1. What types of traffic (HTTP, DNS, FTP, etc.) are present?
   The traffic present in the capture includes:

- **DNS (Domain Name System)**: Numerous DNS queries and responses (e.g., queries for www.google.com, fonts.gstatic.com, etc.).
- **HTTP (Hypertext Transfer Protocol)**: HTTP GET requests and responses (e.g., requests for /decoy2.txt, /encoded.txt, /decoy1.txt).
- **TCP (Transmission Control Protocol)**: TCP handshakes and data transfers (e.g., SYN, SYN-ACK, ACK packets).
- **MDNS (Multicast DNS)**: A single multicast DNS query (e.g., _ftp._tcp.local).
- **HTTPS (HTTP Secure)**: DNS queries for HTTPS records (e.g., HTTPS www.google.com).

2. How many DNS queries were made in total?
   There are **764 DNS queries** in total.
3. What types of DNS queries were made?

The DNS queries include:

- **A (IPv4 address)**: Standard queries for IPv4 addresses (e.g., A www.google.com).
- **AAAA (IPv6 address)**: Queries for IPv6 addresses (e.g., AAAA www.google.com).
- **HTTPS (HTTPS service binding)**: Queries for HTTPS records (e.g., HTTPS www.google.com).
- **PTR (Pointer)**: Reverse DNS lookups (e.g., PTR 14.205.250.142.in-addr.arpa).
- **CNAME (Canonical Name)**: Queries for aliases (e.g., responses with CNAME records).
- **SOA (Start of Authority)**: Authority-related queries (e.g., responses with SOA records).

4. What is a Loopback Interface?

The **loopback interface** is a virtual network interface used for internal communication within a host. It is typically assigned the IP

address 127.0.0.1 (IPv4) or ::1 (IPv6). In this capture, most traffic is between 127.0.0.1 and 127.0.0.53 (local DNS resolver), indicating internal communication.

5. How many .txt files were requested? List their names.

There are **3 .txt files** requested:

1. /decoy2.txt (Packet 180)
2. /encoded.txt (Packet 207)
3. /decoy1.txt (Packet 752)

6. One .txt file contains base64-encoded content. Identify and decode it.

The file /encoded.txt (Packet 207) contains base64-encoded content. The decoded content is:

- **Base64 String**: The actual base64 string is not directly visible in the provided data, but the response to this request (Packet 211) contains the HTTP response. The decoded content would typically be extracted from the payload of this response.
- **Decoded Content**: Without the exact base64 string, the decoded content cannot be provided here. However, you would decode it using a base64 decoder to reveal the hidden message.

7. Was any attempt made to distract the analyst using decoy files? Explain.

Yes, **decoy files** (/decoy1.txt and /decoy2.txt) were likely used to distract the analyst. These files appear legitimate but do not contain meaningful information, while the /encoded.txt file contains the actual hidden or sensitive data.

8. Are there any known ports being used for uncommon services?

Port 8000: Used for the HTTP server in this capture(uncommon for standard HTTP, which typically uses port 80 or 443 for HTTPS). This could indicate a custom or non-standard service.

Port 53: Standard for DNS(common, not uncommon)

9. How many HTTP GET requests are visible in the capture?

Number of HTTP GET Requests. There are 3 HTTP GET requests:

(i) GET /decoy2.txt HTTP/1.1 (Packet 180)
(ii) GET /encoded.txt HTTP/1.1 (Packet 207)
(iii) GET /decoy1.txt HTTP/1.1 (Packet 752)