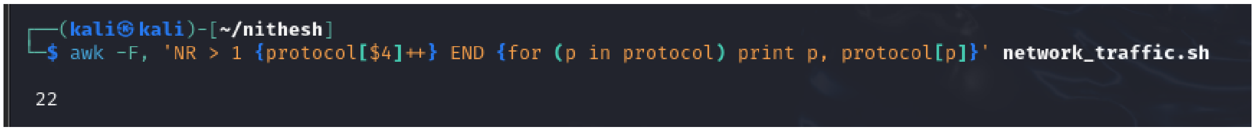
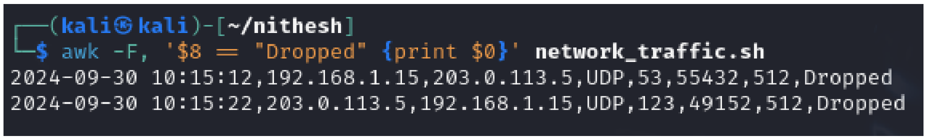
1. count how many packets use each protocol (TCP, UDP, ICMP)



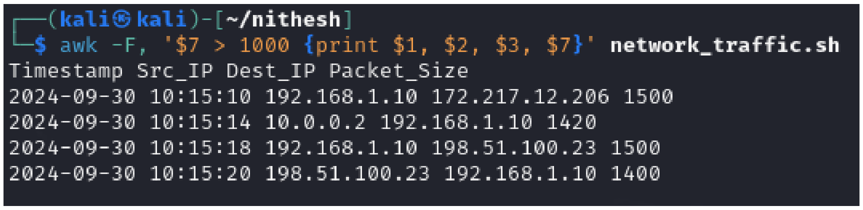
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, 'NR > 1 {protocol[$4]++} END {for (p in protocol) print p, protocol[p]}' network_traffic.sh

 22
```

2. filter and print only the dropped packets.



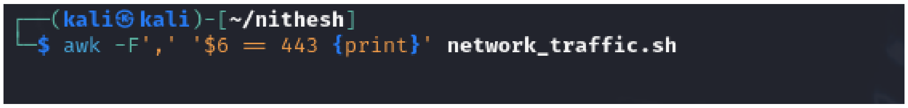
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$8 == "Dropped" {print $0}' network_traffic.sh
2024-09-30 10:15:12,192.168.1.15,203.0.113.5,UDP,53,55432,512,Dropped
2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped
```

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



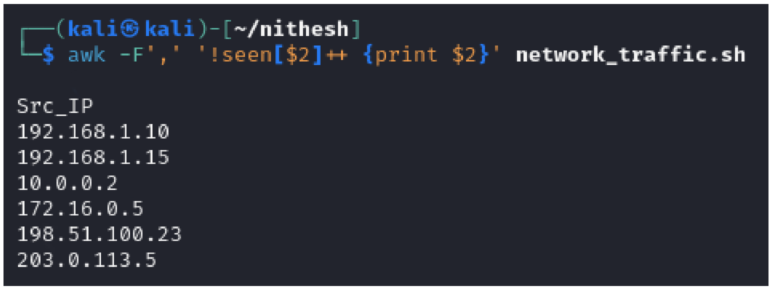
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$7 > 1000 {print $1, $2, $3, $7}' network_traffic.sh
Timestamp Src_IP Dest_IP Packet_Size
2024-09-30 10:15:10 192.168.1.10 172.217.12.206 1500
2024-09-30 10:15:14 10.0.0.2 192.168.1.10 1420
2024-09-30 10:15:18 192.168.1.10 198.51.100.23 1500
2024-09-30 10:15:20 198.51.100.23 192.168.1.10 1400
```

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



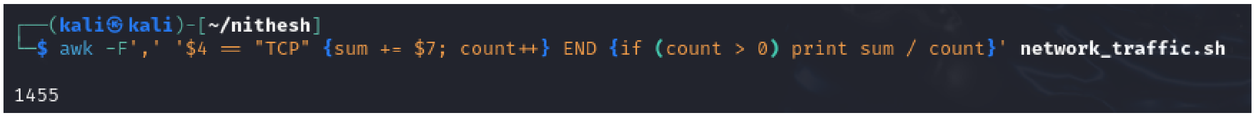
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F',' '$6 == 443 {print}' network_traffic.sh
```

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



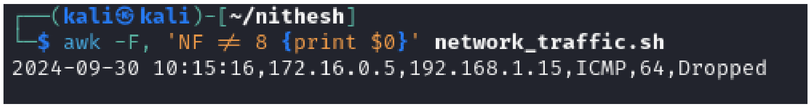
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F',' '!seen[$2]++ {print $2}' network_traffic.sh

Src_IP
192.168.1.10
192.168.1.15
10.0.0.2
172.16.0.5
198.51.100.23
203.0.113.5
```

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



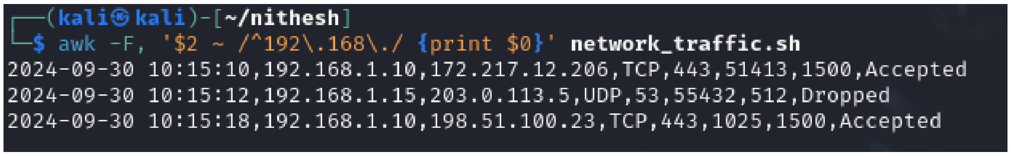
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F',' '$4 == "TCP" {sum += $7; count++} END {if (count > 0) print sum / count}' network_traffic.sh

1455
```

7. Count invalid records



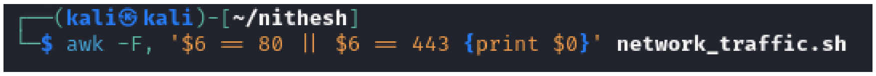
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, 'NF ≠ 8 {print $0}' network_traffic.sh
2024-09-30 10:15:16,172.16.0.5,192.168.1.15,ICMP,64,Dropped
```

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



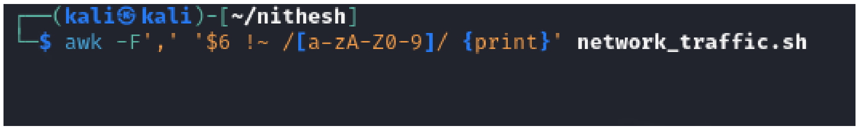
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$2 ~ /^192\.168\./ {print $0}' network_traffic.sh
2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted
2024-09-30 10:15:12,192.168.1.15,203.0.113.5,UDP,53,55432,512,Dropped
2024-09-30 10:15:18,192.168.1.10,198.51.100.23,TCP,443,1025,1500,Accepted
```

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



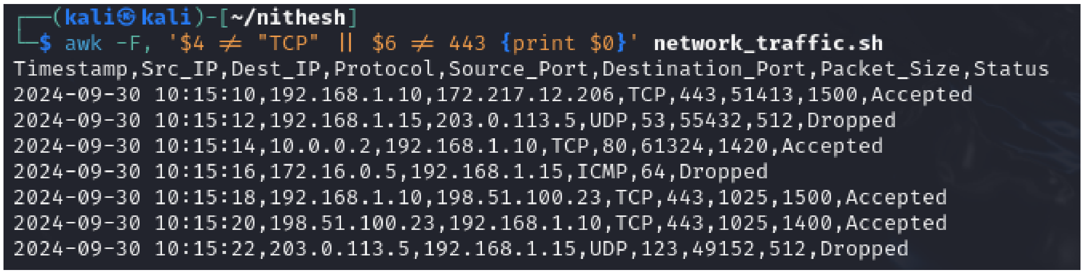
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$6 == 80 || $6 == 443 {print $0}' network_traffic.sh
```

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



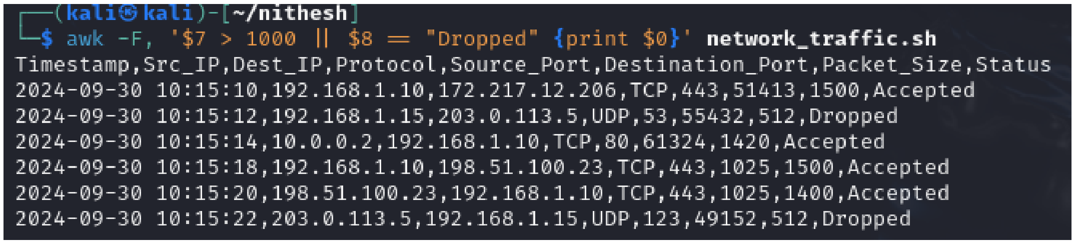
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F',' '$6 !~ /[a-zA-Z0-9]/ {print}' network_traffic.sh
```

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



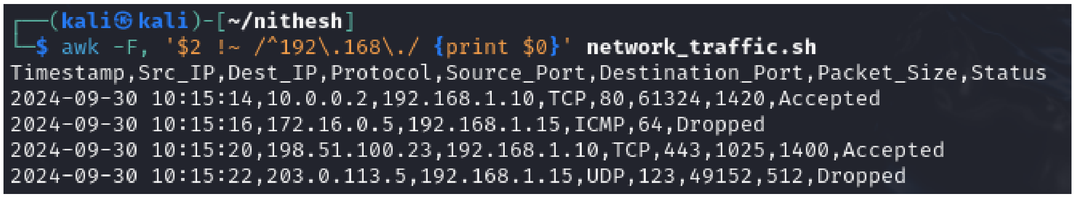
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$4 ≠ "TCP" || $6 ≠ 443 {print $0}' network_traffic.sh
Timestamp,Src_IP,Dest_IP,Protocol,Source_Port,Destination_Port,Packet_Size,Status
2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted
2024-09-30 10:15:12,192.168.1.15,203.0.113.5,UDP,53,55432,512,Dropped
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
```

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



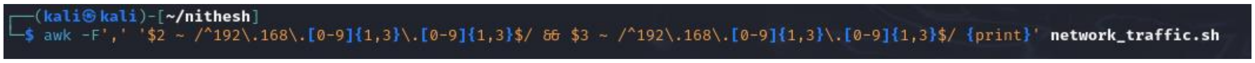
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$7 > 1000 || $8 == "Dropped" {print $0}' network_traffic.sh
Timestamp,Src_IP,Dest_IP,Protocol,Source_Port,Destination_Port,Packet_Size,Status
2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted
2024-09-30 10:15:12,192.168.1.15,203.0.113.5,UDP,53,55432,512,Dropped
2024-09-30 10:15:14,10.0.0.2,192.168.1.10,TCP,80,61324,1420,Accepted
2024-09-30 10:15:18,192.168.1.10,198.51.100.23,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
```

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



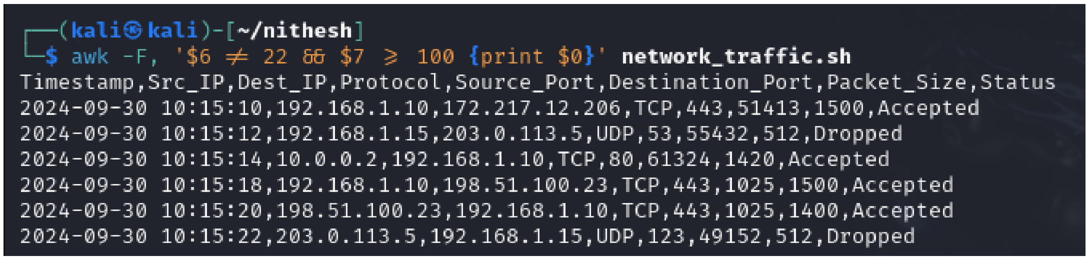
```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$2 !~ /^192\.168\./ {print $0}' network_traffic.sh
Timestamp,Src_IP,Dest_IP,Protocol,Source_Port,Destination_Port,Packet_Size,Status
2024-09-30 10:15:14,10.0.0.2,192.168.1.10,TCP,80,61324,1420,Accepted
2024-09-30 10:15:16,172.16.0.5,192.168.1.15,ICMP,64,Dropped
2024-09-30 10:15:20,198.51.100.23,192.168.1.10,TCP,443,1025,1400,Accepted
2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped
```

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



```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F',' '$2 ~ /^192\.168\.[0-9]{1,3}\.[0-9]{1,3}$/ && $3 ~ /^192\.168\.[0-9]{1,3}\.[0-9]{1,3}$/ {print}' network_traffic.sh
```

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



```
┌──(kali㉿kali)-[~/nithesh]
└─$ awk -F, '$6 ≠ 22 && $7 ≥ 100 {print $0}' network_traffic.sh
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 10:15:18,192.168.1.10,198.51.100.23,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
```