ACO331 Project 2 – Jack Sharkey

```
Attacking device IP: 192.168.0.107 (Device A)
Victim device IP:192.168.0.203 (Device B)

Device A used nmap command to scan ports 1-5000 on device B where the flow capture tool is running.
Both VM's are connected to my local network at home with bridged adapter setting.
Device A is running on my laptop and device B is running on my Desktop.
Source IP Adress 192.168.0.203:

1) Largest Packet Count Flow:

0306.00:58:39.555 0306.00:58:46.777 0 192.168.0.203 41982 0 192.178.48.226 443 6 3 106 18835

2) Largest Byte Count Flow:

0306.00:58:39.555 0306.00:58:46.777 0 192.168.0.203 41982 0 192.178.48.226 443 6 3 106 18835

3) For all records with Source IP Adress 192.168.0.203:

        a) Total flows              - 3261
        b) Total packets            - 4674
        c) Total bytes              - 328715
        d) Unique soruce ports      - 3220
        e) Unique Destination IPs   - 51
        f) Unique Destination Ports - 2407
        g) Unique Protocols         - 3
        h) Top 5 Source Ports based on flow counts:
                Port  | Protocol | Frequency
                ---------------------------
                38531 | Unknown  | 3
                22412 | Unknown  | 3
                55692 | Unknown  | 3
                45027 | Unknown  | 3
                35485 | Unknown  | 3

        i) Top 5 Destination Ports based on flow counts:
                Port  | Protocol | Frequency
                ---------------------------
                53    | domain   | 553
                443   | https    | 45
                80    | http     | 18
                45562 | Unknown  | 3
                37000 | Unknown  | 3

        j) Top 5 Destination IP addresses based on flow counts:
                IP Address     | Flow Count
                ---------------------------
                192.168.0.107  | 2642
                68.105.29.11   | 419
                68.105.28.12   | 78
                68.105.28.11   | 59
                23.220.73.211  | 6

        k) Top 5 Destination IP addresses based on packet counts:
                IP Address     | Packet Count
                ----------------------------
                192.168.0.107  | 2642
                68.105.29.11   | 423
                192.178.49.2   | 125
                192.178.48.226 | 106
                104.19.211.131 | 103

        l) Top 5 Destination IP addresses based on byte counts:
                IP Address     | Byte Count
                ---------------------------
                192.168.0.107  | 105780
                68.105.29.11   | 29422
                192.178.48.226 | 18835
                192.178.49.2   | 14546
                104.19.211.131 | 8323
```

This is the attackers IP Address, from the nmap scan

These addresses are cox servers

IP address from a cloud security company, Akamai Technologies

These IP addresses are from google

This IP Adress is from Cloudflare, Inc. which is a CDN

```
Destination IP Adress 192.168.0.203:

4) Largest Packet Count Flow:

0306.00:58:39.555 0306.00:58:46.777 0 192.178.48.226 443 0 192.168.0.203 41982 6 3 149 226188

5) Largest Byte Count Flow:

0306.00:58:41.506 0306.00:58:46.769 0 104.19.211.131 443 0 192.168.0.203 56394 6 3 135 1633759

6) For all records with Destination IP Adress 192.168.0.203:

        a) Total flows              - 3258
        b) Total packets            - 4753
        c) Total bytes              - 3433391
        d) Unique Source IPs        - 51
        e) Unique soruce ports      - 2406
        f) Unique Destination Ports - 3219
        g) Unique Protocols         - 2
        h) Top 5 Source Ports based on flow counts:
                Port  | Protocol | Frequency
                --------------------------
                53    | domain   | 553
                443   | https    | 45
                80    | http     | 18
                45562 | Unknown  | 3
                37000 | Unknown  | 3

        i) Top 5 Destination Ports based on flow counts:
                Port  | Protocol | Frequency
                --------------------------
                38531 | Unknown  | 3
                22412 | Unknown  | 3
                55692 | Unknown  | 3
                45027 | Unknown  | 3
                35485 | Unknown  | 3

        j) Top 5 Source IP addresses based on flow counts:
                IP Address     | Flow Count
                ---------------------------
                192.168.0.107  | 2642
                68.105.29.11   | 418
                68.105.28.12   | 77
                68.105.28.11   | 58
                23.220.73.211  | 6

        k) Top 5 Source IP addresses based on packet counts:
                IP Address     | Packet Count
                ---------------------------
                192.168.0.107  | 2652
                68.105.29.11   | 419
                192.178.49.2   | 155
                192.178.48.226 | 149
                104.19.211.131 | 135

        l) Top 5 Source IP addresses based on byte counts:
                IP Address     | Byte Count
                ---------------------------
                104.19.211.131 | 1633759
                192.178.48.226 | 226188
                34.120.237.76  | 213420
                192.178.49.2   | 187092
                104.18.130.236 | 161182
```

This is the attackers IP Address, from the nmap scan

These addresses are cox servers

IP address from a cloud security company, Akamai Technologies

This is the attackers IP Address, from the nmap scan

These IP addresses are from google

This IP Adress is from Cloudflare, Inc. which is a CDN

This IP Adress is from Cloudflare, Inc. which is a CDN

These IP addresses are from google

This IP Adress is from Cloudflare, Inc. which is a CDN

This is the script I used to parse the Netflow data:

```
from collections import Counter

from socket import getservbyport


def main():


    victim_ip = '192.168.0.203'

    attack_ip = '192.168.0.107'


    inputFile = 'netflow.data4.txt'


    print(f'\nAttacking device IP: {attack_ip} (Device A)\nVictim device IP:{victim_ip} (Device B)\n')

    print(f'Device A used nmap command to scan ports 1-5000 on device B where the flow capture tool is running.')

    print(f"Both VM's are connected to my local network at home with bridged adapter setting.")

    print(f"Device A is running on my laptop and device B is running on my Desktop.")


    l1 = proccessFlow(inputFile, victim_ip, True)

    l2 = proccessFlow(inputFile, victim_ip, False)


    printItems(True, l1)

    printItems(False, l2)


#Method to proccess the information

def proccessFlow(inputFile, victim_ip, source):


    mostPackets = 0
```

```
mostBytes = 0
total_flows = 0
total_packets = 0
total_bytes = 0

uniq_srcport = Counter()
uniq_ip = Counter()
uniq_dstport = Counter()
uniq_protocol = Counter()
uniq_packet = Counter()
uniq_byte = Counter()

with open(inputFile, 'r') as f:
    next(f) #Skip Headers
    next(f) #Skip newline
    for line in f:
        fields = line.split()

        #Extracting variables...
        src_ip = fields[3]
        src_port = fields[4]
        dst_ip = fields[6]
        dst_port = fields[7]
        protocol = fields[8]
        num_packets = int(fields[10])
        num_bytes = int(fields[11])

        if (src_ip if source else dst_ip) == victim_ip:
```

```
            #Questions 1 and 4
            if mostPackets < num_packets:
                mostPackets = num_packets
                most_packet_flow = fields
            #Questions 2 and 5
            if mostBytes < num_bytes:
                mostBytes = num_bytes
                most_byte_flow = fields
            #Questions 3 and 6
            # a-c
            total_flows += 1
            total_packets += num_packets
            total_bytes += num_bytes
            # d-l
            uniq_srcport[src_port] += 1
            uniq_dstport[dst_port] += 1
            uniq_ip[dst_ip if source else src_ip] += 1
            uniq_packet[dst_ip if source else src_ip] += num_packets
            uniq_byte[dst_ip if source else src_ip] += num_bytes
            uniq_protocol[protocol] += 1

    return [victim_ip, most_packet_flow, most_byte_flow, total_flows,
            total_packets, total_bytes, uniq_srcport, uniq_ip,
            uniq_dstport, uniq_protocol, uniq_packet, uniq_byte]
#Formatting methods...
def printItems(source, l):
    string = ''
    if source:
```

```
        s = 'Source'
        d = 'Destination'
        q1, q2, q3 = '1', '2', '3'
    else:
        s = 'Destination'
        d = 'Source'
        q1, q2, q3 = '4', '5', '6'

    string += f'{s} IP Adress {l[0]}:\n\n'
    string += f'{q1}) Largest Packet Count Flow: \n\n{flowToString(l[1])}\n\n'
    string += f'{q2}) Largest Byte Count Flow: \n\n{flowToString(l[2])}\n\n'
    string += f'{q3}) For all records with {s} IP Adress {l[0]}:\n\n'
    string += f'\ta) Total flows              - {l[3]}\n'
    string += f'\tb) Total packets            - {l[4]}\n'
    string += f'\tc) Total bytes              - {l[5]}\n'
    if source:
        string += f'\td) Unique soruce ports      - {len(l[6])}\n'
        string += f'\te) Unique {d} IPs   - {len(l[7])}\n'
    else:
        string += f'\td) Unique {d} IPs        - {len(l[7])}\n'
        string += f'\te) Unique soruce ports      - {len(l[6])}\n'
    string += f'\tf) Unique Destination Ports - {len(l[8])}\n'
    string += f'\tg) Unique Protocols         - {len(l[9])}\n'
    string += f'\th) Top 5 Source Ports based on flow counts:
{portFreq(l[6].most_common(5))}\n'
    string += f'\ti) Top 5 Destination Ports based on flow counts:
{portFreq(l[8].most_common(5))}\n'
    string += f'\tj) Top 5 {d} IP addresses based on flow counts:
{ipFreq(l[7].most_common(5), "Flow")}\n'
```

```
    string += f'\tk) Top 5 {d} IP addresses based on packet counts:
{ipFreq(l[10].most_common(5), "Packet")}\n'
    string += f'\tl) Top 5 {d} IP addresses based on byte counts:
{ipFreq(l[11].most_common(5), "Byte")}\n'
    print(string)


def portFreq(list):
    s = '-' * 28
    string = f'\n\t\tPort  | Protocol | Frequency\n\t\t{s}\n'

    for port, freq in list:
        try:
            protocol_name = getservbyport(int(port))
        except OSError:
            protocol_name = "Unknown"
        string += f'\t\t{port:<6}| {protocol_name:<8} | {freq}\n'
    return string


def ipFreq(list, s):
    s2 = '-' * (23 + len(s))
    string = f'\n\t\tIP Address     | {s} Count\n\t\t{s2}\n'
    for address, freq in list:
        string += f'\t\t{address:<15}| {freq}\n'
    return string


def flowToString(flow):
    return f'{flow[0]} {flow[1]} {flow[2]} {flow[3]} {flow[4]} {flow[5]} {flow[6]} {flow[7]} {flow[8]}
{flow[9]} {flow[10]} {flow[11]} '
```

```
if __name__ == '__main__':
    main()
```