

Assignment: Lab 2 – Traffic Tracing and Sniffing

By Justin White

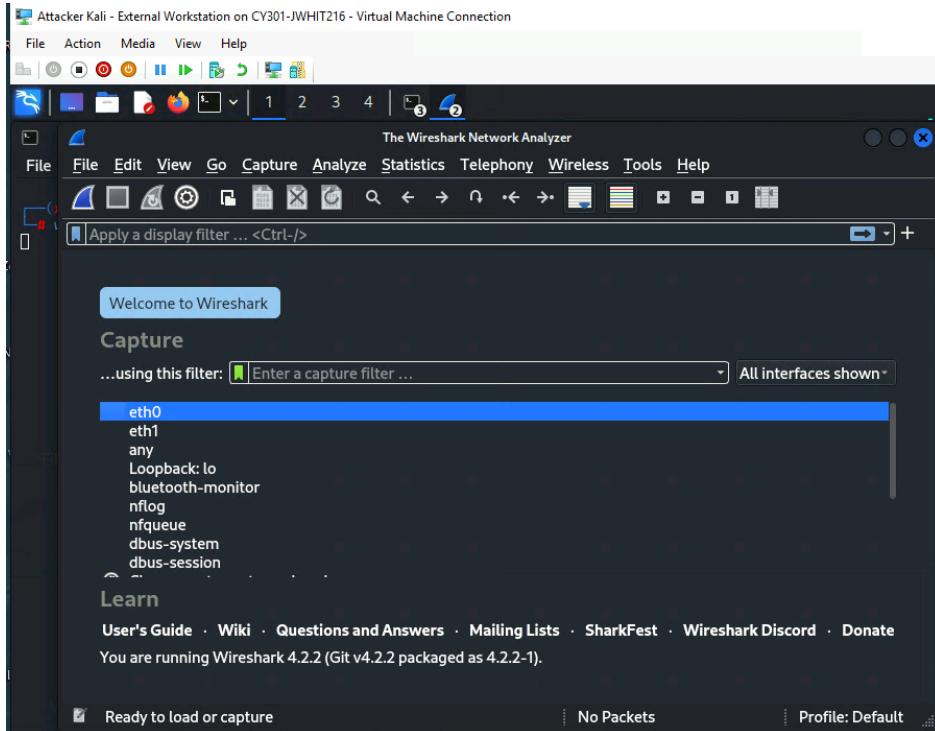
CYSE 301

Professor Vatsa

June 16th, 2025

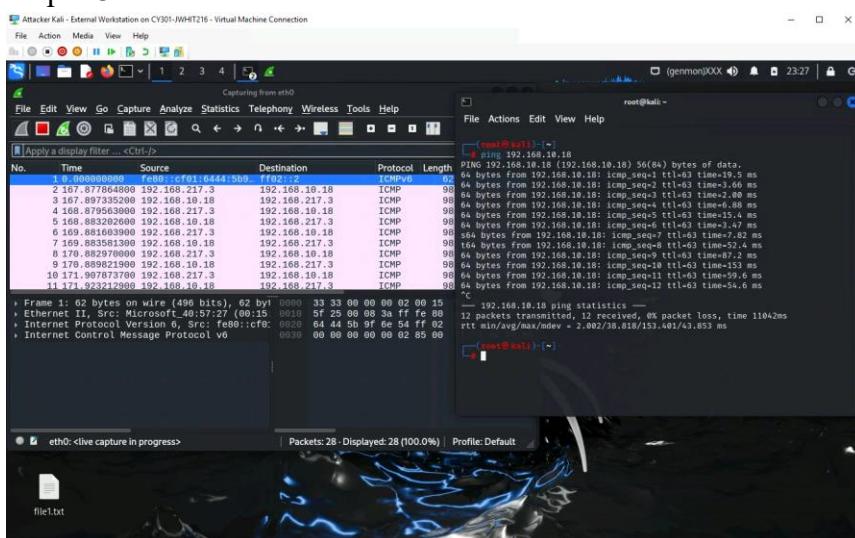
Task A

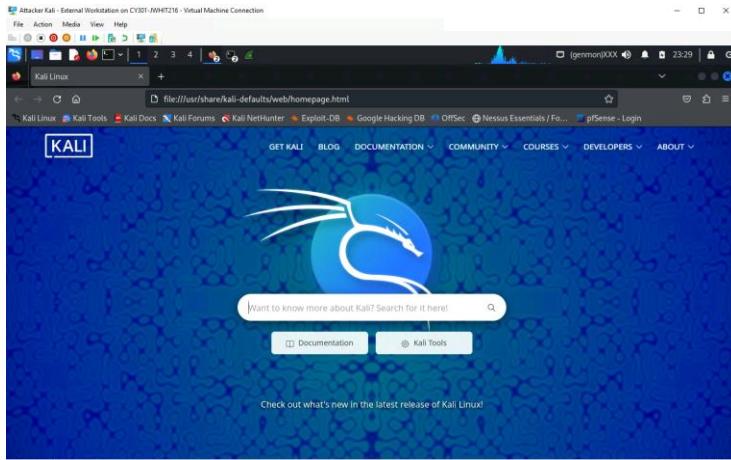
Step 1:



I launched wireshark on external kali and started capturing packets on the eth0 interface to monitor network traffic.

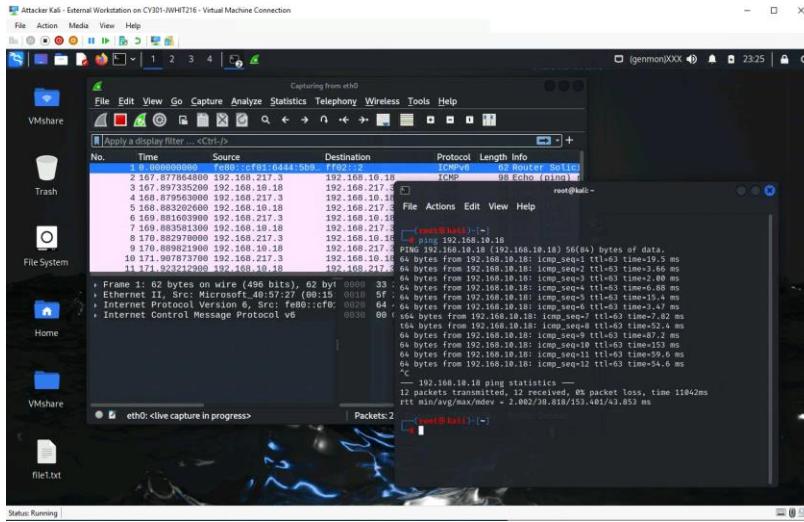
Step 2-3:





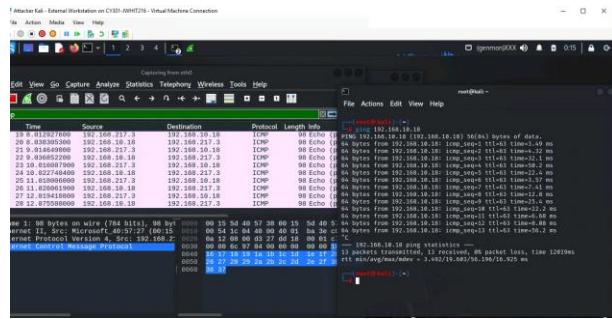
For this step I used the ping command from external kali to ubuntu to generate ICMP traffic that wireshark could capture for analysis as well as opened a web browser to trigger DNS and HTTP requests, simulating normal user behavior and increasing traffic for capture.

Step 4:

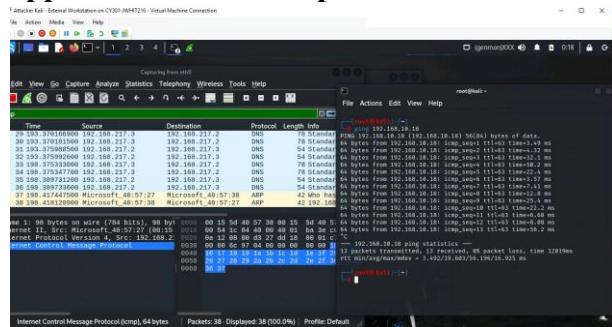


For the last step for task A I stopped wireshark from capturing after interactions were complete to analyze any and all collected data for further detail.

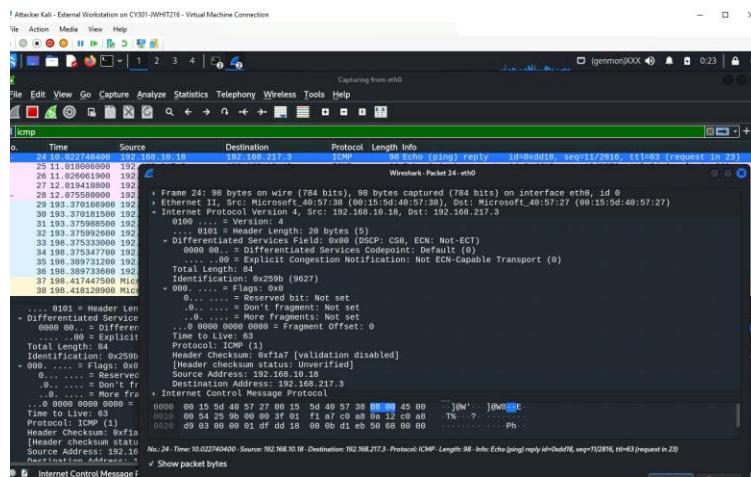
Q1 Answer: The number of packets sent and received were 13 packets however when pinged in wireshark it was 28 packets.



Q2: Answer: When pining Ubuntu while having wireshark up and having the ICMP display applied the amount of packets were 38 and the amount displayed were 38.



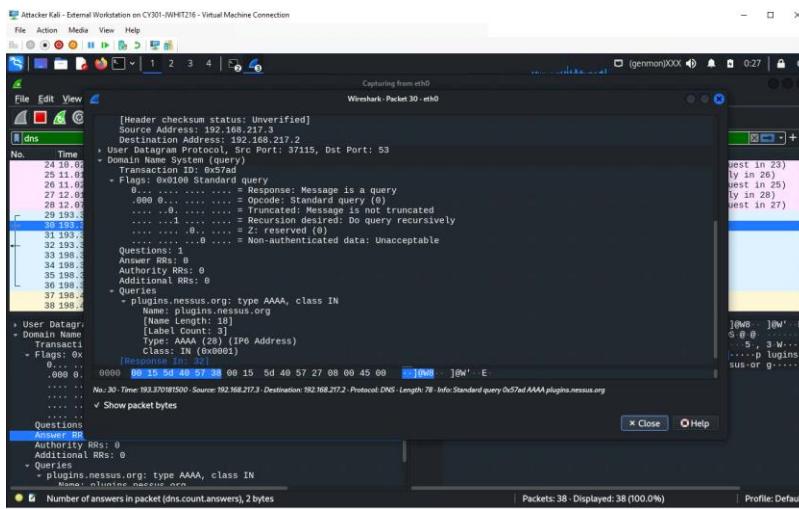
Q3 Answer: The ICMP echo reply selected with source/destination IP, sequence number, and data size.



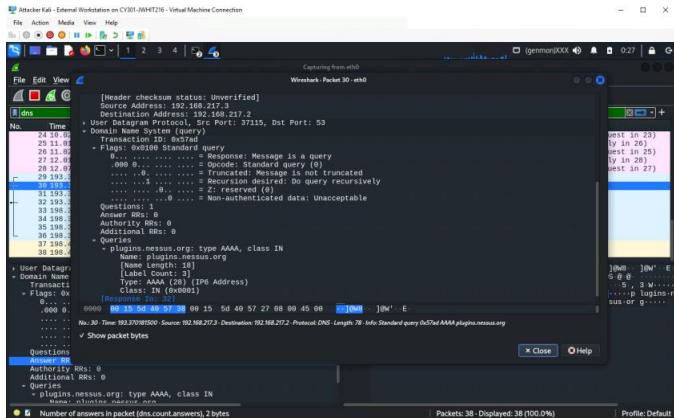
Q4 Answer: Displayed DNS packets are 8 in total

Attacker Kali - External Workstation on CY301-JWHIT216 - Virtual Machine Connection						
File Action Media View Help						
Capturing from eth0						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
No.	Time	Source	Destination	Protocol	Length	Info
24	10.0227404000	192.168.10.18	192.168.217.3	ICMP	98	Echo (ping) reply id=0xdd18, seq=11/2816, ttl=63 (request in 23)
25	11.0180600000	192.168.217.3	192.168.10.18	ICMP	98	Echo (ping) request id=0xdd18, seq=12/3072, ttl=64 (reply in 26)
26	11.0260619000	192.168.10.18	192.168.217.3	ICMP	98	Echo (ping) reply id=0xdd18, seq=12/3072, ttl=63 (request in 25)
27	12.0194108000	192.168.217.3	192.168.10.18	ICMP	98	Echo (ping) request id=0xdd18, seq=13/3328, ttl=64 (reply in 28)
28	12.0755800000	192.168.10.18	192.168.217.3	ICMP	98	Echo (ping) reply id=0xdd18, seq=13/3328, ttl=63 (request in 27)
29	13.0701669000	192.168.217.3	192.168.217.2	DNS	78	Standard query 0x495f A plugins.nessus.org
30	19.3701815000	192.168.217.3	192.168.217.2	DNS	78	Standard query 0x57ad AAAA plugins.nessus.org
31	19.3735988500	192.168.217.2	192.168.217.3	DNS	54	Standard query response 0x495f Refused
32	19.3735992600	192.168.217.2	192.168.217.3	DNS	54	Standard query response 0x57ad Refused
33	19.3735333000	192.168.217.3	192.168.217.2	DNS	78	Standard query 0x495f A plugins.nessus.org
34	19.3735477000	192.168.217.3	192.168.217.2	DNS	78	Standard query 0x57ad AAAA plugins.nessus.org
35	19.3897312000	192.168.217.2	192.168.217.3	DNS	54	Standard query response 0x495f Refused
36	19.3897333600	192.168.217.2	192.168.217.3	DNS	54	Standard query response 0x57ad Refused
37	19.8417447500	Microsoft_40:57:27	Microsoft_40:57:38	ARP	42	Who has 192.168.217.2? Tell 192.168.217.3
38	19.8418120900	Microsoft_40:57:27	Microsoft_40:57:38	ARP	42	192.168.217.2 is at 00:15:5d:40:57:38

Q5 Answer: The DNS query packet showing the domain name and source/destination IP shown within the domain name section

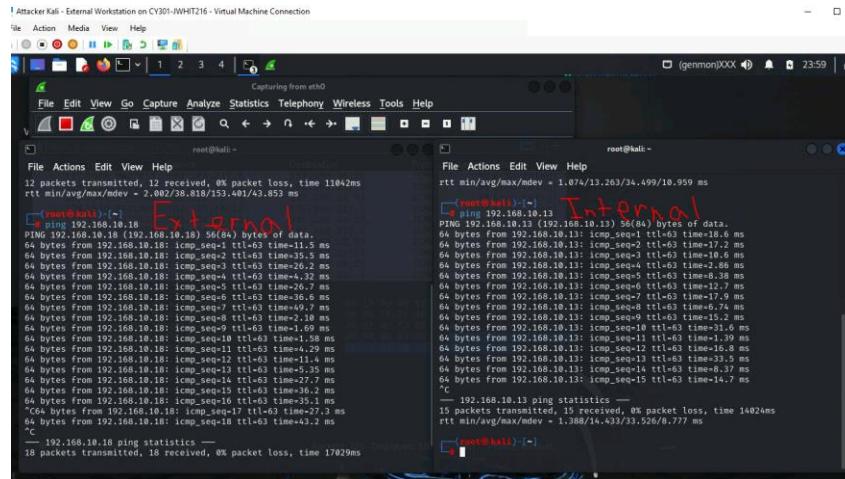
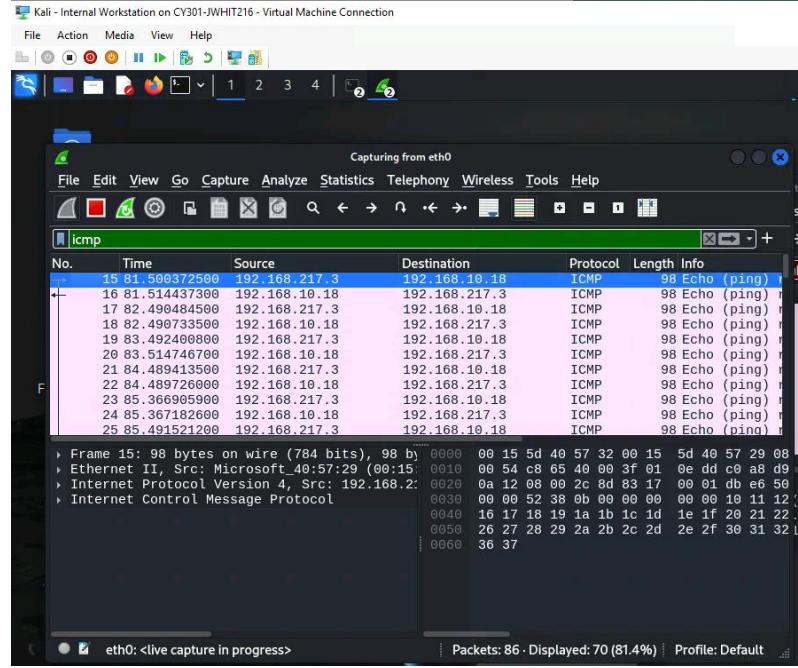


Q6 Answer: The source from the query is shown to have the same IP destination but received a denial is a reply.



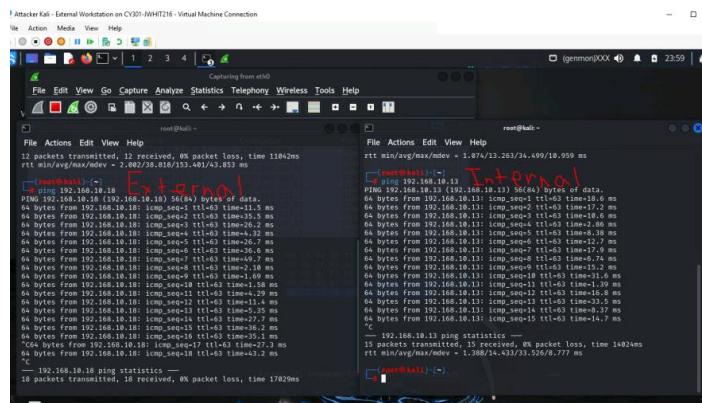
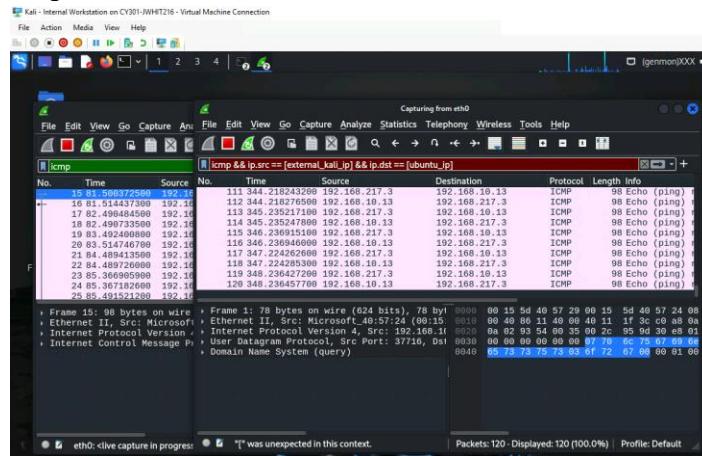
Task B

Step 1a:



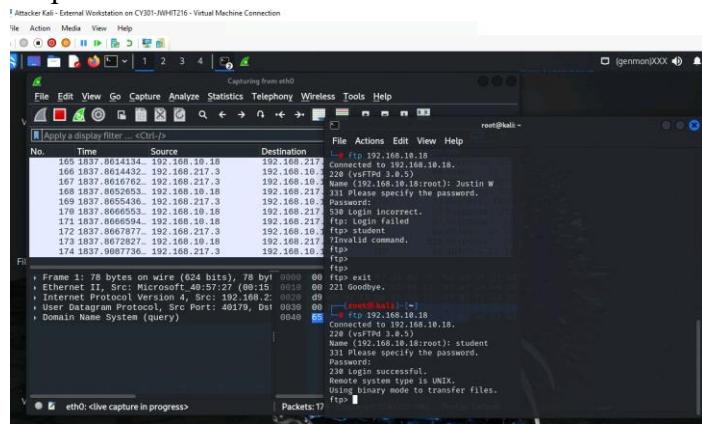
I started on wireshark on internal kali with a filter to capture general ICMP traffic between the two hosts being external kali and ubuntu within the LAN.

Step 1b:



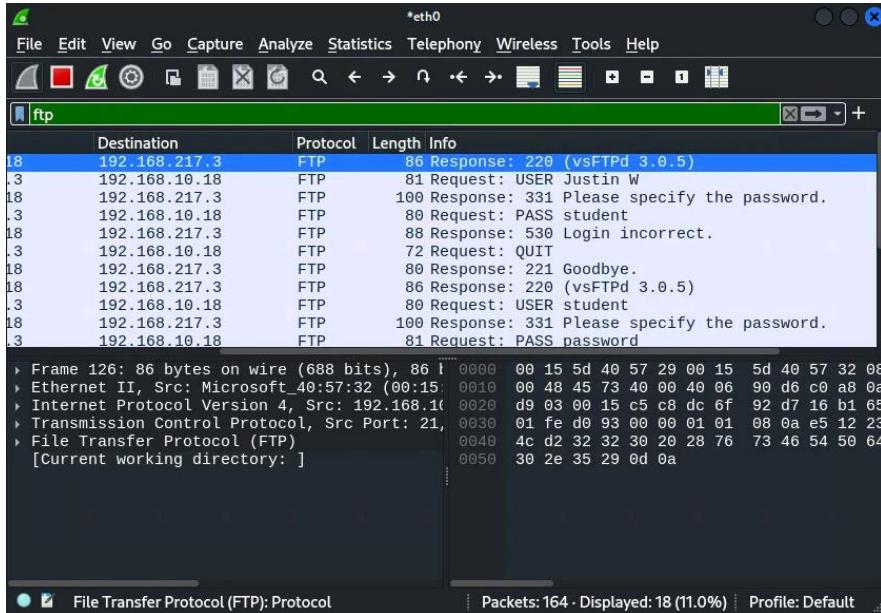
I applied for an ICMP filter to isolate the packets specifically from external kali to ubuntu, to simulate what targeted traffic monitoring looks like.

Step 2a:



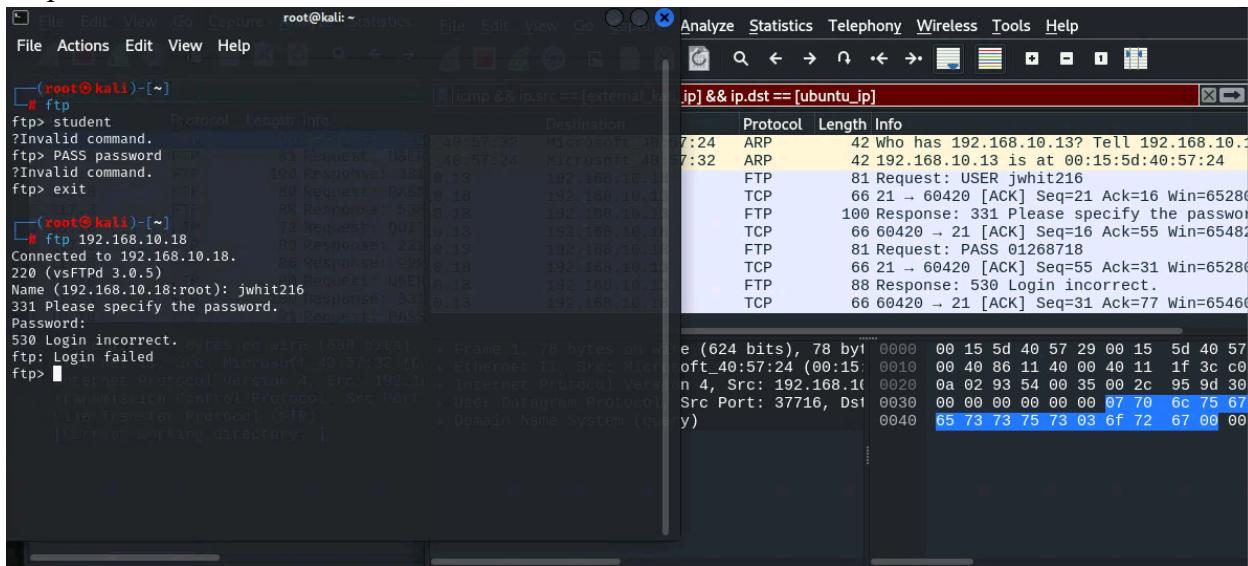
I initiated FTP connection from external kali to ubuntu using the provided credentials which generated the FTP login traffic for the simulated interception of the traffic.

Step 2b:



In this step I intercepted and inspected FTP control traffic on internal kali to reveal login credentials in the plaintext using the FTP display filter.

Step 2c:



For the final step I repeated the FTP login using personal credentials to demonstrate how sensitive information is exposed over unencrypted protocols.