Assignment 2: (Part 2) Exploring Diffie-Hellman Key Exchange Vulnerabilities

Nikhil Gupta (2024JCS2611), Abhishek Gupta (2024JCS2047)

January 27, 2025

1 Introduction

In this part of the assignment, we explored the vulnerabilities inherent in a vulnerable Diffie-Hellman key exchange. By simulating a Man-in-the-Middle (MITM) attack using ARP poisoning, we could intercept the victim's communication and eventually deduce the server's private key.

The setup consisted of three machines:

- **Victim:** A macOS machine.
- **Server:** An Oracle server (IP: 10.208.66.147:5555).
- **MITM:** Kali Linux VM (acting as the attacker).

The Mac and Kali Linux machines were connected using a bridge connection to ensure they were on the same local network.

2 Vulnerability in Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange protocol is widely used to exchange cryptographic keys securely over a public channel. However, the security of this protocol depends on the strength of the parameters used in the exchange, specifically the prime number `P` and the generator `G`. An attacker can exploit these parameters and deduce the private key through brute force or other attacks if these parameters are weak.

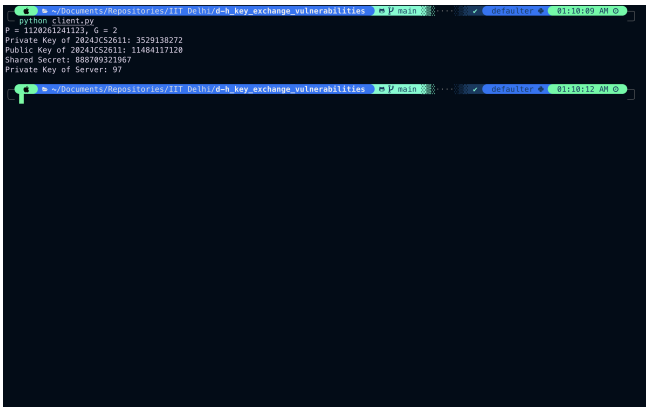

Figure 1: Private Keys (Victim and Server) Without MITM (2024JCS2611).

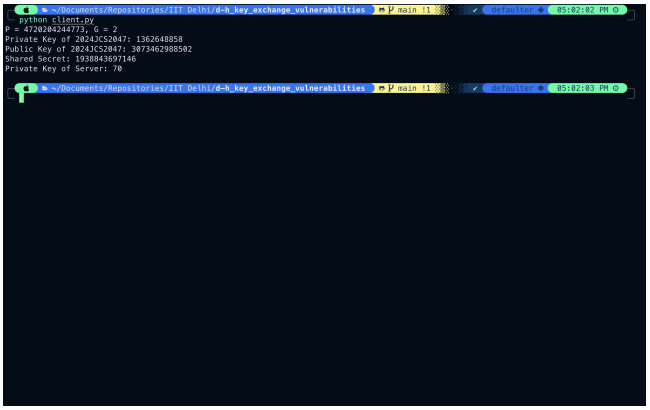

Figure 2: Private Keys (Victim and Server) Without MITM (2024JCS2047).

This demonstrates the critical vulnerability in Diffie-Hellman when weak parameters are used. The ability to deduce the private key undermines the protocol's security and makes it vulnerable to attacks.

3 Setting Up and Executing the MITM Attack

3.1 Step 1: Installation of Ettercap on Kali

To initiate the attack, we first installed Ettercap on Kali Linux. Ettercap is a popular tool for network attacks, particularly Man-in-the-Middle (MITM) attacks. The installation was performed using the following command:

```
1 sudo apt-get install ettercap-common
```

Furthermore, we used the `ifconfig` command to identify IP addresses and network interfaces on the Kali machine to ensure it was properly connected to the network.

3.2 Step 2: Configuring Firewall Rules

We configured the firewall in Kali Linux using `iptables` to manipulate network traffic between the victim and the server. The commands used are:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 5555
    -d 10.208.66.147 -j DNAT --to-destination
    <MITM_IP>:<MITM_PORT>
sudo iptables -t nat -A POSTROUTING -p tcp --dport 5555
    -j MASQUERADE
```

Explanation:

- The first rule directs any incoming traffic to port 5555 on the server IP (10.208.66.147) to the Kali Linux machine's IP address and port, effectively redirecting traffic.
- The second rule allows the Kali machine to masquerade as the server, which means that it will send the responses to the victim as if it were the actual server. This ensures that the communication between the victim and the server is intercepted by the MITM (Kali machine).

3.3 Step 3: Performing ARP poisoning

Before executing the ARP poisoning attack, we inspected the ARP table on the victim machine using the `arp -a` command. This provided a baseline view of the current mappings between IP addresses and MAC addresses. The ARP table initially showed that the gateway IP address was correctly assigned to the router's legitimate MAC address.

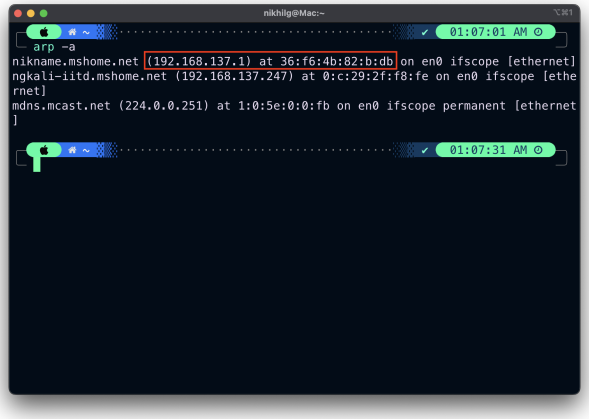

Figure 3: ARP Table before poisoning on Victim: Gateway mapped to the router's MAC address.

To carry out ARP poisoning, we used Ettercap with the following command:

```
sudo ettercap -T -i eth0 -M arp:remote /<victim_IP>//
    /<router_IP>//
```

Explanation:

- `-T`: Text-based interface mode (run in terminal).
- `-i eth0`: Select the interface to use (in this case, eth0).
- `-M arp:remote`: ARP poisoning mode (targeting both the victim and the router).
- `/<victim_IP>//`: The IP address of the victim.
- `/<router_IP>//`: The IP address of the router or gateway.

After executing this command, Ettercap sends malicious ARP packets to both the victim and the router, tricking them into believing that the Kali machine is the router. This enables the Kali machine to intercept and manipulate the victim and server communication.

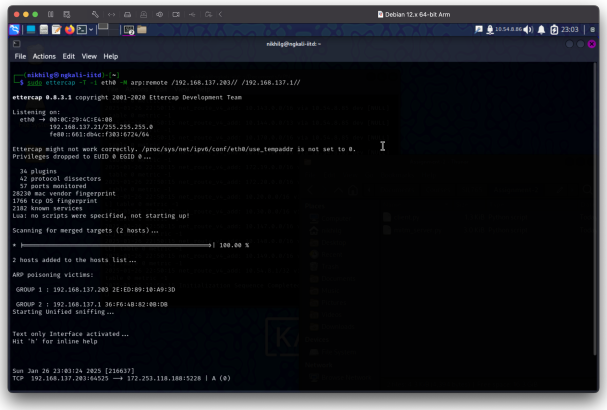

Figure 4: Starting Ettercap on Kali Linux

We re-checked the ARP table on the victim machine using the `arp -a` command. The table now shows that the gateway IP address was mapped to the MAC address of the Kali machine instead of the router, confirming that the ARP poisoning attack was successful.

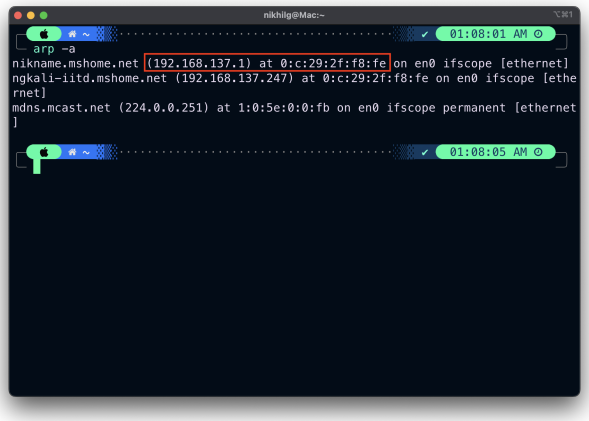

Figure 5: ARP Table after poisoning on Victim: Gateway mapped to Kali's MAC address.

By successfully altering the ARP table, the Kali machine was able to intercept communication between the victim and the router, effectively positioning itself as a man-in-the-middle.

3.4 MITM Attack on Diffie-Hellman Key Exchange

Since all traffic from the client is now directed towards the attacker, it acts as an intermediary between the victim and the server, manipulating the communication to break the system:

- The attacker intercepts the credentials meant for the server from the victim.
- Fake Diffie-Hellman parameters (P and G) are sent to the victim, initiating the key exchange with the attacker.
- The attacker forwards the credentials to the server, receives the correct P and G, and performs the key exchange with the server.
- The attacker calculates the shared secret with the victim and the server and exploits the small prime vulnerability in Diffie-Hellman to derive the server's private key.
- This allows the attacker to decrypt and manipulate the communication, gaining complete control over the interaction between the victim and the server.

The following screenshots show the exchanged values between the attacker, victim, and server during the MITM attack:

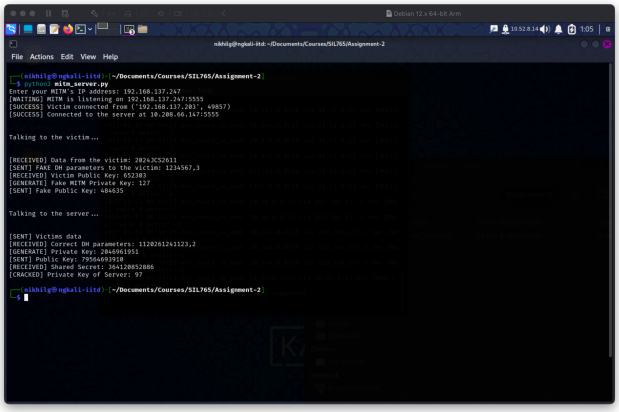

Figure 6: Message exchange between the attacker, server, and victim during the MITM attack.

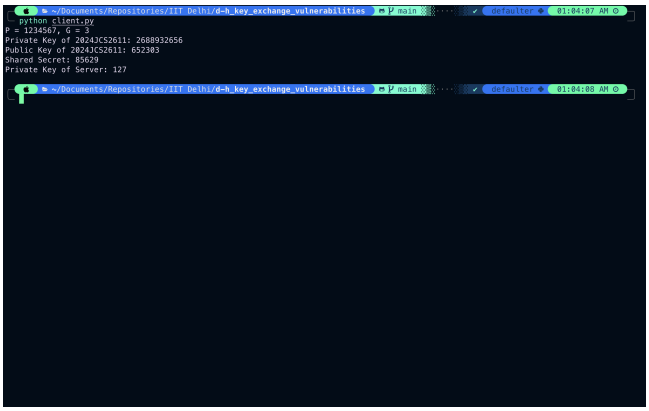

Figure 7: Values received by the victim, showing manipulated Diffie-Hellman parameters.

4 Results

The Man-in-the-Middle (MITM) attack successfully allowed the attacker to intercept the Diffie-Hellman key exchange and deduce the server's private keys. By manipulating the Diffie-Hellman parameters, the attacker could calculate the shared secret and exploit the weak prime number vulnerabilities.

Server private keys deduced:

- Server private key for 2024JCS2611: `97`
- Server private key for 2024JCS2047: `70`

This highlights the vulnerability in Diffie-Hellman when weak parameters are used. The attacker, positioned as a man-in-the-middle, could decrypt and manipulate communication between the victim and the server, revealing the private keys and undermining the security of the entire system.

Insights into the vulnerabilities:

- Weak Diffie-Hellman parameters (P and G) make key exchange susceptible to brute-force attacks.
- ARP poisoning in the network allows attackers to position themselves between the victim and the server, compromising the entire communication.

5 Challenges Faced

During the execution of the MITM attack, several challenges were encountered:

- **DAI (Dynamic ARP Inspection)**: One of the challenges faced was the presence of DAI (Dynamic ARP Inspection) on the router. This security

feature prevented the direct manipulation of the ARP table. To bypass this, we used a mobile hotspot as the network environment, effectively allowing us to conduct the attack without interference from the network's security settings.

- **Network Configuration Issues**: Setting up the network on Kali Linux (to ensure proper communication between the victim and the attacker) involved some trial and error, especially regarding interface configuration and packet forwarding.

Despite these challenges, the overall execution was successful, and the vulnerabilities in the Diffie-Hellman key exchange were demonstrated.