

Minor 2 Project

Project 2: Network Traffic Analysis & Incident Investigation Using PCAP (SOC Analyst Simulation)

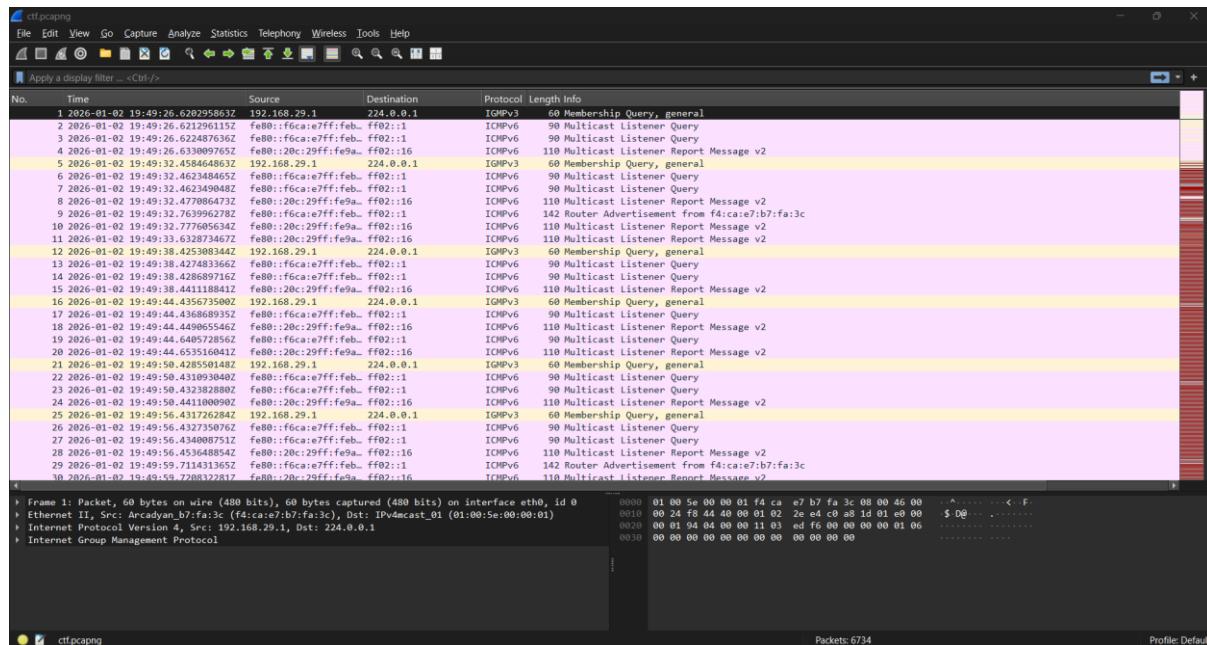
I am working as a Junior SOC Analyst in the Security Operations Centre of my organisation. During routine monitoring, one of the internal systems was suspected to be compromised. As part of the incident response process, the SOC team captured the network traffic (PCAP file) during the suspected incident window and provided it to me for detailed analysis.

My task was to analyse the PCAP file using Wireshark, following real-world SOC investigation methodologies. During the analysis, I focused on identifying the attacker's IP address, determining which internal system was the victim, and understanding the nature of the suspicious activities observed in the network traffic.

I examined the traffic patterns to identify reconnaissance activity, such as port scanning, and further analysed the HTTP traffic to determine how sensitive data was transferred over the network. During this investigation, I identified that a ZIP file containing sensitive data was downloaded using unencrypted HTTP communication.

Using Wireshark's file reconstruction features, I successfully extracted the ZIP file from the PCAP, analysed its contents, and retrieved the hidden flag. All findings were documented with appropriate screenshots and evidence, replicating the workflow of a real SOC analyst during a network-based incident investigation.

1. Open the PCAP file in Wireshark

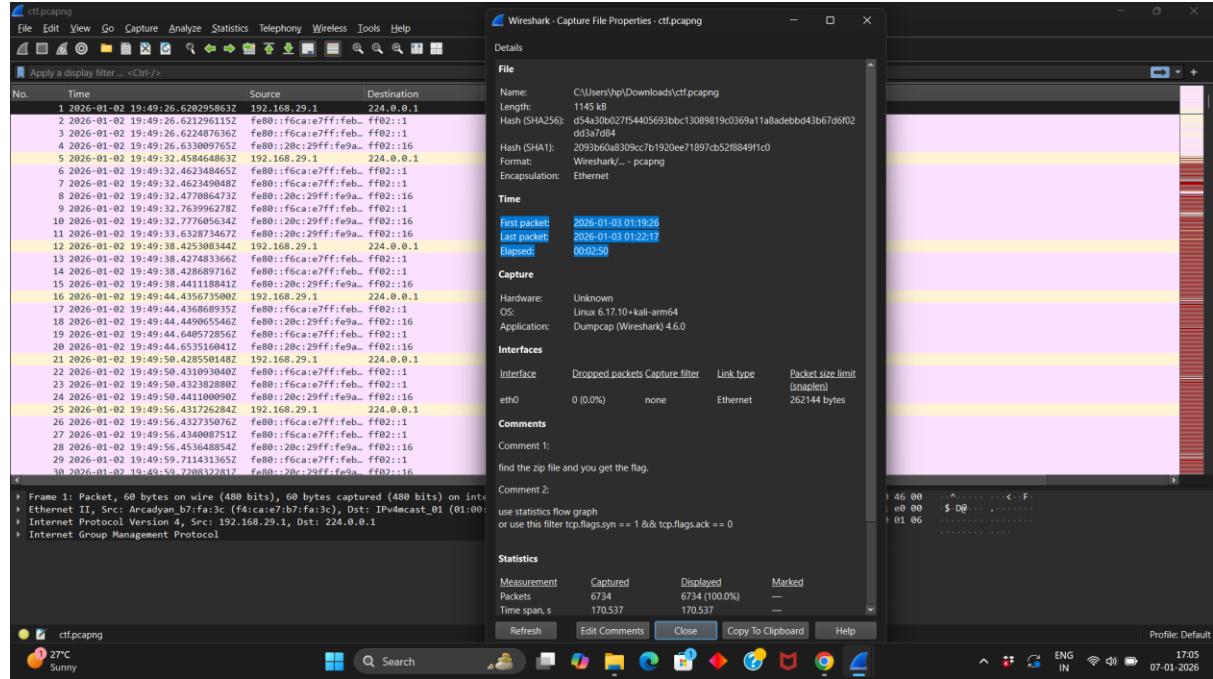


Statistics → Time

When the traffic started – 03 Jan 2026 (01:19:26)

When suspicious activity End – 03 Jan 2026 (01:22:17)

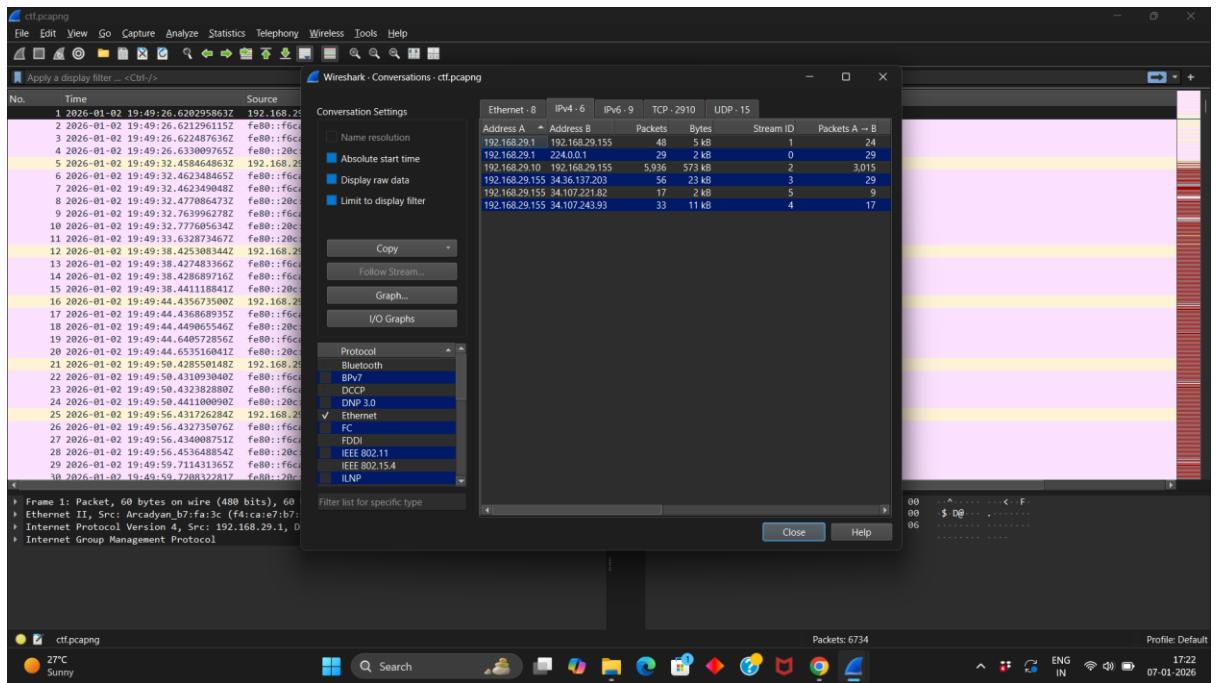
How long the activity Lasted - 02 min 50 sec



2. Analyse network traffic patterns

Conversation → IPv4

- ◆ section of Wireshark, the third row shows communication between Address A (192.168.29.10) and Address B (192.168.29.155). This conversation contains a significantly high number of packets compared to other entries, which indicates abnormal traffic and is considered suspicious.



3. Identify the attacker and victim

Conversation → TCP

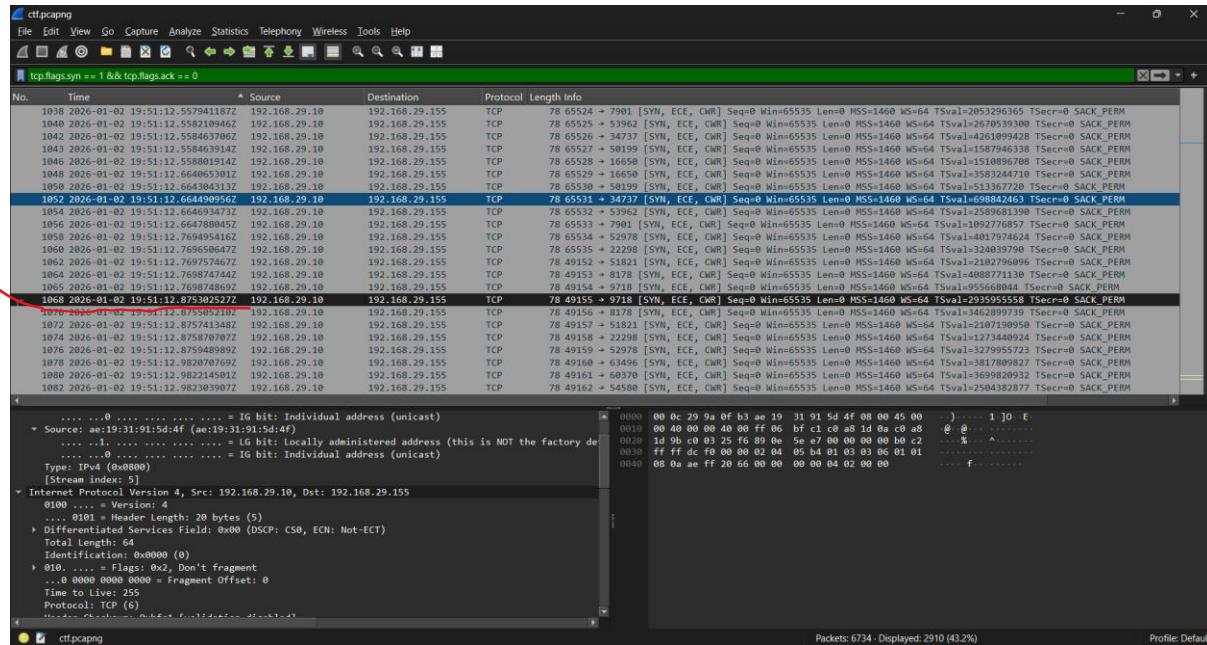
In this image IP (192.168.29.10) it's appears to be hitting a lot of ports towards IP (192.168.29.155).

- ◆ Victim → 192.168.29.155
- ◆ Attacker → 192.168.29.10

Conversation Settings													
	Ethernet - 8	IPv4 - 6	IPv6 - 9	TCP - 2910	UDP - 15								
<input type="checkbox"/> Name resolution													
<input checked="" type="checkbox"/> Absolute start time	192.168.29.10	49152	192.168.29.155	51821	2	132 bytes	459	1	78 bytes	1	54 bytes	106.1495161604	0.000013
<input checked="" type="checkbox"/> Display raw data	192.168.29.10	49153	192.168.29.155	8178	2	132 bytes	460	1	78 bytes	1	54 bytes	106.149579006	0.000010
<input checked="" type="checkbox"/> Limit to display filter	192.168.29.10	49155	192.168.29.155	9718	2	132 bytes	461	1	78 bytes	1	54 bytes	106.255006664	0.000066
	192.168.29.10	49156	192.168.29.155	8178	2	132 bytes	462	1	78 bytes	1	54 bytes	106.255209347	0.000011
	192.168.29.10	49157	192.168.29.155	51821	2	132 bytes	464	1	78 bytes	1	54 bytes	106.25545485	0.000020
	192.168.29.10	49158	192.168.29.155	22298	2	132 bytes	465	1	78 bytes	1	54 bytes	106.255574844	0.000008
	192.168.29.10	49159	192.168.29.155	52978	2	132 bytes	466	1	78 bytes	1	54 bytes	106.255653126	0.000006
	192.168.29.10	49160	192.168.29.155	63496	2	132 bytes	467	1	78 bytes	1	54 bytes	106.39177406	0.000041
	192.168.29.10	49161	192.168.29.155	60370	2	132 bytes	468	1	78 bytes	1	54 bytes	106.451864000	0.000011
	192.168.29.10	49162	192.168.29.155	54580	2	132 bytes	469	1	78 bytes	1	54 bytes	106.46320044	0.000097
	192.168.29.10	49163	192.168.29.155	54580	2	132 bytes	470	1	78 bytes	1	54 bytes	106.362077202	0.000004
	192.168.29.10	49164	192.168.29.155	26234	2	132 bytes	471	1	78 bytes	1	54 bytes	106.362173857	0.000008
	192.168.29.10	49165	192.168.29.155	26234	2	132 bytes	472	1	78 bytes	1	54 bytes	106.464695039	0.000030
	192.168.29.10	49166	192.168.29.155	24868	2	132 bytes	473	1	78 bytes	1	54 bytes	106.464879100	0.000011
	192.168.29.10	49167	192.168.29.155	60370	2	132 bytes	474	1	78 bytes	1	54 bytes	106.464981712	0.000006
	192.168.29.10	49168	192.168.29.155	63496	2	132 bytes	475	1	78 bytes	1	54 bytes	106.465085991	0.000007
	192.168.29.10	49169	192.168.29.155	29160	2	132 bytes	476	1	78 bytes	1	54 bytes	106.465178605	0.000006
	192.168.29.10	49170	192.168.29.155	32298	4	236 bytes	477	3	204 bytes	1	74 bytes	106.465200446	0.000158
	192.168.29.10	49171	192.168.29.155	64650	2	132 bytes	478	1	78 bytes	1	54 bytes	106.465211331	0.000006
	192.168.29.10	49172	192.168.29.155	64650	2	132 bytes	479	1	78 bytes	1	54 bytes	106.570864397	0.000004
	192.168.29.10	49173	192.168.29.155	42681	2	132 bytes	480	1	78 bytes	1	54 bytes	106.571140238	0.000003
	192.168.29.10	49174	192.168.29.155	63788	2	132 bytes	481	1	78 bytes	1	54 bytes	106.571359253	0.000003
	192.168.29.10	49175	192.168.29.155	62924	2	132 bytes	482	1	78 bytes	1	54 bytes	106.571756912	0.000004
	192.168.29.10	49176	192.168.29.155	62924	2	132 bytes	483	1	78 bytes	1	54 bytes	106.676326050	0.000048
	192.168.29.10	49177	192.168.29.155	63788	2	132 bytes	484	1	78 bytes	1	54 bytes	106.676614931	0.000020
	192.168.29.10	49178	192.168.29.155	47387	2	132 bytes	485	1	78 bytes	1	54 bytes	106.676822989	0.000006
	192.168.29.10	49179	192.168.29.155	42681	2	132 bytes	486	1	78 bytes	1	54 bytes	106.677004414	0.000005
	192.168.29.10	49180	192.168.29.155	64650	2	132 bytes	487	1	78 bytes	1	54 bytes	106.677011331	0.000006
	192.168.29.10	49181	192.168.29.155	38781	2	132 bytes	488	1	78 bytes	1	54 bytes	106.715154000	0.000040
	192.168.29.10	49182	192.168.29.155	29649	2	132 bytes	489	1	78 bytes	1	54 bytes	106.732234892	0.000012
	192.168.29.10	49183	192.168.29.155	37107	2	132 bytes	490	1	78 bytes	1	54 bytes	106.732574440	0.000005
	192.168.29.10	49184	192.168.29.155	49183	2	132 bytes	491	1	78 bytes	1	54 bytes	106.732766166	0.000004
	192.168.29.10	49185	192.168.29.155	51530	2	132 bytes	492	1	78 bytes	1	54 bytes	106.729561100	0.000004
	192.168.29.10	49186	192.168.29.155	51530	2	132 bytes	493	1	78 bytes	1	54 bytes	106.886815327	0.000030
	192.168.29.10	49187	192.168.29.155	49183	2	132 bytes	494	1	78 bytes	1	54 bytes	106.886931813	0.000004
	192.168.29.10	49188	192.168.29.155	37107	2	132 bytes	495	1	78 bytes	1	54 bytes	106.891787076	0.000004
	192.168.29.10	49189	192.168.29.155	29649	2	132 bytes	496	1	78 bytes	1	54 bytes	106.891818386	0.000004
	192.168.29.10	49190	192.168.29.155	38781	2	132 bytes	497	1	78 bytes	1	54 bytes	106.891785886	0.000004
	192.168.29.10	49191	192.168.29.155	60860	2	132 bytes	498	1	78 bytes	1	54 bytes	106.899980264	0.000023
	192.168.29.10	49192	192.168.29.155	57516	2	132 bytes	499	1	78 bytes	1	54 bytes	106.900064212	0.000005
	192.168.29.10	49193	192.168.29.155	29160	2	132 bytes	500	1	78 bytes	1	54 bytes	106.990142077	0.000004

4.Determine when the attack started

5.Identify reconnaissance activity (port scanning)



Command: `tcp.flags.syn == 1 && tcp.flags.ack == 0`

TCP Three-Way Handshake

TCP establishes a connection using a three-step handshake process:

- **SYN** → The client sends a request to initiate a connection.
- **SYN-ACK** → The server responds to the client's request.
- **ACK** → The client acknowledges the server's response and the connection is established.

Explanation of the Command:

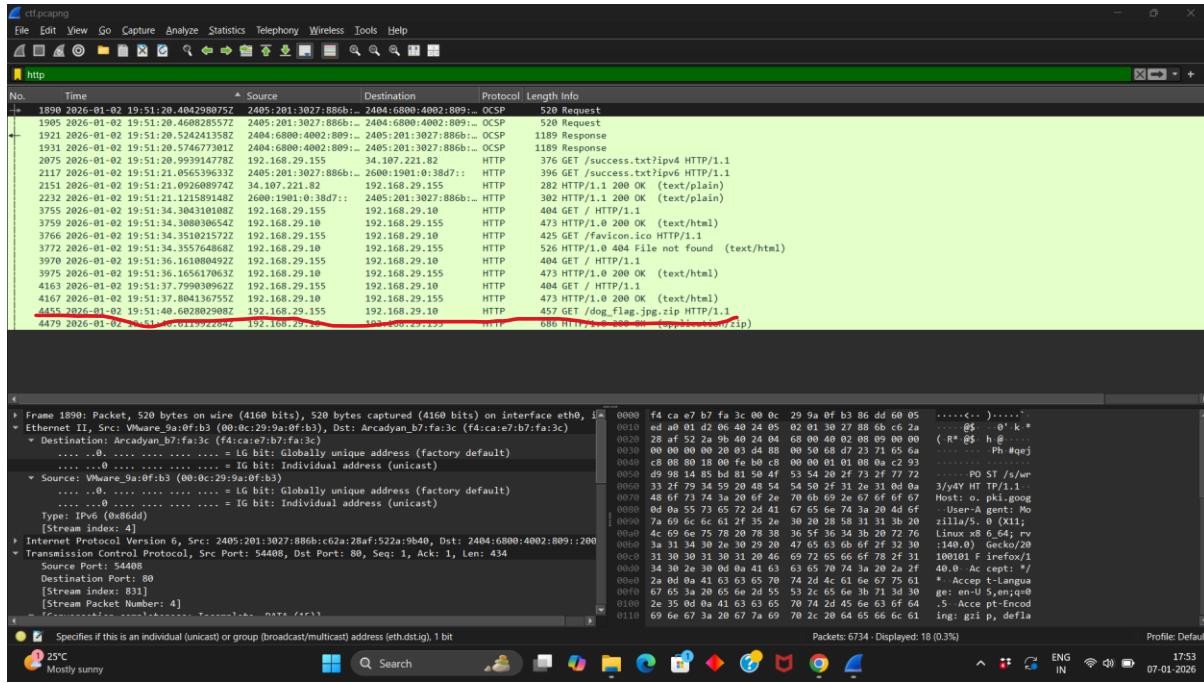
1. `tcp.flags.syn == 1`

This condition selects packets where the **SYN (Synchronize)** flag is set. It indicates an attempt to initiate a TCP connection and is typically used when a client tries to connect to a specific port on a server.

2. `tcp.flags.ack == 0`

This condition selects packets where the **ACK (Acknowledgment)** flag is not set. This means the packet is not acknowledging any previous communication.

6.Find the HTTP file download



After finding the HTTP

dog_flag.jpg.zip

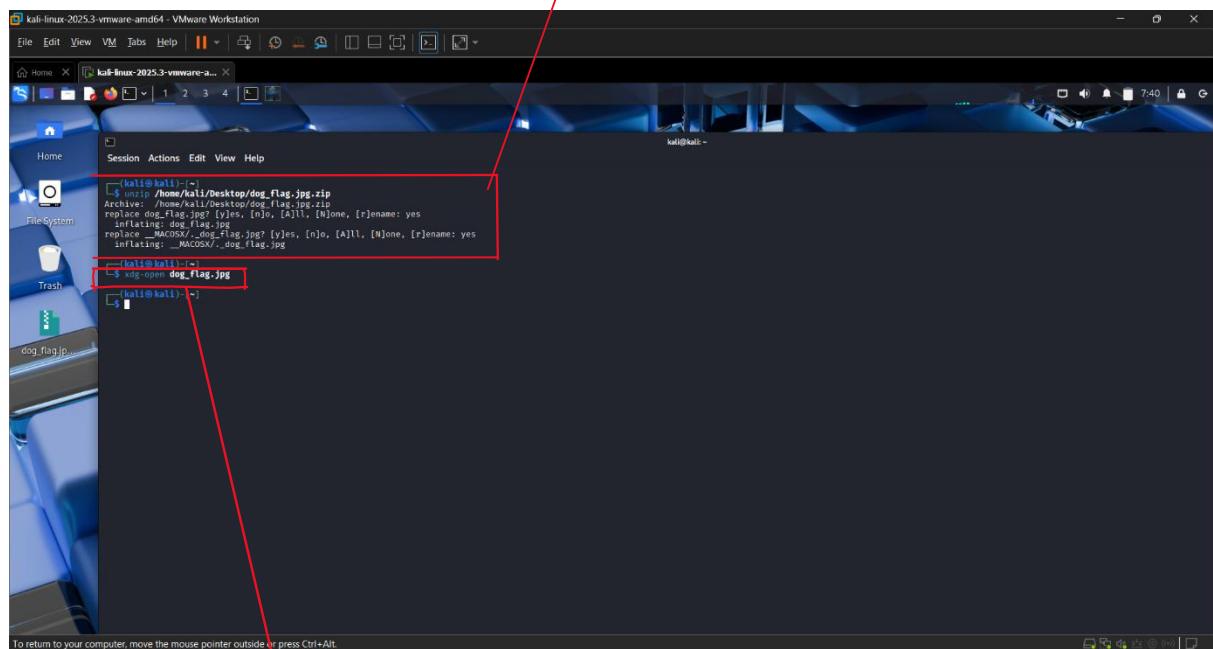
7. Extract the ZIP file from the PCAP

1. Go to file
2. Extract object
3. Choose the file (dog_flag.jpg.zip)
4. Save

8. Unzip the file

1. Go to Linux
2. Drag or copy the zip file in Linux OS
3. Open terminal
4. Run command to unzip

Command to unzip



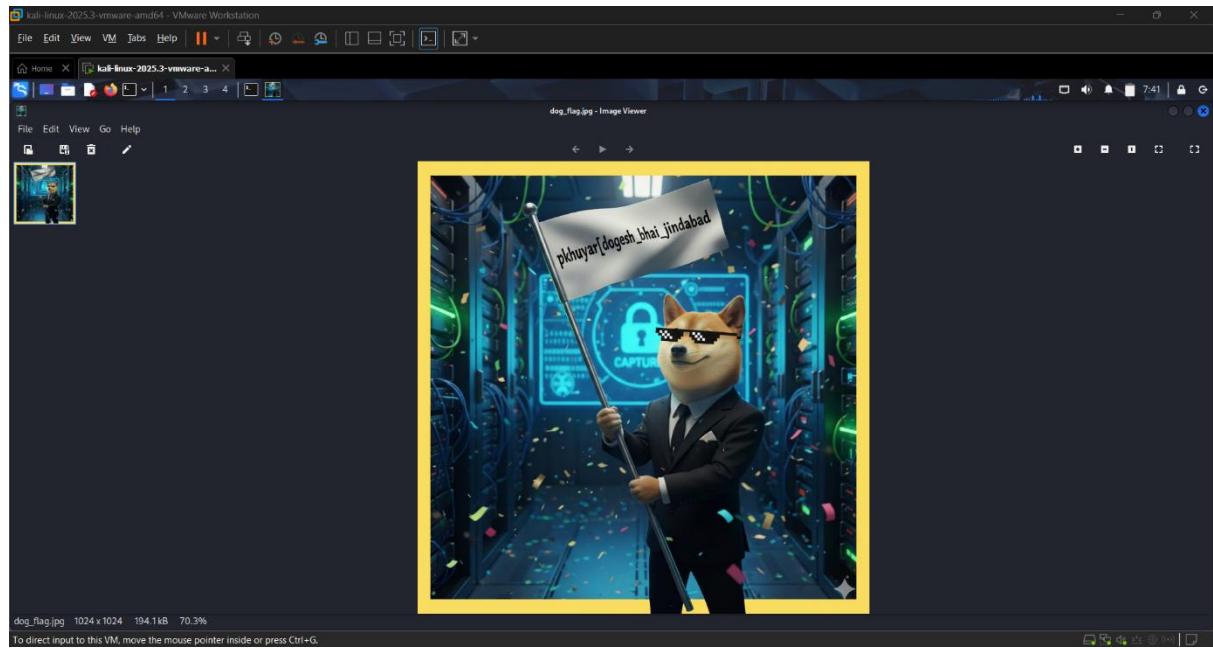
The screenshot shows a terminal window in a Kali Linux environment. The user has run the command `xdg-open dog_flag.jpg.zip`. The terminal output shows the process of extracting the file:

```
[kali㉿kali]:~$ xdg-open /home/kali/Desktop/dog_flag.jpg.zip
Archive: /home/kali/Desktop/dog_flag.jpg.zip
replace dog_flag.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
inflating: dog_flag.jpg
replace /MACOSX/_dog_flag.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: yes
inflating: _MACOSX/_dog_flag.jpg
```

After extraction, the user runs the command `xdg-open dog_flag.jpg`, which opens the image file.

The command **xdg-open** dog_flag.jpg is used in Linux to open a file using the system's default application. It will open the file in default image viewer.

Retrieve the flag



Questions to Answer

Answer the following questions one by one in your report:

1. What is the attacker IP address?

Ans → 192.168.29.10

2. What is the first packet timestamp related to the attack?

Ans → 2026-01-02 19:51:12.875302527Z

3. What evidence suggests that port scanning (reconnaissance) was performed?

Ans → TCP 3-way handshake

4. What is the name of the downloaded ZIP file?

Ans → dog_flag.jpg.zip

5. What is the flag obtained after unzipping the file?

FLAG{pkhuyar_[doges_bhai_jindabad]}