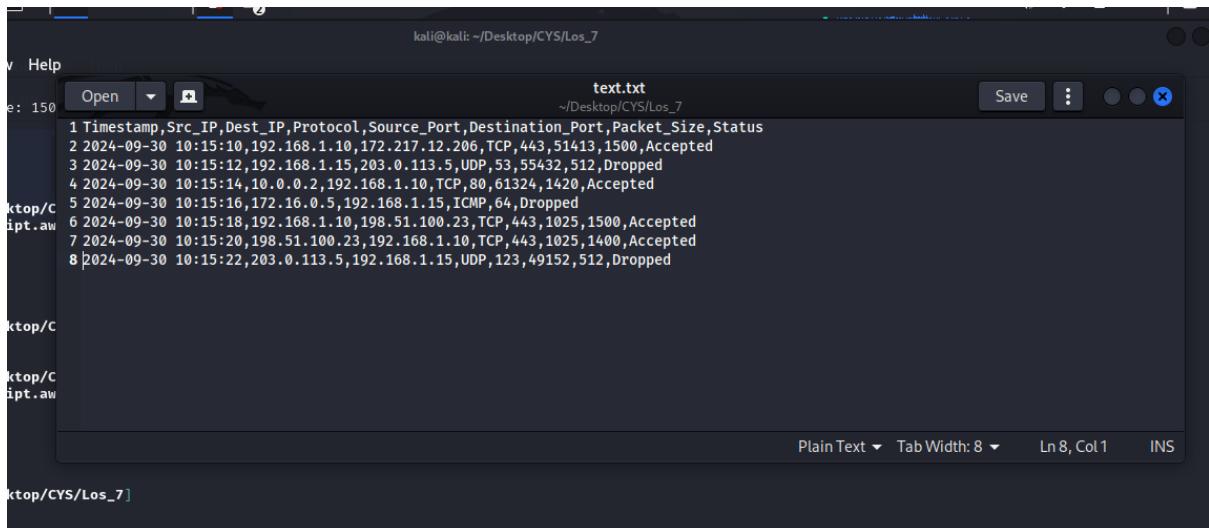


AWK Scripts

Network Packets:

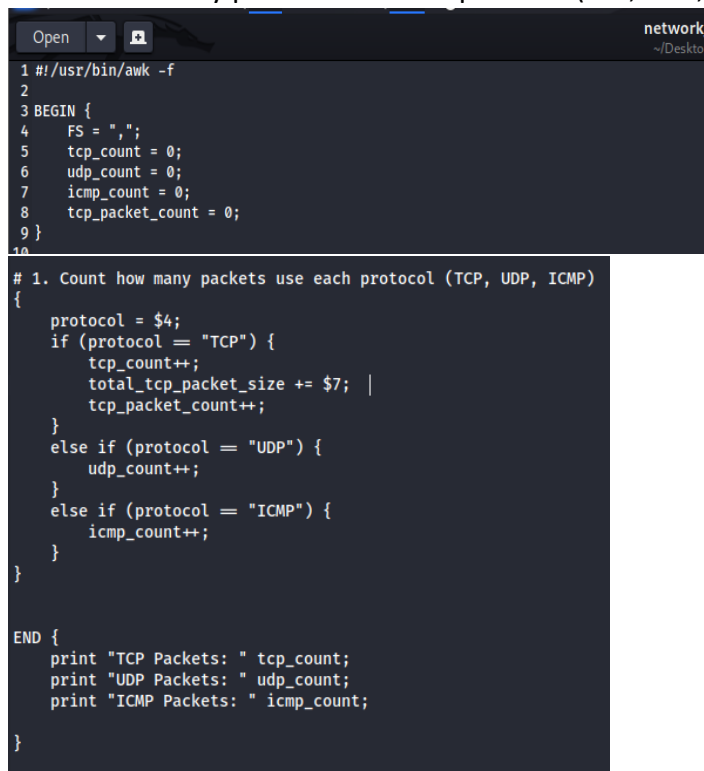


A screenshot of a text editor window titled 'text.txt' located at '~/.Desktop/CYS/Los_7'. The editor displays a list of network packets with the following columns: Timestamp, Src_IP, Dest_IP, Protocol, Source_Port, Destination_Port, Packet_Size, and Status. The data is as follows:

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

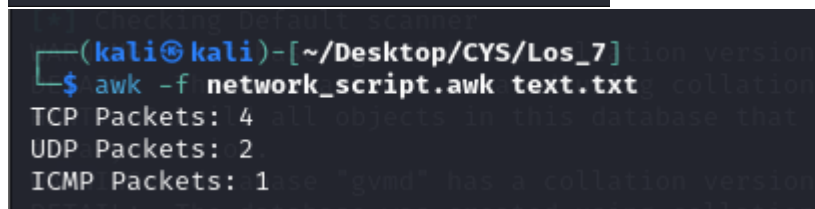
Write a awk script to

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



A screenshot of a text editor window titled 'network_script.awk' located at '~/.Desktop/CYS/Los_7'. The script is as follows:

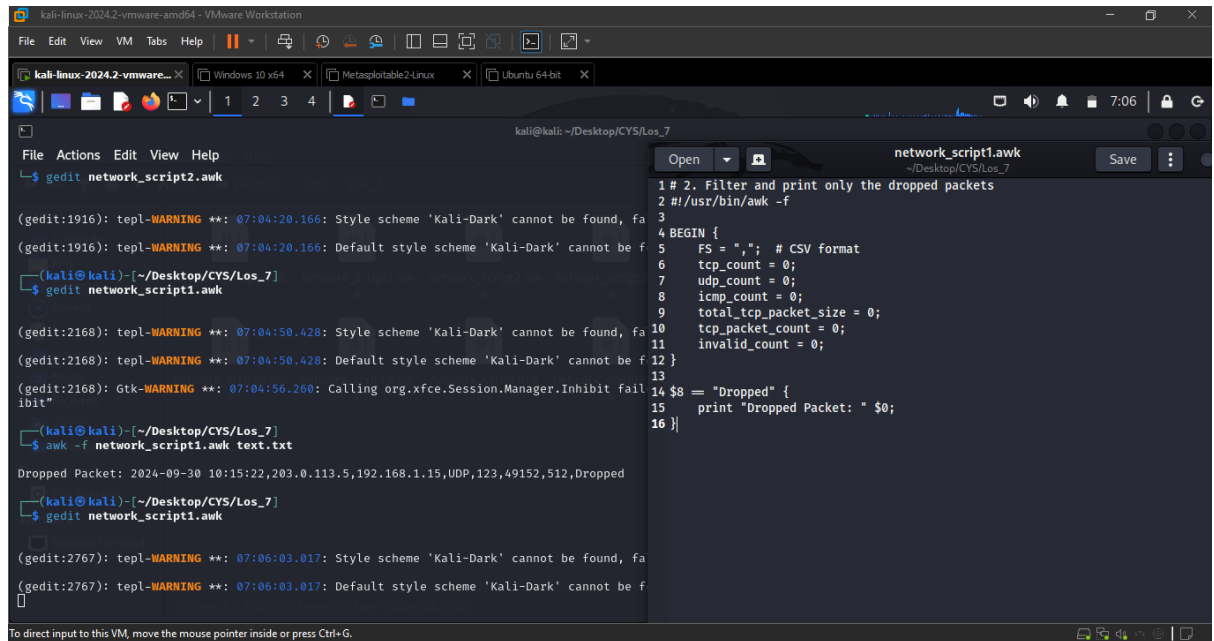
```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ",";
5     tcp_count = 0;
6     udp_count = 0;
7     icmp_count = 0;
8     tcp_packet_count = 0;
9 }
10
11 # 1. Count how many packets use each protocol (TCP, UDP, ICMP)
12 {
13     protocol = $4;
14     if (protocol == "TCP") {
15         tcp_count++;
16         total_tcp_packet_size += $7; |
17         tcp_packet_count++;
18     }
19     else if (protocol == "UDP") {
20         udp_count++;
21     }
22     else if (protocol == "ICMP") {
23         icmp_count++;
24     }
25 }
26
27 END {
28     print "TCP Packets: " tcp_count;
29     print "UDP Packets: " udp_count;
30     print "ICMP Packets: " icmp_count;
31 }
```



A screenshot of a terminal window showing the execution of the AWK script. The prompt is '(kali@kali)-[~/Desktop/CYS/Los_7]'. The command entered is '\$ awk -f network_script.awk text.txt'. The output is:

```
TCP Packets: 4
UDP Packets: 2
ICMP Packets: 1
```

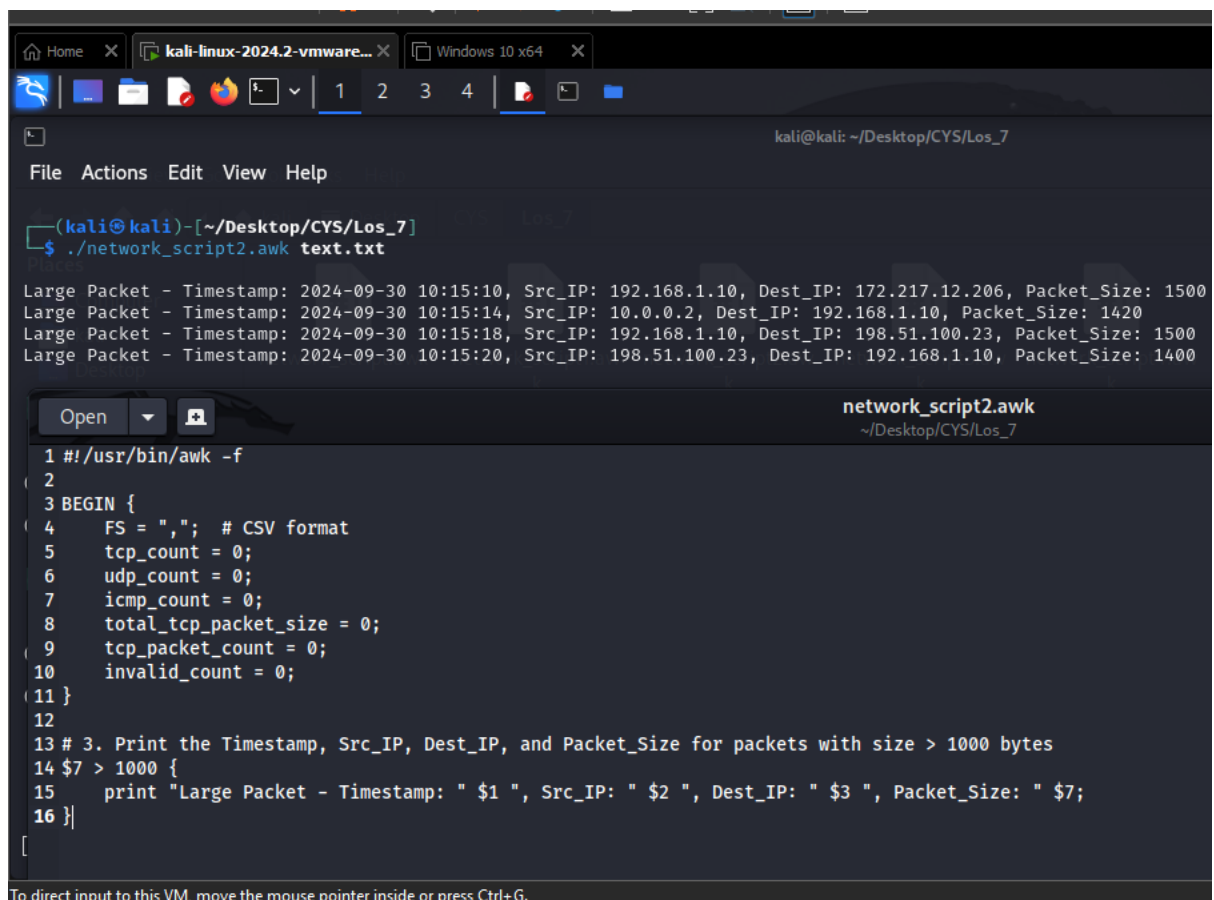
2. Filter and print only the dropped packets.



The screenshot shows a Kali Linux VM terminal window. The user is in the directory `~/Desktop/CYS/Los_7`. They have opened two files: `network_script2.awk` and `network_script1.awk`. The terminal output shows several warning messages from `tepl` and `Gtk` regarding the 'Kali-Dark' style scheme. The user has executed the command `awk -f network_script1.awk text.txt`, which produced the output: `Dropped Packet: 2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped`.

```
kali@kali: ~/Desktop/CYS/Los_7
$ gedit network_script2.awk
(gedit:1916): tepl-WARNING **: 07:04:20.166: Style scheme 'Kali-Dark' cannot be found, fa
(gedit:1916): tepl-WARNING **: 07:04:20.166: Default style scheme 'Kali-Dark' cannot be f
(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script1.awk
(gedit:2168): tepl-WARNING **: 07:04:50.428: Style scheme 'Kali-Dark' cannot be found, fa
(gedit:2168): tepl-WARNING **: 07:04:50.428: Default style scheme 'Kali-Dark' cannot be f
(gedit:2168): Gtk-WARNING **: 07:04:56.260: Calling org.xfce.Session.Manager.Inhibit fail
ibit"
(kali@kali)~/Desktop/CYS/Los_7
$ awk -f network_script1.awk text.txt
Dropped Packet: 2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped
(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script1.awk
(gedit:2767): tepl-WARNING **: 07:06:03.017: Style scheme 'Kali-Dark' cannot be found, fa
(gedit:2767): tepl-WARNING **: 07:06:03.017: Default style scheme 'Kali-Dark' cannot be f
```

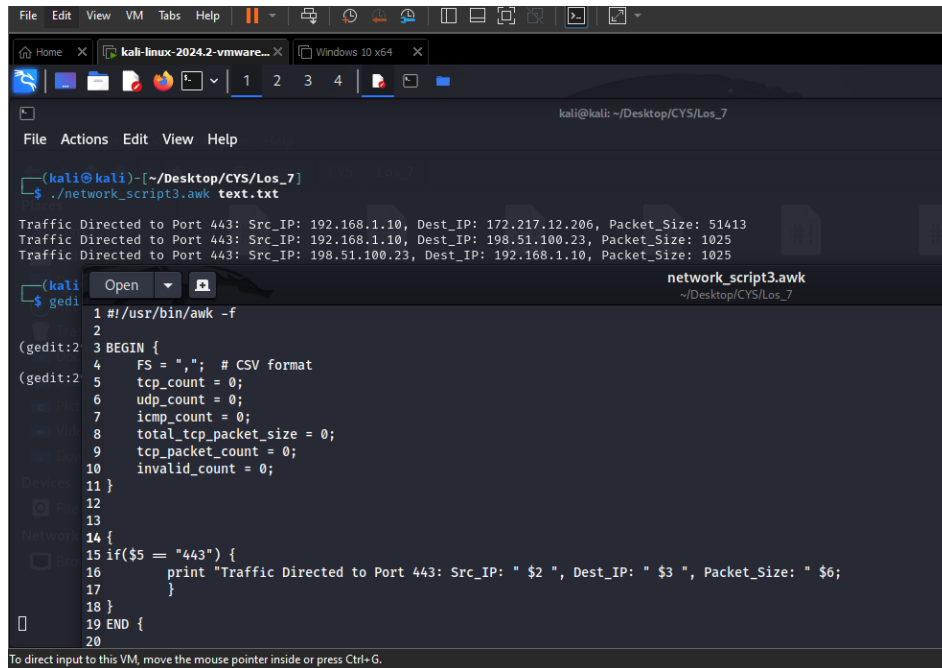
3. Print the Timestamp, Source_IP, Destination_IP, and Packet_Size for packets that have a size greater than 1000 bytes.



The screenshot shows a Kali Linux VM terminal window. The user is in the directory `~/Desktop/CYS/Los_7`. They have executed the command `./network_script2.awk text.txt`, which produced the output: `Large Packet - Timestamp: 2024-09-30 10:15:10, Src_IP: 192.168.1.10, Dest_IP: 172.217.12.206, Packet_Size: 1500`, `Large Packet - Timestamp: 2024-09-30 10:15:14, Src_IP: 10.0.0.2, Dest_IP: 192.168.1.10, Packet_Size: 1420`, `Large Packet - Timestamp: 2024-09-30 10:15:18, Src_IP: 192.168.1.10, Dest_IP: 198.51.100.23, Packet_Size: 1500`, and `Large Packet - Timestamp: 2024-09-30 10:15:20, Src_IP: 198.51.100.23, Dest_IP: 192.168.1.10, Packet_Size: 1400`.

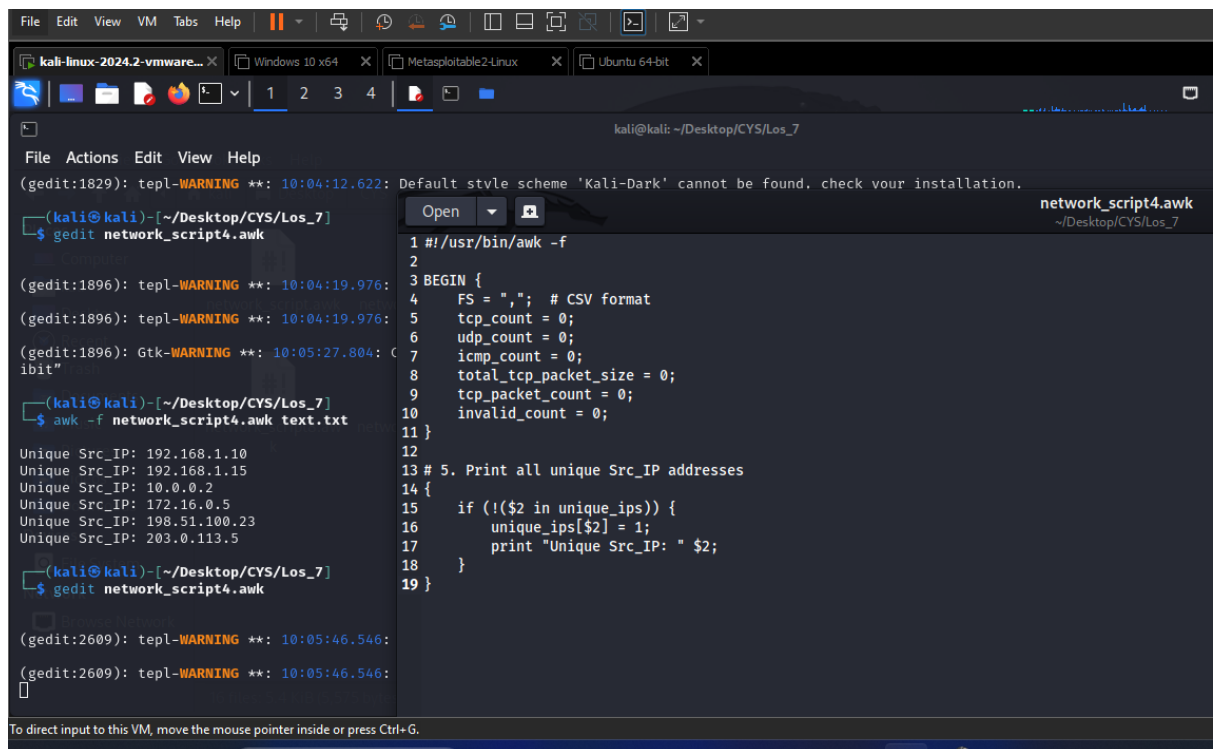
```
kali@kali: ~/Desktop/CYS/Los_7
File Actions Edit View Help
(kali@kali)~/Desktop/CYS/Los_7
$ ./network_script2.awk text.txt
Large Packet - Timestamp: 2024-09-30 10:15:10, Src_IP: 192.168.1.10, Dest_IP: 172.217.12.206, Packet_Size: 1500
Large Packet - Timestamp: 2024-09-30 10:15:14, Src_IP: 10.0.0.2, Dest_IP: 192.168.1.10, Packet_Size: 1420
Large Packet - Timestamp: 2024-09-30 10:15:18, Src_IP: 192.168.1.10, Dest_IP: 198.51.100.23, Packet_Size: 1500
Large Packet - Timestamp: 2024-09-30 10:15:20, Src_IP: 198.51.100.23, Dest_IP: 192.168.1.10, Packet_Size: 1400
(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script2.awk
network_script2.awk
~/Desktop/CYS/Los_7
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","; # CSV format
5     tcp_count = 0;
6     udp_count = 0;
7     icmp_count = 0;
8     total_tcp_packet_size = 0;
9     tcp_packet_count = 0;
10    invalid_count = 0;
11 }
12
13 # 3. Print the Timestamp, Src_IP, Dest_IP, and Packet_Size for packets with size > 1000 bytes
14 $7 > 1000 {
15     print "Large Packet - Timestamp: " $1 ", Src_IP: " $2 ", Dest_IP: " $3 ", Packet_Size: " $7;
16 }
```

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



```
File Edit View VM Tabs Help
kali-linux-2024.2-vmware... Windows 10 x64
kali@kali: ~/Desktop/CYS/Los_7
File Actions Edit View Help
(kali@kali)~[~/Desktop/CYS/Los_7]
$ ./network_script3.awk text.txt
Traffic Directed to Port 443: Src_IP: 192.168.1.10, Dest_IP: 172.217.12.206, Packet_Size: 51413
Traffic Directed to Port 443: Src_IP: 192.168.1.10, Dest_IP: 198.51.100.23, Packet_Size: 1025
Traffic Directed to Port 443: Src_IP: 198.51.100.23, Dest_IP: 192.168.1.10, Packet_Size: 1025
(kali@kali)~[~/Desktop/CYS/Los_7]
$ gedit
network_script3.awk
~/Desktop/CYS/Los_7
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","; # CSV format
5     tcp_count = 0;
6     udp_count = 0;
7     icmp_count = 0;
8     total_tcp_packet_size = 0;
9     tcp_packet_count = 0;
10    invalid_count = 0;
11 }
12
13
14 {
15     if($5 == "443") {
16         print "Traffic Directed to Port 443: Src_IP: " $2 ", Dest_IP: " $3 ", Packet_Size: " $6;
17     }
18 }
19 END {
20 }
```

5. Print all unique Source_IP addresses from the text.txt file.



```
File Edit View VM Tabs Help
kali-linux-2024.2-vmware... Windows 10 x64 Metasploitable2-Linux Ubuntu 64-bit
kali@kali: ~/Desktop/CYS/Los_7
File Actions Edit View Help
(gedit:1829): tepl-WARNING **: 10:04:12.622: Default style scheme 'Kali-Dark' cannot be found. check your installation.
(kali@kali)~[~/Desktop/CYS/Los_7]
$ gedit network_script4.awk
network_script4.awk
~/Desktop/CYS/Los_7
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","; # CSV format
5     tcp_count = 0;
6     udp_count = 0;
7     icmp_count = 0;
8     total_tcp_packet_size = 0;
9     tcp_packet_count = 0;
10    invalid_count = 0;
11 }
12
13 # 5. Print all unique Src_IP addresses
14 {
15     if (!($2 in unique_ips)) {
16         unique_ips[$2] = 1;
17         print "Unique Src_IP: " $2;
18     }
19 }
```

```
(gedit:1896): tepl-WARNING **: 10:04:19.976:
(gedit:1896): tepl-WARNING **: 10:04:19.976:
(gedit:1896): Gtk-WARNING **: 10:05:27.804:
(kali@kali)~[~/Desktop/CYS/Los_7]
$ awk -f network_script4.awk text.txt
Unique Src_IP: 192.168.1.10
Unique Src_IP: 192.168.1.15
Unique Src_IP: 10.0.0.2
Unique Src_IP: 172.16.0.5
Unique Src_IP: 198.51.100.23
Unique Src_IP: 203.0.113.5
(kali@kali)~[~/Desktop/CYS/Los_7]
$ gedit network_script4.awk
(gedit:2609): tepl-WARNING **: 10:05:46.546:
(gedit:2609): tepl-WARNING **: 10:05:46.546:
```

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

The screenshot shows a Kali Linux virtual machine environment. In the terminal, the user runs `awk -f network_script5.awk text.txt`, which outputs statistics for TCP, UDP, and ICMP packets. Then, the user opens `network_script5.awk` in gedit. The script is an awk program that processes a file line by line, checks for TCP traffic (column 4), and calculates the average TCP packet size by dividing the total TCP packet size by the total packet count. The script ends with a print statement for the average packet size.

```
1#!/usr/bin/awk -f
2
3BEGIN {
4    FS = ",";
5    total_tcp_packet_size = 0;
6    tcp_packet_count = 0;
7}
8
9# Process each line in the file
10{
11    # Check if the protocol is TCP (column 4)
12    if ($4 == "TCP") {
13        total_tcp_packet_size += $7;
14        tcp_packet_count++;
15    }
16}
17END {
18    if (tcp_packet_count > 0) {
19        # Calculate and print the average TCP packet size
20        avg_packet_size = total_tcp_packet_size / tcp_packet_count;
21        print "Average TCP Packet Size: " avg_packet_size;
22    } else {
23        print "No TCP traffic found.";
24    }
25}
26
27
```

7. Count invalid records

The screenshot shows a Kali Linux virtual machine environment. In the terminal, the user runs `awk -f network_script6.awk text.txt`, which outputs the invalid records count. Then, the user opens `network_script6.awk` in gedit. The script is an awk program that processes a file line by line, checks for invalid records (column 8), and counts them. The script ends with a print statement for the invalid records count.

```
1#!/usr/bin/awk -f
2
3BEGIN {
4    FS = ",";
5    invalid_count = 0;
6}
7
8{
9    if (NF != 8) {
10        invalid_count++;
11    }
12}
13END {
14    print "Invalid Records Count: " invalid_count;
15}
16
17
18
```

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

The screenshot shows a Kali Linux virtual machine environment. The terminal window displays the execution of an awk script named `network_script7.awk`. The script filters network traffic based on the destination IP address, specifically targeting traffic to `192.168.x.x`. The terminal output shows several lines of network traffic data, including source and destination IP addresses, ports, and protocols. The text editor window shows the content of `network_script7.awk`, which is an awk script that filters traffic based on the destination IP address.

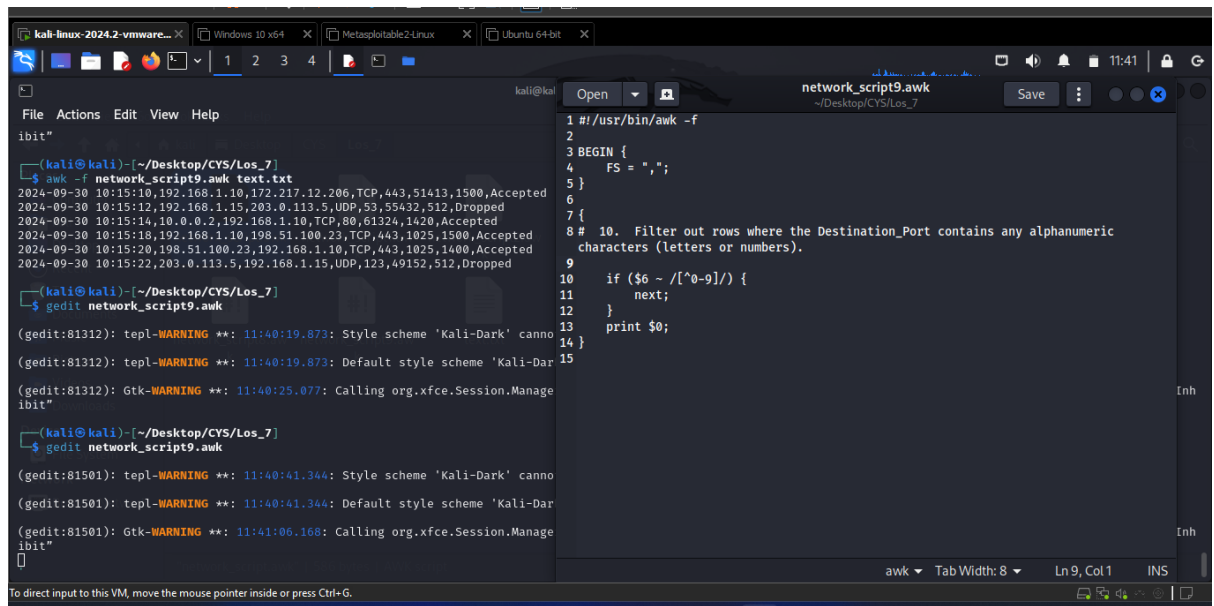
```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ",";
5 }
6
7 {
8     if ($2 ~ /^192\.\.168\./) {
9         print "192.168.x.x Src_IP: " $0;
10    }
11 }
12
```

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

The screenshot shows a Kali Linux virtual machine environment. The terminal window displays the execution of an awk script named `network_script8.awk`. The script filters network traffic based on the destination port, specifically targeting traffic to port 80 or 443. The terminal output shows several lines of network traffic data, including source and destination IP addresses, ports, and protocols. The text editor window shows the content of `network_script8.awk`, which is an awk script that filters traffic based on the destination port.

```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ",";
5 }
6
7 {
8     dest_port = $5
9
10    if (dest_port == 80 || dest_port == 443) {
11        print $0
12    }
13 }
14
15 END {
16 }
```

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



```
File Actions Edit View Help
ibit"
(kali@kali)~/Desktop/CYS/Los_7
$ awk -f network_script9.awk text.txt
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

(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script9.awk

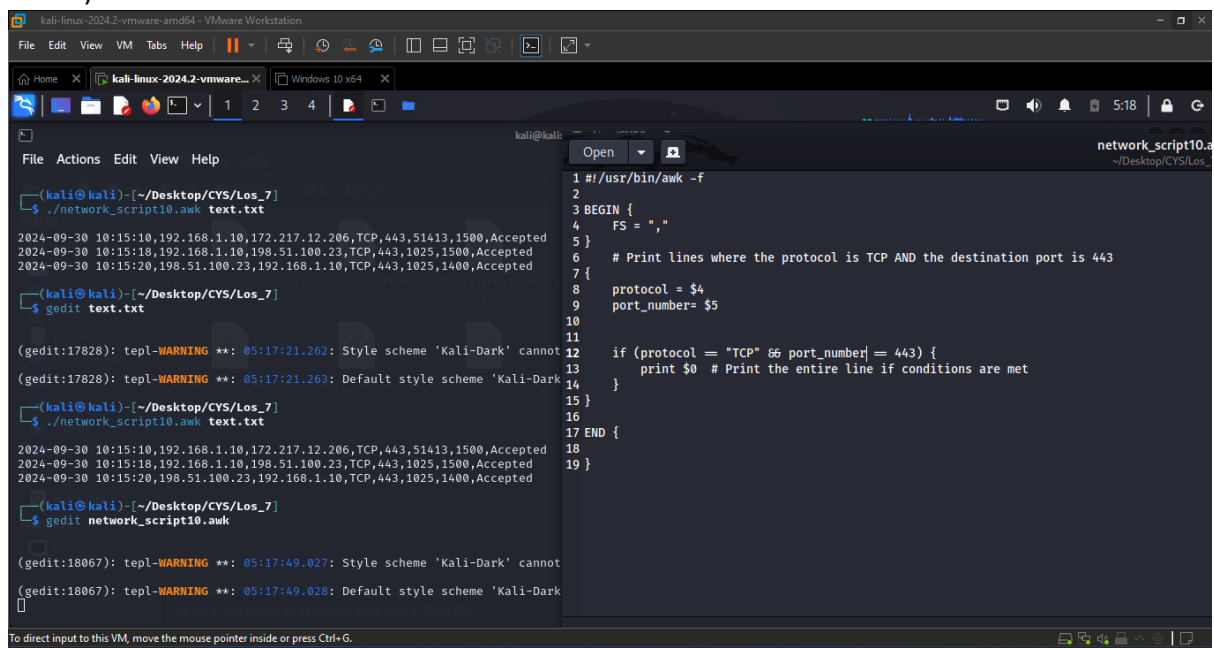
(gedit:81312): tepl-WARNING **: 11:40:19.873: Style scheme 'Kali-Dark' cannot
(gedit:81312): tepl-WARNING **: 11:40:19.873: Default style scheme 'Kali-Dar
(gedit:81312): Gtk-WARNING **: 11:40:25.077: Calling org.xfce.Session.Manage
ibit"

(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script9.awk

(gedit:81501): tepl-WARNING **: 11:40:41.344: Style scheme 'Kali-Dark' cannot
(gedit:81501): tepl-WARNING **: 11:40:41.344: Default style scheme 'Kali-Dar
(gedit:81501): Gtk-WARNING **: 11:41:06.168: Calling org.xfce.Session.Manage
ibit"

1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ",";
5 }
6
7 {
8     # 10. Filter out rows where the Destination_Port contains any alphanumeric
9     # characters (letters or numbers).
10    if ($6 ~ /^[^0-9]/) {
11        next;
12    }
13    print $0;
14 }
15
```

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



```
File Edit View VM Tabs Help
Home X kali-linux-2024.2-vmware... X Windows 10 x64 X
(kali@kali)~/Desktop/CYS/Los_7
$ ./network_script10.awk text.txt
2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,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

(kali@kali)~/Desktop/CYS/Los_7
$ gedit text.txt

(gedit:17828): tepl-WARNING **: 05:17:21.262: Style scheme 'Kali-Dark' cannot
(gedit:17828): tepl-WARNING **: 05:17:21.263: Default style scheme 'Kali-Dark

(kali@kali)~/Desktop/CYS/Los_7
$ ./network_script10.awk text.txt

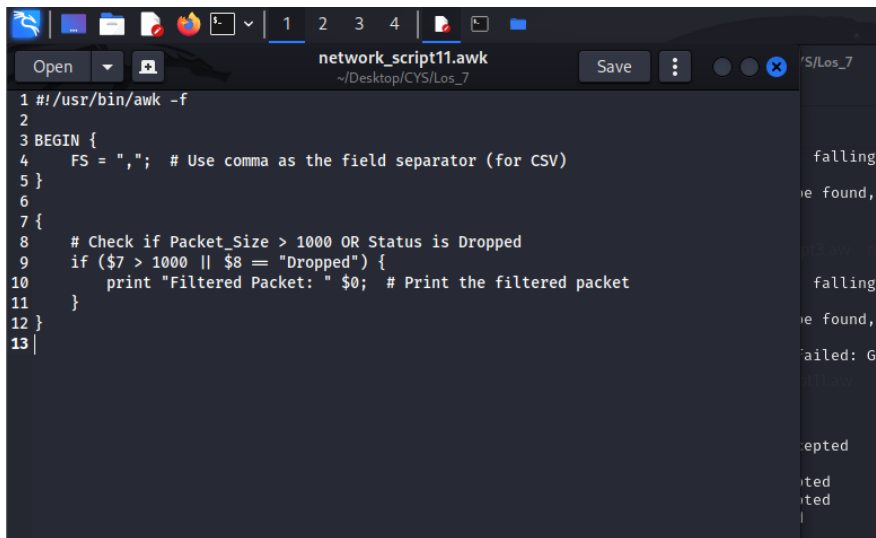
2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted
2024-09-30 10:15:18,192.168.1.10,198.51.100.23,TCP,443,1025,1500,Accepted
2024-09-30 10:15:20,198.51.100.23,192.168.1.10,TCP,443,1025,1400,Accepted

(kali@kali)~/Desktop/CYS/Los_7
$ gedit network_script10.awk

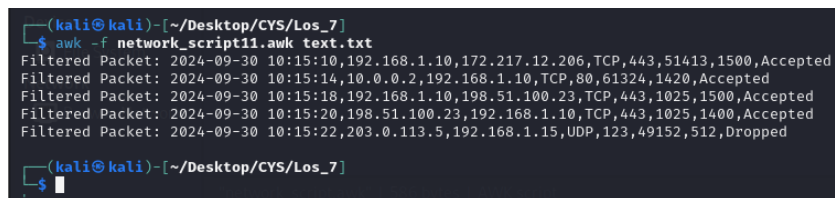
(gedit:18067): tepl-WARNING **: 05:17:49.027: Style scheme 'Kali-Dark' cannot
(gedit:18067): tepl-WARNING **: 05:17:49.028: Default style scheme 'Kali-Dark

1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ",";
5 }
6
7 {
8     # Print lines where the protocol is TCP AND the destination port is 443
9     protocol = $4
10    port_number = $5
11
12    if (protocol == "TCP" && port_number == 443) {
13        print $0 # Print the entire line if conditions are met
14    }
15 }
16
17 END {
18 }
19 }
```

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

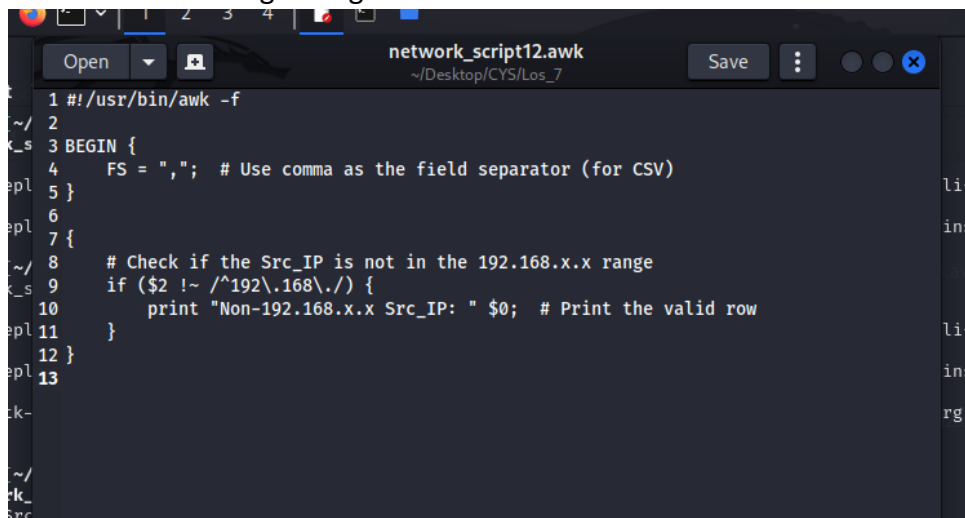


```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","; # Use comma as the field separator (for CSV)
5 }
6
7 {
8     # Check if Packet_Size > 1000 OR Status is Dropped
9     if ($7 > 1000 || $8 == "Dropped") {
10         print "Filtered Packet: " $0; # Print the filtered packet
11     }
12 }
13
```

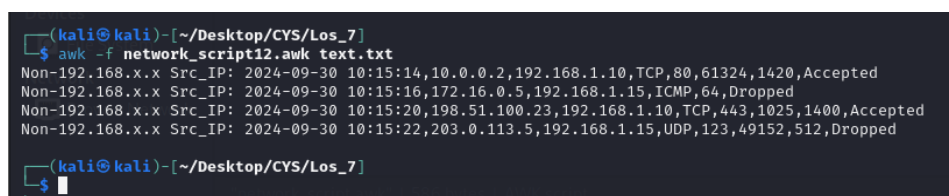


```
(kali@kali)-[~/Desktop/CYS/Los_7]
$ awk -f network_script11.awk text.txt
Filtered Packet: 2024-09-30 10:15:10,192.168.1.10,172.217.12.206,TCP,443,51413,1500,Accepted
Filtered Packet: 2024-09-30 10:15:14,10.0.0.2,192.168.1.10,TCP,80,61324,1420,Accepted
Filtered Packet: 2024-09-30 10:15:18,192.168.1.10,198.51.100.23,TCP,443,1025,1500,Accepted
Filtered Packet: 2024-09-30 10:15:20,198.51.100.23,192.168.1.10,TCP,443,1025,1400,Accepted
Filtered Packet: 2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped
(kali@kali)-[~/Desktop/CYS/Los_7]
```

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

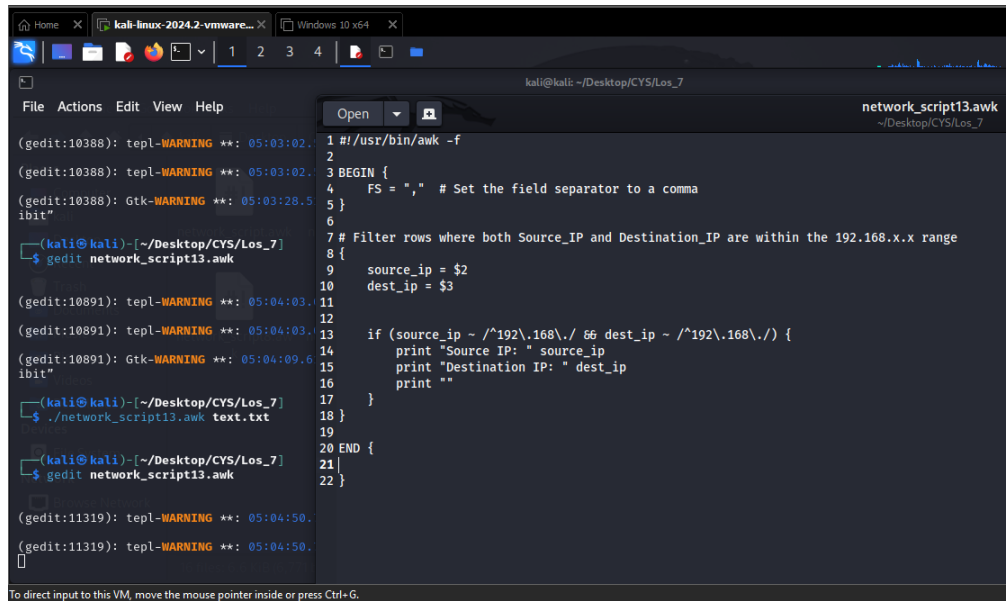


```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","; # Use comma as the field separator (for CSV)
5 }
6
7 {
8     # Check if the Src_IP is not in the 192.168.x.x range
9     if ($2 !~ /^192\.168\.*/) {
10         print "Non-192.168.x.x Src_IP: " $0; # Print the valid row
11     }
12 }
13
```



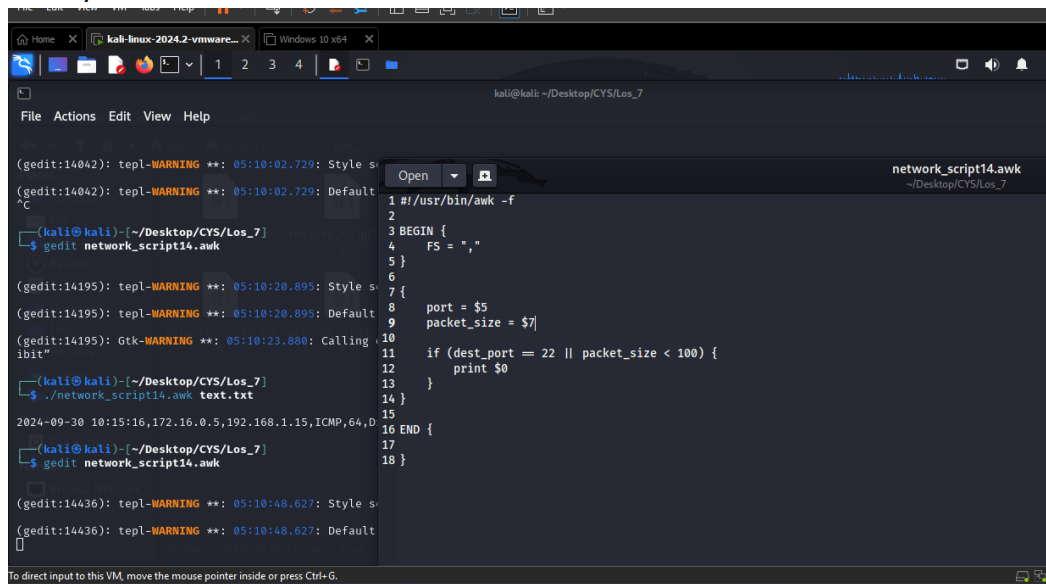
```
(kali@kali)-[~/Desktop/CYS/Los_7]
$ awk -f network_script12.awk text.txt
Non-192.168.x.x Src_IP: 2024-09-30 10:15:14,10.0.0.2,192.168.1.10,TCP,80,61324,1420,Accepted
Non-192.168.x.x Src_IP: 2024-09-30 10:15:16,172.16.0.5,192.168.1.15,ICMP,64,Dropped
Non-192.168.x.x Src_IP: 2024-09-30 10:15:20,198.51.100.23,192.168.1.10,TCP,443,1025,1400,Accepted
Non-192.168.x.x Src_IP: 2024-09-30 10:15:22,203.0.113.5,192.168.1.15,UDP,123,49152,512,Dropped
(kali@kali)-[~/Desktop/CYS/Los_7]
```

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



```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = "," # Set the field separator to a comma
5 }
6
7 # Filter rows where both Source_IP and Destination_IP are within the 192.168.x.x range
8 {
9     source_ip = $2
10    dest_ip = $3
11
12    if (source_ip ~ /^192\.168\.\/ 66 dest_ip ~ /^192\.168\.\/) {
13        print "Source IP: " source_ip
14        print "Destination IP: " dest_ip
15        print ""
16    }
17 }
18 }
19
20 END {
21
22 }
```

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



```
1 #!/usr/bin/awk -f
2
3 BEGIN {
4     FS = ","
5 }
6
7 {
8     port = $5
9     packet_size = $7
10
11     if (dest_port == 22 || packet_size < 100) {
12         print $0
13     }
14 }
15
16 END {
17
18 }
```