CYBERSECURITY :

DHCP STARVATION AND SPOOFING

- Md Miraj Hasan (2005084)
- Wahid Al Azad Navid (2005089)

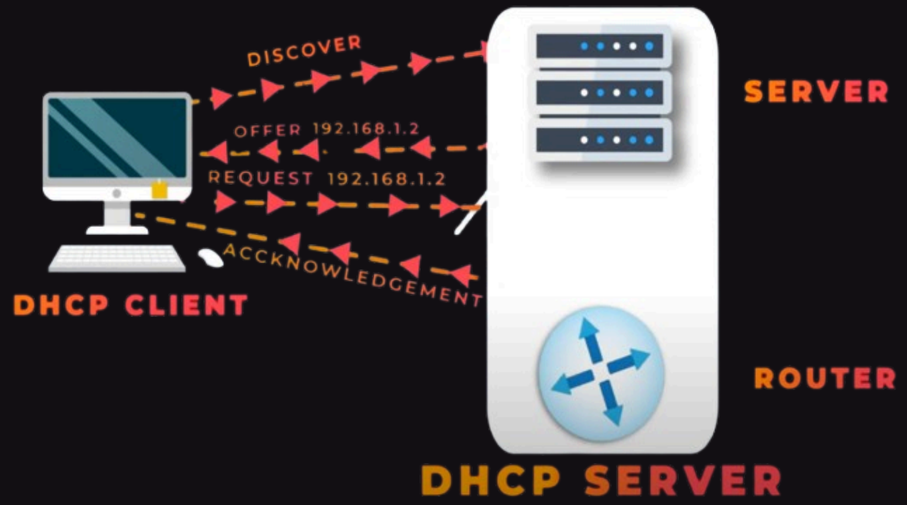
INTRODUCTION

- What is DHCP?

DHCP (Dynamic Host Configuration Protocol) is a network protocol that automatically assigns IP addresses and other network settings to devices when they join a network.

- DHCP – Dynamic Host Configuration Protocol

It ensures each device gets a unique IP without user involvement.



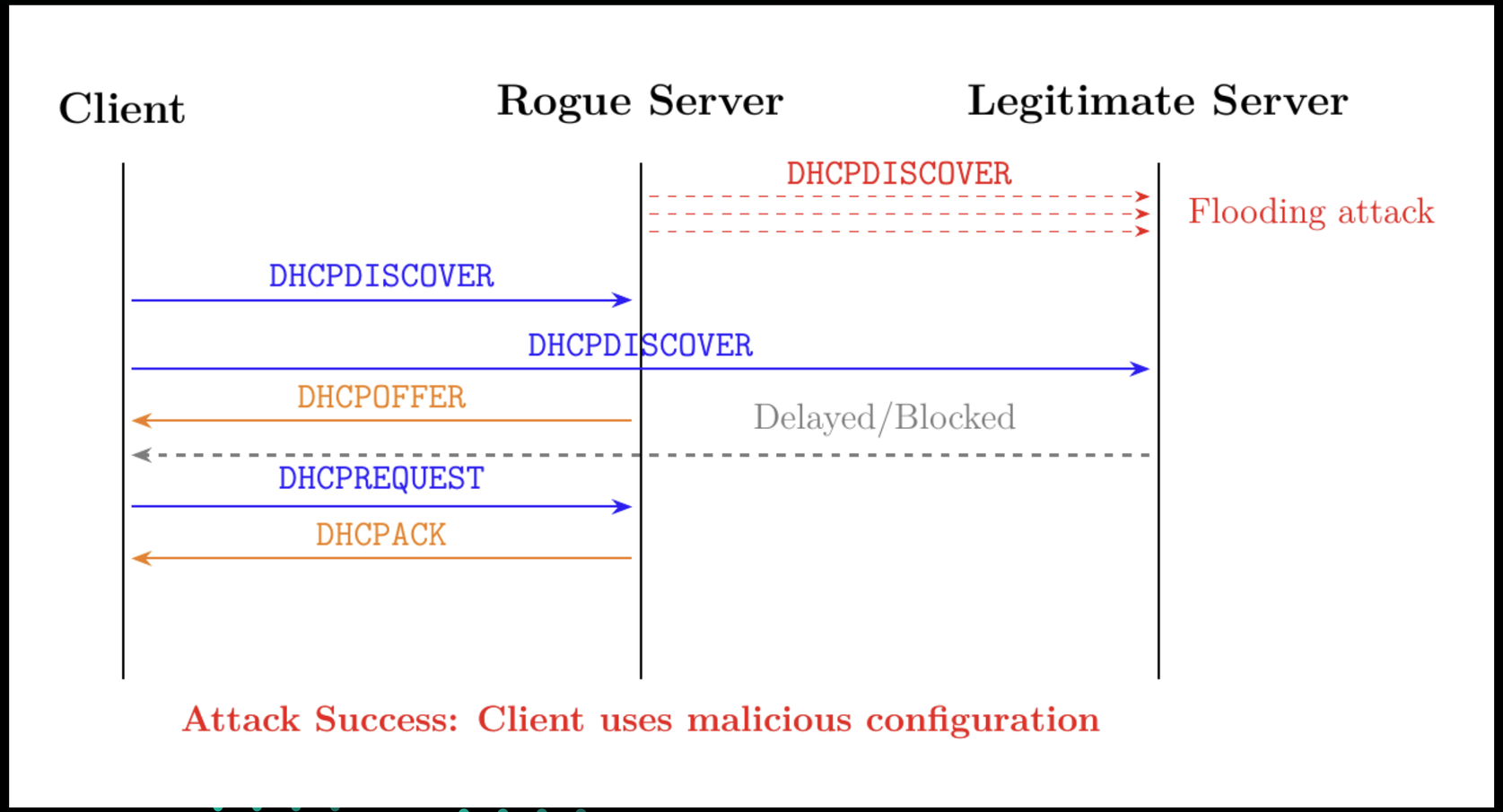
DHCP STARVATION

- An attacker floods the DHCP server with fake DHCPDISCOVER requests using spoofed MAC addresses, quickly exhausting its IP pool. As a result, legitimate clients are denied IP addresses.

DHCP SPOOFING

- After starving the legitimate server, the attacker runs a rogue DHCP server that responds to clients with malicious IP configurations, gaining control over their traffic.

ATTACK DESIGN



CONSEQUENCES OF THESE ATTACKS

- Network access denied to legitimate users (starvation)
- Users unknowingly connect to rogue server (spoofing)
- Leads to data interception, redirection, or MITM attacks
- Can bring entire networks down

ATTACK IMPLEMENTATION (OUR SIMULATION)

- NS-3 based setup with legitimate and rogue DHCP servers
- Multiple clients, 1 acting as attacker
- Rogue sends spoofed DHCP replies / floods DISCOVERs
- Evaluated both starvation and spoofing independently & together

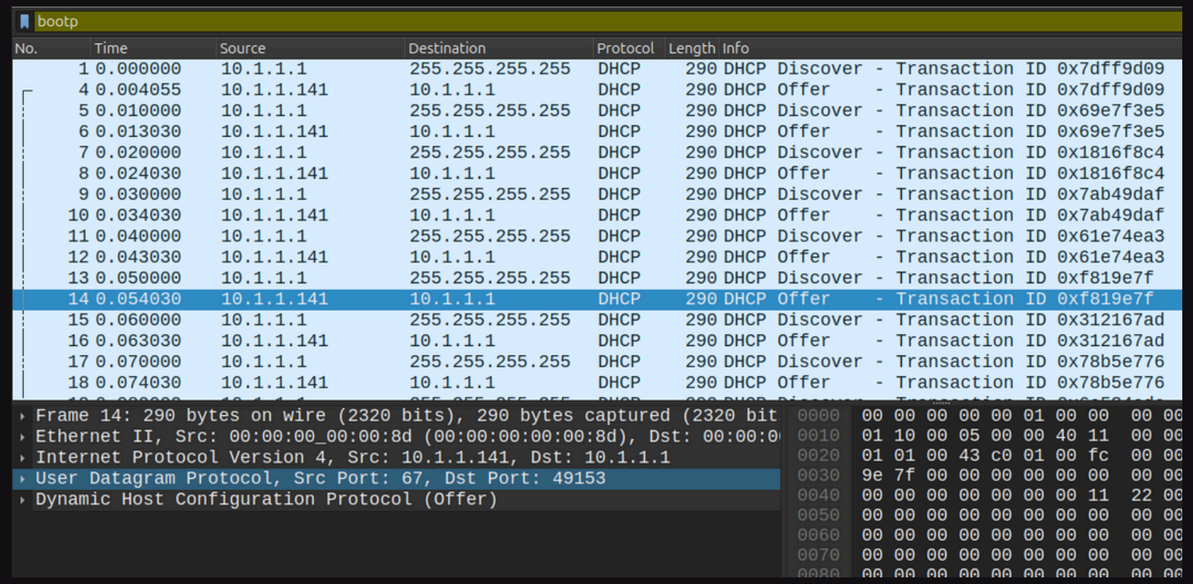
ATTACK IMPLEMENTATION (OUR SIMULATION)

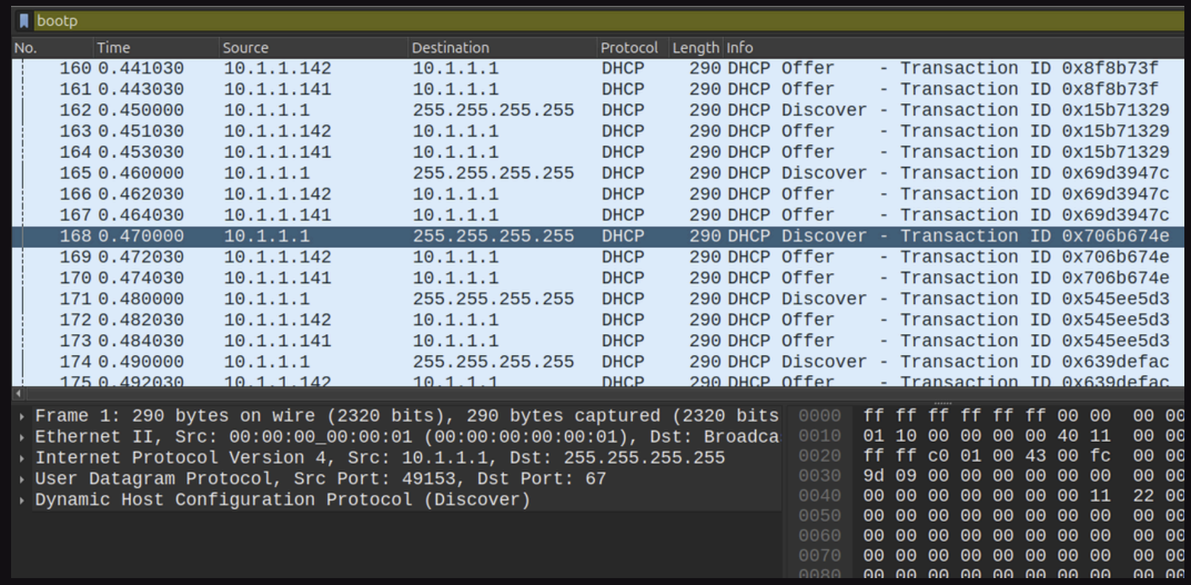
```
int rogue_pool = 250;
// Rogue DHCP Server (responds fast)
Ptr<DhcpServerApp> rogue = CreateObject<DhcpServerApp>();
rogue->Setup(Ipv4Address("192.168.100.1"), rogue_pool, port, MilliSeconds(1)); // fast
rogue->SetStartTime(Seconds(3.0));
rogueServer.Get(0)->AddApplication(rogue);

// Legitimate DHCP Server (slower)
Ptr<DhcpServerApp> legit = CreateObject<DhcpServerApp>();
legit->Setup(Ipv4Address("10.10.10.1"), 100, port, MilliSeconds(3)); // slow
legit->EnableDefense(enableStarvatingDefense); // Enable defense mechanism
legitServer.Get(0)->AddApplication(legit);
legit->SetStartTime(Seconds(0.0));
```

```
    if (m_isAttacker)
    {
        for (uint32_t i = 0; i < m_numSpoofed; ++i)
        {
            Simulator::Schedule(Seconds(1.0) + i * m_interval,
                                &DhcpClientApp::SendSpoofedDiscover,
                                this,
                                i);
        }
    }
```

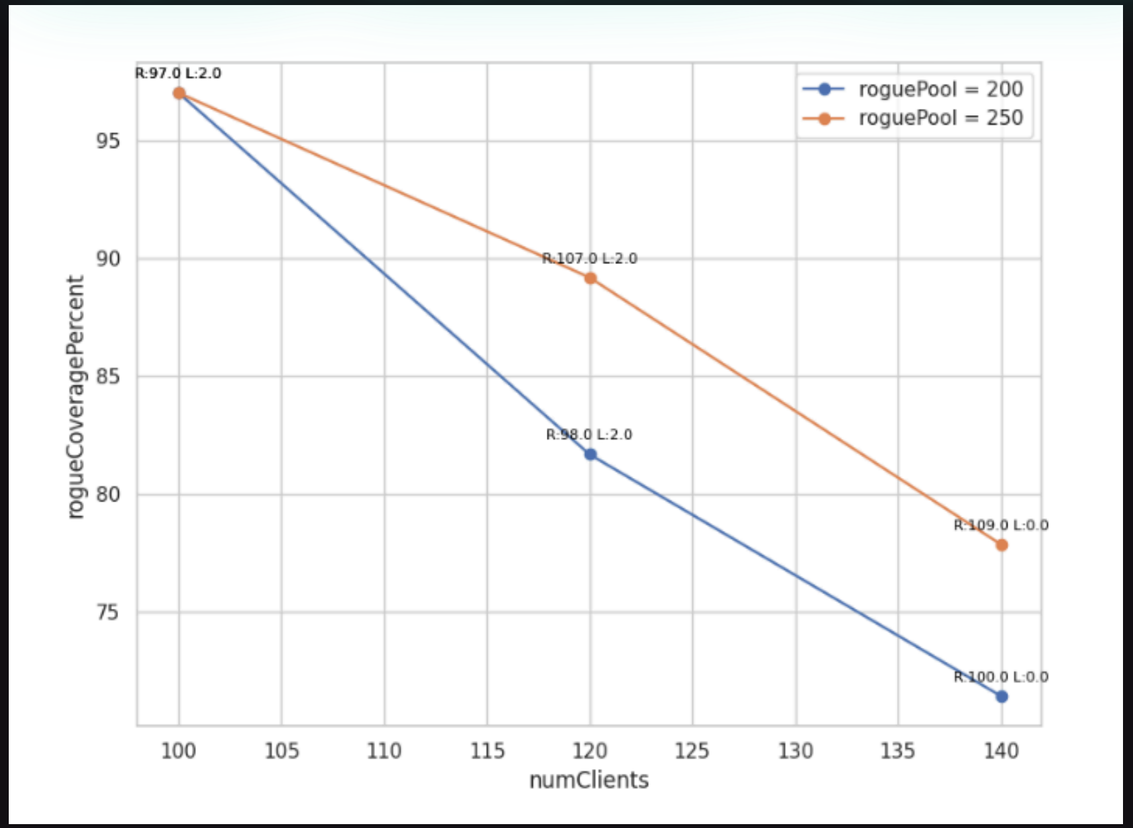
```
  for (uint32_t i = 0; i < numClients; ++i) {
    Ptr<Node> node = clients.Get(i);
    Ptr<DhcpClientApp> client = CreateObject<DhcpClientApp>();
    client->Setup(broadcastAddr, 67);

    if (i == 0) { // only the first node acts as attacker
        client->SetIsAttacker(true);
    }

    if(enableSpoofingDefense) {
        // Add legitimate DHCP server to whitelist
        client->EnableSpoofingDefense(true);
        client->AddTrustedServer(Ipv4Address("10.1.1.141")); // Legitimate server IP
    } else {
        // No spoofing defense, so add rogue server to whitelist
        client->EnableSpoofingDefense(false);
    }

    double jitter = (rand() % 100) / 1000.0; // 0–0.099s
    client->SetStartTime(Seconds(2.0 + i * 0.2 + jitter));
    client->SetStopTime(Seconds(20.0));
    node->AddApplication(client);
}
```

ATTACK IMPLEMENTATION (OUR SIMULATION)

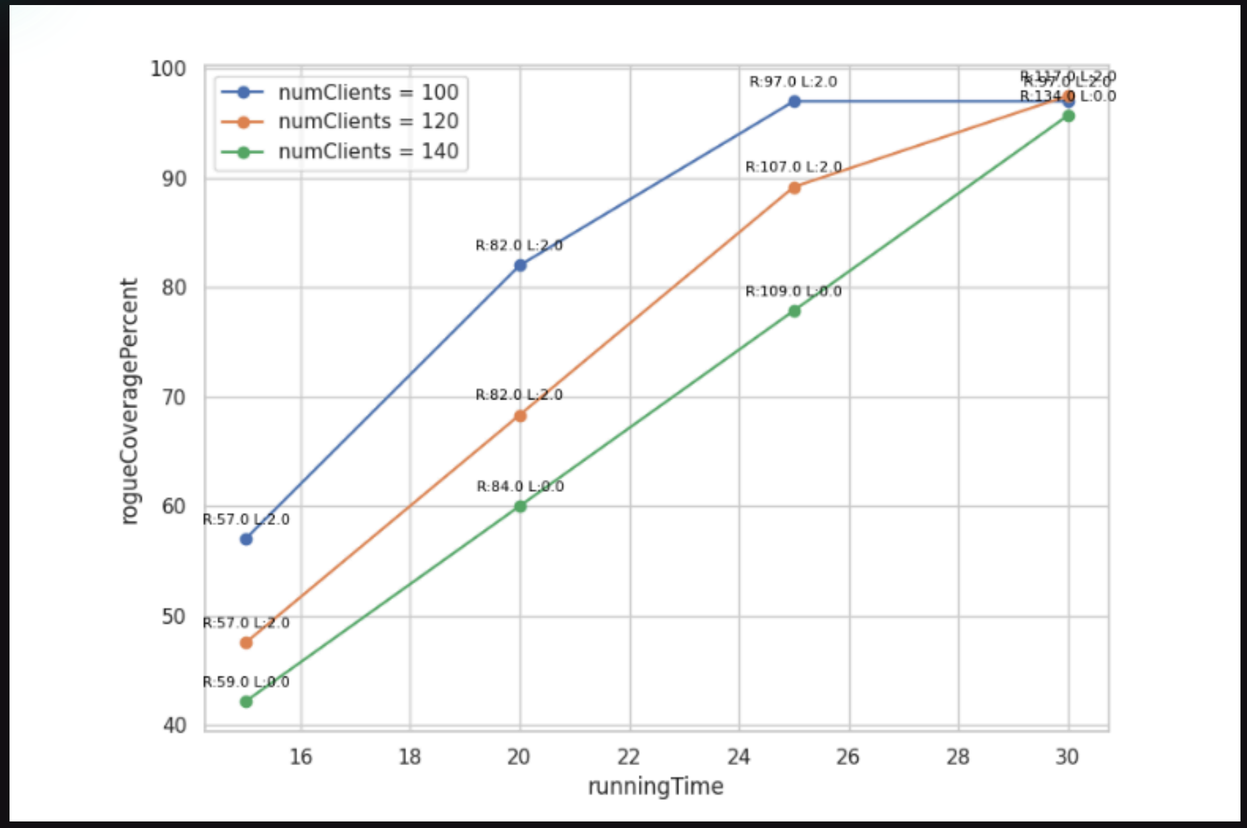




ATTACK UNDER DIFFERENT CONFIGURATIONS



Rogue Coverage % vs Number of Clients
Running time : 25 seconds



Rogue Coverage % vs Running Time
Fixed Rogue Pool Size: 250

DEFENSE MECHANISM

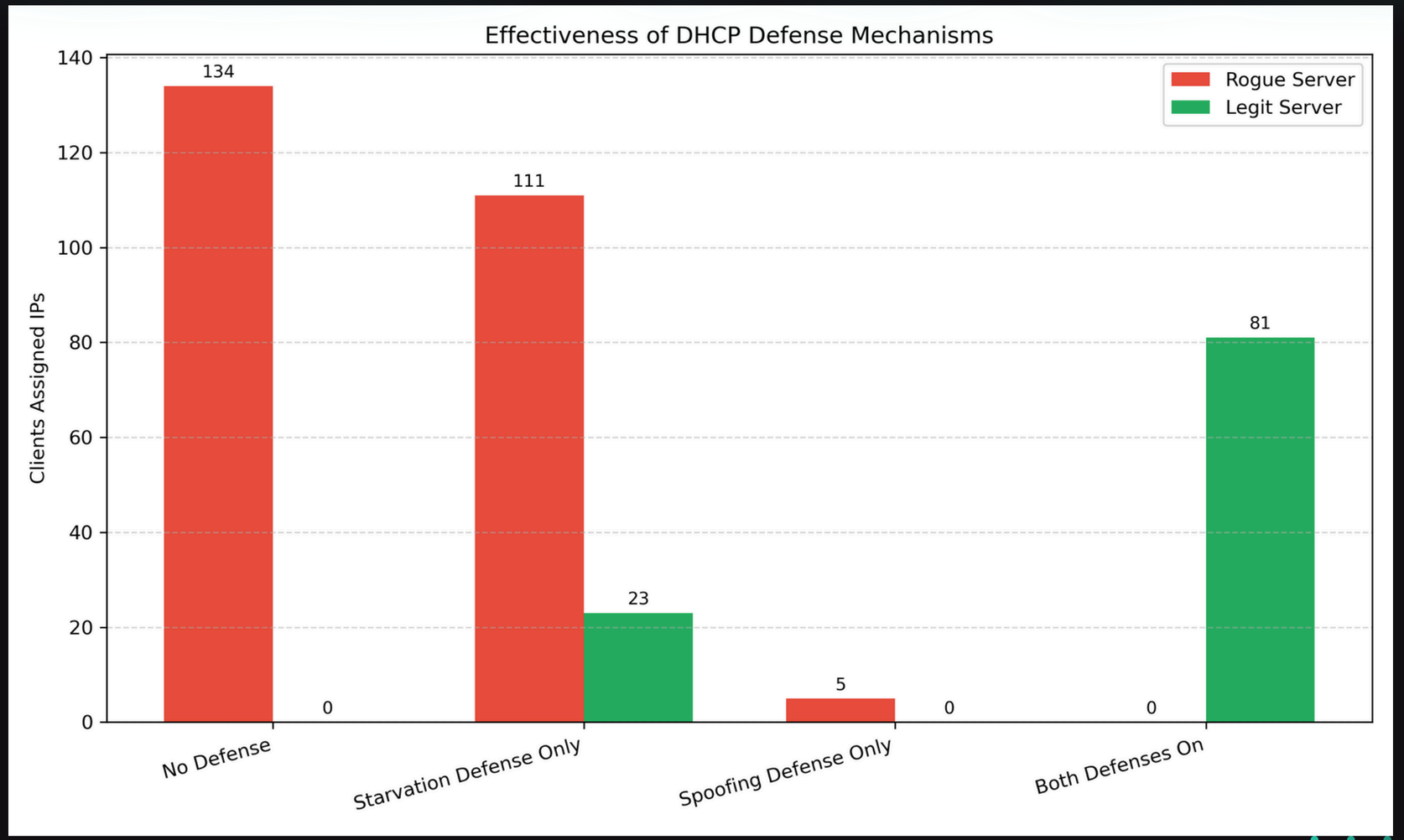
Against DHCP Starvation

We implemented rate limiting on the server side. This mechanism tracks how frequently DHCP requests arrive from clients and flags any unusual burst of activity, effectively preventing an attacker from overwhelming the IP pool with rapid, spoofed requests.

Against DHCP Spoofing

Our defense is based on IP whitelisting. Each client maintains a trusted list of DHCP server IPs. When a DHCPOFFER or DHCPACK is received, the client checks if the sender is on the whitelist. If not, the response is discarded, ensuring only legitimate configurations are accepted.

IMPROVEMENT AFTER DEFENSE



CYBERSECURITY PRESENTATION :

THANK YOU