

EECS 565  
Qixiang Liu  
04/06/2019

## Packet sniffing

### Introduction:

There are two tasks in the project 2. The project works on the Wireshark. The goal is to analyze different sniffed packets and data. The first task is about DNS requests and HTTP traffic in my own IP address. The second task is about other devices that are connected to the same WIFI network, excluding my own IP address.

### Background:

First, I should know some professional terms.

DNS: The domain name system; In other words, I check websites through domain names, like google.com(8.8.8.8) or youtube.com(199.223.232.0, etc). DNS translates IP addresses to domain names;

HTTP: HyperText Transfer Protocol. The protocol defines how messages are formatted and transmitted and how to deal with commands during response. HTTP is not safe, so communication is not protected between users and browser. HTTPS, 'S' is secure, and it is encoded by TLS and SSL.

TCP/IP: Transmission Control Protocol is complemented the IP. It is reliable to transmit data.

ARP: Address Resolution Protocol achieves physical address of TCP/IP address according to IP address. It can communicate with MAC address. ICMP: Internet Control Message Protocol.

DHCP: Dynamic Host Configuration Protocol is used on UDP/IP networks.

How to use Wireshark:

Check local IP: "ipconfig"

Filter interface: "ip src host <IP address>" eg. ip src host 192.168.0.13. or choose WIFI/WLAN interface

Filter more: http.request.method == POST

Result:

192.168.0.13 is my own IPv4 address.

2001...8de... is my own IPv6 address.

The image shows a Wireshark packet capture of a DNS query and response. The packet list on the left shows a standard query from 192.168.1.100 to 192.168.1.1. The packet details on the right show the query for google.com. The packet bytes on the bottom show the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
33	3.181516	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	123	Standard query response 0xc35b AAAA ssl.gstatic.com AAAA 2607:f8b0:4009:804:
34	3.183243	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	111	Standard query response 0xf756 A ssl.gstatic.com A 172.217.4.227
35	3.183248	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	137	Standard query response 0xcd68 AAAA people-pa.clients6.google.com AAAA 2607:
36	3.183250	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	125	Standard query response 0xfdc0 A people-pa.clients6.google.com A 172.217.0.10
37	3.184652	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	103	Standard query 0x5e94 AAAA gmailmail.l.google.com
38	3.184899	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	103	Standard query 0x6160 A gmailmail.l.google.com
41	3.203095	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	131	Standard query response 0x5e94 AAAA gmailmail.l.google.com AAAA 2607:f8b0:40
42	3.203101	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	119	Standard query response 0x6160 A gmailmail.l.google.com A 216.58.216.101
450	6.241796	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	97	Standard query 0x4ee3 AAAA ww3.l.google.com
451	6.241996	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	97	Standard query 0x786a A ww3.l.google.com
452	6.335863	2001:48f8:9004:8de...	2001:48f8:9004:8de...	DNS	125	Standard query response 0x4ee3 AAAA ww3.l.google.com AAAA 2607:f8b0:4009:80

Source Port: 5559  
Destination Port: 53  
Length: 49  
Checksum: 0x2cbc [unverified]  
[Checksum Status: Unverified]  
[Stream index: 6]  
[Timestamps]

▼ Domain Name System (query)  
Transaction ID: 0x5e94  
▼ Flags: 0x0100 Standard query  
0... .. = Response: Message is a query  
.000 0... .. = Opcode: Standard query (0)  
.... ..0... .. = Truncated: Message is not truncated

0000 64 77 7d e3 f3 a2 a8 66 7f 28 29 ce 86 dd 60 02 dw) ... f ( ) ...  
0010 4a b4 00 31 11 ff 20 01 48 f8 90 04 08 de 5c fd J ... H ...  
0020 92 29 67 3a fb cc 20 01 48 f8 90 04 08 de 66 77 )g... H ... fw  
0030 7d ff fe e3 f3 a2 f2 f3 00 35 00 31 2c bc 5e 94 } ... 5, 1, ...  
0040 01 00 00 01 00 00 00 00 00 00 0a 67 6f 6f 67 6c email... googl  
0050 65 6d 61 69 6c 01 6c 06 67 6f 6f 6f 6c 65 03 63 gmail.l. google.c  
0060 6f 6d 00 00 1c 00 01 om ...

The response to this DNS query is in this frame (dns\_response.in)

Packets: 559 · Displayed: 20 (3.6%) · Dropped: 0 (0.0%)

Profile: Default

HTTP: [www.4399.com](http://www.4399.com) is a Chinese game website. I think it is not safe website. The website has many advertisements. In the information tab, GET means get information from the website. When I login the website, I would get POST request from the website. In addition, I got my login information, including username,password, and email.

The screenshot shows the Wireshark network protocol analyzer interface. The top pane (Packet List) displays a list of captured packets, with the selected packet (No. 161) being a POST request to /ptlogin/register.do. The middle pane (Packet Details) shows the structure of the selected packet, including the HTTP request line and the body. The bottom pane (Packet Bytes) shows the raw data of the packet in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
232	13.055713	192.168.0.13	157.185.179.197	HTTP	668	GET /css/sy_v3.css HTTP/1.1
233	13.057306	192.168.0.13	157.185.179.197	HTTP	676	GET /css/newSimpleHead.css HTTP/1.1
234	13.057463	192.168.0.13	157.185.179.197	HTTP	658	GET /jss/click_trace.js HTTP/1.1
235	13.057582	192.168.0.13	157.185.179.197	HTTP	467	GET /upload_pic/2019/4/4/4399_1540221226.jpg HTTP/1.1
236	13.057788	192.168.0.13	157.185.179.197	HTTP	467	GET /upload_pic/2019/4/1/4399_16160499990.jpg HTTP/1.1
237	13.057948	192.168.0.13	157.185.179.197	HTTP	468	GET /upload_pic/2019/3/29/4399_16362292148.jpg HTTP/1.1
238	13.058347	192.168.0.13	157.185.179.197	HTTP	468	GET /upload_pic/2019/3/27/4399_16350058933.jpg HTTP/1.1
239	13.063290	192.168.0.13	157.185.179.197	HTTP	468	GET /upload_pic/2019/3/22/4399_16152625119.jpg HTTP/1.1
240	13.063551	192.168.0.13	157.185.179.197	HTTP	468	GET /upload_pic/2019/3/11/4399_15440604579.jpg HTTP/1.1
241	13.063873	192.168.0.13	157.185.179.197	HTTP	468	GET /upload_pic/2019/3/11/4399_15441251756.jpg HTTP/1.1
242	13.065065	192.168.0.13	203.119.129.114	HTTP	542	GET /heatmap.gif?id=30039538&x=1366&y=230&w=638&s=1280x800&b=safari&c=1&r=6a=1&rar
243	13.065174	192.168.0.13	157.185.179.197	HTTP	668	GET /css/newSimpleHead.css HTTP/1.1
161	51.413169	192.168.0.13	118.184.184.69	HTTP	576	POST /ptlogin/register.do HTTP/1.1 (application/x-www-form-urlencoded)

Form item: "userNameLabel" = "4399用户名"  
 Form item: "level" = "4"  
 Form item: "username" = "159753111"  
 Form item: "password" = "123123"  
 Form item: "passwordveri" = "123123"  
 Form item: "email" = ""

## Task 2:

ARP: The host broadcasts the ARP request to all hosts on the network, and receives the return message, determines the physical address of the target IP address, and stores the IP address and hardware address in the local ARP cache, and directly queries the ARP cache on the next request. Although everyone can get the ARP protocol, the protocol can be ignored except target ip address.

arp						
No.	Time	Source	Destination	Protocol	Length	Info
242...	146.211366	Apple_4a:6a:83	Apple_28:29:ce	ARP	42	192.168.0.43 is at 88:e9:fe:4a:6a:83
247...	155.165645	Apple_28:29:ce	Broadcast	ARP	42	Who has 192.168.0.47? Tell 192.168.0.13
247...	155.225234	Apple_ee:05:7b	Apple_28:29:ce	ARP	42	192.168.0.47 is at 68:64:4b:ee:05:7b
247...	155.225244	Apple_ee:05:7b	Apple_28:29:ce	ARP	42	192.168.0.47 is at 68:64:4b:ee:05:7b
279...	160.854858	HitronTe_e3:f3:a2	Apple_28:29:ce	ARP	42	Who has 192.168.0.13? Tell 192.168.0.1
279...	160.854910	Apple_28:29:ce	HitronTe_e3:f3:a2	ARP	42	192.168.0.13 is at a8:66:7f:28:29:ce
332...	195.463379	Apple_82:df:f4	Broadcast	ARP	42	Who has 192.168.0.12? Tell 192.168.0.40
332...	195.463548	Apple_82:df:f4	Broadcast	ARP	60	Who has 192.168.0.12? Tell 192.168.0.40
336...	196.487381	Apple_82:df:f4	Broadcast	ARP	42	Who has 192.168.0.12? Tell 192.168.0.40
336...	196.487549	Apple_82:df:f4	Broadcast	ARP	60	Who has 192.168.0.12? Tell 192.168.0.40
337...	199.888599	HitronTe_e3:f3:a2	Apple_28:29:ce	ARP	42	Who has 192.168.0.13? Tell 192.168.0.1
337...	199.888704	Apple_28:29:ce	HitronTe_e3:f3:a2	ARP	42	192.168.0.13 is at a8:66:7f:28:29:ce
▶ Frame 33294: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
▶ Ethernet II, Src: Apple_82:df:f4 (8c:85:90:82:df:f4), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
▼ Address Resolution Protocol (request)						
Hardware type: Ethernet (1)						
Protocol type: IPv4 (0x0800)						
Hardware size: 6						
Protocol size: 4						
Opcode: request (1)						
Sender MAC address: Apple_82:df:f4 (8c:85:90:82:df:f4)						
Sender IP address: 192.168.0.40						
Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)						
Target IP address: 192.168.0.12						

This is a request, the other is replay

DHCP: turn off/turn on WIFI. DHCP has request and replay(ACK) packets.

dhcp						
No.	Time	Source	Destination	Protocol	Length	Info
452	52.080686	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x7e7154f9
657	53.942970	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x7e7154f9
671	54.083470	192.168.0.1	192.168.0.13	DHCP	500	DHCP ACK - Transaction ID 0x7e7154f9

There are two types of DHCP. One is request, the other is ACK.

- ▶ Bootp flags: 0x0000 (Unicast)
- Client IP address: 0.0.0.0
- Your (client) IP address: 0.0.0.0
- Next server IP address: 0.0.0.0
- Relay agent IP address: 0.0.0.0
- Client MAC address: Apple\_28:29:ce (a8:66:7f:28:29:ce)
- Client hardware address padding: 00000000000000000000
- Server host name not given
- Boot file name not given
- Magic cookie: DHCP
- ▶ Option: (53) DHCP Message Type (Request)
- ▶ Option: (55) Parameter Request List
- ▶ Option: (57) Maximum DHCP Message Size
- ▶ Option: (61) Client identifier
- ▶ Option: (50) Requested IP Address (192.168.0.13)
- ▶ Option: (51) IP Address Lease Time
- ▼ Option: (12) Host Name
  - Length: 7
  - Host Name: qixiang
- ▶ Option: (255) End
- Padding: 00000000000000000000

Host name: my computer's name  
IP address: my own IP  
MAC address:  
There are lots of information about my private content.

SSH protocol: SSH server connection

First of all, I accept a response from KU cycle server to my IP address. The server needs a password to login server. Information displays key exchange and new keys. When I communicate with KU server, all packets are encrypted.

No.	Time	Source	Destination	Protocol	Length	Info
48	4.116233	129.237.87.113	192.168.0.13	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.8)
53	4.186596	192.168.0.13	129.237.87.113	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_7.6)
55	4.232833	192.168.0.13	129.237.87.113	SSHv2	1426	Client: Key Exchange Init
56	4.233546	129.237.87.113	192.168.0.13	SSHv2	1042	Server: Key Exchange Init
67	4.647891	129.237.87.113	192.168.0.13	SSHv2	430	Server: [TCP ACKed unseen segment] , Elliptic Curve Diffie-Hellman Key Excha
69	4.659234	192.168.0.13	129.237.87.113	SSHv2	82	Client: New Keys
71	4.747767	192.168.0.13	129.237.87.113	SSHv2	110	Client: Encrypted packet (len=44)

## Discussion:

Communication needs to know MAC address on the Internet, so we need ARP translation. DHCP ensures that IP address can only be used by one DHCP client at a time, and it can assign a permanent fixed IP address to the user. DHCP has two other working stages that are DHCP discover and DHCP offer.

## Conclusion:

If we are on the same network as others, when we log in to the http protocol website, others will get your account number and password through this method. Therefore, the outside WiFi should not be connected.

## Worked Cited:

<https://www.youtube.com/watch?v=pBj-7ez1RW0>

<https://blog.csdn.net/longwang155069/article/details/50107911>