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:

Time	Source IP	Source Port	Destination	Destination Port
1.68E+09	192.168.8.5	8116	192.168.8.4	8116
1.68E+09	192.168.8.4	8116	192.168.8.5	8116
1.68E+09	192.168.8.3	11659	194.29.39.47	https
1.68E+09	194.29.39.47	https	192.168.8.3	11659
1.68E+09	194.29.39.47	https	192.168.8.3	11659
1.68E+09	194.29.39.47	https	192.168.8.3	11659
1.68E+09	192.168.8.3	11659	194.29.39.47	https
1.68E+09	192.168.8.3	58351	104.18.21.226	http
1.68E+09	192.168.8.3	58351	104.18.21.226	http
1.68E+09	192.168.8.5	8116	192.168.8.4	8116
1.68E+09	104.18.21.226	http	192.168.8.3	58351
1.68E+09	104.18.21.226	http	192.168.8.3	58351
1.68E+09	192.168.8.4	8116	192.168.8.5	8116
1.68E+09	192.168.8.3	11660	104.18.20.226	http
1.68E+09	192.168.8.3	11660	104.18.20.226	http
1.68E+09	104.18.20.226	http	192.168.8.3	11660
1.68E+09	104.18.20.226	http	192.168.8.3	11660
1.68E+09	192.168.8.3	11659	194.29.39.47	https
1.68E+09	192.168.8.3	45221	20.198.119.84	https
1.68E+09	194.29.39.47	https	192.168.8.3	11659
1.68E+09	194.29.39.47	https	192.168.8.3	11659

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:

Source Public IP	Destination Public IP
194.29.39.47	194.29.39.47
104.18.21.226	104.18.21.226
104.18.20.226	104.18.20.226
20.198.119.84	20.198.119.84
35.154.225.5	138.199.14.78
20.198.119.143	35.154.225.5
52.114.44.85	20.198.119.143
34.199.206.146	52.114.44.85
51.104.167.48	34.199.206.146
20.198.118.190	51.104.167.48
142.250.77.170	20.198.118.190
52.184.217.56	142.250.77.170
40.99.34.146	52.184.217.56
40.126.17.133	40.99.34.146
18.67.161.89	40.126.17.133
23.45.162.53	18.67.161.89
169.51.35.187	23.45.162.53
18.67.161.67	169.51.35.187
13.69.109.130	13.69.109.130
20.191.46.109	18.67.161.67
208.91.0.89	169.50.175.19
69.162.124.234	20.191.46.109
52.114.36.189	208.91.0.89

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:

```
success,Singapore,SG,01,Central Singapore,Singapore,168812,1.283,103.833,Asia/Singapore,Microsoft Corporation,

Process finished with exit code 0
```

The csvtoanalysis.py Output:

S_IP	S_Harmles	S_Maliciou	S_Suspicio	S_Undetec	S_Country	D_IP	D_Harmles	D_Maliciou	D_Suspicic	D_Undete	D_Country
45.83.245.	78	0	0	12	Germany	20.190.14(	78	0	0	12	India
52.205.53.	63	0	0	8	United States	23.58.37.2	63	0	0	8	India
169.50.34.	79	0	0	11	Netherlands	20.190.14(	79	0	0	11	India
91.189.91.	59	0	0	7	United States	23.58.26.2	59	0	0	7	India
20.190.14(	83	0	0	10	India	34.96.84.3	83	0	0	10	United States
23.58.37.2	79	0	0	10	India	23.41.186.	79	0	0	10	India
20.190.14(	76	1	0	13	India	13.89.179.	76	1	0	13	United States
23.58.26.2	77	0	0	13	India	20.118.13(	77	0	0	13	United States
34.96.84.3	75	0	0	15	United States	8.241.137.	75	0	0	15	Singapore
23.41.186.	82	0	0	11	India	8.255.132.	82	0	0	11	Singapore
13.89.179.	82	0	0	10	United States	185.125.19	82	0	0	10	United Kingdom
107.182.1(	79	0	0	9	United States	40.99.34.1	79	0	0	9	India
20.118.13(	75	0	0	17	United States	20.190.14(	75	0	0	17	India
8.241.137.	78	0	0	12	Singapore	20.205.24(	78	0	0	12	Singapore
45.93.16.2	78	0	0	10	Palestinian Territory	8.241.133.	78	0	0	10	Singapore
8.255.132.	76	0	0	14	Singapore	51.104.15.	76	0	0	14	United Kingdom
185.125.19	77	0	0	15	United Kingdom	216.58.19(	77	0	0	15	India
185.224.1(	76	0	0	14	Netherlands	20.42.65.8	76	0	0	14	United States
40.99.34.1	76	0	0	14	India	52.185.21(	76	0	0	14	United States

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()
```