SRT411_Assignment2_joychowdhury

Joy Chowdhury

Interacting With APIs

The Cymon API provides data about malware, botnets, phishing, spam, and other malicious activity on a daily basis from a large repository of sources. The data provided by the API is in a JSON schema, with each object such as domain, ip, malware, and url, contains a response class that provides the details and values in a serializer object. The API is able to analyze if a given domain was created by a human or an algorigthm. It can also look up details of a domain name such as the created/updated date, sources, ips associated with the domain, and urls associated with the domain. For IPs, the API is able to find the created/updated dates, sources, events, domains, and urls. Furthermore, the description of an event associated with an IP is also provided. A malware object associated with an IP contains the hash value, hash type, and events. The code to get an a JSON format response from the API is GET https://cymon.io/api/nexus/v1/ip/x.x.x.x/events/?limit=100&offset=400 The limit specifies the number of items to return, while the offset refers to the point where the output should begin in relation to the complete output. However, the segment of the url after cymon.io can be replaced with other paths to receive a certain kind of data:

Blacklist:

/api/nexus/v1/blacklist/ip/{tag}/? /api/nexus/v1/blacklist/domain/{tag}/

Domain:

/api/nexus/v1/domain/{name}

IP:

/api/nexus/v1/ip/{addr}/timeline/ /api/nexus/v1/ip/{addr}/urls/

Malware:

/api/nexus/v1/malware/{hash_value}/events The data returned can be implemented into R with the json2veris function from the verisr library, and then analyzed graphically using ggplot.

Analyzing Google.com (8.8.8.8) events with Cymon and R

The code for the plot can be found at https://github.com/jchowdhury4/Assignment2 Downloading and observing event data

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
##                                     title                description
## 1  Malware reported by Google SafeBrowsing           Domain: ztxxw.cn
## 2  Malware reported by Google SafeBrowsing          Domain: gzforu.cn
## 3 Phishing reported by Google SafeBrowsing Domain: globals.ssl443.org
## 4  Malware reported by Google SafeBrowsing          Domain: wuzur.com
## 5  Malware reported by Google SafeBrowsing       Domain: 1suckhoe.com
## 6  Malware reported by Google SafeBrowsing     Domain: lwjhhh.08tk.cn
##   details_url              created              updated      tag
## 1        <NA> 2017-03-21T01:07:31Z 2017-03-21T01:07:31Z  malware
## 2        <NA> 2017-03-20T20:51:16Z 2017-03-20T20:51:16Z  malware
## 3        <NA> 2017-03-07T00:54:51Z 2017-03-07T00:54:51Z phishing
## 4        <NA> 2017-03-06T21:43:17Z 2017-03-06T21:43:17Z  malware
## 5        <NA> 2017-03-02T23:24:17Z 2017-03-02T23:24:17Z  malware
## 6        <NA> 2017-03-02T23:11:56Z 2017-03-02T23:11:56Z  malware
```

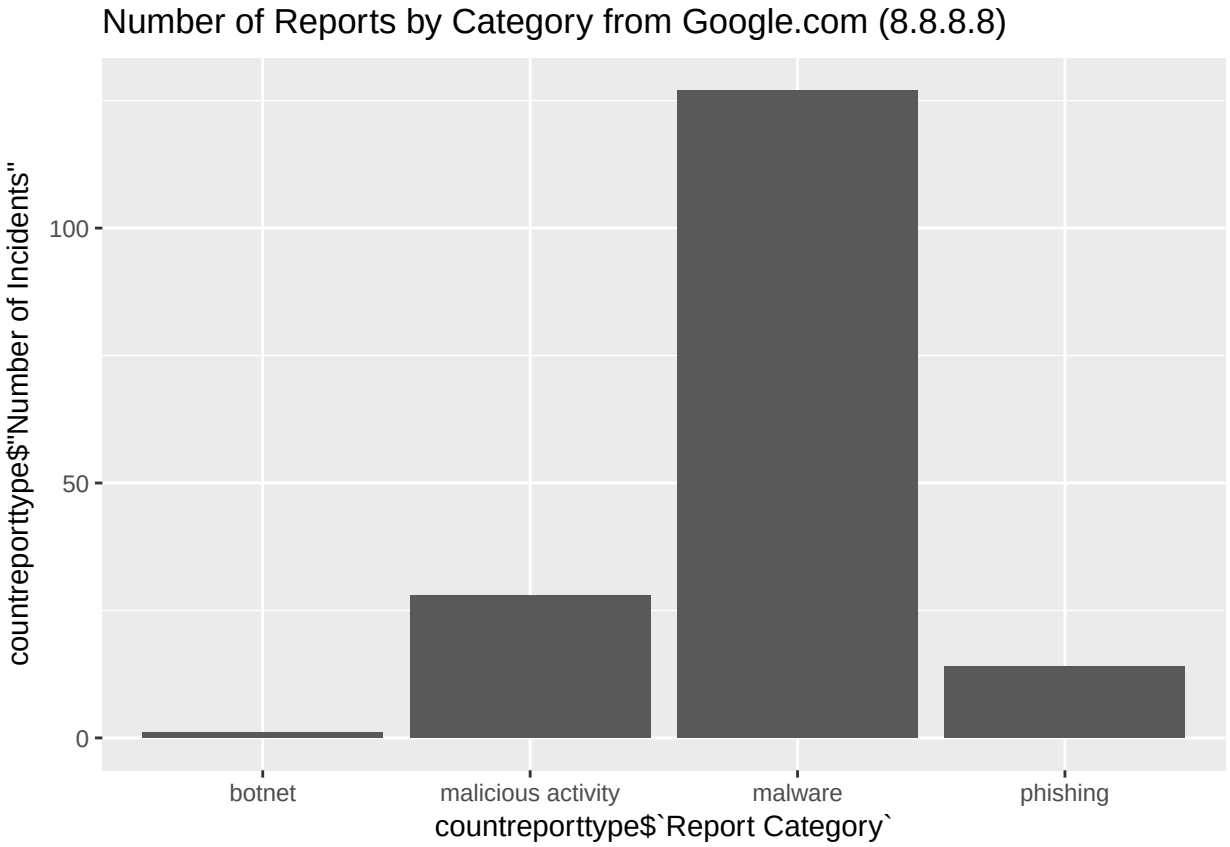
Aggregating data to count number of each type of report

```
countreporttype <- result %>%
  count(tag)
countreporttype <- data.frame(countreporttype)
colnames(countreporttype) <- c("Report Category","Number of Incidents")
countreporttype
```

```
##      Report Category Number of Incidents
## 1             botnet                  1
## 2 malicious activity                 28
## 3            malware                127
## 4           phishing                 14
```

Graphing aggregated data

```
gg <- ggplot(data = countreporttype, aes(x = countreporttype$`Report Category`, y = countreporttype$'Nu
gg
```


Number of Reports by Category from Google.com (8.8.8.8)
countreporttype$"Number of Incidents"
100
50
0
botnet
malicious activity
malware
phishing
countreporttype$`Report Category`