DBSAT – Erfahrungsbericht

Thorsten Grebe, November 2018

Schlüsselworte

DBSAT, Database Security, EU-DSGVO, GDPR, Security-KPI

Einleitung

Oracle hat mit dem Database Security Assessment Tool (DBSAT) ein Werkzeug bereitgestellt, das komfortabel und umfassend den Sicherheitsstatus eines Datenbankservers ermitteln kann. Es analysiert nicht nur sicherheitsrelevante Einstellungen innerhalb der Datenbank, sondern auch Einstellungen von Listener, Datei- und Betriebssystem. Insgesamt können von DBSAT zurzeit über 80 Parameter begutachtet werden. Das Ergebnis wird in vier alternativen Formaten bereitgestellt (CSV, HTML, TXT, JSON). Zusätzlich erstellt DBSAT eine Management-taugliche Übersicht, die als Messlatte verwendet werden kann, um Verbesserungen oder Verschlechterungen im Sicherheitsstatus einer einzelnen Datenbank oder einer gesamten Datenbanklandschaft quantifizieren zu können.

Die erste Version von DBSAT wurde im Mai 2016 zunächst ohne weitere Marketingoffensiven veröffentlicht. Das Freigabedatum der Version 1.0.0 fällt nicht zufällig auf den Monat, in dem die neue Datenschutzgrundverordnung der EU (EU-DSGVO, engl. GDPR) in Kraft trat. Den meisten Administratoren entging dieses neue Werkzeug vermutlich für mindestens ein Jahr. Erst zum Ende 2017 begann Oracle für sein neues Werkzeug zu werben. Bei der DOAG-Konferenz in Nürnberg, bei Regionalveranstaltungen und bei speziellen Customer Events für Endkunden.

Dieser Erfahrungsbericht versucht die Frage zu beantworten, ob und wie DBSAT mit seinen unterschiedlichen Modulen beim gesetzeskonformen Absichern sensibler Oracle Datenbanken unterstützen kann.

Versionen und Funktionen von DBSAT

Die erste Version von DBSAT verfügte über zwei Module: den Collector und den Reporter. Der Collector sammelt sicherheitsrelevante Daten und schreibt sie in eine wahlweise verschlüsselte JSON-Datei. Der Reporter wertet diese aus und stellt sie als menschenlesbare Berichte dar. Die Version 2 wurde im Dezember 2017 zum Download (ausschließlich über MOS ID 2138254.1)

bereitgestellt. Sie bringt drei wesentliche Neuerungen: (1) Zusätzlich zu den Ausgabeformaten Text, HTML und Excel erzeugt der Reporter jetzt auch einen Ergebnisbericht im JSON-Format, der für eine automatisierte Verarbeitung besser geeignet ist. (2) Neu sind auch die ausführlichen Bezüge innerhalb des Berichts zu Paragraphen der EU-DSGVO und zu den CIS Benchmarks. Dadurch wird die Relevanz von Befunden im DBSAT-Bericht verdeutlicht. (3) Besonders hob Oracle jedoch eine Neuerung in DBSAT 2 hervor, die zunächst nicht für das Tool vorgesehen war: den *Discoverer*. Dieser soll in der Lage sein, sensible Daten innerhalb einer Datenbank ressourcenschonend einzig aufgrund von Metadaten-Abfragen aufzuspüren und anzuzeigen. Der Discoverer ist ein unabhängiges Add-on zu DBSAT, das einen eigenen Bericht erstellt, das Programm arbeitet entkoppelt von den Collector und Reporter Modulen.

Oracles Motivation zur Entwicklung von DBSAT

DBSAT ist für Oracle Kunden kostenlos. Es wird von einem Entwicklungsteam weiterentwickelt, Updates sollen regelmäßig zur Verfügung gestellt werden. Ein Dokumentationsteam hat eine solide, ausführliche Dokumentation verfasst, Whitepaper wurden geschrieben, aus dem Englischen ins Deutsche übersetzt. Die Marketing-, Vertriebs- und Event-Maschinerie wurde angeworfen. Warum macht Oracle das?

Die ursprüngliche Motivation zur Entwicklung von DBSAT erwähnt Pedro Lopez, der Product Manager für den Bereich *Oracle Database Security*, in einem Interview (youtube.com/watch?v=XsPuiCPcyA0): Innerhalb Oracle wurde in den Jahren vor der Entwicklung von DBSAT festgestellt, dass weltweit in unterschiedlichen Teams von Oracle unabhängig voneinander individuelle Skriptsammlungen und Hilfsprogramme entwickelt wurden, um den Sicherheitsstatus von Kundendatenbanken auf *Good Practice* Vorgaben zu prüfen. In Deutschland ist aus einem solchen Einzelengagement das 2013 veröffentlichte Sicherheitsbuch „Oracle Security in der Praxis“ von Carsten Mützlitz hervorgegangen, das bis heute in seiner umfassenden und praxisbezogenen Sicht auf den Sicherheitsstatus einer Datenbank konkurrenzlos ist. Offenbar wollte Oracle die unterschiedlichen, redundanten Anstrengungen zusammenführen und statt zahlreicher Einzellösungen, ein gemeinsames Werkzeug entwickeln, das die Erfahrungen und das *Know How* vieler erfahrener Sicherheitsexperten vereint.

In der Dokumentation wird die Motivation zur Bereitstellung von DBSAT etwas knapper formuliert: Oracle möchte seinen Kunden helfen.

In den Ergebnisberichten von DBSAT wird auf die zahlreichen Sicherheitsprodukte von Oracle hingewiesen, die fast ausnahmslos eine Enterprise Edition voraussetzen. Verweise auf Artikel der EU-

DSGVO stimmen besorgt, ob Verstöße gegen Auflagen der neuen, bußgeldgewaltigen Datenschutzregelung nachgewiesen werden könnten.

Die Motivation des Datenbank Administrators zur Verwendung von DBSAT

Für den DBA, der die sichere Konfiguration einer Datenbank gewährleisten soll, stellt DBSAT eine enorme Erleichterung dar. Zwar kann man selbst Skripte und Routinen schreiben, die auf sichere Konfigurationen prüfen. Es gibt dafür ausgezeichnete Anleitungen, Anregungen und Monographien für den Start (Haas, Mützlitz, Knox, Litchfield, Nanda, Natan, u.a.), so dass ein ambitionierter DBA oder Entwickler nicht auf der grünen Wiese beginnen müsste, wenn er eigene Sicherheitswerkzeuge entwickeln wollte. Jedoch ist der initiale Einarbeitungs- und Entwicklungsaufwand hierfür beträchtlich. Das Thema ist umfangreich und komplex. Schlimmer noch: Die in Eigenproduktion geschriebenen Routinen veralten (Einführung von Unified Auditing, Umstellung auf Container Datenbanken, neue Views, veränderte Spalten oder Inhalte in bekannten Views, …). Sie können aus eigener Kraft kaum so aktuell und umfassend sein, wie es DBSAT derzeit vormacht. Hier liegt die Stärke von DBSAT. Es kann als externes, geprüftes und robustes Modul in eigene Reporting-Routinen eingearbeitet werden. Oracle kümmert sich um die Aktualisierung und die fehlerfreie Ausführung. Dem DBA wird eine Sicherheits-Todo-Liste in die Hand gegeben, die er nur noch abarbeiten muss. Auch entfällt die leidige Diskussion um Bedeutung und Schwere von festgestellten Befunden im Kollegenkreis – Oracle gibt schließlich einen scheinbar objektiven und ernst zu nehmenden Standard vor, auf den man sich verständigen kann.

Die Motivation des Security Administrators zur Verwendung von DBSAT

Der Security Administrator interessiert sich nicht für die hoch aufgelösten Details und Handlungsanweisungen, die DBSAT dem Datenbank Administrator zur Verfügung stellt. Der Security Administrator möchte einen verlässlichen Überblick, er interessiert sich für aggregierte Zusammenstellungen oder Scoring-Werte. Verbessert sich der Sicherheits-Status einer Datenbank oder einer kompletten Datenbankumgebung, stagniert er oder verschlechtert er sich sogar? Wichtig für ihn ist, dass die Ermittlung von Übersichten und Scoring-Werten nachvollziehbar ist und nach hinreichend objektiven Kriterien erfolgt. Auch sollte die Erhebung nicht so mühselig und anstrengend sein, dass sie niemand wiederholen mag. Scoring-Werte sollen reproduzierbar, nachvollziehbar und mit geringem Aufwand aktualisierbar sein. Genau hier kann DBSAT auftrumpfen. Die Scoring-Matrix, die DBSAT für jede einzelne Datenbank erstellt, lässt sich über alle Datenbanken aufsummieren und als Basis zur Berechnung eines Security Key-Performance-Indikators für die Oracle Datenbanklandschaft verwenden.

Das Sammeln von Daten: dbsat -collect

Der Aufruf zum Sammeln von Daten einer Instanz in seiner einfachsten Form geht wie folgt:

```
dbsat collect -n "/ as sysdba" /tmp/dbsat-collect-ergebnis
```

Dieser Aufruf analysiert den Sicherheitsstatus einer Datenbank und sammelt die Ergebnisse in einer JSON-Datei mit dem Namen /tmp/dbsat-collect-ergebnis.json. Der Parameter "-n" verhindert die Verschlüsselung der Ergebnisdatei, sonst würde eine Aufforderung zur Kennworteingabe erfolgen, mit der die Datei in einem ZIP-Archiv verschlüsselt würde. Der Connect „/ as sysdba" verbindet dbsat über Betriebssystemauthentifizierung mit der Datenbankinstanz bzw. dem Root Container, für die auf dem lokalen System ORACLE_SID gesetzt ist. Stattdessen lässt sich ein Connect-String angeben wie „`SCOTT/Passwort@pdb1`", wobei SCOTT in der aktuellen Datenbankversion mindestens die Rollen `select_catalog_role` und `audit_viewer` sowie direkte Select-Grants für `registry$history`, `dba_users_with_defpwd` und `audsys.aud$unified` besitzen muss, um die notwendigen DBSAT-Abfragen ausführen zu können.

In einer Container-Datenbank muss ein Aufruf pro Container erfolgen, also einmal für den Root-Container und jeweils einer pro PDB. Der Collector sollte lokal ausgeführt werden, da nur lokal auch die Prüfungen von Listener-, Dateisystem- und Betriebssystemeinstellungen möglich sind.

Das Auswerten von Daten: dbsat -report

Die vom Collector gesammelten Rohdaten werden vom Reporter zu einem Bericht aufbereitet. Hierzu ist keine Datenbankverbindung mehr erforderlich. Die Umwandlung zu Berichtsdateien kann also auch auf einem zentralen Auswerterechner erfolgen. In diesem Beispiel werden die Berichte jedoch gleich an Ort und Stelle erzeugt, und zwar mit folgendem Aufruf:

```
dbsat report -n /tmp/dbsat-collect-ergebnis
```

Die Endung ".json" ergänzt dbsat selbst. Dieser Aufruf erzeugt den DBSAT Ergebnisbericht in vier alternativen Dateiformaten, deren Inhalte überwiegend gleich, aber nicht identisch sind. So fehlt im Excel beispielsweise die Risiko-Score Matrix. Oracle begründet die Unterschiede damit, dass die Dateien für unterschiedliche Verwendungszwecke gedacht seien: die Textdatei für Rückwärtskompatibilität, das Excel als editierbare Arbeitsanweisung, das JSON für die automatisierte Weiterverarbeitung und der navigierbare HTML-Bericht für das detaillierte Studium.

Aggregieren von DBSAT-Ergebnissen

DBSAT aggregiert keine Daten. Man erhält Einzelberichte. Bei 100 Datenbanken sind das 100 Ausführungen von DBSAT mit 100 Ausgabeberichten. Der Traum von der einfach zur ermittelnden Gesamtscore scheitert hier bereits, wenn die Konsolidierung der Daten nicht geskriptet werden kann. Doch das ist möglich. Die CSV-Dateien lassen sich mit dem beigefügten xlsxwriter allerdings nicht zu einem großen Excel konsolidieren. Denn dieses freie Programm kann die erzeugten Exceldateien nicht weiterverarbeiten, es kann lediglich einzelne Read-Only Worksheets erstellen. Die HTML-Ausgabe ist zum Aggregieren von Natur aus ungeeignet. Betagte DBAs verarbeiten daher die Textdateien und ziehen die gewünschten Datenzeilen über bewährte Shell-Kommandos aus den Einzeldateien, um sie zu einem Gesamtergebnis zusammenzuführen. Das geht am einfachsten, ist extrem schnell, hat einen vernachlässigbaren Fußabdruck, ist aber ganz und gar nicht en vogue. Dem Modernen aufgeschlossene Kollegen greifen deshalb zu den JSON-Dateien. Der Aufwand ist beträchtlich größer, es werden unverhältnismäßig viele Daten umgewälzt, der Umgang mit der Syntax ist eine Qual, aber die JSON-Auswertung bietet dafür die höchste Flexibilität. Beispielsweise lassen sich die JSON-Dateien per SQL*Loader direkt in eine Monitor-Datenbank laden. Einmal in einer Tabelle gespeichert, kann auf die einzelnen Bereiche des JSON Berichts unmittelbar per SQL zugegriffen werden (ab 12c gelingt dies nativ mit SQL, unter 11g unter Mitwirkung von Apex-Routinen). Einmal aggregiert, lassen sich die Daten der DBSAT Kollektoren nach Herzenslust weiterverarbeiten. Sie können zum Beispiel zur komplett automatisierten Ermittlung eines Key-Performance-Indikators für die eigenen Oracle Datenbanken verwendet werden, der per Scheduler-Job regelmäßig aktualisiert werden kann.

Wie sind das Potential und der Nutzen des DBSAT Discoverers einzuschätzen?

Der Discoverer arbeitet völlig unabhängig vom Collector/Reporter Duo. Er sucht im Data Dictionary der Oracle Datenbank nach Tabellen mit Spalten, die auf sensible, personenbeziehbare Daten hindeuten. Dabei beschränkt er sich auf den Abgleich von Spaltennamen und Spaltenkommentaren mit einer editierbaren Liste von regulären Suchausdrücken, z.B. die 3-Buchstabenfolge "DOB" für "Date of Birth" oder „ADDR“ zum Finden von Tabellenspalten, die Post- oder Email-Adressen enthalten könnten.

Was auf den ersten Blick wie eine gute Idee klingt, erweist sich in der Praxis jedoch als kaum sinnvoll umsetzbar. Wie schwierig es für den Discoverer ist, allein aufgrund von Abfragen gegen das Data Dictionary sensible Spalten zu finden, zeigt sich an der Oracle Datenbank selbst. Denn würde der Discoverer seine eigene Anwendung durchsuchen, was er überraschenderweise nicht tut, dann würde er zielsicher die Tabellenspalten `password` und `password_versions` in der Tabelle `dba_users`

als sensibel anzeigen. In der ersten Spalte steht allerdings seit über einem Jahrzehnt gar kein Kennwort mehr, in der zweiten Spalte eine belanglose Metainformation über Kennwörter. Der Discoverer hätte also zwei Falsch-Positive Meldungen erzeugt. Der brisante Kennworthash, der einen Hacker ohne großen Aufwand zu den Benutzerkennwörtern führen würde, steht in der Spalte `spare4` der Tabelle `user$`. In dieser Spalte stehen aneinandergereiht die unterschiedlich stark bzw. schwach verhashten Kennwörter aller Datenbank-Benutzerkonten. Diese Spalte würde der Discoverer nicht finden, denn "spare4" müsste erst manuell in seine Suchworteliste eingetragen werden. Das Nichtfinden der tatsächlichen Kennwortspalte wäre ein Falsch-Negativer Befund, der unentdeckt bliebe.

Wie unbeholfen sich das Suchen von sensiblen Daten in der Praxis gestalten kann, zeigen auch andere Anwendungen, beispielsweise SAP, dessen überwiegende Anzahl von Tabellen nach einer Vierzeichensystematik und dessen überwiegende Anzahl von Spalten nach einer Fünfzeichensystematik benannt sind. Bei ca. 100.000 Tabellen in einer durchschnittlichen SAP Anwendung müsste der Discoverer aus über einer Million Spaltennamen erraten, wo die sensiblen Daten liegen. Zur Erhebung des Schutzbedarfes einer SAP-Anwendung dürfte dies kaum ein sinnvolles Vorgehen sein. Mit ein wenig Kommunikation käme man dagegen sehr schnell zu einem tragfähigen Ergebnis: man fragt diejenigen, die sich mit der Anwendung auskennen, anstatt erst viel Zeit in das Justieren von Discoverer-Läufen zu investieren.

In selbstentwickelten Anwendungen, die Patente, geheime Rezepturen, Produktideen, Forschungsergebnisse, aktuelle Ausschreibungsangebote, Newsletter, Kranken-, Kunden-, Adressdaten verwalten, können die Spaltenbezeichnungen so individuell ausfallen, dass es schnell müßig wird, eine Liste mit Spaltennamen zu führen, die der Discoverer finden können soll. Machtlos ist der Metadaten-getriebene Suchansatz des Discoverers auch, wenn sich sensible, schützenswerte, persönliche Informationen in Spalten mit Namen wie "kommentare", "anmerkungen", "hinweise", "notizen" oder "infos" befinden. Eine Anwendungsverantwortliche würde dies jedoch sehr wohl wissen und den Schutzbedarf der Anwendung – nicht der Spalte (wie es der Discoverer suggeriert) – erhöhen.

Dass sich der Discoverer auf eine reine Metadatenabfrage beschränkt, ist dennoch positiv zu bewerten. Denn Terabyte große Datenbanken über Dateninhalte, anstatt über Metadaten zu analysieren, würde nicht wenige Sekunden sondern viele Stunden bis Tage dauern. Die negative Beeinträchtigung eines produktiven Datenbankbetriebs ließe sich bei einer Komplettdurchsuchung nicht ausschließen, sie wäre sogar sehr wahrscheinlich. Wie fragwürdig und überaus müßig dieser erweiterte Wir-Durchsuchen-Alles-Ansatz ist, den einige Beraterfirmen zu verfolgen scheinen, wird klar, wenn man an Feldtypen wie LONG, BLOB, BFILE, XMLTYPE denkt oder an externe Tabellen,

die meist leer, aber dann kurzzeitig doch mit Inhalt gefüllt sein können. BLOBs könnten sensible Informationen in Bildform oder Office-Dokumenten enthalten. BFILEs verweisen auf externe Dokumente beliebiger Art und Größe. Für Ermittlungsbehörden mag es von großer Bedeutung sein, Inhalte in konfiszierten Datenträgern durchsuchen zu können. Im produktiven Datenbankbetrieb scheint eine solche Herangehensweise, die für eine Erhöhung des Datenschutzes und zur Immunisierung gegen die EU-DSGVO kaum zuverlässige Ergebnisse liefern kann, überzogen. Es ist deshalb gut und richtig, dass sich der DBSAT Discoverer auf eine schnelle, ressourcenschonende, Metadaten-Abfrage beschränkt.

Ein Risiko, das durch das automatisierte Suchen von sensiblen Daten in Datenbanken selbst entsteht, liegt darin, sich auf die Suchergebnisse eines agnostischen Programms zu verlassen, statt einen Anwendungsverantwortlichen aufzufordern, sich inhaltlich mit einer Anwendung auseinanderzusetzen und deren Schutzbedarf zu definieren. Falsch-Positive mögen lediglich lästig sein, Falsch-Negative sind heimtückisch.

Im Sicherheitsumfeld kennt man sogenannte *Inference Attacks*, bei denen sensible Informationen durch Aggregation oder Rückschluss aus unsensiblen Informationen gewonnen werden. Kein Discoverer der Welt kann darauf hinweisen, dass aus den Berichten, die eine Anwendung erzeugt, durch indirekte Rückschlüsse sensible Informationen preisgegeben werden. Um solche Schwachstellen abzudichten, muss jemand Verantwortung für eine Anwendung übernehmen und sich Gedanken um den Umgang mit dieser machen. Das kann ihm kein Programm abnehmen. Es kann ihn höchstens in falscher Zuversicht wiegen.

Fazit

Das Collector/Reporter Duo von DBSAT erweist sich als wertvolle Ergänzung im Administrationsalltag. Einmal eingerichtet, hilft es dabei, einen etablierten Sicherheitsstandard für die von einem Team verantworteten Oracle Datenbanken aufrecht zu erhalten. Die Scoring Tabelle ist hervorragend als Grundlage für eine vollautomatisierte KPI Erfassung geeignet. Großartig wäre es, wenn der Collector für eigene Abfragen geöffnet werden könnte oder wenn er um optionale Tests erweiterbar wäre, wie z.B. eine Prüfung auf verschlüsselte Client-Verbindungen oder die Prüfung auf Parametrierung des Audit Trails nach den Vorgaben der eigenen Sicherheitsrichtlinien. Für einige Alarme, die nicht abstellbar sind, weil eine Funktion verwendet werden muss (z.B. Directory-Objekte), wäre es sinnvoll, Ackknowledgements hinterlegen zu können.

Ob man sich mit dem Discoverer auseinander setzen möchte, bleibt Geschmacksache. Er kann sehr viel Zeit kosten, ohne einen Mehrwert zu liefern. Findet der Discoverer etwas, muss geprüft werden,

ob der Treffer Falsch-positiv ist. Findet er keine sensiblen Daten, heißt das längst nicht, dass es keine gibt. Allein diese Einschränkung lässt ihn als verlässliches Werkzeug durchfallen. Ob eine Anwendung sensible Daten beherbergt und einen erhöhten Schutzbedarf erfordert, sollte nicht per Discoverer erraten, sondern beim Dateneigentümer / Anwendungsverantwortlichen erfragt werden. Ist niemand als Verantwortlicher für eine Anwendung verfügbar, wird der Discoverer dieses Problem nicht lösen. Wünschenswert wäre es, dass der Collector für die weitere Entwicklung mehr Fokus erhält, der Discoverer weniger.

Kontaktadresse:

Dr. Thorsten Grebe
twg-it Unabhängige Oracle Datenbankberatung
Geisenheimer Straße 6
D-14197 Berlin

Telefon: +49 (0) 30 2160 2722
E-Mail: Thorsten.Grebe@twg-it.de
Internet: https://twg-it.de