PREDICTION AND REAL TIME DETECTION OF PHISHING URLs & DOMAINS

CODE DOCUMENTATION v1.0

by Shubham Goyal

Preface

This document contains a high level description of all the files, the functions, the inputs and the outputs. Function descriptions are provided in more detail as in-line comments. In addition to this documentation, it is also recommended to refer to the README file included for a quick start or a refresher to the project.

Note: The project will only work at the Qatar Computing Research Institute (QCRI) on the following VMs 10.2.4.195 and 10.2.4.231 as the project and the program requires files and other data stored within these particular servers.

For further information about the project, please contact Dr Mohamed Nabeel at mnabeel@hbku.edu.qa or Shubham Goyal at shubham@psu.edu.

Special thanks to:

Dr Mohamed Nabeel	HBKU
Dr Minhaj Ahmed	HBKU
Dr Yazan Boshmaf	HBKU
Dr Bei Guan	HBKU
@github/elceef	

Table of Contents

README

The first information in the README is the input file that is universal to almost all inputs in this program. If the file is required by a program, it is included in the relevant directories as well (dnstwist & top_brands). The input file is entitled: **inputData-alexa20180516**.

The second set of information in the file is a set of instructions to go through the entire project. The first part of the project was top_brands, therefore it is the first one that is instructed to be run. There is only one file the user runs, it is main.py. The first input is the inputData file from above. The second input is a count limit, which we set to 500 but for testing, we recommend 10 or less. The third input is a date, traditionally the current date is the input for that.

The second instruction is to change directory to dnstwist which will contain the next 2 files that the user can run. The first file is central_dnstwist and the second is unique_dnstwist_domains. Inputs for central_dnstwist are inputData file and a count limit. Input for unique_dnstwist_domains are the same as input for central_dnstwist. Because of the files that are created whilst the code is running, it is strongly recommended to only run one of these files at a time.

The third instruction is to change directory to data_sources. This directory only contains one file to run, phishtank.py. This file takes in no inputs and produces no outputs.

And finally, change directory to rest_api, and run squatting.py. This allows you to open a browser and view the data in the Mongo collections through the REST API. No inputs are taken in for the python file, however the browser will take in a human input. To view the list of all domains in the registered_domains collection, type in browser:
10.2.4.231:5000/squatting/api/v1.0/registered.
To see information for a particular domain, please type in:
10.2.4.231:5000/squatting/api/v1.0/twisted/domain.com where domain.com is replaced with a domain of your choice.

The remaining commands and instructions in README are information on how to use commands in CMD/Terminal or through a python program to manipulate the collection or table. The commands are specifically for pymongo, Mongo shell, psycopg2, PostgreSQL (terminal)

VM LOGIN:

shubham@10.2.4.195
shubham@10.2.4.231
Username: shubham
Password: shubham1

qcri\sgoyal@10.2.4.195
qcri\sgoyal@10.2.4.231
Username: qcri\sgoyal
Password: sN@5vV!4

data_sources

phishtank.py

Input: None
Description: This file connects to the Mongo and PostgreSQL databases. It connects to the PostgreSQL database to get the daily update on the new domains inside phishtank’s database. That information is then taken and if the domain is already in the collection ‘phishtank’, it is passed over, otherwise it is stored inside the collection as a new document.

utils.py

Input: None
Description: Called from inside phishtank.py and is used to connect to MongoDB and the PostgreSQL database

dns_twist

central_dnstwist.py

Input: **TSV file:** inputData, **Limit:** 100
Description: This function calls phase2.py and utils.py. The purpose of the function is to do the bulk of the work from this file. For each domain that is passed in, it first checks if it is a search engine, it then get a list of its twisted domains, it then adds that information to a collection called dnstwist_<YYYYMMDD>. It then performs the phishtank and virustotal checks for a given domain that is passed into the function. Next it passes the domain into a file called phase2.py and the 2 functions inside it. The file is set up currently for a cronjob. To manually test the file, comment in lines 118 & 119 and comment out lines 121 & 122.

phase2.py

Input: None
Description: This file is called directly from inside central_dnstwist.py. There are 2 functions here. The register_domain function's purpose is to add new domains that are seen for the first time into a collection called registered_domain. If the domain is already in the collection, the function skips over the domain. The second function twisted_domain gets a list of the twisted domains for a given legitimate domain. It then checks if a collection under the name of domain_<domain.com> exists, if it does, it updates the last_seen date, if it doesn't, it creates a collection of the formatted name and adds last seen date, first seen date as well as the list of twisted domains for the given legitimate domain.

unique_dnstwist_domains.py

Input: **TSV file:** inputData, **Limit:** 100
Description: This is a simple program whose sole purpose is to get only a list of the unique DNS twisted domains. After getting the DNS twisted domains, the program checks if it already exists in the collection, if it does, the last seen date is updated to the current date. If the domain isn't in the collection, it adds that into the collection along with a first seen and last seen dates. The file

is set up currently for a cronjob. To manually test the file, comment in lines 6 & 7 and comment out lines 9 & 10.

utils.py

Input: None
Description: Called from inside the above files and is used as a utility tool to do various things like connect to our MongoDB, the SQLite database, get DNStwisted domains and the current days' timestamp.

rest_api

squatting.py

Input: None

Description: The file has three functions for the three possible commands it can work with. The first function takes in only one input as a link (check README) and it returns a list of JSON objects that are inside the registered_domain collection. The second function takes in a curl command that allows you to PUT a domain into the list of twisted domains. The third function takes in an input as a link (check README) which would allow the user to see the list of JSON object for the collection following the domain_<domain.com> format.

utils.py

Input: None

Description: Called from inside squatting.py and is used to connect to MongoDB.

wsgi.py

Input: None

Description: None

top_brands

categoryExtractor.py

Input: Passed through domainExtractor.py
Description: If this file is commented in domainExtractor.py, it is one of the earlier functions called which would take in a domain passed in through domainExtractor, and using BeatifulSoup, it would return a category for that domain. If category can't be found, it returns a string saying “Not Found”. To run this function, access to the internet is required since the function gets the category through a changing website link that is queried.

domainExtractor.py

Input: Passed through main.py
Description: If the main program is run, since initial setup is complete, the commented out lines can remain commented out, however if the program is run when domains.json file doesn't exist, then the following lines need to be commented in: 1, 5, 15-23, 61, 64-71, 79 and this need to be commented out: 6. In the current form with all the commented out lines, it essentially calls the webapi.py file and its function for a domain that is passed in, to query the PDNS database. If the commented out lines were commented in, they would essentially get other information about a domain as listed in jsonCreator.py and store it in a text file called outputFile.txt.

jsonCreator.py

Input: Passed through main.py
Description: The only purpose of this file is that if it is commented in from main.py, then for all the domains in outputFile, create a JSON file that contains rank, brand, domain and category for a domain which can then be stored into a collection. Since the setup is completed and a domains.json file comes in the directory, this file is commented out. This domains.json file that is created requires a user to manually push the json file into a collection through terminal.

Main.py

Input: **TSV file:** inputData, **Limit:** data limit, **Date:** YYYYMMDD
Description: This file takes in three inputs and calls all the other functions in the directory. Currently, after the initial setup has been completed, lines 3, 9, 12 and 13 have been commented out. The program, thus only called the function inside domainExtractor.py, which in turn calls the web_api function inside webapi.py. If domains.json and outputFile.txt isn't included as part of the project directory, comment in the above lines, comment out line 11 and run main.py.

webapi.py

Input: Passed through domainExtractor.py
Description: This function was partially abandoned since it was supposed to get PDNS data on various combosquatting domains for a website but there was an error that couldn't be fixed. However, the file comes with 4 functions. The one that's called from domainExtractor.py and calls the other 3 is web_api. It tries to create a collection for combosquatting domains for all the brands on a particular date. Since wildcard3 was the parameter that failed, all lines using it have been commented out. Otherwise, for the working wildcards, the function performs the PDNS query for a domain on a particular day. It then gets all the unique names in the domain and then attempts to store it in the collection "combosquatting_<YYYYMMDD>" in our MongoDB.

cronjobs

The following files are run as part of the cronjobs set up on the 195 VM:

1. phishtank.py - Run daily at midnight. Logfile created as logfile-phishtank.log

2. central_dnstwist.py - Run daily at midnight. Logfile created as logfile-centralDNSTWIST.log

3. unique_dnstwist_domains.py - Run daily at 1 AM. Logfile created as logfile-uniqueDomains.log

MongoDB

The following are the collections stored in the MongoDB “squatting_db”.

- dnstwist_<YYYYMMDD>
    - List of top 100 domains and all the twisted domains for those 100 domains (so far) for a given date YYYYMMDD
- dnstwist_unique_domains
    - Contains a list of unique twisted domains for the top 100 domains (so far)
- domain_brands
    - Contains a list of the top 500 domains, and their brands, ranks and categories
- domain_<domain.com>
    - Contains a list of last seen, first seen dates and twisted domains for a domain
- registered_domain
    - Contains a list of the top 100 registered domains
- phishtank
    - Contains a list of all the domains that are picked up by phishtank as possible phishing websites