

2015 Advanced Computer Networks

Homework 4

Motivation

To learn how to receive, build and send Ethernet packets. You will know how ARP works by this homework.

Specification

First part:

Use the main.c which is included in attachment to make an ARP packet capture program. In order to make program in a common format, please refer to “arp.h” when you do this homework. You can consult your book on page 170 for ARP packet format. Besides, you should implement the filter in this part as well.

Second part:

Send an ARP request and receive the ARP reply to analyze the packet and find the MAC address of the specific IP. Generally, we usually find the MAC address by cleaning the ARP cache, pinging the IP, capturing the packets with something like Wireshark and analyze the packet by yourself. In this part, you should do the same thing by programming.

Third part:

Make an ARP daemon, it can reply a MAC address when it receive specific IP address.

Request

First part:

Show usage when the command with insufficient or excessive parameters. You need to validate IP and MAC address format. You also need to show error message when the program isn't executed by superuser privileges.

```
wolf@wolf:~/tcpip_hw4$ ./arp
ERROR: You must be root to use this tool!
wolf@wolf:~/tcpip_hw4$ sudo ./arp -h
[sudo] password for wolf:
Format:
1) ./arp -l -a
2) ./arp -l <filter_ip_address>
3) ./arp -q <query_ip_address>
4) ./arp <fake_mac_address> <target_ip_address>
wolf@wolf:~/tcpip_hw4$
```

Use “./arp -l -a” command to show all of the ARP packets.

```
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l -a
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.169.197?      Tell 140.117.169.190
Get ARP packet - Who has 140.117.171.183?      Tell 140.117.171.254
Get ARP packet - Who has 140.117.169.99?       Tell 140.117.169.82
Get ARP packet - Who has 140.117.175.51?       Tell 140.117.175.254
Get ARP packet - Who has 192.168.0.1?          Tell 192.168.0.1
Get ARP packet - Who has 140.117.11.1?         Tell 140.117.176.42
Get ARP packet - Who has 140.117.168.117?      Tell 140.117.168.254
Get ARP packet - Who has 140.117.174.102?      Tell 140.117.174.254
Get ARP packet - Who has 140.117.169.89?       Tell 140.117.169.254
Get ARP packet - Who has 140.117.172.166?      Tell 140.117.172.254
Get ARP packet - Who has 140.117.170.154?      Tell 140.117.170.254
Get ARP packet - Who has 140.117.168.171?      Tell 140.117.168.175
Get ARP packet - Who has 140.117.174.106?      Tell 140.117.174.254
Get ARP packet - Who has 140.117.171.197?      Tell 140.117.171.254
Get ARP packet - Who has 140.117.174.5?       Tell 140.117.174.254
Get ARP packet - Who has 140.117.176.108?      Tell 140.117.176.254
Get ARP packet - Who has 140.117.170.106?      Tell 140.117.170.254
Get ARP packet - Who has 140.117.172.14?       Tell 140.117.172.254
Get ARP packet - Who has 140.117.168.171?      Tell 140.117.168.180
^Cwolf@wolf:~/tcpip_hw4$
```

Use “./arp -l <ip address>” command to implement the filter work. Thus, it should show specific ARP packets.

```
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l 140.117.171.216
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.216?          Tell 140.117.171.215
^Cwolf@wolf:~/tcpip_hw4$
```

Second part:

Fill an ARP request packet and send it by broadcast to query the MAC address of the specific IP address.

```
wolf@wolf:~/tcpip_hw4$ sudo ./arp -q 140.117.171.180
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.180 is 44:8a:5b:ba:33:5f
wolf@wolf:~/tcpip_hw4$
```

If the IP is offline, you might not find its MAC address, so you have to check the connection before your homework executed. You can use **ifconfig** on linux or **ipconfig /all** on Windows to check the MAC address of the computer.

Also, you have to install the Wireshark to reconfirm your packets sent and received.

If you obey the order of the homework part, you can use the filter ARP list of the part 1 to detect whether the request packet which part 2 sends is sent successfully or not.

```
wolf@wolf: ~/tcpip_hw4
wolf@wolf:~/tcpip_hw4$ sudo ./arp -q 140.117.171.180
[ ARP sniffer and spoof program ]
### ARP query mode ###
MAC address of 140.117.171.180 is 44 8a:5b:ba:33:5f
wolf@wolf:~/tcpip_hw4$
```

2. query the mac address of specific IP
(send ARP request packet)

```
wolf@wolf: ~/tcpip_hw4
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l 140.117.171.180
[ ARP sniffer and spoof program ]
### ARP sniffer mode ###
Get ARP packet - Who has 140.117.171.180? Tell 140.117.171.216
^Cwolf@wolf:~/tcpip_hw4$
```

1. listen the packets

3. get the ARP packet and list

Third part: You CANNOT use example IP when you test your homework.

When program receive an ARP request for 140.117.171.216 (this is example IP), send a 00:11:22:33:44:55 reply.

The image displays a Wireshark packet capture and a terminal window. The Wireshark interface shows a filter for `arp.opcode==2`. Two ARP packets are visible:

No.	Time	Source	Destination	Protocol	Length	Info
1565	10.705073000	AsustekC_de:3b:e7	HewlettP_4f:6b:5c	ARP	42	140.117.171.216 is at 00:26:18:de:3b:e7 <i>This is the real MAC address.</i>
1566	10.705176000	AsustekC_de:3b:e7	HewlettP_4f:6b:5c	ARP	42	140.117.171.216 is at 00:11:22:33:44:55 <i>This is the fake MAC address we make.</i>

Below the packet list, the details pane shows the Ethernet II header and the ARP (0x0806) section. A yellow highlight indicates a duplicate IP address detected for 140.117.171.216.

The terminal window shows the execution of the `arp` command with the fake MAC address:

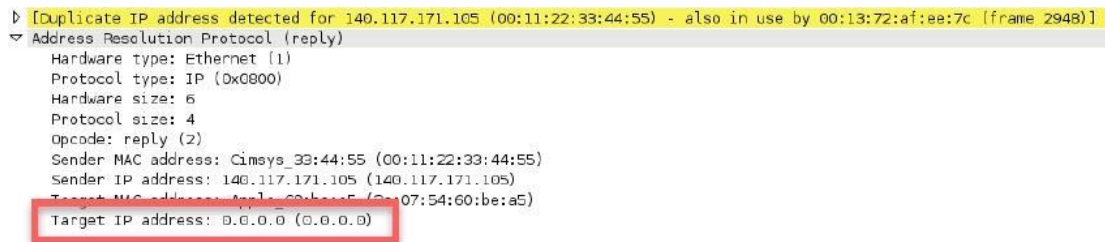
```
wolf@wolf: ~/tcpip_hw4
wolf@wolf:~/tcpip_hw4$ sudo ./arp 00:11:22:33:44:55 140.117.171.216
[ ARP sniffer and spoof program ]
### ARP spoof mode ###
Get ARP packet - Who has 140.117.171.216? Tell 140.117.171.215
Sent ARP Reply: 140.117.171.216 is 00:11:22:33:44:55
send successful
^Cwolf@wolf:~/tcpip_hw4$
```

You can use another computer and ping 140.117.171.216 (this is example IP), it will send an ARP request packet. Your program will send an ARP reply in the same time. (If it's not work, you can clear your ARP cache first.)

You can use Wireshark tool to capture the packet you made. There have two ARP packets, one is from true target (00:13:72:af:ee:7c), another is fake (00:11:22:33:44:55).

Notice

1. In the second and third part, TAs will use Wireshark to verify the ARP reply you made, so make sure your ARP format is as same as the above picture.
2. The packets you send should fully follow the ARP packet standard, every field should be correct and not be empty.



```

▶ [Duplicate IP address detected for 140.117.171.105 (00:11:22:33:44:55) - also in use by 00:13:72:af:ee:7c [frame 2948]]
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: Cimsys_39:44:55 (00:11:22:33:44:55)
  Sender IP address: 140.117.171.105 (140.117.171.105)
  Target MAC address: Apple_Gigahertz5 (08:07:54:60:be:a5)
  Target IP address: 0.0.0.0 (0.0.0.0)

```

The above example is not correct, because of missing target IP address.

3. **ARP spoofing is illegal!** Do not attack device of others!
4. You should build an ARP spoofing target **by yourself**. For the above example, spoofing target is 140.117.171.216.
5. This homework require superuser privileges, so you should build your own Ubuntu Linux 14.04 by yourself, we will not provide server's superuser privileges.
6. In order to make program in a common format, please make your input as follow:

```

./arp -l -a
./arp -l <filter_ip_address>
./arp -q <query_ip_address >
./arp <fake_mac_address> <target_ip_address>

```

7. You should prevent silly input from executing the program.

```
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l 140.256.150.222 // 256 exceeds the limitation of IP, so it is false.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp -a -l // It is not allowed to change the order of the parameters.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l -l // There is not a right input.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l 140.117.170 // IP has 4 digits and 3 dots.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp -l 140.117.170.a // As above.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp -h 140.117.170.210 // -h is a wrong parameter with IP address.
Error Pattern
wolf@wolf:~/tcpip_hw4$ sudo ./arp gg:11:22:33:44:55 140.117.171.216 // MAC address is hexadecimal (0~f), so 'g' is wrong.
Error Pattern
wolf@wolf:~/tcpip_hw4$ \\----At least, you should prevent stupid typing from happening as above.----//
```

This is not a command. It is the suggestion for this homework.

Rules

1. Please do your homework by using C language and ensure your homework can be compiled under Ubuntu 14.04.
2. You have to upload your own Makefile to compile your program.
3. Deducting points if you do not follow above restrictions.
4. Do not copy assignment from anyone. All participants will get **ZERO**.
5. We will notice your demonstration time later by email.
6. You can ask TAs any questions about this assignment except debugging.

TAs email: **net_ta@net.nsysu.edu.tw**

Lab: Network & System Laboratory-**EC5018** (11:00a.m. -5:30p.m.)

Upload

Please compress your homework to **zip** or **tar** and upload to National Sun Yat-Sen Cyber University. Name your homework to “Student ID_TCPIP_HW4”.

Example: **M013040001_TCPIP_HW4.zip**

Deadline

You should upload your assignment before **2015 /11/ 04 (Wed.) 23:59**. Any late submission will not be entertained.

Hint

It is important:

1. structure of **arp_packet** in “arp.h” .
2. **ioctl()**
3. **htons()** and **ntohs()**
4. **Wireshark** can help you know what the packet fields are.