Deeksha Juneja

Cpre 308

306865588

Purpose of the lab:

In this lab, we are supposed to create a script interpreter which is like bash. We are calling our interpreter cash.

Understanding Bash:

The first part requires us to understand bash. We run a small script of hello_world.sh and check the output. I am attaching the output of the image below:-

```
remote: Total 27 (delta 1), reused 0 (delta 0), pack-reused 14
Unpacking objects: 100% (27/27), done.
From github.com:CprE308/lab-03
 * branch            master     -> FETCH_HEAD
Merge made by recursive.
 lab-03/lab03.md   |  223 +++++++++++++++++++++++++++++++++++++++++++++++++++++
 lab-03/lab03.pdf  |  Bin 0 -> 293594 bytes
 lab-03/testing.sh |   57 ++++++++++++++
 3 files changed, 280 insertions(+), 0 deletions(-)
 create mode 100644 lab-03/lab03.md
 create mode 100644 lab-03/lab03.pdf
 create mode 100644 lab-03/testing.sh
[deeksha@co2048-10 labs]$ bash hello_world.sh
bash: hello_world.sh: No such file or directory
[deeksha@co2048-10 labs]$ cd lab-03
[deeksha@co2048-10 lab-03]$ bash hello_world.sh
“Hello World”
[deeksha@co2048-10 lab-03]$
```

Type of commands:

I am attaching an image to show which type of commands I checked:

```
[deeksha@co2048-10 lab-03]$ type commandToCheck
bash: type: commandToCheck: not found
[deeksha@co2048-10 lab-03]$ type cd
cd is a shell builtin
[deeksha@co2048-10 lab-03]$ type ls
ls is aliased to `ls --color=auto'
[deeksha@co2048-10 lab-03]$ type python
python is /usr/bin/python
[deeksha@co2048-10 lab-03]$
```

The Shellshock Bug:

Shellshock, is a family of bugs in the Unix Bash Shell. They are also know as Bashdoor. It uses environment variable to save a function contain malicious code. The first bug causes Bash to unintentionally execute commands when the commands are concatenated to the end of function definitions stored in the values of environment variables. Attackers exploited Shellshock within hours of the initial disclosure by creating botnets of compromised computers to perform distributed denial-of-service attacks and vulnerability scanning. Shellshock could potentially compromise millions of unpatched servers and other systems. Accordingly, it has been compared to the Heartbleed bug in its severity. The bug has now beein fixed.

Tasks of this Lab:

The first thing that I check if I was getting any argument in my command line. Then from there, I opened a file if I was getting one. I am using #! /bin/lab3 to check for a script file. Then I am going over each line to execute the correct command. My knowledge of the string comparing, splitting and concatenating etc came in rather handy for this lab.

I am able to perform commands cd, pwd, export, echo, can open any thing specified with $PATH, I am also able to ignore al the comments, and I am also able to successfully run commands in the background. Though, I must say that running the commands in the back ground was rather tricky.

I tried to make my code need by making small functions like, empty to check for white spaces, start_func to compare strings, end_func, export_func for the environment variable, the open_path function which basically opens the path and path_execution. This made it much easier for me to debug my code when I added background processes. There came a point when all my variable names and brackets got messed up and nothing was executing. Thankfully, at that time, I had a previous working copy of my code, which I used and made careful changes in.

I basically ran into a lot of trouble while I was adding the background pocesses. I added more if statements and had to create many flags. It got quite a bit messy. Even though this lab was very tricky, I can still say that I learnt a lot form this lab. This lab was definitly a lot more hard then the second lab and I feel should have been for two week probably.