# DATENBANKEN & SQL

## – Datenschutz & Sicherheit –

Kurs am RRZK der Universität zu Köln

Rüdiger Voigt, M.A.

13.08.–17.08.2018

# Lage

# Lage

- Digitale Kriminalität ist Alltag.
- Alle aus dem Internet erreichbaren Systeme sind permanent automatisierten Angriffsversuchen ausgesetzt.
- Hacker verkaufen, nutzen oder veröffentlichen sensible Daten.
- Datenbanken sind ein wertvolles Ziel.

# Wie und warum betrifft das Universitäten?

- Forschungsdaten können sehr wertvoll sein:
  - potentielle Patente,
  - Ergebnisse der medizinischen Forschung,
  - vertrauliche Informationen zu Mitarbeitern / Studenten / Patienten,
- Die meisten Teilnehmer dieses Kurses werden in die Wirtschaft wechseln. Hier sind Datenbanken voll mit Geschäftsgeheimnissen.
- Scriptkids ist oft egal auf welchem System sie sich austoben.
- ...

# DSGVO

Am 25.Mai 2018 trat die Datenschutz-Grundverordnung (DSGVO) der EU in Kraft. Sie bildet ein Grundgerüst in der gesamten EU, wird aber gegebenfalls durch nationale Gesetze ergänzt.

Datenschutz wird strenger:

- Vielfach ist nur Opt-In statt Opt-Out erlaubt.
- Umfangreiche Dokumentations- und Auskunftspflichten.
- Verpflichtung zum technischen Datenschutz.
- ...

# DSGVO: Strafen

## Hohe Strafen

Für Verstöße gegen die DSGVO sind Maximalstrafen von 20 Mio. Euro beziehungsweise 4 % des globalen Unternehmensumsatzes möglich.

# personenbezogene Daten

- Name
- Anschrift
- E-Mail-Adresse
- Ausweisnummer
- IP-Adressen
- ...

# personenbeziehbare Daten

# Auskunftsrechte

# Anonymisierung von Daten

- Unumkehrbare Anonymisierung ist schwer!
- Ein Merkmal das einzeln nie einer Person zugeordnet werden könnte, kann in Kombination mit mehreren gleichartigen Merkmalen sehr wohl eindeutig sein.
- Anonymisierung kann Verzicht auf Daten bedeuten, die einen Rückschluss ermöglichen. Oft werden Daten auch aggregiert.

# Angriffswege

# Vielfalt der Angriffswege

- Die Vielzahl der Möglichkeiten ein DBMS anzugreifen sprengt den Rahmen dieses Kurses.
- Es muss noch nicht einmal ein direkter Angriff auf das DBMS sein, sondern Angriffe auf die PCs, Smartphones o.ä. von Personen mit Zugang sind ein Risiko.

## SQL-Injection

SQL Injection ist einer der häufigsten, wenn nicht *der* häufigste Angriff auf DBMS:

- Wenn Eingaben von Nutzern (zum Beispiel in einer Suchmaske oder als Parameter von Scriptaufrufen) nicht, oder nicht ausreichend überprüft werden *bevor* sie an die Datenbank durchgeleitet werden, besteht die **Gefahr, dass Nutzer SQL Befehle einschleusen**. Damit können Sie unter Umständen Daten löschen, einschleusen oder unbefugt abfragen.
- Es gibt Programme für Hacker, welche derartige Angriffe weitgehend automatisieren können.

# Daten schützen

# Datensparsamkeit

Sie brauchen per se einen Grund um personenbezogene Daten zu speichern. In der Praxis mag es manchmal so sein, dass eine Begründung relativ einfach möglich ist.

Datensparsamkeit bedeutet, dass Sie einen Weg suchen sensible Daten gar nicht erheben zu müssen. Das kann unbequem sein, aber Sie ersparen sich den Aufwand diese Daten zu schützen und unberechtigte Dritte können auf nicht-existente Daten nicht zugreifen.

# Rollen- und Rechtesystem

## principle of least privilege

In MariaDB (und in fast allen anderen DBMS) können Sie sehr genau und meist zumindest runter auf Tabellenebene festlegen, zu welchen Aktionen ein bestimmter Account berechtigt ist.

Weisen Sie Ihren Nutzern und Skripten / Programmen jeweils eigene Accounts zu und beschränken Sie deren Rechte auf das für die Nötige. Sollte nun einer dieser Accounts kompromittiert werden, begrenzen Sie damit den Schaden.

# Was tun gegen SQL-Injection?

## GOLDENE REGEL

**NEVER TRUST (USER) INPUT DATA!**

Alle Eingabe-Kontrollen in einem Browser (sei es nun mit JavaScript oder HTML5) können sehr leicht umgangen werden. Im Browser vermeiden Sie in erster Linie, dass Nutzer unsinnige Werte eingeben.
Bevor Daten an eine Datenbank übergeben werden, müssen Sie diese auf einem System unter Ihrer Kontrolle prüfen. Bevorzugte Methoden: Escaping oder noch besser Whitelisting.
Auf jeden Fall sollten Sie **Prepared Statements** zur Interaktion mit der Datenbank verwenden!

# Sicherheit auf Netzwerkebene

- Muss der Server mit dem DBMS aus dem Internet erreichbar sein, oder genügt ein Subnetz wie das eines Instituts?
- Die Anwendung fail2ban (`http://heise.de/-270140`) wird sehr oft genutzt um automatisierte Login-Versuche zu erschweren.
- Mittels einer Passwort-Policy sollten Sie zu simple Passwörter ausschließen. Dienste können mit zufallsgenerierten Passwörtern auf eine Datenbank zugreifen.

# Verschlüsselung

- Der Performanceverlust durch Verschlüsselung ist gering. Die gängigen Algorithmen (AES, . . . ) sind auf modernen CPUs hardwarebeschleunigt.
- Sie können die ganze Datenbank oder auch nur sensible Teile verschlüsseln.

# Backup

- Ein regelmäßiges Backup ist unbedingt notwendig.
- Ein Backup können Sie ebenfalls verschlüsseln.
- Den Fall, dass Sie ihr Backup einmal brauchen, sollten Sie mit einem Testsystem simulieren um zu prüfen, ob es zuverlässig funktioniert.

# Salted Hashes

- Passwörter sind eine beliebte Beute für Angreifer, denn sehr viele Personen nutzen das gleiche Passwort für mehrere Dienste.
- Sie müssen Passwörter nicht im Klartext speichern. Stattdessen speichert man so genannte Hashes (Prüfwerte).
- Einfache Hashwerte reichen Sie nicht aus. Diese sollten mittels eines Salt geschützt werden.

Den kompletten Foliensatz finden Sie in aktueller Version unter:

`https://www.ruediger-voigt.eu/kurs-datenbanken-und-sql.html`