Part 4: Cymon API

Thomas Reid Zuk

March 13, 2016

https://github.com/Freakazoidile/SRT_Assignment_Graphs/tree/master/Part4

Available data:

Cymon API can be sent IP's, tags such as 'Malware' or Spam', domain names, URLs and file hashes to help someone gather information on whether the provided information (IP, URL) is malicious, or has been associated with malicious acitivty in the past.

The Blacklist function of the API takes a supported tags (malware, botnet, spam, phishing, malicious activity, blacklist, dnsbl) and optionally a length of time (3 days) and provides a list of IP or a list of domains associated with the tags.

The DGA (Domain Generator Algorithm) takes a URL like 'virus.com', or 'hsdj34.com' and checks whether or not the domain name is generated by a algorithm or human created.

The Domain function takes a domain URL and provides information such as created date, updated date, the sources, the IP's associated with the domain and urls of the domain.

The IP function takes an IP address and can return the events associated with the IP (malware, spam, ddos), it can provide domains associated with the IP, exact URL's, a timeline of history of events, and any associated malware information (like hashes of malware)

The Malware function takes a hash,MD5, SHA1, SHA256, SHA 512, SSDEEP and finds out of it is known malware. If it is known malware it can return details such as the description (trojan horse, keylogger), when it was created, updated and tags associated with it.

The URL function is similar to the Domain function, but it works for a single page. such as gmail.com/login.php (just an example). It can return the location, the date created, uipdated, sources, IP's associated with the URL and the base domain.

ALL cymon api's return requests in the JSON format.

How to make it useful?

There are many useful programs and tools a security centric person could make so I will explore one such tool/use in some detail.

A security professional could use cymon's API to help find out more detail about abnormal, suspicious network traffic in an organization or use the tool to investigate some software an employee wants to download.

Say an employee wants to download a file from cnet or sourceforge and it is not an the approved software lists. The URL could be given to cymon api to check if the file download has been reported as malicious. A hash of the software, or files the software download contains could be hashed and checked against cymon's malware hash database to check if there is malicious code hidden inside the software.

If someone monitoring a log sees a suspicious link a user visited such as go0gle.com it clearly stands out of a phishing attempt, cymon api can provide more detailed information on to what type of activity has been reported and check what information a user may have exposed.

Malware scanner?

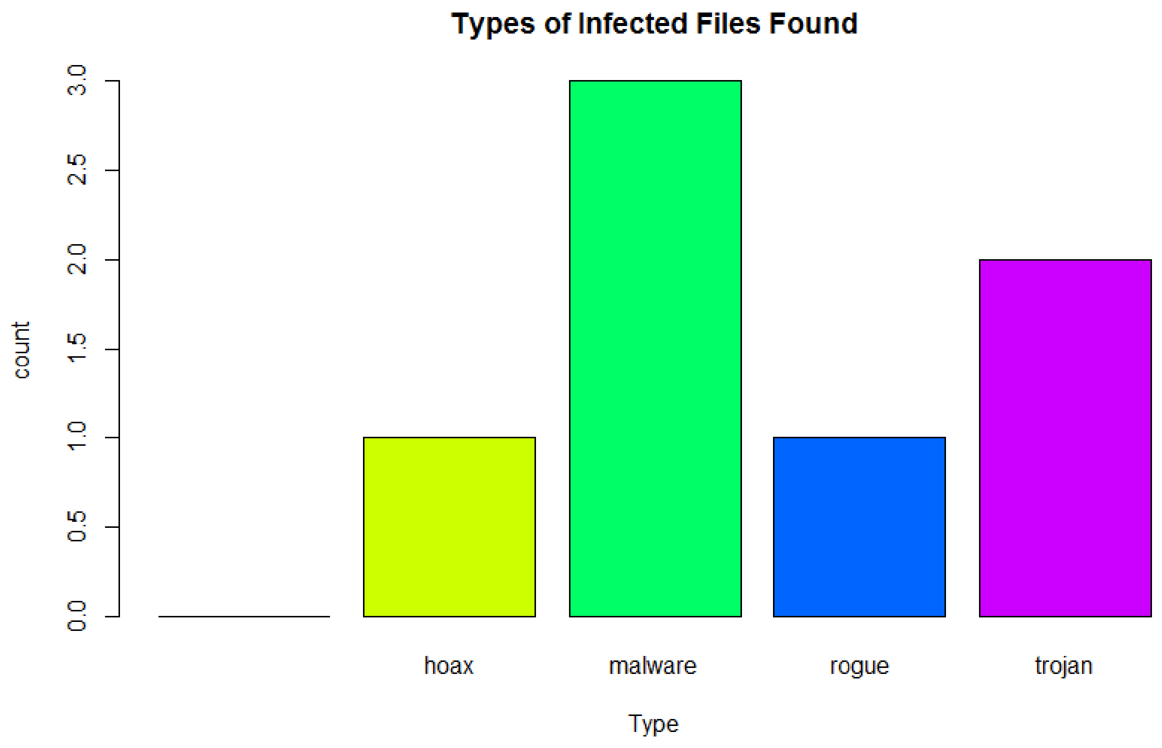
- Can Cymon API be adapted to be used as a malware scanner?
  Goal: Make a directory malware scanner. Scan a user specified directory to check if reported Malware is contained in the directory provided. (Directory level due to rate limits)

```python
import hashlib
import json
import requests
from os import listdir
from os.path import isfile, join


hasher = hashlib.md5()
knownVir = []

def checkThis (resHash, fileName):
    r = requests.get("https://cymon.io/api/nexus/v1/malware/" + resHash + "/events").json()
    if r['count'] == 0:
        print("Hash: " + resHash + "\nsafe")
    else :
        last = r['count']-1
        print("Hash: " + resHash + "\n FILE NOT SAFE!")
        print("Flagged as: " + r['results'][last]['tag'])
        knownVir.append(fileName)

def hashThis (folder):
    files = listdir(folder)
    for i in files :
        thisFile = join(folder,i)
        if isfile(thisFile):

            print("\n----\nFile Name: " + i)
            with open(thisFile, 'rb') as afile:
                buf = afile.read()
                hasher.update(buf)
                checkThis(hasher.hexdigest(), i)



print("==========\nWelcome to Malware Scanner using Cymon\n==========")
folder = input('What directory would you like to scan? ')

hashThis(folder)

print("\n\n===========RESULT===========\nMalicious files found: ", len(knownVir))
print(', '.join(knownVir))
```

The above code can be used on a single directory at a time, such as you Downloads directory, or a uncompressed directory you want to check the files in.

For the report I used known virus hashes and sent the requests to Cymon. I did not download any malware samples to test with, just sent their hashes thru to cymon. After running the scanner and outputing the results to a csv it can be put into R for some simple analysis on the types of infected files founds, the number of files, and the number of each type of file

```
output <- read.csv('output.csv', header = TRUE)
stats <- getenum(output, "tag", add.freq=TRUE, fillzero = FALSE)

barplot(stats$x,
        main="Types of Infected Files Found",
        xlab="Type",
        ylab="count",
        names.arg=stats$enum,
        col=rainbow(nrow(stats))
        )
```



I used my test code and results from cymon to import a CSV and create the graph above of the results of the scan. The graph shows the type of infected files found, groups them by type and shows how many files of each type were found.

References

https://cymon.io/
https://cymon.io/api/docs/