Lab 05 – Searching and Filtering

Raymond Ng: JQG999
IS 1003 Spring 2021
April 25, 2021

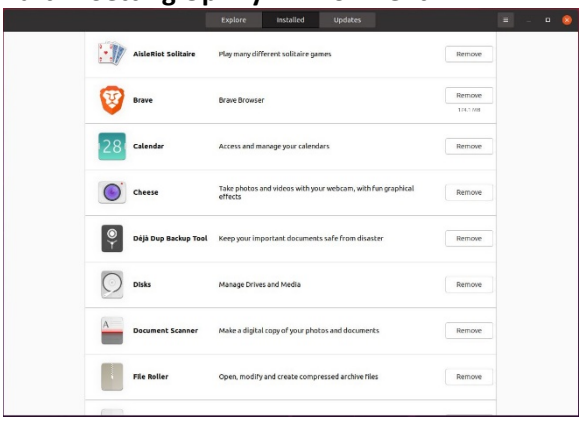
INTRODUCTION

The purpose of this lab was to execute the following:

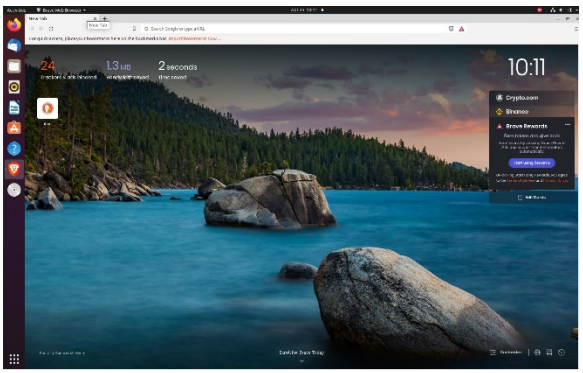
- Secure a browsing environment by using Brave web browser with the DuckDuckGo search engine extension on a Virtual Machine (VM).
- Check to see if my email(s) and password(s) have been discovered via HavIBeenPwned and to establish better password protocols.
- Search my online presence using Google Dorking operators and familiarize with the use of these operators.
- Use and familiarize with the use of the `grep` command and regular expressions (regexes) to filter the Slack workspace.

PROCESS

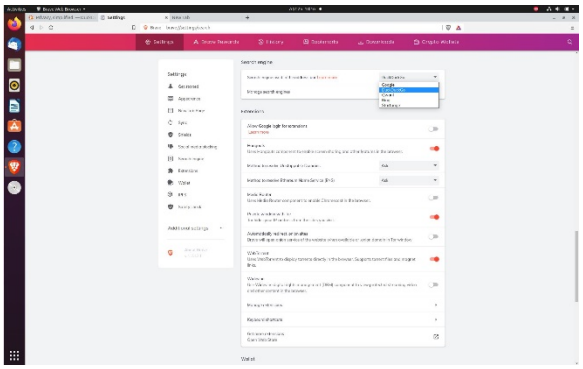
Part 1: Setting Up My Environment



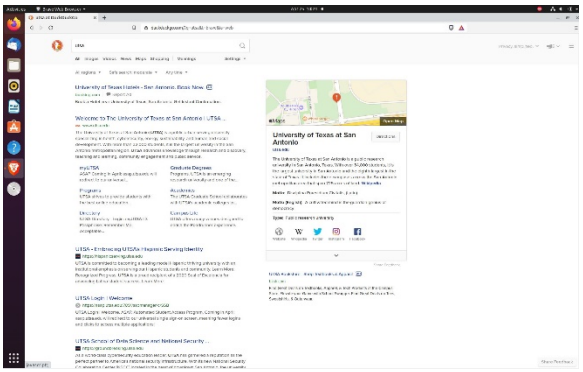
Per the initial block of instructions, I downloaded the Brave browser from the Ubuntu Software Center on my VM. In this lab, I continued using Ubuntu 20.04.2.0 LTS OS that I downloaded via VirtualBox from Lab01.



The installation of Brave took less a minute to install. After the installation was complete, I fired up the browser. There was a walkthrough before I got to the main screen that you see from this screenshot asking simple introductory questions include setting the Brave browser as my default browser.



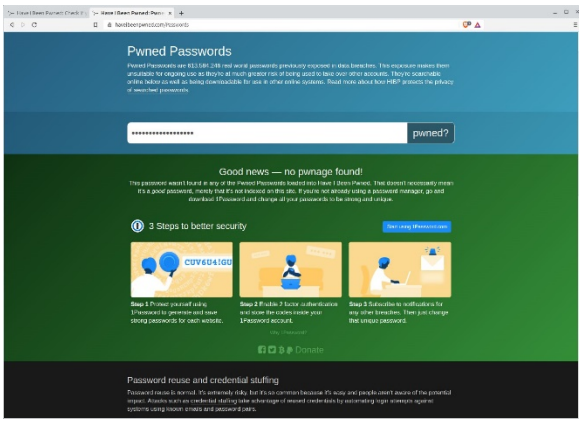
Next, I went to the DuckDuckgo.com and followed the instruction to add the DuckDuckGo extension to my Brave browser. When I clicked on the option to add the extension it prompted a window indicating to go to my search engine settings via the browser to switch it to DuckDuckGo.



Executed a search via DuckDuckGo search engine. The results page looked identical to Google's search engine results. Everything appeared very simple and as expected from a search engine.

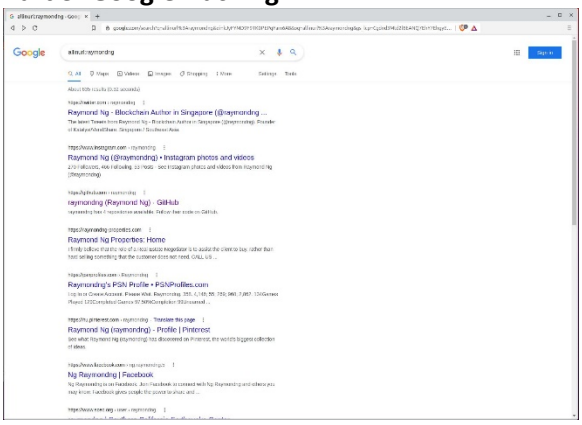
Part 2: Password and E-mail Checks

Using HaveIBeenPwned, I check to see if my personal email had been *pwned*. From the results, apparently my email was part of a data breach that happened back in September 2019 affecting the company Zynga that designs gaming applications. I'm was not too worried about it because the one gaming app I downloaded I used a fake alias as my player name and did not provide any information that would identify myself.

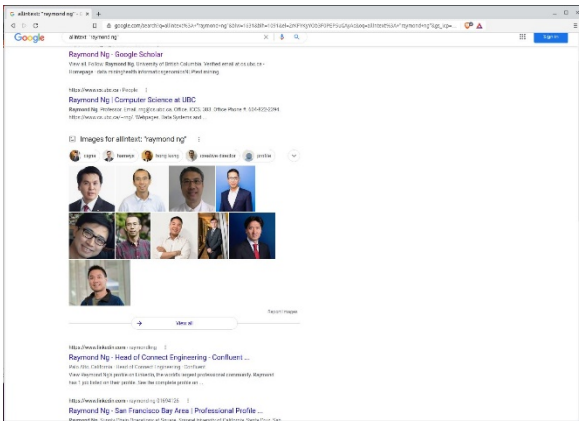


In the second part of this block of instructions I checked my password. The results indicated *no pwnage* was found. It recommended that I used 1Password.com, but I felt it was unnecessary because I have a password manager I use regularly (both mac and windows). Further, from my experience working in a secured environment, I usually change my password every 90 days using the upper/lower case, alphanumeric combinations, with at least one special character.

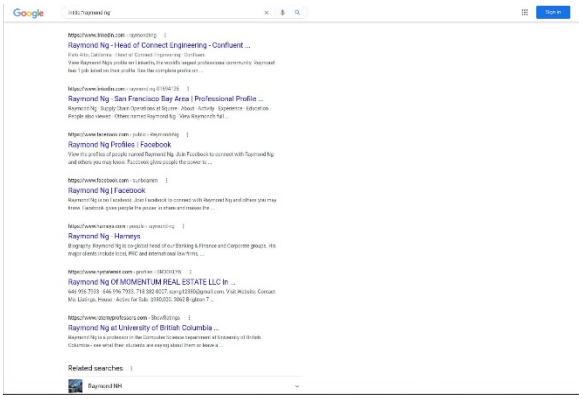
Part 3: Google Hacking



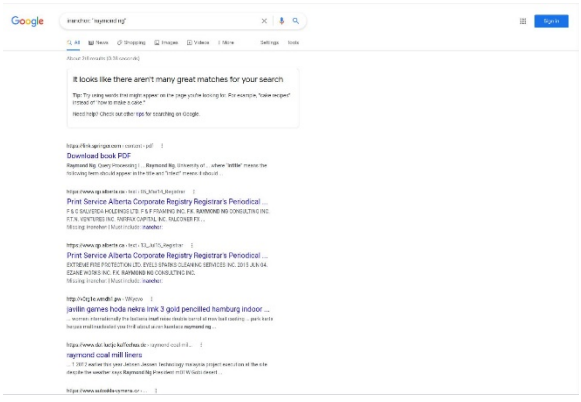
I used the `allinurl` Google Dork operator for a search of my name. It yielded results whose URL contained my name *RaymondNg*. I'm always happy to see that there are so many of me out there via the world wide web. I investigated the results that matched in social media platforms (facebook, Instagram, LinkedIn etc). None of the results were associated with my identity. (Borges, 2021)



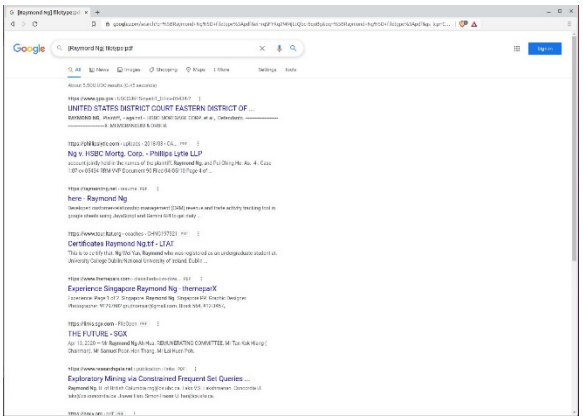
I wanted to continue investigating instances of my name, Raymond Ng, so I used the `allintext` Google Dork operator to locate pages that contained my name inside their text. I investigated the sites on the first two pages just to see if any of them would match my identity. None of them were me. (Borges, 2021)



Here, I used the `allintitle` operator, specifically typed all`intitle:"raymond ng"`, to search for instances of my name in titles. (Borges, 2021)



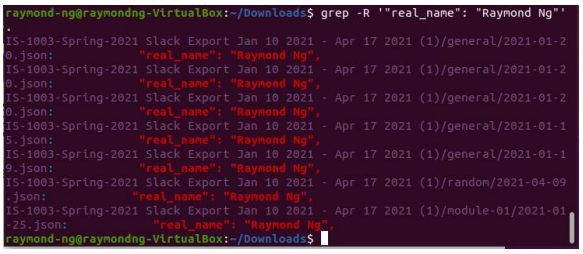
Executed the `inanchor` operator to search anchor text associated with my name. Yielded no results in relation to my identity. (Borges, 2021)



In an effort to continue with my “Defense dorking” of my identity, I wanted to see if there were any files associated with my name so I used `Raymond Ng filetype:pdf` operator to search for any pdf files associated with my name. I also checked for xls, xlsx, doc, docx, ppt files. (Smart Search with GoogleDorking, 2017)

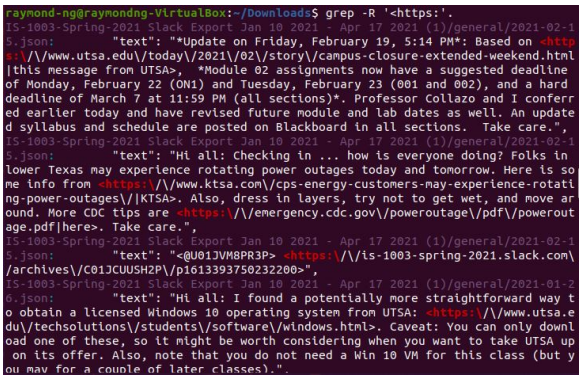
Part 4: Grepping My Slack Workspace

Before I began, I remembered that I had to use a `cd` to change directories, specifically change to my Downloads directory so that I can access the contents in the zip file for this lab.

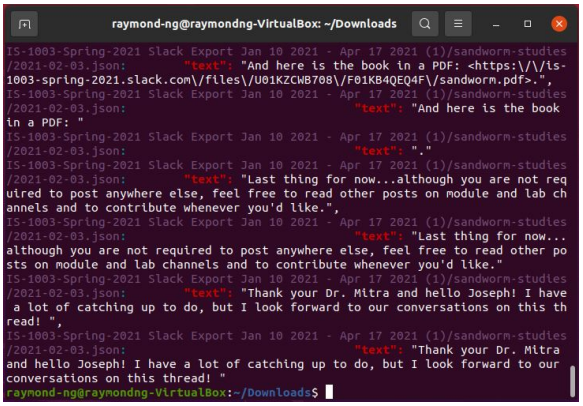


Using the `grep` command I executed a search recursively through all of the folders in my Slack workspace using `-R`, and the `“.”` argument for the filename. The command yielded seven results, which is the number of times I have posted in Slack.

Typed in `grep -R ‘”real_name”: “Raymond Ng”’ . | wc -l` to see how many times my records show up through the entire workspace.

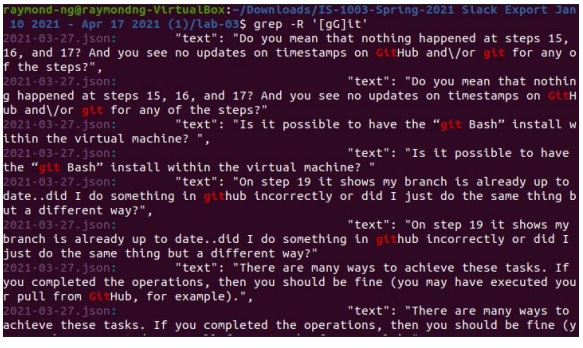


Grabbed all the links in the workspace by searching on `‘<https:’`. Typed in `grep -R ‘<https:’.` I only captured a minimized screen of all the output from the command.

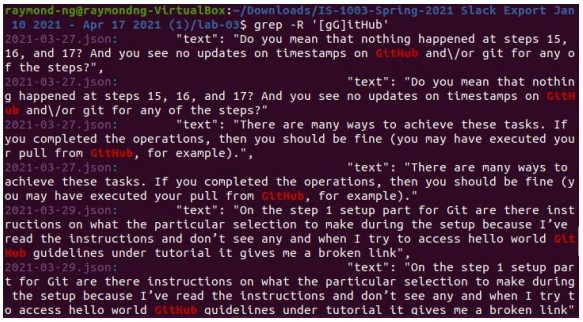


Here I saved all the text of the workspace using the “`text`” keyword. Typed in `grep -R ‘”text”:’.`

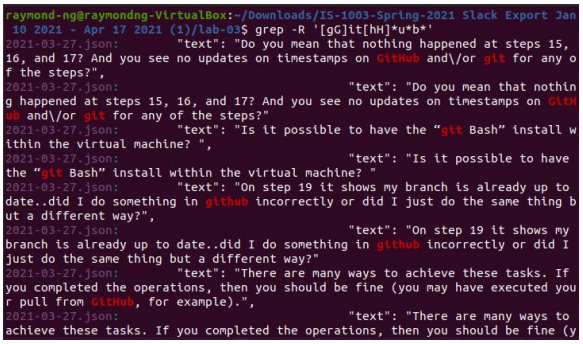
Attempted to execute `grep -R ‘[gG]it lab-03` to find instances of the words “*git*” or “*Git*” in the lab-03 folder, but yielded no results. I tried adding a `‘` at the end of syntax, but that did not work either. I did notice that if I simply just typed in `grep -R ‘[gG]it’` it depicted all instances of the words “*git*” and “*Git*”.



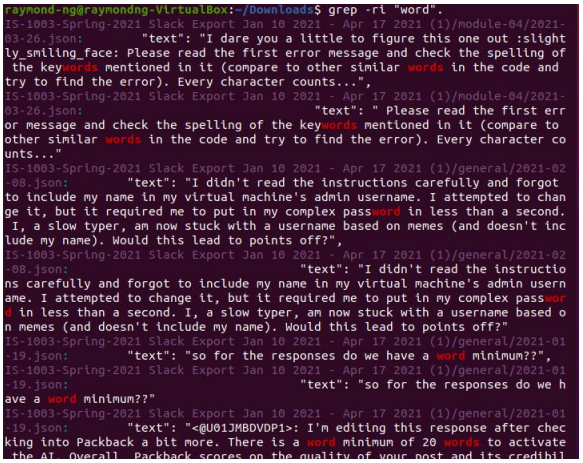
Later, I realized that I had forgot a `‘` after `[gG]it` so the it should have read `grep -R ‘[gG]it’ lab-03’`. However, the output indicated that the lab-03 directory could not be found. Instead, I changed the directory (using the `cd` command) I was working under to the lab-03 folder from the Slack workspace and ran the `grep -R ‘[gG]it’ lab-03’` command again and it searched for all instances of “git” and “Git”.



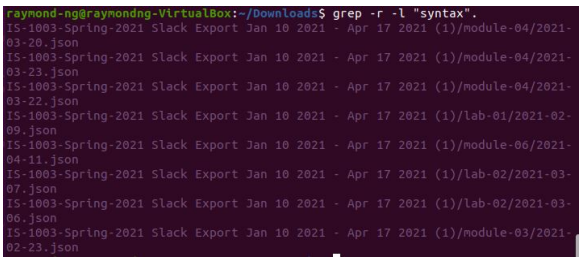
Remaining in the lab-03 file, I search for all instances of the word "GitHub" in my Slack workspace by executing `grep -R '[gG]itHub'` command. The screenshot does not capture the entire output.



This is a screenshot of part of the output from the executing a command using wildcards, specifically the "`*`" wildcard. Here I typed in `grep -R '[gG]it[hH]*u*b*'` into the terminal. The screenshot does not capture the entire output.



I wanted try searching for the word "word" in the Slack workspace without case distinctions. So I typed in `grep -ri "word"`. This screenshot does not capture the entire output of the command. (Gite, 2016)



Lastly, I wanted to search for the word "syntax", but instead of listing all instances of the word in the output I only wanted to see, which files the word would be found in. (Gite, 2016)

CONCLUSION

In conclusion I attempted to secure a safe browsing environment via VM. Additionally, checked to see if my email and passwords have been discovered/compromised. Further, I learned and familiarized myself

with Google Dorking and utilization of the grep command (and regexes) to search and filter in directories in the terminal of linux machines.

This lab felt a little more abstract than the previous ones, particularly when I got to Google Dorking. However, it was great forcing function to research and explore on my own to gain a better understanding. I can see how the combination of the lines of efforts in this lab can help cybersecurity analysts debug in the real world.

REFERENCES

Internet Resources

Borges, E. (2021). *Exploring Google Hacking Techniques - Top Google Dorks*. Retrieved from Security Trails: https://securitytrails.com/blog/google-hacking-techniques

Gite, V. (2016, October 8). *Linux / UNIX Recursively Search All Files For A String*. Retrieved from Cyberciti: https://www.cyberciti.biz/faq/howto-recursively-search-all-files-for-words/

Smart Search with GoogleDorking. (2017, May 29). Retrieved from Exposing The Invisible: https://exposingtheinvisible.org/en/guides/google-dorking/

Collaboration

I attempted to collaborate via Slack, by posting the issue I encountered in Step 4, Part 5 concerning the specific search for a word in a specific lab folder. The instructor noticed I was missing a character in my syntax; however, it did not remedy the issue. I ended debugging on my own by reviewing the `cd` command from a previous lab and the issues was corrected.