Security Inside Out

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

Die neue Datenbank bietet im Security-Bereich zahlreiche Neuerungen. Diese sind in erster Linie nicht revolutionär, sondern vorwiegend evolutionär. Der Artikel zeigt die Neuerungen im Gesamtkontext der Oracle-Datenbank-Security-Strategie.

Zwei Drittel aller gestohlenen Daten stammen aus Datenbanken. Das ist nicht verwunderlich: Einerseits werden Datenbanken in erster Linie zur Speicherung und Verarbeitung wichtiger Daten eingesetzt. Andererseits verlieren viele Unternehmen aus den Augen, dass Datenbanken, die im Zentrum der EDV stehen, einen erhöhten Schutzbedarf haben. Dieser Schutz ist nicht durch die trügerische Sicherung der Peripherie der EDV-Systeme – etwa durch Firewalls und Virenscanner – zu erreichen, sondern muss in der Datenbank selbst und in ihrem unmittelbaren Umfeld erfolgen. Dabei sind drei Bereiche strategisch wichtig: Maßnahmen zur Prävention, Detektion und Administration (siehe Abbildung 1). Datenbank-Features und -Optionen sowie weitere Produkte wie Enterprise Manager Cloud Control mit seinen Packs sowie Oracle Audit Vault and Database Firewall bieten eine technische Unterstützung für diese Maßnahmen.

Prävention: Verschlüsseln von Daten

Grundsätzlich sind zwei Bereiche zu unterscheiden, die für die Verschlüsselung relevant sind; einerseits die Netzwerk-Verschlüsselung, andererseits die Verschlüsselung von Daten auf Speichermedien. Mit der Database 12*c* sind alle Maßnahmen zur Verschlüsselung des Netzwerks, also sowohl die Verschlüsselung über SQL*Net als auch diejenige über SSL nicht mehr Bestandteil der Advanced Security Option (ASO), sondern können von jedem Kunden in allen Datenbank-Editionen (Standard und Enterprise Edition) und in allen Datenbank-Versionen (12.1.0, 11.x.x), die das technisch unterstützen, kostenlos eingesetzt werden. Das gilt übrigens auch für alle Formen der starken Authentifizierung, zum Beispiel über Smartcards, Kerberos-Tickets etc.

Nach wie vor lizenzpflichtig ist der Einsatz von ASO Transparent Data Encryption (TDE). Diese bietet die Verschlüsselung von Tabellen-Spalten und ganzer Tablespaces, ohne dazu Anwendungen verändern zu müssen, die Verschlüsselung von Backups mit RMAN sowie die Verschlüsselung von Dump-Dateien aus Oracle Data Pump.

Die Verschlüsselung von Daten in der Datenbank mit TDE erreicht dabei ein höheres Sicherheitsniveau als die ganzer Festplatten über spezielle Hard- oder Software, da der Zugriff auf die in der Datenbank mit TDE verschlüsselten Daten einen Zugang zur Datenbank und die entsprechenden Objekt-Privilegien erfordert. Dagegen kann jeder, der Zugriff auf das Betriebssystem einer verschlüsselten Festplatte hat, die in der Datenbank gespeicherten Daten über Betriebssystemkommandos auslesen, etwa über den UNIX/Linux-Befehl „strings“.

PRÄVENTION	DETEKTION	ADMINISTRATION
Verschlüsseln	Aktivitäten überwachen	Privilegien analysieren
Redigieren und Maskieren	Auditieren und Berichten	Sensible Daten finden
Privilegierte Benutzer kontrollieren	Database Firewall	Konfigurationen verwalten

Abbildung 1: Security Inside Out – die Oracle-Security-Strategie

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE ,$ORACLE_HOME/net-
work/admin' IDENTIFIED BY einpasswort
```

Listing 1

```
DBMS_REDACT.ADD_POLICY(
   object_schema       => 'scott',
   object_name         => 'emp',
   column_name         => 'empno',
   policy_name         => 'empno_redact',
   function_type       => DBMS_REDACT.PARTIAL,
   function_parameters => '9,2,4',
   expression          => '1=1');
```

Listing 2

Im Bereich von TDE bietet 12*c* vor allem eine neue Syntax zur Verwaltung der Dateien und Hardware Security Module (HSM), in denen die Schlüssel für die Verschlüsselung gespeichert sind. Diese neue Syntax wurde auch durch die veränderten Anforderungen an ein Oracle-System notwendig, das als Container Database mit mehreren Pluggable Databases aufgesetzt ist. Beispielsweise ist es in solchen Systemen nötig und möglich, Schlüssel aus unterschiedlichen Datenbanken in einem Keystore (früher Wallet) zu verwalten. Listing 1 zeigt ein Beispiel. Neben dieser neuen Syntax wurden weitere Views zum Monitoring und zur Verwaltung der Keystores eingeführt.

Prävention: Redigieren und Maskieren

Unter „Redigieren" versteht man bei Oracle das Verändern von Daten ausschließlich für die Ausgabe. Die Veränderung kann für unterschiedliche Benutzer unterschiedlich erfolgen. Wesentlich ist, dass dabei die ursprünglichen Daten nicht verändert werden. Sie können sogar im Rahmen aller DML-Befehle (SELECT, INSERT, UPDATE, DELETE) weiterverwendet werden.

Ein bekanntes Beispiel für die Nutzung ist die Ausgabe von Kreditkarten-Nummern: Die ersten Stellen werden durch ein definierbares Zeichen – häufig ein Asterisk (*) – ersetzt. Nur die letzten Stellen zeigen wirkliche Werte an. Ein praktisches Beispiel zur Anzeige der Personalnummer (EMPNO) aus der Tabelle EMP verdeutlicht die Vorgehensweise und das Ergebnis (siehe Listing 2).

Die Prozedur „ADD_POLICY" aus dem Package „DBMS_REDACT" formuliert eine Regel namens „EMPNO_REDACT", mit der die Personalnummer zum Teil verändert wird: Alle Ziffern von der zweiten bis zur vierten Stelle werden durch die Zahl 9 ersetzt. Das geschieht bei jeder Ausgabe („1=1") und muss nicht mehr in unterschiedlichen Anwendungen neu programmiert werden. Nur bei einer Abfrage durch den Benutzer „SYS" wird die Regel ignoriert, da dieser über das Privileg „EXEMPT REDACTION POLICY" verfügt – vergleichbar mit dem aus Virtual Private Database (VPD) und Oracle Label Security (OLS) bekannten Privileg „EXEMPT ACCESS POLICY". Listing 3 zeigt die Abfrage und das Ergebnis zu diesem Beispiel.

An dem „SELECT"-Befehl wird deutlich, dass die „EMPNO" im Rahmen der Abfrage durchaus verwendet werden kann. Das Redigieren betrifft ja nur die Ausgabe. Das Ergebnis zeigt dann, dass bei der Ausgabe der „EMPNO" von der zweiten bis zur vierten Stelle wie gewünscht immer die Zahl 9 statt der tatsächlich vorhandenen Zahl ausgegeben wird. Abschließend noch der wichtige Hinweis, dass das Redigieren nur im Rahmen von ASO zur Verfügung steht.

In der Oracle-Terminologie bedeutet „Maskieren", dass Produktionsdaten

```
SELECT empno, ename, job, sal, deptno
FROM emp
WHERE empno > 7800;

EMPNO   ENAME    JOB        SAL     DEPTNO
------  -------  ---------  ------  --------
7999    KING     PRESIDENT  5000     10
7999    TURNER   SALESMAN   1500     30
7999    ADAMS    CLERK      1100     20
7999    JAMES    CLERK      950      30
7999    FORD     ANALYST    3000     20
7999    MILLER   CLERK      1300     10
```

Listing 3

```
CREATE AUDIT POLICY doag2013
PRIVILEGES ALTER ANY TABLE,
           CREATE ANY TABLE,
           DROP ANY TABLE, ...
ACTIONS    ALTER USER,
           CREATE ROLE,
           ALTER ROLE, ...
WHEN ,SYS_CONTEXT
     (''USERENV'', ''IP_ADDRESS'') <> (''appserver_ip'')';
AUDIT doag2013;
```

Listing 4

```
DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
   name => 'personalanw',
   type => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
   condition =>
      'SYS_CONTEXT(''USERENV'', ''MODULE'') = ''PERSANW''');
...
EXEC DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ('persanw');
...
EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ('persanw');
EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ('persanw');

SELECT * FROM dba_used_sysprivs;
SELECT * FROM dba_used_objprivs_path;
```

Listing 5

nachhaltig und in der Regel irreversibel verändert werden. Dabei behalten sie typische Eigenschaften wie Primär-/Fremdschlüssel-Beziehungen oder ihre Verteilungs-Charakteristika. Das leistet schon länger das kostenpflichtige Data-Masking-Pack des Enterprise Manager Cloud Control. Neu ist hier, dass die zu maskierenden Daten nicht zuerst dupliziert werden müssen, sondern in Dateien im Data-Pump-Export-Format geschrieben und dann in Test- oder Entwicklungssysteme importiert werden können. Dieses Feature kann auch kombiniert werden mit der Möglichkeit, nur Teilmengen von Produktionsdaten zu extrahieren („data subsetting").

Prävention: Privilegierte Benutzer kontrollieren

Security-Experten haben lange bemängelt, dass viele DBA-Aufgaben die Privilegien des Benutzers SYS erforderten. Das neue Datenbank-Release bietet hier jetzt wirksame Alternativen. Es können nämlich eigene Gruppen für die Bereiche Backup, Keystore und Standby Management festgelegt werden. Das geschieht – unter UNIX und Linux – auf die gleiche Weise, in der schon bisher die DBA-Gruppe festgelegt wird. Das heißt, es werden Betriebssystemgruppen für die einzelnen Bereiche eingerichtet und diesen Gruppen Benutzer zugewiesen. Diese können sich dann bei der Datenbank etwa mit „sqlplus / as syskm" als Keystore-Manager anmelden. Sie haben alle Berechtigungen, die sie für ihre Arbeit mit den Keystores benötigen, aber zum Beispiel keinerlei Rechte zum Anlegen oder Ändern von Benutzern.

Außerdem gibt es analog zum „SYSDBA"-Privileg noch die Privilegien namens „SYSDG", „SYSBACKUP" und „SYSKM". Diese wirken allerdings nur bei geöffneter Datenbank. Auch das ist vom „SYSDBA"-Privileg bekannt, das allein ja zum Beispiel ein Starten der Datenbank nicht ermöglicht. Dazu ist die Zugehörigkeit zur „DBA"-Gruppe erforderlich.

Auch die Optionen Database Vault (DV) und Label Security (OLS) dienen der Kontrolle privilegierter Benutzer. In der neuen Datenbank-Version sind sie automatisch installiert – wie die übrigen Optionen auch. DV und OLS müssen nur noch aktiviert werden. Das vermeidet Probleme, die durch das nachträgliche Installieren immer wieder auftraten, weil die in den entsprechenden Handbüchern beschriebenen Schritte nicht korrekt durchgeführt wurden. Zudem können nun Datenbanken aus einem Oracle-Home-Directory mit und ohne Database Vault betrieben werden.

Detektion: Aktivitäten überwachen

Im Laufe der Jahre hat sich im Bereich des Auditing ein Wildwuchs an Möglichkeiten und Speicher-Formen entwickelt. Das neue Release führt nun mit dem neuen Feature „unified auditing" alles auf eine einheitliche Vorgehensweise und einen einheitlichen Speicherort zurück. Das neue Auditing ist bei Neu-Installationen grundsätzlich konfiguriert; bei älteren Datenbanken, die auf die neue Software umgestellt werden, wird es durch Hinzulinken eines speziellen Codeteils aktiviert.

Alle Audit-Daten – auch die von Database Vault - sind beim „unified auditing" innerhalb der Datenbank in einer einzigen Tabelle gespeichert. Diese ist im Tablespace „SYSAUX" abgelegt, lässt sich aber problemlos in ein anderes Tablespace verlagern.

Die Tabelle gehört dem Benutzer „AUDSYS". Die Rollen „AUDADMIN" und „AUDVIEWER" berechtigen zur Einstellung beziehungsweise zum Lesen des Audit-Trails. Der Eigentümer eines Objekts kann das neue Auditing nur dann verwenden, wenn er auch über die Rolle „AUDADMIN" verfügt.

Das Auditieren mit „unified auditing" ist unabhängig von Initialisierungsparametern – vergleichbar mit dem bekannten Fine Grained Auditing (FGA). Gesteuert wird es ebenfalls wie beim FGA über Regeln („PoliciesW) und deren Aktivierung mit dem Befehl „Audit" (siehe Listing 4).

Das Beispiel listet zunächst eine Reihe von Privilegien und Aktionen auf, die auditiert werden sollen. Im letzten Teil wird eine Bedingung formuliert, die erfüllt sein muss, damit ein Audit-Eintrag erfolgt. Die Steuerung der Einträge ist zwar über die Bedingungen vom FGA bereits bekannt, aber dort funktioniert das nur für DML-Befehle (INSERT, UPDATE, DELETE, SELECT). Im „unified auditing" funktioniert diese Steuerung dagegen für alle Audit-Möglichkeiten. Das ist natürlich ein deutlicher Fortschritt gegenüber dem bisherigen Standard-Auditing. Außerdem ist das Auditing performanter geworden und unterstützt auch SQL*Loader, Data Pump und RMAN. All diese Punkte sollten dafür sorgen, dass sich „unified auditing" sehr schnell als Standard-Variante des Auditing durchsetzen wird.

Detektion: Auditieren und Berichten/ Database Firewall

Die Beschreibung der Oracle-Strategie im Bereich „Security" wäre unvollständig, wenn man nicht auch auf das relativ neue Produkt Oracle Audit Vault and Database Firewall (AVDF) eingehen würde. Diese beiden Komponenten, die das Produkt ausmachen, wurden zunächst separat mit deutlichen

Überschneidungen eingesetzt. Die Audit-Vault-Komponente bietet die Möglichkeit, Audit-Daten unterschiedlicher Oracle- und auch Nicht-Oracle-Datenbanken sowie beliebiger weiterer IT-Komponenten zu konsolidieren. Hinzu kommt die Möglichkeit, Alarme auszulösen und Berichte auf Basis von Application Express sowie Test-Daten über Workflows zu erstellen.

Die Database-Firewall-Komponente schützt Oracle- und eine Reihe von Nicht-Oracle-Datenbanken vor SQL*Injection-Angriffen. Zusätzlich kann sie ein Protokoll aller Zugriffe auf diese Datenbanken erzeugen – allerdings lediglich ein Protokoll der Zugriffe, die über das Netzwerk erfolgen. Das bedeutet, dass lokale Zugriffe anders, etwa bei Oracle-Datenbanken mit „unified auditing“, erfasst werden müssen.

Beide Komponenten, also Audit Vault Server und Database Firewall, werden als Software-Appliance geliefert: Der Kunde stellt die Hardware (X86 64 Bit) und auf dieser wird exklusiv die von Oracle gelieferte Software installiert. Das Ergebnis ist eine Appliance, die nur nach den Vorgaben von Oracle verändert werden darf.

Administration: Privilegien analysieren, sensible Daten finden, Konfigurationen verwalten

Die neueste Version von Database Vault erfasst über das Package „DBMS_PRIVILEGE_CAPTURE“, welche Privilegien ein Anwender oder eine Anwendung tatsächlich nutzt. Dies ist ein bedeutender Schritt auf dem Weg, das Konzept des „least privilege“ durchzusetzen. Das folgende Beispiel erläutert die Vorgehensweise zum Erfassen der Privilegien, die eine fiktive Anwendung zur Personalverwaltung namens „PERSANW“ benötigt.

Zunächst wird mit „CREATE CAPTURE“ das Erfassen der Privilegien definiert, dann mit „ENABLE_CAPTURE“ und „DISABLE_CAPTURE“ das Erfassen gestartet und nach angemessener Zeit, in der mit der Anwendung gearbeitet wird, gestoppt. Schließlich wird das Ergebnis mit „GENERATE_RESULT“ in Hilfstabellen abgelegt. Diese sind als Views aufbereitet und über „SELECT“-Befehle auswertbar. Damit erhält man Auskunft über die verwendeten Privilegien – hier die System-Privilegien, die in „DBA_USED_SYSPRIVS“ abzufragen sind, – und Informationen darüber, auf welchem Weg, zum Beispiel direkt oder über Rollen, diese Anwendung die vorhandenen Objektprivilegien erhalten hat („DBA_USED_OBJPRIVS_PATH“, siehe Listing 5).

Nachdem sich die Frage nach verwendeten Privilegien also durch den Einsatz eines Package aus dem Lieferumfang von Database Vault klären lässt, bleibt die Frage, wie man schützenswerte Daten findet. Die Frage mag auf Anhieb banal klingen, aber für viele, vor allem große Unternehmen stellt sie sich massiv. Sie lässt sich mit der Funktionalität des „sensitive data finding“ aus den Enterprise-Manager-Cloud-Control-Packs „Data Masking“ und „Test Data Management“ klären. Diese sind zwar kostenpflichtig, allerdings lässt sich „sensitive data finding“ kostenlos nutzen, sofern für die betroffene Datenbank eine Security-Option (ASO, DV oder OLS) lizenziert ist.

Als letzte Facette der Oracle-Security-Strategie sei noch auf ein weiteres kostenpflichtiges Pack des Enterprise Manager Cloud Control hingewiesen, das Oracle-Database-Lifecycle-Management-Pack. Es ermöglicht, Konfigurationen von Oracle-Datenbanken zu überwachen, zum Beispiel im Hinblick auf die dort verwendeten Initialisierungs-Parameter zur Sicherheit der Datenbank. Sobald diese Parameter verändert werden, signalisiert das Pack die Veränderung und der (Security)-DBA kann darauf reagieren. Dies unterbindet den sogenannten „configuration drift“, also das Abweichen von individuell festzulegenden Normen im Bereich der Datenbank-Konfiguration.

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