Upgrade to Unified Auditing – Auditing von 11g nach 12c

Sebastian Winkler
CarajanDB GmbH
Erftstadt

Schlüsselworte

Oracle Auditing, Security, Unified Auditing, Fine-Grained Auditing, Audit Vault, 12c

Einleitung

Mit dem Schlagwort „Unified" hat Oracle 12c einige schwerwiegende Eingriffe und Veränderungen bei der Datenbankauditierung vorgenommen. Diese Neuerungen bedürfen aber nicht nur unserer Aufmerksamkeit, wenn wir ein Upgrade von einer älteren Datenbankversion vornehmen. Denn ausgeliefert wird das neueste Release bisher immer im sogenannten „Mixed Mode". Das hat zunächst den Vorteil, dass alles „Alte" aus unserer 11er Welt noch funktioniert. Im neuen „Unified Audit Trail" konsolidiert 12c alle Audit Trails in eine Audit Trail Tabelle. Im „Mixed Mode" aber dann doch erstmal zusätzlich. Leider führt das aber auch zu einer doppelten Datenhaltung. Man sollte sich also an dieser Stelle Gedanken machen, ob und wie genau man an dieser Stelle reagiert. Schwerwiegend wird es, wenn man bisher auf eine Protokollierung im Betriebssystem gesetzt hat um den DBA zu entlasten. Unified Auditing bietet hier aber, abseits von Audit Vault, keine Alternative in Sachen Zugriffsbeschränkung für den SYSDBA. Dieser Vortrag beschäftigt sich zunächst mit den Grundlagen der richtigen Auditierung. Was hat der DBA bei Upgrade und Neu-Installation in Sachen Auditing zu beachten. „Unified Auditing", die wichtigste Neuerung soll das Hauptthema sein. Wie erstellen und verwalten wir unsere Audit Policies? Oracle führt auch hier einige wichtige Vereinfachungen in Sachen Audit Policies sowie Action- und Condition-based Audit Konfiguration ein. Abseits von fehlender OS Auditierung wurden doch einige andere Felder verbessert. Schließlich machen wir uns Gedanken über Security und Performance. Wussten Sie, dass mit 12c erstmals auch RMAN Operationen auditiert werden können? Was passiert mit unseren Audit records in der SGA bei einem Instance Crash? Wer hat Zugriff? Wofür brauche ich noch Fine-Grained-Auditing?

Vorüberlegungen zum Auditing

Man verwendet Auditing um allgemeine Datenbankaktivitäten zu überwachen. In der Oracle Datenbank kann man es auf verschiedenen Ebenen, mit verschiedenen Granularitäten und auf verschiedene Ausgabekanäle konfigurieren.

Bevor man jedoch Auditing konfiguriert und einschaltet, sollte man sich über ein paar wichtige Punkte außerhalb der Oracle Datenbank gewahr werden. Meist fängt die Überlegung mit bestimmten Vorgaben von außen an, welche den eigentlichen Anstoß für eine Überwachung geben. Daher sollte man sich genau überlegen: Welchen Grad an Überwachung will oder muss ich überhaupt erreichen? Stichwort Datenschutz: Darf ich im gewünschten Umfang überhaupt überwachen und wer hat Zugriff auf die gewonnenen Daten? Auswertung: Wer kümmert sich um die angefallenen Audit-Daten und wertet sie entsprechend aus? Archivierung: Was passiert mit alten Audit-Daten, wie lange sollen diese aufbewahrt und im Zugriff bleiben? Kosten: Habe ich entsprechende Performance und Speicherplatz in meiner Umgebung?

Die letzte Frage kommt von technischer Seite meist als erstes: Gibt es Geschwindigkeitsverluste und leidet die Performance der Datenbank? Diese Frage lässt sich natürlich nicht dirket oder allgemein beantworten, denn sie hängt von einer Reihe Faktoren ab. Auditiert man im normalen Umfang, also beispielsweise Failed Logins, SYS Operationen und Zugriffe auf bestimmte sensible Tabellen, dann bewegt sich die Last im kaum wahrnehmbaren Bereich. Fängt man jedoch an sämtliche Datenbankzugriffe und jedes einzelne Statement zu auditieren, dann wird sie wahrscheinlich extrem steigen.

Generell lässt sich für die Praxis sagen, dass man Auditing nur selektiv einschalten sollte, für genau das, was auch ausgewertet werden muss. Nicht vergessen sollte man in jedem Fall die Überwachung von Inhalt und Größe der Audit-Daten.

Upgrade nach 12c

Vorab: Wenn man eine neue Oracle 12c Datenbank erstellt, oder nach einer Migration auf 12c empfiehlt es sich das Auditing möglichst bald zu überarbeiten und auf Unified Auditing und die damit neu eingeführten Audit Policies umzustellen. Hierzu kündigt Oracle in seiner Dokumentation auch noch den desupport für künftige Major Releases an:

„Beginning with Oracle Database 12c, Oracle introduces unified auditing, which provides a full set of enhanced auditing features. For backward compatibility, traditional auditing is still supported. However, Oracle recommends that you plan the migration of your existing audit settings to the new unified audit policy syntax. For new audit requirements, Oracle recommends that you use the new unified auditing. Traditional auditing may be desupported in a future major release.“

(Oracle Database Online Documentation 12c Release 1 (12.1))

Diese Anküdigung lässt also keinen Spielraum für Spekulationen. Man sollte also sein derzeitiges Audit analysieren und ein entsprechendes Konzept zum Umbau entwickeln. Wie bereits erwähnt bietet Oracle für den Übergang den Mixed Mode an. Dies führt dazu, dass man nicht nur altes und neues Audit parallel betreiben kann, sondern jetzt theoretisch auch ein und das selbe doppelt überwachen und damit auch doppelte Audit-Einträge erzeugen kann. Schaut man sich die verschiedenen Audit Trails an, also Orte an denen Auditdaten mit Oracle 12c gespeichert werden kann, fällt auf, dass aktuell nicht alles ins neue Unified Audit Trail läuft wie man vermuten könnte. Nein. Aktuell schleppen wir das „alte“ Standard Auditing in Form der SYS.AUD$ Tabelle und das Fine-Grained Auditing in der Tabelle SYS.FGA_LOG$ weiter mit, in das bei einer Migration, weiter unser „altes“ Auditing reinlaufen würde. Daneben haben wir jetzt das neue Unified Audit in der Tabelle SYS.UNIFIED_AUDIT_TRAIL. Im Unified Audit Trail würde sich aber nach einer erfolgreichen Umstellung auf Unified Audit alles verteilte aus 11g Altlasten bündeln: SYS.AUD$, SYS.FGA_LOG$, DVSYS.AUDIT_TRAIL$ (Database Vault Audit), OS (.aud Dateien), XML sowie EXTENDED Audit.

Im Mixed Mode sind also erstmal alle Audit Trails aktiv. Audit-Daten werden in alle vorhandenen Trails geschreiben, also erzeugen beispielsweise FGA Policies weiterhin Einträge unter SYS.FGA_LOG$. In diesem standardmäßig eingeschalteten Mixed Mode greifen auch schon zwei von Oracle erstellte Policies und erzeugen Audit-Daten im neuen SYS.UNIFIED_AUDIT_TRAIL, nämlich ORA_LOGON_FAILURES und ORA_SECURECONFIG. Zu überprüfen über:

```
SELECT * FROM audit_unified_enabled_policies;
```

Zu Verwirrung führt an dieser Stelle, dass das Unified Auditing nicht wie vorher in 11g allein über einen init.ora Parameter gesteuert wird, sondern über V$OPTION beziehungsweise über ein relinken der Binaries, welches einen Neustart der Datenbank bedeutet. Um die Verwirrung komplett zu machen fragt man jetzt die entsprechende V$OPTION ab und erhält als Ergebnis FALSE:

```
select * from v$option where parameter = 'Unified Auditing';
```

Das bedeutet nicht, dass Unified Auditing nicht funktioniert, sondern nur, dass nicht „Pure Unified Auditing“, also reines Unified Auditing aktiviert ist. Wenn also der Parameter audit_trail nicht auch noch auf none sitzt haben wir Mixed Mode:

```
select value from v$parameter where name = 'audit_trail';
```

Schalten wir Unified Auditing ein und sitzt V$OPTION für Unified Auditing auf TRUE haben wir „Pure Unified Auditing. Der audit_trail Parameter hat jetzt keinerlei Wirkung mehr:

```
cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME
```

Wenn Unified Auditing in V$OPTION auf FALSE sitzt und audit_trail auf NONE wird auch in kein Audit Trail mehr geschrieben.

Unified Auditing lässt sich auch wieder ausschalten:

```
cd $ORACLE_HOME/rdbms/lib

make -f ins_rdbms.mk uniaud_off ioracle ORACLE_HOME=$ORACLE_HOME
```

Vor- und Nachteile

Natürlich hat oder hatte jeder Audit-Trail Typ seine Vor- und Nachteile. Zum Beispiel ist das db Auditing (audit_trail = db) sehr einfach auszuwerten. Kann aber durch den DBA manipuliert werden und generiert REDO Information und somit mehr Last auf der Datenbank. OS Auditing (audit_trail = os) bietet mit entsprechendem Berechtigungskonzept einen besseren Schutz vor dem unberechtigten DBA Zugriff und erzeugt keine zusätzliche Last auf der Datenbank. Es ist dann aber auch schwieriger Auszuwerten und bietet nur eingeschränkte Auditing Informationen. XML (audit_trail = xml) greift an diesen Punkten an. Lässt sich besser auswerten als OS Dateien und kann mit „extended“ auch komplette SQL-Abfragen protokollieren, aber die erzeugten XML Dateien werden teilweise sehr groß und sind damit im Zugriff per View sehr langsam. Bleibt noch SYSLOG bzw. Windows EventLog welcher recht gut geschützt ist vor DBA Manipulationen, aber auch nur sehr eingeschränkte Auditing Informationen liefert.

Unified Auditing bietet hingegen einen besseren Schutz vor DBA Manipulationen. Bietet einen einzigen Audit Trail für alle Auditing Informationen an und ist dadurch relativ einfach auszuwerten. Durch den eigenen dedizierten SGA-Bereich wird auch an der Performance geschraubt. Nachteil ist, dass man seine bisherige Konfiguration komplett anpassen und vorher natürlich auf 12c upgraden muss.

Will man das Risiko der Manipulationen weitgehend verhindern und den Verbrauch von Datenbank-resourcen auf dem Produktivsystem verhindern, bleibt weiterhin nur das zentrale Ablegen der Audit-Daten mittels Oracle Audit Vault und Database Firewall. Hierin besteht auch die einzige Möglichkeit

um Gesetze und Vorgaben, aus Basel II oder SOX in sachen manipulationssichere Aufbewahrung umzusetzen.

Architektur

Wie bereits beim Thema Mixed Mode beschreiben stehen in der 11er Datenbank verschiedene Audit Trails zur Verfügung. Häufig sind dann auch die Audit Trails gleichzeitig in Gebrauch, also beispielsweise SYS Auditing, Standard Auditing und Fine-Grained Auditing parallel. Sicherlich besteht hier die Gefahr relevante Audit-Informationen zu übersehen, Lücken zu lassen oder viel Zeit an der falschen Audit Policy zu verlieren. Letztlich besteht die Schwierigkeit verschiedene Informationen aus unterschiedlichen Audit Trails in den richtigen Kontext zu stellen. Alles unter dem Risiko, dass Informationen bereits manipuliert wurden, oder dass das Audit gleich ganz umgangen wurde. Letztlich bleibt die Frage nach der Performance und die richtige Zuordnung von zusätzlichen Lasten je nach Audit Variante und Konfiguration.

Oracle 12c mit Unified Auditing gibt hier natürlich eine Antwort. Die erste heißt: Wir ziehen alle möglichen Audit Daten zusammen in einem Audit Trail und sichern den Zugriff mittels der Rollen AUDIT_VIEWER und AUDIT_ADMIN. Nur der AUDIT_ADMIN kann mittels PL/SQL Package DBMS_AUDIT_MGMT Audit Daten verwalten oder löschen. Somit hat der Auditor oder DBA über die UNIFIED_AUDIT_TRAIL View nur lesenden Zugriff auf die Audit Daten.

Sämtliche Audit Informationen werden im Unified Audit Trail abgelegt. Dazu gehört dann auch Fine-Grained Auditing, Oracle Label Security (ols_*-Spalten), Oracle Database Vault (dv_*-Spalten), Oracle Real Application Security (xs_*-Spalten), SYS Audit, und neu Data Pump (dp_*-Spalten) sowie bestimmte RMAN Aktionen (rman_*-Spalten). Wobei das RMAN Audit nicht weiter konfigurierbar ist.

Es lassen sich alle Informationen über die UNIFIED_AUDIT_TRAIL View auslesen:

```
select * from unified_audit_trail;
```

Eine Ausnahme gibt es aber dann doch und die betrifft den Fall, dass Oracle nicht schreibend auf das Unified Audit Trail zugreifen kann, werden die Audit Informationen in Binärdateien auf das Betriebssystem ins Verzeichnis $ORACLE_BASE/audit/$ORACLE_SID/ geschreiben. Die betrifft z.B. den NOMOUNT- und MOUNT-Status, READ ONLY gestartet, oder im Falle einer Standby Datenbank. Die so erzeugten Daten werden aber trotzdem über die UNIFIED_AUDIT_TRAIL View abgerufen und können mittel DBMS_AUDIT_MGMT Package und LOAD_UNIFIED_AUDIT_FILES in die Datenbank geladen und spätestens über CLEAN_AUDIT_TRAIL gelöscht werden.

Die Antwort in Sachen Performance lautet: Unified Audit SGA Queue. Erzeugte Audit Einträge werden erst in der SGA Queue gesammelt, bevor sie dann asynchron auf Platte geschrieben werden. Eine Zwischenablage in der SGA erzeugt natürlich das Problem des möglichen Datenverlustes bei einem Crash der Datenbank oder des Servers. Auf Kosten der Performance kann man hier statt des standarmäßigen Queue Mode auch auf Immediate Mode umstellen. Die Größe lässt sich über den init.ora Parameter UNIFIED_AUDIT_SGA_QUEUE_SIZE festlegen und liegt zwischen 1 MB und maximal 30 MB.

```
select value from v$parameter where name =
'unified_audit_sga_queue_size';
```

Queue Mode abfragen:

```
select parameter_value from dba_audit_mgmt_config_params where
parameter_name = 'AUDIT WRITE MODE';
```

Queue Mode auf IMMEDIATE ändern:

```
BEGIN
  DBMS_AUDIT_MGMT.set_audit_trail_property(
    audit_trail_type          => DBMS_AUDIT_MGMT.audit_trail_unified,
    audit_trail_property                                          =>
DBMS_AUDIT_MGMT.audit_trail_write_mode,
    audit_trail_property_value                                    =>
DBMS_AUDIT_MGMT.audit_trail_immediate_write
  );
END;
/
```

Ein manuelles „flushen“ der SGA Queue ist auch möglich. Der gesamte Inhalt der SGA Queue wird direkt auf Disk geschrieben:

```
EXEC DBMS_AUDIT_MGMT.flush_unified_audit_trail;
```

Audit Policies

Mit Unified Audit führt Oracle auch die Unified Audit Policies ein. Mit diesen kann man also das Standard-Audit welche über Objekt- oder Systemprivilegien überwachen ablösen. Unified Audit Policies sind an dieser Stelle mächtiger und erlauben auch die Überwachung in komplexeren Kontexten, welche mit Standard Auditing nur schwierig bis garnicht umsetzbar gewesen ist. Schwierig wurde es, wenn man Zugriffe zu bestimmten Zeiten, nur aus bestimmten Netzwerken oder nur für DBA Aktionen des Application Owners.

Eine Audit Policy fasst eine Anzahl Audit Einstellungen zusammen, um bestimmte Aktionen in der Datenbank zu überwachen. Es können eine Reihe von Aktionen überwacht werden. Zum Beispiel: Die Anwendung von Systemprivilegien (SELECT ANY TABLE), Objektprivilegien (DROP TABLE) oder Zugriffe, einzelne Benutzerkonten, administrative Benutzer (SYSDBA, SYSOPER), Rollen (DBA-Rolle), das ganz CDB- oder PDB-Abhängig. Es können Oracle Komponenten wie Data Pump oder Applikations-kontextabhängige Aktionen überwacht werden.

Die Erstellung erfolgt über CREATE AUDIT POLICY. Um alle Aktionen der User mit DBA Rolle zu überwachen erstellt man folgende einfache Policy:

```
CREATE AUDIT POLICY pol_dbarole_all ROLE dba;
```

Unter Umständen produziert eine solche Policy zuviele Daten. Einschränken ließe sich diese natürlich mit einer Bedingung (WHEN).

Im anschließend Beispiel beschränken wir eine neue Policy auf Zugriffe außerhalb des Datenbank Servers, die nicht mit unserer Applikation erfolgt. Allerdings zeichnen wir hier jedwede Aktion mit:

```
CREATE AUDIT POLICY pol_notapp_all
ACTIONS ALL
WHEN
```

```
'(upper(sys_context('userenv','module')) != "MyOwnApp")
OR
'(sys_context('userenv','ip_address') != "127.0.0.1")'
EVALUATE PER SESSION;
```

Das war aber lediglich die Definition der beiden Policies. Aktiviert werden sie mit nachfolgenden Statements. Mit WHENEVER SUCCESSFUL legen wir fest, dass nur bei einem erfolgreichen Befehl auditiert werden soll. Der umgekehrte Fall ergibt sich aus WHENEVER NOT SUCCESSFUL. Bei der pol_notapp_all schließen wir Anmeldungen mit HR aus. Wollten wir diese Policy lediglich auf HR beschränken müssten wir BY statt EXCEPT nutzen.:

```
AUDIT POLICY pol_dbarole_all WHENEVER SUCCESSFULL;

AUDIT POLICY pol_notapp_all EXCEPT HR;
```

Natürlich lässt sich die Komplexität unendlich steigern und an die eigenen Bedürfnisse anpassen. Es kann aber keine beliebige PL/SQL Funktion für die Bedingung angewandt werden. Eingeschränkt wird man hier auf bestimmte numerische Funktionen (INSTR, CEIL, FLOOR und weitere) und Zeichenfunktionen (CONCAT, LOWER, UPPER und weitere) sowie SYS_CONTEXT, SYSDATE und so weiter. Diese lassen sich mit AND, OR, IN, NOT IN, =, <, >, <> bzw. != verknüpfen.

Oracle liefert bereits acht vordefinierte Policies mit, von denen zwei aktiv sind. ORA_SECURECONFIG, welche die Audit Konfiguration überwacht und ORA_LOGON_FAILURES, welche die Logons protokolliert. Alle bereits definierten Audit Policies finden sie hier:

```
SELECT DISTINCT policy_name FROM audit_unified_policies;
```

All aktiven Audit Policies finden sie hier:

```
SELECT * FROM audit_unified_enabled_policies;
```

Ausschalten erfolgt über NOAUDIT löschen wie gewohnt über DROP.

Weitere Beispiele finden Sie in der Präsentation.

Abschluss

Mit 12c gibt es ein paar wichtige Neuerungen im Bereich Auditing. Audit Statements und Audit Policies existieren im Mixed Mode problem nebeneinander. Dies vereinfacht zunächst die Migration, sollte aber nicht dazu verleiten, ein doppeltes Auditing zu produzieren. Schalten Sie bei Erstellung einer neuen Audit Policy ein unter Umständen regelgleiches „altes“ Audit Statement umgehend ab. Vergessen Sie nicht, unabhängig von der Auditing Methode, müssen die gewonnen Informationen abschließend ausgewertet und entsprechend archiviert werden. Oracle hat das DBMS_AUDIT_MGMT Package hierfür auch um den Support für Unified Audit Trail erweitert. Vergessen sie nicht die Audit-Daten, gleich am Anfang in einen eigenen Tablespace zu verschieben. Setzen Sie die Berechtigungen mit den neuen Rollen AUDIT_ADMIN und AUDIT_VIEWER sinnvoll um. So erspart man sich unnötige Konflikte, wenn auch nur in der Theorie, für DBA‘s und Applikationsadministratoren. Falls die neuen Hürden noch immer nicht reichen, bleibt nur ein Blick auf Audit Vault um entsprechende Anforderungen in Sache manipulationsverhinderung umzusetzen.

Kontaktadresse:

Sebastian Winkler

CarajanDB GmbH

Siemensstraße 25

D-50374 Erftstadt

Telefon:	+49 (0) 2235-170 91 88
Fax:	+49 (0) 2235-170 97 78
E-Mail	sebastian.winkler@carajandb.com
Internet:	www.carajandb.com