Oracle-Datenbank-Security & PCI-DSS-Zertifizierung

Johannes Kraus, Herrmann & Lenz Services GmbH

Die IT-Sicherheit und im Speziellen die Datenbank-Sicherheit wurden in der Vergangenheit in vielen Fällen recht stiefmütterlich behandelt. Spätestens seit den NSA-Enthüllungen im Sommer 2013 bekam dieses Thema jedoch immer mehr Aufmerksamkeit und ist inzwischen ein ständiger Begleiter einer jeden IT-Lösung. Für Firmen, die mit Kreditkarten in Berührung kommen, sei es bei der Abwicklung von Transaktionen, der Übermittlung von Kreditkarten-Daten oder der Speicherung dieser Daten, ist das Thema „Sicherheit" und speziell die PCI-DSS-Zertifizierung kein Neuland mehr.

Die Abkürzung „PCI-DSS" steht für „Payment Card Industry Data Security Standard". Hierbei handelt es sich, einfach ausgedrückt, um einen Datensicherheits-Standard der Kreditkarten-Industrie, der im Jahr 2006 von American Express, Discover Financial Services, JCB International, MasterCard und VISA Inc. ins Leben gerufen wurde. Er soll die Kreditkarten-Daten der Endkunden sowohl vor Diebstahl als auch vor Missbrauch schützen. Zertifiziert werden müssen alle Systeme, die mit Kreditkarten-Daten in Berührung kommen. Dazu zählt neben den üblichen Kreditkarten-Daten wie der Kreditkarten-

Nummer auch die dreistellige Kreditkarten-Prüfnummer.

Kommen firmeneigene Systeme zu keiner Zeit mit Kreditkarten-Daten in Berührung, ist auch keine Zertifizierung notwendig. Im Falle einer Überschneidung, auch wenn es sich beispielsweise nur um die Weiterleitung der Daten an einen Zahlungs-Dienstleister handelt, muss eine Zertifizierung vorliegen. Bei Missachtung werden hohe Strafzahlungen fällig bis hin zum vollständigen Entzug der Erlaubnis, Kreditkarten-Transaktionen verarbeiten zu dürfen.

Die Firma Herrmann & Lenz Services GmbH hat in der Vergangenheit bereits mehrfach erfolgreich Kunden bei der PCI-DSS-Zertifizierung begleitet und dabei unter anderem den Bereich „Datenbanken" übernommen, der in diesem Artikel näher beleuchtet wird. Dabei werden verschiedene Einstellungsmöglichkeiten und Lösungen zur Erhöhung der Sicherheit gezeigt, die in fast jeder Edition (SE, SE One, SE Two, EE) eingesetzt werden können.

Oracle Binaries

Die Sicherheit einer Datenbank beginnt bereits mit einer erfolgreich abgeschlossenen Oracle-Server-Installation. Im „bin"-Verzeichnis der Installation („ORACLE_HOME") liegen die Executables einer jeden Datenbank. Dort ist auch das Binary „oracle" (unter Unix/Linux) beziehungsweise „oracle.exe" (unter Windows) zu finden. Unter Linux/Unix besitzt dieses Binary die Berechtigungen „-rwsr-s--x oracle".

Auf den ersten Blick ist zu erkennen, dass der Eigentümer der Datei alle Rechte auf diese Datei hat (lesend, schreibend und ausführend). Das „s" bedeutet dabei, dass die Datei beziehungsweise das Programm zusätzlich zu den Rechten des Users, unter dem das Programm gerade läuft, mit den Rechten des Eigentümers läuft.

Bei den Angaben zu den Gruppenberechtigungen wird sichtbar, dass die Datei ein „r" für lesend und ein „x" beziehungsweise ein „s" für ausführend vorweist. Das „s" steht in diesem Fall dafür, dass das Programm zusätzlich zu den Rechten des ausführenden Users mit den Berechtigungen der Gruppe läuft. Alle anderen Benutzer, ausgenommen der Eigentümer und die Benutzer der gleichen Gruppe, dürfen das Programm nur ausführen.

Um nun das eigentliche Problem zu verstehen, ist es gut zu wissen, wie die Datenbank die Verarbeitung von Server-Verbindungen aufbaut und verwaltet.

Wird eine Verbindung zur Datenbank ohne den Listener aufgebaut, erstellt die Datenbank einen User-Prozess und verwaltet diesen (Stichwort: „bequeathing" oder „beq") über einen weiteren Server-Prozess, dessen Child-Prozess-ID mit der Parent-Prozess-ID des Anwendungsprozesses, etwa „sqlplus", übereinstimmt. Wird also nach dem Programm „sqlplus" auf dem Datenbankserver gesucht *(siehe Listing 1)*, erhält man in diesem Beispiel die Parent-Prozess-ID 26138. Mit deren Hilfe können nun weitere involvierte Prozesse (der Datenbankserver-Prozess und der „sqlplus"-Userprozess) gefunden werden *(siehe Listing 2)*.

Wird nun die Verbindung über den Listener aufgebaut, erstellt dieser einen Prozess für den User (Stichwort: „forking") und baut dabei die Verbindung mit der Datenbank auf. Dabei spielt es keine Rolle, ob die Verbindung lokal vom Datenbank-Server oder von einem Client aus über den Listener erstellt wurde. Eine Suche, wie sie oben beschrieben wurde, führt in diesem Fall zu keinem Ergebnis. Es kann zwar der Applikationsprozess gefunden werden, in diesem Fall „sqlplus", jedoch nicht der passende Server-Prozess. Aus diesem Grund müssen wir die Datenbank mit einem SQL-Befehl zu Rate ziehen *(siehe Listing 3)*. Mithilfe der gefundenen Prozess-ID kann nun auf dem Datenbank-Server danach gesucht werden *(siehe Listing 4)*.

Aus den eben erwähnten Informationen lässt sich ableiten, dass die Datenbank den Server-Prozess startet und verwaltet – und nicht der User selbst. Der Server-Prozess verwendet dabei das Binary „oracle" oder „oracle.exe". Die Erkenntnis daraus ist, dass die zuvor beschriebenen Dateiberechtigungen nicht nötigt sind und mit „chmod 0700 $ORACLE_HOME/bin/oracle" entfernt werden können.

Bei diesem Befehl wird auch das SUID-Bit „s" entfernt und durch ein „x" ersetzt. Dies bedeutet jedoch, dass keine „bequeathing"-Verbindungen mehr zur Datenbank von anderen Usern, ausgenommen ist der Eigentümer des Binary, geöffnet werden können. Verbindungen über den Listener sind jedoch nach wie vor möglich.

Neben dem Binary „oracle" oder „oracle.exe" gibt es noch weitere Executables, die in Bezug auf ihre Rechte angepasst werden sollten. Es ist im Rahmen des Artikels allerdings nicht möglich, auf weitere Beispiele einzugehen.

```
ps –ef | grep sqlplus
26138 22094 sqlplus jkadmin
```

Listing 1

```
ps -ef | grep 26138
26139 26138 oracleORCL (DESCRIPTION=(LOCAL=YES)
(ADDRESS=(PROTOCOL=beq)))
26138 22094 sqlplus jkadmin
```

Listing 2

```
select spid
from v$session s, v$process p
where s.audsid = sys_context('USERENV','SESSIONID')
and p.addr = s.paddr;
```

Listing 3

```
ps –ef | grep 1266
1266 1 oracleORCL (LOCAL=NO)
```

Listing 4

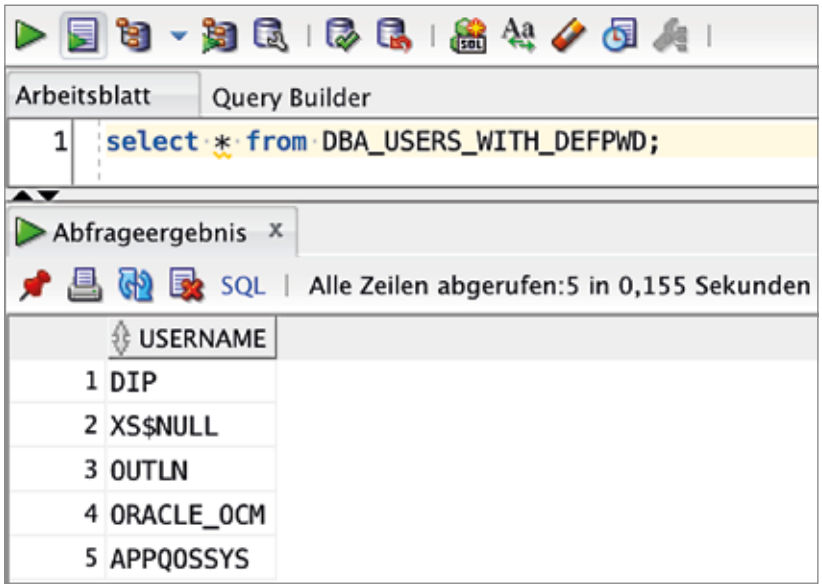

Abbildung 1: Benutzer mit Standard-Passwort

Oracle User Management

Immer wieder ist in verschiedenen Nachrichten zu lesen, dass Benutzerkonten aus verschiedensten Gründen geknackt wurden. Aus diesem Grund sieht es die PCI-DSS-Zertifizierung unter anderem vor, dass zum einen keine unnötigen Datenbank-Benutzer existieren, und zum anderen, dass eine bestimmte Passwort-Komplexität etabliert ist. Es gibt jedoch Fälle, in denen ein Benutzerkonto nicht gelöscht werden kann. In diesem Fall, seien es nun Oracle-System-User, technische User oder ehemalige Mitarbeiter, müssen diese gesperrt werden. Zusätzlich müssen alle Standard-Passwörter geändert werden.

Sowohl im „Database 2 Day + Security Guide“ als auch im „Database Installation Guide 12*c* Release 1“ kann eine Liste aller Oracle-Standard-Benutzer mit ihren Standart-Passwörtern eingesehen werden. Mithilfe des SQL-Statements „Select * from DBA_USERS_WITH_DEFPWD;“ können alle User auf einer Datenbank angezeigt werden, die aktuell ein Oracle-Standard-Passwort besitzen. *Abbildung 1* zeigt ein mögliches Ergebnis dieses SQL-Befehls.

In Bezug auf den User „XS$NULL“ gibt es einen Hinweis im „Database 2 Day + Security Guide“: „An internal account that represents the absence of database user in a session and the actual session user is an application user supported by Oracle Real Application Security. XS$NULL has no privileges and does not own any database object. No one can authenticate as XS$NULL, nor can authentication credentials ever be assigned to XS$NULL.” Mit anderen Worten: Es ist nicht möglich, das Passwort dieses Accounts zu ändern.

Passwort-Regeln

Auch bei der Passwort-Komplexität ist ein gewisser Level erforderlich. Dabei sollte zwischen technischen und persönlichen Benutzern unterschieden werden. Mithilfe von Datenbank-Profilen lässt sich diese Differenzierung realisieren. Ein Grund für eine Trennung in zwei verschiedene Profile ist zum Beispiel, wenn es für die Funktionalität des Systems kontraproduktiv sein kann, wenn ein Passwort nach 90 Tagen automatisch abläuft. Im folgenden Abschnitt werden einige Beispiele für die Passwort-Komplexität aufgezeigt und ein Beispielprofil gezeigt:

- Bei der ersten Anmeldung muss das Passwort geändert werden.
- Das Passwort besteht aus mindesten acht Zeichen und setzt sich aus Groß- und Kleinbuchstaben, Sonderzeichen und Zahlen zusammen. Bei der Umsetzung des Passworts sollte dabei nicht auf die Ehrlichkeit des Users vertraut werden, sondern vielmehr dieses mithilfe einer Passwortprüfungs-Funktion automatisiert überprüft werden.
- Die letzten vier Passwörter dürfen nicht wiederverwendet werden.
- Ein Passwort wird nach 90 Tagen ungültig und muss geändert werden.

Es existieren noch viele weitere Möglichkeiten, das Passwort vor möglichen Angreifern zu sichern. Die oben genannte Liste beinhaltet nur einen kleinen Ausschnitt der möglichen Optionen. *Listing 5* zeigt eine Beispiel-Implementierung eines Profils.

Das Profil weist folgende Konfiguration vor:

- Das Passwort ist maximal 90 Tage lang gültig.
- Ein Passwort kann 365 Tage lang nicht wiederverwendet werden.
- Die letzten vier Passwörter sind ebenfalls gesperrt.
- Nach sechs fehlgeschlagenen Anmeldeversuchen wird der Account gesperrt.
- Das Passwort wird mithilfe einer Funktion überprüft.

Daneben sind noch weitere Einstellungen möglich.

Audit-Einstellungen

Um im Falle eines Vorfalls eine nahezu lückenlose Aufklärung betreiben zu können, bietet Oracle mit seinen Audit-Einstellungen ein hohes Maß an Überwachung einzelner Aktionen an. Im Zuge einer geplanten Zertifizierung ist es unabdingbar, dass Aktionen – vor allem von personenbezogenen Accounts – überwacht werden.

Die Überwachung der technischen User ist an vielen Stellen nicht sinnvoll, da etwa das Löschen oder Verändern von Datensätzen durchaus im Rahmen gewisser Prozesse nötig ist. Dennoch könnte es beispielsweise sinnvoll sein, gewisse Tabellen-Inhalte von technischen Usern zu überwachen. Ein Beispiel wäre die Überprüfung verschlüsselter Kreditkarten-Nummern dahingehend, dass diese auch wirklich verschlüsselt in der Tabelle abgespeichert wurden. Neben der Überwachung der persönlichen und administrativen Benutzer muss auch eine lückenlose Überwachung aller Aktionen der

Oracle-Systemuser („SYS“ und „SYSTEM“) garantiert sein.

Bei der Audit-Einstellung „audit select table by %USERNAME% by access;“ werden alle Select-Statements auf Tabellen von einem Benutzer protokolliert. „%USERNAME%“ ist dabei durch den entsprechenden Benutzernamen zu ersetzen. Mehrere User können Kommaseparierat angegeben werden. Bei der Einstellung „audit session by %USERNAME% by access;“ wird jede Erstellung einer Session eines Users protokolliert.

Im Beispiel von *Listing 6* werden die Anpassung einer Sequenz, die Änderung einer Tabelle, das Löschen einer Tabelle und das Ausführen einer Prozedur für einen oder mehrere Benutzer überwacht. Weitere Einstellungen können dem „Database Security Guide“ entnommen werden.

Um die Aktivitäten des „SYS“-Users zu überwachen, können keine Audit-Einstellungen verwendet werden. Aus diesem Grund ist es notwendig, dass der Datenbank-Parameter „AUDIT_SYS_OPERATIONS=TRUE“ gesetzt ist. Die Audit-Informationen können dabei entweder in der Datenbank abgespeichert oder aber im Filesystem abgelegt sein. Eine Ablage im Filesystem hat den Vorteil, dass zum einen keine Datenbank-Operationen wie das Löschen der Audit-Informationen durchgeführt werden können, zum anderen lässt sich der Zugriff auf diese Dateien mithilfe von Verzeichnis-Berechtigungen beschränken und im Falle eines Datenbank-Ausfalls weiterhin lesen. Zudem kann man Dateien durch Monitoring-Lösungen automatisiert überwachen. Unternehmen, die im Besitz einer Enterprise Edition sind, haben die Möglichkeit, weitere detailliertere Audit-Einstellungen mittels Oracle Fine Grained Auditing (FGA) vorzunehmen *(siehe Listing 7)*.

Das Beispiel sorgt für eine Überwachung der Tabelle „T_CREDITCARD“ im Schema „SCOTT“ für die SQL-Befehle „SELECT“, „INSERT“, „UPDATE“ und „DELETE“. Von der Überwachung sind alle User außer dem Tabellen-Eigentümer „SCOTT“ betroffen. Mithilfe von FGA wäre auch die Überwachung der Selektion einer einzelnen Tabellenspalte möglich. Seit der Version 12*c* stehen mit dem Feature „Unified Auditing“ zusätzliche Möglichkeiten zur Verfügung. Weitere Informationen dazu stehen im „Database Security Guide“.

Listener-Konfigurationen

Auch der Listener ist immer wieder Angriffspunkt für mögliche „man-in-the-middle“-Attacken. Spätestens seit Veröffentlichung der Sicherheitslücke „CVE-2012-1675“ von Oracle, auch bekannt unter „TNS Listener Poison Attack“, sollte jeder Listener bis Version 11.2.0.3 den Parameter „SECURE_REGISTER_LISTENER_PROD=(IPC)“ enthalten. Dieser sorgt dafür, dass der Listener die Registrierung zur Datenbank nur noch über das IPC-Protokoll aufbaut. Da sich für dieses Protokoll die Datenbank und der Listener auf demselben Host befinden müssen, sind „man-in-the-middle“-Attacken nicht mehr ohne Weiteres möglich.

Ab Version 11.2.0.4 gibt es eine andere Möglichkeit, den Listener zu sichern (Stichwort: „VALID_NODE_CHECKING“, siehe MOS-Artikel „14538831.1“). Die zuvor beschriebene Lösung ist jedoch weiterhin möglich. Zusätzlich sollte, sofern kein Data Guard, RAC oder das PL/SQL Gateway für Apex im Einsatz ist, der Parameter „dynamic_registration_LISTENER_NAME=off“ gesetzt sein. Diese Einstellung bewirkt, dass der Listener keine dynamischen Registrierungen mehr zulässt. Zusätzlich sollte man das Listener-Log automatisiert überwachen. Nur so lassen sich mögliche Attacken auch frühzeitig erkennen.

„SQLNET.ORA“-Parameter

Mithilfe von SQLNET.ORA-Parametern ist es ebenfalls möglich, die Sicherheit der Datenbank zu erhöhen. Die beiden Parameter „TCP.VALIDNODE_CHECKING=YES“ und „TCP.INVITED_NODES=(%node1%,%node2%)“ konfigurieren die Datenbank so, dass nur von bestimmten Servern oder Clients aus Verbindungen zur Datenbank aufgebaut werden können. Der erste Parameter sorgt dabei für eine generelle Aktivierung dieses Features und dementsprechend für einen Abgleich der IP-Adressen zwischen der Adresse, von der die Verbindunganfrage kommt, mit denen, die in der White List stehen. Die White List ist durch den zweiten Parameter „TCP.INVITED_NODES“ dargestellt. Hier werden alle IP-Adressen Komma-separiert aufgelistet, die für eine Verbindung zur Datenbank freigegeben sind.

```
create profile personal_users limit
password_life_time 90
PASSWORD_REUSE_TIME 365
PASSWORD_REUSE_MAX 4
FAILED_LOGIN_ATTEMPTS 6
PASSWORD_VERIFY_FUNCTION f_check_personal_pwd;
```

Listing 5

```
audit alter sequence, alter table, delete table, execute procedure by
%USERNAME% by access;
```

Listing 6

```
begin
DBMS_FGA.ADD_POLICY(OBJECT_SCHEMA => 'SCOTT',
OBJECT_NAME => 'T_CREDITCARD',
POLICY_NAME => 'CC_varString_Audit',
AUDIT_CONDITION => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'')!= ''SCOTT'' ',
AUDIT_COLUMN => NULL,
ENABLE => TRUE,
STATEMENT_TYPES => 'SELECT,INSERT,UPDATE,DELETE',
audit_trail => DBMS_FGA.XML + DBMS_FGA.EXTENDED);
end;
/
```

Listing 7

„PFILE/SPFILE"-Parameter

Neben den bisherigen beschriebenen Möglichkeiten gibt es ganze Listen weiterer Möglichkeiten, mit denen die Datenbank mithilfe von Konfigurationsparametern gehärtet werden kann. Der Datenbank-Parameter „SEC_MAX_FAILED_LOGIN_ATTEMPTS=3" legt zum Beispiel die maximale Anzahl fehlgeschlagener Verbindungsaufbau-Versuche zur Datenbank fest. Sollte, wie in diesem Fall, der dritte Versuch eines möglichen Logins fehlschlagen, wird die Verbindung automatisch beendet.

Neben den zuvor genannten Parametern ist „SEC_PROTOCOL_ERROR_TRACE_ACTION=ALERT" ein weiterer wichtiger Kandidat. Mit dieser Einstellung werden sogenannte „Bad Packets" protokolliert. Je nach Konfiguration werden diese Informationen entweder ignoriert oder aber, wie bei der oben genannten Initialisierung, in Form eines kurzen Einzeilers in ein Trace-File und in das „Alert.log" der Datenbank geschrieben. Das eigentliche Problem wurde dadurch jedoch nicht gebannt oder gelöst.

Der Parameter „SEC_PROTOCOL_ERROR_FURTHER_ACTION=Delay,3" kann einen „denial-of-service"-Angriff etwas entschärfen. Er gibt die Zeit in Sekunden an, die die Datenbank nach der Identifizierung eines Client- oder Server-Protokoll-Fehlers („Bad packets") etwa aufgrund eines „denial-of-service"-Angriffs wartet, bevor die nächste Anfrage desselben Clients entgegengenommen wird.

Patch-Management (PSU)

In regelmäßigen Abständen (einmal pro Quartal) werden von Oracle sogenannte „Patch Set Updates" (PSU) veröffentlicht. Diese beinhalten neben normalen Bug Fixes auch Sicherheits-Updates. Nach Erscheinen eines solchen PSU sollte dieses umgehend auf allen Datenbanken installiert werden. Das Patch-Management ist auch ein wichtiger Bestandteil einer PCI-DSS-Zertifizierung. Weitere Informationen können unter dem fünften Punk der „Links" eingesehen werden.

Monitoring

Auch das oft stiefmütterlich behandelte Monitoring dient als zusätzlicher Schutz vor möglichen Angreifern. Um frühzeitig über mögliche Gefahren oder Probleme auf der Datenbank informiert zu werden, ist eine automatisierte Überwachung diverser Dienste und Dateien durch eine Monitoring-Lösung, wie das Monitoring-Module der Firma Herrmann & Lenz Solutions GmbH, unabdingbar.

Fazit

Das Thema „Sicherheit" ist in der heutigen Zeit eines der zentralen Themen in einer je-

den IT-Landschaft. Oracle bietet mit seiner Datenbank-Lösung eine Vielzahl von Sicherheitslösungen an. Viele dieser Einstellungen können dabei bereits in einer Standard-Edition eingesetzt werden. Sollten die in einer Standard Edition zur Verfügung stehenden Sicherheitsvorkehrungen nicht ausreichen, kann mit der Enterprise Edition auf weitere Funktionalitäten zugegriffen werden.

Mithilfe der lizenzpflichtigen Zusatzoption „Oracle Advanced Security" stehen nochmals weitere Features wie die Verschlüsselung sämtlicher Tabellen-Inhalte zur Verfügung. Eine Verschlüsselung von Datenbank-Exports und RMAN-Backups ist mit dieser Option ebenfalls möglich.

Die im Verlauf dieses Artikels gezeigten Möglichkeiten reichen jedoch nicht aus, um eine PCI-DSS-Zertifizierung erfolgreich bestehen zu können. Zudem ist zu beachten, dass nicht alle Bereiche und Einstellungen sowie Konfigurationen beleuchtet werden konnten. Dieser Artikel soll als Denkanstoß dienen und über die Vielfalt der möglichen Sicherheitseinstellungen und Konfigurationen einer Oracle-Datenbank informieren.

Weiterführende Links

[1] PCI Security Standards Council: *https://de.pcisecuritystandards.org/minisite/en*
[2] Database 2 Day + Security Guide: *https://docs.oracle.com/database/121/TDPSG/toc.htm*
[3] Database Installation Guide 12c Release 1: *https://docs.oracle.com/database/121/LADBI/toc.htm*
[4] Database Security Guide: *https://docs.oracle.com/database/121/DBSEG/toc.htm*
[5] Critical Patch Updates: *http://www.oracle.com/technetwork/topics/security/alerts-086861.html*
[6] HL-Monitoring: *https://hl-solutions.de/produkte/monitoring-module*
[7] OraInfo: *https://hl-solutions.de/produkte/orainfo*

Johannes Kraus
johannes.kraus@hl-services.de

OpenStreetMap und Oracle Spatial im Zusammenspiel

Markus Lindner, CISS TDI GmbH

Dieser Artikel stellt das OpenStreetMap-Projekt kurz vor und zeigt, wie OSM-Daten in eine Oracle-Datenbank importiert werden können. Dabei kommen die Herausforderungen beim Import, die notwendige Datenmodellierung und die vielfältigen Nutzungsmöglichkeiten in Oracle zur Sprache.

Seit mehr als zehn Jahren gibt es das OpenStreetMap-Projekt, das sich seit der Gründung im Jahr 2004 durch Steve Coast zum Ziel gesetzt hat, eine freie Datenbank für weltweite Kartendaten bereitzustellen. In den letzten Jahren erfreut sich das Projekt immer größerer Beliebtheit, so stieg die Zahl der registrierten Benutzer von 2009 mit etwa 100.000 über 2013 mit einer Million bis auf mehr als zwei Millionen im Jahre 2015.

Im Grunde ist OSM eine Datensammlung über Gegenständliches, Nichtgegenständliches und Benennungen. Unter „gegenständlich" sind Themen wie Wege, Eisenbahnstrecken, Wasserläufe, Freizeitanlagen, Naturflächen und andere zu verstehen. Insgesamt gibt es 23 Themen mit vielen weiteren Unterthemen. „nichtgegenständlich" sind Routen, Grenzen oder Informationen über Nachbarschaften und Beschränkungen. „Benennungen" schließlich sind Straßennamen, Ortsangaben, Adressen oder ähnliche Informationen.

Die Nutzung der Daten geschieht in der Regel online als Karten im Web-Browser, oder als Online-Routenberechnung, vergleichbar mit den bekannten Möglichkeiten von Google oder Bing. Im Unterschied dazu sind die Daten allerdings auch offline verfügbar; so gibt es zum Beispiel Möglichkeiten, die OpenStreetMap-Daten direkt in GPS-Geräten ohne Internet-Verbindung zu nutzen. Den aktuellen Umfang der Datenbasis kann man sich jederzeit über *„http://www.openstreetmap.org"* ansehen, *Abbildung 1* zeigt die entsprechende Statistik im Juni 2015.

Um jedoch einen besseren Überblick über die Datenmengen zu erhalten, lohnt sich ein Blick auf die Dateigrößen, die ein Datenbank-Auszug ergibt. Für die Nutzung, die nicht auf bestehenden Diensten basiert, können OpenStreetMap-Daten in einem XML-Format bezogen werden, das es in verschiedenen Komprimierungen gibt. *Tabelle 1* gibt die Dateigrößen für ausgewählte Gebiete an, in XML unkomprimiert und im „bz2"-Format komprimiert.

Zusammenfassend lässt sich sagen, dass das OpenStreetMap-Projekt mittler-