ACO331 Project 3 – Jack Sharkey

1. **Top 3 Source IP Addresses by flow count**:

    192.168.128.3 | 151298

    192.168.122.52 | 47415

    192.168.125.17 | 42699

2. These IP Addresses are private IP Addresses that were assigned to users in the network.

3. Source IP Cluster1.txt - 192.168.128.3:

    a) 151298 flows

    b) 7526055 packets

    c) 6840786276 bytes

    d) Source Port relative uncertainty: 0.86025

    e) Destination IP relative uncertainty: 0.42488

    f) Destination Port relative uncertainty: 0.10534

    Source IP Cluster2.txt - 192.168.122.52:

    a) 47415 flows

    b) 1319428 packets

    c) 1018225772 bytes

    d) Source Port relative uncertainty: 0.87552

    e) Destination IP relative uncertainty: 0.48674

    f) Destination Port relative uncertainty: 0.06985

    Source IP Cluster3.txt - 192.168.125.17:

    a) 42699 flows

    b) 1722879 packets

    c) 2204671814 bytes

    d) Source Port relative uncertainty: 0.75952

    e) Destination IP relative uncertainty: 0.65257

    f) Destination Port relative uncertainty: 0.26226

4. **Top 3 Source Ports by flow count**:

   6881 | 35414 - commonly used for P2P applications like BitTorrent.

   37450 | 30789

   53116 | 13569

5. ports 37450 and 53116 are often used with time-sensitive applications like gaming or audio/video streaming.

6. Source Port Cluster1.txt - 6881:

   a) 35414 flows

   b) 6754614 packets

   c) 4594105706 bytes

   d) Source IP relative uncertainty: 0.17862

   e) Destination IP relative uncertainty: 0.95800

   f) Destination Port relative uncertainty: 0.71692

   Source Port Cluster2.txt - 37450:

   a) 30789 flows

   b) 1440096 packets

   c) 921493365 bytes

   d) Source IP relative uncertainty: 0.20950

   e) Destination IP relative uncertainty: 0.94100

   f) Destination Port relative uncertainty: 0.79653

   Source Port Cluster3.txt - 53116:

   a) 13569 flows

   b) 52791 packets

   c) 34659371 bytes

   d) Source IP relative uncertainty: 0.00891

   e) Destination IP relative uncertainty: 0.99897

   f) Destination Port relative uncertainty: 0.00631

7. **Top 3 Destination IP Addresses by flow count**:

    172.16.255.200 | 741899

    172.16.255.183 | 147765

    172.16.141.250 | 135485

8. These IP Addresses are also private IP Addresses from within the network.

9. Destination IP Cluster1.txt - 172.16.255.200:

    a) 741899 flows

    b) 1894161 packets

    c) 189141046 bytes

    d) Source IP relative uncertainty: 0.43399

    e) Source Port relative uncertainty: 0.75463

    f) Destination Port relative uncertainty: 0.00483

    Destination IP Cluster2.txt - 172.16.255.183:

    a) 147765 flows

    b) 376401 packets

    c) 37424405 bytes

    d) Source IP relative uncertainty: 0.44843

    e) Source Port relative uncertainty: 0.85845

    f) Destination Port relative uncertainty: 0.01308

    Destination IP Cluster3.txt - 172.16.141.250:

    a) 135485 flows

    b) 1256612 packets

    c) 639933866 bytes

    d) Source IP relative uncertainty: 0.41955

    e) Source Port relative uncertainty: 0.77904

    f) Destination Port relative uncertainty: 0.04148

10. **Top 3 Destination Ports by flow count:**

    443 | 1039599 – This port is for HTTPS.

    53 | 909590 – This port is for DNS.

    80 | 179414 – This port is for HTTP.

11. Answered above

12. Destination Port Cluster1.txt - 443:

    a) 1039599 flows

    b) 137124792 packets

    c) 161483705992 bytes

    d) Source IP relative uncertainty: 0.41334

    e) Source Port relative uncertainty: 0.72598

    f) Destination IP relative uncertainty: 0.51043

    Destination Port Cluster2.txt - 53:

    a) 909590 flows

    b) 2082152 packets

    c) 216212802 bytes

    d) Source IP relative uncertainty: 0.42594

    e) Source Port relative uncertainty: 0.74794

    f) Destination IP relative uncertainty: 0.04629

    Destination Port Cluster3.txt - 80:

    a) 179414 flows

    b) 43530844 packets

    c) 63878920434 bytes

    d) Source IP relative uncertainty: 0.46705

    e) Source Port relative uncertainty: 0.80372

    f) Destination IP relative uncertainty: 0.53847

Script/Code

This is the script I wrote to analyze the data:

```
from collections import Counter
import math

def main():
    inputFile = 'Unicauca-dataset-April-June-2019-Network-flows.txt'
    #Top 3 most common
    print('Finding top 3 most common source ip, port, destination ip, port...')
    src_ip_mc, src_port_mc, dst_ip_mc, dst_port_mc = mostCommon(inputFile)

    #Create clusters for each

    print('Creating clusters for each...')
    createClusters(src_ip_mc, inputFile, 10, 'Source IP')
    print('Source IP clusters created.')
    createClusters(src_port_mc, inputFile, 11, 'Source Port')
    print('Source Port clusters created.')
    createClusters(dst_ip_mc, inputFile, 12, 'Destination IP')
    print('Destination IP clusters created.')
    createClusters(dst_port_mc, inputFile, 13, 'Destination Port')
    print('Destination Port clusters created.')

    #Process Clusters
    print('Processing Clusters...')
    index = 10
    mapping = {
        'Source IP' : ['Source Port', 'Destination IP', 'Destination Port'],
        'Source Port' : ['Source IP', 'Destination IP', 'Destination Port'],
        'Destination IP' : ['Source IP', 'Source Port', 'Destination Port'],
        'Destination Port' : ['Source IP', 'Source Port', 'Destination IP']
    }
    names = ['Source IP Addresses', 'Source Ports', 'Destination IP Addresses',
'Destination Ports']
    j = 0
    for values in [src_ip_mc, src_port_mc, dst_ip_mc, dst_port_mc]:
        print(f'Top 3 {names[j]} by flow count:')
        for x, freq in values:
            print(f'\t{x} | {freq}')
        print()
        j += 1
    for s in ['Source IP', 'Source Port', 'Destination IP', 'Destination Port']:
        x, y, z = mapping[s]
        for i in range(3):
            a, b, c, d, e, f, fixed = processCluster(f'{s} Cluster{i+1}.txt',
index)
```

```
            print(f'{s} Cluster{i+1}.txt - {fixed}:\n\ta) {a} flows\n\tb) {b}
packets\n\tc) {c} bytes\n\td) {x} relative uncertainty: {d:.5f}\n\te) {y} relative
uncertainty: {e:.5f}\n\tf) {z} relative uncertainty: {f:.5f}\n')
        index += 1

def createClusters(top3freq, inputFile, check, name):
    top3 = [x for x, _ in top3freq]
    cluster_files = []

    # Open files for writing clusters
    for i in range(3):
        filename = f'{name} Cluster{i+1}.txt'
        cluster_files.append(open(filename, 'w'))

    with open(inputFile, 'r') as f:
        for line in f:
            fields = line.split()
            x = fields[check]

            if x in top3:
                index = top3.index(x)
                cluster_files[index].write(line)

    # Close all cluster files
    for file in cluster_files:
        file.close()

def mostCommon(inputFile):

    uniq_srcip = Counter()
    uniq_srcport = Counter()
    uniq_dstip = Counter()
    uniq_dstport = Counter()

    with open(inputFile, 'r') as f:
        for line in f:
            fields = line.split()

            #Extracting variables...
            src_ip = fields[10]
            src_port = fields[11]
            dst_ip = fields[12]
            dst_port = fields[13]
            #Mapping frequency
            uniq_srcip[src_ip] += 1
            uniq_srcport[src_port] += 1
            uniq_dstip[dst_ip] += 1
            uniq_dstport[dst_port] += 1

    return uniq_srcip.most_common(3), uniq_srcport.most_common(3),
uniq_dstip.most_common(3), uniq_dstport.most_common(3)

def processCluster(inFile, fieldIndex):
    mapping = {
        10: (11, 12, 13),
        11: (10, 12, 13),
        12: (10, 11, 13),
        13: (10, 11, 12)
    }

    var1, var2, var3 = mapping[fieldIndex]
    flowCount, packetCount, byteCount = 0, 0, 0
    v1, v2, v3 = [], [], []
    with open(inFile, 'r') as f:
        for line in f:
            fields = line.split()

            fixed = fields[fieldIndex]
            v1.append(fields[var1])
            v2.append(fields[var2])
            v3.append(fields[var3])
            packet = fields[15]
            byte = fields[16]

            flowCount += 1
            packetCount += int(packet)
            byteCount += int(byte)
    return flowCount, packetCount, byteCount, cal_entropy(v1), cal_entropy(v2),
cal_entropy(v3), fixed

def cal_entropy(values):
    total = 0
    entropy = 0
    freq_dict = {}

    for value in values:
        value = value.rstrip()
        total += 1

        if value in freq_dict:
           freq_dict[value] += 1
        else:
           freq_dict[value] = 1

    for v in freq_dict:
        f = freq_dict[v]
        prob = f / total
        entropy += -1 * prob * math.log(prob, 2)

    max_entropy = math.log(total, 2)

    std_entropy = entropy / max_entropy

    return std_entropy

if __name__ == '__main__':
```

```
main()
```