Einsatz und Realisierung von Datenbanksystemen

ERDB Übungsleitung

Alice Rey, Maximilian Bandle, Michael Jungmair

i3erdb@in.tum.de

Folien erstellt von Maximilian Bandle & Alexander Beischl

Organisatorisches
Disclaimer

Die Folien werden von der Übungsleitung allen Tutoren zur Verfügung gestellt.

Sollte es Unstimmigkeiten zu den Vorlesungsfolien von Prof. Kemper geben, so sind die Folien aus der Vorlesung ausschlaggebend.

Falls Ihr einen Fehler oder eine Unstimmigkeit findet, schreibt an i3erdb@in.tum.de mit Angabe der Foliennummer.

Sicherheitsaspekte

Sicherheitsaspekte
k-Anonymität

- Nur Aggregatanfragen erlaubt (SUM, COUNT, AVG…)

- Mindestens k Datensätze müssen aggregiert werden

```
create view HärteDerVorlesung(VorlNr, Härte) as
  select VorlNr, avg(Note)
  from prüfen
  group by VorlNr
  having count(*) > 11;
```

Aufgabe 1

Die AMU (Alexander-Maximilians-Universität) hat eine Datenbank mit den Durchschnittsnoten aller Studenten mit Name.

Schema: {[Name | Durchschnittsnote]}

Der einzige Schutzmechanismus dieser Datenbank ist, dass immer mindestens 3 Tupel aggregiert werden. Als Ausgabe sind nur `COUNT` und `AVG` zulässig.

1. Beschreibe eine Methode, um herauszufinden, was die Note des schlechtesten Studenten der AMU ist
2. Max will Alex' Durchschnittsnote herausfinden. Dazu stellt er folgende Anfragen mit Ergebnis:

```
SELECT AVG(Durchschnittsnote), COUNT(*) FROM Noten => (2,5 ; 10.000)
SELECT AVG(Durchschnittsnote), COUNT(*) FROM Noten WHERE Name != 'Alex'
=> (2,5001 ; 9.999)
```

   Kann er aus den Ergebnissen Alex' Note berechnen? Wenn ja, wie, wenn nein, wieso nicht?

Sicherheitsaspekte
SQL-Injections

- Hinter nahezu allen WebApps stehen Datenbanksysteme
- Aus den eingegebenen Parametern werden SQL-Anfragen generiert
- Den Eingaben darf niemals getraut werden, da sie SQL-Statements enthalten können
- Nutzer könnte auf dem Server Code ausführen

Sicherheitsaspekte
SQL-Injections

Name	Passwort	Geheimnis
Max	MaxIstToll!	Lässt Übungen halten
Alex	MaxIstDoof;)	Ist fleißig und schreibt MA ;)
Hacker	54321	Hat große Pläne

Beispiel WebApp

Name: Max

Passwort: MaxIstToll!

Dein Geheimnis ist: *Hält keine Übung mehr*

Sicherheitsaspekte
SQL-Injections

Beispiel WebApp

Name $name

Passwort $passwort

Dein Geheimnis ist: $geheimnis

Code der WebApp

```
$geheimnis = SELECT Geheimnis FROM Geheimnisse
  WHERE Name = '$name' AND Passwort = '$passwort';

print “Dein Geheimnis ist: “ + query($geheimnis)
```

Sicherheitsaspekte
SQL-Injections

Beispiel WebApp

Name: Max

Passwort: MaxIstToll!

Dein Geheimnis ist: *Lässt Übungen halten*

Code der WebApp

```
$geheimnis = SELECT Geheimnis FROM Geheimnisse
  WHERE Name = 'Max' AND Passwort = 'MaxIstToll!';

print “Dein Geheimnis ist: “ + query($geheimnis)
```

Sicherheitsaspekte
SQL-Injections

Beispiel WebApp

Name: Alex

Passwort: ???' OR 'x'='x

Dein Geheimnis ist:

Lässt Übungen halten
Ist fleißig und schreibt MA ;)
Hat große Pläne

Hinweis:
AND bindet stärker als OR, deshalb werden hier alle Geheimnisse ausgegeben.

Code der WebApp

```
$geheimnis = SELECT Geheimnis FROM Geheimnisse
  WHERE Name = 'Alex' AND Passwort = '???' OR 'x'='x';

print “Dein Geheimnis ist: “ + query($geheimnis)
```

Sicherheitsaspekte
SQL-Injections

Beispiel WebApp

Name: Alex

Passwort: ???'; DELETE FROM Geheimnisse WHERE 'x' = 'x

Dein Geheimnis ist: ERROR

Code der WebApp

```
$geheimnis = SELECT Geheimnis FROM Geheimnisse
  WHERE Name = 'Alex' AND Passwort = '???';
  DELETE FROM Geheimnisse WHERE 'x' = 'x';

print “Dein Gehemnis ist: “ + query($geheimnis)
```

Aufgabe 2

Die Prüfungs-Datenbank der AMU wurde leider von einem unachtsamen Programmierer geschrieben. Es gibt ein Formular, in dem man nach seinen Klausurnoten suchen kann, allerdings wird die Benutzereingabe nicht geprüft.

Schema: Prüfung: {[Vorlesung, Note, Matrikelnummer]}

Benutzte Anfrage:

```
SELECT *
FROM Prüfung
WHERE Matrikelnummer='{MatrikelNr}' AND Vorlesung='{Benutzereingabe}'
```

- Benutzereingabe: ist die Benutzereingabe.
- MatrikelNr: wird automatisch mit deiner Matrikelnummer befüllt.

Schreibe eine Benutzereingabe, mit der du alle deine Noten auf 1.0 setzen kannst.

Aufgabe 3

Sie wollen der AMU helfen, ihre Datenbank sicherer zu machen, und finden bei einer kurzen Suche im Internet *Prepared Statements* und *Input Sanitization*.

1. Erklären Sie kurz beide Methoden, besonders deren Unterschiede in der Behandlung von bösartigen Eingaben.
2. Beschreiben Sie Vorteile von *Prepared Statements*, die über Sicherheit hinaus reichen.
3. Nachfolgend sehen Sie die Funktion `exec_unsafe()`, die das Formular serverseitig aufruft, um die Klausurnote abzufragen. Ersetzen Sie diese durch eine Funktion `exec_prep()`, die mittels eines *Prepared Statements* auf die Datenbank zugreift.

```
#include <pqxx/pqxx>
#include <iostream>
#include <string>
pqxx::result exec_unsafe(pqxx::connection& conn, int matrnr, std::string vorl){
  std::string q = "SELECT * FROM pruefung WHERE matrikelnummer=",
    q2 = "AND vorlesung='", q3 ="'";
  pqxx::work tx{conn, ""}; // Begin of transaction
  pqxx::result r(tx.exec(q + std::to_string(matrnr) + q2 + vorl + q3));
  tx.commit(); // Commit transaction
  return r;
}
int main(int argc, char* argv[]){
  pqxx::connection conn;
  auto r = exec_unsafe(conn, 123, "Grundzuege");
  for(auto row: r)
     std::cout << row["note"] << std::endl;
  return 0;
}
```

Sicherheitsaspekte
Input Sanitization

Beispiel WebApp

Name: Alex

Passwort: ???“ OR “x“=“x

Dein Geheimnis ist: Kein Ergebnis

sanitize löscht bzw. ersetzt alle Zeichen die aus einer Zeichenkette “ausbrechen” können.

```
sanitize(???“ OR “x“=“x) =>
“??? OR x=x“
```

Code der WebApp

```
$geheimnis = SELECT Geheimnis FROM Geheimnisse
  WHERE Name     = sanitize(Alex) AND
        Passwort = sanitize(???“ OR “x“=“x);

print “Dein Geheimnis ist: “ + query($geheimnis)
```

Sicherheitsaspekte
Stored Procedures

Beispiel WebApp

Name: Alex

Passwort: ???“ OR “x“=“x

Dein Geheimnis ist: Kein Ergebnis

prepare bereitet die Anfrage vor und legt bei den Platzhalten den Datentyp fest. Die Struktur kann sich nicht mehr ändern!

queryPrepare ersetzt die Platzhalter als String. Damit ist die bösartige Eingabe unwirksam, da sie die Struktur nicht ändert .

Code der WebApp

```
$geheimnis = prepare(SELECT Geheimnis FROM Geheimnisse
  WHERE Name = $1 AND Passwort = $2;)

print “Dein Geheimnis ist: “ +
      queryPrepare($geheimnis, Alex, ???“ OR “x“=“x)
```

Aufgabe 4

Bob hat ein Vorlesungsverzeichnis für die Universität programmiert und unter `http://db.in.tum.de/~schuele/sql_verzeichnis.html` online gestellt.

Um die Suche zu erleichtern, kann die Anzahl der SWS durch ein Parameter eingeschränkt werden. Finden sie einen speziell präparierten Parameter, bei dessen Eingabe statt der Vorlesungen die Liste der Studenten ausgegeben wird. Die Datenbank folgt dem bekannten Universitätsschema.

Bob erfährt von der Sicherheitslücke und schlägt vor die bekannten Tabellen einmalig mit zufälligen Namen umzubennen, so seien sie nicht zu finden. Würde diese *Sicherheitsmaßname* helfen?

Aufgabe 5

Sie haben die User-Tabelle zweier Pizzalieferanten ausgelesen, jedoch scheinen die Passwörter uncharakteristisch kompliziert zu sein. Das von Ihnen erhaltene Resultat ist das Folgende:

```
id          name        password
----------  ----------  ----------------------------------------
1           luigi       4d75e8db6a4b6205d0a95854d634c27a
2           mario       fe78ea401158dd5847c4090b8bb22477e510febf
```

- Was könnte der Grund für diese hexadezimalen, 32 bzw. 41 Stellen lange Passwörter sein?
- Können Sie trotzdem den Klartext finden?
- Wie können Sie das Passwort sicherer Speichern?
- Wie können Sie für diese Art von Passwortspeicherung Bruteforce-Attacken erschweren?

Aufgabe 6

Sie fangen die folgende, mit RSA verschlüsselte Nachricht ab: 13. Sie kennen den öffentlichen Schlüssel (3,15). Wie lautet die Nachricht im Klartext? Geben Sie die komplette Herleitung an.

Aufgabe 6
RSA

Öffentlicher Schlüssel (e,N)
Privater Schlüssel (d,N)

$$N = p \times q$$

mit p und q sehr großen Primzahlen. e, der sogenannte Verschlüsselungsexponent wird als teilerfremde Zahl zu $\phi(N)$ gewählt, wobei gilt $1 < e < \phi(N)$. $\phi(N)$ ist hierbei definiert als $\phi(N) = (p - 1) \times (q - 1)$. d, der sogenannte Entschlüsselungsexponent, ist gerade das multiplikative Inverse von e bezüglich des Moduls $\phi(N)$. Die Berechnung erfolgt mittels erweiterten euklidschen Algorithmus.
Die Entschlüsselung einer verschlüsselten Nachricht C zu ihrem Klartext K erfolgt mittels der Formel

$$K = C^d \bmod N$$

Aufgabe 6
RSA

Aus der Angabe wissen wir:
- N = 15
- $e = 3$
- C = 13

Wir müssen also zunächst d berechnen. Dies wäre einfach, wenn wir $\varphi(N)$ wüssten. Hierzu ist die Primfaktorzerlegung von N nötig. Dies ist für die Zahl 15 äußerst einfach, es gilt $N = 5 \times 3$. Damit ist $\varphi(N) = 4 \times 2 = 8$. Die Lösung der Kongruenz

$$e \times d \equiv 1 \bmod \varphi(N)$$

bzw. im konkreten Fall

$$3 \times d \equiv 1 \bmod 8$$

können wir raten, indem wir alle im Bezug auf 8 teilerfremden Zahlen z betrachten, für die gilt: $1 < z < 8$.
Für $z = 3$ gilt: $3 \times 3 \equiv 1 \bmod 8$, womit $d = 3$ ist.
Wir entschlüsseln nun die Nachricht:

$$K = 13^3 \bmod 15 = 7$$

Der Klartext K ist also 7.

Aufgabe 7

Ein bekannter Anwendungsbereich des RSA-Kryptosystems ist das Verschlüsseln und Signieren von E-Mails. Die beiden Standards sind S/MIME und OpenPGP. Für ersteres benötigen Sie ein Zertifikat, ausgestellt von einer Zertifizierungsstelle, wie sie die TUM für alle E-Mail-Adressen ausgibt: `https://my.ito.cit.tum.de/zertifikat/` für `in.tum.de` oder `ma.tum.de` oder `https://go.tum.de/343149` für `tum.de`

Ein Schlüsselpaar für OpenPGP können Sie sich jederzeit selbst und für jede E-Mail-Adresse zulegen. Beschäftigen Sie sich näher mit OpenPGP oder S/MIME, damit Sie Ihre E-Mails verschlüsseln können und schicken Sie Ihrem Tutor eine verschlüsselte und signierte E-Mail. Ihr Tutor erklärt Ihnen während der Übungsstunde, an welche Adresse Sie Ihre E-Mail schicken sollen und wo Sie den entsprechenden öffentlichen Schlüssel erhalten. Dafür erhalten Sie einen Bonuspunkt.

Aufgabe 7

- Mail-Adresse:

- Key: pgp.mit.edu

Fragen?