Datenbanksicherheit mit Konzept

Dr. Günter Unbescheid
Database Consult GmbH
Jachenau

Schlüsselworte

Datenbank, Sicherheit, Verschlüsselung, Authentifizierung, Autorisierung

Einleitung

Security war schon immer ein wichtiges Thema, das jedoch lange nur zögerlich umgesetzt wurde. Dies hat sich in jüngster Zeit gründlich geändert: viele Betriebe haben sich mittlerweile die sicherheitstechnische Aufarbeitung ihrer Rechnerumgebungen auf die Fahnen geschrieben und sehen sich dabei einer Fülle von technische Optionen und organisatorischen Hürden gegenübergestellt.

Dieser Beitrag konzentriert sich auf die Kernfragen bei der Erstellung von Sicherheitskonzepten für Oracle Datenbanken der neuesten Generationen und stellt die zur Umsetzung wichtigsten Schlüsseltechnologien dar. Nur der konzeptionell aufeinander abgestimmte Einsatz von Technologien und Prozessen kann den Herausforderungen des Themas Security auf Dauer gewachsen sein.

Das Umfeld

Der Terminus „Datenbanksicherheit“ ist mit Wikipedia schnell und klar umrissen:

Datenbanksicherheit bezeichnet die Verwendung einer breiten Palette von Informationssicherheitskontrollen, um Datenbanken zu schützen (unter Umständen einschließlich der Daten, der Datenbankanwendungen …, der Datenbankserver und der dazugehörigen Netzwerkverbindungen) gegen Gefährdungen der Vertraulichkeit, Integrität und Verfügbarkeit. Es beinhaltet verschiedene Typen oder Kategorien der Kontrolle, etwa technische, verfahrensorientierte / administrative und physische.

Bruce Schneier, ein weltweit anerkannter Experte in Sachen Kryptografie und Computersicherheit bringt es lapidar auf den Punkt:

Software has vulnerabilities because it's written by people …

und an anderer Stelle (“*Secrets an lies*”) gibt er den Rat:

Security, ... involves people, ... (it) ***is a process****, not a product*

Die Definition der oben genannten drei Kernbegriffe von Security ist ebenfalls wohl definiert und vielfältig kommentiert worden:

Vertraulichkeit – ist die Eigenschaft einer Nachricht, nur für einen beschränkten Empfängerkreis vorgesehen zu sein.

Integrität – ist die vollständige sowie unveränderte Übermittlung von Daten an den Empfänger, aber auch – allgemeiner – die Korrektheit von Daten, frei von unerlaubten Modifikationen und die Nachvollziehbarkeit (erlaubter) Modifikationen.

Verfügbarkeit – ist die „Uptime“ pro Zeiteinheit (in Prozent), sofern die Antwortzeit eine bestimmte Kenngröße nicht überschreitet. Sie ist damit ein Qualitätskriterium.

Diese wenigen Begriffsbestimmungen geben zwar die Ziele des Themas vor, lassen uns aber zunächst alleine mit den für diese Prozesse notwendigen Detailentscheidungen. Immerhin können wir an dieser Stelle aus dem Gesagten weitere Ziele für das Thema im allgemein und Datenbanksicherheit im Besonderen ableiten:

- Sicherheit ist ein permanenter und iterativer Prozess. Technologischer Fortschritt gebiert stets neue Schwachstellen, die erkannt, bekämpft und behoben werden müssen.
- Sicherheit erfordert Prozesse, Technologien sind die Basis und nicht mehr.
- Sicherheit muss demnach Technologien sinnvoll kombinieren und integrieren.
- Sicherheit muss in wohldefinierter Balance zwischen technischer Perfektion auf der einen und organisatorischer Praktikabilität und Wartbarkeit auf der anderen Seite geplant und umgesetzt werden.
- Sicherheit ist Teamarbeit und weist über die Gruppe der DBAs hinaus.
- Sicherheit wird als ganzheitliches Konzept geplant, jedoch schrittweise implementiert. Dabei müssen vorhandene Infrastrukturen so weit wie möglich integriert werden. Keine Systemlandschaft lässt sich in endlicher Zeit und mit vertretbaren Kosten komplett umkrempeln.
- Sicherheit als Gesamtkonzept ist notwendig, gleichwohl darf über dem Konzept die Pragmatik nicht vergessen werden. Ein lückenhaft umgesetztes Konzept ist mit Sicherheit besser als ein perfektes Security-Paper.
- Sicherheit darf die Nutzung und Administrierbarkeit von Systemen nicht in Frage stellen. Im Zweifelsfall steht der Nachweis von Aktionen (Nachvollziehbarkeit) über ihrer Verhinderung bzw. Unterbindung.
- Sicherheit muss auf persönlichen Identitäten aufbauen. Anders kann der Nachweis von Aktionen nicht gelingen.
- Sicherheit erfordert angepasste personelle und organisatorische Strukturen (*segregation of duties*).

Datenbanken sind generell folgenden funktionalen Anforderungen ausgesetzt, die im Umfeld des Themas Sicherheit berücksichtigt werden müssen. Die genaue Kenntnis der Schnittstellen steht damit am Anfang eines jeden Sicherheitskonzeptes:

- Applikationen greifen von *remote* auf Daten unterschiedlicher Vertraulichkeitsstufen zu. Dieser Zugriff kann über personifizierte Benutzer, aber auch über Pool-Benutzer (Applikation Server) erfolgen.
- Administratoren konfigurieren die Systeme mit hochrangigen Privilegien sowohl auf den umgebenden Betriebssystemen, als auch innerhalb der Datenbanken selbst und haben zunächst vollständigen Zugriff auf alle Daten.
- Entwickler kopieren und klonen Systeme zu Testzwecken.
- Datenbanken kommunizieren über Links und Dump Dateien mit anderen Systemen.

Datenbanken sind demnach in ein Netzwerk aus Abhängigkeiten eingebettet, die zur Umsetzung des Themas Sicherheit unbedingt berücksichtigt werden müssen. Aus den Zugriffsprofilen, die sich durch diese Abhängigkeiten ergeben, können wir den konkreten Schutzbedarf von Systemen ableiten, der letztlich vor dem Hintergrund möglicher finanzieller Schäden durch die Kompromittierung derselben beziffert werden kann. Durch die Bezifferung von potentiellen Schäden können wir eine Klassifizierung von Systemen durchführen, beispielsweise in Form der Schutzklassen „intern“, „vertraulich“ und „streng vertraulich“. Technische und organisatorische Maßnahmen zur Sicherung können dann anhand dieser Schutzklassen gruppiert und implementiert werden. Diese Strategie kann dabei helfen, sowohl die Lizenzkosten als auch die organisatorischen Aufwände zu optimieren und reduzieren, indem bei-

spielsweise die Technologie *transparent data encryption* (TDE) nur in Systemen mit streng vertraulichen Daten eingesetzt wird.

Erforderliche Maßnahmen können wir zusätzlich gliedern in:

- Grundlegende Maßnahmen („Grundhärtung"), die auf allen Systemen unabhängig von den ihnen zugeteilten Schutzklassen umgesetzt werden und
- Klassenspezifische Maßnahmen, die abhängig von der betreffenden Schutzklasse implementiert werden

Vor diesem Hintergrund werden wir im Folgenden das Thema Datenbanksicherheit sowohl im Umfeld von Betriebssystemen und Servern als auch vor dem der Datenbank selbst betrachten.

Klassifizierung von Systemen

Die Klassifizierung von Systemen auf Basis des Schutzbedarfs zur reduzierten und optimierten Konfiguration von Security-Features wurde vorstehend bereits erwähnt. Sollte diese Klassifizierung prinzipiell in Betracht gezogen werden, ist es sinnvoll, sie an den Anfang von Security Projekten und entsprechenden Planungen zu stellen, um unnötige Konfigurationsaufwände zu vermeiden.

Für die Klassifizierung sind unterschiedliche Strategien denkbar, die alle vor dem Hintergrund denkbarer Schadensszenarien entworfen werden können[1]:

- Die Betrachtung kann sich auf den Schutzbedarf der Daten in der betreffenden Datenbank konzentrieren. Dieser Ansatz ist – aus Sicht der Datenbankabteilungen – recht pragmatisch und konzentriert sich bewusst auf das unmittelbare „Hoheitsgebiet" der DBAs.
- Ein alternativer und umfassenderer Ansatz bezieht zusätzlich auch die Applikationen und ihr konfiguratorisches Umfeld sowie die betreffenden Schnittstellen mit ein. Diese Betrachtung wird dem Anspruch nach Datensicherheit wesentlich gerechter, ist in der Praxis jedoch wegen ihres Abteilungs- und Kompetenz-übergreifenden Charakters sehr aufwändig in der Umsetzung.

Ganz gleich welche der genannten Strategien zum Einsatz kommen, sie sollten stets auf Basis von Erhebungen bzw. Umfragen stattfinden, bei denen die betreibenden Fachabteilungen, die auch die Datenverantwortung haben, sich verbindlich und aktenkundig äußern. Um Entscheidungen möglich zu machen, sind im Rahmen der Erhebung klar die entscheidenden Kriterien, d.h. Schadensszenarien, aber auch die zum Schutz geplanten konfiguratorischen Maßnahmen mitsamt der entstehenden Lizenz- und Administrationskosten transparent zu machen. Nicht selten müssen Anforderungen an die Datensicherheit aufgrund von potentiellen Kosten neu bewertet werden.

Betriebssysteme und Datenbank Server

Ist eine Klassifizierung geplant, ist es dringend anzuraten, die Datenbanken „klassenkonform" auf spezifische Serversysteme zu verteilen. Eine zusätzliche Abschottung durch die Aufteilung in Subnetze ist ebenfalls empfehlenswert. Werden virtuelle Systeme für den Betrieb der Datenbanken genutzt, ist zu entscheiden, inwieweit sich unterschiedlich klassifizierte virtuelle Systeme einen gemeinsamen physikalischen Server teilen dürfen oder nicht.

Die Grundhärtung wird, wie bereits erwähnt, pauschal für alle System unabhängig von ihrer Klassifizierung durchgeführt. Zu diesem Thema existiert ein umfangreiches Arsenal an *best practises*, die auf

[1] siehe hierzu beispielsweise: „Leitfaden zur Entwicklung sicherer Webanwendungen", auch „BSI Standard 100-2, IT-Grundschutz-Vorgehensweise" (beide: Bundesamt für Sicherheit in der Informationstechnik)

jeden Fall mit den in jedem Unternehmen vorhandenen Installationsrichtlinien abgeglichen werden sollten. Auf dieser Basis müssen die eigenen Richtlinien ggf. erweitert und angepasst werden. Beispielsweise existieren für nahezu alle Unix Derivate eigene Security und Hardening Guides[2]. Darüber hinaus bieten u.a. auch die National Security Agency (NSA), das Center for Internet Security (CIS) und das Bundesamt für Sicherheit in der Informationsverarbeitung (BSI) ausführliche Anleitungen.

Eine der wesentlichen Säulen der Serversicherheit ist die persönliche Identifizierung von Benutzern, um – neben der Autorisierung – vor allem die Nachweisbarkeit von Aktionen zu gewährleisten. Ohne diese eindeutige Identifizierung in Form von persönlichen OS-Benutzern ist jedes Sicherheitskonzept wertlos. In diesem Zusammenhang sind in der Regel auch die Zugriffswege auf die Datenbank-Server zu betrachten. Im Einzelnen ergibt sich für Oracle-Umgebungen häufig folgendes Bild:

- Endbenutzer greifen nahezu ausnahmslos über Anwendungen *from remote* auf die Daten in der Datenbank zu und haben daher keine persönlichen OS-Benutzer auf den DB-Servern.
- Im Gegensatz dazu müssen DBAs zahlreiche Aufgaben direkt auf den DB-Servern erledigen, wie beispielsweise aber nicht nur das Installieren und Patchen von Oracle Software. Hier sind demnach persönliche Accounts unerlässlich.
- Für diese administrativen Accounts müssen die Zugriffswege auf die Target-Server klar definiert werden. Unterschiedliche Modelle sind in diesem Zusammenhang denkbar:

  (a) Target Server dürfen nur über sogenannte Jump-Hosts (JH) erreicht werden. Die JH werden i.d.R. von Windows Workstations aus erreicht. Die Anmeldung auf den JH erfolgt über **`ssh`** entweder per separatem (Linux-)Password oder – falls ein Authentifizierungs-Framework konfiguriert ist (beispielsweise PowerBroker) – automatisiert über vorhandene AD Kennungen.
  Die Einwahl auf den Target-Systeme kann dann von den JH aus Passwort-los über ssh-key pairs oder AD-Frameworks erfolgen.
  (b) Administratoren verbinden sich direkt und ohne die JH mit den Target-Systemen, die dem entsprechend von allen relevanten Client Workstations aus erreichbar sein müssen.

  Auf den Target-Systemen haben die Administratoren entsprechende **`sudo`**-Rechte, um privilegierte Kommandos ausführen zu können. Gleichfalls sollte der Shell-Wechsel zu **`oracle`** über entsprechende **`sudo`** Einstellungen unterbunden werden.
  Target-Systeme können für Administratoren entweder permanent navigierbar sein oder aber im Falle von Service-Arbeiten explizit freigeschaltet werden. Hierfür empfiehlt sich der Einsatz eines *privileged identity management* Systems, über das Passwort-Zuteilungen gesteuert und dokumentiert werden können.

- Software Owner zu Installation der Oracle Software sind in den meisten Fällen die Accounts **`oracle`** und **`grid`**. Wo immer möglich sollten diese Accounts gesperrt oder mit einem „unmöglichen“ Passwort versehen werden. Aktionen, die durch Administratoren unter **`oracle`** durchgeführt werden müssen, laufen dann über entsprechende **`sudo`** Privilegien.

[2] z.B. Für SLES 11 SP3: https://www.suse.com/documentation/sles11/singlehtml/book_hardening/book_hardening.html oder für Oracle Linux unter http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-harden-oracle-linux-1695888.html um nur einige Beispiele zu nennen.

Für die Nachweisbarkeit administrativer Aktionen auf den Jump- und Target Servern – Endbenutzer verbinden sich in der Regel nicht mit diesen Systemen – können folgende Optionen in Betracht gezogen werden:

- **`ssh`**-Verbindungen werden standardmäßig im SYSLOG erfasst. Wenn DBAs nicht auch gleichzeitig **`root`** Rechte innehaben, ist damit auch die Revisionssicherheit sichergestellt. Diese kann noch dadurch erhöht werden, dass Log-Einträge auf remote System weitergeleitet werden, die eine eigenständige Verwaltungsstruktur vorweisen.
- Gleiches gilt für die Nutzung von **`sudo`**-Privilegien, die ebenfalls im SYSLOG erfasst werden.
- Alle anderen Aktionen, die direkt unter den persönlichen OS-Benutzern stattfinden, können ebenfalls protokolliert werden. Hier bieten sich sogenannte Key-logger an oder spezielle Shells, wie **`rootsh`** oder **`sudosh`**. Vorsicht ist in diesen Fällen geboten, da einige Technologien auch Passwörter mitschreiben. Die Revisionssicherheit der Einträge und das entstehende Datenvolumen, das nicht nur Eingaben, sondern auch Ausgaben loggt, muss im Einzelfall sehr genau geprüft werden. Wenn alle privilegierten Kommandos über **`sudo`** erledigt werden, ist es durchaus legitim, auf das Logging personifizierter Aktionen zu verzichten.
- Welche Optionen auch immer konfiguriert werden, generell gilt, dass Logging Einträge regelmäßig und automatisiert forensisch analysiert werden sollten, um auf eventuelle Übergriffe aufmerksam zu werden. Die Abstimmung mit dem Betriebsrat ist in den meisten Fällen unerlässlich.

Oracle Datenbanken

Wie auch bei den Servern ist eine Grundhärtung der Oracle Datenbanken unabhängig von ihrer möglicherweise geplanten Klassifizierung angesagt und sollte pauschal auf allen Systemen durchgeführt werden. Auch hier existieren umfangreiche *best practises*, die im Abgleich mit den bereits im Unternehmen existierenden Regeln konsultiert werden müssen. An dieser Stelle sind nicht nur das Center for Internet Security (CIS) und das SANS Institute zu nennen oder die – immer noch – sehr gute Darstellung von Ron Ben Nathan[3]. Nach Eingabe von „hardening oracle database“ in Suchmaschinen erscheint eine umfangreiche Trefferliste mit zahlreichen Dokumenten. Die in diesen Dokumenten vorgeschlagenen Maßnahmen beziehen sich zum einen auf die OS-Umgebung der Oracle Software und ihrer Dateien und Verzeichnisse, zum anderen auf Einstellungen „innerhalb“ der Datenbank. Die Darstellung der Maßnahmen ist teilweise sehr detailliert, indem neben den Gründen auch die zur Einstellung nötige Syntax mitgeliefert wird.

Auch im Bereich der Datenbank ist die gezielte Zuteilung von Privilegien und die eindeutige, d.h. Personen-spezifische Nachweisbarkeit von Aktionen von entscheidender Bedeutung. Im besonderen Fokus stehen hier zunächst die DBAs wegen ihrer üppigen Rechtestruktur. Folgende Strategien können in diesem Zusammenhang zur Anwendung kommen:

- Die Nachweisbarkeit von administrativen Datenbank-Aktionen muss über den Audit Trail der Datenbank sichergestellt werden. SYSDBA-Aktionen lassen sich bekanntlich über den init-Parameter **`audit_sys_operations`** pauschal und vollständig erfassen Auch hier ist die Revisionssicherheit der Einträge unbedingt zu gewährleisten, vornehmlich über die Nutzung des SYSLOG Dienstes (Parameter **`audit_syslog_level`**).
- Administratoren, die lokal auf den Target-Systemen arbeiten und über persönliche Benutzer beim Betriebssystem angemeldet sind, können stets über den im Audit Trail mitgeführten OS-Benutzer

[3] Ron Ben Nathan, HOWTO Secure and Audit Oracle 10g and 11g, Auerbach Publications

eindeutig identifiziert werden, auch dann, wenn sie via SYSBA in der Datenbank als SYS fungieren.
- Für *remote* Zugriffe unter SYSDBA gilt das nur dann, wenn die Identität der Personen auf den Client Workstations durch zentrale Authentifizierungsmaßnahmen, wie z.B. Active Directory, zweifelsfrei und lückenlos sichergestellt werden kann. Wenn diese Voraussetzungen nicht erfüllt werden können, muss auf remote SYSDBA-Verbindungen verzichtet werden (kein Password File) und die Verbindung über persönliche Datenbank-Benutzer aufgebaut werden. Diese persönlichen nicht-SYSDBA-Benutzer benötigen dann auch konventionelle Audit-Optionen für den Nachweis der unter ihrem Account durchgeführten Aktionen.
- Ebenso muss die Nachweisbarkeit aller nicht administrativen Aktionen, die über Applikationen und dort genutzte persönliche oder Gruppen-Benutzer durchgeführt werden über konventionelle Audit-Optionen bzw. Policies (12c) eingestellt werden. Hierzu sind in der Regel allgemeine, d.h. für alle Applikationen gültige, Optionen durch spezifische, d.h. für einzelne Applikationen und Schutzklassen sinnvolle zu ergänzen.

Die Authentifizierung lokaler Administratoren via SYSDBA erfolgt bekanntlich über Gruppenzugehörigkeiten und ohne Angabe von Passwörtern. Für alle weiteren Datenbankbenutzer hingegen sollten die Authentifizierungserfahren genau geplant werden:

- Authentifizierungen über Passwörter müssen in der Datenbank über entsprechende Benutzerprofile und in ihnen eingebettete Passwortregeln abgesichert werden. Die Komplexität und Gültigkeitsdauer derselben sollte in Abhängigkeit von der betreffenden Schutzklasse – z.B. „intern“, „vertraulich“, „streng vertraulich“ – und nach dem Typ des Benutzers geregelt werden. Persönliche Benutzer sollten häufigere Passwortwechsel durchführen als Funktionsbenutzer, deren Passwörter nicht selten in Konfigurationsdateien „versteckt“ sind. Die Komplexität und Länge der Passwörter von Funktionsbenutzern kann dadurch jredoch drastisch heraufgesetzt werden.
  Die Einführung von Passwortregeln muss in sehr enger Abstimmung mit den zugreifenden Applikationen erfolgen. In vielen Fällen müssen wir hier aufgrund von technologischen Altlasten deutliche Abstriche bei den Konfiguration in Kauf nehmen.
- Oracle bietet bekanntlich ein Reihe von alternativen Authentifizierungsverfahren. Die wichtigsten seien an dieser Stelle nur kurz genannt: Kerberos nutzt die via AD ausgestellten Service Tickets für Passwort-lose Authentifizierungen; bei Vorliegen einer PKI-Infrastruktur können auch Zertifikate genutzt werden, ebenso wie SecureID Tokens oder die Einbindung von LDAP-Verzeichnissen wie Oracle Internet Directory. Diese Verfahren sind vor allem dort interessant, wo eine entsprechende Infrastruktur bereits vorliegt und genutzt werden kann.

Abhängig von dem ermittelten Schutzbedarf und der darauf aufbauenden Klassifizierung müssen – zusätzlich zum bereits erwähnten Grundschutz – zusätzliche Maßnahmen geplant werden:

- Geheime und streng vertrauliche Daten sollten vor unerlaubten Zugriffen durch Verschlüsselung geschützt werden. Da *transparend data encryption* (TDE) bekanntlich innerhalb der Datenbank transparent ist, kann dieser Schutz nur vor Zugriffen aus dem Kontext des Betriebssystems (z.B. durch Blockdumps) wirksam sein.
- Im Netzwerk können Verschlüsselung und Prüfsummen ebenfalls sehr effizient über wenige Parameter konfiguriert werden.

Während diese Verfahren zentral und transparent vom DBA durchgeführt werden können, erfordern die im Folgenden stichwortartig aufgeführten Technologien eine Umsetzung im Kontext der betref-

fenden Anwendungen und sind aus diesem Grunde aufwändiger zu realisieren, können aber den Datenschutz maßgeblich steigern helfen. Um nur die wichtigsten zu nennen: Die Maskierung sensibler Daten für Testzwecke und beim Klonen; Verschlüsselung von Dump-Dateien für Import und Export, Rollen- und Privilegien Strukturen in der Datenbank, die mit den Zugriffsmustern der Applikation „synchronisiert“ sind; zusätzliche, maßgeschneiderte Audit-Optionen, welche die Endbenutzer transparent machen, Kontext-abhängige Filterung von Daten (*virtual private database*) u.v.m. Alle hier angesprochenen Verfahren sind mittlerweile technologisch voll in die Oracle Datenbanken integriert.

Ein häufig diskutierter und geforderter Bereich ist der Schutz der Fachdaten vor hoch privilegierten Zugriffen, die vor allem durch die SYSDBA Authentifizierung von Administratoren nötig wird. In diesem Kontext sind zwei unterschiedliche Strategien denkbar:

- Die Kontrolle dieser Privilegien durch umfassende Auditing-Regeln sowie die lückenlose Auswertung der Zugriffsprotokolle, um Missbräuche zeitnah zu erfassen.
- Die Abschottung vorgegebener Datenbereiche, um auf dieser Weise ANY-Systemprivilegien außer Kraft zu setzen. Diesen Ansatz verfolgt beispielsweise das Werkzeug Database Vault.

Fazit

Aktuelle Oracle Releases verfügen über eine breite Palette an Security Technologien, die in der Regel gut dokumentiert sind. Die technische Implementierung ist aus dieser Sicht nicht sehr aufwändig. Die Herausforderung liegt vielmehr in der sinnvollen Abstimmung dieser Technologien, die dem jeweiligen Schutzbedarf gerecht wird, sich nahtlos in bestehende Infrastrukturen eingliedert und die Sicherheit erhöht ohne die Nutzbarkeit der Systeme drastisch einzuschränken. Hierzu muss der Blick auch auf die umgebenden Infrastrukturen gerichtet sein. Darüber hinaus erfordert Security immer auch einen Eingriff in bestehende organisatorische Abläufe und personelle Strukturen.

Der vorliegende Beitrag hat einen Überblick über die in diesem Kontext wichtigen Strategien gegeben.

Kontaktadresse:

Dr. Günter Unbescheid

Database Consult GmbH

Laich 9 1/9

D-83676 Jachenau

Telefon:	+49 (0) 8043 1010
Fax:	+49 (0) 8043 1011
E-Mail	g.unbescheid@database-consult.de
Internet:	www.database-consult.de