Einführung Oracle Forensik, Tamper Detection, Log Analyse

Dr. Günter Unbescheid

Database Consult GmbH

Jachenau

Schlüsselworte

Datenbank, Sicherheit, Forensik, Log-Analyse

Einleitung

Die Veröffentlichungen der letzten Jahre haben den Stellenwert der IT-Sicherheit beträchtlich erhöht. Unternehmen sind mehr denn je bereit, in dieses wichtige Thema zu investieren. Die Oracle-Datenbank bietet bekanntlich neben umfangreichen Security-Features vielfältige Möglichkeiten des Logging, Tracing und Auditing. Doch die Konfiguration und Aktivierung dieser Features alleine genügt in den meisten Fällen keineswegs, sondern muss ergänzt werden durch eine sinnvolle, d.h. an die Sicherheitsanforderungen und technischen Voraussetzungen des Kunden angepasste, forensische Analyse der vorliegenden Daten.

Der Vortrag gibt eine Einführung in die technischen Möglichkeiten und die konzeptionellen Herausforderungen dieses Themas.

Das Umfeld

Es gibt im Netz zahllose Definitionen des Begriffes Computer-Forensik oder IT-Forensik. Eine liefert auch das Bundesamt für Sicherheit in der Informationstechnik (BSI), wonach Forensik

... die streng methodisch vorgenommene Datenanalyse auf Datenträgern und in Computernetzen zur Aufklärung von Vorfällen unter Einbeziehung der Möglichkeiten der strategischen Vorbereitung insbesondere aus der Sicht des Anlagenbetreibers eines IT-Systems (ist).

Diese ausladende Definition lässt sich lapidar und knapp als *Datenanalyse zur Aufklärung von Vorfällen* zusammenfassen. Die im Rahmen der Forensik betriebene Aufklärung kann – im schlimmsten Falle – Aktionen aus dem Bereich der Computerkriminalität betreffen, die dann nicht nur analysiert, sondern auch zur nachträglichen Verwendung vor Gericht und im Einklang mit der aktuellen Gesetzesgrundlage gesichert werden müssen. Im einfacheren Falle lassen sich die Erkenntnisse genauso gut für Zivilverfahren oder zur Bearbeitung von Supportfällen einsetzen.

In vollem Umfang betrieben sind forensische Untersuchungen sehr komplex und umfassen diverse Bereiche des Hardware- und Software-Stacks kompletter IT-Infrastrukturen. Grundsätzlich lassen sich zwei Bereiche unterscheiden:

- Bei der Post-mortem-Analyse (auch Offline-Forensik) werden nicht-flüchtige Spuren auf Datenträgern bzw. Datenträgerabbildern untersucht, wohingegen
- bei der Live-Forensik (auch Online-Forensik) das Augenmerk auf flüchtige Daten wie beispielsweise Hauptspeicherinhalte, Prozesslisten und Netzwerkverbindungen gelegt wird. Hier ist minimal invasives und rasches Vorgehen gefordert, um vorhandene Spuren nicht unnötig zu verwischen.

Neben dem Nachweis und der Erkennung von Manipulationen ist die Forensik im Rahmen eines Notfallmanagements naturgemäß auch an der Schadensbegrenzung, dem Wiederanlauf (Recovery) und der Wiederherstellung (Restauration) betroffener Systeme interessiert.

Das Ziel einer forensischen Untersuchung ist in der Regel die Beantwortung folgender Fragen:

- Was ist geschehen?
- Wo ist es passiert?
- Wann ist es passiert?
- Wie ist es passiert?

Zusätzlich können zusätzliche Fragestellungen relevant werden, insbesondere auch im Fall der Strafverfolgung oder einer Sicherheitsbewertung :

- Wer hat es getan?
- Was kann gegen eine Wiederholung getan werden?

Vor diesem Hintergrund ist es recht einfach, den Begriff Datenbank Forensik im Allgemeinen und Oracle Forensik im Besonderen einzugrenzen:

Database forensics is a branch of digital forensic science relating to the forensic study of databases and their related metadata (Wiki).

Die Oracle Forensik baut demnach auf denselben Grundlagen wie die allgemeine Forensik auf und konzentriert sich lediglich auf das unmittelbare Umfeld von Oracle-Systemen.

Vor diesem Hintergrund ist es wichtig zu betonen, dass die Oracle-Forensik nur als ein strategisches Element in einem sicherheitstechnischen Gesamtkonzept zu betrachten ist. Da Datenbanken stets in einen größeren infrastrukturellen Kontext eingebunden sind, müssen auch angrenzende Komponenten wie beispielsweise Firewalls, Router und Webserver für umfassende Lösungen mit in Betracht gezogen werden. Sicherheit erreicht man bekanntlich nicht durch den Einsatz einzelner Features.

Methoden und Datenquellen

Der vorliegende Beitrag möchte vor diesem Hintergrund eine Einführung in den forensischen Teilbereich der Oracle Forensik bieten. Bei forensischen Recherchen sind, wie bei allen anderen Sicherheitsvorfällen auch, die altvertrauten Sicherheitsaspekte der Vertraulichkeit, Integrität, Verfügbarkeit, Nichtabstreitbarkeit und Authentizität gefordert. Besonders zur Erfüllung der beiden letztgenannten Aspekte sollte jedes System zur Sicherstellung effektiver und gesetztes-konformer Analysen bestimmte „strategische Voraussetzungen“ erfüllen. Die damit verbundenen konfiguratorischen und administrativen Vorarbeiten sind letztlich identisch mit den in jedem guten Security-Projekt geforderten Maßnahmen:

- Nutzung eindeutiger, persönlicher Benutzer-Accounts oder vergleichbarer Techniken, zumindest auf der Ebene des Betriebssystems, optional auch innerhalb der Datenbank. Nur auf diese Weise kann die Forderung nach „Nichtabstreitbarkeit“ gewährleistet werden.
- Auditing und Logging sicherheitsrelevanter Vorgänge auf OS- und DB-Ebene unter Wahrung der Revisionssicherheit und Auswertbarkeit. Audit Optionen sind für Administratoren (OS- und DB-Umfeld) und Applikations-Benutzer (in der Regel nur DB-Umfeld) in Abhängigkeit von den Schutzanforderungen der Daten zu aktivieren.
- Alle Logging Maßnahmen müssen firmenintern sorgfältig abgestimmt werden (Betriebsrat etc.).
- Log- und Audit-Daten sollten zur Sicherstellung der Revisionssicherheit und aus Gründen der Auswertbarkeit auf einem oder mehreren zentralen Log-Servern zusammengeführt werden. Diese Log-Server sind „gehärtet“ zu konfigurieren. Der Zugang ist auf das absolute Minimum zu beschränken (*segregation of duties*)
- Auf diesen Logservern müssen die Daten unterschiedlicher Quellen für spätere Analysen vorbereitet werden (effiziente Speicherstrukturen, Vorsortierung, Anreicherung mit zusätzlichen Informationen).
- Die Verwaltung und Analyse der Log-Daten sollte nach Möglichkeit in strikter personeller Trennung zu den Systemadministratoren stattfinden. Die Nutzung entsprechender Frameworks – selbst programmiert oder kommerziell – ist dringend anzuraten.
- Persönliche Informationen sind in der Regel maskiert und nur in Sonderfällen direkt auszugeben.
- Zur Erstellung präziser „*timelines*“ (zeitliche Zusammenschau von Events aus unterschiedlichen Log-Quellen) ist eine präzise Synchronisation der Systemzeiten unabdingbar.

Es kann nicht oft genug betont werden, dass die vorstehend skizzierten Maßnahmen mit Augenmaß implementiert werden müssen, so dass die Effizienz administrativer Prozesse ebenso berücksichtigt wird wie die Belange der Sicherheit und gesetzliche Rahmen.

Diese Konfigurationsbereiche sind darüber hinaus eingebettet in einen „forensischen Stack“ der fortschreitend aus den folgenden Stufen bestehen sollte:

- Durchführung einer Schutzbedarfsanalyse zur Feststellung sensibler Daten, darauf aufbauend
- die Einführung von Systemklassen und mit ihnen verbundenen Schutzmaßnahmen
- für die höchste Schutzklasse sollten Systeminventare festhalten welche Applikationen, von welchen Servern aus mit welchen Benutzern welche Daten verarbeiten. Auf dieser Basis lassen sich „ungewöhnliche“ Aktivitäten leichter ausmachen.
- Ebenfalls für die höchste Schutzklasse werden dann die festgelegten konfiguratorischen Maßnahmen durchgeführt.
- Die Einstellungen müssen permanent geprüft werden (Monitoring). Ebenso müssen Audit- und Log-Daten ausgewertet werden (Reporting)
- Auf dieser Basis können dann auffällige Maßnahmen lokalisiert und bewertet werden.

Grundsätzlich stehen im Umfeld der Forensik sogenannte „Security Information and Event Management“ (SIEM) und „Intrusion Detection“ (IDS) Systeme zur Verfügung, die Log-Daten und Netzverkehr analysieren und – im Falle von SIEM – auch für das Security Event Management zuständig sein können. Die Konfiguration und der Betrieb dieser Systeme wird allgemein als sehr komplex und aufwändig empfunden und ist weit davon entfernt, als „rundum-glücklich-Lösung“ schlecht oder gar nicht implementierte Security Policies zu kompensieren.

In den folgenden Abschnitten werden Details für die wichtigsten Konfigurationsbereiche besprochen.

Basis für die Nachweisbarkeit und Unbestreitbarkeit (*nonrepudiation*) von administrativen Aktionen ist eine sichere Authentifizierung auf der Grundlage von persönlichen Benutzern (und technischen Benutzern für Batch-Abläufe). Für Administratoren, die sich sowohl in der Datenbank als auch im Umfeld des Betriebssystems bewegen, ist dies auf OS-Ebene ein Muss. Aktionen innerhalb der Datenbank können dann – auch wenn per SYSDBA gearbeitet wird – über die OS-Accounts zugeordnet werden. Endbenutzer verbinden sich in der Regel über Applikationen und benötigen keine persönlichen OS-Accounts auf den Zielsystemen. Nur auf der Basis spezifischer Authentifizierungen lassen sich die *need-to-know*- und *least-privilege*-Forderungen umsetzen

Auditing sollte für Administratoren sowohl im OS- als auch im DB-Kontext konfiguriert werden. Hier ist sorgfältig auf eine gute Balance zwischen den Sicherheitsanforderungen und dem Datenaufkommen zu achten. Schließlich müssen Audit-Daten regelmäßig analysiert werden, um Auffälligkeiten zu entdecken. Die Abstimmung der Audit-Maßnahmen mit dem Betriebsrat ist unbedingt notwendig. Linux bietet in diesem Zusammenhang unterschiedliche Möglichkeiten und spezielle Shells, die teilweise nur den Input, teilweise Input und Output inklusive eingegebener Passwörter mitschreiben. Hier ist Vorsicht geboten. Eine gute Balance bietet das beispielsweise das Logging von SUDO- und SSH-Aktionen, die revisionssicher in das Systemlog geschrieben werden können und vorzugsweise auch zentral zusammengeführt werden sollten. Für das Datenbank-Auditing ist ebenfalls auf Revisionssicherheit zu achten. Leider ist für das unter 12c eingeführte sogenannte „Unified Auditing“ nicht über SYSLOG-NG oder RSYSLOG zu konfigurieren. Es versteht sich, dass die Konfiguration von Auditing ganz eng an die im Unternehmen üblichen und gelebten Zugriffsmuster angelehnt sein muss.

Es ist unabdingbar, die durchgeführten konfiguratorischen Einstellungen permanent zu überwachen, d.h. in das System Monitoring einzubinden. Darüber hinaus: Auditing ohne Auswertung der Daten ist wertlos. Werden die Daten auf einem oder mehreren zentralen Log-Servern zusammengeführt, können übergreifende Analysen durchgeführt werden. Für die Aufbereitung der Daten auf den Log-Servern stehen unterschiedliche Strategien zur Verfügung:

- Aufbereitung im Rahmen eines DB-Repositories/Data Warehouses. Hier lassen sich unterschiedliche „Audit-Marts“ definieren. SQL bietet darüber hinaus flexible Abfragemöglichkeiten inklusive der Maskierung persönlicher Daten für Unbefugte.
- Speicherung im Kontext des Filesystems. Die Verzeichnisstrukturen sind sorgfältig zu planen. Abfragen, Datenanalysen und –Maskierungen sind über entsprechende Frameworks zu realisieren.

Für übergreifende Analysen müssen die Daten unterschiedlicher Quellen zeitsynchron zusammengeführt werden (Timelines). Die Abstimmung der Systemzeiten, die ja auch innerhalb der Datenbank abgegriffen werden, ist hier unerlässlich und muss spätestens beim Übertrag auf den Logserver erfolgen.

Die Audit-Daten des Betriebssystems und der Datenbank sollten bei forensischen Untersuchen durch weitere Datenbank-Quellen ergänzt werden. In Fällen, wo das Auditing nicht aktiviert wurde, bieten diese Quellen oft die einzige Möglichkeit für den „Blick in die Vergangenheit“. Die vielfältigen und in Teilen redundanten Datenquellen lassen die Wahrscheinlichkeit steigen, dass auch bei möglicher Vertuschung subversiver Aktionen, verdeckte Spuren nicht vollständig getilgt und damit sichtbar gemacht werden können:

- Listener.log – liefert u.a. Zeitstempel, Connect Strings, Protokoll Informationen, benutzte Servicenamen, Hostname, IP-Adressen und Return Codes

- login.sql/glogin.sql – diese Dateien, die beim Aufruf von sqlplus automatisch ausgeführt werden, können Schadcode enthalten und sollten überwacht werden.
- Trace- und Alert-Files – zeigen administrative Aktionen und Fehlersituationen an.
- Redo Log Files – bieten im Zusammenhang mit dem Logminer-Werkzeug die Möglichkeit, Transaktionen zurückzuverfolgen, sofern in der Datenbank das *supplemental logging* eingeschaltet wurde. Die in den Logs verzeichneten Zeiten entsprechen jedoch nicht den tatsächlichen Ausführungszeiten, sondern den Zeiten des *log syncs*, d.h. des Schreibens der Redo Records aus dem Log-Buffer. Darüber hinaus werden diese Zeiten nur Sekundengenau ausgegeben (Spalte timestamp von *v$logmnr_contents*). Dies ist bei der Erstellung von Timelines zu berücksichtigen.
- Flashback Queries – erlauben bekanntermaßen den Blick zurück, sofern die eingestellten Retention-Zeiten nicht überschritten wurden. In diesem Zusammenhang können sie vor Allem interessant werden, um Metadaten, beispielsweise Tabellendefinitionen zu prüfen.
- recyclebin – Der Papierkorb kann – falls er nicht gellert wurde – ebenfalls Hinweise auf gelöschte Objekte enthalten.
- V$ / AWR / ASH – der Blick in die Aktivitäten von Sessions und die in ihrem Kontext aufgeführten Statements ist über die Objekte von AWR und ASH sowie diverser V$-Views wie beispielsweise VSQL, VSQL_BIND_DATA, V$SQL_PLAN und weiterer Views möglich. Bei den Abfragen sollten nach Möglichkeit Basistabellen (z. B. WRH$_ACTIVE_SESSION_HISTORY) statt der übergeordneten Views verwendet werden, um zu verhindern, dass kontaminierte Objekte den Analysten in die Irre führen.
- Informationen über Datenzugriffe lassen sich auch über Objekte wie col_usage$ (Tabellenspalten und angewandte Prädikate) und mon_mod$ (DML-Statistiken von Segmenten) ermitteln.
- Datenblöcke – enthalten ebenfalls, zumindest für eine gewisse Zeitspanne, Informationen über gelöschte Datensätze, die über Blockdumps ermittelt werden können. Diese Methode ist hochgradig aufwändig und kann nur in Ausnahmefällen und begrenzt zum Einsatz kommen.

Die Durchschlagskraft der Untersuchung ergibt sich aus der Kombination aller verfügbaren Quellen. Dabei sollte zunächst der flüchtige Bereich (V$-Objekte) und dann der permanente Bereich revisionssicher erfasst und auf ein separates Analysesystem übertragen werden.

Wichtige konfiguratorische Metadaten wie beispielsweise die Konfiguration von Rollen und Privilegien, der Code von DDL- oder Logon-Triggern oder zentralen Packages sollte ebenfalls erfasst und kontrolliert werden, um auf Manipulationen frühzeitig reagieren zu können. Hierfür können Checksums (von Result-Dateien) oder Hash-Werte genutzt werden.
Über DBMS_UTILITY.GET_HASH_VALUE oder DBMS_SQLHASH.GETHASH können Hash-Werte von Resultsets bzw. Cursorn generiert werden.

Zusammenfassung

Forensische Analysen im allgemeinen sind sehr komplex und müssen ständig an neue Bedrohungsszenarien und aktuelle Software-Versionen angepasst werden. Dies gilt auch für Analysen im Bereich von Oracle-Datenbanken, die stets nur als Teil gesamthafter Analysen begriffen werden können. Hierbei ist eine gute Basiskonfiguration der betroffenen Systemen ebenso wichtig, wie eine gute Detailkenntnis der technischen Zusammenhänge und Angriffsszenarien. Ersteres können und müssen die Betroffenen selbstständig vor Ort lösen, letzteres sollte mit zusätzlicher externer Unterstützung in Angriff genommen werden. Einmal vorbereitet ist das permanente Monitoring und regelmäßige Auswertungen der Log-Daten unerlässlich, um Vorfälle hinsichtlich ihres Bedrohungscharakters bewerten zu können.

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