DGA Botnet detection using Collaborative Filtering and Density-based Clustering

Abstract:

In recent years, the botnet phenomenon is one of the most dangerous threat to Internet security, which supports a wide range of criminal activities, including distributed denial of service (DDoS) attacks, click fraud, phishing, malware distribution, spam emails, etc. An increasing number of botnets use Domain Generation Algorithms (DGAs) to avoid detection and exclusion by the traditional methods. This paper presents a novel method to detect DGA botnets using Collaborative Filtering and Density-Based Clustering. This paper proposes a combination of clustering and classification algorithm that relies on the similarity in characteristic distribution of domain names to remove noise and group similar domains.

Introduction:

This research article is published in The Sixth International Symposium on Information and Communication Technology **(SoICT 2015)** which is held in Hue City in Vietnam in **December 2015**. This article has been downloaded from **ACM Digital Library**. The paper can be found at https://dl.acm.org/doi/10.1145/2833258.2833310

Steps & Algorithms:

There are three mains steps involved. They are

- Domain Filtering
- Domain Clustering
- Botnet Detection

1. Domain Filtering

There are two main approaches in domain Filtering. They are whitelist filtering and clustering-classification.

- In whitelist filtering, top benign domains from alexa.com is chosen and wildcards are applied to them to remove the possibly present benign domains in the dataset.
- Using clustering-classification method, 2-gram frequency of domain names are found and so the probability of those 2-grams occurring in the list of domain names are also found. Then using the probabilities, the k-means clustering algorithm is applied with k=2 referring to botnet and non-botnet domains.

2. Domain Clustering

The domains in the botnet cluster are only chosen for this step. In this domain clustering, DBSCAN algorithm (the density based clustering algorithm) is used. This algorithm works by determining the distance between two domains (i.e the number of user connections between the two domains). The clusters are then verified based on the number of benign or non-benign present in them.

3. Botnet Detection

In this step, Collaborative Filtering(CF) is used. Generally CF is used for recommender systems. The idea is to use CF to find all the users whose behaviour is similar with a user that is considered to be a bot in the botnet. The CF performs the following steps:

- Collect the set of domains that is accessed by users in monitored network and give each domain a score.
- Analyse access log to find users who have same behaviour and same set of connected domains.
- List all users who have similar behaviour with user who is considered as a bot in botnet.

Visualisations:

The below visualisations are done for this paper:

- N-gram frequencies comparison
- K-means clustering result
- DBSCAN clusters

Additional works:

- As the dns traffic dataset is really huge (around 3GB), we tried using **MapReduce** to do some of the preprocessing and computation purposes.
- Tried visualising data via **Tableau.**

Map Reduce:

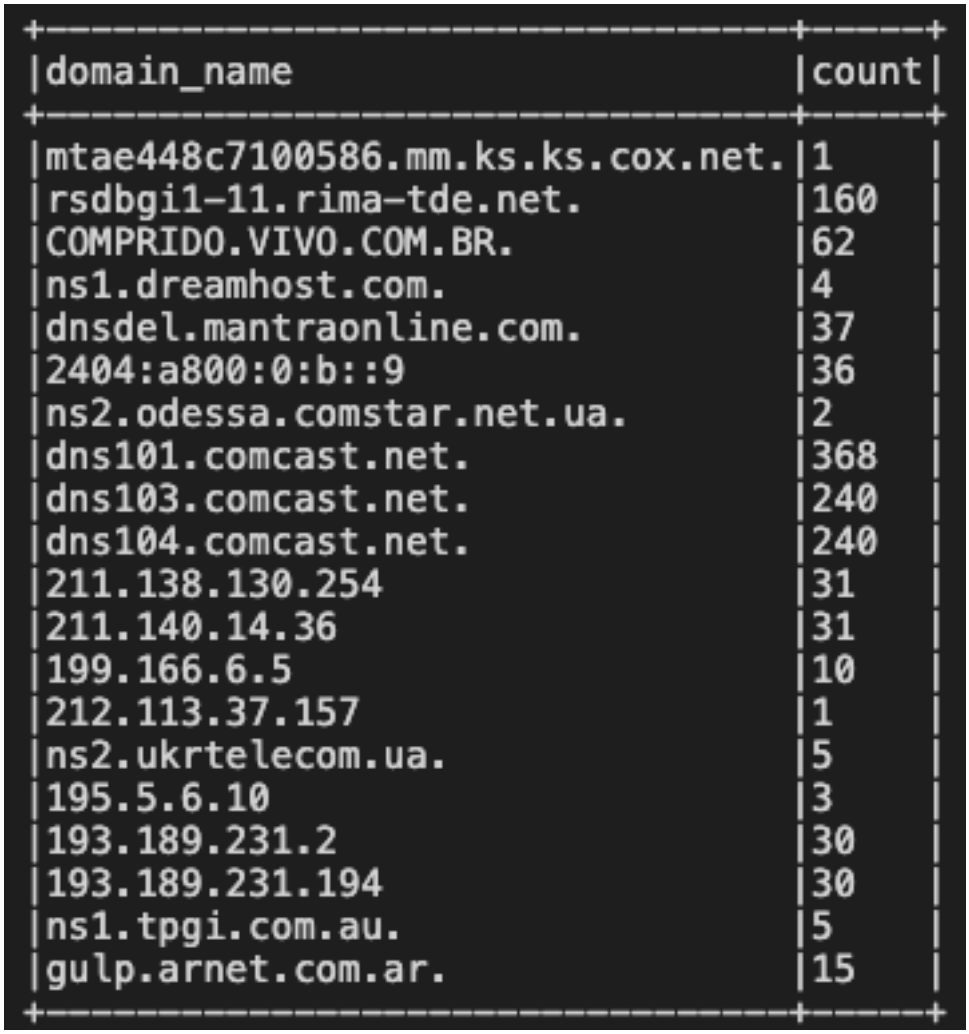
The map reduce code is given below:

```
pairkeynames = (domain_filt_words
                    .map(lambda na: (na, 1))
                    .reduceByKey(lambda o,p: o+p)
                    .collect())
print (pairkeynames)
```

A sample output of the mapper code is given below:

domain_name	count
[a-zA-Z0-9]*.youtube.com	1
[a-zA-Z0-9]*.tmall.com	1
[a-zA-Z0-9]*.baidu.com	1
[a-zA-Z0-9]*.qq.com	1
[a-zA-Z0-9]*.sohu.com	1
[a-zA-Z0-9]*.facebook.com	1
[a-zA-Z0-9]*.taobao.com	1
[a-zA-Z0-9]*.360.cn	1
[a-zA-Z0-9]*.jd.com	1
[a-zA-Z0-9]*.yahoo.com	1
[a-zA-Z0-9]*.amazon.com	1
[a-zA-Z0-9]*.sina.com.cn	1
[a-zA-Z0-9]*.wikipedia.org	1
[a-zA-Z0-9]*.weibo.com	1
[a-zA-Z0-9]*.xinhuanet.com	1
[a-zA-Z0-9]*.netflix.com	1
[a-zA-Z0-9]*.reddit.com	1
[a-zA-Z0-9]*.live.com	1
[a-zA-Z0-9]*.alipay.com	1
[a-zA-Z0-9]*.zhanqi.tv	1

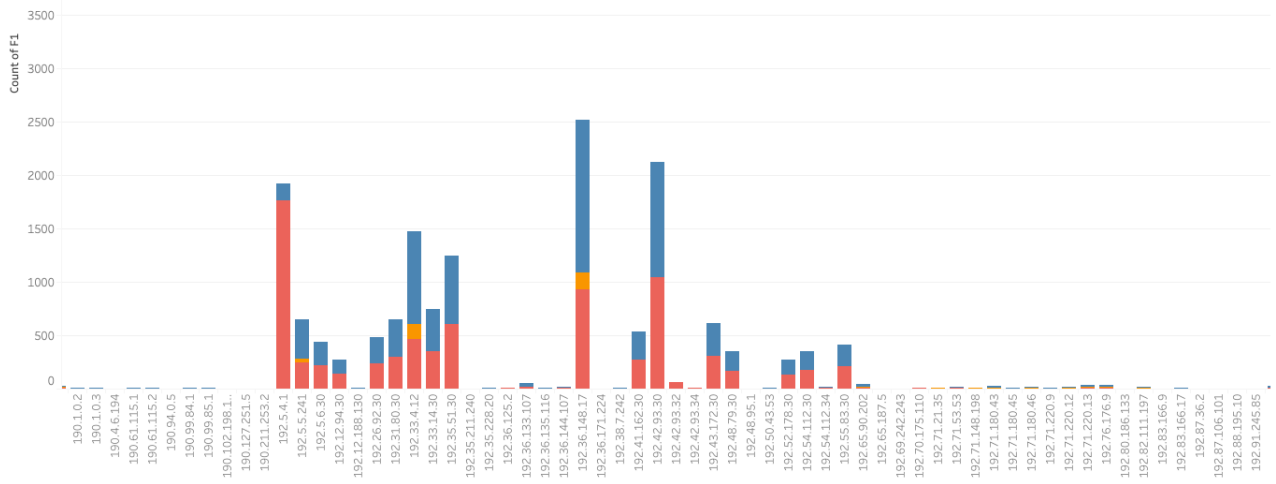
A sample output of the reducer code is given below:



domain_name	count
mtae448c7100586.mm.ks.ks.cox.net.	1
rsdbgi1-11.rima-tde.net.	160
COMPRIDO.VIVO.COM.BR.	62
ns1.dreamhost.com.	4
dnsdel.mantraonline.com.	37
2404:a800:0:b::9	36
ns2.odessa.comstar.net.ua.	2
dns101.comcast.net.	368
dns103.comcast.net.	240
dns104.comcast.net.	240
211.138.130.254	31
211.140.14.36	31
199.166.6.5	10
212.113.37.157	1
ns2.ukrtelecom.ua.	5
195.5.6.10	3
193.189.231.2	30
193.189.231.194	30
ns1.tpgi.com.au.	5
gulp.arnet.com.ar.	15

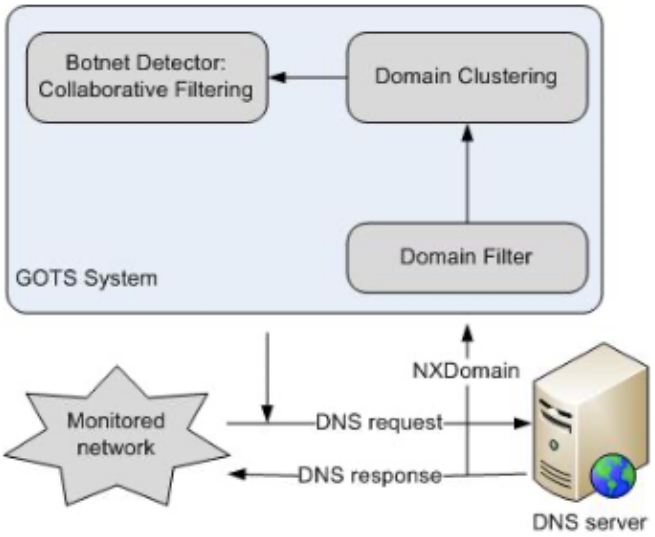
Tableau Visualization:

The dataset is visualized for the count of ip addresses to suspect any malicious activities in the list of domains using the Tableau software. The visualization is given below:



Conclusion:

In this paper, the DGA-botnet detection system – GOTS system is discussed which is shown in the image below. A distance metric that was applied in DBSCAN algorithm that successfully clusters domains was proposed. This system used Collaborative Filtering algorithm to finding out offline infected-machine in each botnet, this is a new approach in this system. GOTS has the advantages that can be performed automated and discovered botnet faster than existing methods.



Working Code:

The working code along with the datasets, research paper and visualizations can be found at the below directory.
https://github.com/KaranrajMokan/DGA-botnet-detection/

Team members:

- Harisaipravin SV (17PW13)
- Karanraj M (17PW18)