IBM Security Guardium for zOS Overview and Best Practice

Roy Panting
CTO, Guardium for zOS

Vikalp Paliwal
Offering Manager
Guardium for zOS

Chris Born
Rocket Software

Julie Bergh
Rocket Software

IBM **Security**

IBM

Agenda

Guardium for ZOS

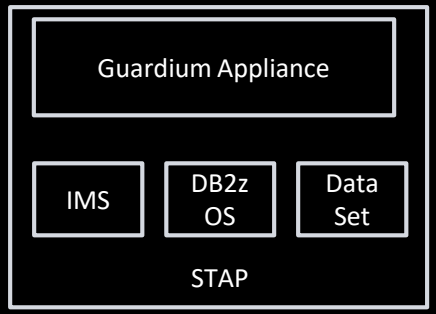
- Overview - Guardium components for z/OS
- Data collection
- Collection profile policy
- Performance

Guardium zOS – Solution Overview

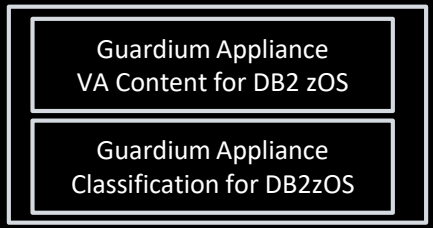
Guardium Data Protection for zOS

Guardium Encryption for zOS

Guardium Vulnerability Assessment for DB2 zOS



- Encryption for DB2zOS and IMS



- STAP agents feeds to Guardium Appliance
- Reporting, Alerts, Policies and Integration in Appliance

Understanding Guardium

- Guardium Data Protection intercepts incoming database activity
- A policy determines if this activity needs to be stored
  - If it does, it is sent to a repository that is off the mainframe
- Think of it as a video recorder that is watching activity occurring on a database

Guardium Use Cases or “Why Guardium?”

Compliance and Privacy Regulations

Require some method of identifying what is occurring on the systems

Privilege User Monitoring

Highly privileged users are a risk exposure

User Access

Compromised credentials and expanding authorization are a major source of data breaches

Audit Reporting

Auditor questions need to be answered quickly

….to address risk in the organization

Guardium's Evolution

- Originally, Guardium focused on compliance requirements
- Significant attention on filtering non-essential activity
- Auditing expanded from users to users and objects and beyond
- Architecture enhancements improved performance
- Requirements are expanding to trust no one

Guardium meets requirements for the future

- Efficient data gathering is key
- Minimal impact on MIPS
- Changing internal and external security requirements change what and how the data is collected and processed
- Advancing from compliance to security analytics
- Identification of unusual activity using machine learning
- So, how do can we do more with less?

Today’s presentation – DB2 STAP

- **We will Focus on DB2 STAP**
  - How can Guardium efficiently meet your organizations security requirements?
  - How can Guardium be tuned for optimization?
  - How will understanding the architecture provide an understanding of efficiency?
  - How can simple changes to the policy impact efficiency?

Guardium components for z/OS

Source: Add data source citation (name, author, date)

Collect activity for Db2 Subsystems on z/OS

DB2 S-TAP on Z

Collects data access information from a variety of DB2 resources to produce a comprehensive view of business activity for auditors

Collection based on collection profile policy, filtered at collection point

Authorization IDs, Objects, Plans, Programs, connection information etc.

Collection via shared DB2 Subsystem intercepts

TCP/IP stream audit events to Guardium System

Guardium System (aka Appliance, Collector)

- Policy definitions
- Hardened hardware or virtual appliance to securely store audit events
- Reporting and audit processes (workflow)
- Alerting
- Analytics

Data Collection

Source: Add data source citation (name, author, date)

Audited events collected

Security Guardium S-TAP for DB2 collects and correlates the following types of data

- DML (Data Manipulation Language)
  - Modifications(changes) to an object (SQL UPDATE, INSERT, DELETE)
  - Reads of an object (SQL SELECT)
- DCL (Data Control Language)
  - Explicit GRANT and REVOKE operations
- DDL (Data Definition Language)
  - CREATE, ALTER, and DROP operations against an object (such as a table)
- Assignment or modification of an authorization ID
- Authorization attempts that are denied because of inadequate authorization
- Utility access to an object (IBM utilities only)
- DB2 commands entered, including which users are issuing specific commands
- User selected DB2 negative SQL events
- Commits and Rollbacks events (optional - config parm controlled)

Data Gathering

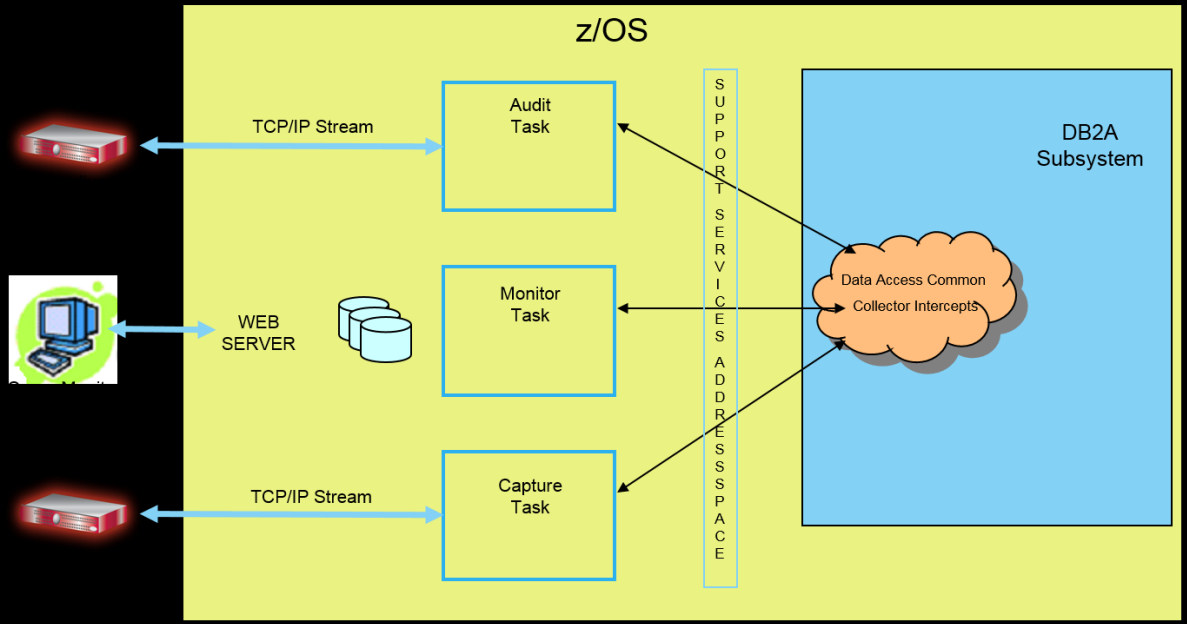
The DB2 S-TAP is responsible for gathering the DB2 SQL requests and filtering those requests against collection policies that have been established by the Guardium/Audit administrator

- As an SQL statement is processed by DB2, the S-TAP intercept, captures the SQL and a variety of additional fields to characterize the event for downstream processing
- All SQL statements must be minimally inspected to determine if they are events of interest based on collection policy
- For all SQL collection, DB2 S-TAP on Z uses the **IBM DB2 Data Access Common Collector for z/OS** (CQC)

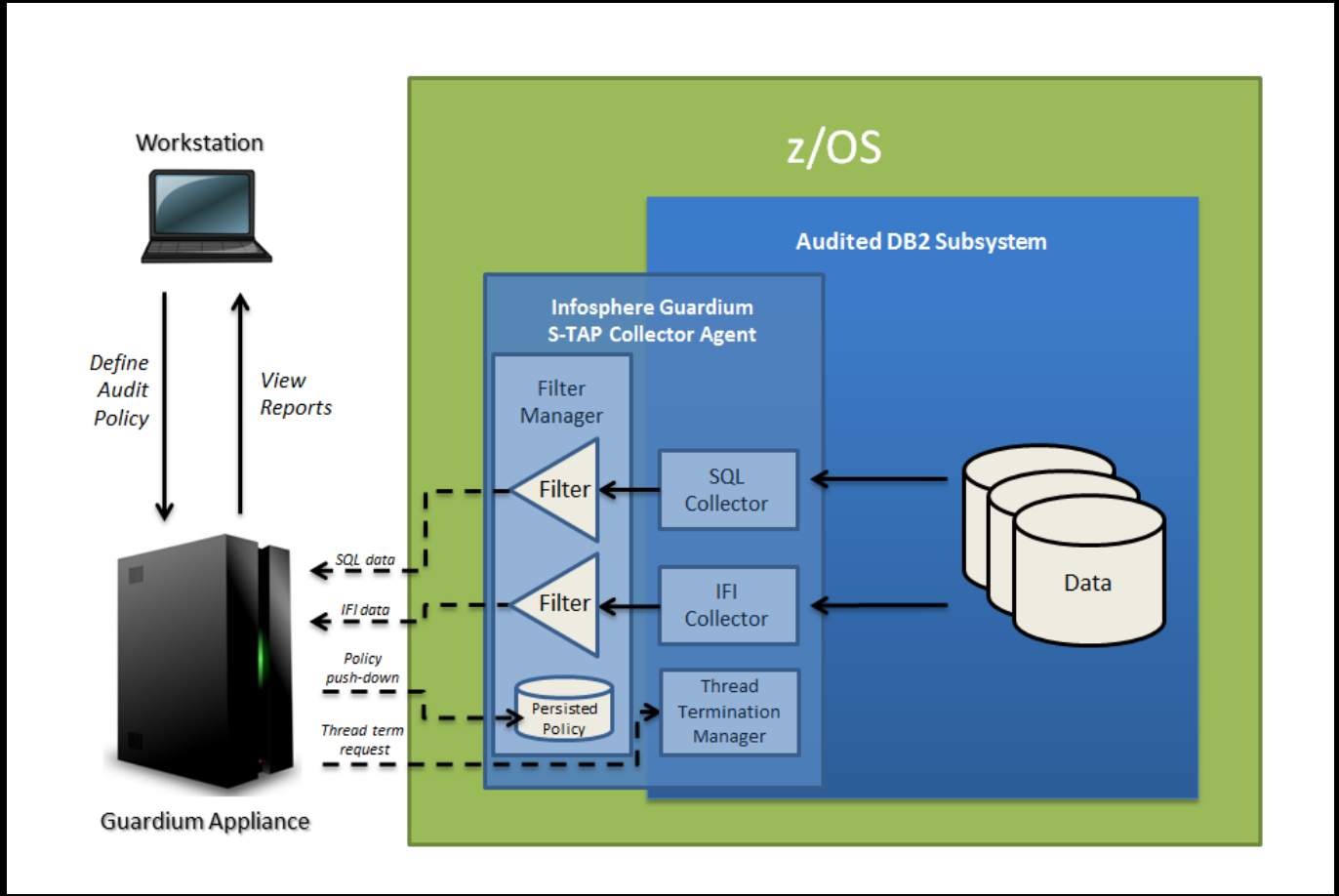
IBM DB2 Data Access Common Collector for z/OS (CQC)

IBM DB2 Data Access Common Collector for z/OS is a delivery vehicle for common collector technology leveraged across multiple IBM offerings

- CQC is a prerequisite product for IBM DB2 Query Monitor for z/OS, IBM Security Guardium S-TAP for DB2 on z/OS and IBM InfoSphere Optim Workload Replay for DB2 for z/OS



Architecture



Collection Policy (aka Collection Profile Policy)

Source: Add data source citation (name, author, date)

Enable collection of specific event types

SELECT/UPDATE/INSERT/DELETE (SUID), CREATE/ALTER/DROP, SET CURRENT SQLID, DB2 COMMANDS

- Collection is enabled through the presence of a Stage 1 filter or a non-blank wildcard value in the **Object** field of the rule

GRANT/REVOKE, DB2 UTILITIES, FAILED LOGINS (83/87)

- Collection is enabled via like entry in COMMAND setting

NEGATIVE SQLCODES

- Collection is enabled through the presence of a negative SQLCODE list

COMMIT/ROLLBACK

- Collection is enabled by including COMMIT_ROLLBACK setting in policy rule.

Event types and filtering

Event type	Filtered?
SELECT/UPDATE/INSERT/ DELETE (SUID)	Yes
CREATE/ALTER/DROP	No
GRANT/REVOKE	No
SET CURRENT SQLID	No
DB2 COMMANDS	No
DB2 UTILITIES	No
FAILED LOGINS (83/87)	No
NEGATIVE SQLCODEs	No
COMMIT/ROLLBACK	No

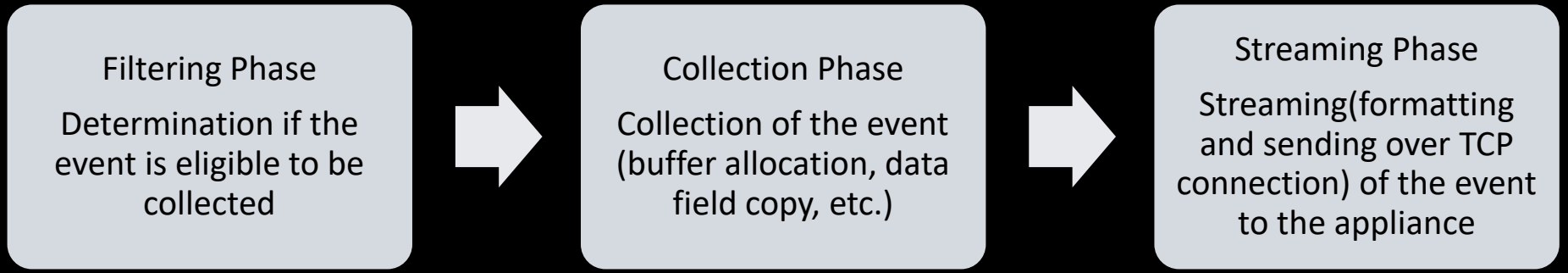
Collection Rule logic

- Collection of SELECT/UPDATE/INSERT/DELETE (SUID) audit events
  - The following policy rule fields are and-ed for collection logic
    - » CONNTYPE (NET PRTCL) Specifies the appliance connection type to DB2
    - » Plan Name (APP.USER, PLAN=) Specifies a valid DB2 plan name
    - » Program Name (APP.USER, PROG=) Specifies a valid DB2 program name
    - » Primary AUTHID (OS USER) Specifies the original operator user ID that is used to connect to DB2
    - » Current SQLID (DB USER) Specifies the primary AUTHID that is used for authorization within DB2
    - » WSNAME (CLIENT INFO, WKSTN=) Specifies a valid user workstation name
    - » WSTRAN (CLIENT INFO, APPL=) Specifies a valid program (or user workstation transaction)
      WSUSER (CLIENT INFO, USER=) Specifies a valid user name
    - » OBJECT Schema.Table name
    - » DATABASE

Performance

Source: Add data source citation (name, author, date)

Capturing of an Event is comprised of 3 main processing phases



Performance can be split into two areas.

CPU overhead per event.

- Filtering + Collection + Streaming

Total CPU overhead for the LPAR.

- Total number of events streamed.

Focus on filtering overhead and total number of events streamed.

- Fewer events streamed results in less Collection and Streaming overhead.
- May come at cost of complicated filtering.

Filtering overhead – Stage 1 (Thread level) filter

Once an SQL event is gathered by the SQL intercept, it needs to be inspected and filtered against the configuration policies that have been established in the collection profiles

Filtering is done very efficiently in the DB2 Address space using S-TAP compiled filtering.

Filtering is performed at two levels or “stages”

- Stage 1 filtering is the most basic and efficient form of filtering where the user can filter based on all filter fields except objects and Dbnames.
  - Filtering is at the thread level. If the fields do not change from event to event, filter evaluation is avoided.

Filtering overhead – Stage 1 filter (Continued)

The following fields would be Stage 1 filtering criteria

- CONNTYPE (NET PRTCL) Specifies the appliance connection type to DB2
- Plan Name (APP.USER, PLAN=) Specifies a valid DB2 plan name
- Program Name (APP.USER, PROG=) Specifies a valid DB2 program name
- Primary AUTHID (OS USER) Specifies the original operator user ID that is used to connect to DB2
- Current SQLID (DB USER) Specifies the primary AUTHID that is used for authorization within DB2
- WSNAME (CLIENT INFO, WKSTN=) Specifies a valid user workstation name
- WSTRAN (CLIENT INFO, APPL=) Specifies a valid program (or user workstation transaction)
  WSUSER (CLIENT INFO, USER=) Specifies a valid user name

Filtering overhead Stage 2 (Statement level)

Stage 2 filtering occurs if rule contains objects or DBName filters

- If a rule contains an object or DBName filter, it maybe subjected to ‘Stage 2’ or Statement level filtering.
- Stage 2 filtering is evaluated if

  The rule contains Stage 1 filters and the event passes the stage 1 filters

  Or

  The rule does not contain Stage 1 filters.

If all rules in the active collection profile are able to be filtered in Stage 1, then no Stage 2 filtering will occur

This will result in CPU reduction for the filtering process.

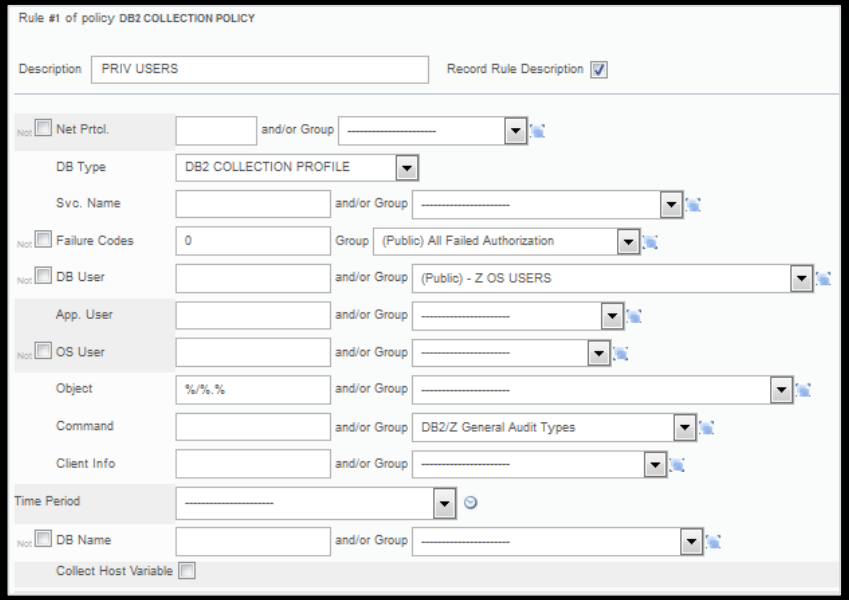
Performance - scenarios

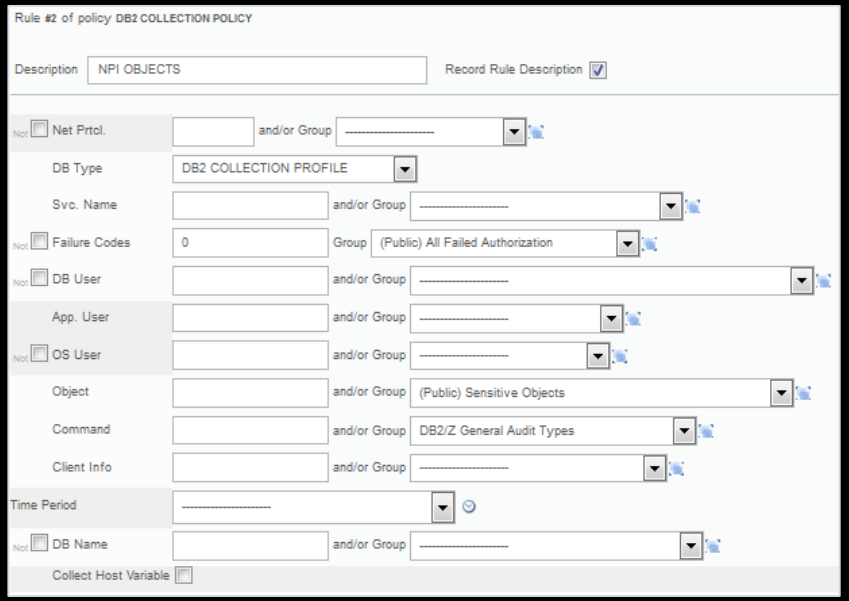
- **Scenario 1: No filtering, collect all activity.**
  - Pros: Policy is simple to construct, resulting filter CPU Overhead is minimized. All events streamed to appliance.
  - Cons: Collection and streaming phases are driven for every event resulting in increased CPU, Significant amount of data streamed to appliance which will need to be processed.
- **Scenario 2: Complex filtering, collect minimal activity.**
  - Pros: Only wanted events streamed, minimal event processing required at appliance.
  - Cons: Policy is potentially complex, too much data is filtered out prior to streaming to appliance, CPU Overhead of filter may be significantly greater than avoiding the collection and streaming phases.
- **Scenario 3: Middling policy, potentially collect more events than wanted**
  - Pros: Policy is simpler to construct, easier to understand, additional events streamed to appliance maybe wanted at some future date. Filtering CPU Overhead maybe minimized offsetting additional CPU overhead resulting from collection and streaming of unwanted events.
  - Cons: Avoidance of Collection and streaming phases not optimized resulting in potentially higher CPU than necessary. Unwanted events will potentially require filtering at appliance.

Filtering performance

Most common types of Collection Rules

- Privileged Users for all objects
- List of tables containing sensitive data touched by any Users



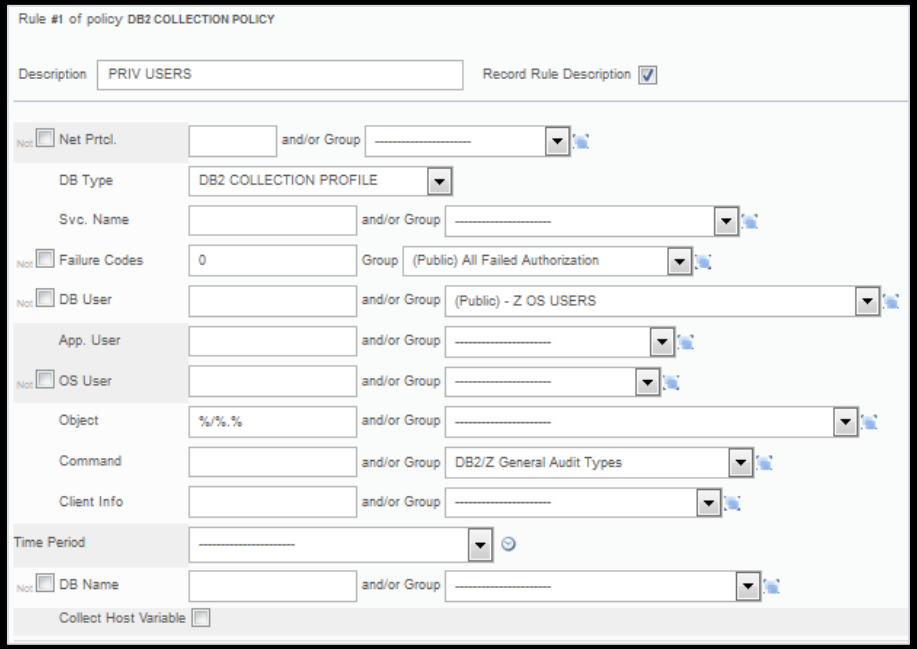


Filtering performance ART-101

If possible we want to modify our NPI OBJECT Rule to results in some work being doing in STAGE 1

To do this we can add a stage 1 filter criteria to rule

- Here we added a NOT list of PRIV USERS to NPI OBJECT Rule
- PRIV USERS rule will pick up any work by these users on NPI tables and make part of NPI OBJECTS rule Stage 1



SAMPLE collection policy rules concepts

Exclude CICS activity

- Single rule: Net Prtcl = NOT CICS
- Pro: Filtering cpu cost low while potentially having significant impact on removing volume of data
- Con: All other activity flows to appliance which can be significant if CICS activity low

Exclude CICS activity, collect Distributed activity for specific workstations

- Rule 1: Net Prtcl = NOT CICS
- Rule2: Net Prtcl = DRDA, Client Wrkstation = WS%
- Pro: Filtering cpu cost low while potentially having significant impact on removing volume of data, isolate collection of activity further to distributed activity with WS%.
- Con: Activity from other workstations not collected.

Exclude trusted users, Include activity for PLAN=DSNTEP2 and Object(Table) Filter.

- Rule 1: DB User = NOT ADMIN1, NOT ADMIN2, NOT ADMIN3
- Rule2: Object = %/REGION1.RECEIVABLES
- Pro: Structurally simple.
- Con: ALL activity is analyzed, potentially resulting in increased CPU Usage.

Control collection of HOST Variables

Many customers

- Do not need to collect host variables and want to reduce any performance impact
- Host variables could contain NPI data values that we don't want 'externalized'

Defaults:

- Prior to V10 default: host variables were collected by default
- V10 default changed: Will not collect host variables

To include them:

- Specify, on a per-rule basis, whether host variable information will be sent to the appliance for activity that matches that rule

If no rules in policy request HOST Variables we will see some reduced overhead

Thanks You

Questions

Next Tech Talk – Guardium for zOS - Troubleshooting