Agenda

- Why are audit records of Database Operations required in some cases? And why is collecting them difficult?
- Given the requirement what technical alternatives are available for database audit data collection?
- What issues should be considered in choosing among these alternatives?
  - Review and explain the alternatives above in light of the issues that need to be considered
- Beyond collection technology choice what other requirements should be considered
Why Audit: blame the government?

Legal + Regulatory + Industry = Compliance
Why Audit: *blame the DBMS vendors?*

Scalability, Recoverability, Availability, …
Why Audit: blame the DBMS vendors?

Scalability, Recoverability, Availability, ...

Auditability?
Why Audit: *blame ourselves*?

Most of our important, sensitive and private data is in the database. We MUST audit access to the database. We MUST audit usage data. We MUST audit maintenance/control/admin usage. If we don’t, we will not be in \textit{COMPLIANCE}.

Hey – audit the database

What do you need to audit?

Hmmm… Well – audit all reads of data, all writes, all changes to the database, who did it, from where, when, how….

Are you serious?

OK- audit just config changes
No wait – also schema changes
..And – access to sensitive data. And…
Why Audit: Plenty of Blame to go around

- Sarbanes-Oxley
- HIPAA
- GLBA
- SB 1386
- SAS70
- USA Patriot Act
- Moody’s
- 21CFR11
- Basel II
- ...
Database Auditing Motivation

Critical Issues:

- Passing the Audit
- Reduce cost of IT controls
- Regulatory and Industry Compliance
- Less DBA Workload also less insider risk
- Resulting in ...
  - Mandated Record Keeping for ...
  - Operations on Sensitive Data and Privacy
  - Forensic Analysis
Agenda

• Why are audit records of Database Operations required in some cases? And why is collecting them difficult?

• Given the requirement what technical alternatives are available for database audit data collection?

• What issues should be considered in choosing among these alternatives?
  – Review and explain the alternatives above in light of the issues that need to be considered

• Beyond collection technology choice what other requirements should be considered
Activity Monitoring: 
*Database Types & Versions*

**Audited Systems**

- **SQL Server 7, 2000 & 2005**
- **Oracle 8, 9i & 10g**
- **Sybase 11, 12 & 15**
- **DB2 “UDB” 8.1, 8.2 & 9**

**Audit Trail Repository**

- **Policy Adherence**
- **Compliance Reporting**
- **Anomaly Detection**

Oracle, SQL Server or DB2
What High Level Alternatives Exist?

• Go back to paper?

• Application &/or database schema modification
  – Shadow audit tables with application functionality to load them
  – Database Triggers

• Native Database System Audit Capability
  
  
  DB2 Audit Facility, Oracle Audit, SQL Server Trace, SybSecurity, …

• Transaction Log Files

• Collecting Network Activity
The Road Less Traveled

• Application &/or database schema modification
  Shadow audit tables, triggers, ...

• Reasons not to use Triggers:
  – Significant performance overhead
  – Easy to circumvent (alter table disable trigger all)
  – Costly to maintain (schema changes, code changes)
  – Incomplete capture
    • Typically can only capture DML (not DDL and not SELECTS)
What High Level Alternatives Exist?

• Go back to paper?
• Application &/or database schema modification
  *Shadow audit tables, triggers, …*
• **Native Database System Audit Capability**
  *DB2 Audit Facility, Oracle Audit, SQL Server Trace, SybSecurity, …*
• Transaction Log Files
• Collecting Network Activity
Native Database Audit Tools

• What is it?
• How does it work?
• Strengths and Weaknesses
• Why is anything other then this needed?
Native Database Audit Tools

- Event systems useful for performance and audit monitoring
- Generally under the control of the DBA
- Usual output is to a “local” table or text file
- Often comprehensive but in no case complete
- Performance wise you get what you pay for
- Output is hard to make unimpeachable (trustworthy)
- Usually Low Ease of Use
Native Database Audit Tools

- **Strengths**
  - DB Operation coverage
  - Supported by Database Vendor
  - Comes with the database

- **Weaknesses**
  - Impeachability (AKA will the auditor trust the records?)
  - Performance for anything that happens frequently
  - Data impact records, (what changed, how did it change, what data did the user see, …)
  - Specific to the data container (the DBMS system) and not to the data contents
  - Incomplete
What High Level Alternatives Exist?

- Go back to paper?
- Application &/or database schema modification
  
  *Shadow audit tables, triggers, …*

- Native Database System Audit Capability
  
  *DB2 Audit Facility, Oracle Audit, SQL Server Trace, SybSecurity, …*

- **Transaction Log Files**

- Collecting Network Activity
Transaction Log Post Processing

• When we say “Transaction Log” what do we mean?
• Strengths and Weaknesses
Database Transaction Log

- A transaction log is a “write ahead” journal of all activity that alters the state of the database
- Binary file, platform specific, internals undocumented
- A database always writes to the log even when the database is in circular (non-archived) mode
- A database writes undo and redo records to the log for:
  - Roll forward recovery
  - Transaction atomicity (roll back to a known state)
- An on-line transaction log is backed up to archived files
- The archived log files may serve as a rich source for audit activity
Example of a Transaction Log Record

Record 887
Log Page Offset = 04CD = 1229
Log File Offset = 0001D4CD = 120013
Record LSN = 0000 0428 34CD
Record Size = 0084 = 132
Record Type = 4E = Normal
Log Header Flags = 0002
Record TID = 00000000006F2
Back Pointer LSN = 0000 0000 0000
Originator Code = 01 = Data File Manager
Function ID = 78 = 120 : UPDREC_DP
Pool ID = 2 Object ID = 6
RID = 00000004 Pool Page = 0 Slot = 4
Update Log section = Undo section
oRecType = NORMAL
oTotRecLen = 38
TRHEAD Rec Type = FIXEDVAR
# Bytes In Fix Length Data = 14

Old Data:
0E 00 03 00 64 00 00 00 00 11 00 0D 00 00 69 62  *....d..........ib*
6D 43 72 75 65 6C 6C 61 20 42 6C 76 64 2E  *mCruella Blvd. *

Update Log section = Redo section
reclen = 38 free_sp = 2744
recoff = 2846
oRecType = NORMAL
oTotRecLen = 38
TRHEAD Rec Type = FIXEDVAR
# Bytes In Fix Length Data = 14

New Data:
0E 00 03 00 64 00 00 00 00 11 00 0D 00 00 69 62  *....d..........ib*
6D 31 20 57 61 6C 6C 20 73 74 72 65 64 74  *m1 Wall street *
What is in the Transaction Log?

- All DML operations
  - For row Insert & Delete, all column values are generally present
  - For column Update, before/after values for all modified columns are generally present
  - Status (success or failure)
  - Time stamp / Transaction ID
- On some platforms
  - All DDL operations
  - Application name shown on operation records
  - Session/UserID Identifiers shown on operation records
  - SQL Text
  - Client host name (or IP address)
What is generally **Not** in the Transaction Log?

- Read / SELECT operations
- Failed operations
- Unlogged operations

*example: SQL Server Simple Recovery Mode*
Transaction Log - Strengths

- Small performance impact database does the work in advance
  - Reading an OS file, outside of the database
  - Potentially using a machine that is not the DBMS Server
    
    **Work shifting**
    
    - Potentially at a time well after when the operation happened
      
      **Time shifting**

- Appropriate for auditing privileged users
  
  - Logs complete activity
  
  - Hard to circumvent

- Appropriate for auditing local access and Stored procedure contents
  
  - Shared Memory, named pipes, … connections

- Appropriate for data modifications
  
  - Provides a full audit trail of changes to column data

- Resilient to system failures
  
  - Audit data is never lost; because it uses the same recovery mechanism as the DBMS

- Not affected by network encryption
Transaction Log - Weaknesses

- Unsuitable for implementing privacy auditing
  - Selects are not in the transaction log
- Unsuitable for real-time alerting
  - Log reading is typically batch oriented that runs during scheduled times
- May not be deployed on databases that run in (non-archived log mode)
- Log files are binary and direct access is not a “supported” interface, hard to write your own
- Incomplete
What High Level Alternatives Exist?

• Go back to paper?

• Application &/or database schema modification
  
  Shadow audit tables, triggers, …

• Native Database System Audit Capability
  
  DB2 Audit Facility, Oracle Audit, SQL Server Trace, SybSecurity, …

• Transaction Log Files

• Collecting Network Activity
Network Capture

• When we say “Network Capture” what do we mean?
• Issues
• Strengths and Weaknesses
Network Capture

- A network appliance only
  - Resides on the network between the database and the users
  - Captures TCP traffic
  - Extracts SQL text and stores in a file backed cache
- Can be either a network appliance and/or host based software agents
How the Internet Works

- The Internet is “dumb”
  - All connections are initiated by the connected client server systems at each end
  - All data synchronization and retransmissions are controlled by the end systems
  - The network merely pushes packets along from hop to hop
  - If a packet is dropped in the network the end points will detect it and retransmit.
Sniffers Listen to Packets as They Go By

- Packets do get dropped in networks
  - Sniffers can drop packets if there is local congestion
    - Local congestion can be caused by non audit traffic
  - If the sniffer in the middle drops a packet it may not be retransmitted because the endpoints may not drop the same packets
What is in a TCP Packet?

- A SQL Text command, exactly as it was sent to the database
- The SQL batch must be parsed to obtain:
  - Object names (affected tables)
  - Individual SQL commands (e.g., nested selects)
  - Prepared statements must be correlated, if possible
- Other fields must be obtained from a previous login request (if not encrypted)
  - User name
  - Application id
  - Client host
Network Capture - Strengths

• Suitable for privacy
• Very small additional load performance impact
  \(\text{Same work and time shifting possibilities as log reading}\)
• Suitable for basis of real-time alerting
• Less Integration and therefore complication with the DBMS itself
• If you are a DBA the network topology and network security are “Somebody Else's Problem”
Network Capture - Weaknesses

- Difficulty auditing privileged users and or local host access
- Difficulty handling session encryption
  - SSL, IPSec
  - Encrypted login packets
- Inability to audit server side logic
  - Cannot capture activity of stored procedures, triggers packages
- Inability to track changes to data values
- Not resilient to failure
  - If a network capture component fails to record traffic, the audit data is lost and cannot be recovered
- Hard to do, not a supported interface
Agenda

• Why are audit records of Database Operations required in some cases? And why is collecting them difficult?

• Given the requirement what technical alternatives are available for database audit data collection?

• What issues should be considered in choosing among these alternatives?
  – Review and explain the alternatives above in light of the issues that need to be considered

• Beyond collection technology choice what other requirements should be considered
The Data Compliance Spectrum

- Highly tuned production system
- Very high service level requirements
- No stored data values

- Multiple DBs
- Migration of servers
- Rolling out encryption

- Financial and operational data
- Past privileged user access issues
- Before and after value audit trail required

Companies often blanket the spectrum
## Mapping Technology to Real-World Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Transaction Log Reading</th>
<th>Network Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>All my database activity is encrypted (i.e. SSL, IPSec, etc…)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I need to monitor <em>all</em> forms of privileged user access</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I need to see the results from stored procedures &amp; triggers</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I need to see the data impact for SOX compliance and provide “before and after” values</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I need real-time security alerts</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I need to capture a high volume of SELECTs for PCI compliance</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I cannot have any impact on the database</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>I don’t have database logging enabled and I cannot change this</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Over all Conclusion

- None of the alternatives are comprehensive
- Avoid use of Native Audit Tools for operations that happen frequently
  - Log Reading is best for DML auditing
  - Network Capture is best for Select/Read auditing
**Agenda**

- Why are audit records of Database Operations required in some cases? And why is collecting them difficult?
- Given the requirement what technical alternatives are available for database audit data collection?
- What issues should be considered in choosing among these alternatives?
  - Review and explain the alternatives above in light of the issues that need to be considered
- Beyond collection technology choice what other requirements should be considered
Beyond Collection Method

- Audit Policy
- Deployment Configuration
Audit Policy Configuration “What Records to keep?”

- Policies can conditionally qualify otherwise raw database events to:
  - Produce a more effective and accurate audit trail
  - Identify violations of policy
  - Classify events based on context
  - Issue alerts to initiate immediate response
- Definition of a complete, specific, database monitoring filter; abstracted away from details of database type and collection method.

<table>
<thead>
<tr>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>All privileged user activity: Audit</td>
</tr>
<tr>
<td>If activity is outside of maintenance window: Critical</td>
</tr>
<tr>
<td>If activity results in update to SOX scope data: Alert</td>
</tr>
<tr>
<td>If host or application is determined to be un trusted</td>
</tr>
</tbody>
</table>

POLICY
Consolidated definition of acceptable use
Audit Policy Rules?

- Supported attributes that can be used:
  - User ID
  - Application Name (Oracle and SQL Server data sources only)
  - Object (table, column, view, stored procedure)
  - Database Operation (insert, update, delete, grant, select …)
  - Client Host Name
  - Date Time range

- Supported Actions:
  - Collect, record the audit record in the Audit DB repository, mark the audit record with the policy name that was “true”
  - Annotate the collected audit record with a severity level
  - Send an alert
Grant of new Privileges → Company (enterprise) policy

Changes to Stored Procedures → Division policy

DML and Selects on employee table → Department policy

… → Instance policy

… → Database policy

- Multiple Policies with distinct objectives collected concurrently
- Each collected audit record annotated with the policy or policies that triggered collection
- Each audit record prioritized
- Conditional immediate alerting
Multi-Layer Data Protection Framework

- Data Assets
  - DB Version, Patches & Configuration
  - DB Access Control
  - Host – OS Firewall, Version, Patches, Configuration
  - IDS – Deep Packet Inspection
  - Network - Firewalls & Access Control
Deployment Configuration

- **Audited System**
  - SQL Server 7, 2000 & 2005
  - Oracle 8, 9i & 10g
  - Sybase 11, 12 & 15
  - DB2 "UDB" 8.1, 8.2 & 9

**Audit Trail Repository**

- Policy Adherence
- Compliance Reporting
- Anomaly Detection

Oracle, SQL Server or DB2
Deployment Configuration

Assumptions:
- 8 database instances per DB Server
- DB2 per instance
  - 90 Trans./Sec
  - 500 Megabytes/DB
- Oracle per instance
  - 90 Trans./Sec
  - 1.50 Gigabytes/DB
- Sybase (per instance)
  - 40 Trans./Sec
  - 300 Megabytes/DB
- SQLServer per instance
  - 3 Trans./Sec
  - 50 Megabytes/DB

Operations Parameters:
- 141 transactions/second for first level sniffers
- 10 first level sniffers on an aggregation box
- 2 levels of aggregation

Diagram:
- Clients
- SQL Server
- DB2
- Oracle
- Sybase
- Building 1
- Building 2
- Building 3
- Building 4
- Building 5
- Campus 1
- Campus 2
- Campus 3
- Campus 4
- Campus 5

Legend:
- First level sniffers
- First level aggregation
- Second level aggregation
Summary

• Compliance presents multiple auditing requirements:
  – Privileged user, privacy, data auditing, completeness, separation of duties, etc.

• The requirements need to be implemented in a variety of existing production environments:
  – Performance sensitive systems
  – Systems that use network encryption
  – Warehouses that run in circular log mode

• No single technology alone can meet all auditing requirements.
• No single technology alone can fit all IT environments.
• An appropriate combination of native audit, transaction log reading, and network capture can be selected to match specific audit requirements and IT infrastructure.
Thank you! Questions?

Lumigent Technologies, Inc.

henry.parnell@lumigent.com
 Auditing Stored Procedure Example

create procedure sp_salary @sal int, @emp int as
update overtime set rate = @sal where empid = @emp
delete overtime where empid = @emp+1
go

declare @handle int
set @handle = NULL
exec sp_prepare @handle OUTPUT, N'@p1 varchar(80)',
N'update overtime set rate = @p1 where empid = 5', 1
exec sp_execute @handle, 100
go

declare @str nvarchar(500)
set @str = N'delete overtime where empid = 12'
execute sp_executesql @str

CREATE TABLE [dbo].[overtime] (  
    [emp_id] [empid] NULL ,  
    [empid] [int] NULL ,  
    [rate] [decimal](18, 0) NULL  
) ON [PRIMARY]
<table>
<thead>
<tr>
<th>EXECUTE PROCEDURE</th>
<th>127.0.0.1 / 127.0.0.1 (MSSQL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-03-02 20:14:11</td>
<td>ALL</td>
</tr>
<tr>
<td>2220000000009</td>
<td>hr.</td>
</tr>
<tr>
<td></td>
<td>sp_salary 100, 7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EXECUTE PROCEDURE</td>
<td>127.0.0.1 / 127.0.0.1 (MSSQL)</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>2007-03-02 20:17:59</td>
<td>ALL</td>
</tr>
<tr>
<td>2220000000014</td>
<td>hr.</td>
</tr>
<tr>
<td></td>
<td>declare @handle int set @handle = NULL exec sp_prepare @handle OUTPUT, N'@p1 varchar(80)', N'update overtime set rate = @p1 where empid = 5', 1 exec sp_execute @handle, 200</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EXECUTE PROCEDURE</td>
<td>127.0.0.1 / 127.0.0.1 (MSSQL)</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>2007-03-02 20:18:08</td>
<td>ALL</td>
</tr>
<tr>
<td>2220000000015</td>
<td>hr.</td>
</tr>
<tr>
<td></td>
<td>declare @str nvarchar(500) set @str = N'delete overtime where empid = 12' execute sp_executesql @str</td>
</tr>
</tbody>
</table>
### Auditing Stored Procedure Both NetCap and TLog 1of3

<table>
<thead>
<tr>
<th>ACTIVITY TYPE</th>
<th>Object Host / Instance / Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Time</td>
<td>Login Name</td>
</tr>
<tr>
<td>Activity ID</td>
<td>Policy Name DML Column Name: DML Old Value:</td>
</tr>
<tr>
<td></td>
<td>DML Object,... DML SQL</td>
</tr>
<tr>
<td><strong>EXECUTE PROCEDURE</strong></td>
<td>127.0.0.1 / 127.0.0.1 (MSSQL)</td>
</tr>
<tr>
<td></td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>hr.</td>
</tr>
<tr>
<td></td>
<td>sp_salary 100, 7</td>
</tr>
<tr>
<td><strong>UPDATE</strong></td>
<td>LUMIGENT-L3 / LUMIGENT-L3 (MSSQL) / hr.dbo.overtime SQL Query Analyzer on LUMIGENT-L3</td>
</tr>
<tr>
<td></td>
<td>intruder SOX_404_Audit_Policy</td>
</tr>
<tr>
<td></td>
<td>◇ empid : 7</td>
</tr>
<tr>
<td></td>
<td>rate : 10.5</td>
</tr>
<tr>
<td></td>
<td>hr.dbo.overtime</td>
</tr>
<tr>
<td><strong>DELETE</strong></td>
<td>LUMIGENT-L3 / LUMIGENT-L3 (MSSQL) / hr.dbo.overtime SQL Query Analyzer on LUMIGENT-L3</td>
</tr>
<tr>
<td></td>
<td>intruder SOX_404_Audit_Policy</td>
</tr>
<tr>
<td></td>
<td>◇ empid : 8</td>
</tr>
<tr>
<td></td>
<td>hours : 0</td>
</tr>
<tr>
<td></td>
<td>rate : 10.5</td>
</tr>
<tr>
<td></td>
<td>hr.dbo.overtime</td>
</tr>
</tbody>
</table>
### Auditing Stored Procedure Both NetCap and TLog 20f3

<table>
<thead>
<tr>
<th>ACTIVITY TYPE</th>
<th>Object Host / Instance / Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Time</td>
<td>Login Name</td>
</tr>
<tr>
<td>Activity ID</td>
<td>Policy Name</td>
</tr>
<tr>
<td></td>
<td>DML Column Name: DML Old Value:</td>
</tr>
<tr>
<td></td>
<td>DML Object,...</td>
</tr>
<tr>
<td></td>
<td>DDL SQL</td>
</tr>
</tbody>
</table>

**EXECUTE PROCEDURE**

2007-03-02 20:17:59
22200000000014

127.0.0.1 / 127.0.0.1 (MSSQL)

**UPDATE**

2007-03-02 20:17:59
22400000000002

**LUMIGENT-L3 / LUMIGENT-L3 (MSSQL) / hr.dbo.overtime**

<table>
<thead>
<tr>
<th>intruder</th>
<th>SQL Query Analyzer on LUMIGENT-L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOX_404_Audit_Policy</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>empid</th>
<th>rate</th>
<th>hr.dbo.overtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

```sql
declare @handle int
set @handle = NULL
exec sp_prepare @handle OUTPUT, N'@p1 varchar(80)', N'update overtime set rate = @p1 where empid = 5', 1
exec sp_execute @handle, 200```
<table>
<thead>
<tr>
<th>ACTIVITY TYPE</th>
<th>Object Host / Instance / Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Time</td>
<td>Login Name</td>
</tr>
<tr>
<td>Activity ID</td>
<td>Policy Name</td>
</tr>
<tr>
<td></td>
<td>DML Column Name:</td>
</tr>
<tr>
<td></td>
<td>DML Old Value:</td>
</tr>
<tr>
<td></td>
<td>DML New Value:</td>
</tr>
<tr>
<td></td>
<td>DML Object,...</td>
</tr>
<tr>
<td></td>
<td>DML SQL</td>
</tr>
</tbody>
</table>

**EXECUTE PROCEDURE**

2007-03-02 20:18:08  
22200000000015

127.0.0.1 / 127.0.0.1 (MSSQL)

**DELETE**

2007-03-02 20:18:08  
22400000000001

LUMIGENT-L3 / LUMIGENT-L3 (MSSQL) / hr.dbo.overtime intruder

SOX_404_Audit_Policy

empid : 12
hours : 0
rate : 10.5
hr.dbo.overtime
Thank you! Questions?

Lumigent Technologies, Inc.

henry.parnell@lumigent.com
Deployment Details – Network Capture

Monitoring “out of band” on a mirrored/span switch port

Monitoring on the DB server (no additional hardware required)