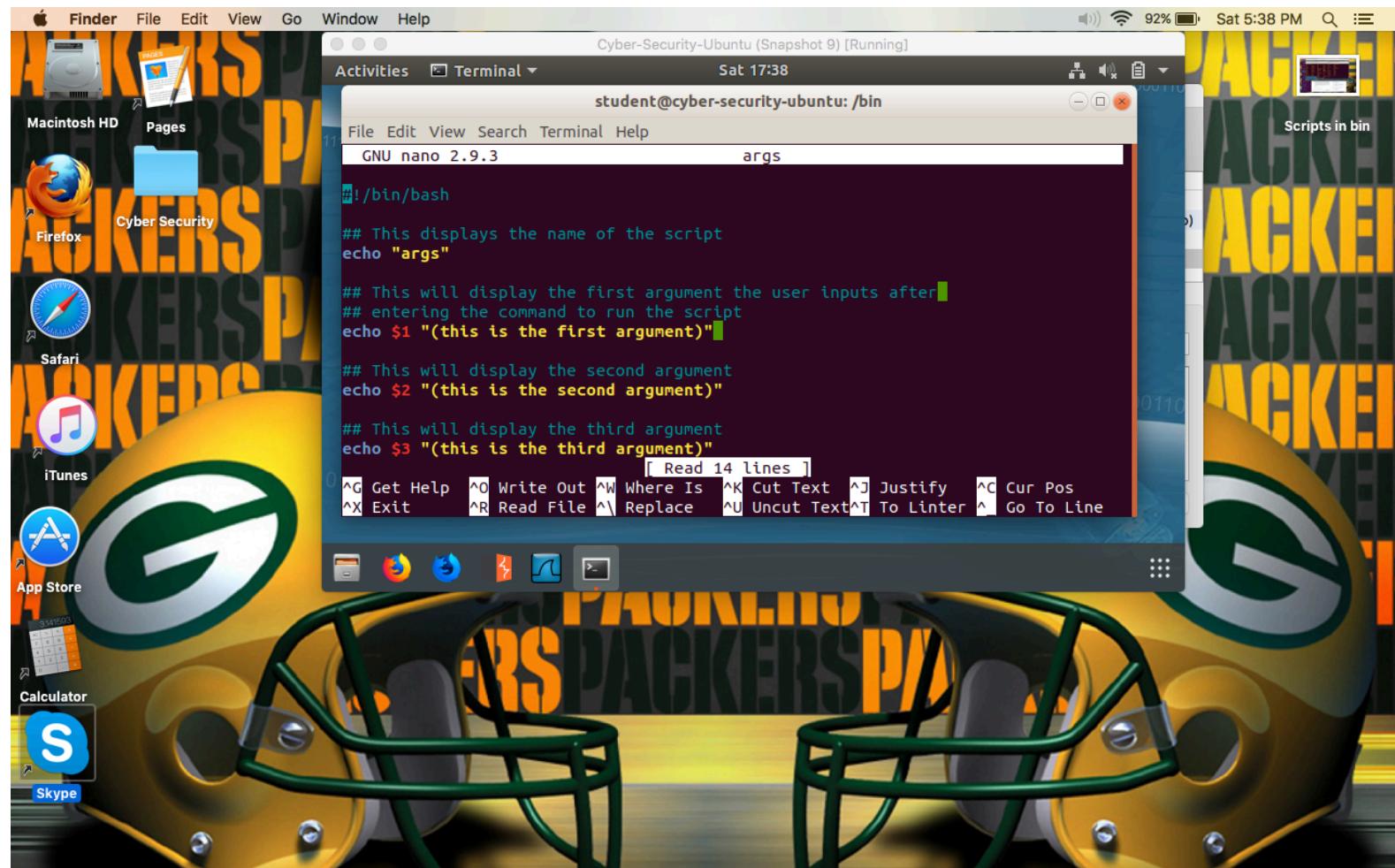


Here within the /bin/ folder, are the scripts that were created for this week's homework assignment as executable commands. In the left terminal window, at the top of the first column you can see the args script, as a command, and in the middle column at the bottom you can see the mkdircd script, also as a command. In the right terminal window, in the third column, in the fifth line you can see the ping\_sweep script, also as a command.

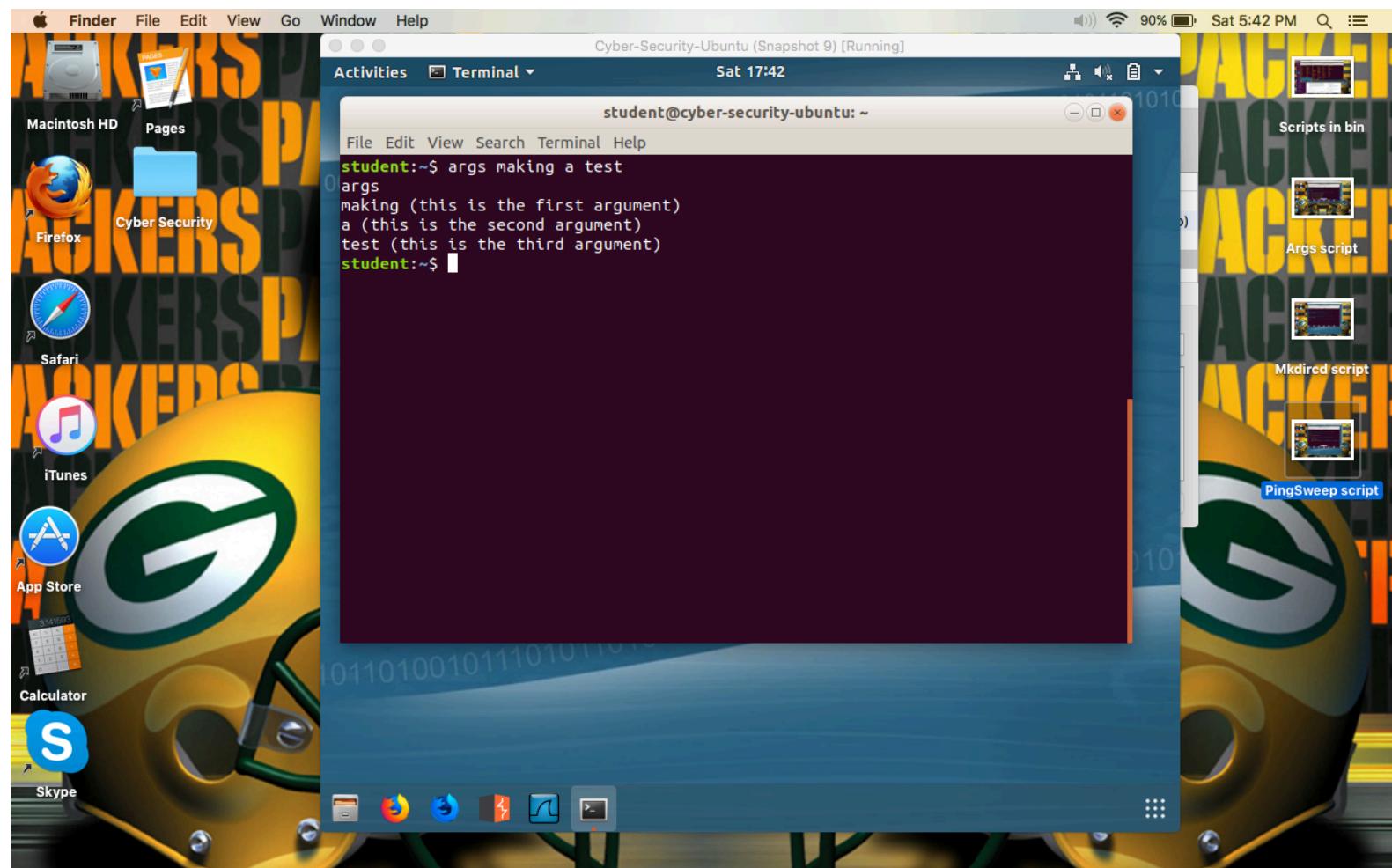
The screenshot shows a Mac OS X desktop environment. On the left, there is a Dock with icons for Finder, Safari, Mail, iCal, Address Book, System Preferences, and several application icons. A Finder window titled "Cyber-Security-Ubuntu (Snapshot 9) [Running]" is open, displaying a list of files and folders. In the center, there are two terminal windows. The left terminal window is titled "student" and shows the command "student:/bin\$ ls". The right terminal window is also titled "student" and shows the command "student@cyber-security-ubuntu:/bin\$". A context menu is open over the right terminal window, with options like "Show", "Reset", and "Take". The background features a "HACKER" theme wallpaper.

```
student:/bin$ ls
args      fuser      netwochvt    pidof      udevadm
bunzip2   fusermount  nisocp       login     unlockmgr_server
bzcat     grep       ntfs-dash    lowntfs-3g ping
bzcmp    gunzip     ntfsdate    lsblk      ping4
bzdiff   gzexe     ntfsdd      lsmod      ping4
bzegrep  hciconfig  ntfsdf     mkdircd   ping6
bzexe    hostname   ntfsdmesg  mknode    readlink
bzfgrep  journalctl ntfsdomainname mount   red
bzgrep   kbd_mode   ntfsdumpkeys more    rm
bzip2    kill       ntfsrecho   mountpoint run-parts
bzless   less       ntfsfbootdump mt      rmdir
bzmore   lessfile   ntfsfbootmgr  gnu      sed
cat      lesskey    nc          mv      setfacl
chacl   lesspipe   open false   nc      setfont
chgrp   chmod      nc.openbsd  sh.distrib zgrep
chmod   chown     pidoffgrep  netcat   sleep
loadkeys  login     ping_findmnt netstat  ss
loadkeys  ping4     ping_sweep  uname
login    ping6      ping_sweep  uncompress
ping4   ping6      ping_sweep  unicode_start
ping6   ping6      ping_sweep  vdir
ping6   ping6      ping_sweep  wdctl
ping6   ping6      ping_sweep  which
ping6   ping6      ping_sweep  whiptail
```

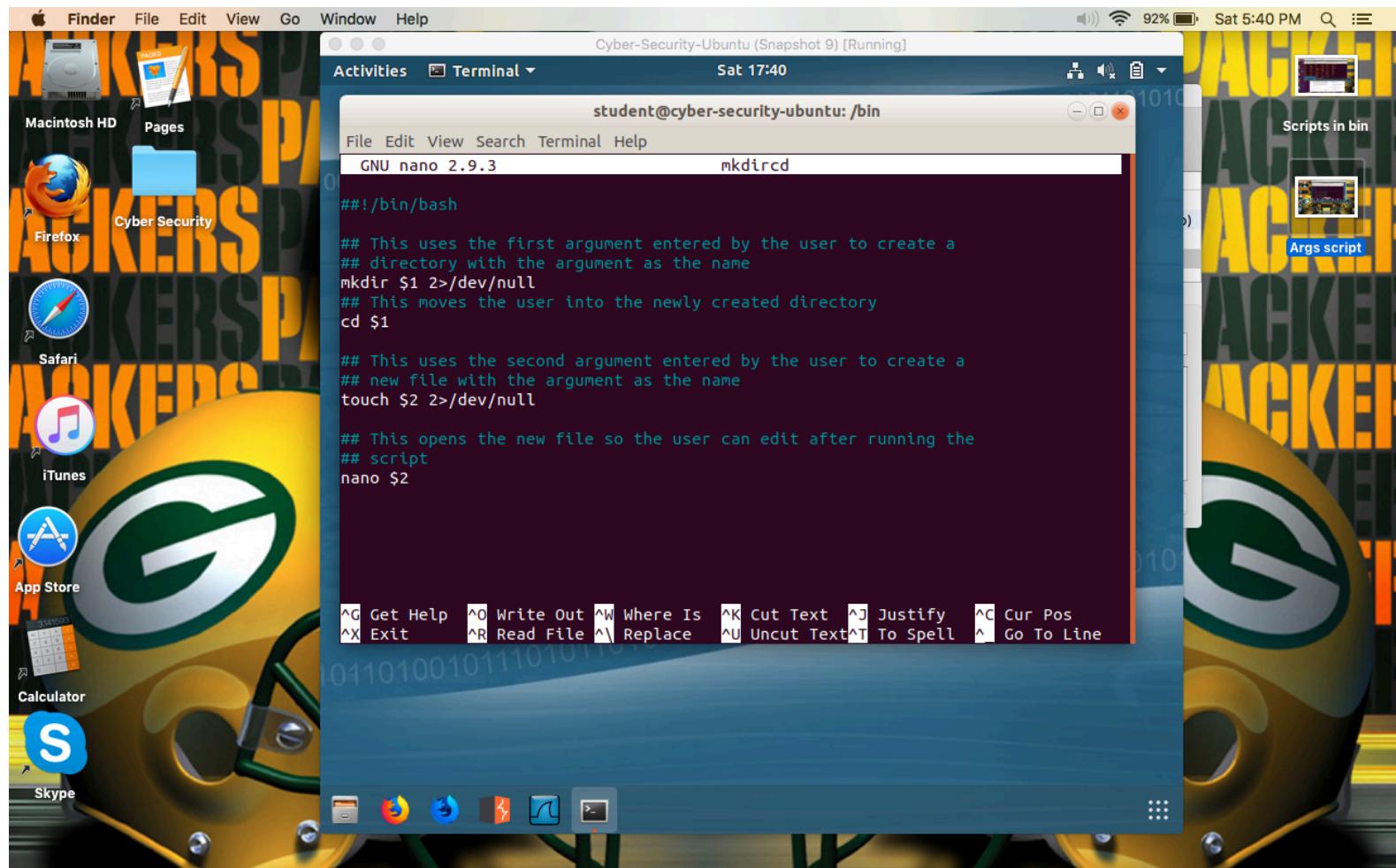
Here I am showing the command for the args.sh file. First I am displaying the name of the command "args". In the following 3 lines I am using "echo \$1, echo \$2, echo \$3" to display the 3 positional arguments the user enters after running the command args.



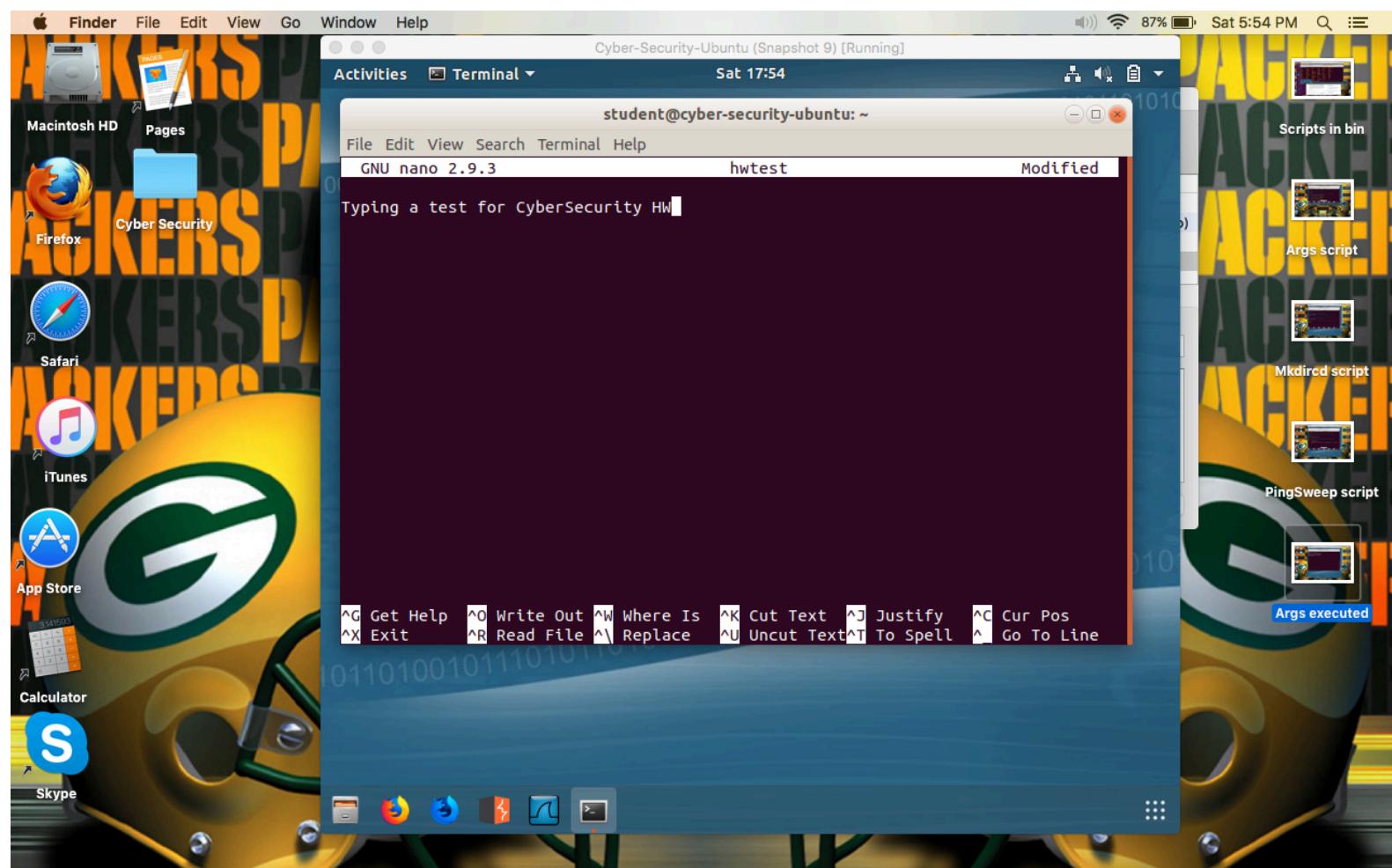
Here I show running the command within the user "student" home folder. "Making a test" are displayed after the command name as separate lines in the display.



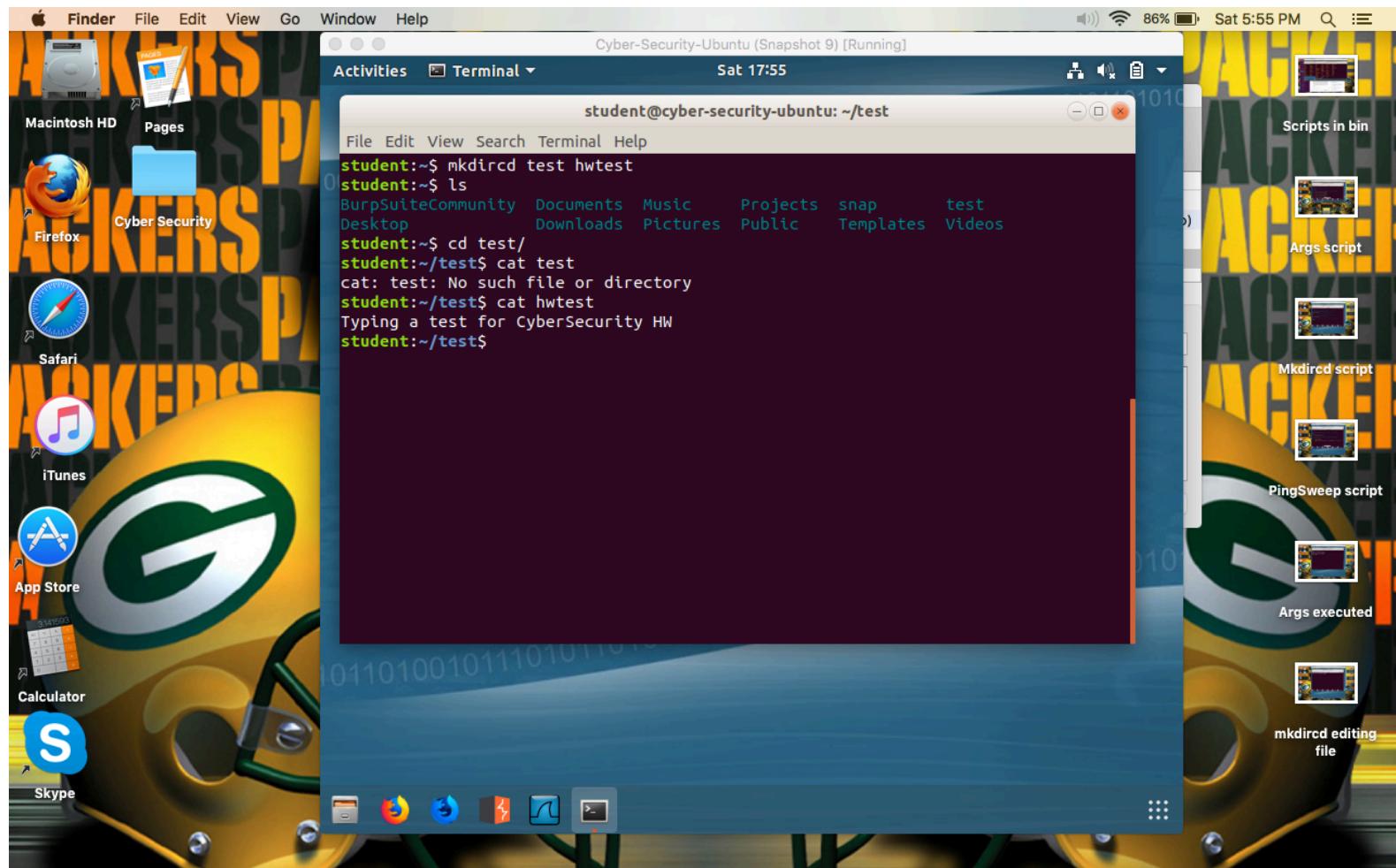
Here I am showing the script for the mkdircd command. It is taking the first and second positional argument and utilizing them to create a new directory, the first argument, and a new file, the second argument. I include the 2>/dev/null line to send any errors to the to the /dev/null file. If there is no second argument used in the command the user will still create a directory and a file will be made with the user being prompted to name the file when exiting and saving.



Here I am showing the editing of the newly made file.



Here I am showing the new folder added to the user "student" home directory. I move into the folder and use a cat command to show the text from the earlier file.



Here I am showing the script for the ping\_sweep command. I am creating the variable PREFIX to use the user's first argument after running the command, in this case would be an IP address. Next I echo the scanning message. Finally I create a for loop so that IP address that the user enters will iterate through .1 to .255 and ping each address. The prefix variable from earlier will be combined with a seq variable ranging from 1 to 255 to create the IP addresses that will be pinged. I utilize ">/dev/null" to send any information from ping command to the /dev/null file. If the address is live and up the message "\$TARGET is up" will go to file called live\_hosts but if the IP address is down the message "\$TARGET is down" will go to a file called down\_hosts.

The screenshot shows a Mac desktop environment. On the left, there is a Dock with various application icons: Finder, File, Edit, View, Go, Window, Help; Macintosh HD; Pages; Cyber Security; Firefox; Safari; iTunes; App Store; Calculator; and Skype. In the center, a terminal window is open in the Activities overview. The title bar says "Cyber-Security-Ubuntu (Snapshot 9) [Running]" and the status bar shows "Sat 17:41". The terminal window has a dark background and contains the following script:

```
student@cyber-security-ubuntu:/bin
File Edit View Search Terminal Help
GNU nano 2.9.3          ping sweep
#!/bin/bash

## setting the positional argument for the user's input in use
## within the following for loop
PREFIX=$1

echo "Scanning $PREFIX .0/24..."

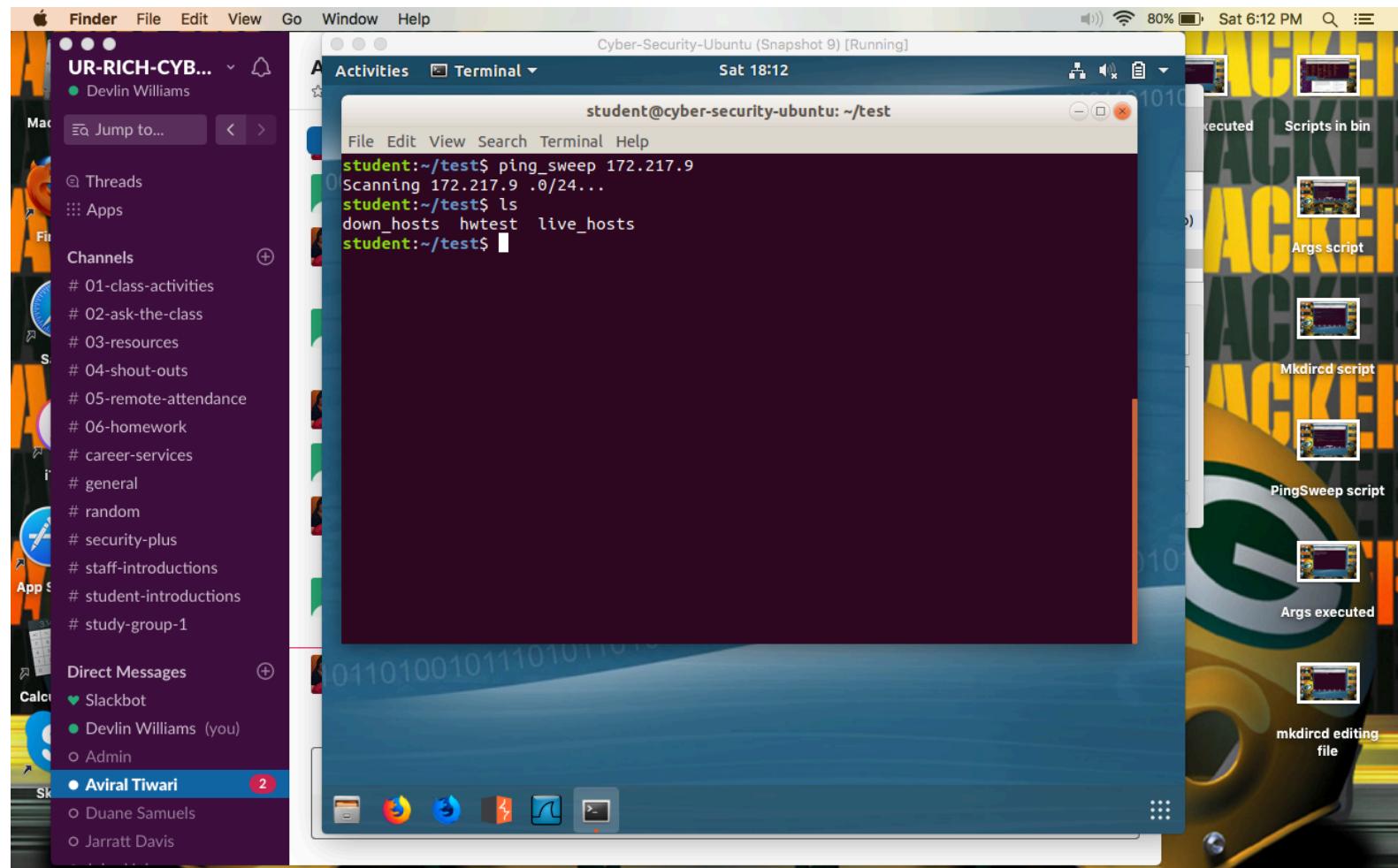
## Using a for loop to cycle through the ending of any IP address
## to send the live IPs to live_hosts and down IPs to down_hosts
for x in $(seq 1 255)
do
    TARGET="$PREFIX.$x"
    ping $TARGET -c 1 &>/dev/null && echo "$TARGET is up" >> live_hosts || $done
```

At the bottom of the terminal window, there is a menu of keyboard shortcuts:

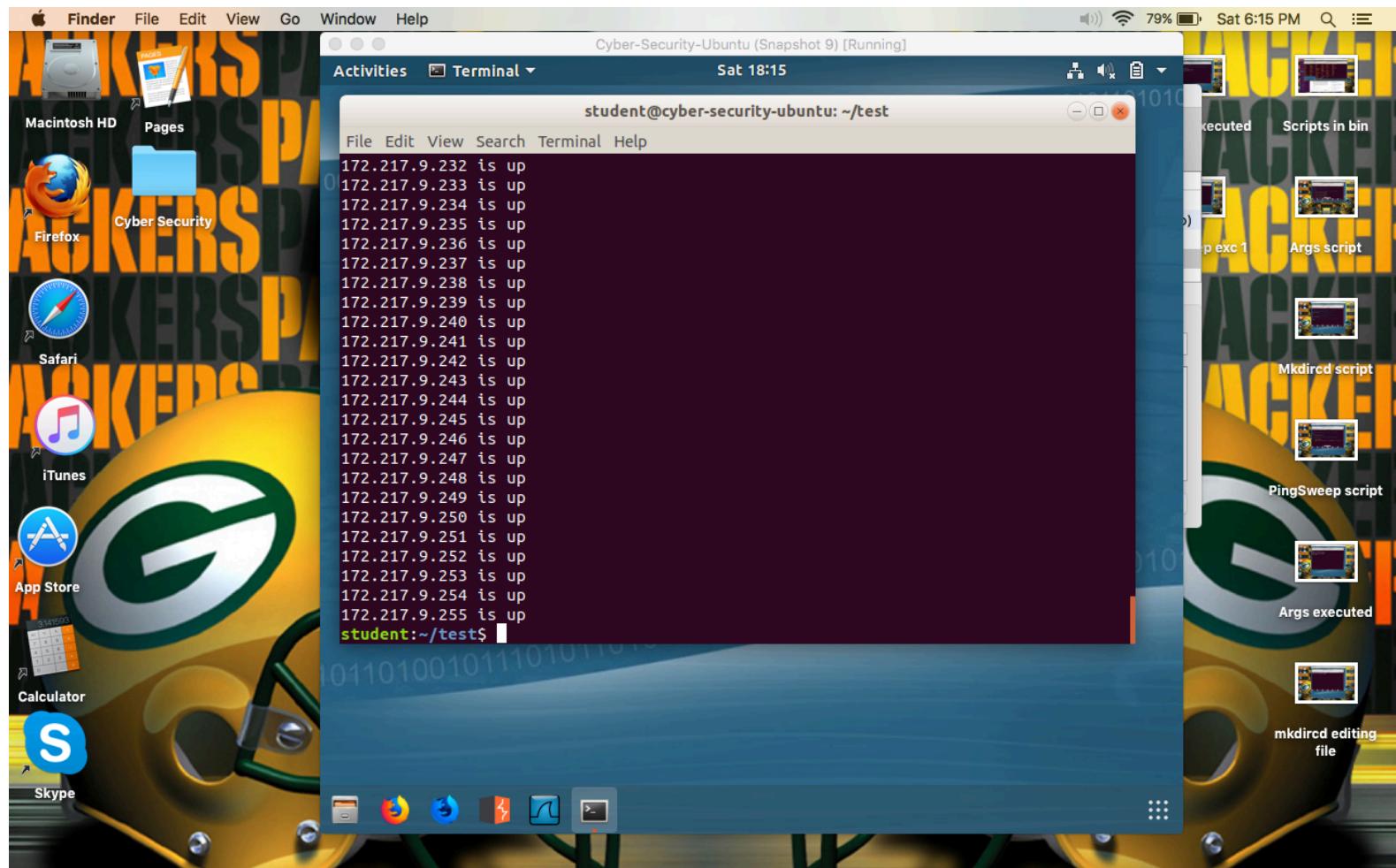
- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^C Cur Pos
- ^X Exit
- ^R Read File
- ^L Replace
- ^U Uncut Text
- ^T To Linter
- ^A Go To Line

[ Read 16 lines ]

Here I am showing the two files, down\_hosts and live\_hosts, being created in the user "student" home directory within the test subdirectory from the mkdircd command.



Here I am showing some of the IP addresses withing the live\_hosts file.



Here I am showing some of the IP addresses that were in the down file.

