1 Setting up SSH on a department machine

To hand in your assignments, you’ll be using the `cs125_handin` script, which you’d normally run on a Linux box. In CS125, you’ll be doing most (if not all) of your work on the Mac machines. Because life is already hard enough, you don’t want to have to log into a Linux machine just to hand in your work. Instead, you can set up SSH in a terminal, and run the handin script right from a Mac.

Starting Fall 2017, there’s a bit of a new (somewhat clunky) way to SSH from Mac to Linux, since the Linux department machines are on the new AD Kerberos authentication system. This is a two-part process: a one-type key setup on (any) Linux machine, and a command you run every time you log into a Mac.

1. Log in to a Linux in the MSLab. We need to first set up ssh as if the department machines were a personal machine.

2. Type `ssh-keygen -t rsa` and hit enter. You should be prompted for a filename to save the key; just hit enter to accept the default. Then, at the prompt, type a password and confirm it. In the terminal output you’ll see a bunch of nonsense about your new key.

3. Next, run `ssh-key-setup ~/.ssh/id_rsa.pub`. This will leave any existing ssh keys intact.

2 Using SSH on a Mac

1. Now, each time you want to use SSH from the Mac, open Terminal using Spotlight (cmd+spacebar) and type in "terminal."

2. Type `kdestroy` and press Enter. This will wipe out your Kerberos authentication cache temporarily for the current login session (not permanently). This is why you need to do these steps each time you log into a Mac and want to use SSH. See the end of this doc for more info on why this works.

3. Type `ssh cslab9f` or any machine and press Enter. You can guess different machines by naming convention or pick your favorite. Enter the password you entered for the SSH key above. If all goes well, you’ll see some terminal output, and finally, you’ll see a prompt that looks like `cslab9f ~$`. You have successfully logged into a Linux machine from your Mac!

3 Handing in from a department machine

1. In the terminal, change to your first assignments directory by typing `cd ~/course/cs125/model_progress` and hitting return. The `cd` command stands for change directory, and does exactly that; any command you run in the terminal will run with your current directory as its working directory.

   Now, type `touch REPORT.txt` and hit enter to create an empty file called REPORT.txt in your assignment directory. When you hand in an assignment, there MUST be a file called REPORT.txt in the assignment directory, or nothing will get handed in (don’t worry, the handin script will tell you if you’re missing it).

2. Normally you’d actually write up a short report in REPORT.txt, but for now, we just want to make sure the handin script is working. From the assignment directory, type `cs125_handin model_progress` and hit enter. The script will spit out the names of all the files it’s handing in (in this case it will just be ./ and ./REPORT.txt), and should end with Handin Successful! If not, let a TA know! You’ll get an email confirming that your handin was successful.

   You just handed in your first assignment completely blank! It’s alright though: every time you run the handin script, it overwrites your previous submission.
This is how you’ll go about handing in all your assignments. From now on, all you have to do is type ssh incoming in a terminal and enter your password to log into a Linux box. Then, enter the assignment directory (you’ll have to create the rest of these yourself) and run the cs125_handin script with the appropriate assignment name, and your files will be compressed and handed in.

4 Appendix: Why this works

*Courtesy of bnaacar*

The long explanation: if you run ssh with the -v option, you’ll see it trying several different authentication methods in succession. It tries Kerberos (GSSAPI) before it tries ssh keypairs.

Kerberos is what gets initialized whenever you log in in person. If you ssh into the department, Kerberos doesn’t get initialized unless you type `kinit`. Back when every machine was bound to the CS realm, you could log into any machine, Mac or Linux, and then ssh to any machine, Mac or Linux, and it would just work because of Kerberos.

The problem is that CS Kerberos and AD Kerberos don’t communicate properly, and the miscommunication isn’t symmetric. If you have active Kerberos credentials on an AD-bound machine and try to ssh to a CS-bound machine, Kerberos will fail and ssh will proceed to try ssh keypairs if you have them. But if you have active Kerberos credentials on a CS-bound machine and try to ssh to an AD-bound machine, Kerberos *starts* to work and do the handshake, and then it fails in such a way that ssh fails too and doesn’t try any further authentication methods. (You can see this happen with ssh -v.) The trick in this situation is to wipe out your active Kerberos cache before ssh’ing—then ssh will skip over Kerberos and try your keypair.

**Super Power User Tip:** Once you’ve done the steps in Using SSH on a Mac, you can even SSH from a Mac to a Mac! Type `kinit` and press Enter. Enter your login/AD/Banner password (your Brown Shibboleh password that you use for Banner. Not necessarily the password you set up in Part 1).