This project is about httpd.

If you did not load the http package when you loaded the system you can install it from `jaguar:/sdb/slack13.1/slackware/n/httpd-2.2.15-i486-1_slack13.0.txz`.

The web server should be fully configured. This project is about modifying the configuration and modifying Web access (security) to directories.

You can enable the webserver by `chmod a+x /etc/rc.d/rc.httpd` and rebooting, but I would recommand doing the `chmod` and starting by simply running that file with the `start` parameter.

Setup home pages for both bob and ftp. A simple home page is available in `~djv/index.html`. In bob's version of the home page change "CECS 476" to bob. In ftp's version of the home page change "CECS 476" to ftp. This allows you to distinguish them. Make sure you can access the home pages for both ftp and bob.

Now that you have http working, configure your http security as follows:
1) The machine home page `http:labxx.net.cecs.csulb.edu` should be available
2) The home page for bob `http:labxx.net.cecs.csulb.edu/~bob` should be available
3) The home page for ftp `http:labxx.net.cecs.csulb.edu/~ftp` should NOT be available

That is, even though ftp has a legal home page, you are declaring through the http security mechanisms that no one is allowed to access that home page.

To test you will need to start a browser on some machine. If you want, you may configure X on your machine for this project or use another machine in the lab whose administrator has X configured.

Note: do the security with the http mechanisms, do NOT do this by using `chmod` on the ftp files or directories to make them unreadable.

Submit: You had to change several things in the configuration files; report each of the changes you made. (Don't report the whole file, only the changes.)

4) Create a new directory and put a page into it. Call the directory `http:labxx.net.cecs.csulb.edu/~bob/limited`. A copy of bob's home page is fine. Add an `.htaccess` file to control access to this directory. Access should be controlled by password. Place the password file inside the `limited` directory. (This is not the most secure place to put it, it's just convenient.)

There should be two users who have access to the `limited` directory: `bob` using password `access31` and `joe` using password `access33`.

Note: `joe` deliberately is a http "user" only, he does not have a login account on your machine. Testing: make sure when you try to access the secure page, a password is required and the password works.

Submit: the contents of your `.htaccess` file. The exact form of the `htpasswd` command you used for `bob`.

5) Disallow access to all files ending in `.secret` Place in `~bob` a test file called `test.secret`. It should be readable by everyone (test using your account or confirm using `ls`). Make sure you cannot access that file from the web browser.

Submit: Report the change you made to the configuration files.

6) Enable cgi scripts in the user home directories. Make a copy of the script `~djv/msg.cgi` into the public portion (`public_html`) of bob's web directory. It's text, you can look at it to see what it should display. Make sure you can access the page.

Submit: Report the changes you made to the configuration files.

Leave the `httpd` server running so I can test your configurations.