Wireshark DNS Packet Capture & Analysis - Summary

1. What We Did

We used Wireshark on Kali Linux (running inside UTM on Mac) to capture and analyze DNS, HTTP, TLS, and ICMP packets.
Steps performed:
1. Opened Wireshark and selected the active network interface.
2. Started packet capture.
3. Generated safe traffic: DNS queries, HTTP requests, HTTPS/TLS traffic, and ICMP pings.
4. Applied display filters to inspect specific protocols.
5. Exported only safe packets using 'Export Specified Packets' -> Displayed.
6. Saved the filtered capture as task5_capture_safe.pcap.
7. Transferred the file from VM to Mac using UTM shared folders.
8. Prepared README.md, report.md, and uploaded the capture with screenshots to GitHub.

2. Interview Prep Questions & Answers

Q1: What is DNS and why is it important?
A1: DNS translates human-readable domain names into IP addresses.

Q2: How do you filter DNS packets in Wireshark?
A2: Use the display filter: dns

Q3: What is the difference between a display filter and a capture filter?
A3: Capture filters are applied before capture to limit recorded packets. Display filters are applied after capture to show only matching packets.

Q4: What are common DNS record types?
A4:
- A: IPv4 address
- AAAA: IPv6 address
- CNAME: Canonical name (alias)
- MX: Mail exchange

Q5: How do you ensure no sensitive data is uploaded?
A5: Review packets and export only safe packets using filters.

Q6: How can .pcap files be opened and analyzed?
A6: Open in Wireshark or use CLI tools like tcpdump.

Q7: What is the risk of uploading raw .pcap captures?
A7: They may contain sensitive data like IPs, credentials, or session info.