Bangladesh University of Engineering and Technology

Computer Security Sessional

CSE 406

DHCP Spoofing

Prepared By:
Tawhidul Hasan Bhuiyan, 1505008

September 9, 2019

1 Introduction

Dynamic Host Configuration Protocol(DHCP) dynamically assigns IP addresses. The four step process is discover, offer, request, and acknowledgment. When a host joins the network it doesn't have an IP address. So it begins the DHCP process by broadcasting a discover packet to the network. Because DHCP doesn't have a native authentication process, the client can be a victim of DHCP spoofing.

2 Definition

DHCP spoofing is a kind of attack where attacker can gain access to network traffic by spoofing responses that would be sent by a valid DHCP server.

When a client tries to get an IP address through DHCP, it may get an offer from the real DHCP server, but in addition it may also get an offer from a rogue DHCP server and accept that offer. When a client requests an IP address from a DHCP server, the client has no reassurance that the server is legitimate. In addition, the server has no way of knowing that the client requesting the address is a legitimate client on the network.

3 Topology diagram

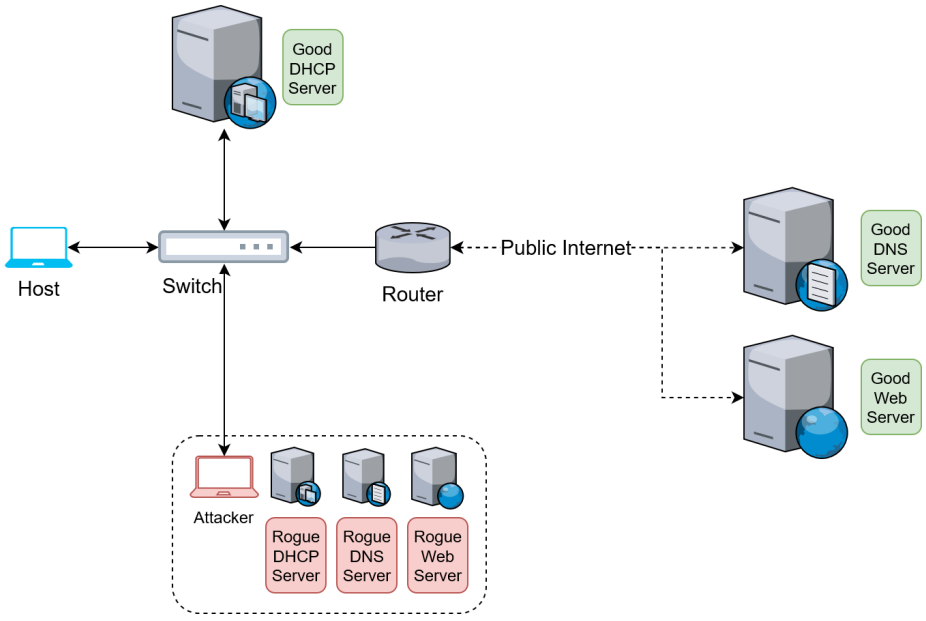

Figure 1: Topology Diagram of DHCP Spoofing Attack

When a *Host* joins a network, it normally gets an IP address from a *Good DHCP Server* in that network. Additionally, the DHCP Server will provide address of one or more *Good DNS Servers*, so that the *Host* can connect to *Correct Web Server* through the *public internet*.

An *Attacker* that is in the same network of *Host* can try to spoof it by responding to its DHCP discover message by running a *Rogue DHCP Server* in his machine. Also, it sends the IP of a *Rogue DNS Server* run by the *Attacker* as only DNS server. So, when the *Host* tries to connect to the internet, it uses that *Rogue DNS Server*. The *Rogue DNS Server* might sen the *Host* to a *Rogue Web Server*. So, that can be use for further spoofing.

Figure 1 is a demonstration of the topology.

4 Timing diagrams

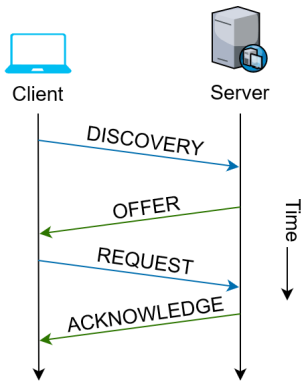

Figure 2: Original Timing Diagram of DHCP Protocol

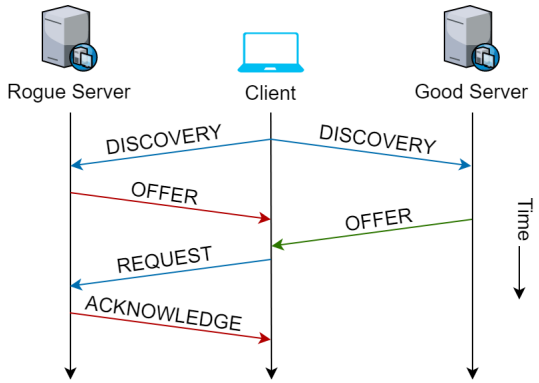

Figure 3: Timing diagram for DHCP Spoofing

DHCP protocol normally goes as follows:

A client connects to the network and tries to get an IP address that starts with a *DHCP DISCOVER* packet. Then each DHCP server in the network responds with a *DHCP OFFER* packet. After that the client responds to one of the DHCP servers with a *DHCP REQUEST*. Finally, the server sends a *DHCP ACKNOWLEDGE* as a respond to the *DHCP REQUEST* packet of client.

We will do the attack by mimicking a server to the client. But our server responds to the client with IP address of a **Rogue DNS server** as DNS server address. As the client cannot differentiate this from a good DNS server, he might accept it. Now, when the client tries to reach a web server, our **Rogue DNS server** sends it to a **Rogue Web server**.

5 Strategies

To make a DHCP Spoofing attack, some steps needs to be followed.

- Set up a **Rogue DNS Server** in the network.
- Also, set up a **Rogue Web Server** for testing.
- Start sniffing for any *DHCP DISCOVERY* packet.
- When a *DHCP DISCOVERY* packet is found, reply to it with a spoofing IP address to use, own IP address as server address and IP address of the **Rogue DNS Server** as DNS server.
- The reply is a *DHCP OFFER* message. It should be sent within a reasonable time frame to be successful.
- After *DHCP OFFER* reply is sent, wait for some time. If the client responds with a *DHCP REQUEST* message, send it a *DHCP ACKNOWLEDGE* message.
- This settings make it work like a **Rogue DHCP Server**. So, we need to handle available IPs accordingly.

6 Protocol Frame Details and Modifications

We are only concrened with **Your IP address**, **Server IP address** and **DNS IP address** in **DNS configuration option**. We will fill

- **Your IP address** with an IP from a given range.
- **Server IP address** with IP address of our rogue DHCP server.
- **DNS IP address** with IP address of our rogue DNS server.

(a) Structure of a DHCP Frame

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Opcode								Hardware type								Hardware address length								Hop count							
Transaction ID																															
Number of seconds																Flags															
Client IP address																															
Your IP address																															
Server IP address																															
Gateway IP address																															
Client hardware address (16 Bytes)																															
Server host name (64 Bytes)																															
Boot filename (128 Bytes)																															
Options (Variable length)																															

(b) Structure of a DHCP Header

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Code (6 for DNS)								Length (option data size)								IP address[] (One or more IP address of DNS Server)															

(c) Structure of DNS configuration option of DHCP Header

Figure 4: DHCP Protocol

(a) Structure of a DHCP Frame with Spoofing

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Opcode								Hardware type								Hardware address length								Hop count							
Transaction ID																															
Number of seconds																Flags															
Client IP address																															
Your IP address (from Rogue Server)																															
Server IP address (of Rogue Server)																															
Gateway IP address																															
Client hardware address (16 Bytes)																															
Server host name (64 Bytes)																															
Boot filename (128 Bytes)																															
Options (Variable length)																															

(b) Structure of a DHCP Header for Spoofing

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Code (6 for DNS)								Length (option data size)								IP address[] (IP address of Rogue DNS Server)															

(c) Structure of DNS configuration option of DHCP Header for Spoofing

Figure 5: DHCP Protocol with Spoofing

7 Prerequisites

Mininet A virtual network emulator. A network of multiple PCs, switches and routers is needed to implement the topology that runs the attack. Mininet creates a realistic virtual network, running real kernel, switch and application code, on a single machine (VM, cloud or native), in seconds, with a single command. So, it is safe to say the result found here works in a real network.

Dnsmasq A tool used to run a DNS server in the attacker's PC.

Busybox A tool used to run the good DHCP server.

8 Network Creation

First we use mininet to create a virtual network with the topology shown in figure 1. The topology is implemented in a python file `topo.py`. The good DHCP server is implemented using **Busybox** tool. A NAT is used as the router connecting to Internet.

Run the topology file with following command:

```
$ sudo python topo.py
```

After running the program it opens Xterm terminal for each node. Here,

h1 is the victim

evil is the attacker

dhcp is the good DHCP server

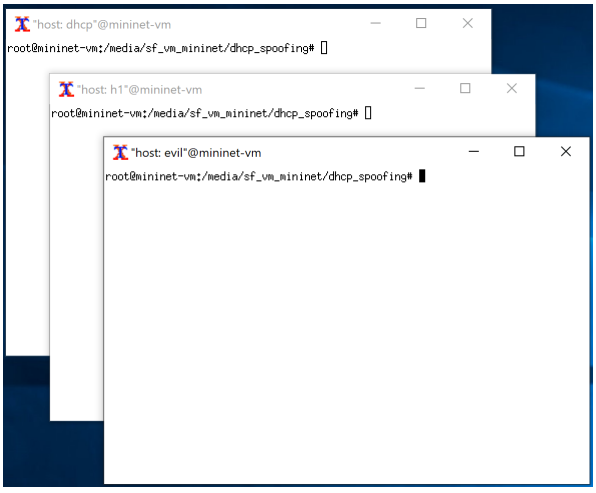

Figure 6: Xterm Terminals Created

9 DHCP spoofing with Rogue DHCP server

A rogue DHCP server is implemented in a python file `dhcp_server.py`. It sniffs the network for DHCP DISCOVER packet. As soon as getting the packet it responses with a DHCP OFFER packet before the good DHCP server. So, the victim responses with DHCP REQUEST (hence, getting spoofed) and the server sends back a DHCP ACKNOWLEDGE packet. The victim keeps sending DHCP REQUESTs to the server as the lease time gets near to end. The server sends DHCP ACKNOWLEDGEs back for each of them.

Run the rogue DHCP server with following command:

```
$ python dhcp_server.py <interface> <ip_address> [-m] [-e]
```

Here, interface is the interface connceted to network and ip_address is the IP address of that interface. The -m, -e flags are discussed in the next section.

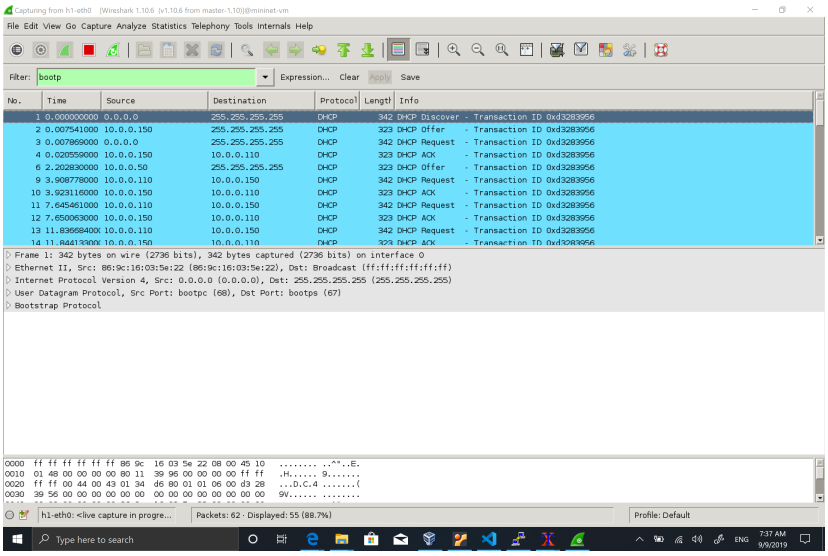

Figure 7: Victim receiving response of rogue DHCP server early and getting spoofed

10 Attacks

DHCP spoofing allows two different type of attacks.

Masquerade DHCP sever will send IP of a rogue DNS server as name server. This server will send the victim to a rogue website while he belives it to be the real.

Evesdroping DHCP server will send IP of a rogue PC as router. So, any packet the victim sends, goes through that PC.

The tool implemented here can do both(-m flag for Masquerade, -e flag for Evesdroping). Each one needs to be rundown seperately.

A bash file `evil.sh` is used to run commands and other files to perform the attacks with ease. Command to run this is:

```
$ ./evil.sh <interface> [-m] [-e]
```

10.1 Masquerade

Masquerade attack is performed through sending IP of a rogue DNS server as name server. That DNS server can send the victim to a rogue Web server that the victim might believe to be a legitimate good server. Here, the evil PC runs a rogue DNS server. So, DHCP sends IP of evils PC as name server. The DNS server is implemented using Dnsmasq tool. The tool looks in a file `/etc/dnsmasq.conf` for resolving names where we can store a fake mapping from names to IPs. A bash file `dns_server.sh` implements these.

Run the following code to create rogue DNS server:

```
$ sudo ./dns_server.sh <ip_address>
```

Next, the evil PC runs a rogue Web server(`web_server.py`). Now, the IP of evil PC can be used for mapping from any domain name.

Run the following code to create rogue Web server:

```
$ sudo python web_server.py <ip_address>
```

Running `evil.sh` with `-m` flag does the whole thing.

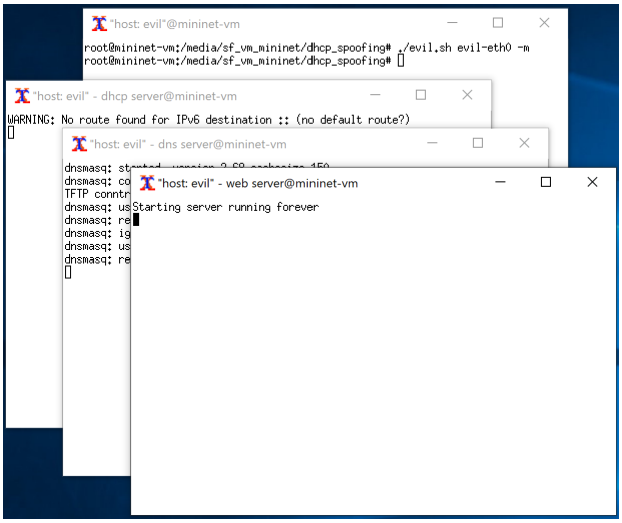

Figure 8: Starting Masquerade Attack

Without masquerade attack, the victim goes to the right `www.amazon.com` website.

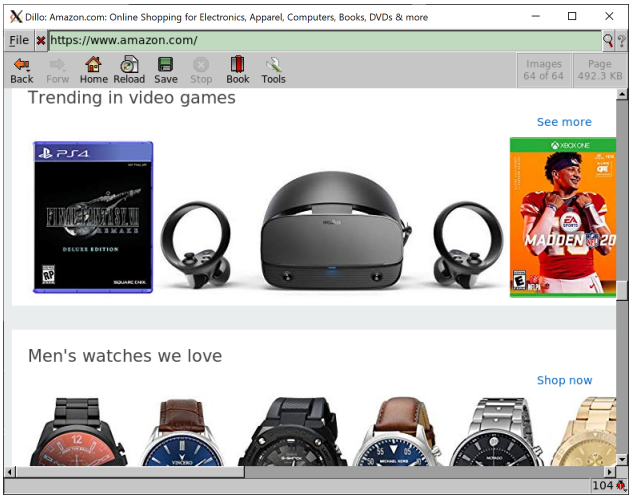

Figure 9: Victim going to right Website when not Masqerading

But with masquerade attack, when serching for `www.amazon.com`, the victim goes to the rogue server the attacker created.

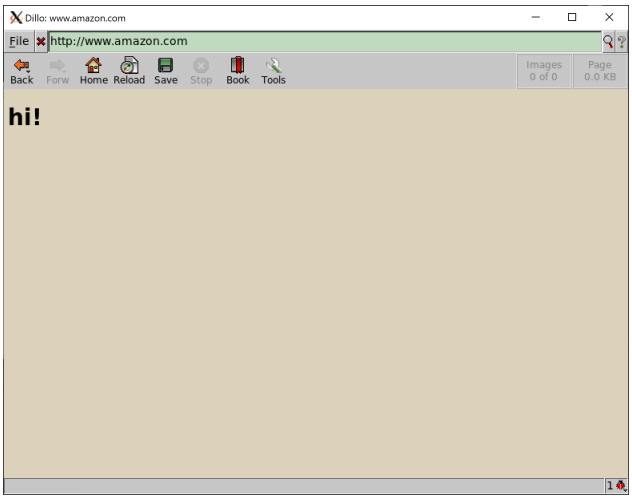

Figure 10: Victim going to rogue Website when Masqerading is enabled

10.2 Eavesdropping

If in the DHCP packets the evil PC's IP is used as the router address all packets going out of victim PC comes to the evil PC. Then the evil PC can sniff all those packets. The sniffer is implemented in a python file `sniff.py`. IP Forwarding can be enabled in the evil PC so that the attack does not turn out to be a DoS.

Run sniffer:

```
$ python sniff.py <interface>
```

Enable IP forwarding:

```
$ echo 'net.ipv4.ip_forward = 1' > /etc/sysctl.conf
$ sysctl -p /etc/sysctl.conf
```

Running `evil.sh` with `-e` flag does the whole thing.

This enables to eavesdrop what the victim sends and receives through the network.

Here, the victim goes to `http://biis.buet.ac.bd`. Everything looks to be normal.

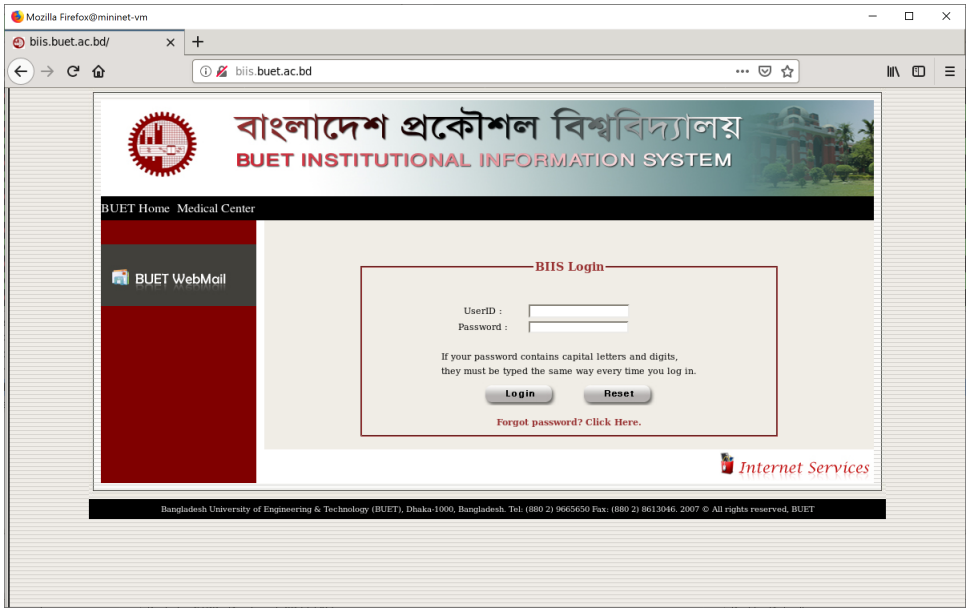

Figure 11: Victim sees nothing different when attacker eavesdrops

But, the attacker can listen everything.

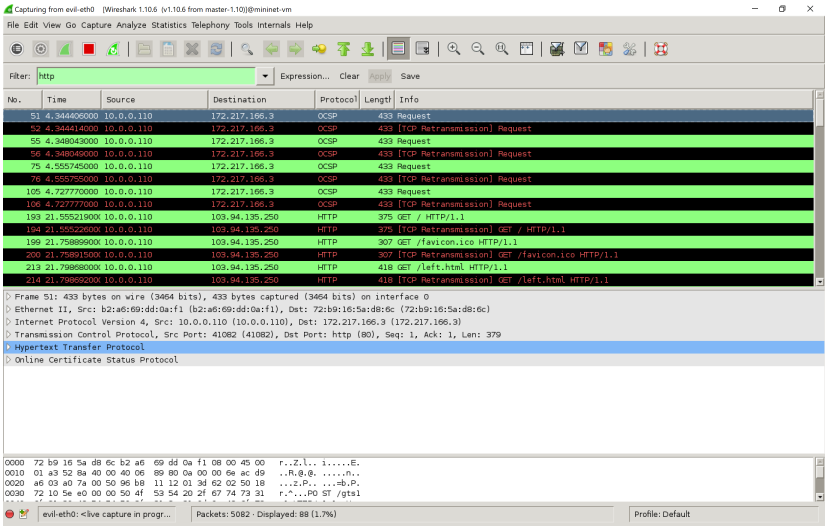

Figure 12: Attacker listens everything victim does through network

11 Countermeasures

DHCP Snooping is the best mechanism to avoid this kind of attacks. The idea of this feature is to differentiate between two types of ports in a switched environment: a reliable port side (trusted port) and, secondly, untrusted ports (untrusted host). The first have no restrictions on the type of DHCP messages that can receive, as they will be those connected to a controlled environment (in this case the server / DHCP servers). However, the latter can only send those packets that under normal conditions a client needs to send to get its DHCP configuration (DHCP Discover, DHCP Request, DHCP Release). Therefore, the untrusted ports will correspond to those ports connected to end users, and in the case that one of those ports receives a spoofed DHCP offer packet or a DHCP ack (as in our case), it will be blocked.