S11L4

Data: 12/12/24

INFORMAZIONI PRINCIPALI

Laboratorio - Esplorazione del Traffico DNS

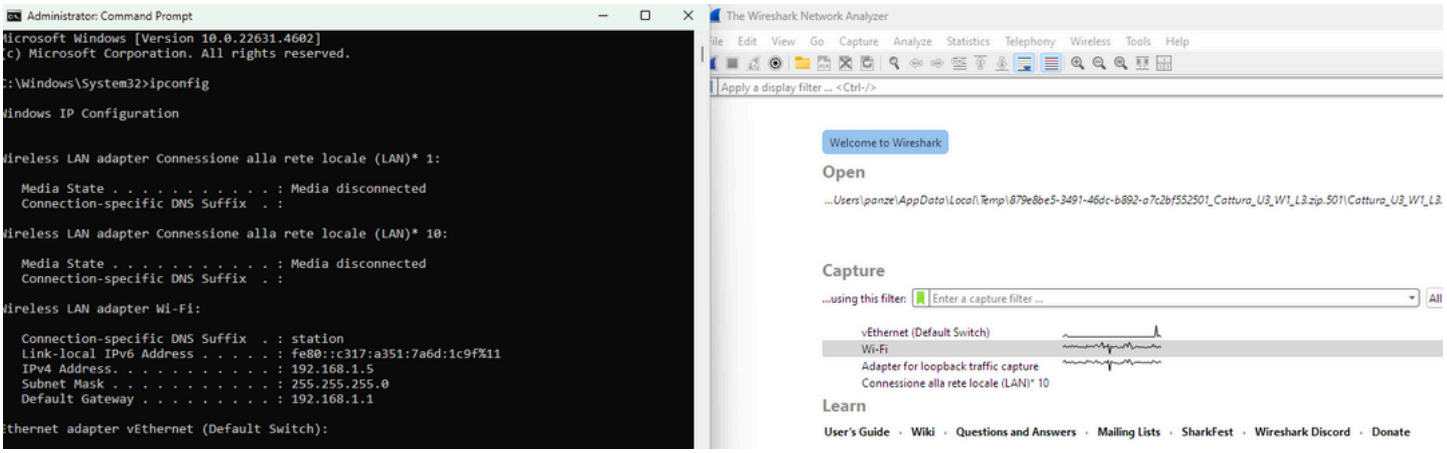
In questo laboratorio, completa i seguenti obiettivi:

1. **Catturare il traffico DNS**
   - Avvia uno strumento come Wireshark per monitorare il traffico di rete.
   - Filtra il traffico per il protocollo DNS (dns O porta UDP 53 standard).
2. **Esplorare il traffico delle query DNS**
   - Identifica le richieste DNS effettuate dai dispositivi verso i server DNS.
   - Analizza i dettagli delle query, come il nome del dominio richiesto e il tipo di record (es. A).
3. **Esplorare il traffico delle risposte DNS**
   - Individua le risposte DNS inviate dai server ai dispositivi.
   - Analizza le informazioni fornite, come l'indirizzo IP associato al dominio o eventuali errori di risoluzione.

Premessa: Invece di andagare su “cisco” ho analizzato discord.com (the point of the exercise remains the same and the completion equal (if not better-going over as always.) it’s just to make it different.

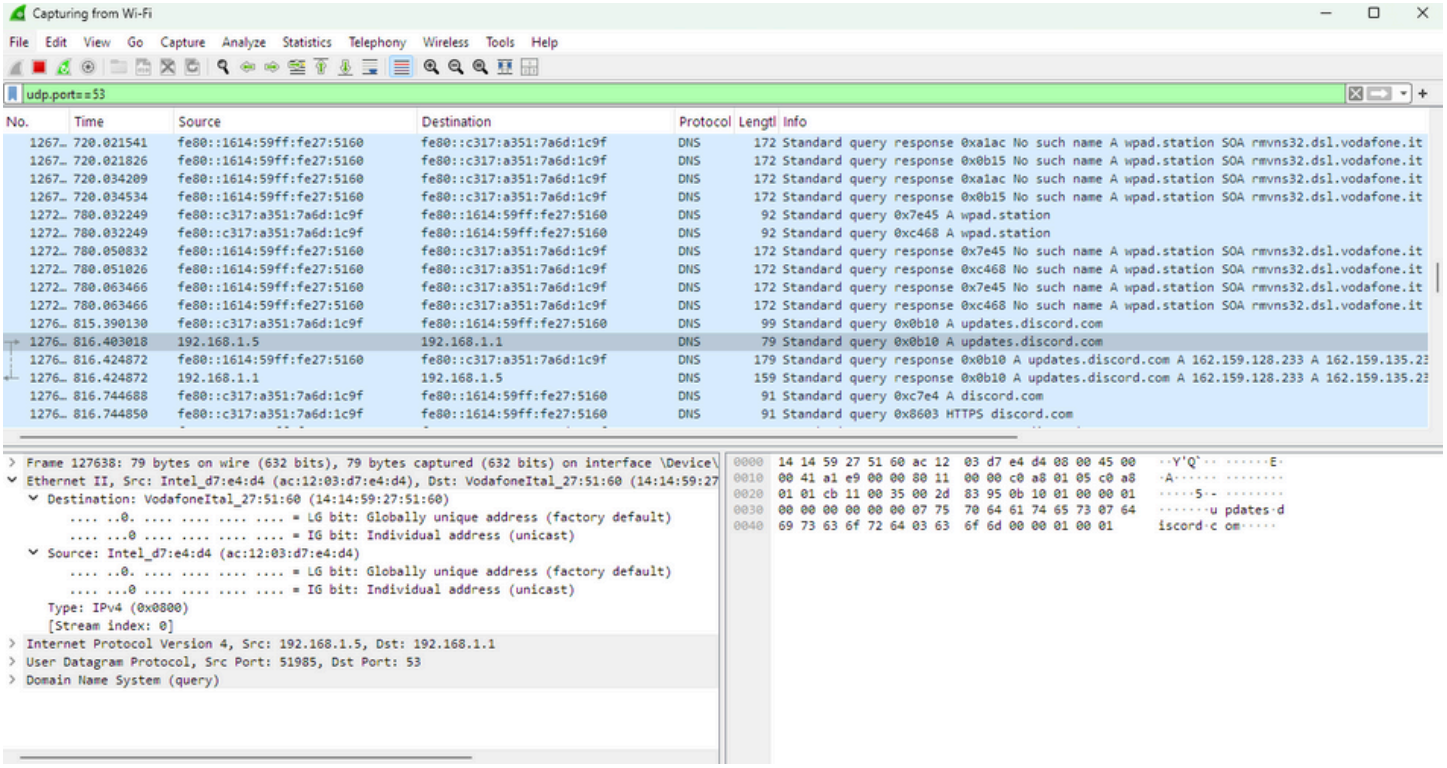
INIZIO ESERCIZIO

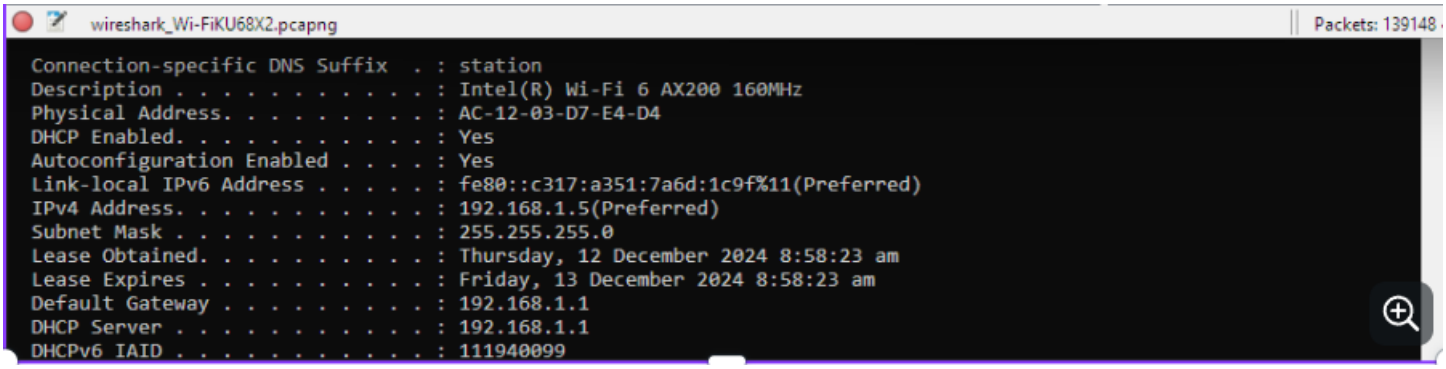
Apriamo wireshark e selezioname la nostra rete (for showcase i just also did an ipconfig showing that it’s associated with a wireless adapater)



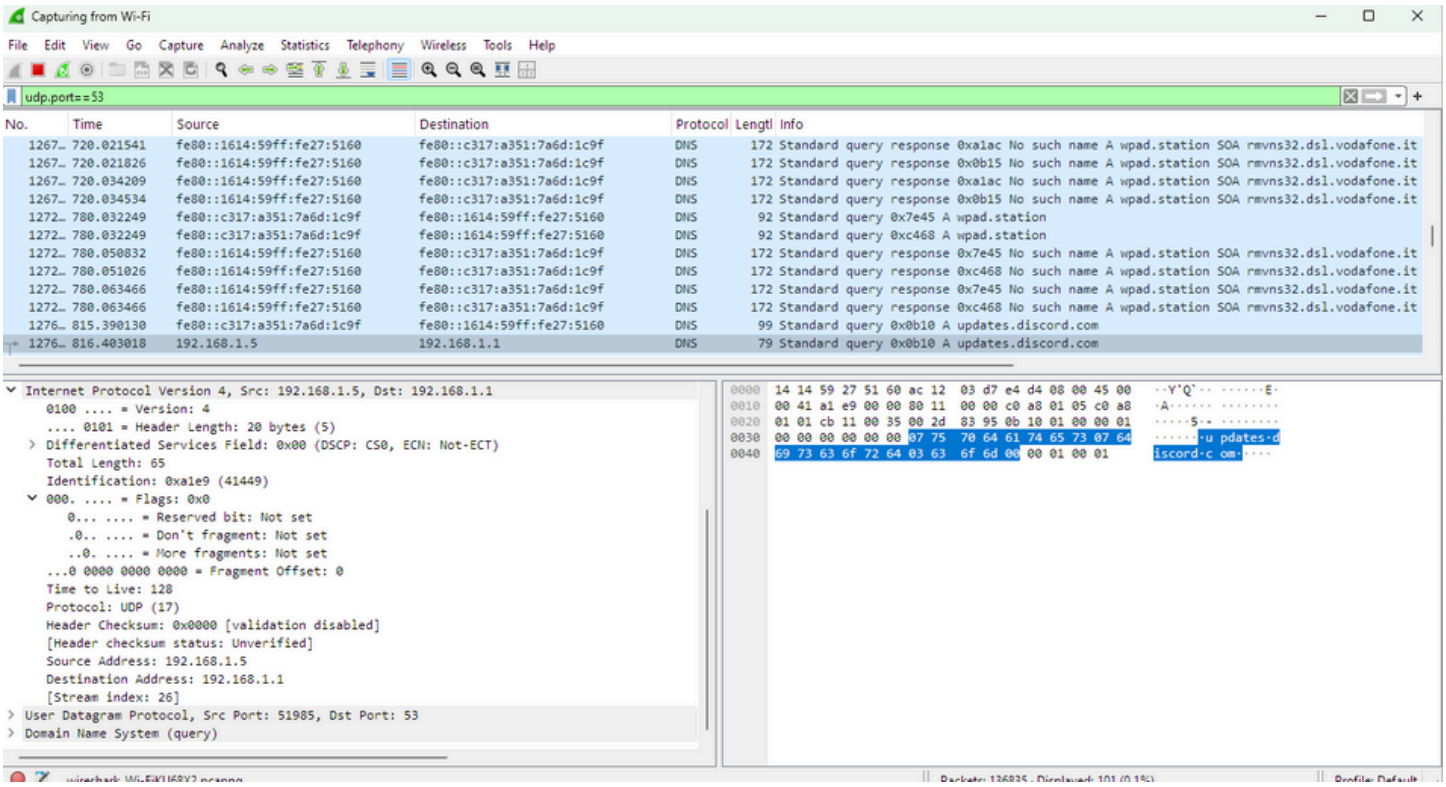
Eseguiamo un ipflush su windows mentre wireshark è in “mezzo” e andremo ad osservare le richieste. Nel filtro possiamo specificare il protocollo UDP (totale) o restringere la ricerca alla porta 53, specifica per il DNS. Poi apriamo il pannello dettagliato e espandiamo l’opzione ethernet2 che contiente l’intestazione i dati/payload e il CRC. (((Crc per il controllo di errori di trasmissione (check) dati e payload es. pacchetti ip* (intestazione: time to live, ipv4 o 6 etc payload richieste http.)))

Vediamo che “source” è associato a alla mia scheda di rete “intel” con indirizzo mac fisico AC120307 e la destinazione è il gateway o il mio router.

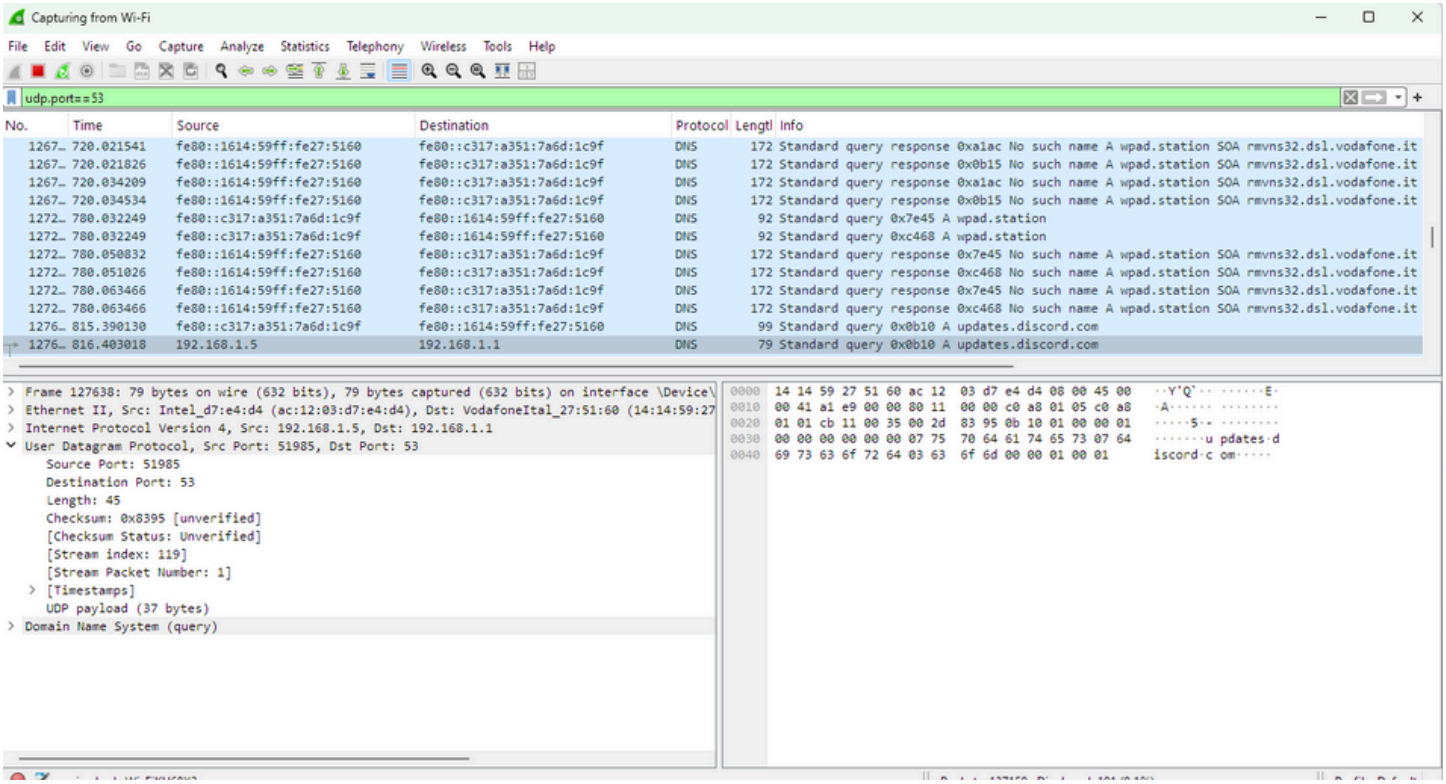




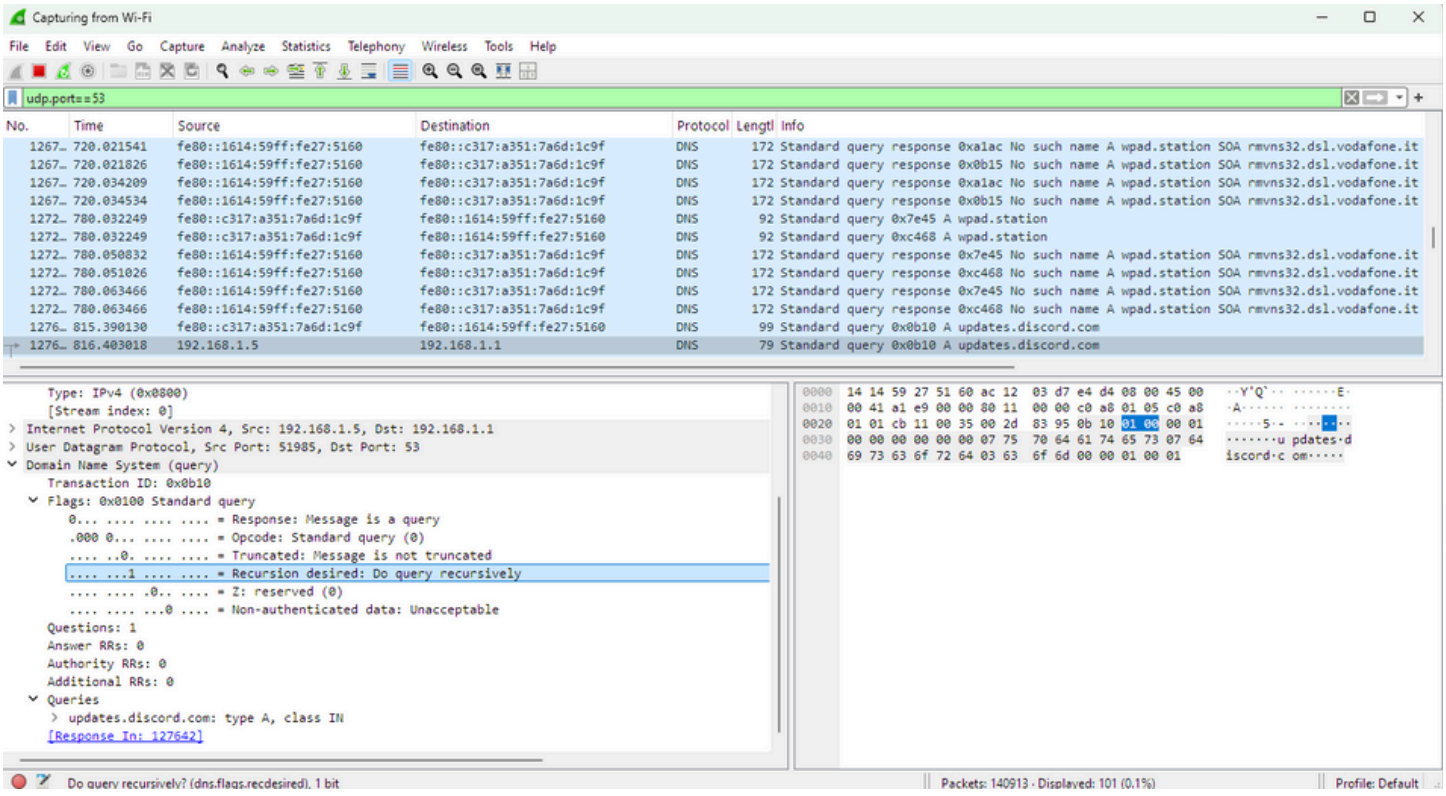
Estendiamo internet protocol e vediamo che l'indirizzo ip del source è associata alla nostra scheda di rete mentre alla destinazione riconfermiamo che è il gateway.



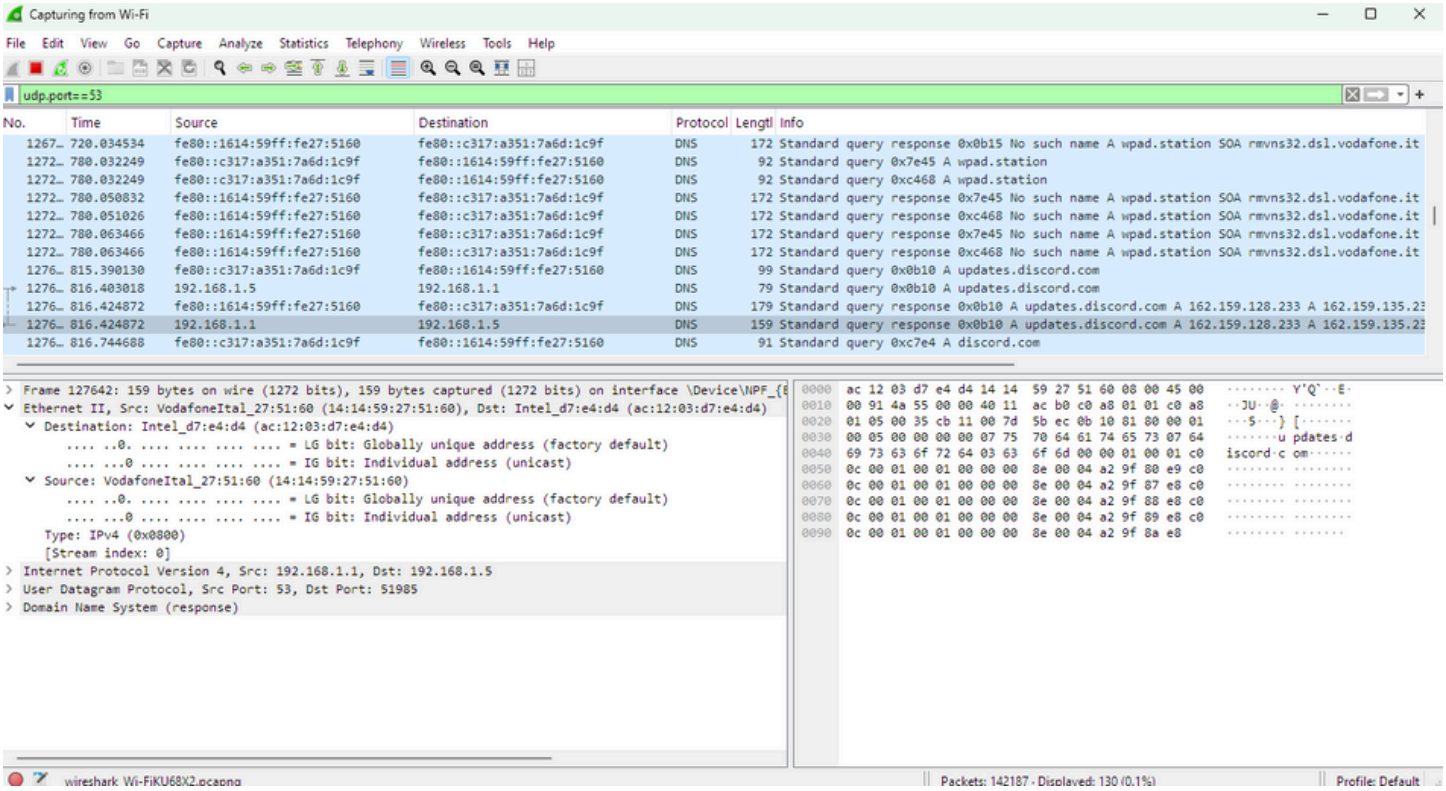
La porta da parte della "sorgente" (in questo caso noi 1.5) è 51985 (casuale/random every time just for this part is entire connection is set to 51985, next time it may be something else). E la destinazione (1.1) sulla porta 53 (DNS) (Standard DNS)



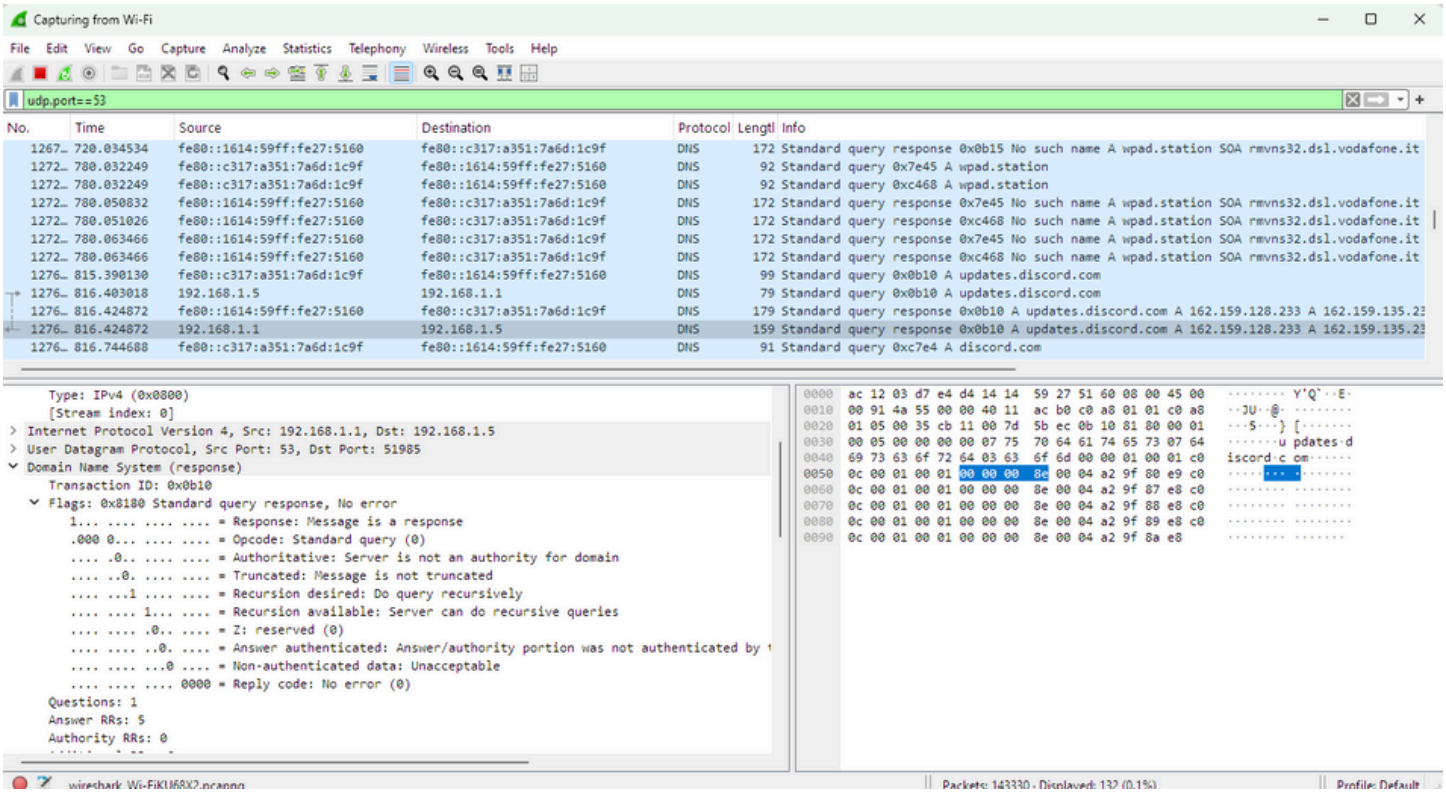
Adesso espandiamo domain system e troviamo il tipo di risposta (il messaggio è una query) il tipo (standard) se è stato "truncated" o se ha dei "byte" riservati e sei dati non authenticati sono "accettabili". A noi interessa per l'esercizio la recursion, ed è impostata su "do query". In caso della risoluzione ricorsiva, il server DNS interroga altri server DNS fino a ottenere una risposta completa da restituire al client. altrimenti se sarebbe impostato su do not query recursively Il server DNS non esegue la risoluzione completa, ma risponde con un riferimento al prossimo server DNS da interrogare (modalità iterativa). il server dns risponderà al client u chi dovrà dopo interrogare per ottenere gli indirizzi ip associti ai nomi di dominio.



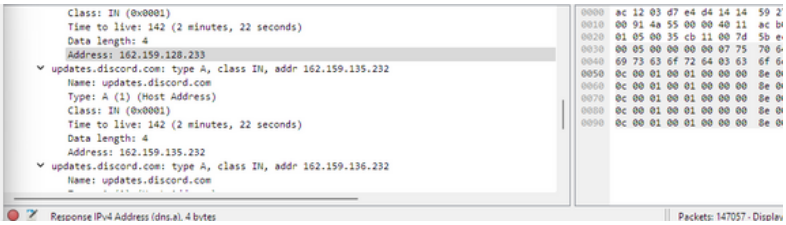
Procediamo con la risposta (ipv4 for showing purposes, you can clearly see there is both ipv4/ipv6 “calls” and answers.) Adesso si sono invertiti i ruoli. la precedente "destinazione" è adesso diventata "sorgente" e Sorgente ora è destinazione <>. Lo confermiamo sia a primo impatto da l'indirizzo ip 1.1 su source ma lo confermiamo con il mac nella sezione ethernet2 (exececise has to be this way) lui sta rispondendo a noi 1.1>1.5 con le liste DNS (per poter associare i nomi di dominio ad indirizzi ip 233 e 232) La porta si è invertita in equalmodo. Porta 53>51985e anche lui in modo recursivo



Stiamo chiedendo al server di fare il "lavoro completo" e lui ci risponde si ok va bene con do recursive. Se il server non supporta la risoluzione ricorsiva, invierà una risposta parziale o iterativa, con informazioni sul server successivo da interrogare.



Scorriamo in basso e controlliamo con nslookup (il nostro server DNS impostato e indirizzo) e con nslookup discord.com tutti gli indirizzi ip associato al dominio discord.com e vediamo che le richieste sono corrette e coincidono, completando l'esercizio. (Type A-ANSWER risponde con la richiesta a Chi è discord.com? discord è 162.x.x.232 e 162.x.x.233 associando nome di dominio a indirizzi ip.)



RECAP

1. Dopo aver avviato Wireshark, filtriamo il traffico DNS utilizzando il filtro udp.port==53. Questo ci permette di osservare sia le query che le risposte DNS.
2. Espandendo l'opzione Ethernet nel pannello dettagliato, vediamo che l'indirizzo MAC sorgente appartiene alla nostra scheda di rete (Intel, MAC AC:12:03:07). La destinazione iniziale è il gateway (1.1).
3. Analizzando il protocollo IP, notiamo che la sorgente è l'indirizzo del nostro dispositivo (1.5), mentre la destinazione è il server DNS del gateway (1.1). La porta UDP sorgente è casuale (51985), mentre la destinazione utilizza la porta standard 53.
4. All'interno del protocollo DNS, analizziamo i dettagli della richiesta:
   - Tipo di messaggio: Query.
   - Tipo di record richiesto: A (per IPv4).
   - Flag di ricorsione: *Recursion Desired* impostato su 1 (richiesta ricorsiva).

Answer analysis:

- Nella risposta, i ruoli di sorgente e destinazione si invertono. Il server DNS (1.1) risponde al client (1.5) fornendo gli indirizzi IP associati al dominio richiesto. La porta sorgente diventa 53, mentre la destinazione utilizza la porta 51985.
- Il flag *Recursion Available* nella risposta indica che il server DNS ha completato la risoluzione ricorsiva, interrogando altri server DNS e restituendo una risposta completa.

Il server DNS del gateway supporta richieste ricorsive, completando la risoluzione dei nomi di dominio (es. discord.com → 162.x.x.232, 162.x.x.233). Se la ricorsione non fosse supportata, il server avrebbe restituito una risposta iterativa con l'indirizzo del server successivo da interrogare. Nlookup per la corretta associazione di nomi di dominio agli indirizzi IP.