**Analysis of Source and Destination IP using Virus Total**

In this world everyone has there own name like wise every gadget which is connected to Internet has its own name that is IP(Internet Protocol). An IP is a unique identifier assigned to every device connected to the internet. It serves as a numerical label that allows different devices to communicate with each other over the internet.

Since computer can only understand binary values 1s and 0s IP addresses are named in numerical. So, this project will take logs from Wireshark and will calculate the malicious data which is being received or which is being sent.

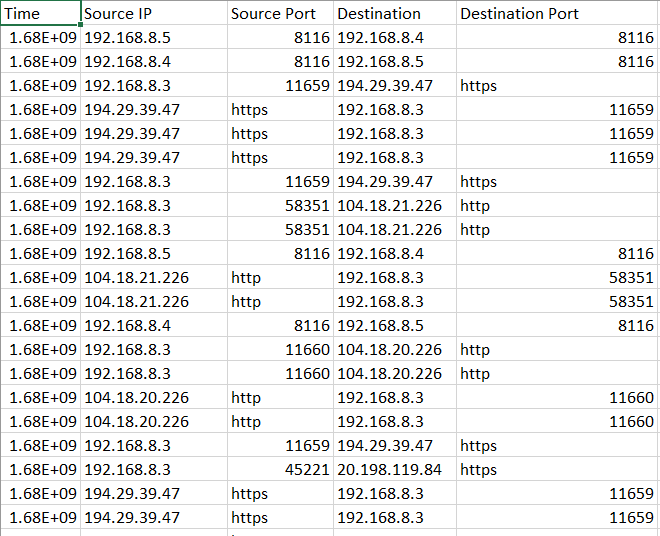
**Step 1:** Capture the packet from Wireshark and store it in .pcap file using Wireshark tool.

**Step 2:** Now run the pcap2csv.py python script to read data from .pcap file using scary library in which it line by line reads the raw data only from pcap file and then it will store it in csv file in order which will be separated with source, source IP and destination, destination IP .

from scapy.all import \*  
import csv  
  
  
# Print the summary info for each packet  
with open('savefile.csv', 'w', newline='') as csvfile:  
 fieldnames = ['Time', 'Source IP','Source Port', 'Destination','Destination Port']  
 writer = csv.DictWriter(csvfile, fieldnames=fieldnames)  
 writer.writeheader()  
  
 for packet in rdpcap('pesce1\_10Lac.pcap'):  
 if packet.summary()[-1] == "w":  
 (source,destination) = (packet.summary().split()[5],packet.summary().split()[7])  
 try:  
 row = {  
 'Time': packet.time,  
 'Source IP': source.split(":")[0],  
 'Source Port': source.split(":")[1],  
 'Destination': destination.split(":")[0],  
 'Destination Port': destination.split(":")[1]  
 }  
 writer.writerow(row)  
 except:  
 writer.writerow({  
 'Time': packet.time,  
 'Source IP': "",  
 'Source Port': "",  
 'Destination': "",  
 'Destination Port': ""  
 })

So, this program will save the raw data from pcap to csv file.

Output:



Step 3: Read Source and Destination IP address from csv file and check Only Public IP

Since we are talking only Public IP address we should first check whether the IP is Public IP or not , If Public IP then store it in separate column in other csv file.

Run this Python Script which will take data from .csv and check IP and then again save back to other .csv file.

SourceCheck.py

import csv  
from ipcheck import is\_private\_ip  
  
with open('savefile.csv', 'r', newline='') as csvfile:  
 data = list(csv.reader(csvfile))  
  
 with open('Public\_IP.csv', 'w', newline='') as savefile:  
 fieldnames = ['Source Public IP','Destination Public IP']  
 writer = csv.DictWriter(savefile, fieldnames=fieldnames)  
 writer.writeheader()  
  
 S\_IP = 0  
 D\_IP = 0  
 lsts = set()  
 lstd = set()  
  
 stks = []  
 stkd = []  
  
 for lines in data[1:]:  
 if is\_private\_ip(lines[1]) == False and lines[1] not in lsts:  
 lsts.add(lines[1])  
 stks.append(lines[1])  
 if is\_private\_ip(lines[3]) == False and lines[3] not in lstd:  
 lstd.add(lines[3])  
 stkd.append(lines[3])  
 if stks and stkd:  
 row = {  
 'Source Public IP': stks.pop(0),  
 'Destination Public IP': stkd.pop(0)  
 }  
 writer.writerow(row)  
 while stks:  
 row = {  
 'Source Public IP': stks.pop(0),  
 }  
 writer.writerow(row)  
 while stkd:  
 row = {  
 'Destination Public IP': stkd.pop(0),  
 }  
 writer.writerow(row)

This program will take source and destination IP from csv and check Public IP only.

Since the function is written in another file which is ipcheck.py

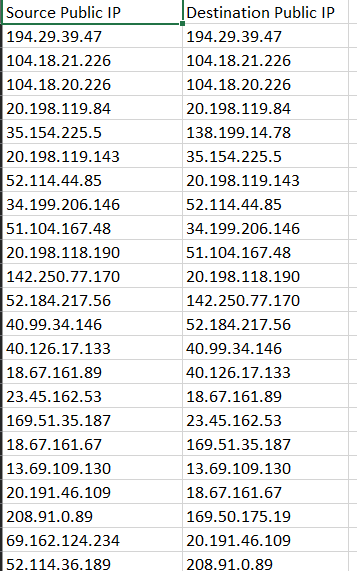
def is\_private\_ip(ip):  
   
 if ":" in ip:  
 # IPv6 address  
 if ip.startswith("fc") or ip.startswith("fd"):  
 return True  
 return False  
 else:  
 # IPv4 address  
 ip = ip.split(".")  
 if ip[0] == "10":  
 return True  
 elif ip[0] == "172" and 16 <= int(ip[1]) <= 31:  
 return True  
 elif ip[0] == "192" and ip[1] == "168":  
 return True  
 return False

This function will check IPs if IP is starting from 10,172,192 which is Private IP address then it will return

True then that IP is Private if not it will return False which is for Public IP address.

Finally, the script sourcecheck.py will check all IP and save to Public\_IP.csv.

Output:



These are Public IP addresses.

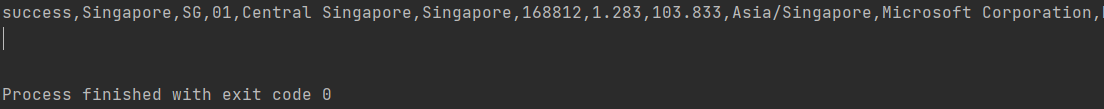
Step 4: Now its time to analysis each and every IP address run this script csvtoanalysis.py which will take Public IP and get back json data and their Country Name save it to csvtoanalysis.csv.

import csv  
import virustotal\_python  
from base64 import urlsafe\_b64encode  
from api import api\_function  
  
  
with open('Public\_IP.csv', 'r', newline='') as csvfile:  
 data = list(csv.reader(csvfile))  
  
 with open('csvtoanalysis.csv', 'w', newline='') as savefile:  
 fieldnames = ['S\_IP','S\_Harmless','S\_Malicious','S\_Suspicious','S\_Undetected','S\_Country',' ','D\_IP', 'D\_Harmless','D\_Malicious','D\_Suspicious','D\_Undetected','D\_Country']  
 writer = csv.DictWriter(savefile, fieldnames=fieldnames)  
 writer.writeheader()  
  
 for lines in data[1:]:  
 print(lines)  
 IP\_S = lines[0]  
 IP\_D = lines[1]  
 country\_S = api\_function(IP\_S).split(',')  
 country\_D = api\_function(IP\_D).split(',')  
 country\_S = country\_S[1] if country\_S[0] == 'success' else 'NO COUNTRY NAME AVAILABLE'  
 country\_D = country\_D[1] if country\_D[0] == 'success' else 'NO COUNTRY NAME AVAILABLE'  
 flag\_S = 0  
 with virustotal\_python.Virustotal(  
 "4905a54d2595dd657db8b0ed31a5b043b0f3abfa120cff8ffe806db537b139eb") as vtotal:  
 flag\_S = 0  
 try:  
 flag = 0  
 resp = vtotal.request("urls", data={"url": IP\_S}, method="POST")  
 url\_id = urlsafe\_b64encode(IP\_S.encode()).decode().strip("=")  
 report = vtotal.request(f"urls/{url\_id}")  
 result = report.data['attributes']['last\_analysis\_stats']  
 except virustotal\_python.VirustotalError as err:  
 flag = 1  
 result = f"Failed to send URL:{IP\_S} for analysis and get the report: {err}"  
  
 flag\_D = 0  
 with virustotal\_python.Virustotal(  
 "4905a54d2595dd657db8b0ed31a5b043b0f3abfa120cff8ffe806db537b139eb") as vtotal:  
 flag\_D = 0  
 try:  
 flag\_D = 0  
 resp = vtotal.request("urls", data={"url": IP\_D}, method="POST")  
 url\_id = urlsafe\_b64encode(IP\_D.encode()).decode().strip("=")  
 report = vtotal.request(f"urls/{url\_id}")  
 result = report.data['attributes']['last\_analysis\_stats']  
 except virustotal\_python.VirustotalError as err:  
 flag = 1  
 result = f"Failed to send URL:{IP\_D} for analysis and get the report: {err}"  
  
 if flag\_S == 0 and flag\_D == 0:  
 row = {  
 'S\_IP': IP\_S,  
 'S\_Harmless': result['harmless'],  
 'S\_Malicious': result['malicious'],  
 'S\_Suspicious': result['suspicious'],  
 'S\_Undetected': result['undetected'],  
 'S\_Country': country\_S,  
 'D\_IP': IP\_D,  
 'D\_Harmless': result['harmless'],  
 'D\_Malicious': result['malicious'],  
 'D\_Suspicious': result['suspicious'],  
 'D\_Undetected': result['undetected'],  
 'D\_Country': country\_D  
 }  
 writer.writerow(row)  
 else:  
 print(result)

This program has some function which is api\_function which is api.py which will return csv data in that its Country name will be available which can be used for analysis.

import requests  
  
def api\_function(IP):  
 api\_base\_url = f"http://ip-api.com/csv/{IP}"  
 response = requests.get(api\_base\_url)  
 return response.text  
  
print(api\_function('52.114.15.109'))

Output:



The csvtoanalysis.py Output:

Table

Description automatically generated with medium confidence

Final Output.

Some other code:

1.Check whether the Virus total API is working Good - > virus\_total\_quota\_check.py

import virustotal\_python  
from base64 import urlsafe\_b64encode  
IP\_S = '20.42.65.90'  
  
with virustotal\_python.Virustotal(  
 "4905a54d2595dd657db8b0ed31a5b043b0f3abfa120cff8ffe806db537b139eb") as vtotal:  
 flag\_S = 0  
 try:  
 flag = 0  
 resp = vtotal.request("urls", data={"url": IP\_S}, method="POST")  
 url\_id = urlsafe\_b64encode(IP\_S.encode()).decode().strip("=")  
 report = vtotal.request(f"urls/{url\_id}")  
 result = report.data['attributes']['last\_analysis\_stats']  
 except virustotal\_python.VirustotalError as err:  
 flag = 1  
 result = f"Failed to send URL:{IP\_S} for analysis and get the report: {err}"  
print(result)

2.To get json data of only one IP address -> Single\_IP\_Analysis.py

import csv  
import virustotal\_python  
from base64 import urlsafe\_b64encode  
from api import api\_function  
  
with open('final.csv', 'r', newline='') as csvfile:  
 data = list(csv.reader(csvfile))  
  
 with open('final - Copy.csv', 'w', newline='') as savefile:  
 fieldnames = ['S\_IP','S\_Harmless','S\_Malicious','S\_Suspicious','S\_Undetected','S\_Country']  
 writer = csv.DictWriter(savefile, fieldnames=fieldnames)  
 writer.writeheader()  
  
 for lines in data[1:]:  
 print(lines)  
 IP\_S = lines[0]  
 country\_S = api\_function(IP\_S).split(',')  
 country\_S = country\_S[1] if country\_S[0] == 'success' else 'NO COUNTRY NAME AVAILABLE'  
 flag\_S = 0  
 with virustotal\_python.Virustotal(  
 "4905a54d2595dd657db8b0ed31a5b043b0f3abfa120cff8ffe806db537b139eb") as vtotal:  
 flag\_S = 0  
 try:  
 flag = 0  
 resp = vtotal.request("urls", data={"url": IP\_S}, method="POST")  
 url\_id = urlsafe\_b64encode(IP\_S.encode()).decode().strip("=")  
 report = vtotal.request(f"urls/{url\_id}")  
 result = report.data['attributes']['last\_analysis\_stats']  
 except virustotal\_python.VirustotalError as err:  
 flag = 1  
 result = f"Failed to send URL:{IP\_S} for analysis and get the report: {err}"  
  
  
 if flag\_S == 0: # and flag\_D == 0:  
 row = {  
 'S\_IP': IP\_S,  
 'S\_Harmless': result['harmless'],  
 'S\_Malicious': result['malicious'],  
 'S\_Suspicious': result['suspicious'],  
 'S\_Undetected': result['undetected'],  
 'S\_Country': country\_S  
 }  
 writer.writerow(row)  
 else:  
 print(result)

3.IF need to clear csv files -> clearcsv.py

#Program to clear the csv file  
# f = open("csvtoanalysis.csv", "w")  
# f = open("savefile.csv", "w")  
f = open("Public\_IP.csv", "w")  
f.truncate()  
f.close()