Lab *03 – Scripting*

Author: David Murillo
Course Section: IS-1003
Date: 4/3/2023

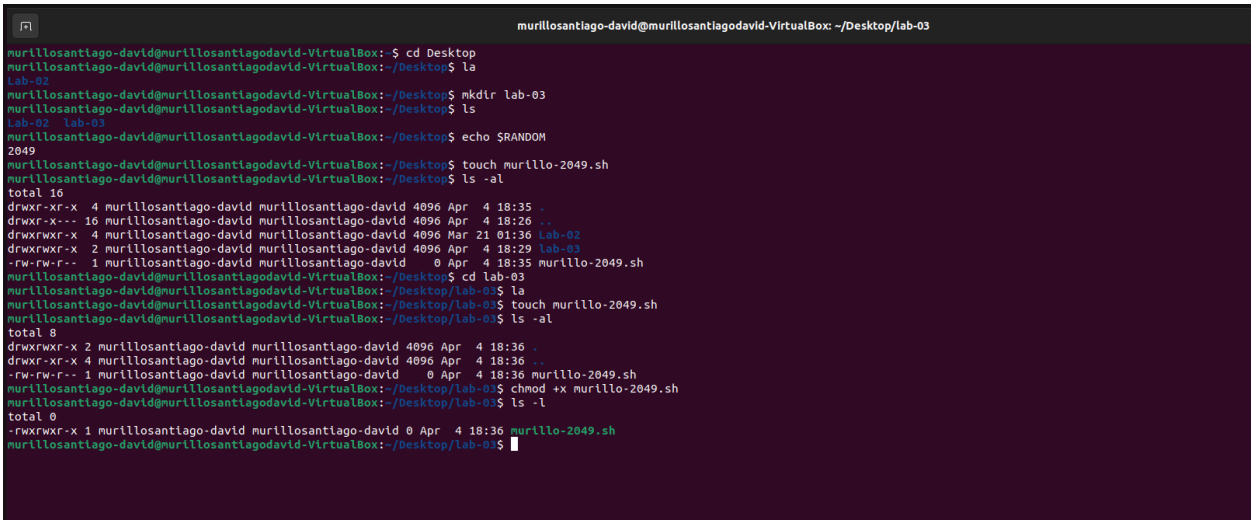
INTRODUCTION

In this lab, I will be learning basic Bash scripting. The Bash shell provides an interface to interact with other programs, retrieve or store data, and execute various tasks using simple commands. Scripting is an important skill to learn for cybersecurity as it can be used for both offensive and defensive maneuvers.

PROCESS

Step *0*: *Set Up Your Environment*

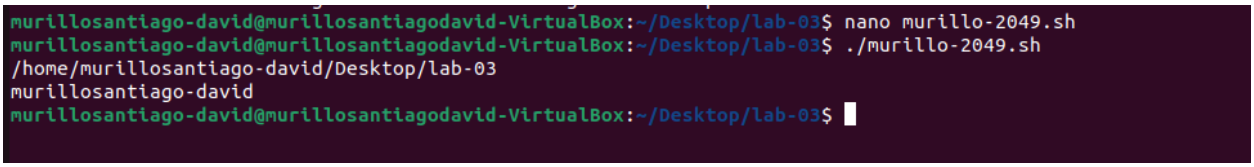
In this step, I created a subfolder for the lab on my virtual machine. I chose to create a folder on my desktop directory using the ***mkdir Lsb-03*** command. Then I used the ***echo $RANDOM*** command to generate a pseudorandom integer. The number that was generated was ***2049.*** I added the generated number to the file name. To create the file, I used the ***touch murillo-2049.sh*** command. After creating the file, I gave the file permission to run. To do this I used the ***chmod +x murillo-2049.sh*** command. I completed this step successfully with no outside help, the only mistake I made was, when I created the ***.sh*** file, I forgot to change directory onto the ***lab-03*** folder. To fix this, I deleted the file I created outside of the terminal, then I returned to the terminal and continued to follow the lab instructions.



*Screenshot of my terminal, revealing how I used the **mkdir** command to create a folder. The **echo $RANDOM** command to generate a random number. The **touch** command to create a **.sh** file. The **chmod** command to give the file permission to run. And the **ls** command, **ls -al** command, and **ls -l** command to verify all files and folder were created successfully. Of the following, the **cd, mkdir, echo $RANDOM, touch, ls,** and **chmod** commands are built-in commands. The variable defined by me, "the programmer", was the file name **murillo-2049.sh***

Step *1*: *Setup a Lab Folder and File*

In this step, I used the ***nano murillo-2049.sh*** command to open the file with the nano editor. Next, I entered the ***pwd*** and ***whoami*** commands to verify my working directory, username, and to create some output to test my script. After that, I pressed CNTRL X on my keyboard to exit the file and I pressed the "y" key to save the file. I then executed the script by entering ***./murillo-2049.sh***

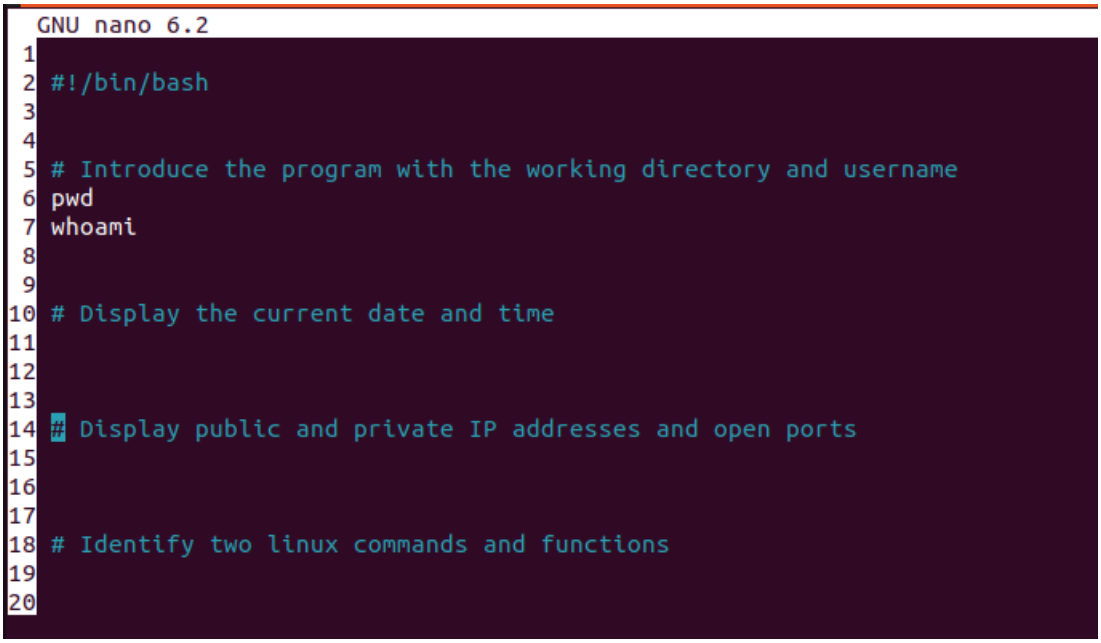


*Screenshot of my command terminal demonstrating my use of the **nano** command to edit my **.sh** file. It also shows my use of the **./murillo-2049.sh** command to run my **.sh** file.*

*Screenshot of my **.sh** file as I'm editing it with nano. The first line is the shebang, where I specify that I will be using the Bash shell. To declare the shebang, I entered "**#!**". The Shebang does not show up in the output as the shebang is simply a declaration of which shell should be used to execute the script.*

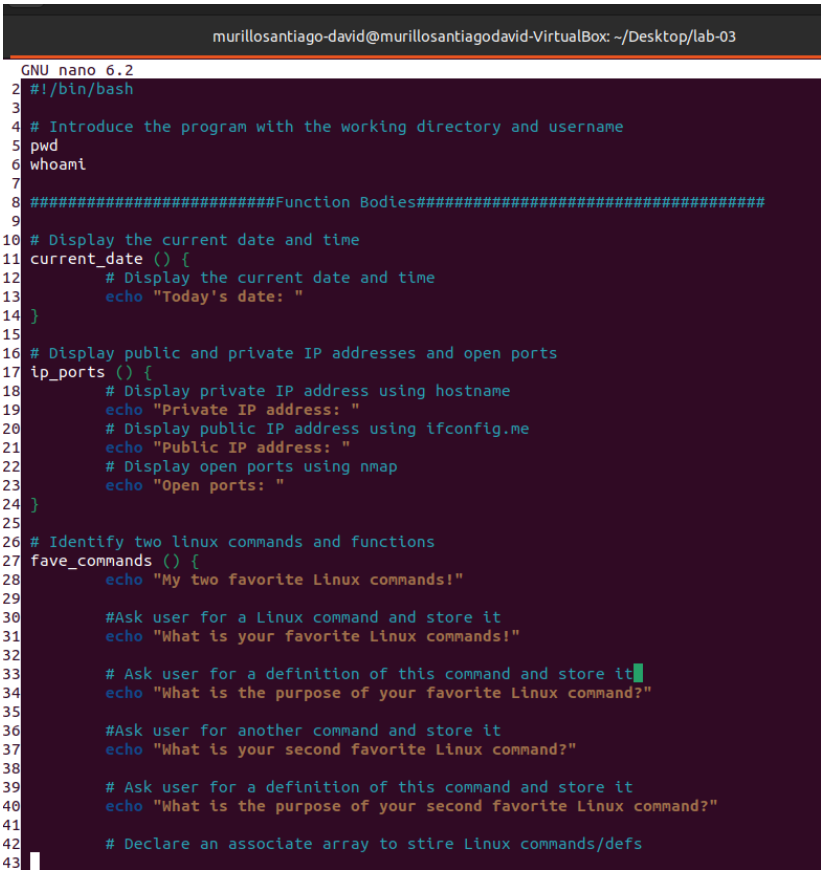
Step *2*: Design Your Script

In this step, I needed to create a script which prints the working directory and username, displays the current date and time, displays public and private IP addresses, and my machine's open ports, and identifies two Linux commands and functions, and displays them. In the previous step, I had already created a command to print the working directory and username.

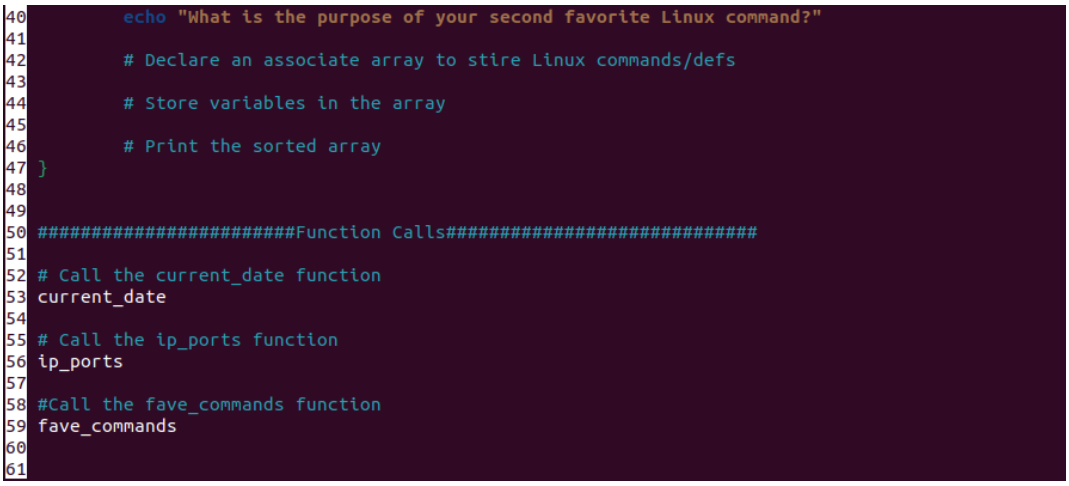


Prior to coding I added comments to the terminal to specify the function of the blocks of codes which I will create.

After creating the listed outline, I added a section for "Function Bodies" and "Function Calls." Originally, I had understood that I needed to create a script which displays the answers to the prompts in the outline, but I understand now that I need to create a script which prints the set up for the answers. For example, today's date. Rather than using a built-in command to print the date, I needed to use the ***echo*** command to literally print "Today's date." I used the ***echo*** command for each of the prompts and I added comments to detail the purpose of each line of code. I modeled my code off the lab instruction video and the included screenshots. Prior to viewing the video, the error I kept making was I would forget to add a function call, thus my code would not execute when I would attempt test runs.



Screenshot of my **.sh** file script. T*he script needed to print the working directory and username, display the current date and time, show the public and private IP addresses, list the open ports on my machine, and identify two Linux commands and functions while displaying them. I had already created a command to print the working directory and username in the previous stage. To begin, I created an outline for the required functionalities, and added sections for "Function Bodies" and "Function Calls." Afterwards, I used the echo command to generate the necessary output for each prompt and added explanatory comments to detail the purpose of each line of code. Finally, I completed the script by adding the required sections for the Linux commands and functions.*



Second half of my script. Reveals my "Function Calls" section.

Step *3*: Write a Date Function

In this step, I had to create a script that revealed the current date and time. To find the command to achieve this I used the following website: https://www.cyberciti.biz/faq/linux-unix-formatting-dates-for-display/ created by nixCraft. From this website, I learned the ***date + %m-%d-%y*** command which allowed me to display the current date by month, day, and year. I also learned the ***%H:%M*** command which displays a 24-hour hour and limit.

```
##########################Function Bodies#######################################

# Display the current date and time
current_date () {
        # Display the current date
        today=$(date +"%m-%d-%y")
        echo -e  "\nToday's date: $today "
        # Display the current time
        current_time=$(date +"%H:%M:%S %p")
        echo -e "The current time: $current_time\n"
}

# Display public and private IP addresses and open ports
ip_ports () {
        # Display private IP address using hostname
        echo "Private IP address: "
        # Display public IP address using ifconfig.me
        echo "Public IP address: "
        # Display open ports using nmap
        echo "Open ports: "
}
```

*Screenshot of my script. I defined the variable today with the built in **date** command and separated month (%m), day (%d), and year (%y) with a dash as that's how I want it to be displayed when executed. The "-e" and "\n" are used to display items on separate lines. I only used them because the instructions used them, but I also executed the code without the use of these commands and got the same output. I followed the same general steps to display current time except the commands I used were **%H:%M:%S %P***

```
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$ ./murillo-2049.sh
/home/murillosantiago-david/Desktop/lab-03
murillosantiago-david

Today's date: 04-05-23
The current time: 07:27:30 AM

Private IP address:
Public IP address:
Open ports:
My two favorite Linux commands!
What is your favorite Linux commands!
What is the purpose of your favorite Linux command?
What is your second favorite Linux command?
What is the purpose of your second favorite Linux command?
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$
```

Screenshot of me executing the script on my terminal. The current date is displayed as "04-05-23" and the time is "07: 27: 30 AM." Two conventions I used in the script were naming conventions (function names) and formatting conventions (indentation). For the function names, the convention is to use lower-case letters and use underscores to separate words. For indentations, the convention is to indent blank lines between blocks. To find the information on conventions I used the following website:

https://google.github.io/styleguide/shellguide.html#s5.1-indentation The site states to never use tab to indent, instead use 2 spaces. I did not follow that convention when creating the script.

Step 4: Finding Your IP Addresses and Ports

In this step, I needed to create a script to display both my public and private IP address and my open ports. To do this, I needed install two programs, “Curl” and “Nmap.” To install curl, I entered the ***sudo apt install curl*** command in the terminal. To install nmap I entered the ***sudo apt install nmap*** command in the terminal. Next, I went back into the script and under the ip_ports function I entered **hostname -I Curl ifconfig.me** and **nmap scanme.nmap.org**. I also used the **echo -e** command as I did in the previous step.

```
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$ sudo apt install curl
[sudo] password for murillosantiago-david:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 223 not upgraded.
Need to get 194 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.10 [194 kB]
Fetched 194 kB in 0s (389 kB/s)
Selecting previously unselected package curl.
(Reading database ... 232726 files and directories currently installed.)
Preparing to unpack .../curl_7.81.0-1ubuntu1.10_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.10) ...
Setting up curl (7.81.0-1ubuntu1.10) ...
Processing triggers for man-db (2.10.2-1) ...
```

*Screenshot of my terminal after I entered the **sudo apt install curl** to install the curl command.*

```
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$ sudo apt install nmap
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libblas3 liblinear4 lua-lpeg nmap-common
Suggested packages:
  liblinear-tools liblinear-dev ncat ndiff zenmap
The following NEW packages will be installed:
  libblas3 liblinear4 lua-lpeg nmap nmap-common
0 upgraded, 5 newly installed, 0 to remove and 223 not upgraded.
Need to get 5,973 kB of archives.
After this operation, 26.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 libblas3 amd64 3.10.0-2ubuntu1 [228 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 liblinear4 amd64 2.3.0+dfsg-5 [41.4 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 lua-lpeg amd64 1.0.2-1 [31.4 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap-common all 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [3,940 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 nmap amd64 7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1 [1,731 kB]
Fetched 5,973 kB in 1s (5,967 kB/s)
Selecting previously unselected package libblas3:amd64.
(Reading database ... 232733 files and directories currently installed.)
Preparing to unpack .../libblas3_3.10.0-2ubuntu1_amd64.deb ...
Unpacking libblas3:amd64 (3.10.0-2ubuntu1) ...
Selecting previously unselected package liblinear4:amd64.
Preparing to unpack .../liblinear4_2.3.0+dfsg-5_amd64.deb ...
Unpacking liblinear4:amd64 (2.3.0+dfsg-5) ...
Selecting previously unselected package lua-lpeg:amd64.
Preparing to unpack .../lua-lpeg_1.0.2-1_amd64.deb ...
Unpacking lua-lpeg:amd64 (1.0.2-1) ...
Selecting previously unselected package nmap-common.
Preparing to unpack .../nmap-common_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_all.deb ...
Unpacking nmap-common (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Selecting previously unselected package nmap.
Preparing to unpack .../nmap_7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1_amd64.deb ...
Unpacking nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Setting up lua-lpeg:amd64 (1.0.2-1) ...
Setting up libblas3:amd64 (3.10.0-2ubuntu1) ...
update-alternatives: using /usr/lib/x86_64-linux-gnu/blas/libblas.so.3 to provide /usr/lib/x86_64-linux-gnu/libblas.so.3 (libblas.so.3-x86_64-linux-gnu) in auto mode
Setting up nmap-common (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Setting up liblinear4:amd64 (2.3.0+dfsg-5) ...
Setting up nmap (7.91+dfsg1+really7.80+dfsg1-2ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
```

*Screenshot of my terminal after I entered the **sudo apt install nmap** to install the nmap command.*

```
# Display public and private IP addresses and open ports
ip_ports () {

        # Display private IP address using hostname
        echo -e  "My private IP address: "
        hostname -I

        # Display public IP address using ifconfig.me
        echo -e  "\nMy public IP address: "
        curl ifconfig.me
        echo
        # Display open ports using nmap
        echo -e "\nThe open ports on my machine: "
        nmap scanme.nmap.org
}
```

*Screenshot of my script. I used the **hostname** command to find my private IP, the **curl ifconfig.me** command to find my public IP address, and the **nmap scname.nmap.org** command to find my open ports.*

```
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$ ./murillo-2049.sh
/home/murillosantiago-david/Desktop/lab-03
murillosantiago-david

Today's date: 04-05-23
The current time: 09:22:56 AM

My private IP address:
10.0.2.15

My public IP address:
104.51.44.251

The open ports on my machine:
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-05 09:22 CDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.053s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 5.83 seconds
My two favorite Linux commands!
What is your favorite Linux commands!
What is the purpose of your favorite Linux command?
What is your second favorite Linux command?
What is the purpose of your second favorite Linux command?
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$
```

*Screenshot of my script being executed on my terminal. My public and private IP addresses are successfully shown as well as my open ports. Based on the provided links, the **curl** command is used to retrieve or send information to or from a server. Nmap is a network exploration tool and port scanner, I used it to determine my open ports.*

Step 5: Using an Associative Array

In this step, I needed to print out an associative array. The following screenshots detail my process:

```
# Identify two linux commands and functions
fave_commands () {
        echo "My two favorite Linux commands!"

        #Ask user for a Linux command and store it
        echo "What is your favorite Linux commands!"
        read first_command

        # Ask user for a definition of this command and store it
        echo "What is the purpose of your favorite Linux command?"
        read first_def

        #Ask user for another command and store it
        echo "What is your second favorite Linux command?"
        read second_command

        # Ask user for a definition of this command and store it
        echo "What is the purpose of your second favorite Linux command?"
        read second_def

        # Declare an associate array to stire Linux commands/defs

        # Store variables in the array

        # Print the sorted array

}

#######################Function Calls#############################
```

*In this step I added the **read** command to the function statement to allow user input. This is needed so the user could enter the answers to the questions the script will show.*

```
My two favorite Linux commands!
What is your favorite Linux commands!
ls
What is the purpose of your favorite Linux command?
it lists contents of directories
What is your second favorite Linux command?
woahmi
What is the purpose of your second favorite Linux command?
it identifies the user
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$
```

Screenshot of me testing the script I had just written. I was able to answer the questions successfully.

```
# Declare an associate array to stire Linux commands/defs
declare -A faves

# Store variables in the array
faves[$first_command]=$first_def
faves[$second_command]=$second_def
```

*I used the **declare -A** command to declare that I will be using an associative array rather than an indexed array. Next, I wrote a script to store the values input by user.*

```
        # Print the sorted array
        echo -e "\nMy favorite Linux commands:"
        for key in "${!faves[@]}"; do
          echo "$key: ${faves[$key]}"
        done | sort
}
```

I created a for loop that iterates through the keys of the associative array. The $ is used to indicate variable substitution. The @ is used to expand the elements in the array and the ! is used to expand the keys in the associative array.

```
My two favorite Linux commands!
What is your favorite Linux commands!
ls
What is the purpose of your favorite Linux command?
lists contents of a directory
What is your second favorite Linux command?
woahmi
What is the purpose of your second favorite Linux command?
identifies the user

My favorite Linux commands:
ls: lists contents of a directory
woahmi: identifies the user
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$
```

Screenshot of me executing the script I just created. It ran successfully.

```
/home/murillosantiago-david/Desktop/lab-03
murillosantiago-david

Today's date: 04-05-23
The current time: 12:18:37 PM

My private IP address:
10.0.2.15

My public IP address:
104.51.44.251

The open ports on my machine:
Starting Nmap 7.80 ( https://nmap.org ) at 2023-04-05 12:18 CDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.051s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 7.39 seconds
My two favorite Linux commands!
What is your favorite Linux commands!
ls
What is the purpose of your favorite Linux command?
lists contents of a directory
What is your second favorite Linux command?
woahmi
What is the purpose of your second favorite Linux command?
identifies the user

My favorite Linux commands:
ls: lists contents of a directory
woahmi: identifies the user
murillosantiago-david@murillosantiagodavid-VirtualBox:~/Desktop/lab-03$
```

Screenshot of my completed script running successfully.

LIMITATIONS/CONCLUSION

In this lab, I learned basic Bash scripting and how to use the Bash shell to interact with other programs, retrieve or store data, and execute various tasks. I successfully completed the lab, including setting up a lab folder and file, designing a script, writing a date function, finding IP addresses and ports, and creating an associative array. Although I was able to complete the lab successfully, one limitation of the

lab was my misunderstanding of Step 2. Originally, I had understood that I had to create the entire script in that single step. It wasn't until I watched the lab video that I realized I was simply creating an outline for the rest of the script. Another limitation was my disregard for the indentation convention. Rather than use 2 spaces to indent, I used the tab key, which I later came to realize was against convention. I continued in this manner in order to keep my script consistent and readable.

REFERENCES

This website helped me find the command to display the date and time:

CyberCiti. "Linux / Unix: Formatting Dates for Display." CyberCiti.biz, 18 June 2008, https://www.cyberciti.biz/faq/linux-unix-formatting-dates-for-display/. Accessed 5 Apr. 2023.

This website helped me understand conventions for scripting with bash:

Google. "Shell Style Guide." Google Developers, Google, n.d., https://google.github.io/styleguide/shellguide.html#s5.1-indentation

This website helped me understand the purpose of the curl command:

"curl - Linux command." Computer Hope, Computer Hope, 12 Mar. 2021, https://www.computerhope.com/unix/curl.htm.

This website helped me understand the purpose of the nmap command:

"nmap command." Computer Hope, Computer Hope, 12 Mar. 2021, https://www.computerhope.com/unix/nmap.htm.

This website helped me understand the read command:

Linux Hint. "Bash read command." Linux Hint, Linux Hint LLC, 3 July 2020, https://linuxhint.com/bash_read_command/

COLLABORATION

I did not collaborate with anyone for this lab, I only used the references I listed and the lab instructions and video.