Using SSH

Learning Objectives

You will use Terminal to transfer a compressed file via SCP. This file contains a complete website. You will access the server via SSH and will decompress the file so that you are left with the working website.

****

User Account

The account that you will be using in this assignment is the same account that you will use throughout the rest of this program. It is for school use only! It is not to be used for storage! Only material for your current class should be in your account. There is a size limit as well as a file limit on your accounts. It is your responsibility to keep yourself under those limits.

****

Open Terminal

Open the Terminal application as outlined in the previous *Introduction to Terminal* tutorial.

Download the Working File

In your web browser download the file that you will be working with from:
http://66.192.104.111/NSS/files/chappalinos.tar.gz

When asked for a username and password use: *username*: **student** *password*: **?KNSS1**
(Note: if you use Safari make sure to go into your preferences and **uncheck** "Open 'safe' files after downloading." If you do not do this Safari will automatically strip off the .gz file extension and the rest of this activity will not work.)

Upload the Working File

Upload the file using the ***scp*** command to the server, replacing *username* with your username and *server* with the appropriate server address. In this case the username will be [*first initial*][*last name*] (all lower case) and the server will be 66.192.104.111. In Terminal navigate to the directory containing the file you just downloaded and then enter the scp command:

```
scp chappalinos.tar.gz username@server:~/
```

If you are asked whether you want to add the server to the list of known hosts, answer by typing **yes**.

SCP is used to securely transfer data from one computer to another. When using the ***scp*** command, the input file should come first followed by ***username@server:*** with the destination file or directory following directly after it. In this case, you are uploading the file chappalinos.tar.gz to your home directory, symbolized by the ~. When asked for a password use **?KNSS1**. Nothing will be displayed while you are typing your password. This is a security feature of Unix / Linux systems.

Connecting To the Server

Now that the file is uploaded, you have to connect to the server via SSH in order to manipulate it. To do so, run the following command from your local machine, replacing ***username*** with your username and ***server*** with the appropriate server.

```
ssh user@server
```

Once again, when prompted for a password use **?KNSS1**. When you have successfully connected, you will be prompted with your shell environment. This shell environment should function nearly identically to that of your local machine.

Moving the Working File

Now that you have uploaded the file and connected to the server, you now have to move the file to a place on the server that is publicly available via the HTTP server. To do so, move the file to the *public_html* directory located inside your home directory. Refer to the previous activity on Terminal commands for help on moving files in Terminal.

Untar the Working File

Now the files need to be extracted from the tar.gz file.

```
tar -xzpvf chappalinos.tar.gz
```

Perform a list (ls) and you'll see a new directory in public_html called chappalinos.

Fix the site

Now that you have decompressed the file you have a bunch of new files and directories. These files and directories make up a fully functional website, however right now many of them have the wrong permissions, wrong names, and may be in the wrong place. Using the Terminal commands you learned previously you now need to change permissions, move files, rename files, rename directories, and create directories so that you have an exact match of the site map on the following page.
It's best to use long-format listing for the rest of the activity so we know what the permissions are and so we can differentiate between files and directories.

Site Map **(the numbers are the permissions) This is what we want the files to look like.**

- ***~/public_html/chappalinos/*** (755)
    - contact.html (644)
    - **css**/ (755)
        - main.css (644)
    - gallery.html (644)
    - **images**/ (777)
        - bg_repeat.jpg (644)
        - bottom.jpg (644)
        - couple.jpg (644)
        - couple-tn.jpg (644)
        - family.jpg (644)
        - family-tn.jpg (644)
        - header.jpg (644)
        - kitchen.jpg (644)
        - kitchen-tn.jpg (644)
        - location.gif (644)
        - maria.jpg (644)
        - maria-tn.jpg (644)
        - nav.jpg (644)
        - pasta-dish1.jpg (644)
        - pasta-dish1-tn.jpg (644)
        - pasta-dish2.jpg (644)
        - pasta-dish2-tn.jpg (644)
        - rexalls.jpg (644)
        - top.jpg (644)
        - window.jpg (644)
        - window-tn.jpg (644)
    - index.html (644)
    - location.html (644)
    - menu.html (644)

First off, if you try to change directory into chappalinos you'll get a "permission denied" message. This is a big clue for what you need to do. Perform a long-format list and you'll see that the chappalinos directory has a file permission of rw------- (600), making it inaccessible. To get into a directory you need to have an execute permission. Looking at the site map we can see it's supposed to be 755 so let's change it.

```
chmod 755 chappalinos
```

Now go into the chappalinos directory and list the contents. We can see some files and directories. If you look at the site map you'll notice that we have several things in the wrong place and some things are named incorrectly. Everything here also has the wrong permission. We need to fix that first before we can fix anything else. Change all the permissions to what they should be according to the site map. Even if the name is wrong, change the permission first so that you can change the name to what it should be. Can't figure out what it's supposed to be? Just change it to 644 for now so you can work with it and figure it out. Remember directories need the execute permission set in order for you to access it.

We have a directory called "gallery" that is not on the site map. This directory is probably supposed to be named "images" So its permission should be 777. Now let's rename that directory.

```
mv gallery images
```

We can see a couple other files here that have the wrong name as well. indexx.html and contact.shtml should be index.html and contact.html so change those the same way you did gallery. Don't confuse the index.html file here with the index.html file up in the public_html directory. They are completely different. You'll learn more about those in your HTML classes.

The remaining image files are supposed to be in the images directory so go ahead and move them there.

```
mv bg_repeat.jpg images
mv bottom.jpg images
mv header.jpg images
```

…etc

Now go into images. We can see a file that doesn't belong here, gallery.html. It needs to be moved up a level.

```
mv gallery.html ..
```

Now that that's out of the way change the permissions for the remaining files.

Now go back out to the main chappalinos directory. We have a folder called "location" so let's go see what's in it. location.gif and location.html. Set their permissions and put them in the right place.

Back out and go into the menu directory. Change the permission on menu.html and move it to where it should be.
Back out again and you can see we're almost done. We seem to be missing a directory called css. Create it and set its permission.

```
mkdir css
```

Now move main.css into the newly created css folder. We have the location and menu folders that have nothing in them now and they are not on the site map so go ahead and delete them. Remember you have to use the recursive option to delete directories.

```
rm –R location menu
```

That's all there is to it!

Confirm Completion and Delete Working File

To confirm that you can access the site, you can go the following URL, replacing ***username*** with your username, in your web browser.

http://66.192.104.111/~username/chappalinos

Change your password

This user account is one you'll be using throughout the rest of the degree program. It's a good idea to change your password for security. **If you do not change your password you will lose points on this assignment and all other assignments using SSH including the final exam.** To do this, use the passwd command. (Do not include the brackets.)

```
passwd [username]
```

You'll then be asked to enter the current password, which is the same one you logged in with. You'll see something similar to the following.

```
[username]@Alfalfa:~$ passwd [username]
Changing password for [username].
(current) UNIX password:
```

Once you type in the current password and hit return it'll ask you to enter a new password. It'll have you type it again to confirm and then if you've done it correctly it will say "password updated successfully." If you don't see "password updated successfully" then the password was not changed.

DO NOT forget your password as you will be using this account again on the final exam and throughout the degree program.

Export Terminal Output

Export your Terminal session to a text file by clicking on Shell > Export Text As… and name the file [*last name*]_[*first initial*]_SSH and submit it on the dropbox.

This assignment is exactly like what you'll have to do on the final exam. You won't need to understand HTML or CSS to accomplish the task (I don't even know what all these files do) but know that .html files are the pages that are displayed when someone visits the site in a web browser. The index.html file is the default page that is displayed when someone visits the site. It's usually a splash page or main page of the site. The .html files link to the other resources, such as .jpg file, .mp3 files, .mov files, etc, so the user can access them. That should get you on the right path. Let me know now if you're confused by anything here. Don't wait until the final.