CSE 406: Computer Security Sessional
DHCP Spoofing

Department of CSE
Bangladesh University of Engineering and Technology

July 2025

Team Members

Name: Wahid Al Azad Navid
Student ID: 2005089

Name: Md Miraj Hasan
Student ID: 2005084

1 Definition of the Attack with Topology Diagram

Overview

In this project, we implement a hybrid network attack that combines **DHCP starvation** and **DHCP spoofing**. This dual-layer strategy increases the effectiveness of the spoofing attack by exhausting the legitimate DHCP server's IP pool, forcing clients to accept configurations from the attacker's rogue DHCP server.

- **DHCP Starvation:** The attacker sends a large number of DHCP `DISCOVER` messages using spoofed MAC addresses to the legitimate server. This consumes all available IPs in the pool, making the server unable to respond to genuine clients.
- **DHCP Spoofing:** The attacker simultaneously operates a rogue DHCP server that rapidly replies to DHCP `DISCOVER` requests from victim clients. The rogue server assigns a malicious IP configuration (e.g., attacker's IP as gateway or DNS), potentially redirecting traffic.

Attack Flow Summary

1. The victim sends a DHCP `DISCOVER` request upon connecting to the network.
2. The legitimate DHCP server is either exhausted (due to starvation) or too slow to respond.
3. The rogue DHCP server sends a malicious `DHCPOFFER` packet first.
4. The victim accepts the rogue offer, using attacker-controlled DNS or gateway.

Topology Diagram

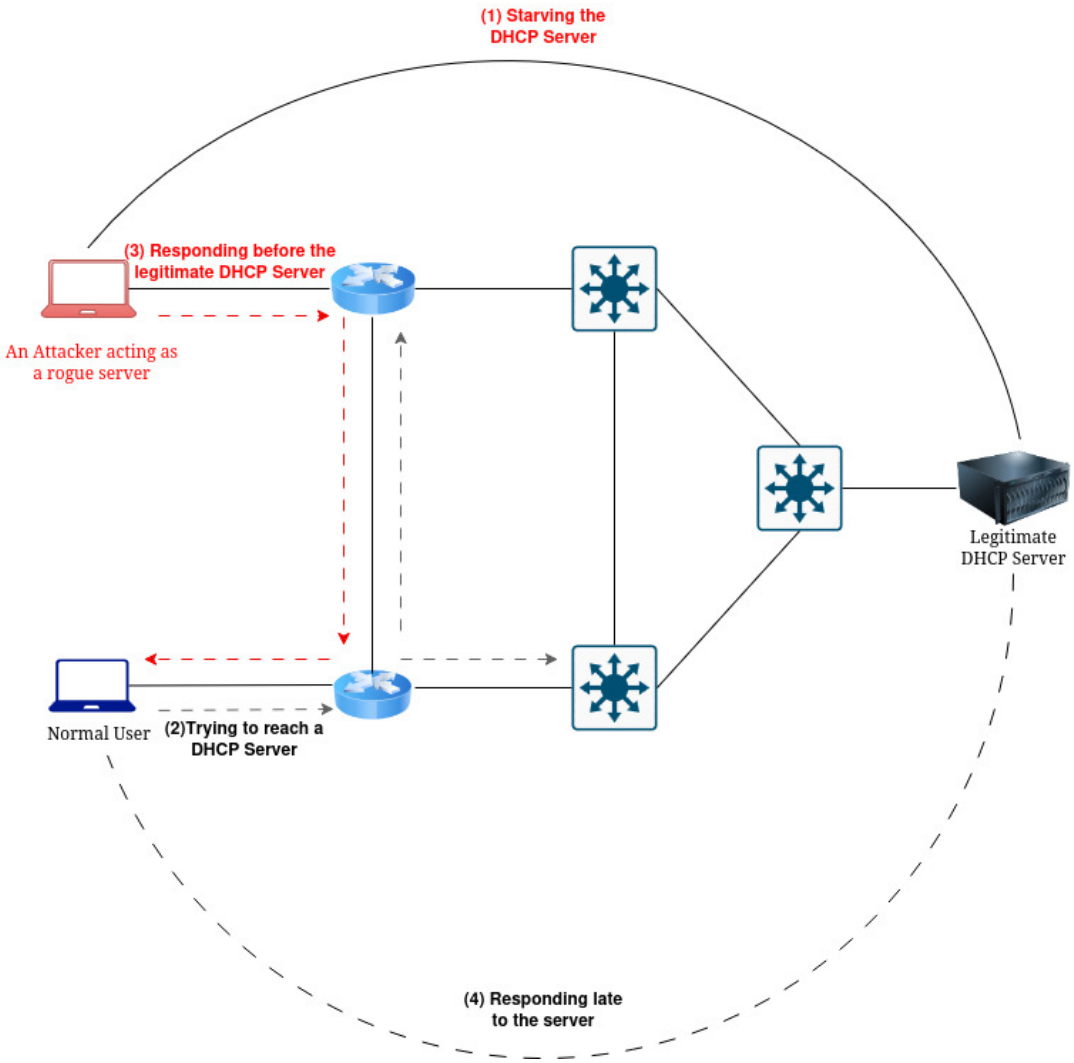
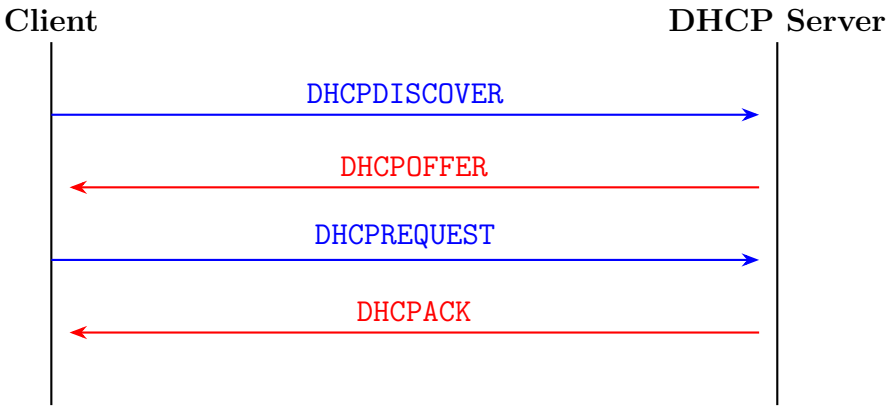


Figure 1: Combined DHCP Starvation and Spoofing Attack Topology

2 Timing Diagram of the Original Protocol and the Attack

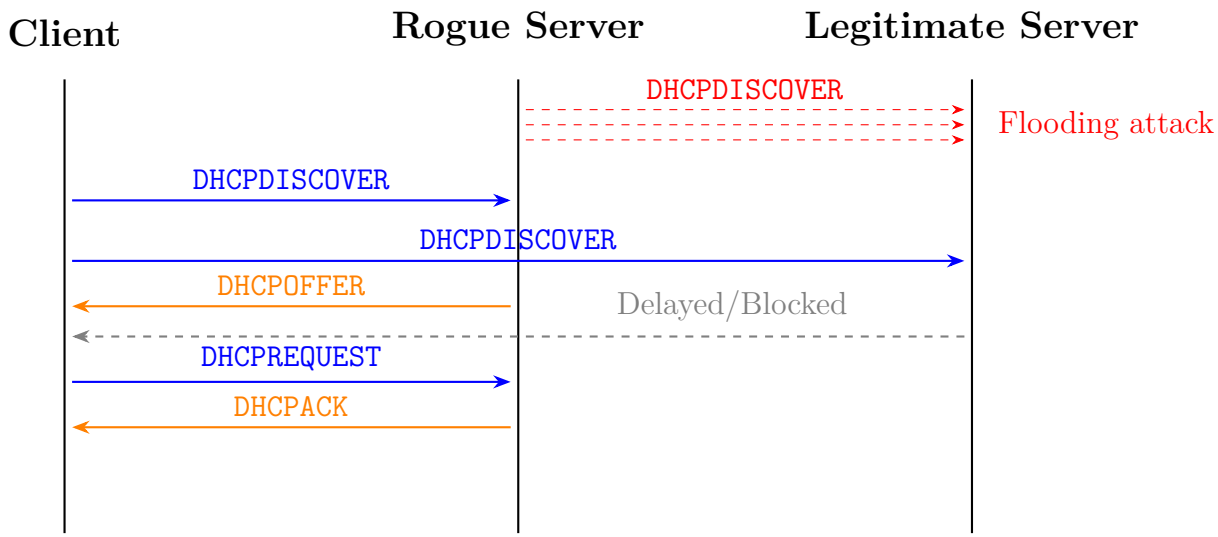
Normal DHCP Protocol Timing (4-way Handshake)

- Client broadcasts `DHCPDISCOVER`
- Server responds with `DHCPOFFER`
- Client replies with `DHCPREQUEST`
- Server confirms with `DHCPACK`



Attack Timing with Starvation and Spoofing

- Attacker floods the legitimate server with `DHCPDISCOVER` requests (starvation).
- Client sends `DHCPDISCOVER`.
- Rogue server responds immediately with `DHCPOFFER`.
- Client accepts rogue server's IP via `DHCPREQUEST`, and attacker replies with `DHCPACK`.



Attack Success: Client uses malicious configuration

3 Packet / Frame Details of the Attack

DHCP Starvation Packets

During DHCP starvation, the attacker sends a flood of `DHCPDISCOVER` packets using randomized, spoofed MAC addresses.

- **Source MAC:** Randomly generated fake addresses
- **Destination MAC:** `ff:ff:ff:ff:ff:ff` (broadcast)
- **UDP Port:** Source 68 → Dest 67
- **Bootp Flags:** Broadcast = 1
- **Effect:** Legitimate DHCP server allocates IPs to fake clients until the pool is exhausted.

Example DHCPDISCOVER Packet (Starvation Phase)

```
Frame 7: 342 bytes on wire
Ethernet II, Src: 0a:1b:2c:3d:4e:5f, Dst: ff:ff:ff:ff:ff:ff
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Transaction ID: 0x7a6b3c2d
    Seconds elapsed: 0
    Bootp flags: 0x8000 (Broadcast)
    Client MAC address: 0a:1b:2c:3d:4e:5f

    Option: (53) DHCP Message Type = DHCP Discover
    Option: (61) Client identifier = 0a:1b:2c:3d:4e:5f
    Option: (12) Host Name = "fake-client-23"
    Option: (55) Parameter Request List
        1 (Subnet Mask)
        3 (Router)
        6 (DNS)
        51 (IP Lease Time)
        54 (Server Identifier)
```

Spoofed DHCP Server Packets

The rogue server sends `DHCPOFFER` and `DHCPACK` responses before the legitimate server can respond.

- **Assigned IP Address:** Any available IP from rogue server's fake pool
- **Default Gateway Option (Option 3):** IP address of attacker machine
- **DNS Server (Option 6):** Can be attacker-controlled DNS
- **IP Address Lease Time (Option 51):** Can be short to retain control

Example DHCP Offer Packet (Spoofed)

```
Frame 52: 342 bytes on wire
Ethernet II, Src: aa:bb:cc:dd:ee:ff (Attacker), Dst: ff:ff:ff:ff:ff:ff (Broadcast)
Internet Protocol Version 4, Src: 192.168.1.100, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hops: 0
    Transaction ID: 0x12345678
    Your IP address: 192.168.1.50
    Next server IP address: 192.168.1.100
    Relay agent IP address: 0.0.0.0
    Client MAC address: 08:00:27:15:8d:44

    Option: (53) DHCP Message Type = DHCP Offer
    Option: (1) Subnet Mask = 255.255.255.0
    Option: (3) Router = 192.168.1.100              ← attacker as gateway
    Option: (6) DNS Server = 192.168.1.100          ← attacker-controlled DNS
    Option: (51) IP Address Lease Time = 300 secs   ← short lease
```

4 Justification: Why Our Design Should Work

Our attack combines two complementary techniques, DHCP starvation and DHCP spoofing, to maximize reliability and control over the victim's network configuration. The justification behind the design:

- **Starvation ensures legitimacy is disabled:** By flooding the legitimate DHCP server with `DISCOVER` requests using fake MAC addresses, we exhaust its IP pool. This guarantees that real clients will not receive valid IP leases from the real server.
- **Spoofing ensures malicious control:** Our rogue DHCP server is configured to respond faster than the legitimate server. When a real client sends a `DHCPDISCOVER`, it accepts the first response, our malicious `DHCPOFFER`.
- **Control over network routing:** The rogue DHCP server assigns a default gateway (Option 3) and DNS (Option 6) pointing to the attacker. This gives full control over traffic and resolution, enabling MITM, phishing, or surveillance attacks.
- **No authentication in DHCP:** DHCP lacks built-in authentication or trust mechanisms. Any device on the same LAN can act as a server, making this attack feasible on most networks without triggering alarms.
- **Short lease time maintains persistence:** The rogue server sets a very short IP lease time (Option 51), ensuring that the victim devices will re-request frequently and continue to accept malicious responses.

Therefore, the design is robust, stealthy and highly effective in local network environments, especially on unprotected switches or public Wi-Fi setups.