Best Practices für das Datenbank-Audit in Oracle 11g und 12c

Dr. Elke Fritsch
Muniqsoft GmbH
Unterhaching

Schlüsselworte
Audit, Unified Auditing, Standard Edition

Einleitung
Firmen, die mit personenbezogenen Daten arbeiten, sind gesetzlich zum Audit verpflichtet. In der Praxis sieht das allerdings häufig so aus, dass man im Vertrauen darauf, dass die Default-Audit-Einstellungen von Oracle schon alles Wesentliche abdecken werden, die Inhalte der Audit-Trails eine bestimmte Zeit aufhebt und dann löscht oder irgendwohin exportiert, ohne jemals einen Blick auf den Inhalt geworfen zu haben.

Und wenn ein entnervter DBA unter Oracle 11g kurz nachsehen will, welcher 🕷🌪💥☠🌊💣 an den Datenbankparametern gedreht hat und es wieder mal keiner gewesen sein will, wird er im Audit-Trail der Datenbank vergeblich danach suchen, selbst wenn er den Parameter `AUDIT_SYS_OPERATIONS` auf TRUE gesetzt hat. Unter Linux müsste er dazu unzählige aud-Files im Betriebssystem und unter Windows das Eventlog durchforsten.

Um wirklich das auditiert zu bekommen, was man gerne hätte und die Auswertung nicht allzu zeitraubend zu gestalten, muss man die Audit-Einstellungen und das Housekeeping der Daten manuell einrichten und testen. Auch die Übertragung eines fertigen 11g-Audit-Konzepts auf Oracle 12c ist eine Menge Arbeit.
Dieser Vortrag beschäftigt sich anhand eines fiktiven, aber nicht untypischen Szenarios mit den Schwachstellen der Default-Audit-Konfiguration in Version 11g und 12c und stellt Lösungsansätze für das Audit vor, mit denen man die Erfassung der Daten und die Auswertung vereinfachen kann.

Ausgangslage für das Audit-Szenario
Der neue, motivierte und experimentierfreudige DBA einer mittelständischen Firma soll ein neues Audit-Konzept umsetzen. Im Einsatz sind Oracle-Datenbanken der Version 11.2.0.4 (Standard Edition One) auf Linux und auf Windows.
Alle Angestellten der Firma loggen sich unter dem selben Account , aber von unterschiedlichen Rechnern aus ein und können entweder direkt (über SQL Developer) oder über ein APEX-Interface auf die Tabellen zugreifen. Der DBA-Kollege nutzt manchmal auch den Enterprise Manager. Die DBAs arbeiten als `SYSDBA` auf der Instanz, alle anderen haben die Rollen `CONNECT` und `RESOURCE`.

Auditiert werden sollen:

1. Änderungen (DML) an der Tabelle Kundendaten im Schema KUNDEN
2. DROP oder TRUNCATE-Vorgänge an Tabellen im Schema KUNDEN
3. Änderungen am PL/SQL-Code im Schema KUNDEN
4. fehlgeschlagene Login-Versuche
5. Für einen begrenzten Zeitraum alle Aktionen des neuen Werkstudenten, der eine neue Applikation in einem eigenen Schema (mit vollem Zugriff auf das Kundenschema) entwickeln soll.
6. Alle Aktionen des externen Consultants, der sich ebenfalls als `SYSDBA` auf der Instanz einloggt.
7. Stop, Starts, Änderungen an Parametern und Datenbankdateien
8. Rechte-Vergabe bzw. Entzug

Weitere Forderungen

- Bis jetzt wurde die Tabelle aud$ einmal pro Woche über einen DBMS_JOB-Aufruf geleert. In Zukunft sollen die Daten 120 Tage aufbewahrt und "zeitnah" ausgewertet werden.
- Alle Audit-Einträge sollen über eine zentrale View selektierbar sein, um die Auswertung zu erleichtern.
- Das Audit soll mit möglichst wenig Aufwand (bzw. Kosten) einzurichten sein.
- Es soll sowohl für Windows- als auch für Linux-Datenbanken umsetzbar sein und möglichst keine zusätzlichen Shell- oder Batch-Skripte für das Housekeeping etc. erfordern.

Probleme bei Einrichtung, Auswertung und Housekeeping des Audits unter 11g

- *Gewöhnungsbedürftige Syntax für das Einrichten der Audit-Optionen*
  Von den oben geforderten Audits wird nur ein kleiner Teil über die Default-Einstellungen abgedeckt. Es ist also viel Handarbeit nötig, wobei die Syntax der AUDIT-Anweisung einige Fallstricke enthält.
  So kann z.B. ein unvorsichtiges `SELECT TABLE BY <username>` den Audit-Trail exponentiell aufblasen, wenn man ein Tool wie den SQL Developer benutzt.

- *Überflüssige Informationen im Audit-Trail*
  Wenn man den Enterprise Manager konfiguriert hat, der sich im Sekundentakt bei der Datenbank anmeldet, führen die Default-Einstellungen dazu, dass der Audit-Trail größtenteils mit den Logon- und Logoff-Einträgen der User SYSMAN und DBNSMP gefüllt wird.
  Beim SYS-Auditing werden unter 11g alle Statements aufgezeichnet. Hier muss man den Overhead der Data-Dictionary-Zugriffe (recursive SQL) beim Auswerten herausfiltern.
  Auch der RMAN erzeugt eine Flut von Einträgen, wenn er sich als SYSDBA anmeldet.

- *Unterschiedliche Speicherorte der verschiedenen Audit-Informationen*
  Connects als SYSDBA bzw. SYSOPER, Stops und Starts der Datenbank werden per default auditiert (mandatory auditing). Alle weiteren Aktionen des Users SYS können durch Umstellung des Parameters AUDIT_SYS_OPERATIONS auf TRUE auditiert werden.
  Unter Unix werden die Informationen jedoch in das durch den Parameter audit_file_dest angegebene Verzeichnis geschrieben, unter Windows ins Eventlog.
  Um dieses Audit auswerten zu können, müsste man die Inhalte der *.aud-Files unter Linux bzw. die Oracle-spezifischen Inhalte des Eventlogs unter Windows parsen und in die Datenbank laden. Beides ist möglich, aber aufwendig.

- *Unzulänglichkeiten des Packages DBMS_AUDIT_MGMT*
  Das Package DBMS_AUDIT_MGMT ist umständlich zu bedienen, vor allem, wenn man sowohl in der Datenbank als auch im Filesystem aufräumen will. Um die Einträge im Windows Event Log muss man sich selber kümmern.

Vorschlag für ein "Unified Auditing" unter 11g

Wenn man den Parameter AUDIT_TRAIL auf XML setzt, werden alle zu auditierenden Statements des Standard, Mandatory, SYS- und Fine Grained Auditing (letzteres natürlich nur in der Enterprise Edition) im AUDIT-FILE-DEST-Verzeichnis als XML-Files gespeichert und können über die View V$XML_AUDIT_TRAIL gemeinsam ausgewertet werden, ähnlich wie die XML-Files des Alert.logs über die View V$DIAG_ALERT_EXT.
Die seit der Oracle-Version 10.2 verfügbare View DBA_COMMON_AUDIT_TRAIL vereint diese Einträge mit denen aus DBA_AUDIT_TRAIL und DBA_FGA_AUDIT_TRAIL, so dass auch die Inhalte der Tabellen AUD$ und FGA_LOG$ noch sichtbar bleiben.

Diese Lösung kann man durchaus als abgespecktes "Unified Auditing" für 11g bezeichnen. Die Auswertung und das Housekeeping werden dadurch wesentlich erleichtert. Auch die Performance des Audits leidet nicht darunter, dass man die Audit-Files im Betriebssystem speichert , eher im Gegenteil. Wenn man den XML-Audit-Trail vor Zugriffen durch übel gesinnte DBAs schützen will, muss man dafür sorgen, dass die Files in kurzen Abständen auf einen anderen Server außer Reichweite gesichert werden.

Im Vortrag wird gezeigt, wie man die oben genannten Anforderungen an das Audit mit dieser Methode, ein paar Views für die bequeme Auswertung und einem datenbankgesteuerten Housekeeping umsetzen kann.

Und wie sieht's unter 12c aus ?
Unsere mittelständische Firma wächst und gedeiht und denkt an einen Upgrade auf die Version 12c. Das unter 12c neu eingeführte *Unified Auditing* soll schneller, leichter einzurichten und zu verwalten und für alle Editions verfügbar sein.
Im Prinzip ja, aber ...

Die Standard Edition One (SEO) gibt es nur noch in der Version 12.1.0.1. Hier ist das Feature wegen des Bugs 17466854 (CANNOT SET UNIFIED AUDITING IN STANDARD EDITION) nicht zu aktivieren. Dafür gibt es einen Patch 17466854, den man aber nicht einspielen kann, wenn man seine Datenbank mit dem aktuellen Patchset versorgt hat. Der Bugfix wurde für den Patchset 12.1.0.1.5 (vom 14.10.2014) geschrieben und funktioniert nicht für höhere Levels.
Für SEO-Datenbanken unter Windows soll der Bugfix im Patchset 12.1.0.1.18 eingeschlossen sein.

Da die SEO allerdings sowieso ein Auslaufmodell ist, steigt unsere Firma doch lieber auf die Standard Edition 2 (SE2) um. Bei der Version 12.1.0.2 funktioniert der Umstieg auf Unified Auditing problemlos. Es macht sich jedoch deutlich bemerkbar, dass bei jedem Patch diverse Bugs gefixt werden. Man sollte also unbedingt die neuesten Patches einspielen !

Nach der Migration auf die Version 12c ist zunächst der sogenannte *Mixed Mode* aktiv, unter dem nur die Policies `ORA_SECURECONFIG` und zusätzlich `ORA_LOGON_FAILURES` (erst ab 12.1.0.2) aktiviert sind. Diesen beizubehalten, ist keine wirklich gute Option, vor allem hinsichtlich des Housekeepings und der Sicherheit der Audit-Daten gegenüber Manipulation durch den DBA.
Zudem gibt es einige wirklich nützliche zusätzliche Features beim neuen Unified Auditing, die man sich nicht entgehen lassen sollte. Dazu zählt vor allem die größere Sicherheit der Audit-Daten.

Was ist beim Umstieg auf Unified Auditing unter anderem zu beachten ?

- *Das sys-Audit ist nicht mehr das, was es einmal war*
  Änderungen am System, an den Audit-Einstellungen etc. werden im Unified Auditing immer auditiert, egal, ob man sich als `SYSTEM` oder `SYS AS SYSDBA`, `SYSOPER`, `SYSDG`, `SYSBACKUP` bzw. `SYSKM` anmeldet.
  Ansonsten wird der User `SYS` aber genauso behandelt wie Hinz und Kunz !!! Auch seine Selects erscheinen nicht mehr im Audit-Trail.
  Wenn man z.B. mitbekommen will, ob er sich per DML an irgendwelchen User-Tabellen vergreift, muss man dieses Audit explizit in einer Policy einrichten! Der Parameter `AUDIT_SYS_OPERATIONS` hat keinen Einfluss mehr.

- *Per Default werden nur failed logins auditiert*
  Dass nicht mehr jeder LOGON und LOGOFF registriert wird, macht den Audit-Trail zwar übersichtlicher, aber für Auswertungen des Audit-Trails hinsichtlich der Dauer der Sessions oder der Anzahl nicht korrekt beendeter Session (LOGOFF BY CLEANUP) muss man sich nun eigene Policies schreiben.

- *Die alten Audit-Optionen sind nicht 1:1 in den neuen Policies abzubilden*
  Ein `AUDIT PROCEDURE BY consultant;`
  unter 11g wird unter 12c zu
  ```
  CREATE AUDIT POLICY consultant_plsql_pol
  ACTIONS CREATE FUNCTION,
          DROP FUNCTION,
          CREATE LIBRARY,
          DROP LIBRARY,
          CREATE PACKAGE,
          DROP PACKAGE,
          CREATE PACKAGE BODY,
          CREATE PROCEDURE,
          DROP PROCEDURE;
  AUDIT POLICY consultant_plsql_pol BY consultant;
  ```
  Das ist eigentlich ein Vorteil, weil diese Syntax besser verständlich ist, aber die Policies können schnell unübersichtlich werden.
  Die Anforderungen an das Audit von Seite 1 lassen sich über die beiden Default-Policies und 3 zusätzliche maßgeschneiderte Policies verwirklichen. Zuviel Policies wirken sich eher schädlich auf die Performance aus.

- *Das neue Audit ist per Default "extended"*
  Während man unter der Version 11g den Parameter audit_trail auf `DB, extended` oder `XML, extended` setzen musste, um auch die SQL-Statement und Bindvariablen zu erfassen, ist das in Version 12c per Default der Fall. Das lässt den Audit Trail entsprechend schnell anwachsen, wenn die Statements etwas komplexer werden. Auch das obligatorische RMAN-Audit braucht einiges an Platz. Deshalb ist ein durchdachtes Housekeeping fast noch wichtiger als unter 11g.

Wie der (immer noch motivierte) DBA den Umstieg von der selbstgestrickten Unified Auditing Lösung unter 11g auf die Audit-Konfiguration unter 12c meistert und was ein vernünftig konfiguriertes Audit alles an Verbrechen gegen die Datenbank im allgemeinen und die guten Sitten im besonderen zutage fördern kann, erfahren Sie im Vortrag.

Kontaktadresse:
Dr. Elke Fritsch
Muniqsoft GmbH
Witneystraße 1
82008 Unterhaching

Telefon:	+49 (089) 6228-6789-49
Fax:	+49 (089) 6228-6789-50
E-Mail	elke.fritsch@muniqsoft.de
Internet:	www.muniqsoft.de