Lab: Fixing Vulnerabilities

You're going to install and run a pre-built web application and see how secure it is. The name of the lab should be a clue: *it's not*. And you're going to fix it.

Setup

Go to GitHub and fork the NodeGoat project so you can work on it. Clone your forked repo, **not the original or you can't submit your work!**

```
$ git clone https://github.com/YOUR-USERNAME/NodeGoat
$ cd NodeGoat
```

Now install MongoDB and get it running.

```
$ brew tap mongodb/brew
$ brew install mongodb-community
$ brew services start mongodb-community
```

Open the config/env/development.js file in your editor and uncomment line 8. This points the app to your local Mongo server.

```
  // remove the comment from this line:
  db: "mongodb://localhost:27017/nodegoat",
```

Finally, you'll run the following commands to install the necessary dependencies and get the project running. You'll need a current version of Node and npm, run brew install node if you don't.

```
$ npm install
$ npm run db:seed
$ npm start
```

You should now be able to see the app running on your machine at http://localhost:4000. You can login with any of the following credentials:

- **username : password**
- user1 : User1_123
- user2 : User2_123
- admin : Admin_123

Fix It!

Once it's running, log in as a user and as an admin and figure out how the app is supposed to work. Then take a few minutes to look through the code and see if you can get a basic understanding of what it's doing. Most of the JavaScript code you'll need to work with is contained in the app directory.

On the login page, there is a link to the tutorial. Read through each vulnerability and follow along with lecture slides to understand it. Watch the videos they provide for each about what an attacker can do and try them on your local version. Think like a hacker and try to break it!

Grading

Make a new new branch in your repo called owasp-fixes that will contain all of your changes. You should be committing frequently since this is a fragile app and you're actively trying to break it. When you're done, push the branch to GitHub and submit a PR to your fork, **not the real repo.**

Submit a link to your repo in Slack on the #cset-170 channel to help each other out and for me to grade the lab.