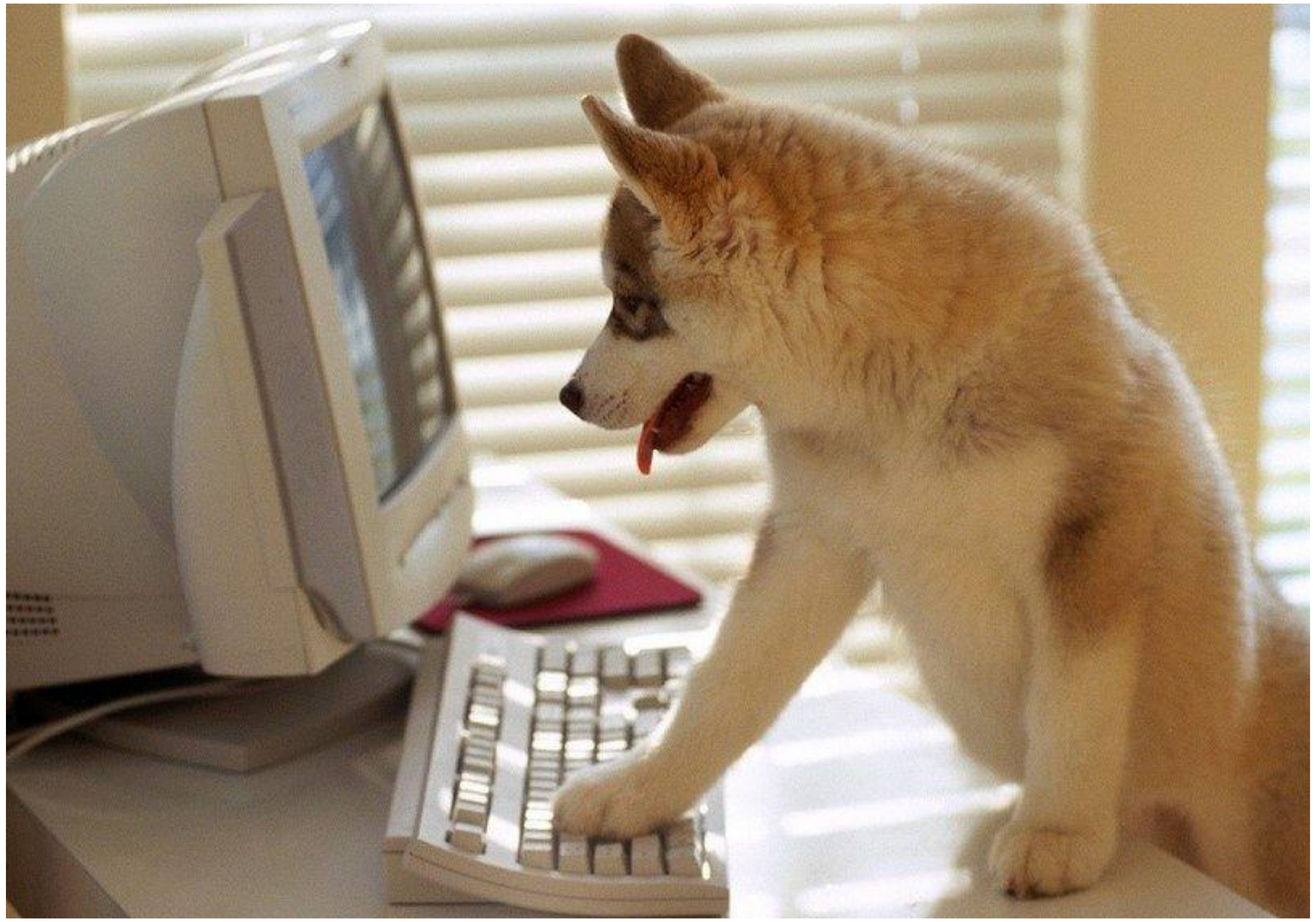
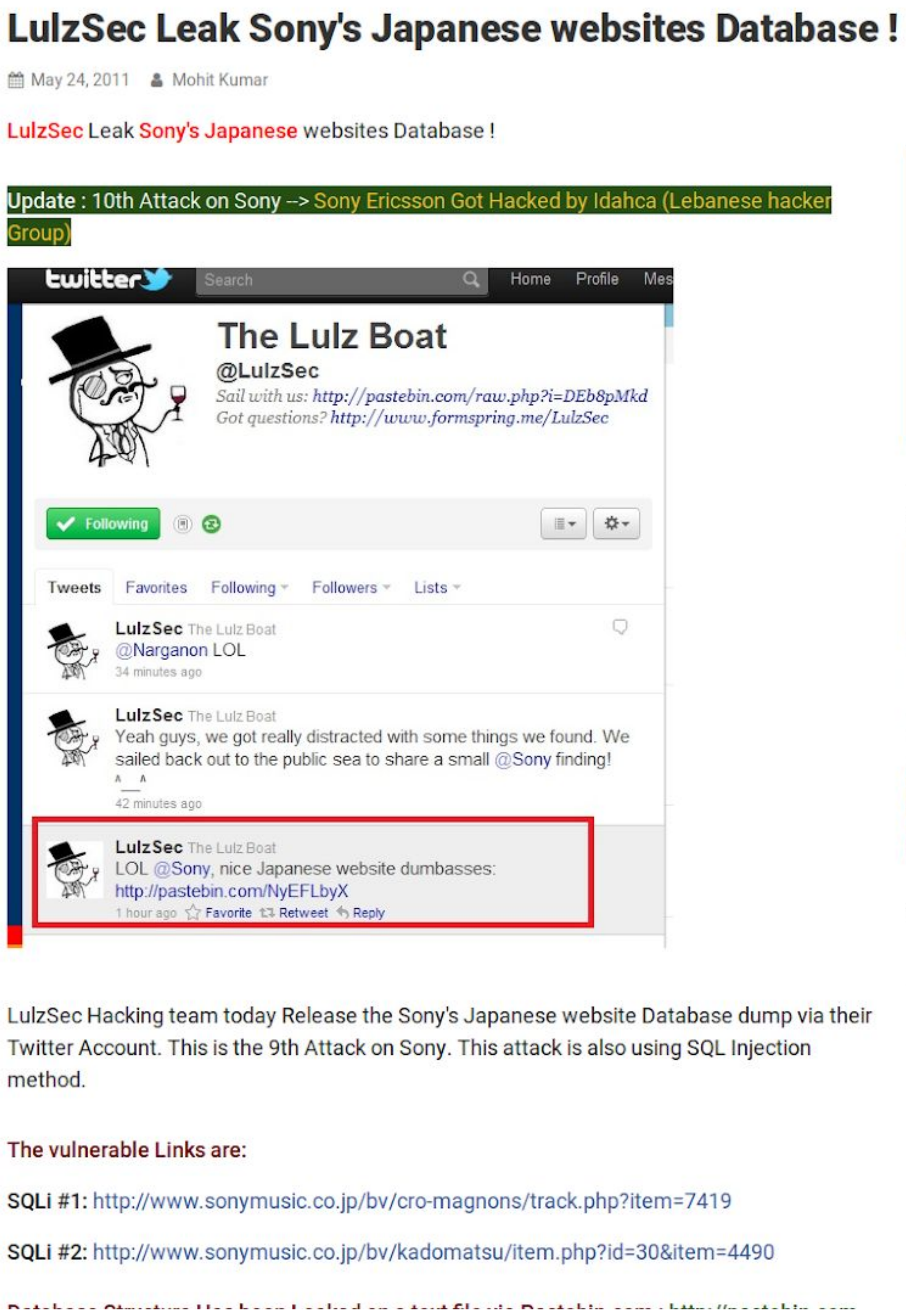
# Programming Tutorial [Advanced]



# SQL Injection



**LulzSec Leak Sony's Japanese websites Database !**

May 24, 2011 Mohit Kumar

LulzSec Leak Sony's Japanese websites Database !

Update : 10th Attack on Sony --> Sony Ericsson Got Hacked by Idahca (Lebanese hacker Group)

The Lulz Boat
@LulzSec
*Sail with us: http://pastebin.com/raw.php?i=DEb8pMkd*
*Got questions? http://www.formspring.me/LulzSec*

LulzSec The Lulz Boat
@Narganon LOL
34 minutes ago

LulzSec The Lulz Boat
Yeah guys, we got really distracted with some things we found. We sailed back out to the public sea to share a small @Sony finding!
^__^
42 minutes ago

LulzSec The Lulz Boat
LOL @Sony, nice Japanese website dumbasses:
http://pastebin.com/NyEFLbyX
1 hour ago Favorite Retweet Reply

LulzSec Hacking team today Release the Sony's Japanese website Database dump via their Twitter Account. This is the 9th Attack on Sony. This attack is also using SQL Injection method.

**The vulnerable Links are:**

**SQLi #1:** http://www.sonymusic.co.jp/bv/cro-magnons/track.php?item=7419

**SQLi #2:** http://www.sonymusic.co.jp/bv/kadomatsu/item.php?id=30&item=4490

Database Structure Has been Leaked on a text file via Pastebin.com : http://pastebin.com

Image Source:
https://thehackernews.com/2011/05/lulzsec-leak-sonys-japanese-websites.html

# SQL Injection

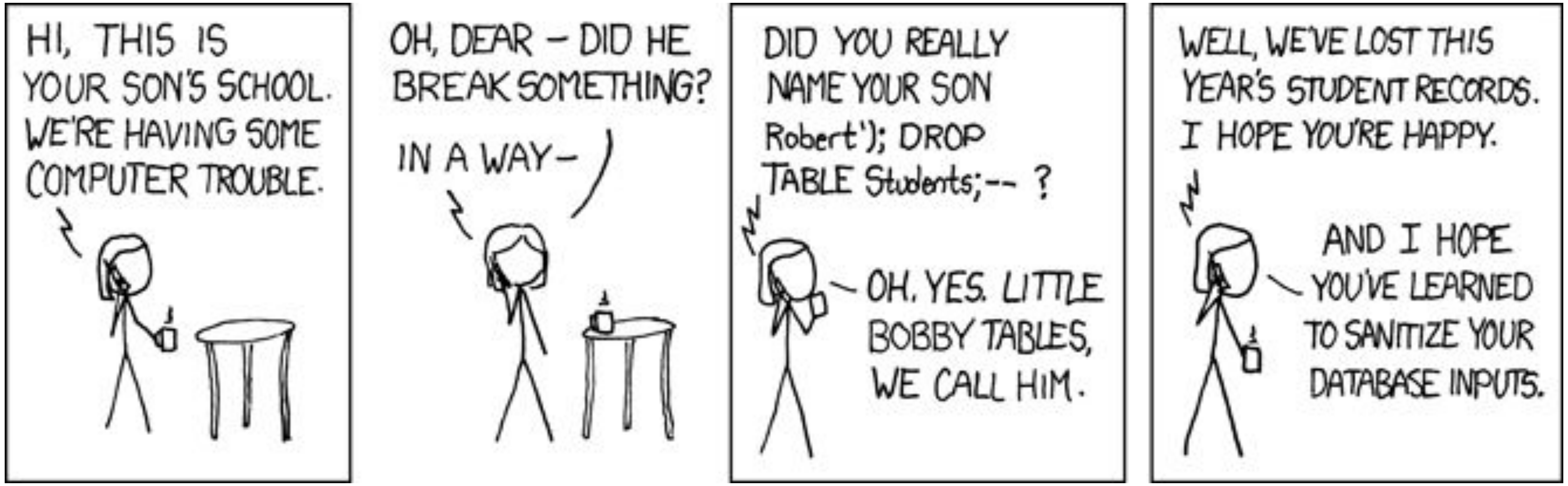

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Image Source: https://xkcd.com/327/

# Statement

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   Statement stat = conn.createStatement();
   stat.executeUpdate(“CREATE TABLE people (name, occupation);”);
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# PreparedStatement

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   conn.setAutoCommit(false);
   PreparedStatement prep = conn.prepareStatement(
      “INSERT INTO people VALUES (?,?);”);
   prep.setString(1, “Turing”);
   prep.setString(2, “computers”);
   prep.addBatch();
   prep.setString(1, “Einstein”);
   prep.setString(2, “physics”);
   prep.addBatch();
   prep.executeBatch();
   conn.commit();
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# PreparedStatement

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   conn.setAutoCommit(false);
   PreparedStatement prep = conn.prepareStatement(
      “INSERT INTO people VALUES (?,?);”);
   prep.setString(1, “Turing”);
   prep.setString(2, “computers”);
   prep.addBatch();
   prep.setString(1, “Einstein”);
   prep.setString(2, “physics”);
   prep.addBatch();
   prep.executeBatch();
   conn.commit();
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# PreparedStatement

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   conn.setAutoCommit(false);
   PreparedStatement prep = conn.prepareStatement(
     “INSERT INTO people VALUES (?,?);”);
   prep.setString(1, “Turing”);
   prep.setString(2, “computers”);
   prep.addBatch();
   prep.setString(1, “Einstein”);
   prep.setString(2, “physics”);
   prep.addBatch();
   prep.executeBatch();
   conn.commit();
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# ResultSet

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   Statement stat = conn.createStatement();
   ResultSet rs = stat.executeQuery(“SELECT * FROM people;”);
   while(rs.next()){
      System.out.println(“name = ” + rs.getString(“name”));
      System.out.println(“job = ” + rs.getString(“occupation”));
   }
   rs.close();
   conn.close();
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# ResultSet

```
try{
   Class.forName(org.sqlite.JDBC);
   Connection conn = DriverManager.getConnection(jdbc:sqlite:test.db);
   Statement stat = conn.createStatement();
   ResultSet rs = stat.executeQuery(“SELECT * FROM people;”);
   while(rs.next()){
      System.out.println(“name = ” + rs.getString(“name”));
      System.out.println(“job = ” + rs.getString(“occupation”));
   }
   rs.close();
   conn.close();
}
catch(SQLException e){
   e.printStackTrace();
}
catch(ClassNotFoundException e){
   e.printStackTrace();
}
```

# Hints

Execute the following Statement every time you start your code (Not applicable in *real life*):

```
DROP TABLE IF EXISTS <YOUR_TABLE_NAME>
```

Download the latest version of sqlite-jdbc here:

https://bitbucket.org/xerial/sqlite-jdbc/downloads/

Compile and run your code as follows:

```
javac -cp /path/to/sqlite.jar FileName.java
```

```
java -cp .:/path/to/sqlite.jar ClassName
```

*(Note: For Windows use ‘;’ instead of ‘:’)*

# Databases

In this course we will use Github Classroom

1. Get a Github Account if you don’t have one
2. Go to: https://classroom.github.com/a/JRpxxXpn (or scan the QR Code with your phone)
3. Authorize Github and accept the assignment
4. Click on the repository