Find-Malicious-links

Done by:- Aniket Kumar (2016CS50398)
Anubhav Palway (2016CS10368)

ABOUT

This assignment is about making a python script to find all the malicious links in personal web pages of people(Faculties and Students) hosted by IIT Delhi Computer Science Department.

HOW IT WORKS

First challenge for us was to get web links of all the people. For this we used SSH in CSC to find all the usernames. We copied all the usernames in a text file and copied it back to our system. We made a bash script to do this.

The web-link hosted by CSE has a general format as

www.cse.iitd.ac.in/~<username>

Next part was to find all the weblinks in a page and check if it is malicious or not. For this we have made a python script. We used different python libraries for different tasks.

1. **Requests :** This downloads the html file of the webpage. We passed all the links through this library to download its html.
2. **BeautifulSoup (bs4) :** This library is used for parsing an html. We passed html which we got from requests and parsed it to get a list of all the links in that webpage.
3. **Safe-Browsing-Google-API (gglsbl) :** This library is used for checking if a link is malicious or not and returns the threat which our link exhibit. We passed all the link got from beautifulSoup lists through this to find threats in them.

FUNCTIONAL DESCRIPTION:

1. **Check_format(link) :** This function takes a link as input and checks if the link is going to download some specified files types or not. This had to be done to remove downloading

those files which are generally of large size and does not contain much threat. E.g. mp4, pdf, mp3, jpeg, m4a, ppt, etc. We have given an option to read these formats from a yaml file to make it easy for deciding which to ignore.

2. **Find(domain) :** This returns a set of all the links inside any webpage in the domain. It calls function find_links for this.
3. **Find_links(link, domain, links) :** It is a recursive function to find all the links starting from given domain inside the link. We made it sure not to go back to the link which is already parsed once. This would had caused an infinite recursion. For this we passed the set of lists which is already visited as links.
4. **Find_threats(link, api_key) :** It takes a link and an api_key given by google to use its API and returns the threats inside the link. It returns 'No threat' in case of no threat found.
5. **Main() :** It organizes all the work. It reads an input file containing usernames and create link (www.cse.iitd.ac.in/~<username>) and calls the function find(link) to get all the links in the domain, passes all these links to find_threats to get threat and write it in another file if a threat is found.

HOW TO RUN:

1. **Running python script directly :**

   python3 find-malicious.py <username.txt> <outfile.txt>

2. **Running through bash script :** It gets all the usernames from CSC and runs python script.

   ./run.sh <username> <user type> <out-file>

   e.g. ./run.sh cs5160398 dual output.txt