged users, ALL use of Ad-Hoc SQL query tools, and excludes activity from well vetted applications tends to represent the best performing combination; providing all necessary audit visibility while exhibiting the best performance characteristics of the Guardium solution.
- For the DB2 on z/OS environment, close collaboration and coordination between the security and database administration teams are essential to ensure that no inadvertent performance changes are introduced whenever audit collection policies are changed for information security reasons.
- The use of professional services as part of a production implementation will provide the best implementation experience for any Guardium customer. Customers should also consider a “health check” periodically so that they can apply best practices as new features are introduced over time.

It is important to remember that investment in and implementation of any database activity monitoring solution needs to result in a manageable and actionable audit trail. Deploying an approach which is unwieldy or generates excessive amounts of auditing detail will be ineffective, and only serve as a checkbox solution.

**IBM InfoSphere Guardium – A Brief Overview**

The Guardium approach provides an enterprise wide solution which is completely scalable with manageable impact to application performance. The solution comprises two components; the first is a software agent, running on z/OS as an address space, referred to as the STAP Agent/Collector. This address space is responsible for inspection of ALL DB2 database activity, and based on collection profiles, as part of an audit policy, to selectively collect and publish audit events to the Guardium appliance. This second component, the Guardium appliance, acts as the audit repository and provides all the reporting, alert, and audit processing infrastructure to the security organization. The appliance is based on a hardened LINUX operating system, with an imbedded relational database manager acting as the audit repository. There is no root access privilege or any native database privilege delivered with the appliance component, this allows us to provide a tamper resistant solution for database activity monitoring on z/OS.

**Database Access Concepts**

Data which is stored in databases is often referred to as structured data. We will limit our conversation to such data, but be aware that sensitive data can be stored in other formats,
referred to as unstructured data, and this data needs protection and monitoring as well, since sensitive information can be located in these repository types. In a relational database environment, such as DB2 for z/OS, data access typically is through the use of SQL (Structured Query Language).

Access to database information can comprise one of the following general classifications or characteristics:

- **Application Access**: This is access to corporate data in a consistent and repeatable manner, as part of a defined process that represents some business function. An example of this might be an application used by customer service representatives to check account balances. Many times, these application processes are executed in an OLTP (On-line Tele-Processing) environment such as CICS® or IMS™.

- **Ad-Hoc Access**: This approach is where there is no predefined pattern of data access, rather various tools are used by different business users to process data in a database in a way that will change based on the question being asked or researched. One example of this might be the use of an Ad-Hoc query tool such as QMF or Crystal Reports to look for trends or pattern analysis across a large portion of data.

- **Administrative Access**: In order to provide for the care and feeding of the database environment, there are many situations which require data access by administrators (DBA, System Administrators, and Application Developers). This access might be needed in order to perform actions such as database backup and recovery, reorganization of data for optimal performance, schema changes to support application migrations, etc. This access will be typically via the use of administration tools and utilities, which might or might not issue SQL.

**Access controls on DB2 for z/OS**

For the z/OS environment, customers will use either the native security elements of the DBMS, for DB2 this is referred to as “Grant/Revoke” security, or will control access to DB2 data and resources through the use of an external security agent. This agent on z/OS will be either RACF or the CA-ACF2 or CA-Top Secret equivalent security product. Access privileges to data can be granted for direct table access, or indirectly via execution privileges on application plans or packages.

**Privileged Users**

While most privileges need to be granted by through a security function, some privileges are inherited by users with elevated levels of authority. Examples of this will include the implied ability to access data of the object owner (creator), or the plan or package execution privilege given to the binder of the application package/plan. In addition, some classes of users, for example SYSADM users, will gain access to all objects, since DB2 will not perform authorization checking on any SYSADM related activity.

Our definition of privileged users are those users who have broad based access privileges, with little or no direct control over what data they are accessing. In general, due to the nature of their responsibilities as custodians of the DBMS, DBA and System Administrators continue to need this broad level of access privileges in order to successfully conduct their day-to-day activities, so restricting access to such categories of users is not feasible. Such users could, without a business need-to-know requirement, access data in an inappropriate fashion.
Because of this juxtaposition of elevated privilege required for their day-to-day activities combined with an absence of capability to restrict access, it is imperative that ALL activities associated with privileged user activity in the DBMS environment be monitored in such a way as to be:

- Completely monitored
- Stored in a tamper proof event repository precluding privileged user interference
- Audit data collection and reporting approach independent of privileged user involvement

**Trusted Applications**

As mentioned before, application access is a defined and repeatable process to access database information. As an application that executes on z/OS is developed, there are usually a series of steps which are involved in moving that prepared application from the development environment, across the various lifecycle phases, to where it finally ends up as a production application. These migration processes are typically well separated from span of control standpoint, and well protected from a security and approval perspective.

Once implemented, access to these applications can be granted and controlled by the data or business owner, and access enforcement is typically implemented via the use of RACF or the equivalent 3rd party security manager. This enforcement will typically consist of several different layers where RACF authentication will occur. Take, as an example, the CICS application execution scenario where the following authorization processes can occur:

- In order to use execute an application, the end-user must first gain access to the corporate network, which can result in the first RACF authentication, many times in conjunction with the use of multiple factor authentication such as certificate processing.
- Once the TN3270 session is established, the end-user will then need to log into the CICS TOR (Terminal Owning Region), which requires a second RACF authentication to be satisfied.
- The end-user then will enter the CICS TXID (Transaction Identifier) which initiates the CICS application and is subject to a 3rd RACF check.
- Finally, once the application issues the first SQL statement, there is a 4th RACF authorization check that can occur on behalf of the scheduling of a DB2 package or plan by the DB2/CICS interface.

In the above scenario, most customers’ view that this form of data access is “trusted”, as it is well controlled by RACF. Because of this view, this form of access, for many customers, is viewed as being outside of the audit scope, and SQL statements from applications are not collected as audit events.

In a similar fashion, most distributed applications are initiated with either a well-known IP address that represents the location of the application server and/or a RACF ID used to authenticate the remote application SQL access to DB2.
One potential source of access abuse would be where an individual with administration privileges on the distributed application server would use the application server credentials and attempt to establish a local connection (via TSO for example) and attempt unauthorized access. By correlating the combination of IP address and credentials, these local connection attempts can be detected.

**Activity Monitoring Solution Characteristics and Attributes**

The following is a list of different forms of auditing capabilities and characteristics that the typical customer might employ:

- **Privileged User Monitoring**—This is where a population of users with elevated access privileges, typically identified by RACF or 3rd party authorization ID, is monitored for ALL activity performed in the DBMS by the targeted users. This approach to monitoring allows for the construction of an audit trail that verifies that only administration related activities are performed, and allows for detection of data access outside the business need-to-know. Some customers also view the use of certain database query tools as activity that requires detailed inspection, so activity associated with the use of these tools would also be included in the privileged user monitoring scenario. Guardium administrators, based on observed access patterns, are able to construct “white/black lists”, as part of the monitoring profile, which describes the population of these users/packages/connections. The privileged user monitoring approach typically represents the least intrusive amount of overhead, results in a manageable amount of audit data being collected, and presuming that all other application access is viewed as well vetted and secure, is the auditing approach used by most of our large z/OS customers. Guardium has developed a default report to help customers identify connections to become part of this “white or black list”, which is called “connection profiling”. This is a must to incorporate security best practices for databases.

- **Sensitive Object Monitoring**—This approach is where, after a thorough inventory of database artifacts, a list of objects (tables) is created which contain one or more data elements (columns) which are deemed as sensitive. Some customers with accurate and current data models find the creation of this inventory to be relatively straightforward, while others will devote a significant amount of time and resource to the creation of this sensitive object list. Once created, any access to a table contained in this sensitive object list would be subject to auditing. As described before, this access would encompass both application access as well as ad-hoc access events. For objects which are heavily accessed, particularly by application based SQL, this could constitute a large amount of SQL statement collection activity and subsequent network traffic. Guardium can help ease this task with a “classifying” feature which can help you identify where your sensitive data is located.

- **Comprehensive Monitoring**—As it sounds, this technique will collect EVERY SQL statement executed in the DBMS. While providing complete visibility, there are obvious considerations when attempting to implement a comprehensive approach on DB2 for z/OS. Due to the potentially large amount of data that is collected, one can anticipate higher application impact than the previously outlined approaches. Due to the amount of inbound event traffic, more appliance resource would be need to process and store the audit events. We see very few z/OS customers taking the comprehensive monitoring approach.
A typical z/OS implementation

While each customer has unique requirements, we see some common thread between implementations that could be viewed as “best practices” when constructing monitoring policy and associated profiles:

- **“Trusted Applications”—**While someone who is not well-versed in z/OS might take exception to this, most mature z/OS customers are comfortable with the level of protection that RACF provides, both from the OS as well as the middleware layer. As described above, application access, particularly through OLTP environments such as CICS or IMS, have been completely vetted by RACF. As a result, most customers will exclude, by DB2 attachment type, certain categories of access. When these attachment types are excluded, via a filter type in the established policy, it allows the STAP for DB2 to make a qualification decision extremely early in the SQL inspection process, and provides for the ability to exclude a large percentage of SQL Statements with a very low CPU impact. Most of our large z/OS customers exclude “trusted application” access from collection.

- **“Privileged User based profiles”—**Customers have a very good inventory over who has access to RACF credentials with elevated privileges. This information (inventory) can usually be derived from the RACF or 3rd party security product database, and/or from the security authorization tables of the DB2 catalog. Once captured, this information can be automatically fed into the appliance to automate the creation of a “Privileged User” Guardium group. UserID based filtering is also very efficient and represents the best performance characteristics with very low CPU impact.

- “SQL Generation Tools”—Packages and/or Plans associated with certain DB2 SQL generation tools should also be used for the formulation of policy rules. These tools allow access to tables with Ad-Hoc SQL generation. Some examples of these tools might include QMF, File-Aid for DB2, File Manager, etc. Many customers view the use of these tools in production DB2 environments as potentially hazardous, and monitor their use. Plan based monitoring uses the same approach as connection type and UserID filtering, and is also extremely efficient.

The above approach represents a very comprehensive “first step”, and for many of our large DB2 on z/OS customers, provides the necessary visibility to accomplish their auditing goals. These customers find that the combination described above strikes a good balance between having activity monitoring for all of the potentially damaging access, while at the same time experiencing a manageable impact to application performance. This balance provides a good risk reduction to their operational environment.

Another observation that we’ll make is that the choice of monitoring profiles can dramatically impact the performance overhead associated with the STAP. While we provide ample instrumentation on the STAP side whenever a new policy is “pushed” to the appliance, this can be somewhat after the fact, and performance surprises are not usually welcomed. As a result, we encourage close communication between the security function that is responsible for policy administration, and the database administration organization that is responsible for performance monitoring. Some of our larger customers will require that the two groups collaborate and agree on any changes, and even scheduling these during change control windows. This allows for an understanding of when changes are being implemented and allows for the performance of the system to be monitored closely, thereby minimizing any unforeseen changes to performance characteristics.
Any production implementation can require some sophisticated planning for topology decisions such as balancing appliance workload across the different sysplexes, planning for failover and redundancy, and ongoing care and feeding of the Guardium ecosystem. We strongly recommend that customers engage the IBM professional services Guardium practitioners in this process. They bring a high degree of deployment experience and a long history of successful Guardium implementations.

Performance – Technical Details Supplement
IBM InfoSphere Guardium STAP for DB2 on z/OS STAP collects and correlates data access information from a variety of DB2 for z/OS resources to produce a comprehensive view of business activity for security monitoring auditing. As the processing involves capturing SQL Statements on z/OS, there is typically interest and concern over the overhead of running the event collection on the mainframe. For reasons described below, it is difficult to be precise in estimating the expected overhead as many factors influence the processing costs. This interest and concern over performance is therefore frequently handled with a response of “It depends” as unsatisfying as that sort of response might be.

This section will first offer an overview of the STAP architecture. It will then describe the key elements of that architecture that should be considered in understanding the performance characteristics of the STAP in your environment.

IBM InfoSphere Guardium operates in a client-server environment using a combination of one or more agents, repositories, and data collectors. IBM InfoSphere Guardium STAP for DB2 on z/OS is comprised of several components that work together to collect audit DB2 activity for DB2 subsystems. The figure below shows these components.

The STAP agent/collector is responsible for the collection of audit data in a Guardium on z/OS environment. The IBM InfoSphere agent/collector runs as a started task on z/OS and acts as a “container” to run the various collectors that are appropriate to the specific type of system on which it operates. There are two basic types of data collected roughly characterized as low volume events (Commands, Utilities, etc.) and high volume events (SQL DML and DDL). The low volume events are collected using the standard DB2 trace capability with the appropriate trace classes enabled and routed to the DB2 Instrumentation Facility Interface (IFI). The high volume events are captured using specialized DB2 intercepts known as the Audit SQL Collector (ASC) without the need for DB2 tracing. Both IFI and ASC collected SQL Statements are streamed in real-time directly to the appliance. The agent/collector also maintains the necessary communications link to send information back and forth to the IBM InfoSphere server address space.
To describe the configuration for the STAP, if we take a typical environment where we have a single DB2 subsystem in the z/OS LPAR which is of auditing scope, the following components will be needed:

- A single MASTER address space, which is only used as a control block anchor point and consumes no CPU, and has no system resources allocated.
- A single Agent/Collector address space, which is responsible for the collection and management of the IFI based information, is responsible for intercept placement and SQL inspection within the DB2 address space.

If we were to introduce a second monitored DB2 subsystem into the previously described z/OS LPAR, we would see the need for these additional components:

- A second Agent/Collector address space

**Key performance considerations**

The STAP is responsible for gathering the DB2/z SQL requests and filtering those requests against policies that have been established by the STAP administrator. If the policy matches the content of the SQL, it is passed to the Guardium appliance for collection, analysis and reporting. Several functions are performed by the STAP, each of which contributes to the performance overhead:

**Data Gathering**
As an SQL statement is processed by DB2, the STAP ASC intercepts will capture that SQL and a variety of necessary additional fields, to characterize the event for downstream processing. It is important to consider that all SQL statements must be at least minimally inspected by the STAP to see if they are SQL Statements that are of interest. Thus, even if you have a need to capture very few events for downstream processing (e.g. only SQL from a single user), all SQL will need to be minimally inspected to be able to harvest those of interest (those of that user). The performance overhead of the data gathering will then be correlated to the rate of SQL that passes through DB2.

**Data Filtering**
Once an SQL event is gathered by the STAP intercepts, it needs to be inspected and filtered against the configuration policies that have been established in the collection profile. Only those SQL Statements of interest will be sent on to the Guardium appliance. Filtering is performed at two levels or “stages”. The STAP can perform some filtering very efficiently in the DB2/z address space using STAP stage 0 filtering. Examples of these very efficient stage 0 filters are those for users or plans. Any filtering that cannot be done by a stage 0 filter is performed in a stage 2 filter in the STAP address space (the Agent/Collector) outside of DB2/z. Examples of stage 2 filters are those that operate on an object (such as a table). To minimize overall overhead on the LPAR, the filtering policies should, whenever possible, be targeted at use of stage 0 filtering criteria. However, recall that these stage 0 filters are performed in the intercepts running in the DB2/z address space, so stage 0 filtering will be charged as DB2 class 2 processing. Should there be a need to minimize DB2 class 2 costs, it is possible to turn off stage 0 filtering. However this means that all filtering of gathered SQL Statements will be done in stage 2 (in the Agent/Collector address space) which is less efficient overall and would negatively effect the total overhead of the STAP in the LPAR.
**Data Movement**

As the SQL Statements are gathered (and potentially filtered by stage 0 filtering) the resulting events need to be moved out of the DB2 address space to the Agent/Collector (and this would then appear in DB2 class 2 times). This involves a memory move from the DB2 address space, above the bar, to be accessed by the STAP ASC address space. One benefit of stage 0 filtering is the reduction in the number of SQL Statements that might qualify to move to the ASC and the resulting reduction in memory moves of the data. With no stage 0 filtering, all the captured SQL Statements will need to be moved in memory. Once in Agent/Collector, stage 2 filtering will be applied where necessary and the resulting events are streamed to the Guardium appliance over TCP/IP. If the filtering is extensive, potentially few events will be streamed, limiting the performance overhead of the TCP/IP transfers. Note that if there is no filtering (stage 0 nor stage 2), the effect is to stream all the SQL Statements processed through DB2 over the network to the Guardium appliance.

This can result in significant TCP/IP processing and the resulting CPU overhead. Stage 2 filtering cost and preparing the data for streaming will be borne by the STAP Agent/Collector task (not DB2). Note that stage 2 filtering is eligible for zIIP processing should a zIIP processor be available and have capacity to have this work dispatched to the zIIP subject to the workload manager policies. The TCP/IP cost to do the transfer (stream) to the appliance is borne by the TCP/IP task, but that work is also zIIP processing eligible.

**Conclusion**

Significant control of the STAP overhead can be made through careful selection of filtering criteria and type, limited to only that which is necessary for your security and compliance requirements. The overall volume of SQL events per second is also a key determinant of the overhead of the STAP. A workload that has heavier weight (complex) SQL and thus fewer per second, as in a BI environment, would show less overhead as fewer SQL statements would be processed. In all cases, regardless of STAP configuration and filtering criteria, consider that all SQL events must be inspected by the STAP to determine if they are within the scope of interest.
For more information
To learn more about the InfoSphere Guardium solutions for System z, please contact your IBM representative or IBM Business Partner, or visit the following website:
ibm.com/software/data/db2imstools/solutions/information-governance.html

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We’ll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit:
ibm.com/financing

© Copyright IBM Corporation 2013
IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
August 2013

IBM, the IBM logo, ibm.com, DB2, RACF, InfoSphere, z/OS IMS, and Guardium are trademarks of International Business Machines 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 on the web at “Copyright and trademark information” at ibm.com/legal/copytrade.shtml

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

Please Recycle