Eliza Nip
CS373

Homework 4

First I want to come up with rules. These rules would help me classify rather the URL is malicious because of their characteristics. So how could I determine an URL is malicious? From the hints I was told about various features that could help classifying an URL. I started by counting the number of malicious urls in the train.json file, since it does contain pre-classified malicious urls. I did a simple count on malicious_url == 1 and malicious_url == 0 to get the number of the pre-known malicious urls.

```
#Count
print("Malicious url:")
print(sum(1 for record in urldata if record['malicious_url'] == 1))
print ("Clean url:")
print(sum(1 for record in urldata if record['malicious_url'] == 0))
```

```
Malicious url:
934
Clean url:
1072
```

Train.json file has a total of 2006 urls. Out of these 2006 urls, 934 of them are pre-known malicious url and 1072 of them are not pre-known malicious. I could only do a pre screening with the train file because the classify.json does not have information on pre-known malicious url. I need to find another way to investigate.

The first thing I look at is the domain age. Younger domain age is most likely to be a malicious url. I made the cut off to consider young domain age at 200 days. Any url has the domain age >200 days would pass the test. I did a quick print statement to print all the urls that have domain age < 200 days and their pre-known malicious index(0,1). Turns out all the urls that are printed are the pre-known malicious one. Based on the result, I adjusted the cut off to 700 days, which is a little bit less than 2 years old. I started to receive “clean” urls when I set the cut off to 800 days, so I think 700 days is a fair number.

The next thing I look at is the Alexa rank. Alexa is a useful tool for determining general site popularity and prevalence. The Alexa rank could tell you the popularity ranking of the url. For example, google.com ranks first in Alexa rank. The higher it ranks, the more convincing that it is a legitimate website. By using the Alexa rank, we can filter those that do not have ranking at all(Alexa rank = null) or have a very low ranking(Alexa rank > 1000000). If the website has significantly low Alexa ranking or does not even have one, high chance that it is malicious.

The next thing I look at is the query tag of each url. Domain that does not return IP address, failing DNS query is more likely to be malicious. These domains could be fast-flux domains that

botnets use to hide phishing/malware delivery. Based on that, I filter urls by looking at their query tag. If the url has “query: null” , I classify that as malicious.

The next thing I look at is the file extension. Urls that have “file_extension” are considered malicious.Usually executables (.exe file) are malicious.Therefore, any url that has file_extension(!= null) will be viewed as malicious.

The next thing I look at is the number of path tokens. This number represents the number of paths that need to be route through before getting to the final url destination. I suppose the number of path tokens would be a lot higher for the malicious urls. Therefore I set the filter at num_path_token > 7 that urls that have 7 or more paths before getting to the final url is considered suspicious.

The next thing I look at is the port number of the url. Port 80 and 443 are both internet ports that 80 is the www and 443 is the encrypted https. If the url doesn’t use the standard port, there is a high chance that it is not legitimate. By using such rule, I can filter urls that use non-standard ports, and those would be considered potentially malicious.

I use a point system to give each url a point for not passing the malicious rules.The more points that a url receives, the more convincing that it is malicious. By using the point system, I was able to filter out the “not so malicious looking” url because they have a lower point under my point scale. In my classifier, urls that received more than 2 points are malicious. Below is part of the result in my classify.JSON and train.JSON.

```
http://cnnnewday.wordpress.com/wp-includes/wlwmanifest.xml
Number of points: 9
http://bbs.banq.cn/hotbbs.js
Number of points: 5
http://static.meteorsolutions.com/metsol.js
Number of points: 5
http://download.taobaocdn.com/sns/taoban.apk?file=taoban.apk&t=
Number of points: 9
http://www.eyubogluparke.com/~zeedee/9a70c0acdb2584924f5d0/web.php?#/confirm.php?cmd=login-submit&amp;amp;dispatch=27ad2219b8326ea34c8eb720635e6da5b4bf53c0
db18fd85h0a134
Number of points: 3
http://www.pimei.com/
Number of points: 9
https://www.pinterest.com/join/register/email/
Number of points: 9
http://khanehkeshavarz.ir/modules/mod_articles_category/tmpl/2014/
Number of points: 3
http://www.voisfriend.it//components/com_users/jan.php
Number of points: 3
https://lh6.ggpht.com/hizC1CjCW0oIO1aDB2oZuSHdePaAXkMpP0vP131xo2SDuJdPL1yDw_X1nlFAY_k0xqLY7HoHHATGNv8iXepqW2yCDiEjo5gn2pfdXUU=s660
Number of points: 9
http://cgi.ebay.com-itm137597881453-css-others.session15id-sj3mzbasf3k12z58183115.login-wpadmin.buyitnow.sign-in.pr3ocess253943sd4h53qwg34235hj61rj.xml.con
1fig.pieroth-fran.de/lic/inc/moneyback.htm?ViewStatusj3mzbasf3k12z58138115itm180597802458
Number of points: 3
http://www.compras-ok.com/f
Number of points: 3
http://abateagostina.blogdns.com:8080/
Number of points: 6
http://idergachova.hobby-site.com:8080/
Number of points: 6
http://www.zelkovaconsulting.com.au/wp-admin/includes/Cdocs/Cdocs
Number of points: 3
http://secure-system-online.com/eBay_Buyer_Protection.htm
Number of points: 3
http://cdn-static.denofgeek.com/sites/denofgeek/modules/features/denofgeek_seek_a_geek/promo_box_styles.css?n4j3sw
Number of points: 9
http://remax.com.believeyourheart.com/remax
Number of points: 3
http://jiaxing.auto.sohu.com/
Number of points: 9
http://www.hkej.com/template/dailynews/jsp/detail.jsp?dnews_id=3997&cat_id=7&title_id=679870
Number of points: 9
```

train.JSON- Part of the result

classify.JSON-Part of the result

Turns out that about 985 of the 2006 urls in classify.JSON are malicious. That is about 49% of the total. For the train.JSON, I found about 943 of the 2006 urls are malicious. That is about 47% of the total. Although I didn't get to 50% malicious rate, I think my finding is pretty close to what is supposed to be. I think if I added a few more features, looking into the sub-domain, I would be closer to the 50% malicious rate. I think the result would be more accurate if I filter out all the .php, .js etc. I looked at some of the urls that in the beginning I thought were legitimate but after running it on my classifier, it is indeed malicious. Without the classifier, I might consider a malicious url as legitimate and things would've gone terribly wrong. Imagine users that are not in the IT industry and not familiar with url classification, it would be very difficult for them to spot the malicious urls, since they are very similar to the legitimate ones.