IT PROJECT **DOCUMENTATION**

DHCP Discover Flood Attack –
Detection, Analysis & Recovery

MORGAN BURERA

JULY 2025

Google IT Support
Professional Certificate

Ubuntu, Terminal

🧭 Table of Contents

This document is part of a personal project portfolio developed during the Google IT Support Certification. All simulations and analyses are performed in a controlled lab environment.

slow_network.pca
png

I. Project Overview

Simulate a DHCP Discover Flood attack using Yersinia in a virtual lab, capture and analyze the traffic using `tcpdump` and Wireshark, then implement effective client recovery to restore IP allocation.

Furthermore, an investigation was done on a normal DHCP discovery and ip address allocation on my client vm. This small part was get a deeper understanding on how dynamic DHCP works. A short video explaining each steps of the four way handshake is available.

II. Environment Setup

Component	Details
OS	Ubuntu 24.04.2 2 desktop and server (Virtual Machines)
Network Type	ens33 (Wi-fi, coffe shop)
Capture Tools	`tcpdump` (packet capture) Wireshark (analysis of `.pcap` file)
Tools	Yersinia (DHCP attack tool)

III. Test Setup and Process

- **Attack Initialization**
  - Yersinia is launched in DHCP flood mode, rapidly generating DHCP DISCOVER packets with spoofed MAC addresses.

- **Traffic Capture**
  - `tcpdump` is used on interface `ens33` to capture DHCP traffic to a file:

```
sudo tcpdump -i ens33 -n port 67 or port 68 -w dhcp_flood1.pcap
```

- **Symptom Observation**
  - Manual IP request using:

```
sudo dhclient -v ens33
```

  - Repeated DHCP discoveries with no corresponding DHCP offerss indicate a saturated DHCP pool.

- **Attack Termination**
  - On the attacking VM:

```
sudo pkill yersinia
```

  - Or simply : q on yersinia's interface

- **Client Interface Recovery**
    - Clear previous DHCP leases and reset interface:

```
 sudo dhclient -r ens33
 sudo ip link set ens33 down
sudo ip link set ens33 up
sudo dhclient -v ens33
```

- **Verification**
    - Check IP allocation:

```
ip a
 ip route
```

- **Test internet connectivity:**

```
ping 1.1.1.1 -c 3
```

4. Packet Capture and Analysis

- **Total Packets**: 40.000+
- **Attack Signature**: Continuous DHCP Discover packets with source IP ‘0.0.0.0’ and broadcast destination ‘255.255.255.255'.
- **MAC Spoofing Detected** : Yersinia uses randomized MAC addresses (c3:2d:65:55:7f:72 etc.)
- **After attack ends**:
- Valid DHCP exchange observed (DISCOVER → OFFER → REQUEST → ACK).
- Client receives IP ‘192.168.251.135’ from DHCP server ‘192.168.251.254'.

V/ Resolution / Outcome

✅ DHCP server recovered after flood stop
✅ Interface reset successfully
✅ Client reassigned valid IP
✅ Network access restored (Ping success to 1.1.1.1)

VI/ Key Takeaways & Skills Demonstrated

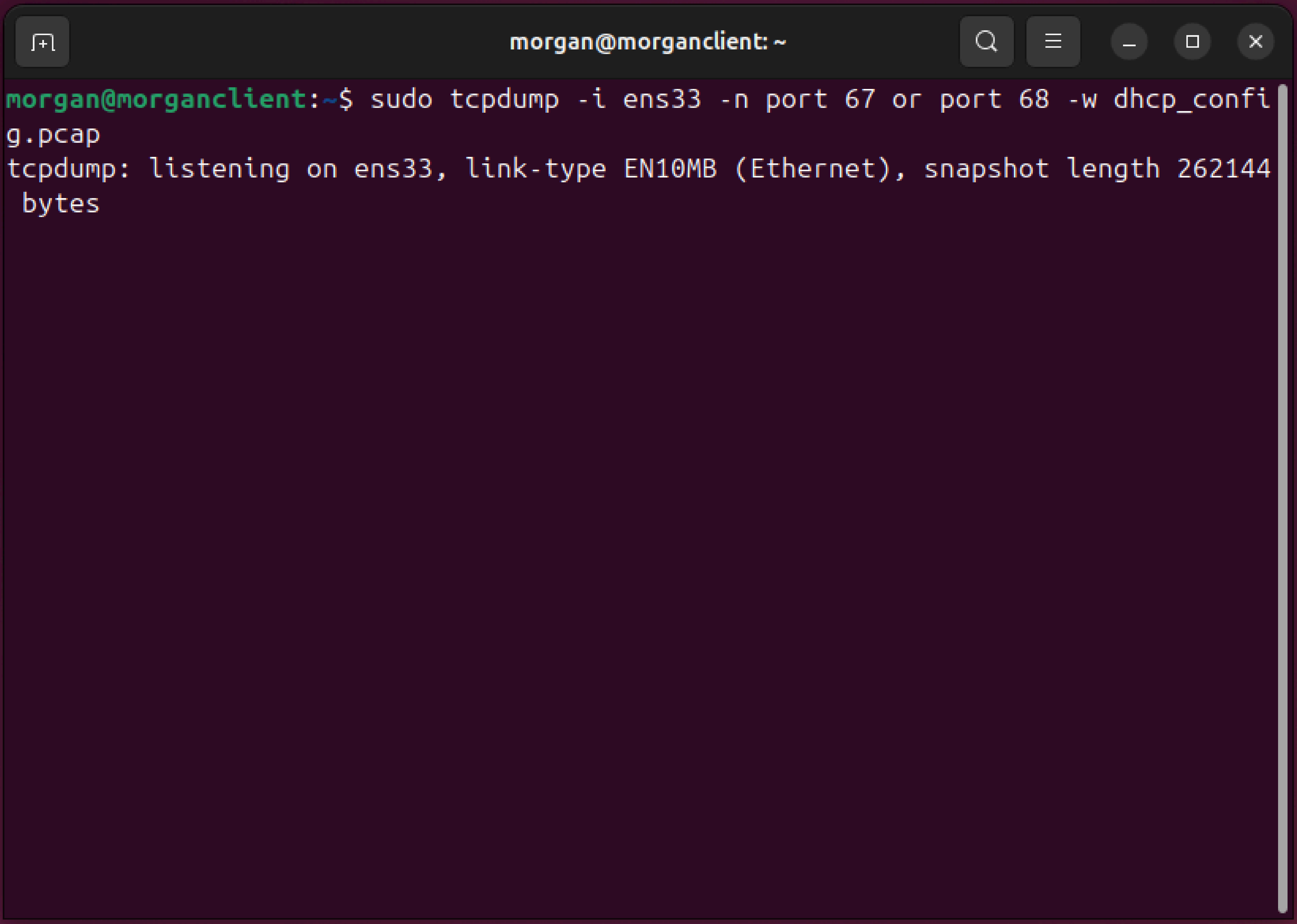
- How DHCP flooding can exhaust server leases
- How to detect DHCP-based denial attacks using packet capture and hands on yersinia's interfacce
- How to manually reset a client interface to request fresh DHCP after attack
- How to correlate CLI output with `.pcap` Wireshark analysis for deeper insight
- Decent understanding of IP addresses's dynamic assignment using DHCP and being able to analyze the capture then explain it

VII/ Screenshots & Visuals

The screenshots section of this part being too large I put them on another file.

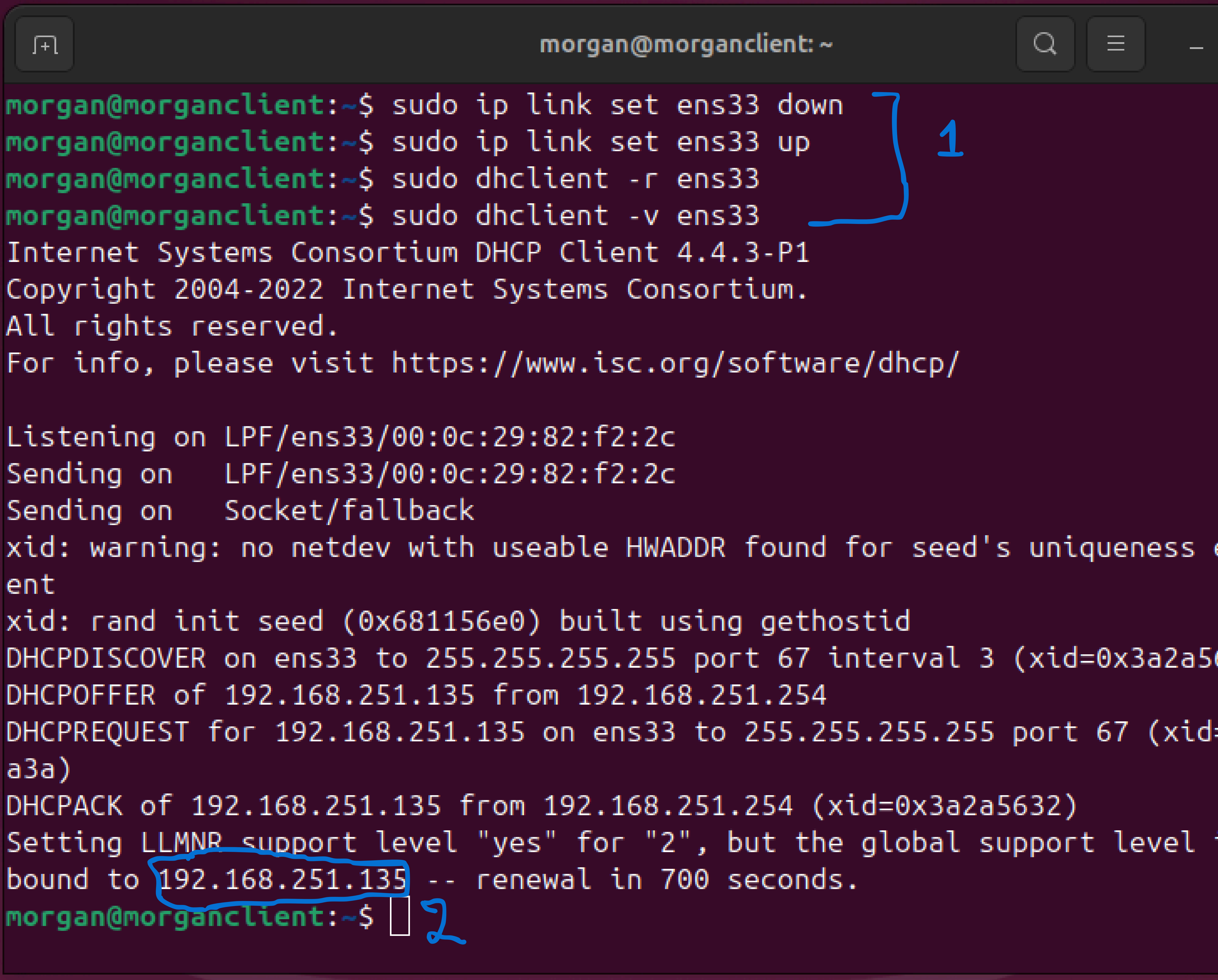
VIII/ Part 2, bonus : Interpreting a standard DHCP exchange

Breackdown with screenshots



```
morgan@morganclient:~$ sudo tcpdump -i ens33 -n port 67 or port 68 -w dhcp_confi
g.pcap
tcpdump: listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144
 bytes
```

I start capturing the packets on the right ports using tcdump.



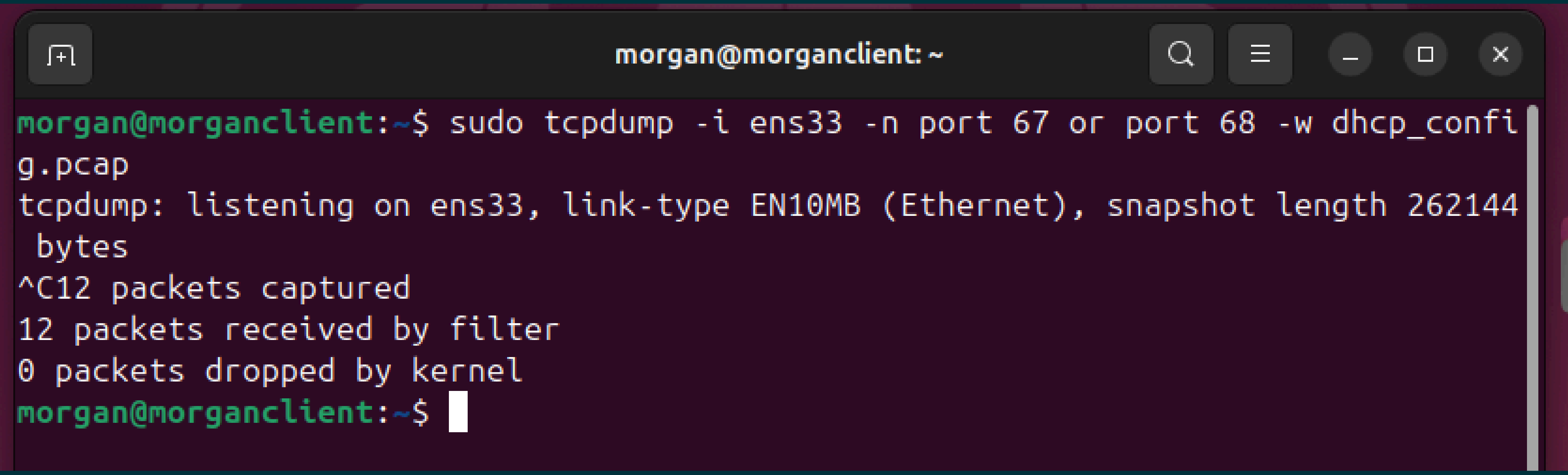
```
morgan@morganclient:~$ sudo ip link set ens33 down
morgan@morganclient:~$ sudo ip link set ens33 up
morgan@morganclient:~$ sudo dhclient -r ens33
morgan@morganclient:~$ sudo dhclient -v ens33
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens33/00:0c:29:82:f2:2c
Sending on   LPF/ens33/00:0c:29:82:f2:2c
Sending on   Socket/fallback
xid: warning: no netdev with useable HWADDR found for seed's uniqueness e
ent
xid: rand init seed (0x681156e0) built using gethostid
DHCPDISCOVER on ens33 to 255.255.255.255 port 67 interval 3 (xid=0x3a2a5
DHCPOFFER of 192.168.251.135 from 192.168.251.254
DHCPREQUEST for 192.168.251.135 on ens33 to 255.255.255.255 port 67 (xid=
a3a)
DHCPACK of 192.168.251.135 from 192.168.251.254 (xid=0x3a2a5632)
Setting LLMNR support level "yes" for "2", but the global support level "
bound to 192.168.251.135 -- renewal in 700 seconds.
morgan@morganclient:~$
```

1. Reset of the network interface (ens33, Wi-fi) and manually requesting and ip address with ‘dhclient -v’

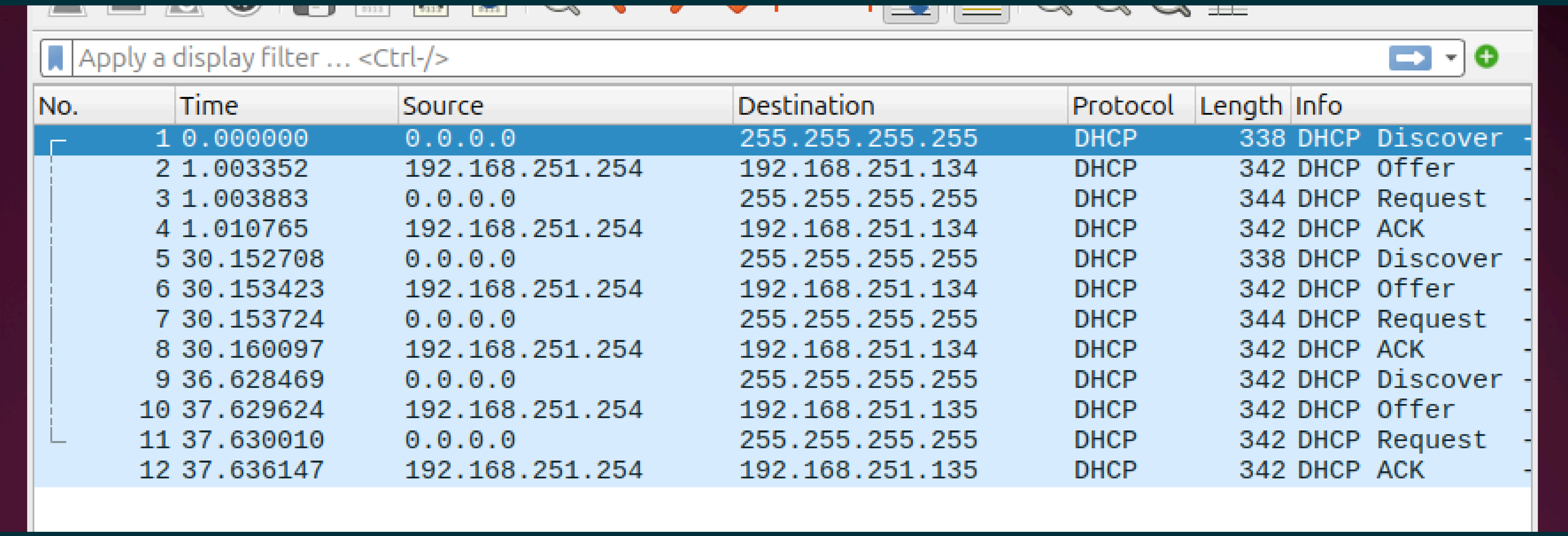
2. The IP address has been assigned.

We can stop the capture and inspect it



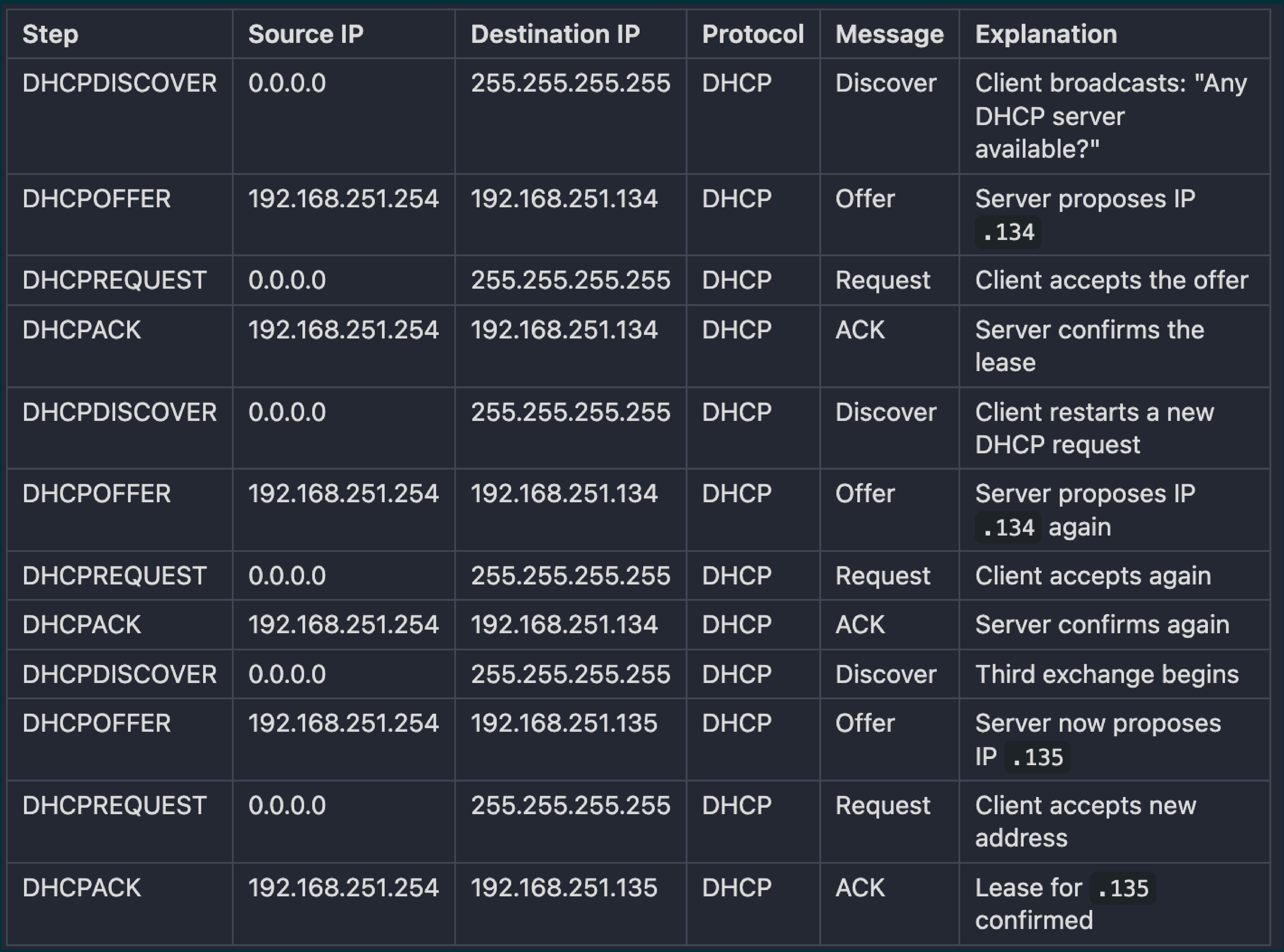
```
morgan@morganclient:~$ sudo tcpdump -i ens33 -n port 67 or port 68 -w dhcp_confi
g.pcap
tcpdump: listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144
 bytes
^C12 packets captured
12 packets received by filter
0 packets dropped by kernel
morgan@morganclient:~$ 
```

dhc_config.pcap folder's content on Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	338	DHCP Discover
2	1.003352	192.168.251.254	192.168.251.134	DHCP	342	DHCP Offer
3	1.003883	0.0.0.0	255.255.255.255	DHCP	344	DHCP Request
4	1.010765	192.168.251.254	192.168.251.134	DHCP	342	DHCP ACK
5	30.152708	0.0.0.0	255.255.255.255	DHCP	338	DHCP Discover
6	30.153423	192.168.251.254	192.168.251.134	DHCP	342	DHCP Offer
7	30.153724	0.0.0.0	255.255.255.255	DHCP	344	DHCP Request
8	30.160097	192.168.251.254	192.168.251.134	DHCP	342	DHCP ACK
9	36.628469	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover
10	37.629624	192.168.251.254	192.168.251.135	DHCP	342	DHCP Offer
11	37.630010	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request
12	37.636147	192.168.251.254	192.168.251.135	DHCP	342	DHCP ACK

Packet Breakdown (Standard DHCP Exchange), written on Obsidian for a better display



Step	Source IP	Destination IP	Protocol	Message	Explanation
DHCPDISCOVER	0.0.0.0	255.255.255.255	DHCP	Discover	Client broadcasts: "Any DHCP server available?"
DHCPOFFER	192.168.251.254	192.168.251.134	DHCP	Offer	Server proposes IP `.134`
DHCPREQUEST	0.0.0.0	255.255.255.255	DHCP	Request	Client accepts the offer
DHCPACK	192.168.251.254	192.168.251.134	DHCP	ACK	Server confirms the lease
DHCPDISCOVER	0.0.0.0	255.255.255.255	DHCP	Discover	Client restarts a new DHCP request
DHCPOFFER	192.168.251.254	192.168.251.134	DHCP	Offer	Server proposes IP `.134` again
DHCPREQUEST	0.0.0.0	255.255.255.255	DHCP	Request	Client accepts again
DHCPACK	192.168.251.254	192.168.251.134	DHCP	ACK	Server confirms again
DHCPDISCOVER	0.0.0.0	255.255.255.255	DHCP	Discover	Third exchange begins
DHCPOFFER	192.168.251.254	192.168.251.135	DHCP	Offer	Server now proposes IP `.135`
DHCPREQUEST	0.0.0.0	255.255.255.255	DHCP	Request	Client accepts new address
DHCPACK	192.168.251.254	192.168.251.135	DHCP	ACK	Lease for `.135` confirmed

VIII/ Optional Enhancements, Reflection

I really enjoyed doing this project — but the issue was surprisingly easy to fix. So I thought to myself:

- **But What About a Real Network Attack?**
  - **Nothing guarantees the attacker will stop, right?**
    If it were me, I'd take as much time as needed — or disrupt the network just enough to carry out whatever malicious plan I had in mind.

  - In that case :
    - ▪ The DHCP server becomes overwhelmed and can no longer respond (the IP pool is exhausted by fake requests).
    - ▪ The legitimate client fails to obtain an IP → Denial of Service (DoS).

  - **How to Stop the Attack in Practice?**
    - ▪ **This goes beyond the client's responsibilities, but for awareness:**

⚠️ **Actions :**

- ○ (Network) Enable DHCP Snooping (on managed switches)
- ○ (Security) Identify the malicious machine using 'tcpdump' or DHCP server logs
- ○ (OS) Isolate the attacker (block IP or MAC address)