Timestamp,Src_IP,Dest_IP,Protocol,Source_Port,Destination_Port,Packet_Size,Status

2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted 2024-09-30 10:15:12,192.168.1.15,203.0.113.5,UDP,53,55432,512,Dropped 2024-09-30 10:15:14,10.0.0.2,192.168.1.10,TCP,80,61324,1420,Accepted 2024-09-30 10:15:16,172.16.0.5,192.168.1.15,ICMP,64,Dropped 2024-09-30 10:15:18,192.168.1.10,198.51.100.23,TCP,443,1025,1500,Accepted 2024-09-30 10:15:20,198.51.100.23,192.168.1.10,TCP,443,1025,1400,Accepted 2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped

Src IP – Source Ip

Dest Ip – Destination Ip

Write a awk script to

1. **count how many packets use each protocol (TCP, UDP, ICMP)**

```
BEGIN { FS="," }
NR > 1 {
  protocols[$4]++
}
END {
  for (protocol in protocols) {
    print protocol ": " protocols[protocol]
  }
}
```

2. **filter and print only the dropped packets.**

```
BEGIN { FS="," }
NR > 1 && $8 == "Dropped" {
  print $0
}
```

3. **print the Timestamp, Source_IP, Destination_IP, and Packet_Size for packets that have a size greater than 1000 bytes.**

```
BEGIN { FS="," }
NR > 1 && $7 > 1000 {
  print $1 "," $2 "," $3 "," $7
}
```

4. **display traffic that is directed to destination port 443.**

```
BEGIN { FS="," }
NR > 1 && $6 == 443 {
  print $0
}
```

5. **print all unique Source_IP addresses from the network_traffic.csv file.**

```
BEGIN { FS="," }
NR > 1 {
  if (!($2 in seen)) {
```

```
        seen[$2] = 1
        print $2
    }
}
```

6. **filter only TCP traffic and calculate the average packet size.**

```
BEGIN { FS="," }
NR > 1 && $4 == "TCP" {
    sum += $7
    count++
}
END {
    if (count > 0) {
        print "Average TCP packet size: " sum/count
    } else {
        print "No TCP packets found"
    }
}
```

7. **Count invalid records**

```
BEGIN { FS="," }
NR > 1 && NF != 8 {
    invalid++
}
END {
    print "Invalid records: " invalid
}
```

8. **extract and print all rows where the Source_IP is in the 192.168.x.x range.**

```
BEGIN { FS="," }
NR > 1 && $2 ~ /^192\.168\./ {
    print $0
}
```

9. **match traffic directed to either port 80 (HTTP) or port 443 (HTTPS).**

```
BEGIN { FS="," }
NR > 1 && ($6 == 80 || $6 == 443) {
    print $0
}
```

10. **filter out rows where the Destination_Port contains any alphanumeric characters (letters or numbers).**

```
BEGIN { FS="," }
NR > 1 && $6 ~ /^[0-9]+$/ {
    print $0
}
```

11. **filter out traffic where the protocol is TCP AND the destination port is 443 (HTTPS traffic).**

```
BEGIN { FS="," }
NR > 1 && !($4 == "TCP" && $6 == 443) {
```

```
  print $0
}
```

12. filter out and print traffic where the Packet_Size is greater than 1000 OR the Status is Dropped.

```
BEGIN { FS="," }
NR > 1 && ($7 > 1000 || $8 == "Dropped") {
  print $0
}
```

13. print traffic NOT originating from 192.168.x.x IP addresses.

```
BEGIN { FS="," }
NR > 1 && $2 !~ /^192\.168\./ {
  print $0
}
```

14. filter rows where both Source_IP and Destination_IP are within the 192.168.x.x range.

```
BEGIN { FS="," }
NR > 1 && $2 ~ /^192\.168\./ && $3 ~ /^192\.168\./ {
  print $0
}
```

15. filter out traffic where the destination port is 22 OR the packet size is less than 100 bytes.

```
BEGIN { FS="," }
NR > 1 && !($6 == 22 || $7 < 100) {
  print $0
}
```