**CIS-449 Intro to Software Security**

**With Professor Dr. Anys Bacha**

**Assignment 1**

**Student: Demetrius Johnson**

**02 February 2023**

**Due: 08 February 2023 at 9am**

Table of Contents

[Abstract 4](#_Toc126707439)

[TASK 1: Enabling and connecting to virtual machine over SSH and SFTP 5](#_Toc126707440)

[SSH via PuTTY 5](#_Toc126707441)

[Background 5](#_Toc126707442)

[How to connect via PuTTY 9](#_Toc126707443)

[How to connect via Windows CMD 10](#_Toc126707444)

[SFTP via winSCP 11](#_Toc126707445)

[TASK 2: Prepare your home directory 12](#_Toc126707446)

[TASK 3: Linux Bash Shell commands 1 13](#_Toc126707447)

[1. List the contents of the "hw1.1" directory using the **"ls"** command. You must include the full command in your report and include a screenshot capturing the first page of the output. The output must show: 13](#_Toc126707448)

[2. Run the following command "cat /usr/share/dict/words | grep -i hello > /tmp/words.log" 16](#_Toc126707449)

[3. Copy the file numbers.txt from the current directory to the java subdirectory. Include the command in your report. 16](#_Toc126707450)

[4. Rename the file Burrot.java to Borat.java (renaming is done using the same command as moving). Include the command in your report. 17](#_Toc126707451)

[5. Delete the files diff.html and diff.css. This must be done with a single command and not multiple commands. Include the command in your report. 18](#_Toc126707452)

[6. List all web page files (files whose names end with the extension .html or .css) in the current directory. Note that the ls command can accept parameter(s) for what files you want it to list. You must include the full command in your report. 18](#_Toc126707453)

[7. Copy all the text files (files whose names end with .txt) from the current folder to the website subdirectory. Include the command in your report. 18](#_Toc126707454)

[8. Display the contents of all files whose names begin with verse and end with the extension .txt, such as verse1.txt and verse2.txt . (Write a single command that displays all their contents concatenated.) You must include the full command in your report. 19](#_Toc126707455)

[TASK 4: Linux Bash Shell commands 2 20](#_Toc126707456)

[1. The file animals2.txt contains an alphabetized list of animal names. It includes many duplicates. Output the first 16 distinct animals from the file, one per line. (The last one should be adlie penguin .) You must include the full command in your report and include a screenshot capturing the output. 20](#_Toc126707457)

[2. Combine the contents of files verse1.txt, verse2.txt, and verse3.txt into a new file lyrics.txt. Include the full command in your report. 21](#_Toc126707458)

[3. Display all lines from animals.txt that contain the word "growl" ignoring case, in reverse-ABC-sorted order and with no duplicates. Output the lines themselves only. Include the full command in your report. 22](#_Toc126707459)

[TASK 5: Linux Bash Shell commands 3 23](#_Toc126707460)

[1. Set the file example1.txt in the current directory so that its group and other can write to the file. (You don't need to change any other permissions that might currently be set on the file.). Include the full command in your report. 23](#_Toc126707461)

[2. Set all files with extensions .dat and .doc to be read/writable (but not executable) by their owner, and to have no access from others. (Your command should grant these exact permissions and revoke any others.) Do this using the standard letter code arguments for granting and removing permissions. Include the full command in your report. 24](#_Toc126707462)

[3. The file /etc/passwd stores a list of all users' names and user account names on the system, along with a bit of other information such as what shell program they use. (The default shell for most users, and the one we have been learning about in this course is the Bash shell, stored in the file /bin/bash ) **How many users exist on this Linux system that use the Bash shell by default?** 25](#_Toc126707463)

[4. Create a file foo.txt using the “touch” command. Change the file's last-modified date to be January 4 of this year at 8:56am. Include the full command in your report. 26](#_Toc126707464)

[Conclusion 27](#_Toc126707465)

[From README.TXT FILE 27](#_Toc126707466)

[1) To compress the assg1\_complete.tar.gz file: 27](#_Toc126707467)

[2) To decompress (extract) the file contents of my assg1\_complete.tar.gz file: 28](#_Toc126707468)

# Abstract

In this lab, we explore the use of various Linux commands when running a Bash shell; to do this, we use a virtual machine that is running Linux Ubuntu 20.04, using VirtualBox Hypervisor to run and manage the machine. After, we log into the machine over SSH and SFTP in order to transfer files and interface with the operating system of the virtual machine without needing to use the GUI of the virtual machine, which is much slower. Some of the primary commands we run to learn Bash syntax are **grep, cp, ls, mv, rm, touch,** and **cd.** We also work with flags/options of the functions to gain more control from the power of the functions, as well as some of the syntax of regular expressions such as **\*, [],** or **^.** Lastly, we used some of the bash operators such as pipe **|,** and redirect **>** in order to redirect output of functions/commands to another function or file, respectively. Over the course of this lab, you will learn to navigate the Linux system, and how to connect to virtual machines using other interface methods.

# TASK 1: Enabling and connecting to virtual machine over SSH and SFTP

## SSH via PuTTY

### Background

I will write some notes here of my solution, and why the solution works, as a I discovered/researched according to my knowledge about computer networks:

1. FIRST OF ALL MAKE SURE THE VIRTUAL MACHINE IS POWERED OFF BEFORE TOGGLING THE NECESSARY SETTINGS!
   1. Select your machine to configure
   2. Then click “settings”, as shown below:

Graphical user interface, text, application

Description automatically generated

1. Now go to the “Network” tab and make sure the following settings are configured as shown below (circled in red):
   1. Then, click “Advanced”, then “Port Forwarding” as highlighted below:

Graphical user interface, text, application

Description automatically generated

1. Now, add a name (optional), make the protocol TCP (this can also work with UDP, but TCP will make connection reliable), set **Host Port to a value such as 2281**, and the **Guest Port to 22** (default ssh port number = 22, but it does not have to be the port used).

Graphical user interface, text, application

Description automatically generated

* 1. We leave the Host and Guest IP blank, and in doing so all IP traffic on the Host machine that is tagged to port 2281 will be forwarded to a default IP address and port number of the virtual machine 🡪 VirtualBox configuration sets IP address of virtual machines to 10.0.x.15, where x depends on which interface id + 2 the machine will use for network connectivity.
  2. Now, according to [How to configure port forwarding in VirtualBox for NAT Networking | LaptrinhX](https://laptrinhx.com/how-to-configure-port-forwarding-in-virtualbox-for-nat-networking-2625274867/#:~:text=Configure%20port%20forwarding%20for%20NAT%20using%20GUI%201,forwarding%20rule%20for%20the%20respective%20vm%20More%20items), Virtual Box uses NAT Networking to configure the networking/network adapter of the virtual machine.
  3. The virtual machine is a on a separate private network than the local host machine.
  4. So, we must request to connect to the loopback address of the host machine (127.0.0.1) on port 2281. Even though every machine that runs IP has a loopback address of 127.0.0.1, and the host and virtual machine are technically the same hardware, VirtualBox keeps loopback traffic of the local host separate from the virtual machine loopback traffic.
  5. VirtualBox will then map 127.0.0.1:2281 of local host traffic to be 10.0.x.15:23 and forward it to the virtual machine which is on the 10.0.2.1 network (thus acting as a router running NAT = network address translation). The default gateway of the virtual machine is 10.0.2.2, and thus when I ran a ss -atp to check TCP traffic on the virtual machine, you will see a connection is established on the virtual machine on local address 10.0.2.15, to peer address 10.0.2.2.
     + Graphical user interface

       Description automatically generated with medium confidence
  6. And when I ran netstat command on my local Windows machine, notice port 2281 on local machine (denoted by 0.0.0.0, and 127.0.0.1).

A picture containing text

Description automatically generated

### How to connect via PuTTY

1. Open PuTTY
2. Enter [username@127.0.0.1](mailto:username@127.0.0.1); username is seed thus do [seed@127.0.0.1](mailto:seed@127.0.0.1)
3. Set port number to be as Host Port that you configured in previous step; in this case it is **2281** as following with my example.
4. Thus you are connecting the Host machine to itself over SSH on port 2281, which virtualbox will see that IP traffic and forward it to the virtual machine on port 22 (and also changing the destination IP to 10.0.x.15).

Graphical user interface, application

Description automatically generated

1. Now you just type in the password and you are connected:

Text

Description automatically generated

### How to connect via Windows CMD

([ssh(1) - OpenBSD manual pages](https://man.openbsd.org/ssh.1))





Text

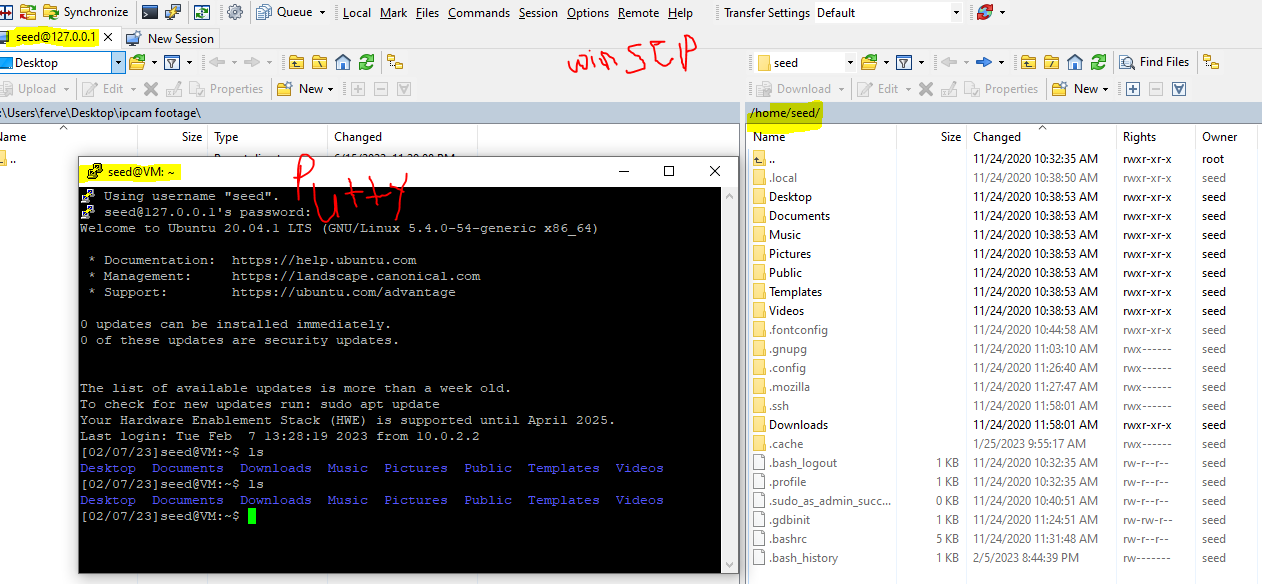
Description automatically generated

## SFTP via winSCP

Do the same steps as for SSH:

Graphical user interface, application

Description automatically generated



# TASK 2: Prepare your home directory

I used WinSCP file transfer via SFTP client to transfer the tar ball files from my local machine to the VM, inside the home/seed/Documents directory:

Chart

Description automatically generated with medium confidence

Now I navigate to the home/seed/Documents directory of the VM in PuTTY so that I can run “tar -xvf” command to decompress the tar ball files:

-Going to the Documents Directory and displaying using ls:

Text

Description automatically generated

-Now here is the output after running tar -xvf <tar file> 🡪-x means extract, -v means verbose (show messages during extraction), and -f means archive file expected:

A screenshot of a computer

Description automatically generated with medium confidence

-Now showing I have decompressed all tar ball files and they are in the Documents Directory:

Graphical user interface, text, application

Description automatically generated

# TASK 3: Linux Bash Shell commands 1

*For each of the numbered items below, determine a single bash shell statement that will perform the operation(s) requested. Each of your solutions must be a single one-line shell statement and should not use Linux’s multi-statement joining operators such as |, &&, ||, and ; unless told otherwise. You can use the following references:*

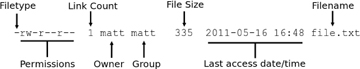
The numbered statements must be done from within the “/home/seed/Documents/hw1.1” directory. As such, you must use the “cd” command to change into the appropriate directory:

Text

Description automatically generated

## 1. List the contents of the "hw1.1" directory using the **"ls"** command. You must include the full command in your report and include a screenshot capturing the first page of the output. The output must show:

A long listing of all files and directories within hw1

* -l option
* 

Any hidden files/directories

* -a option

All files and directories within hw1 listed in reverse chronological order (oldest file/directory at the top and newest file/directory at the bottom).

* -R option for recursively listing all files and directories **in current directory** **and subdirectories**
* -t option: sort by modification time, newest first
* -r option: reverse order while sorting

Readable sizes of all files and directories

* -s (list size of files **in *blocks*, UNLESS -h option is included**)
* -h option: with -l and -s, print **sizes** like 1K 234M 2G etc.

Here is the output without the **less** function:

* Text

  Description automatically generated
* Graphical user interface, text

  Description automatically generated

The output must be piped to “less” to show the first page.

* Thus we need to do the following command and pipe it to **less** function:
  + ls -laRtrsh | less
  + Here is the output when we pipe to **less:**
  + A screenshot of a computer

    Description automatically generated with medium confidence
  + Note that page is relative to window size; **less** function will go to next page, up and down arrows moves to next line; press ‘q’ to quit (as I learned when using the function).

## 2. Run the following command "cat /usr/share/dict/words | grep -i hello > /tmp/words.log"

*Write a paragraph explaining what the command does:*

* The **cat** command is reading from a file on the path ***/usr/share/dict/***words, where **words** is the file.
* Then, instead of the standard output, the output from **cat** is redirected (via the pipe operator **|** )**,** to the **grep** function to be used as an input.
* Then, **grep** function (global regular expression search function) with -i parameter specifies to search for a value and ignore the case (upper or lower, i.e. e == E).
* Thus, **grep -i hello**  searches for string “hello” (from **words** file) and outputs matches regardless of case (i.e. not case-sensitive; HELlo == hello).
* Finally, redirect operator **>** sends the output of **grep** to a file called **words.log** from the directory ***/temp***
  + Text

    Description automatically generated

## 3. Copy the file numbers.txt from the current directory to the java subdirectory. Include the command in your report.

Showing contents of hw1.1 (java directory and numbers.txt directories are present)

Text

Description automatically generated

Before running command: showing contents of java directory:

Graphical user interface, text, chat or text message

Description automatically generated

Now run this command to copy **numbers.txt** to the **java** subdirectory:

**cp numbers.txt ./java**

Graphical user interface, text, application

Description automatically generated

## 4. Rename the file Burrot.java to Borat.java (renaming is done using the same command as moving). Include the command in your report.

* Use mv command
  + mv - move (rename) files
  + Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.
* Showing output before changing Burrot.java:
* Text

  Description automatically generated
* Now run this command to change the name: **mv Burrot.java Borat.java**
* Notice file contents are the same as before, showing that the file is the same and it only has a different name:
* Text

  Description automatically generated

## 5. Delete the files diff.html and diff.css. This must be done with a single command and not multiple commands. Include the command in your report.

Before deletion of files:

Graphical user interface, text, application, chat or text message

Description automatically generated

After running command to delete files: **rm diff.css diff.html**

Text

Description automatically generated

## 6. List all web page files (files whose names end with the extension .html or .css) in the current directory. Note that the ls command can accept parameter(s) for what files you want it to list. You must include the full command in your report.

You can use a \* (asterisk) as a "wild-card" character to specify a group of files. For example, \*foo means all files whose names end with foo , and foo\* means all files whose names begin with foo . You can use a wildcard in the middle of a file name, such as foo\*bar for all files that start with foo and end with bar .

Run this command: **ls -l \*.html \*.css**

Text

Description automatically generated

## 7. Copy all the text files (files whose names end with .txt) from the current folder to the website subdirectory. Include the command in your report.

The **website** directory before running the copy command:

Graphical user interface

Description automatically generated

Now run **cp ./\*.txt ./website** (note ./ means check current directory, \*.txt means check for any instances where file ends in .txt

Graphical user interface

Description automatically generated

## 8. Display the contents of all files whose names begin with verse and end with the extension .txt, such as verse1.txt and verse2.txt . (Write a single command that displays all their contents concatenated.) You must include the full command in your report.

Run the command: **cat verse\*txt** (this means print to standard output the contents of any file that begin with **verse** and end with **txt**):

Text

Description automatically generated

# TASK 4: Linux Bash Shell commands 2

*For each item below, determine a single bash shell statement that will perform the operation(s) requested. Each solution must be a one-line shell statement, but you may use input/output redirection operators such as >, <, and |.*

The number statements must be done from within the “/home/seed/Documents/hw1.2” directory. As such, you must use the “cd” command to change into the appropriate directory:

A screenshot of a computer

Description automatically generated with medium confidence

## 1. The file animals2.txt contains an alphabetized list of animal names. It includes many duplicates. Output the first 16 distinct animals from the file, one per line. (The last one should be adlie penguin .) You must include the full command in your report and include a screenshot capturing the output.

* First, we need to get all of the contents of the animals2.txt file using **cat** command
* Then, we need to redirect the output from **cat** to **sort** function
  + Then we need to use options for **sort:**
    - -u option: deletes all duplicates of sorted lines
* Then, we redirect **sort** output to **grep** and use options:
  + -m <number> option: only output the first m matches
  + ^ denotes that grep will search for all lines starting with any value
* Now if we combine everything together we do:
  + **cat animals2.txt | sort -u | grep -m 16 ^**

Text

Description automatically generated

## 2. Combine the contents of files verse1.txt, verse2.txt, and verse3.txt into a new file lyrics.txt. Include the full command in your report.

The hw1.2 directory before running the command:

A screenshot of a computer

Description automatically generated

Run this command: **cat verse\*.txt > lyrics.txt** (cat will output all contents of files that start with **verse** and end with **.txt**, the then *>* will redirect contents to a file called lyrics.txt, which was not yet created, thus it will be created); also note lyrics.txt is white because it does not have execute permissions:

A screenshot of a computer

Description automatically generated with medium confidence

Now here are the contents of lyrics.txt:

Text

Description automatically generated

## 3. Display all lines from animals.txt that contain the word "growl" ignoring case, in reverse-ABC-sorted order and with no duplicates. Output the lines themselves only. Include the full command in your report.

* First, we need to get all of the contents of the **animals.txt** file using **cat** command
* Then, we need to redirect the output from **cat** to **sort** function
  + Then we need to use options for **sort:**
    - -u option: deletes all duplicates of sorted lines
* Then, we redirect **sort** output to **grep** and use options:
  + -i option: ignores case (i.e., e == E)
  + -n option: output line numbers
  + and then we need to search for string “**growl**”
* Finally, we need to redirect output from **grep** to another instance of **sort** function with -r option in order to reverse the sort.
* Now if we combine everything together we do:
* **cat animals.txt | sort -u | grep -in growl | sort -r**

Text

Description automatically generated

I redirected it one more time to grep command just for sake of color:

A screenshot of a computer

Description automatically generated with medium confidence

# TASK 5: Linux Bash Shell commands 3

The number statements must be done from within the “/home/seed/Documents/hw1.3” directory. As such, you must use the “cd” command to change into the appropriate directory:

A screenshot of a computer

Description automatically generated

## 1. Set the file example1.txt in the current directory so that its group and other can write to the file. (You don't need to change any other permissions that might currently be set on the file.). Include the full command in your report.

Here are the original permissions of the file:

Text

Description automatically generated

Now run the command: **chmod go+w example1.txt** (where **go** refers to letters g and o for *groups* and *others*, and +w means “add write permissions”)

Text

Description automatically generated with low confidence

## 2. Set all files with extensions .dat and .doc to be read/writable (but not executable) by their owner, and to have no access from others. (Your command should grant these exact permissions and revoke any others.) Do this using the standard letter code arguments for granting and removing permissions. Include the full command in your report.

Before changing permissions:

Calendar

Description automatically generated

Run the command: **chmod u+rw-x,go-rwx \*.dat \*.doc** (this adds rw but no x for user/owner, then no rwx for g and o, and do this for all files ending with .dat and .doc)

Now I will output contents of hw1.3 directory of only .dat and .doc files, showing the permissions:

Text

Description automatically generated with medium confidence

## 3. The file /etc/passwd stores a list of all users' names and user account names on the system, along with a bit of other information such as what shell program they use. (The default shell for most users, and the one we have been learning about in this course is the Bash shell, stored in the file /bin/bash ) **How many users exist on this Linux system that use the Bash shell by default?**

(Hint: To figure this out, you will need to search for lines in the passwd file that mention bash.)

You may assume that no line of /etc/passwd contains the phrase "/bash" other than to specify the Bash shell). Include the full command in your report.

The following information I got from <https://www.ibm.com/docs/bg/aix/7.2?topic=passwords-using-etcpasswd-file> to help me use for referencing what each field means:

*Traditionally, the /etc/passwd file is used to keep track of every registered user that has access to a system.*

*The /etc/passwd file is a colon-separated file that contains the following information:*

* *User name*
* *Encrypted password*
* *User ID number (UID)*
* *User's group ID number (GID)*
* *Full name of the user (GECOS)*
* *User home directory*
* *Login shell*

Example: root:x:0:0:root:/root:/bin/bash

* First, **cat** the passwd file
* Then, pipe output of **cat** as input to **grep** using -in options to ignore case and to output line numbers (just for visual convenience) and search for value “bash”
* Thus the full command is: **cat /etc/passwd | grep -in bash**
  + Text

    Description automatically generated
* Thus only 2 users use the bash shell by default: **root**, and **seed**.
* The following command will output number of lines returned by grep from using wc -l:
  + **grep bash /etc/passwd | wc -l**
  + A picture containing text

    Description automatically generated

## 4. Create a file foo.txt using the “touch” command. Change the file's last-modified date to be January 4 of this year at 8:56am. Include the full command in your report.

I will use this reference from <https://www.ibm.com/docs/en/aix/7.1?topic=t-touch-command> in order to format the time for the touch command:

Text

Description automatically generated

* I will use command **touch** command and use options -m (modify only the date modified time) and -t to set the date based on the -m flag, and using the format as shown above when -t is used:
* **touch -mt 202301040856.22 foo.txt**
* Notice in the screenshot below when I run **stat** command, date Modified is the only thing changed on foo.txt to the jan 4 2023 8:56am time:

A screenshot of a computer

Description automatically generated with medium confidence

# Conclusion

Overall, I learned how to greatly express the power of Linux command line interface using the Bash shell. I especially enjoy the power of the pipe and redirect operators as I see how they can be combined into very elaborate and useful statements that give me some desired output. I think one of the most powerful commands to utilize and understand is undoubtedly the **grep** command. It is very extensive in its options and functionality for a reason, since all file systems rely on regular expressions that humans require in order to use and manage a file system (particularly searching and modifying the file system). Thus, if you can understand how to use **grep, cat,** and the **redirect >** and **pipe |** operators, you will likely be able to understand most of all of the other commands/functions, and be able to harness (through combining) the power of these fundamental commands and operators as combined also with other commands.

# From README.TXT FILE

I use this source to help me compress <https://www.cyberciti.biz/faq/how-do-i-compress-a-whole-linux-or-unix-directory/> :

## 1) To compress the assg1\_complete.tar.gz file:

Graphical user interface, text, application, email

Description automatically generated

* Here, I will explain how I compressed (and how you can decompress) the compressed file assg1\_complete.tar.gz that I uploaded to CANVAS. It contains the modified versions of hw1.1, hw2.2, and hw1.3 that resulted from me working in them upon completion of this lab.
* To compress, I ran the following command:
  + **tar -zcvf assg1\_complete.tar.gz ./hw1.1 ./hw1.2 ./hw1.3**
* This is the directory I was in (Documents)
* 
* After running the above command, this is the result:
* A screenshot of a computer

  Description automatically generated with medium confidence
* You see the .tar.gz file is present.

## 2) To decompress (extract) the file contents of my assg1\_complete.tar.gz file:

* Run the command: **tar -xvf assg1\_complete.tar.gz (note, x flag is for “extract”)**
* This will extract contents of the tar file into the current directory of where the tar file is stored.
* In my case, I created a directory called **test** , and made a copy of the tar file into that directory, then navigated to that directory and ran the command.
* Then **test** directory has these contents after extraction of tar file:
* 
* As you can see, my tar file contains the 3 directories that I worked in as required for the assignment.