ORACLE DATABASE

Datenbank-Sicherheit

Grundüberlegungen

ORACLE WHITE PAPER | AUGUST 2017

ORACLE

Inhaltsangabe

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Warum ist Datenbank-Sicherheit wichtig

Die Größe, Komplexität und Relevanz von Datenbanken nimmt kontinuierlich zu und es gestaltet sich immer schwieriger, Datenbanken ausreichend abzusichern. Die Auswirkungen eines Missbrauchs sind jedoch immens, denn in den Datenbanken lagert das wichtigste Gut eines Unternehmens in konzentrierter Form: Informationen. In vielen Unternehmen hat die Sicherung der Datenbanken mit der gestiegenen Bedrohung nicht Schritt gehalten. Zudem planen oder implementieren bereits viele Kunden private und virtuelle Datenbank-Clouds. Bei der damit verbundenen Konsolidierung werden unterschiedliche Daten mit verschiedenen Sicherheitsanforderungen zentral auf einer Hardware-Plattform zusammengeführt. Damit erhöht sich das Risiko maßgeblich, wenn nicht erhöhtes Augenmerk auf Sicherheit gelegt wird.

Trügerisches Sicherheitsgefühl

Datenbanken stehen unerreichbar in der hinteren Ecke unseres Rechenzentrums. Vor dem Internet durch mindestens drei Firewalls geschützt. Auf jedem Client-PC sind Virenscanner aktiv. Auch noch viele weitere aktive Sicherheitskomponenten wie Intrusion Detection Systems (IDS) sind im Einsatz. Kein weiterer Handlungsbedarf? Aber warum sind trotzdem Datenbanken das Angriffsziel Nr. 1?

Entsprechend des Verizon Data Breach - Reports[1] 2012 laufen 33% aller Angriffe direkt über den Datenbankserver, wobei 98% der Daten kompromittiert werden. Sind hier Insider aktiv? Unterschreiben Sie mit Ihrem Namen, dass in Ihrem Unternehmen keine Schadsoftware[2] aktiv ist? Was ist wenn doch? Wird dann ein Mitarbeiter ungewollt zum Innentäter? Sind Ihre Mitarbeiter so sensibilisiert, dass niemand jemals den Mail-Anhang eines Freundes, Bekannten oder Familienmitgliedes, welches an die private E-Mail Adresse gesendet wurde, mit dem im Büro freigegebenen Browser-Mailclient auf dem Dienst-PC öffnet? Das lässt sich nicht ausschließen!

Der Dienst-PC bleibt unberechenbar, egal welche Schutzmaßnahmen dort implementiert sind. Er ist die Schnittstelle nach außen, die Schnittstelle zu den Kunden und Geschäftspartnern. Über den Dienst-PC werden E-Mails verarbeitet, Dokumente empfangen und versendet. Außerdem werden ab und zu private Aktivitäten durchgeführt, was in der heutigen Zeit durchaus erlaubt ist. Es ist nicht auszuschließen, dass Dienst-PCs technisch hoch privilegierter Mitarbeiter, wie zum Beispiel Datenbankadministratoren oder Systemadministratoren, bereits kompromittiert sind und damit auch entsprechende Schadsoftware IT-Systeme bedroht, welche sich hinter diversen Firewalls in Sicherheit wiegen. Hierunter fallen neben anderen wichtigen IT-Infrastruktur-Komponenten auch Datenbanken. Es sollte nicht die Frage danach gestellt werden, ob Ihr Unternehmen bereits kompromittiert ist, sondern

1 http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2012-ebk_en_xg.pdf

2 https://de.wikipedia.org/wiki/Schadprogramm

nach Lösungen gesucht werden, wie Sie Ihr Unternehmen nachhaltig vor den Auswirkungen eingeschleuster Schadsoftware schützen können.

Das Einschleusen von Schadsoftware wird immer professioneller. Sogenannte Script-Kiddies[3], oft technisch unerfahren, erstellen meist aus Neugier Schadsoftware mittels im Internet verfügbarer „Schadsoftware-Baukästen“. Diese so erstellte Schadsoftware wird über bekannte, aber noch nicht behobene Sicherheitslücken, falsch konfigurierte Clients und Server oder unachtsame Computer-Benutzer eingeschleust.

Eine weitere wesentlich größere Bedrohung durch Schadsoftware kompromittiert zu werden, geht von der organisierten Computer-Kriminalität aus. Das Ziel krimineller Organisationen ist im Prinzip das gleiche wie bei legalen Unternehmen, nämlich Umsatz. Allerdings sind die Methoden krimineller Organisationen jenseits des Legalen. Hier wird meist durch Erpressung (z.B. durch angedrohte Löschung oder Verschlüsselung persönlicher Daten), Identitätsdiebstahl (Kreditkarten Informationen), aber auch durch den Missbrauch des Computers bzw. Servers als Teil eines Bot-Netzes, Umsatz erwirtschaftet.

Die größte Bedrohung jedoch geht von den Regierungen selber aus. Wirtschaftsspionage, Sabotage, Ausspähen von Geheimnissen und vieles mehr gehören heutzutage zum Alltagsgeschäft enorm spezialisierter Regierungs-Abteilungen. Hier sind die Möglichkeiten schier unbegrenzt. Methoden wie das gezielte Ausnutzen sogenannter Zero-Day Exploits[4], Einschleusen von Schadsoftware in Firmware oder das Infiltrieren anderer Regierungen bzw. Unternehmen durch eigene Personen, stellen nur einen Teil der Möglichkeiten dar. Mit genügend Zeit und nahezu endlosen Ressourcen lässt sich jedes computergestützte System kompromittieren.

Beispiel: Datenbank-SQL-Injection

Die meisten Angriffe auf Datenbanken werden durch SQL-Injections durchgeführt. Ein kurzes Beispiel soll demonstrieren, wie ein Schadprogramm, welches den Weg auf einen internen Dienst-PC gefunden hat, Datenbanken kompromittiert und damit jegliche dazwischenliegende Sicherheitstechnologie, wie zum Beispiel Firewalls, mühelos überwindet.

Hierzu führt das Schadprogramm über eine im Unternehmen verwendete Web-Anwendung native Datenbank-SQL-Befehle (SQL-Injections[5]) aus. Diese SQL-Befehle unterliegen nicht den Zugriffsregeln der Applikation, sondern werden mit den Berechtigungen des technischen Datenbank-Benutzers, der

3 https://de.wikipedia.org/wiki/Scriptkiddie

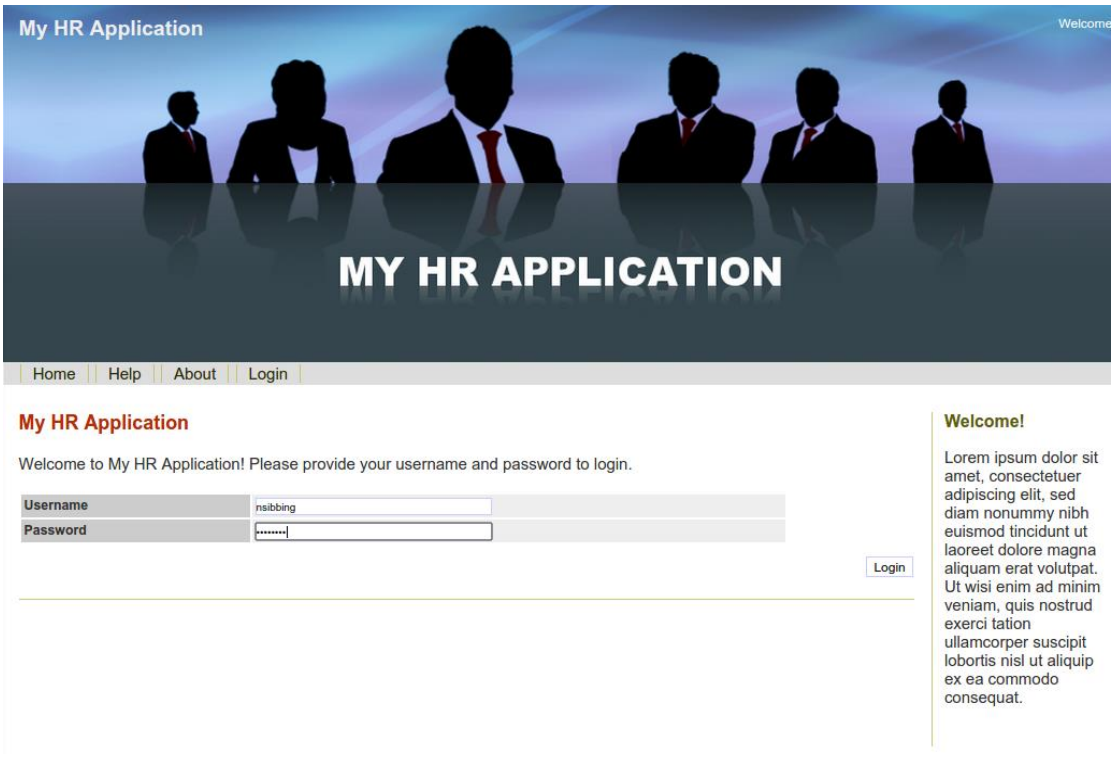
4 https://en.wikipedia.org/wiki/Zero-day_(computing)

5 https://de.wikipedia.org/wiki/SQL-Injection

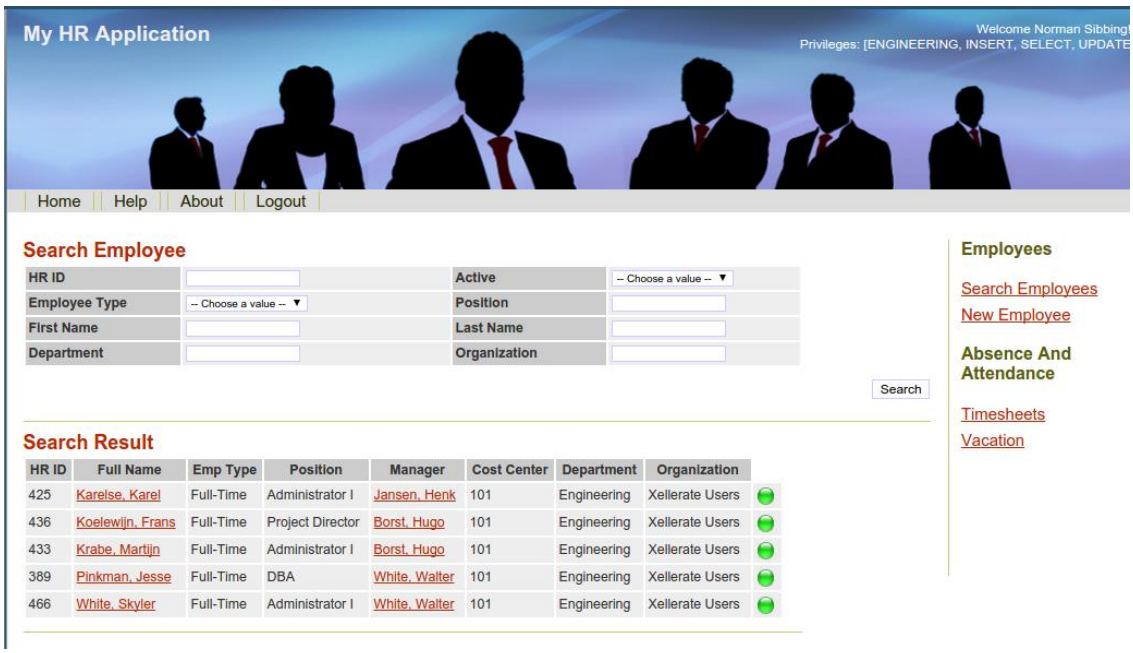
die Anwendung mit der Datenbank verbindet, ausgeführt. Dieser technische Datenbank-Benutzer ist in der Regel hoch privilegiert (meist sogar über-privilegiert). Im schlimmsten Fall lässt sich über ihn die gesamte Datenbank, der komplette Datenbank-Server oder sogar das komplette Netzsegment übernehmen.

Die hier verwendete Web-Anwendung stellt ein Mitarbeiter-Verzeichnis zur Suche von Mitarbeitern zur Verfügung. In Abhängigkeit der Benutzerrolle (HR-Mitarbeiter, Abteilungsleiter, Mitarbeiter) werden unterschiedliche Sichten auf die Mitarbeiterdaten ermöglicht. HR-Mitarbeiter sehen alle Mitarbeiter-Daten des Unternehmens, Abteilungsleiter erhalten die Sicht auf Mitarbeiterdaten ihrer Abteilungen und Mitarbeiter haben einen stark eingeschränkten Zugriff auf Daten der Mitarbeiter ihrer Abteilung.
Die Applikation besitzt hierzu eine eigene Benutzerverwaltung. Diese besteht aus zwei Datenbank-Tabellen: „Benutzer“ und „Rollen“. In Abhängigkeit der Benutzer-Rollen-Beziehung werden entsprechende SQL-Bedingungen an jedes durch die Applikation abgesetzte SQL-Statement angefügt. Somit erhalten die Benutzer nur Zugriff auf Daten, welche sie laut Rolle sehen dürfen. Diese Vorgehensweise ist heutzutage die Standardimplementierung einer Zugriffssteuerung in nahezu jeder Web-Applikation.

Zur leichteren Demonstration verwende ich für die SQL-Injection die Anwendungsoberfläche. Alle hier gezeigten SQL-Injections lassen sich auch ohne Anwendungsoberfläche im Text-Modus durchführen. Im ersten Schritt meldet sich ein Mitarbeiter (Abteilungsleiter) mittels Benutzerkennung und Kennwort an.

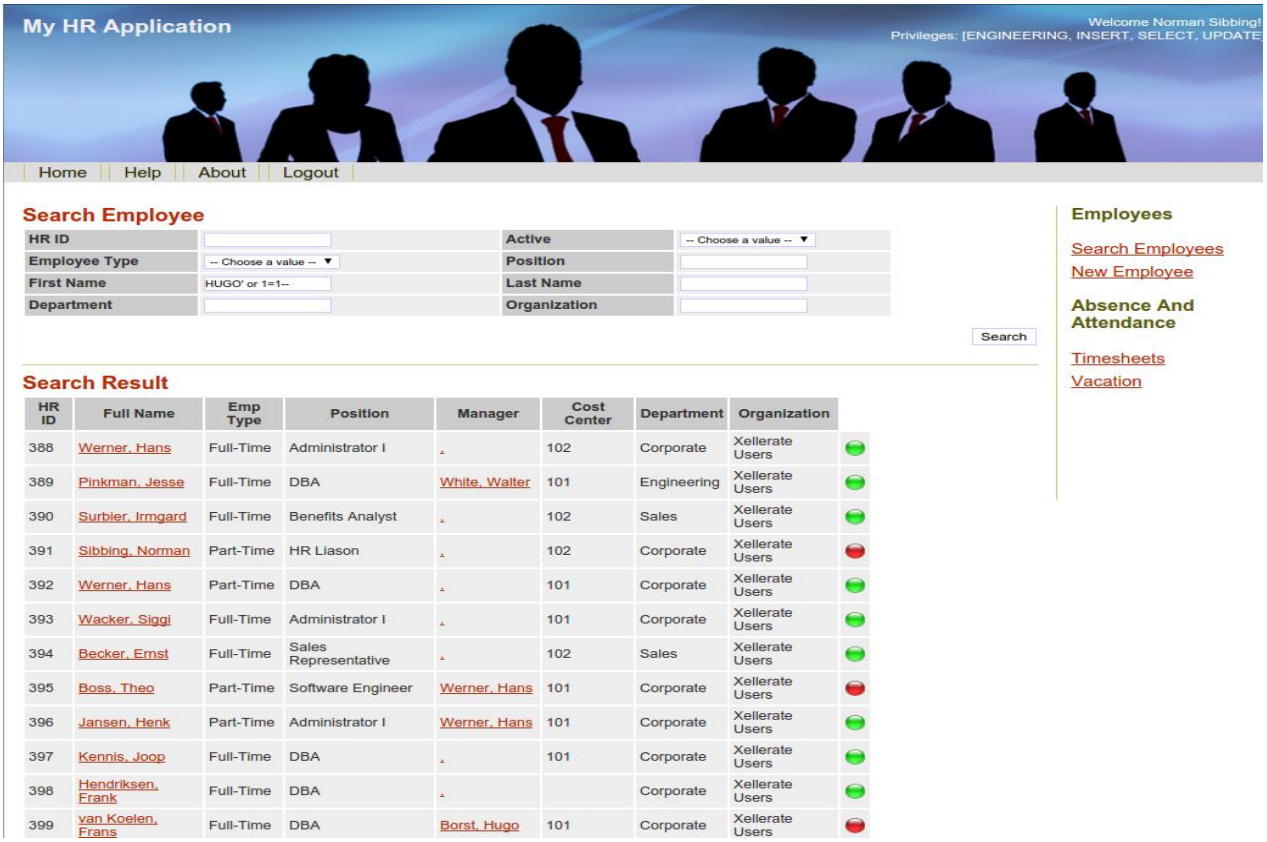


Der Benutzer hat nun gemäß seiner Rolle die Möglichkeit, alle Mitarbeiterdaten seiner Abteilung einzusehen. Um alle seine Mitarbeiter sehen zu können, gibt er keine Suchkriterien an. Durch drücken des Such-Buttons werden nun die Mitarbeiterdaten angezeigt. Die Applikationslogik hat entsprechend der Benutzerrolle das SQL-Statement um eine rollen-abhängige SQL-Bedingung ergänzt.



Wie hier zu sehen ist, wird nur eine kleine Anzahl an Mitarbeiterdaten angezeigt. Das ist auch bezüglich der Benutzerrolle absolut korrekt.

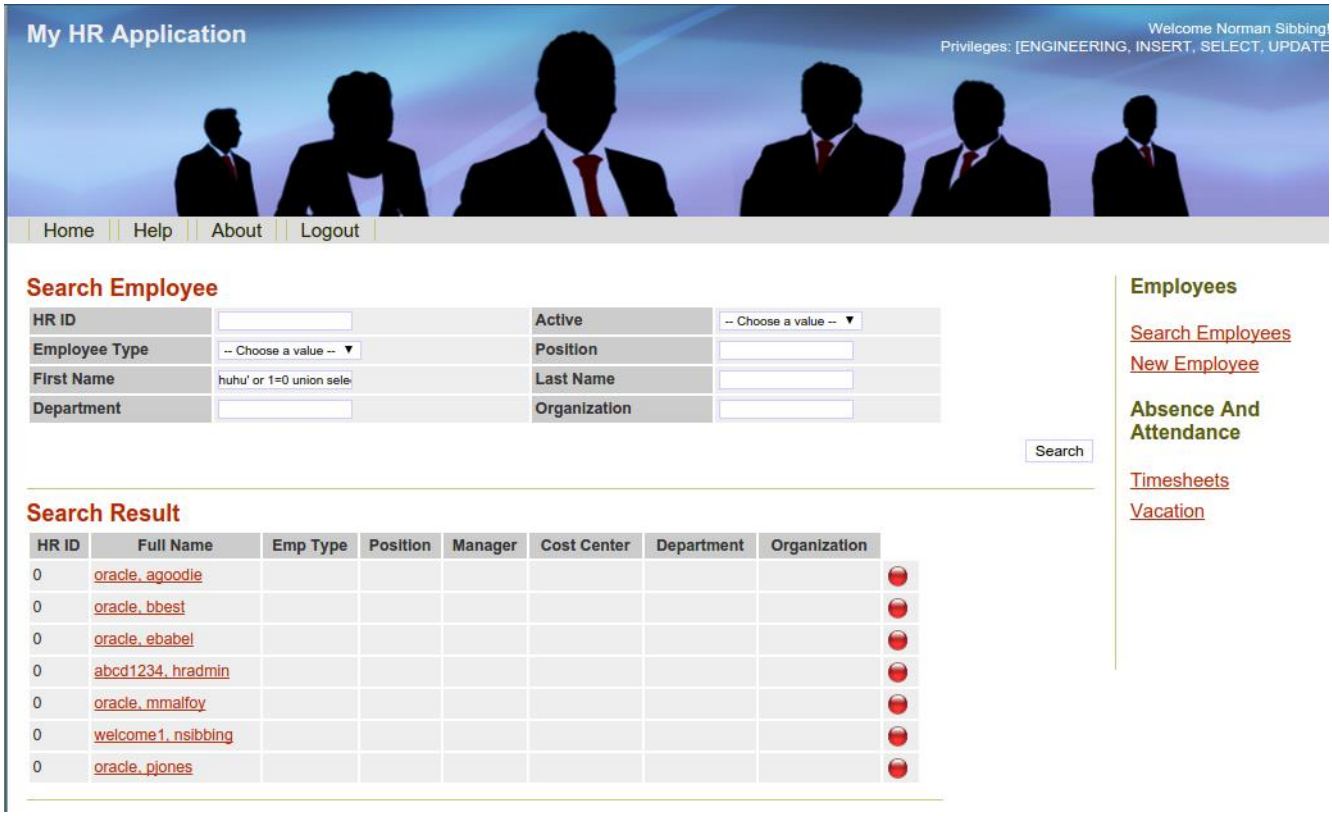
Bei dieser schlecht programmierten Applikation besteht nun die Möglichkeit, eine SQL-Syntax über ein Formular-Feld einzugeben. Anstelle eines normalen Suchbegriffs wie „Becker“ wird ein Ausdruck wie „HUGO‘ or 1=1—„ eingegeben. Der Teil „‘or 1=1—„ ist die SQL-Injection und bedeutet letztendlich, dass alle durch die Applikation angefügten Bedingungen immer zutreffen, das heißt „True, (1=1)“ ergeben und es somit keine Einschränkung in Bezug auf die angezeigten Datensätze mehr gibt. Deshalb werden alle Mitarbeiterdaten angezeigt, obwohl das entsprechend der Benutzerrolle nicht korrekt ist.



Durch Injektion anderer SQL-Kommandos lassen sich auch Informationen aus der Datenbank holen, die mit der reinen Applikation nichts mehr zu tun haben. Wichtig ist hier zu wissen, dass alle SQL-Kommandos mit den Datenbankberechtigungen des technischen Applikationsbesitzers ausgeführt werden. Diese sind in der Regel fast immer „über-privilegiert“.

So lassen sich mit der Injektion dieses SQL-Kommandos auch die Benutzerkennungen und Kennwörter der Applikation auslesen. Eingesetzt wird:

huhu' or 1=0 union select null,USERID,PASSWORD,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null,null from DEMO_HR_USERS—



Diese Beispiel sollte zeigen, dass der Schutz Ihrer Unternehmensdaten durch Sicherheitstechnologien wie Web Application Firewalls, Verschlüsselung und so weiter alleine nicht ausreichend ist. Eine erfolgreiche Abwehr von SQL-Injections kann nur durch den Einsatz von SQL-Firewalls gewährleistet werden.

Kosten für IT-Sicherheit

Oft wird die Frage nach den Kosten für IT-Sicherheit gestellt. Leider gibt es hierzu oft keine zufriedenstellende Antwort. Sicherlich lassen sich Kosten für Lizenzen und Wartung von Sicherheitsprodukten leicht ermitteln, jedoch ist es schwer, die Kosten für den operativen Teil der IT-Sicherheit zu bestimmen.

Hinzu kommt noch die Tatsache, dass es nahezu unmöglich ist, ein Return on Investment (ROI) für die Umsetzung von IT-Sicherheit zu berechnen. Die Frage nach der Eintrittswahrscheinlichkeit eines Angriffs sowie dessen Auswirkungen lässt sich wenn überhaupt nur statistisch ermitteln.

Ein pragmatischer Ansatz, um ein Gefühl für die ungefähren Kosten zu erhalten, ist die Annahme, dass die Kosten für IT-Sicherheit ungefähr den Kosten entsprechen, welche das Unternehmen für den Bereich Verfügbarkeit investiert. Zusätzliche Aufgaben wie Log-Auswertung, Unterstützung bei IT-Audits, fortlaufende Ausbildungsmaßnahmen für IT-Forensiker und Sicherheits-Spezialisten, ständige Anpassungen des IT-Sicherheitskonzeptes auf neue Bedrohungslagen sowie regelmäßiges Einspielen von Patches sind keine „Nebenbei“-Aktivitäten, sondern Vollzeit-Jobs und müssen zweckgebunden ins IT-Budget eingeplant werden.

Leider stieg aber die Anzahl der Cyber-Angriffe in den letzten Jahren dramatisch an. Hinzu kommt, dass die Cyber-Angriffe immer präziser und raffinierter werden. Ein guter Ausgangspunkt ist die Annahme, dass bereits potentiell gefährliche Software im Unternehmen aktiv ist. Ihre Verteidigungsstrategie sollte also dieser Annahme Rechnung tragen.

Verteidigungsstrategie

Bei der Verteidigungsstrategie ist es wichtig, nicht in blinden Aktionismus zu verfallen, sondern zielgerichtet einen Schritt nach dem andern zu machen. Viele Projekte im Bereich IT-Sicherheit scheitern an Komplexität, Missverständnissen, fehlendem Budget und Wissensmangel. Die meisten Softwarekomponenten lassen sich von der Basis her bereits sicherer betreiben als vermutet.

Der erste Schritt sollte immer sein, sich ein Lagebild zu beschaffen. Hierzu gehören Antworten auf folgende Fragen:

- Welche gesetzlichen oder branchenspezifischen Anforderungen existieren für das Unternehmen?
- Wer ist verantwortlich (Datenschutz, Revision, Risikomanagement)?
- Existiert ein Budget für IT-Sicherheit?
- Haben wir angemessen ausgebildetes Personal?

- Welche Daten betreiben wir?
- Welche Hardware wird eingesetzt?
- Welche Software ist im Betrieb?
- Welche Versionen werden verwendet?
- Welche Betriebskonzepte existieren (Sicherheit, Patchmanagement, Backup/Recovery,...)?
- Wie ist der aktuelle Sicherheitszustand?

Von den Antworten lässt sich bereits eine grobe Vorgehensweise ableiten. Wenn alle Verantwortlichen involviert sind (Datenschutz, IT-Betrieb, Risikomanagement, Revision und gegebenenfalls Personalrat), sowie IT-Budget und Personal zur Verfügung stehen, wird der zweite Schritt geplant.

Welche Art von Daten haben wir und wie müssen wir diese schützen? Nicht alle Daten müssen gleich geschützt werden. Damit Schutzmaßnahmen richtig und zielgerichtet umgesetzt werden können, ist es notwendig, einen Überblick über die Art der zu schützenden Daten zu erhalten. Das Schlüsselwort heißt Datenklassifizierung.
International werden Daten mindestens nach den Klassen Vertraulichkeit, Integrität und Verfügbarkeit eingestuft. Bekannt ist hier das englische Kürzel CIA (für die Kategorien **C**onfidentiality, **I**ntegrity, **A**vailability). Jede der Kategorien besitzt mindestens drei Abstufungen.

Beispiel zur Einstufung von Daten:

- C3 I3 A1 für Personenbezogene Daten
- C1 I2 A3 für Produktinformationen im Onlineshop
- C1 I3 A3 für Börsendaten (Live-Ticker)

Die Einstufung nimmt jedes Unternehmen individuell für sich vor. In Abhängigkeit der Einstufung der Daten sollten entsprechende Sicherheitsmaßnahmen umgesetzt werden. Wichtig ist hier, dass die Kosten für den Betrieb umso höher sind, je höher die Einstufung der Daten ist.

Nachdem bekannt ist, welche Daten welchen Schutz benötigen, muss der aktuelle Sicherheitszustand der Datenbank-Infrastruktur erfasst werden. Dieser sollte, wenn vorhanden, mit entsprechenden Tools ermittelt werden. Die Informationen hieraus helfen, einen Abgleich zwischen der Realität (aktueller Sicherheitszustand) und den gesetzlichen bzw. branchenspezifischen Anforderungen durchzuführen. Oft ist die Lücke zwischen der gefühlten Sicherheit und der geforderten Sicherheit signifikant größer als vermutet. Die hieraus ermittelten Schwachstellen werden anhand ihrer Kritikalität priorisiert.

Zudem hat es sich als vorteilhaft erwiesen, mit den technisch einfachen Maßnahmen zu beginnen und sich kleine Zwischenziele zu setzen. Ebenfalls sollten Maßnahmen, welche mit wenig Aufwand einen großen Sicherheitsmehrwert liefern, also das Risiko mit wenig Aufwand stark verringern, zuerst umgesetzt werden.

Die wesentlichen Punkte zur Erhöhung der Sicherheit einer Datenbank sind:

- Ausbildung und Sensibilisierung des Personals

  Mitarbeiter sollten über Gefahren und wie diese im Alltag bösartig ausgenutzt werden informiert sein. Wissen darüber, was Schadsoftware ist, über welche Wege sie ins Unternehmen gelangen kann und welche Schäden sie verursachen kann, dient zur ersten Sensibilisierung der Mitarbeiter. Es müssen Vorgehensweisen aufgezeigt werden, wie mit externen Quellen, besonders E-Mails mit Anhang, angemessen umgegangen werden soll. Diese zwei Schritte können bereits wertvolle erste Maßnahmen zur Verbesserung der IT-Sicherheit im Unternehmen sein. Im Prinzip einfache Verhaltensweisen der Mitarbeiter tragen wesentlich zur Sicherheit bei. Das Unterlassen des Öffnens von Mails vermeintlich vertrauenswürdiger Quellen wie Freunde und Bekannte oder das Vermeiden der Verwendung von ein und demselben USB Stick für zu Hause und im Büro ist essenziell und bietet den größten Schutz vor Cyber-Attacken. Hier kann das Unternehmen durch regelmäßige Trainings (wie zum Beispiel durch Computer-Based Training) die Sensibilität der Mitarbeiter steigern. Diese Sensibilisierung hilft nicht nur dem Unternehmen, sondern auch den Mitarbeitern im privaten Umfeld sich vor Cyber-Attacken zu schützen.

  Die Umsetzung und Wirksamkeit von technischen und organisatorischen Maßnahmen zur Steigerung der IT-Sicherheit ist stark davon abhängig, wie gut die IT-Mitarbeiter ausgebildet sind, wie deren eigene Einschätzung der potentiellen Bedrohungslage ist und ob ein ausreichendes Budget vorhanden ist. Zudem werden aus Kosten- und Zeitgründen oft notwendige Ausbildungsmaßnahmen der IT-Mitarbeiter verschoben oder sogar unterlassen. Dies führt zu Wissenslücken insbesondere in den Bereichen moderner Sicherheitsmaßnahmen und Sicherheitstechnologien. Durch falsche beziehungsweise mangelhafte Handhabung von Technologien entsteht mitunter der trügerische Eindruck von Sicherheit. Ausserdem lässt die hohe Arbeitsbelastung der IT-Mitarbeiter durch die Übernahme zusätzlicher Aufgaben aus dem Bereich IT-Sicherheit meist keinen optimalen Schutz zu. Hier muss das Unternehmen durch Implementierung standardisierter Prozesse, automatisierter Arbeitsabläufe und Auslagern von Routinearbeiten für Entlastung der IT-Mitarbeiter sorgen. Geschieht dies nicht, bleibt die IT-Sicherheit zwar eine der höchsten Prioritäten im Unternehmen, aber nicht in der Praxis, sondern nur auf dem Papier.

- Härtung des Betriebssystems

  Ist das Betriebssystem unsicher, sind alle darauf betriebenen Anwendungen und Verfahren ebenfalls unsicher. Unsicher sind dann auch Systeme, die sich im gleichen Netzwerk-Segment befinden, wie das unsichere Betriebssystem. Grundsätzlich ähneln die Maßnahmen zur Härtung eines Betriebssystems den Maßnahmen zur Härtung einer Datenbank:
  Restriktive Rechtevergabe, Deaktivierung und Deinstallation unsicherer Protokolle und Software, regelmäßige Softwareaktualisierung und ein zielgerichtetes Auditing.

Das Unternehmen sollte klare Richtlinien zur Härtung des Betriebssystems definieren und nachhaltig umsetzen und überwachen.

Solaris Security Guide
http://docs.oracle.com/cd/E53394_01/html/E54807/index.html

Linux Security Guide
https://docs.oracle.com/cd/E52668_01/E54670/html/index.html

Oracle Cloud Control und Lifecycle Management
http://www.oracle.com/technetwork/oem/lifecycle-mgmt/config-learn-more-2687517.html

- Sichere Datenbank-Konfiguration (ggf. versionsabhängig)
  Das Härten von Datenbanken verfolgt zwei wesentliche Ziele: Risikominimierung und Nachweisbarkeit. Es sollte überprüft werden, ob Standardfunktionen bzw. notwendige Zusatzfunktionen entsprechend den Anforderungen sicher eingestellt sind. Überprüft werden sollten z. B. der Verzicht auf die Nutzung von Standardkennwörtern, die Zurücknahme von nicht notwendigen Privilegien, die Kontrolle des Zugriffs auf sensible Daten, die Datenverschlüsselung etc.
  Zudem sollte dass zugrundeliegende Betriebssystem gehärtet sein.

  Oracle Database Security Guide
  https://docs.oracle.com/database/122/DBSEG/toc.htm

- Automatisierte Überwachung hoch privilegierter Aktivitäten (Auditing)
  Nicht alle Datenbank-Aktivitäten werden über das Netzwerk ausgeführt. Die meisten hoch privilegierten Aktivitäten (> 80%) werden lokal auf den Datenbankservern durchgeführt. Dazu gehören Aktivitäten wie Jobs und Batchläufe, aber auch das lokale Ausführen von Datenbank-Patches oder Applikations-Patches. Um alle Aktivitäten lückenlos zu erfassen, sollte das datenbankeigene Auditsystem genutzt werden. Die Oracle Datenbank ermöglicht das Auditing/Logging von Benutzern- und DBAs in der Datenbank. Die Auditing-Funktionalitäten können dabei hochflexibel an die unterschiedlichen Unternehmensanforderungen angepasst werden. Auch wenn der Begriff Auditing einen „faden“ Beigeschmack für so manchen Datenschützer hat, sollte nie außer Acht gelassen werden, dass Sicherheit nicht unbedingt Anonymität bedeutet! Aufgrund der Anpassungsmöglichkeiten des Oracle Auditings und seiner Auswertung kann jedoch immer ein Kompromiss zwischen Sicherheitsanforderung und Datenschutz gefunden werden.

  Oracle Datenbank Audit Dokumentation
  https://docs.oracle.com/database/122/DBSEG/part_6.htm#DBSEG006

Die erzeugten Protokolldaten müssen umgehend auf eine separate Umgebung übertragen und vom erzeugenden Produktivsystem gelöscht werden.

Oracle stellt mit dem Produkt „Audit Vault and Database Firewall“ (AVDF) eine Lösung zur Verfügung, welche diese Aufgaben vollständig automatisiert übernehmen kann. Der Oracle Audit Vault Server ist ein dedizierter Server und ist für die sichere Speicherung und Auswertung der Audit Daten verantwortlich.
Die so zentral gesammelten Protokolldaten stehen annähernd in Echtzeit zur Verfügung. Dies ermöglicht eine zeitnahe Analyse zum Beispiel bei einem Sicherheitsvorfall. Die Protokolldaten sind nicht manipulierbar und stehen nur autorisierten Benutzern zur zweckgebundenen Auswertung zur Verfügung. Ein entsprechendes Zugriffs- und Berechtigungskonzept ist implementiert.
Aus Datenschutzgründen, dürfen Protokolldaten mit personenbezogenen Inhalten nur auf bestimmte Zeit vorgehalten werden. Demzufolge ist es möglich, entsprechende Löschfristen automatisiert einzuhalten. Berichte können zeitgesteuert an bestimmte Benutzerkreise zum Beispiel in Form von PDFs verteilt werden und unterliegen ebenfalls den vorgegebenen Löschfristen. Spezielle Datenbank-Ereignisse können bei Bedarf an ein angeschlossenes SIEM Tool zur weiteren Analyse weitergeleitet werden. Dies ermöglicht eine genaue Analyse der Vorgänge bei einem Sicherheitsvorfall.

Audit Vault and Database Firewall
http://www.oracle.com/us/products/database/security/ds-security-audit-vault-firewall-1883353.pdf

- Kontinuierliche Log-Überwachung
  Logdateien enthalten viele wertvolle Informationen. Die überwiegend technischen Informationen helfen bei der Erkennung und Eingrenzung von Problemen. Sie sind aber auch unverzichtbar für die forensische Analyse nach und während eines Sicherheitsvorfalls. Leider werden die Logdateien häufig zu selten ausgewertet und dann unausgewertet gemäß des entsprechend gültigem Datenschutzgesetzes fristgerecht gelöscht. Viele Sicherheitsvorkommnisse bleiben aber unentdeckt oder werden erst lange nach dem Vorkommnis bekannt. Dabei muss es in den meisten Fällen nicht so weit kommen. Entscheidend ist hier zu wissen, welche Log-Dateien mit welchen Informationen vorhanden sind und wonach in diesen gesucht werden soll. Einen Standard, wie Log-Dateien auszusehen haben, gibt es leider nicht. Das erschwert die Analyse und Auswertung. Die meisten Hersteller von IT-Komponenten liefern entsprechende Informationen über die Log-Dateien aber in ihrer Dokumentation. Die Dokumentation beinhaltet meist Informationen über den Speicherort, über Einstellungsmöglichkeiten des Auditing sowie über den Inhalt und die Struktur der Log-Dateien. In vielen Fällen werden Log-Dateien im XML Format gespeichert.

Siehe Diagnostik-Struktur der Oracle Datenbank:
https://docs.oracle.com/database/122/CNCPT/oracle-database-instance.htm#CNCPT1912

Die Analyse erfolgt meist klassisch über das Parsen mittels regulärer Ausdrücke. Diese Vorgehensweise stößt allerdings schnell an ihre Grenzen. Abhilfe schafft hier die modernere Big Data Technologie. Big Data Technologien sind geradezu prädestiniert dafür, Analysen auf eine sehr große Anzahl von unstrukturierten Daten (Log-Dateien) durchzuführen.

Oracle LOG-Analytics
http://www.oracle.com/us/solutions/cloud/omc-log-analytics-cloud-service-ds-2714886.pdf

- Toolbasierte Schwachstellenerkennung
  In der Regel entwerfen Unternehmen einen Sicherheitsstandard und setzen diesen manuell um. Die Erfahrung zeigt aber eindeutig, dass entsprechende Sicherheitsverantwortliche, wie der CSO oder Datenschutzbeauftragte, keine Transparenz und Kontrolle über die Durchsetzung der definierten Sicherheitsstandards im Unternehmen haben. Daraus folgt, dass im Allgemeinen eine deutliche Diskrepanz besteht zwischen der geforderten und der tatsächlichen Umsetzung von Sicherheitsmassnahmen.
  Diese Lücke kann unter anderem toolbasiert geschlossen werden. Gerade in Bezug auf Datenbank-Härtung beziehungsweise „sichere Konfigurationen im Allgemeinen“ bietet Oracle eine umfangreiche „Regelbiobliothek“ an, die hier unterstützen kann und zum Beispiel die Datenbank auf aktuelle Schwachstellen hin prüft. Die Bibliothek wird regelmäßig durch Oracle hinsichtlich neuer Schwachstellen aktualisiert. Diese Funktionalität verbessert die Kontrolle und Transparenz bei der Durchsetzung unternehmensweiter Sicherheitsstandards nachhaltig.

  Oracle Cloud Control und Lifecycle Management
  http://www.oracle.com/technetwork/oem/lifecycle-mgmt/config-learn-more-2687517.html

- Aktuelle Zustandsberichte
  Wer seine aktuelle IT-Landschaft nicht kennt, kann diese auch nicht effektiv schützen. Aktuelle Informationen darüber, welche Softwarekomponenten im Einsatz sind, welche Versionen diese haben, wie sie aktuell konfiguriert sind und wie deren Historie aussah, sind nicht nur für ein anstehendes IT-Audit essenziell, sondern sind auch unerlässlich bei der Einschätzung des aktuellen Risikos und der aktuellen Bedrohungslage.
  Existieren für diese IT-Komponenten Sicherheits-Updates? Entspricht der aktuelle Zustand meiner IT-Infrastruktur meinen Sicherheitsrichtlinien?
  Diese Fragen lassen sich nur zeitnah beantworten, wenn ein entsprechendes Software Inventarisierungssystem beziehungsweise ein Configurations und Change Management System im Einsatz ist. Diese Werkzeuge arbeiten meist vollkommen automatisiert und liefern aktuelle Zustandsberichte sowie historische Informationen und Berichte über Konfigurationsänderungen.

Oracle Cloud Control und Lifecycle Management
http://www.oracle.com/technetwork/oem/lifecycle-mgmt/config-learn-more-2687517.html

- Überwachung des Normal-Verhaltens von Datenbankzugriffen
  Im Wesentlichen geht es hier um die Abwehr von SQL-Injections (siehe Beispiel im Dokument). Die Datenbank an sich ist nicht in der Lage, die für eine Applikation normalen SQL-Statements von untypischen SQL-Statements zu unterscheiden. Die einzigen Maßnahmen, welche hier Risikominderungen versprechen, sind zum einen eine gewissenhafte Anwendungsentwicklung nach aktuellen Sicherheitsstandards und zum andern der Einsatz einer sogenannten SQL-Firewall. Eine SQL-Firewall ist im Gegensatz zu einer „normalen Firewall“ eine spezielle Applikation-Firewall, welche Datenbank SQL-Statements versteht.
  Die Realität zeigt, dass leider meist aus Kosten und Zeitgründen das Thema Sicherheit bei der Entwicklung von Anwendungen und Verfahren auf das notwendigste reduziert wird. Das hierdurch latent vorhandene Risiko, durch eine erfolgreiche SQL-Injection den Verlust der Integrität, Vertraulichkeit und Verfügbarkeit der Daten zu erleiden, ist relativ hoch.
  SQL-Firewalls sind hier eine sinnvolle Maßnahme zur Abwehr von SQL-Injections. Moderne SQL-Firewalls sind in der Lage, das Normalverhalten einer Applikation zu erlernen. Es entsteht somit eine sogenannte „White List“ je Anwendung, in der alle „normalen“ SQL-Statements der entsprechenden Anwendung in SQL-Syntax Clustern gespeichert sind. Ist die SQL-Firewall erst einmal angelernt, werden alle SQL-Statements, welche die Anwendung zur Datenbank sendet, in Echtzeit analysiert und mit der erlernten White List verglichen. Sollte nun ein aktuelles SQL-Statement nicht auf der White List sein, gilt dieses als verdächtig und kann entweder geblockt oder durch ein ungefährliches SQL-Statement ersetzt werden. Auf jeden Fall wird dieser Vorfall protokolliert. Somit stellt die SQL-Firewall eine Maßnahme dar, die wesentlich zur Risikominderung in Bezug auf SQL-Injections beiträgt.

  Audit Vault and Database Firewall
  http://www.oracle.com/us/products/database/security/ds-security-audit-vault-firewall-1883353.pdf

Weitere Informationen

Oracle Database 12*cR2* Security Guide

https://docs.oracle.com/database/122/nav/security.htm

Oracle Database 11*g* Security Guide

http://docs.oracle.com/cd/E11882_01/server.112/e10575/toc.htm

Oracle Database Security

http://www.oracle.com/us/products/database/security/overview/index.html

Securing Oracle Database 12*c* eBook: A Technical Primer

http://books.mcgraw-hill.com/ebookdownloads/Oracle12cSecurity

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Datenbank-Sicherheit - Grundüberlegung
August 2017
Author: Norman Sibbing