csce215 — UNIX/Linux Fundamentals
Fall 2021 — Remote Access to the Labs

The lab assignments for this course are meant to be completed using the computers in room 1d43. Being physically present in the lab is the recommended way to do this.

However, it also also possible to access those computers remotely, using a protocol called `ssh`, which stands for **secure shell**. The idea is to establish a connection from a remote computer, i.e. your laptop or other device, to one of the lab computers. Commands typed on your computer are sent across this connection to a shell process running on the lab computer, and responses are sent back to be displayed on your screen.

This document has some instructions for accessing the lab computers in this way, which may be useful if you would like to complete the lab assignments without returning physically to the lab, or if you would like to access your Linux account for other reasons. However, because it depends on many factors that are beyond our control, using this sort of access is *not* something that the TAs and I can provide support or troubleshooting for.

The explanation 'I had trouble with my SSH connection' will not be treated as an acceptable reason for any request to submit a lab assignment after the deadline. Please plan accordingly.

With that explanation and word of caution in place, there are three basic elements that you'll need to access the labs remotely.

Duo Two-Factor Authentication

The university requires two-factor authentication, using a tool called Duo, for remote access to many computing services, including `ssh` access to our Linux lab.

If you have not configured Duo before, these are the basic steps:

- Go to `https://myaccount.sc.edu` and log in using your USC Network ID and password.
- Choose 'Update Account Settings'.
- Answer and/or establish security questions.
- Go to the 'Multi-Factor' tab.
- Enter your phone number, select mobile or landline, choose your mobile OS if applicable, and click 'Submit'.

- A line entry for the phone will appear. Click 'Activate' and follow on screen instructions.

Connecting to UofSC VPN

Next, you'll need to configure and connect to the university's VPN service. Details of the process are here:

`https://scprod.service-now.com/kb_view.do?sysparm_article=KB0010878`

You'll need to download a VPN client from here:

`https://my.sc.edu/software/`

The client appears under:

Home → Security → Cisco AnyConnect VPN for Students - Personal Use

Download and install the application. Be sure to follow the instructions linked above. In particular, you should only install the Core and VPN packages; all other packages are unnecessary for this purpose. If you encounter any issues with this process you should reach out to the Division of Information Technology: `https://uts.sc.edu/`

You'll need to have this VPN client installed and activated for any of the commands below to work.

Establishing a secure shell connection

With the two previous steps complete, you should be able to connect to that lab computers using a program called an **SSH client**.

There's a decent chance that your computer has an SSH client installed by default. (For example, this seems to be the case with Macs. Some recent Windows versions also have an option to enable `ssh` easily.) If so, then using a command sort of like this, in whatever sort of terminal program is available on your computer, should work.

```
ssh -p222 user@L-1D43-00.cse.sc.edu
```

However, you should replace '`user`' with your username, and '`00`' with your favorite number between `01` and `36`. In this case, the `222` is the *port number* to use for the connection and `L-1D43-`xx`.cse.sc.edu` is the hostname of the specific lab computer you'd like

to connect to. (There's no reason to care about which lab computer you connect to. You'll see the same files in your home directory across all of them.)

A few details about this kind of remote access are here:

`https://cse.sc.edu/resources/cse-linux-workstations`

If your computer does not have an SSH client installed, you can install one. There are lots of options, both free and paid. For Windows, one popular option is the free version of MobaXterm. For Android, you might try Termux. Many of these are graphical clients, which means that instead of typing an `ssh` command, you'll enter the hostname and port number into a window and click a button to start the connection.

Securely copying files

Because the commands you execute in an SSH session are running on the remote lab computer, any files you create will be stored on that remote machine. So if, for example, you want to submit a file created on the lab machine to `dropbox.cse.sc.edu`, you'll first need to copy it from the lab machine to your computer.

If you are using a command-based SSH client, the command to do that is `scp`, which is short for 'secure copy'. Here's an example:

```
scp -P222 user@L-1D43-00.cse.sc.edu:~/lab1.cast .
```

Just like for the `ssh` command, you'll need to replace `user` with your username and `00` with the number of a computer (between 01 and 36) to connect to. In this example, `/lab1.cast` is the name of the file we want to copy from the lab computer, and `.` is the destination it should be copied to (i.e. the current directory). After issuing this command and giving your password, you will need to approve a DUO Push notification, which will be sent automatically.

Important: This command must be executed on your local computer, not from within an ssh connection to a lab computer.

If you are using a graphical SSH client, obviously it won't make sense to try to use the `scp` command. However, most graphical SSH clients include graphical tools for browsing and downloading files; you may see this labeled as 'scp' or 'sftp' or even something like 'remote files'. Check the documentation for your SSH client.