

# CS166 Project 4, Part II: Banner

Due on April 30, 2009 at 10:00 pm EST

Tiers Out: April 21, 2009

## Silly Premise

You've revealed your exploits to the University. Now you have to fix them or get expelled.

## Setup

The workspace is in `/course/cs166/asgn/Gradeview`. Copy it over to your personal folders and you should be able to open eclipse using that workspace.

For this project you will have to use the ULTRA AWESOME 166 ECLIPSE. You can do so by typing `'cs166_eclipse'`, and if you have the courage, pressing enter.

## Database

We are using SQLite to back our data for the grades server. If you want to interactively explore the the database, run

```
cs1ab9b: ...banner/data% /course/cs166/share/sqlite/bin/sqlite3 database file
```

Type `.help` for basic help, or visit <http://www.sqlite.org/lang.html> for official documentation. We are actually running sqlite through a Java wrapper, which you can find documentation for at <http://www.ch-werner.de/javasqlite/> (or google "sqlite java" for the top result).

```
SQLite package: /course/cs166/share/sqlite
```

```
SQLite JNI: /course/cs166/share/sqlitejni
```

```
SQLite Jar: /course/cs166/share/sqlitejar/sqlite.jar
```

## Eclipse

While not necessary, an IDE is heavily suggested to allow you to quickly learn the code and refactor what is needed. Some useful Eclipse shortcuts:

Find references to an instance: Right click on a class name (in explorer or code), References → Workspace, or Ctrl-Shift-G

Find constructor for an instance: Ctrl-click on a class name to jump to its constructor

Rename class, instance, or argument throughout the project. Switch parameter names and order throughout the project. Right click on the item of interest and choose the appropriate refactor option.

## Dates

**April 30, 10pm**

Handin: Run our handin script.

**April 30 - May 5th**

Interactive Grading. Your grader will contact you this week and setup a date.

## Exploits

You are responsible for fixing ALL the exploits listed below, any exploits not mentioned here will still be worth fixing for contest pwnage, just don't mention them to anyone else!

- **Injection:** The shoutbox has a few javascript injection/XSS problems.
- **Injection:** SQL injection. Just about everything on the site is vulnerable to SQL injection, sanitize the inputs.
- **Session:** The 'Remember me' checkbox saves the password in plaintext on the user's computer, making it vulnerable to cross site scripting attacks.
- **Forgot?:** The default 'Forgot password?' scheme has easy questions. There are much better ways to retrieve forgotten passwords, through email, a reset option, just don't leave it as something for Firefox to do!
- **Handbacks:** Handbacks are accesible by anyone if they have the correct path on the website, and this path itself is merely the MD5 of the name of the handback. Fix this scheme so it is impossible to view someone else's handback.

Note: Not all solutions to this are equal, security by obscurity is **NEVER** an ideal solution.

- **Arbitrary upload:** A TA can upload a homework as handback which can contain any valid html code, which includes any valid javascript/JSP, so on, which can be used to do all sorts of nasty things to student's accounts and more. Perhaps a real format like pdf would be superior?
- **TAs FOREVER:** Once you are a TA for a class, you can forever and always upload and change the grade of any student you have ever TAed. This is not desirable behavior.
- **SSL:** https://. This entire protocol communicates over the http protocol, completely in the clear, so even if you manage to fix all the other vulnerabilities you can still very easily intercept packets between client and server. Switch the entire app over to using SSL.
- **Change any table:** Modification of the 'hidden' values in the Change Grades view gives you the option to change basically any table in the database you want. There are better ways to change a database than to store the table and column information on the client side.

## A Cautionary Note

This is intense stuff. You will have to majorly redesign major portions of the code, we suggest svn. You WILL step on each other's toes as you code this up, and you will need to go back to a working version. Subclipse is already installed on the cs166 version of eclipse, if you would like to use it. If there is some part of the code which is unclear, let the TA staff know, we'll do our best to explain it or redo it. Most of the code is straightforward, though it does require some basic understanding of the structure of Servlets, which can be easily googled and javadoced.