DHCP Starvation and Spoofing Attack

CSE-406

Submitted by

Md. Miraj Hasan
ID: 2005084
Wahid Al Azad Navid
ID: 2005089

July 27, 2025

Contents

1 Introduction

In this project, we simulate a DHCP Starvation and Spoofing Attack using NS-3. This attack exploits vulnerabilities in the DHCP protocol, where a rogue client exhausts the IP pool of a legitimate DHCP server and then sets up a rogue DHCP server to serve fake IPs, potentially redirecting victim traffic.

2 Attack Methodology

2.1 Steps of the Attack

The attack was implemented in the following steps:

1. **DHCP Starvation:** The attacker sends multiple DHCP DISCOVER messages with spoofed MAC addresses to exhaust the legitimate server's IP pool.

2. **DHCP Spoofing:** After the starvation, the attacker runs a rogue DHCP server to respond faster than the legitimate server to new clients.

3. **Hijacking:** New clients receive IP configurations from the rogue server, potentially with malicious gateway/DNS settings.

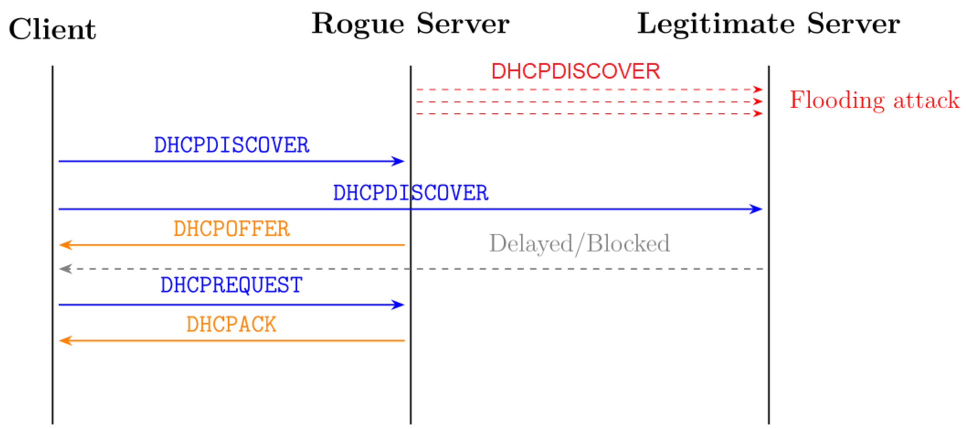

Figure 1: Timing Diagram of the Attack

2.2 NS-3 Simulation Environment

The simulation was built using NS-3 with the following structure:

- **Number of Clients:** Up to 140
- **Legitimate DHCP Server:** Responds with slower delay (3ms)
- **Rogue DHCP Server:** Responds with faster delay (1ms) and a large pool
- **Topology:** All nodes connected over CSMA network
- **Attacker:** First client node acts as attacker

2.3 Simulation Code

Below is a snippet of the main simulation code:

Listing 1: NS-3 Simulation Code Snippet

```
Ptr<DhcpServerApp> rogue = CreateObject<DhcpServerApp>();
rogue->Setup(Ipv4Address("192.168.100.1"), rogue_pool, port, MilliSeconds(1));

Ptr<DhcpServerApp> legit = CreateObject<DhcpServerApp>();
legit->Setup(Ipv4Address("10.10.10.1"), 100, port, MilliSeconds(3));

for (uint32_t i = 0; i < numClients; ++i) {
    Ptr<Node> node = clients.Get(i);
    Ptr<DhcpClientApp> client = CreateObject<DhcpClientApp>();
    client->Setup(broadcastAddr, 67);

    if (i == 0) client->SetIsAttacker(true); // attacker node

    double jitter = (rand() % 100) / 1000.0;
    client->SetStartTime(Seconds(2.0 + i * 0.2 + jitter));
    client->SetStopTime(Seconds(20.0));
    node->AddApplication(client);
}
```

3 Simulation Results

3.1 Observed Outputs

Attacker

The attacker was able to send multiple DHCP requests with spoofed MACs, causing the legitimate server pool to be exhausted.

Victim Clients

Many clients received IP addresses from the rogue server, indicating the attack's success.

Server Logs

The legitimate server stopped responding after pool exhaustion, while the rogue server continued to assign IPs.

3.2 Snapshots

3.2.1 DHCP Starvation Phase

During the starvation phase, the attacker floods the network with a large number of DHCPDISCOVER packets, each of which carries a spoofed MAC address. This rapidly exhausts the legitimate DHCP server's IP address pool, making it unable to serve legitimate clients. In particular, the rogue DHCP server initially remains silent, allowing the legitimate server to deplete its pool. Once starvation is complete, the rogue server begins responding to client requests, ensuring that it becomes the only responder in the network.

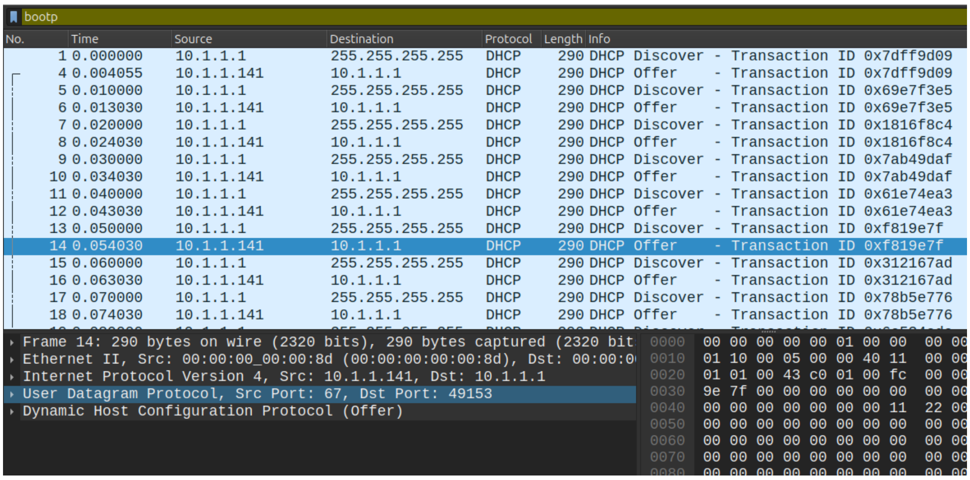

Figure 2: Wireshark Snapshot: Attacker Flooding DISCOVER Packets with Spoofed MACs, Forcing Starvation

3.2.2 DHCP Spoofing and Starvation Phases

The hybrid attack consists of two key phases - spoofing during availability and spoofing after starvation — both of which highlight the dominance of the rogue DHCP server over the IP allocation process.

Phase 1: Rogue Wins the Race (Spoofing with Available Legit Server) In this phase, the legitimate DHCP server still has valid IP addresses in its pool, but the rogue server, acting as a man-in-the-middle, responds faster to client DHCPDISCOVER messages. Due to this faster response, clients accept the DHCPOFFER of the rogue server and subsequently are assigned fake IP addresses, completely bypassing the legitimate server.

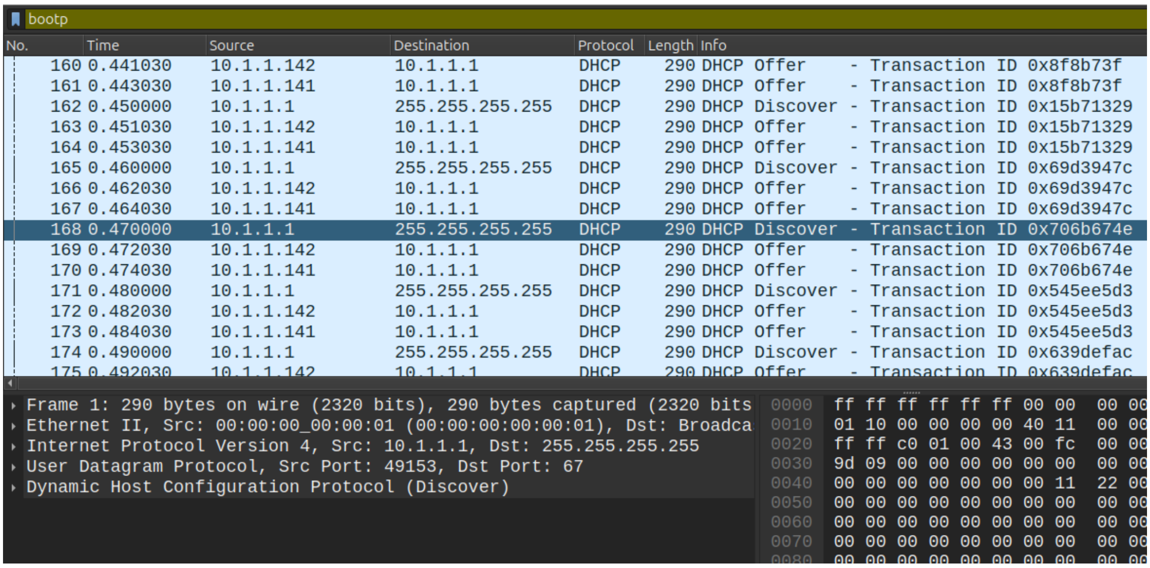

Figure 3: Wireshark Capture: Rogue Server Responding Faster Than Legitimate Server Despite Available IPs

Phase 2: Starvation Complete (Only Rogue Offers Remain) After the legitimate server's IP pool is exhausted via DHCP starvation, it no longer responds to client DHCPDISCOVER messages. At this point, only the rogue server sends DHCPOFFER packets, effectively taking over the DHCP process without competition. Clients have no alternative but to accept IPs from the rogue server.

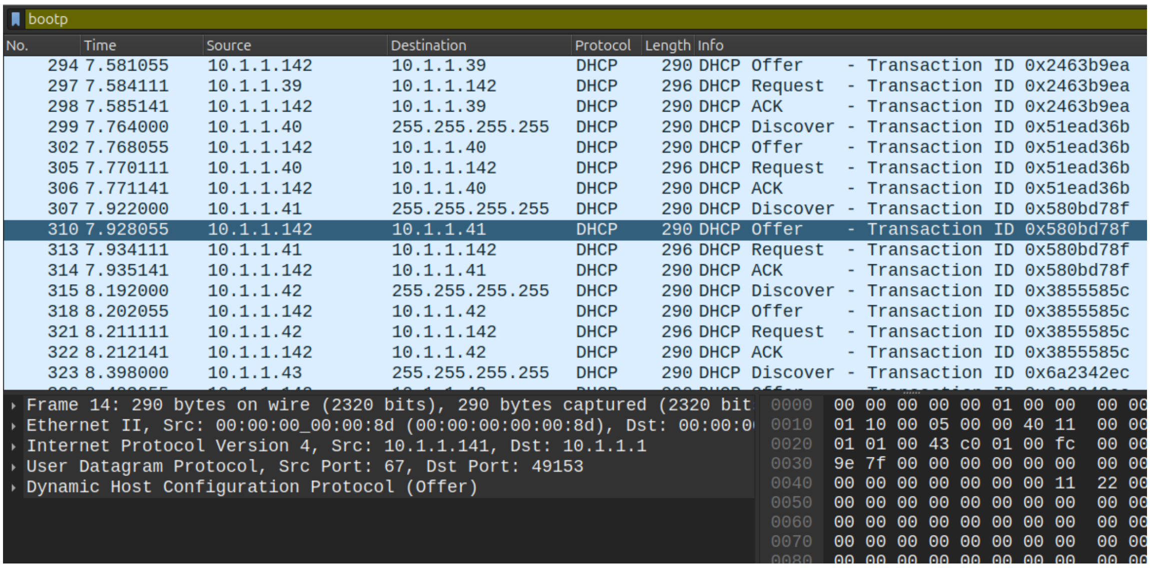

Figure 4: Wireshark Capture: Rogue Server Offers IPs After Legitimate Server is Starved

3.3 Graphical Analysis

The following graphs analyze the DHCP starvation and spoofing attack in terms of the rogue coverage percentage (i.e., the percentage of clients served by the rogue server out of all clients). These plots help visualize how different parameters (number of clients, rogue pool size, and simulation time) influence the attack's effectiveness.

Impact of Number of Clients (Fixed Running Time = 25s) As shown in Figure 5, the rogue coverage increases as the number of clients increases from 100 to 140 though the rogue coverage percentage is decreasing. The rogue server's ability to handle more clients is evident, especially when the rogue pool size is larger (250). This is because a higher client count creates more opportunities for the rogue server to dominate the DHCP offer-response race, and a larger rogue pool ensures it doesn't run out of IPs.

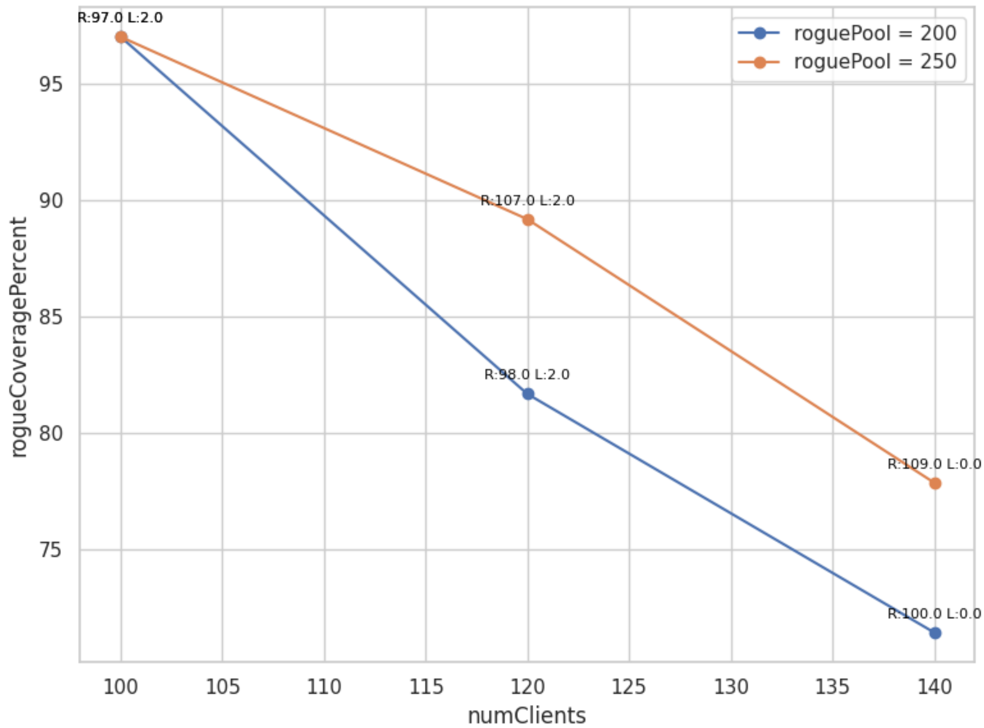

Figure 5: Rogue Coverage % vs Number of Clients (Fixed Running Time = 25s)

Impact of Running Time (Fixed Clients = 120) Figure 6 demonstrates that increasing simulation time allows more clients to complete the DHCP transaction, thereby giving the rogue server more chances to respond first. This results in increased rogue coverage, especially when the rogue pool size is sufficient to support the client load. The attack becomes nearly fully successful at longer durations when paired with a rogue pool of 250.

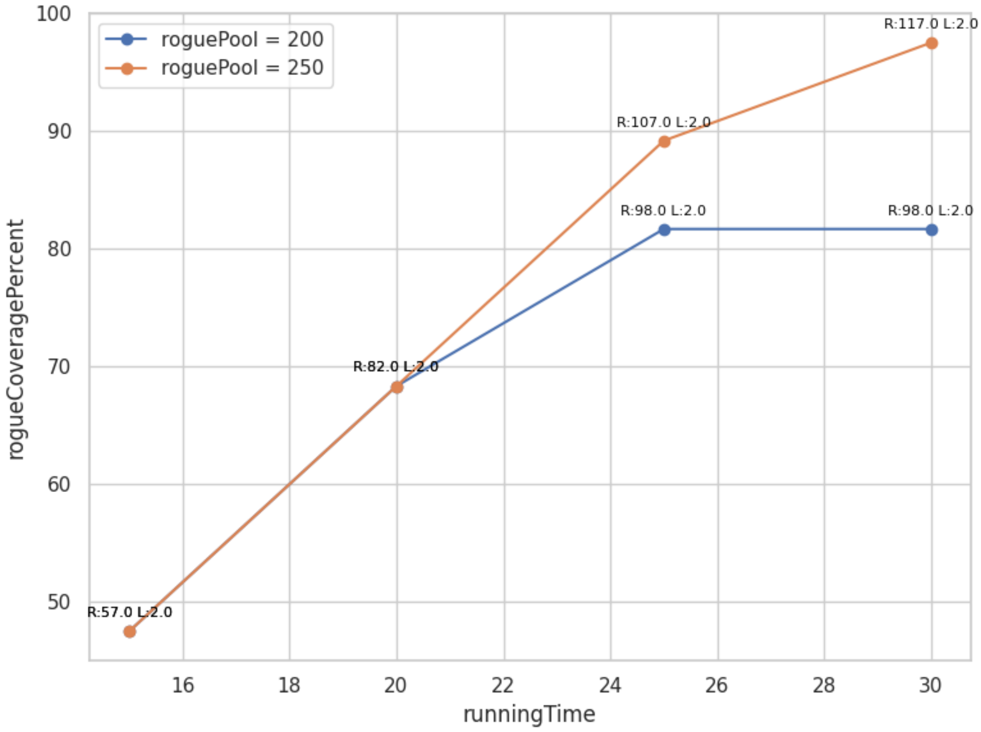

Figure 6: Rogue Coverage % vs Running Time (Fixed Clients = 120)

Impact of Running Time (Fixed Rogue Pool = 250) In Figure 7, with a large rogue pool fixed at 250, the attack's success is predominantly influenced by simulation time and number of clients. For lower client counts, full coverage is achieved quickly. But for higher numbers like 140, extended time is needed to reach near-complete coverage. This suggests that when the rogue server is not limited by IP pool exhaustion, it can dominate the DHCP process over time.

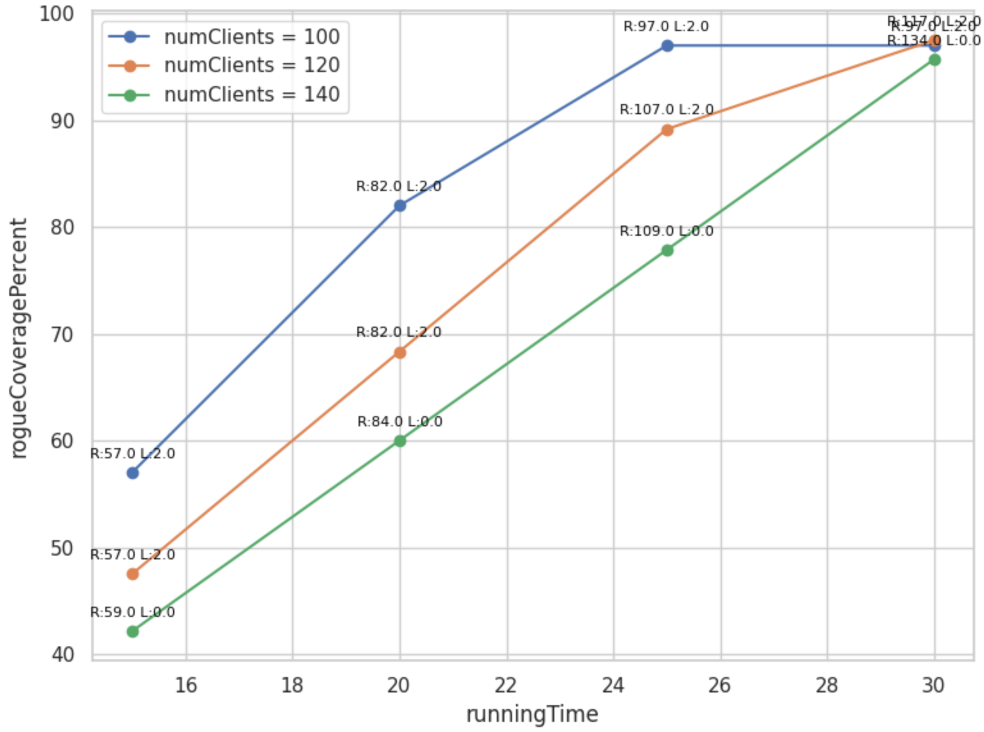

Figure 7: Rogue Coverage % vs Running Time (Fixed Rogue Pool Size = 250)

3.4 Tabular Summary

Table 1: DHCP Attack Simulation Summary

numClients	roguePool	runningTime	rogueAssigned	legitAssigned	totalAssigned	rogueSuccessPercent	rogueCoveragePercent
100	200	15	57	2	59	96.61	57.00
100	200	20	82	2	84	97.62	82.00
100	200	25	97	2	99	97.98	97.00
100	200	30	97	2	99	97.98	97.00
100	250	15	57	2	59	96.61	57.00
100	250	20	82	2	84	97.62	82.00
100	250	25	97	2	99	97.98	97.00
100	250	30	97	2	99	97.98	97.00
120	200	15	57	2	59	96.61	47.50
120	200	20	82	2	84	97.62	68.33
120	200	25	98	2	100	98.00	81.67
120	200	30	98	2	100	98.00	81.67
120	250	15	57	2	59	96.61	47.50
120	250	20	82	2	84	97.62	68.33
120	250	25	107	2	109	98.17	89.17
120	250	30	117	2	119	98.32	97.50
140	200	15	59	0	59	100.00	42.14
140	200	20	84	0	84	100.00	60.00
140	200	25	100	0	100	100.00	71.43
140	200	30	100	0	100	100.00	71.43
140	250	15	59	0	59	100.00	42.14
140	250	20	84	0	84	100.00	60.00
140	250	25	109	0	109	100.00	77.86
140	250	30	134	0	134	100.00	95.71

4 Analysis

From the tabulated data and graphs, we observe:

- **Effect of Time:** Longer simulation time allows more clients to request IPs, increasing rogue assignment.
- **Effect of Rogue Pool Size:** When the rogue pool increases from 200 to 250, more clients get rogue IPs.
- **Success Percentage:** Rogue success rate remains consistently high (above 96%), showing the attacker replies faster.
- **Coverage Limitation:** Even with high success rate, coverage is limited by the total simulation time and number of clients.
- **Complete Spoofing:** In cases like 140 clients and rogue pool of 250, the rogue server alone handles all IP assignments.

5 Why the Attack is Successful

The success of the attack is due to some factors:

- **IP Pool Exhaustion:** The attacker floods the network with DHCPDISCOVER packets using spoofed MAC addresses, quickly consuming all IPs from the legitimate server. As a result, genuine clients receive no IPs from the trusted source.
- **Faster Rogue Response:** The rogue DHCP server is configured to respond faster than the legitimate one. Since clients accept the first DHCPOFFER they receive, the rogue server gains control over IP assignment.
- **Lack of Authentication:** DHCP lacks built-in authentication. Clients cannot distinguish between legitimate and rogue servers, making spoofing trivial.

6 Implemented Defense Mechanisms

To mitigate the effects of DHCP starvation and spoofing attacks, we implemented defense mechanisms on both the server and client sides.

6.1 Server-side Starvation Defense

The legitimate DHCP server was enhanced to detect a high rate of DHCPDISCOVER packets within a short time window (e.g., more than 20 requests per second). If this threshold is exceeded, the server temporarily drops incoming requests, assuming an ongoing starvation attack.

- **Trigger:** More than 20 DISCOVERs in 1 second.
- **Effect:** Excess traffic is ignored, preserving the IP pool.
- **Reasoning:** Legitimate clients do not send bursts of requests; such behavior is typical of an attack.

6.2 Client-side Spoofing Defense

Each client maintains a whitelist of trusted DHCP server IPs. When a DHCPOFFER or DHCPACK is received, the client verifies the sender's IP address. If the server is not in the whitelist, the packet is dropped and ignored.

- **Trigger:** DHCPOFFER/DHCPACK received from untrusted IP.
- **Effect:** Client ignores rogue DHCP responses.
- **Reasoning:** Prevents acceptance of configuration from unauthorized DHCP servers.

6.3 Evaluation of Defenses

The graph below summarizes the impact of enabling different combinations of the above defense mechanisms. Each bar shows how many clients the rogue or legitimate server served in different configurations.

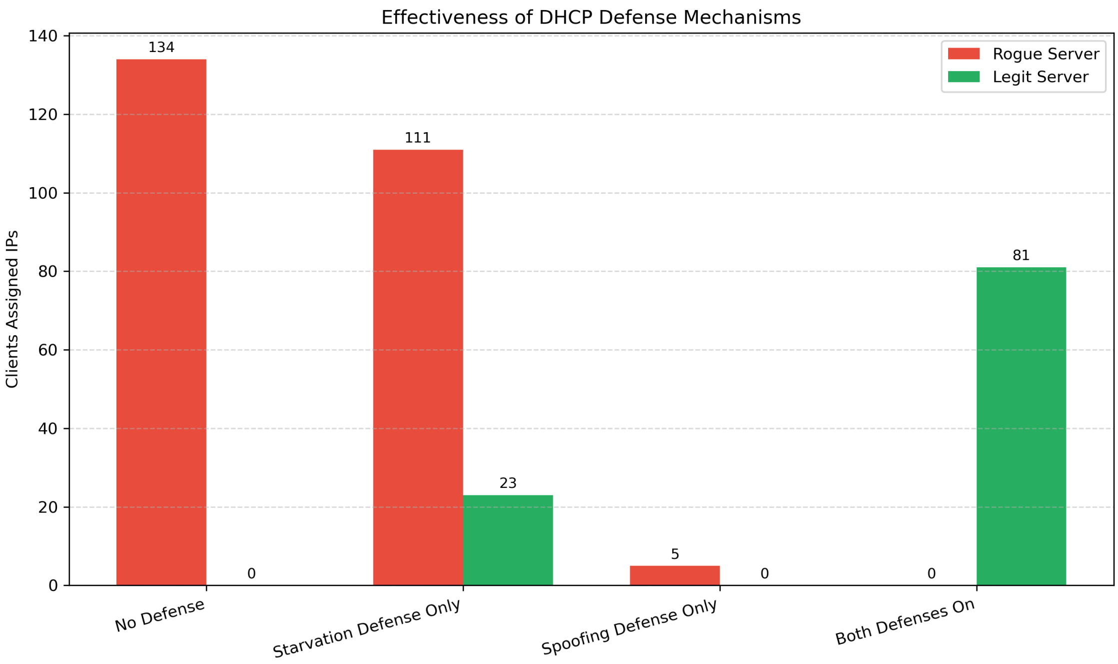

Figure 8: Comparison of DHCP Assignment With/Without Defenses

As evident:

- Without defense, all clients received IPs from the rogue server.
- With only spoofing defense, few rogue responses were filtered.
- With only starvation defense, some legitimate responses were preserved.
- With both defenses, the legitimate server was able to dominate IP assignments.

7 Conclusion

The DHCP Starvation and Spoofing attack was successfully simulated in NS-3. By exhausting the legitimate server's pool and deploying a faster rogue server, the attacker could control IP allocation to clients. This highlights the importance of DHCP snooping and port security mechanisms in real-world networks.

8 Contribution of Team Members

Md. Miraj Hasan: Designed and implemented the DHCP starvation using ns-3. Assisted in the analysis of simulation results and handled parameter tuning (number of clients, rogue pool size, etc.), prepared visualization scripts using Python (Matplotlib, Seaborn, Pandas) and contributed in writing the reports. Evaluated the results of the defense.

Wahid Al Azad Navid: Designed and implemented the spoofing attack simulation using ns-3. Developed the simulation logic, analyzed the simulation results and ensured accurate logging of client-server interactions including interpretation of graphs and tables for evaluating attack success. Designed and implemented the defense. Did Wireshark analysis and contributed in writing reports.