Die Anwender in der DB im Zaum halten - Über Rechte/Rollen und den sicheren Betrieb der DB

Gunther Pippèrr

GPI Consult

Witzenhausen / Roßbach

Schlüsselworte
Datenbank Sicherheit, User Verwaltung, Proxy User, Rollen

Einleitung

Mit Rollen/Rechten und dem Proxy User ein persönliches User-Konzept auf der Datenbank umsetzen.

Gerade mit den gestiegenen Anforderungen der Datenschutz-Grundverordnung DSGVO wird es immer wichtiger, die Anwender und Entwickler auf einer Datenbank besser zu strukturieren und die Rechte aller Beteiligten so passend wie möglich einzurichten.

Besonders in einer Data Warehouse Umgebung lässt es sich oft nicht vermeiden, dass sich Endanwender mit MS Offices Tools, TOAD oder BI Werkzeugen wie Tableau direkt an der Datenbank anmelden um Ihre Abfragen adhoc durchzuführen.

Die Oracle DB stellt dabei schon seit Jahren sehr gute Funktionen zur Verfügung, die helfen die Zugriffe auf die Daten auf das Notwendigste einzuschränken.

Konzepte, wie die „Secure Application Role“ und der „Proxy User“, „Virtual private Database“ VPD, die normalen Rollen und Rechte und das Auditing, helfen uns dabei, diese Anforderungen umzusetzen und zu protokollieren.

Im Vortrag wird aufgezeigt, wie aus diesen Komponenten ein schlüssiges Konzept für den proaktiven Datenschutz in einer Datenbank Umgebung 12c zusammengestellt werden kann.

Wie aber so ein Konzept in einer gewachsenen Umgebung umsetzen?

Als Praxis Beispiel wird ein aktuelles Projekt vorgestellt, in dem diese Anforderungen umgesetzt werden mussten, und welche Erfahrungen mit den Anwendern dabei gesammelt werden konnten.

Das Szenario

In einer „normalen“ traditionellen Applikation meldet sich die Applikation über ihr zugehöriges Datenbank Schema an der Datenbank an, ein beliebtest Password ist dabei oft der Name des Schemas (Username=Password).

Das Password der Applikation kann nur schwer geändert werden, da dieses Password in vielen ODBC Verbindungen hart hinterlegt ist, Schnittstellen direkt damit arbeiten und im Unternehmen kein Wissen mehr besteht, was alles mit der Datenbank über die verschiedenen Applikationsschemas kommuniziert.

Da sich nicht jede Funktionalität in der Anwendung umsetzten lässt, melden sich Sachbearbeiter und Analysten mit den allgemein bekannten Zugangsdaten direkt an der Datenbank per SQL*Net an, um dann in Excel oder TOAD direkt mit dem Daten der DB zu arbeiten.

Wer in der Datenbank wann und was geändert hat, lässt sich im Nachhinein nur schwer nachvollziehen.

Der Nachweis, dass die Datenschutz-Vorgaben wirklich eingehalten wurden, ist damit nur sehr schwierig zu erbringen. Die Gefahr das Daten „verloren“ steigt.

Der Lösung Ansatz

Im ersten Schritt muss eine Matrix erstellt werden, **was** und **wer** mit der Datenbank **wie** arbeiten soll.

Ziel ist es hier eine saubere Trennung zwischen technischen und persönlichen User zu erreichen,

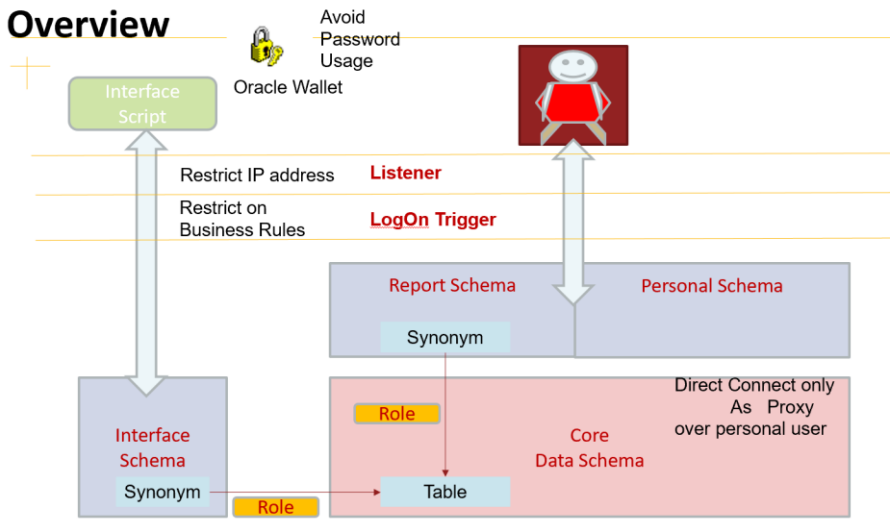
Define a matrix of responsibility

User Type	Data Type	Example	Concept	Role Name	DB Profile Name
Technical	Data Owner	Application Core Schema	Very Strong Password with min. 20 letters		data_owner
Technical	Application (Select/DML)	Interfaces to other programs	Synonym Schema, minimal rights on table and PL/SQL	<appname>_role	interface
Personal	Only Select	Analysts	Synonym Schema, minimal rights on table and PL/SQL	For example Analyst_role_<schema>	personal_user
Personal	Select / DML /DDL	Developer and Deployment User	Oracle Proxy Authentication	Personal user only create Session	personal_user

Es wird zwischen zwei Typen von technischen Usern und zwei Typen von persönlichen Usern unterschieden. Für jede Gruppe müssen die entsprechenden, passenden Konzepte entwickelt werden.

Jede Gruppe sollte über ein eigenes Datenbank-Profile verfügen, einmal um hier auch die User in der DB besser zu unterscheiden und zum zweiten Feature wie Workload Management und Passwort Regeln je Gruppe granular einstellen zu können.

Mit dem Gesamtkonzept können wir nun auf jeder Ebene in der Datenbank Umgebung ansetzten und mit dem Einsatz der möglichen Oracle Funktion eine große Verbesserung der aktuellen Situation erzielen, ohne die Applikation selber groß anpassen zu müssen.



Zum Einsatz kann in der Oracle Welt kommen:

- Schema User und komplexe Passwörter
- User Profiles
- Oracle Wallet für die Authentifizierung von SQL Skripten
    - Alternativ verschlüsselte Passwörter in SQL Skripten
- Oracle Proxy User Concept
- Objects Grant und Synonym
- Rollen mit allen Möglichkeiten, wie mit Passwort schützen (Secure Application Role)
- VPD – Virtual Private Database und Fine Access Control (Nur EE!)
- Oracle Workload Management um die Ressourcen Verwendung einzuschränken
- Login Trigger
- Listener Security auf IP Adressen
- Oracle Auditing

Gleichzeit muss die Überwachung der Datenbank optimiert werden um so viel Daten wie nötig (aber eben nicht zu viel) einzusammeln und auszuwerten.

Schema User und komplexe Passwörter

Im ersten Schritt sollte die Unsitte „Schemaname“ = „Passwort“ bekämpft werden.

Dazu ist es aber notwendig zu wissen, wer sich an der Datenbank so alles anmeldet.

Im ersten Schritt wird dazu das Auditing auf Login aktiviert, das hilft aber nur, wenn die Audit Daten auch eine gewisse Zeit zur Verfügung stehen und nicht vom Hoster nach einer Woche alle gelöscht werden.

Im nächsten Schritt kann eine Kontrolle über das AWR (wenn die Diagnostik Option der EE erworben wurde!) erfolgen.

Alternativ ist eine eigene Routine zur Aufzeichnung aller Anmeldungen, die einfach alle paar Minuten die V$SESSION in eine feste Tabelle mit Angabe eines Zeitstempels überträgt, sehr hilfreich. Das hat auch den Vorteil, dass auch Sessions mit geschnitten werden die nicht „active“ sind (in der AWR Session History werden nur aktive Sessions protokolliert!). So können wir nun besser erkennen, wer sich nur anmeldet aber eigentlich nichts tut.

Falls erreichbar, sollte auch das Listener.log ausgewertet werden, hier lassen sich auch viele Informationen auslesen, was sich so alles an der Datenbank anmeldet.

In diesem Schritt können wir aber nur analysieren, da es ohne Umsetzung der Responsibility Matrix zu viele Seiteneffekte gibt, wenn wir Passwörter einfach so ändern.

Oracle Wallet (Oracle Secure External Passwort Store) für die Authentifizierung von SQL Skripten

Neben der Option Passwörter verschlüsselt zu hinterlegen, kann auch die Oracle Wallet für den Zugriff auf die DB ohne Passwort Eingabe verwandt werden. Siehe dazu mehr unter https://www.pipperr.de/dokuwiki/doku.php?id=dba:oracle_secure_external_passwort_store

Nachteil: Jeder der auf dem System sich anmelden kann, braucht nur kein Passwort mehr für die Datenbank!

Alternativ: Passwörter verschlüsselt auf dem System hinterlegen und im Skript nur referenzieren , siehe auch = >
https://www.pipperr.de/dokuwiki/doku.php?id=dba:passwort_verschluesselt_hinterlegen

Oracle Proxy User Concept

Die Idee hinter dem Proxy Connect ist die Anmeldung an einem Schema mit den Credentials eines anderen DB Users, z.B. soll ein persönlicher User sich an dem Schema eines technischen Users anmelden. D.h. der persönliche User, z.B. ein Datenbank Entwickler, benötigt nicht das Passwort des technischen Schemas um mit diesem Schema zu arbeiten.

Für eine erfolgreiche Anmeldung darf keiner der User gelocked sein. Das Passwort des technischen Users kann aber abgelaufen sein.

Mehr dazu siehe https://www.pipperr.de/dokuwiki/doku.php?id=dba:proxy_connect

Objects Grant und Synonym

Ganz klassisch die Rechte in der DB organisieren.

So lässt sich einfach ein Schema einrichten, auf dem Business Anwender sich mit Ihren persönlichen Passwort anmelden und dann mit Hilfe von Rechte Vergaben und Synonymen nur die Daten sehen dürfen die der Sachbearbeiter auch sehen darf.

Rollen mit allen Möglichkeiten wie mit Passwort schützen (Secure Application Role)

Eine Rolle bündelt die Rechte in der DB, die einzelnen Grants müssen nicht jedes Mal neu an einen Anwender vergeben werden.

Nachteil: Die Rollen Verwendung ist in PL/SQL stark eingeschränkt, für das Lesen von Tabellen werden zum Beispiel direkte Grants auf den Schema Owner des PL/SQL Programms benötigt.

VPD – Virtual Private Database and Fine Access Control (Nur EE!)

Mit Application Context and Fine-Grained Access Control kann der Zugriff auf Daten innerhalb einer Tabelle sehr gut eingeschränkt werden.

Ein Anwender, eine Applikation kann nur die Daten sehen, wenn die entsprechenden Regeln zum Sehen auf die Daten erteilt wurden. Im Gegensatz zu Rechten auf Tabellen reden wir hier von Rechten auf die Daten in der Tabelle.

Technisch wird das über ein verstecktes Prädikat in der where Klausel einer jeden SQL Abfrage erreicht. Nach der Anmeldung an der DB wird über eine hinterlegte Regel jedes SQL damit so erweitert, dass nur die „richtigen“ Daten gelesen werden können.

User Profiles

Über User Profile wird unteranderem das Passwort Verhalten geregelt, zum Beispiel müssen Persönliche User ihr Password alle 90 Tage ändern, technische User müssen das nicht tun aber dann ein sehr langes Passwort verwenden.

Oracle Workload Management um die Ressourcen Verwendung einzuschränken

Die Datenbank kann dann über das hinterlegte Profil am User (bzw. über diverse andere Kriterien auf der aktuellen Session) entscheiden, ob der Anwender alle CPU’s der Maschine bei einer parallel Query erhalten darf oder ob nur z.B. 2 CPU’s verwendet werden dürfen.

Damit kann verhindert werden, das zum Beispiel auf einer EXADATA ein Analyst die gesamte Rechenleistung abzieht und wichtige ETL Prozesse zum Erliegen kommen.

Login Trigger

Bei jedem Login wird geprüft ob der Anwender bestimmten Kriterien genügt, falls nicht wird ein Login nicht zugelassen.

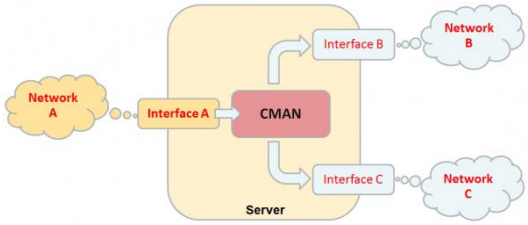
Risiko: Die Login Check Routine sollte sehr betriebssicher entwickelt werden (z.B. möglichst keine Abhängigkeiten auf andere Anwendung PL/SQL Routinen in der Datenbank)! Wird der Login Trigger ungültig kann sich nicht mehr an die DB angemeldet werden!

Listener Security auf IP Adressen

Der Oracle Listener kann auf IP Adressen eingeschränkt werden. Mit dieser einfachen Maßnahme kann schon viel erreicht werden. Gerade in große Umgebung ist so sichergestellt, dass prinzipiell nur erlaubte Anwendungen mit der Datenbank reden.

Damit das wartbar bleibt, ist es sehr praktisch, wenn die DBA's am Arbeitsplatz und bei VPN Einwahl spezielle IP Netze zugeteilt bekommen. So bleibt die Wartung der DB ohne Einschränkung möglich, Fachfremde können aber nicht so einfach sich auf jede DB in der Firma anmelden.

Mit dem Oracle CMAN, Connection Manager lässt sich das dann noch so erweiterten, dass auch sehr komplexe Anforderungen an die Netzwerk Sicherheit umgesetzt werden können.



Siehe => https://www.pipperr.de/dokuwiki/doku.php?id=dba:sqlnet_cman_connection_manager

Oracle Auditing

Auch sollte in der DB alles Wichtige überwacht werden, DDL, DML Befehle und bei Bedarf auch SELECT auf empfindliche Daten.

Allerdings sollte nur so viel aufgezeichnet werden, wie auch später ausgewertet werden kann. Wird zu viel aufgezeichnet, wird oft nur viel Platz verschwendet.

Hier also lieber weniger aufzeichnen, dafür das Ganze aber öfters überwachen und prüfen

Die Erfahrungen bei der Umsetzung in der Praxis

In der Praxis hat sich gezeigt, dass in einer bestehenden Umgebung, die von einem externen Hoster in einem sehr fernen Land betrieben wird, nur sehr langsam und wenig vom Konzept umgesetzt werden konnte.

Die Schwierigkeit in einer solcher Umgebung besteht darin, dass eben nicht nur einfach ein Konzept auf Zuruf umgesetzt werden kann. Out Sourcing heiß eben nicht, dass proaktiv gehandelt wird, sondern meist wird nur das bestehendes mit Strom versorgt wird bis es nicht mehr funktioniert.

Gerade in sehr stark und sehr weit outgesoursten Umgebungen bleibt dem Kunden hier nichts anders übrig, als dem extern Full Service Dienstleister auszubilden und viel Zeit für ständiges Zureden

einzuplanen Die genannten Features, wie z.B. Datenbank Rollen, waren dort in dieser Umgebung meist völlig unbekannt.

D.h. Skill Bulding ist einer der wichtigsten Herausforderungen, um ein solches Konzept umzusetzen.

Jede Änderung an einem solchen System wird mit komplexen Changemanagement Prozessen so stark wie möglich verzögert. Das ist im Prinzip ja auch im Sinne des Hosters, der Marge nur erzielen kann, wenn er so aufwandsarm wie möglich arbeitet.

Zusätzlich müssen die internen Mitarbeiter intensiv ausgebildet werden, der Hauptnachteil liegt darin das für das Ändern eines Passworts viel Eigen-Initiative notwendig ist.

Die Tools wie Excel unterstützen das zwar im Prinzip, für die Mitarbeiter stellt so ein zusätzlicher Dialog aber oft eine kaum zu überwindender Hürde dar.

Fazit – Sicherheit ist möglich aber nur in der Theorie

Technisch lässt sich Sicherheit in der DB relativ aufwandsarm erreichen indem die Oracle Feature, die auch teuer bezahlt wurden, auch eingesetzt werden.

Um so ein Projekt aber wirklich in der Praxis umzusetzen, reicht es leider nicht nur auf die Features der Datenbank zu vertrauen.

Der Hauptaufwand liegt in der Motivation der Zulieferer und Mitarbeiter sich mit dem Thema zu beschäftigen und hier die Überzeugungsarbeit zu leisten das etwas Neues gelernt werden muss.

Es muss daher genau abgewägt werden, was den Mitarbeitern zugemutet werden kann und es muss langsam Schritt für Schritt voran gegangen werden.

Kontaktadresse:

Gunther Pippèrr

GPI Consult
Bergweg 14
D-37216 Witzenhausen

Mobil: +49(0)171 8065113
E-Mail gunther@pipperr.de
Internet: **http://www.pipperr.de/dokuwiki/doku.php**