모바일 컴퓨팅 HW6

2018037356 안동현

1. DHCP exhaustion attack 이 무엇인지 설명하고, 이를 막기위한 방법을 제시하세요. (7 pts)

DHCP exhaustion attack(starvation attack) 이란 공격자가 의도적으로 위조 MAC 주소를 반복적으로 계속 생성해 DHCP 서버를 향해 DHCP Discover 를 날려서 IP 를 예약받아, DHCP 의 IP pool 을 소진시키는 공격입니다. 이러한 공격 때문에 저희의 정상적인 클라이언트는 IP 주소를 할당받지 못하게 되는 일종의 DOS 공격이라고 할 수 있습니다.

이러한 공격이 먹히는 이유는 근본적으로 위조 MAC 주소들이 DHCP 가 실제 서로 다른 머신들이라고 판단하기 때문입니다. 따라서 L3 장비에서 한 포트에 허용하는 MAC 주소에 제한을 주면 해당 공격을 완화할 수 있을거라고 판단됩니다.

이러한 방법으로는 Port-Security 를 설정하는 방법이 있습니다. 해당 방법은 특정 포트에 미리 지정된 MAC 주소들만을 허용하고 나머지 MAC 주소의 연결이 감지되면 해당 포트를 차단 시키는 방법입니다.

Relay Agent 를 사용하면 특정 홉에 존재하는 머신들의 DHCP 요청은 모두 해당 위치로 가게 되고, 그곳에서부터 Unicast 로 DHCP 서버로 보내지게 됩니다. 만약 해당 Agent 에서 위조된 MAC 주소를 차단시킬 수 있다면 공격에 대한 방어책이 될 것입니다.

2. DHCP MITM (man in the middle) attack 이 무엇인지 설명하고, 이를 막기위한 방법을 제시하세요. (7 pts)

DHCP MITM attack 이란 공격자가 위조 DHCP 서버의 역할을 수행해서, 클라이언트가 DHCP 요청을 할 때 자신의 IP 주소를 게이트웨이나 DNS 서버의 주소로 속이는 공격 방법입니다.

이 경우 클라이언트가 외부 네트워크로 보내는 모든 패킷이 공격자에게 들어오기에 중간자 공격으로 패킷을 스니핑하거나 변조하는 것이 가능해집니다.

이를 막기 위해서는 L3 장비에서 정상적인 DHCP 서버가 연결된 포트를 제외한 나머지 포트를 신뢰할 수 없는 포트로 지정해서 필터링 과정을 거쳐 DHCP offer 와 ack 를 차단시킵니다. 즉 서버 -> 클라이언트 패킷을 차단시키는 것입니다. 하지만 요정은 가능해야하므로 Discover, Request 는 차단하지 않습니다.

위 방법을 사용하기 위해서 DHCP snooping 설정시 스위치의 포트를 Trusted 포트와 Untrusted 포트로 구분합니다.

공격자의 위조된 DHCP 서버가 untrusted 포트 뒤에 있다면 이것이 클라이언트에게 offer, ack 를 통해 잘못된 정보를 보내는 것을 막을 수 있습니다.

이 역시 Relay Agent 를 통해 해당 Agent 에서 정상적인 DHCP 가 연결된 포트만 trusted 로 설정하고 나머지는 전부 untrusted 로 설정할 수 있기에 Relay Agent 는 DHCP 보안에 있어서 여러모로 이득이 많은 방법입니다.

3. 다음의 trace 를 보고 답하여라.

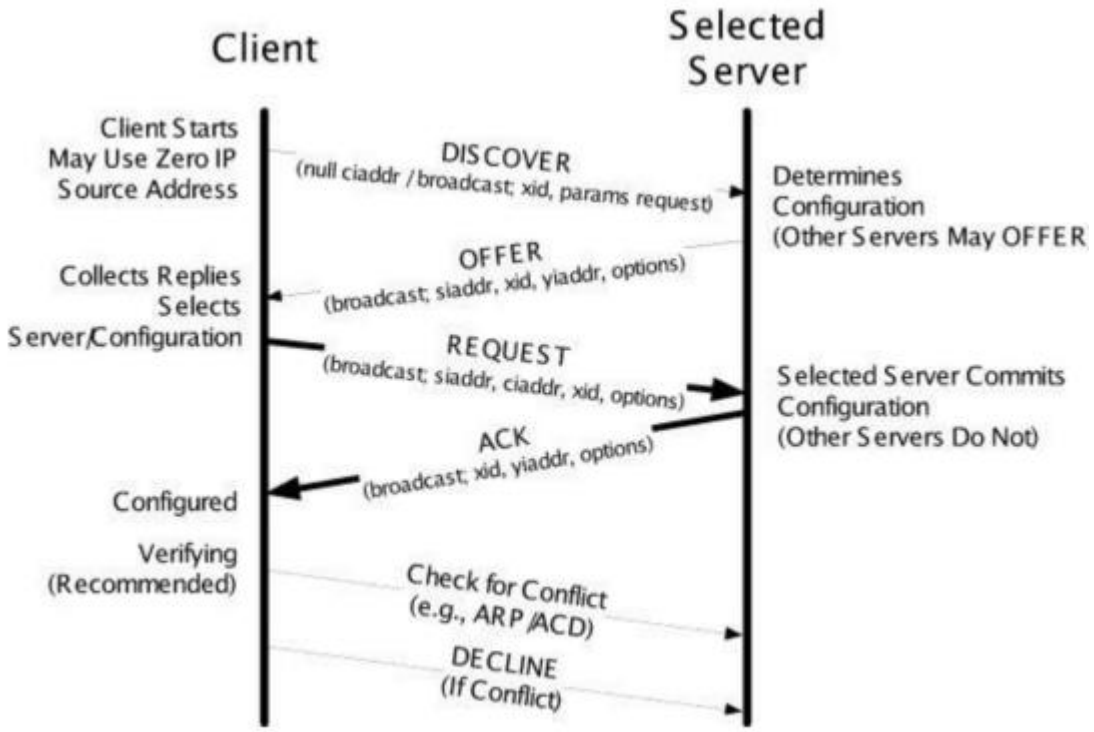
192.168.0.6	192.168.0.1	DHCP	342	DHCP Release - Transaction ID 0xab114d66
0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x317959da
192.168.0.1	192.168.0.6	DHCP	590	DHCP Offer - Transaction ID 0x317959da
0.0.0.0	255.255.255.255	DHCP	356	DHCP Request - Transaction ID 0x317959da
192.168.0.1	192.168.0.6	DHCP	590	DHCP ACK - Transaction ID 0x317959da
AskeyCom_46:be:82	Broadcast	ARP	42	Gratuitous ARP for 192.168.0.6 (Request)
192.168.0.6	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x775e024c
192.168.0.1	192.168.0.6	DHCP	590	DHCP ACK - Transaction ID 0x775e024c

a. 위의 trace 에서 gratuitous ARP 가 사용된 이유는 무엇인가? (2 pts)

할당받은 IP 가 중복된 IP 인지 아닌지 확인하기 위해서입니다. 만약 요청에 대해서 이상한 MAC 주소가 온다면 해당 IP 주소는 정복돼서 할당된 IP 입니다.

b. 만일 위의 gratuitous ARP 에 대해서 누군가가 응답을 한다면 어떠한 DHCP message 가 DHCP server 로 전달되는가? (2 pts)

누군가가 응답을 했다는 소리는 IP 가 중복 할당이 되었다는 소리이므로,



해당 자료에 의하면 DHCP Decline 을 보낼 것입니다.

c. 만일 Rapid DHCP 가 사용된다면 위의 trace 에서 몇번 packet 은 생략될 수 있는가? (3pts)

Discover 와 ACK 만 있으면 되므로,

3,4 번 패킷은 생략이 가능합니다.

d. DHCP 를 사용하여 configuration 할 수 있는 4 개의 parameter 는 무엇인가? (2pts)

IP Address (IP 주소):

DHCP 를 통해 클라이언트는 동적으로 할당된 IP 주소를 받을 수 있습니다.

Subnet Mask (서브넷 마스크):

서브넷 마스크는 네트워크의 IP 주소 범위를 식별합니다. DHCP 를 사용하면 클라이언트에게 서브넷 마스크가 할당됩니다.

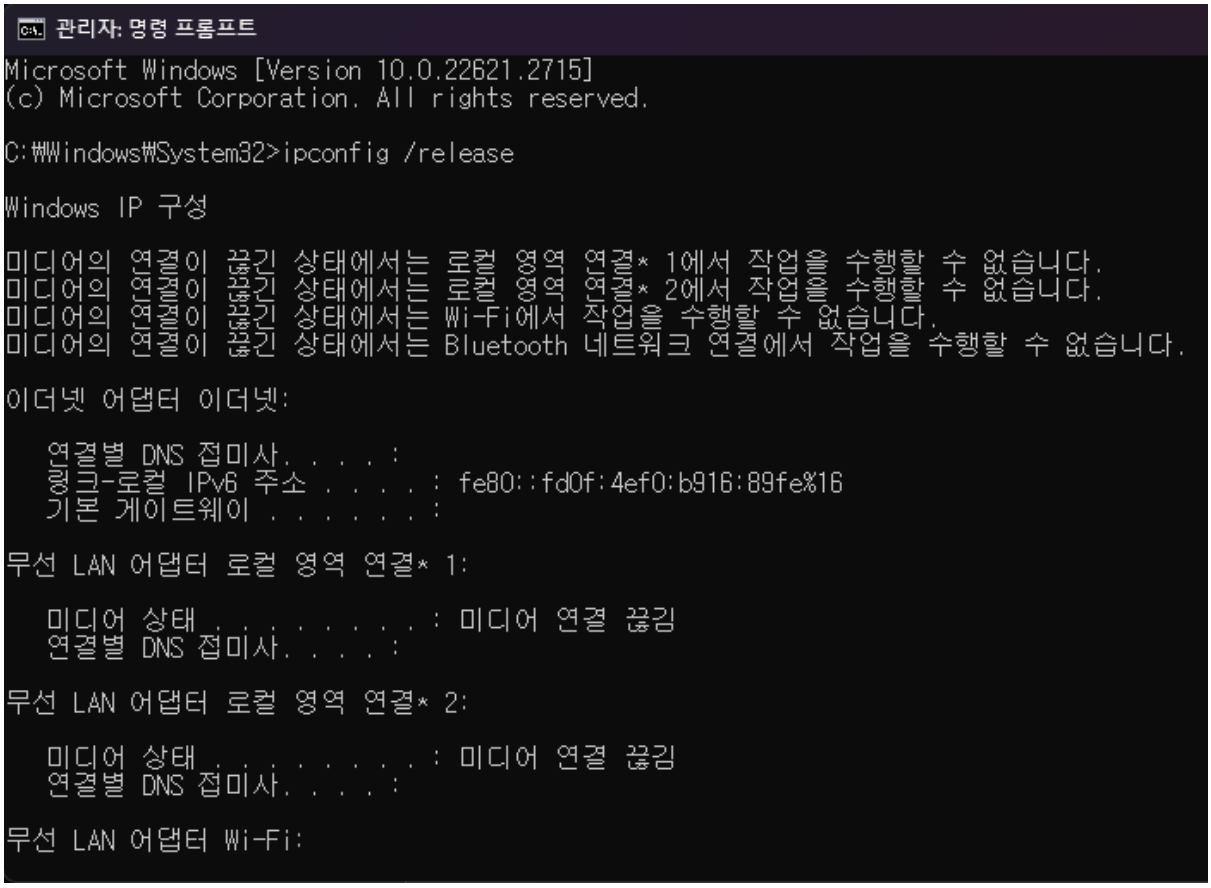
Default Gateway (기본 게이트웨이):

기본 게이트웨이는 클라이언트가 다른 네트워크로 통신할 때 사용하는 라우팅 포인트를 나타냅니다. DHCP 를 통해 클라이언트는 기본 게이트웨이 주소를 받을 수 있습니다.

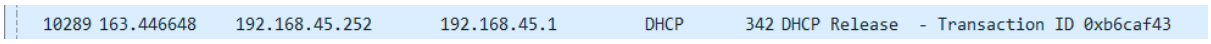
DNS Server Addresses (DNS 서버 주소):

DHCP 를 사용하면 클라이언트에게 동적으로 할당된 DNS 서버의 IP 주소를 제공할 수 있습니다. DNS 서버는 도메인 이름을 IP 주소로 해석하는 데 사용됩니다.

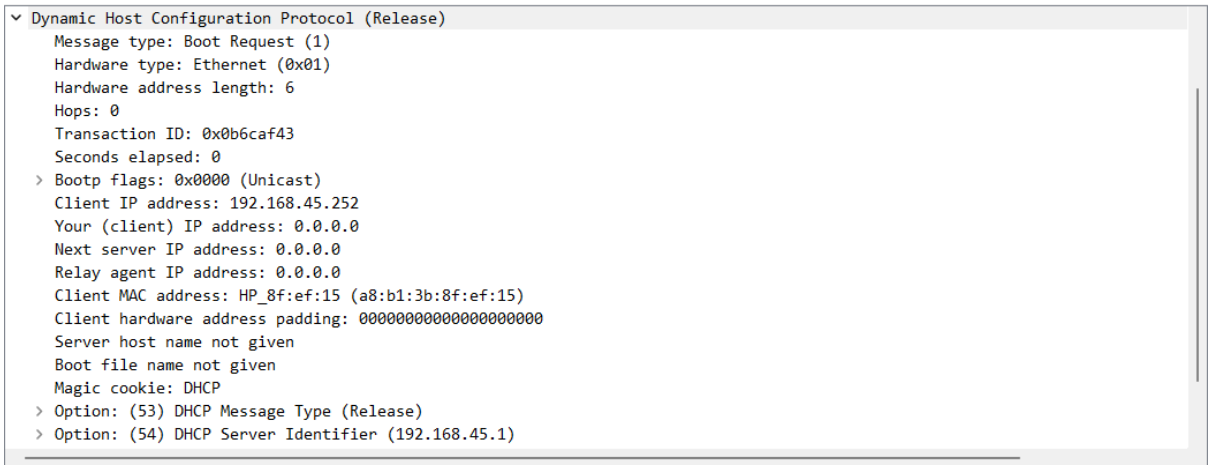
4. Wireshark 을 활용하여 DHCP packet 을 capture 하고 DHCP packet 의 header 를 자세히 설명하세요. pcap file 을 업로드 하세요. (7 pts)



먼저 와이어 샤크를 키고, 현재 IP를 반환해서 인터넷 연결을 끊어봤습니다.



이렇게 저의 클라이언트에서 DHCP로 Release 메시지를 통해 할당되었던 IP를 반환해주는 것을 확인할 수 있습니다.



해당 DHCP 패킷의 헤더 정보입니다.

클라이언트에서 반환 요청을 한 만큼 Message type 이 Request임을 확인할 수 있고, Hardware type은 이더넷, 그리고 나머지 정보들과 현재는 IP가 할당되어있기에 Client IP address 에 값이 들어있는 모습을 볼 수 있습니다.

이후 your IP address, server IP address, gateway IP address에는 값이 없음을 확인할 수 있고 해당 패킷을 보내는 클라이언트의 MAC주소가 담겨있습니다. 이후에는 패딩값이 채워져있는데 이는 검색해보니 0의 192옥텟 또는 추가 옵션을 위한 오버플로 공간임을 알 수 있었습니다.

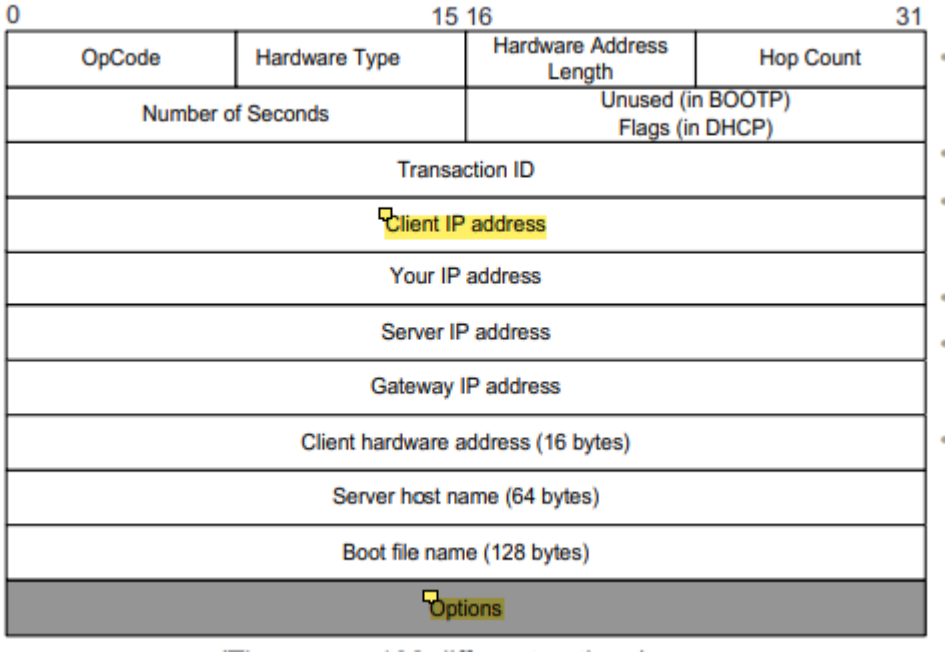
이후 옵션을 제외한 정보들이 전부 들어있지 않은 것을 보아 정말로 IP 반환을 위해서 클라이언트의 IP와 MAC 주소만이 들어있는 것이라고 추측할 수 있었습니다.

또한

```
Magic cookie: DHCP
```

이렇게 Magic cookie값을 가지고 있었는데,

DHCP는 BOOTP의 확장인만큼, 이 둘은 동일한 메시지를 가지고 있어서 해당 쿠키값이 없으면 구별하기가 불가능합니다. 따라서 쿠키값을 통해서 아래에서 나오는 정보들은 DHCP로 해석하라는 의미라고 볼 수 있습니다.



이러한 패딩값과 쿠키값은 강의 자료에 없었던 만큼 그렇게까지 중요한 정보가 아닐지도 모르지만, 찾아보는 과정이 즐거웠던 경험이었습니다.

```
C:₩Windows₩System32>ipconfig /renew

Windows IP 구성

미디어의 연결이 끊긴 상태에서는 로컬 영역 연결* 1에서 작업을 수행할 수 없습니다.
미디어의 연결이 끊긴 상태에서는 로컬 영역 연결* 2에서 작업을 수행할 수 없습니다.
미디어의 연결이 끊긴 상태에서는 Bluetooth 네트워크 연결에서 작업을 수행할 수 없습니다

이더넷 어댑터 이더넷:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::fd0f:4ef0:b916:89fe%16
   IPv4 주소 . . . . . . . . . : 192.168.45.252
   서브넷 마스크 . . . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . . : 192.168.45.1

무선 LAN 어댑터 로컬 영역 연결* 1:

   미디어 상태 . . . . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :

무선 LAN 어댑터 로컬 영역 연결* 2:

   미디어 상태 . . . . . . . . : 미디어 연결 끊김
   연결별 DNS 접미사. . . . :
```

다음은 renew를 통해서 다시 DHCP 요청 메시지를 보내봤습니다. 만약 지금 인터넷이 연결되어있었다면 해당 명령어는 DHCP "갱신"을 위해서 DHCP Request 메시지를 보냈을 것입니다. 하지만 방금 전 Release를 통해 IP를 반환했기에 위에서의 renew는 DHCP Discover 메시지를 보내게 됩니다.

```
10529 173.713772   0.0.0.0         255.255.255.255   DHCP    342 DHCP Discover - Transaction ID 0xa759ba56
10530 173.748153   192.168.45.1    192.168.45.252    DHCP    590 DHCP Offer    - Transaction ID 0xa759ba56
10531 173.752177   0.0.0.0         255.255.255.255   DHCP    352 DHCP Request  - Transaction ID 0xa759ba56
10532 173.837971   192.168.45.1    192.168.45.252    DHCP    590 DHCP ACK      - Transaction ID 0xa759ba56
```

이후에는 배웠던 대로 Offer -> Request -> ACK 순으로 메시지가 보내져 정상적으로 IP를 할당받아서

이렇게 인터넷 연결이 되는 것을 확인할 수 있었습니다.

자 이제 해당 패킷들의 헤더들을 살펴보겠습니다.

(1) Discover

```
Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xa759ba56
    Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
      0... .... .... .... = Broadcast flag: Unicast
      .000 0000 0000 0000 = Reserved flags: 0x0000
    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: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  > Option: (53) DHCP Message Type (Discover)
  > Option: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.45.252)
  > Option: (12) Host Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  > Option: (255) End
    Padding: 00000000000000
```

먼저 DHCP 서버를 찾기 위한 메시지 이기 때문에 Request 타입입니다.

하드웨어 타입은 Ethernet이고, Hops는 Relay Agent가 카운트를 시켜주는 것이기에 클라이언트에서 캡쳐되는 해당 패킷들은 무조건 0일 것이라고 추측됩니다. 또한 Transaction ID를 볼 수 있는데 위쪽의 Release와 ID가 다름을 확인할 수 있습니다!

여기서 신기한 점이 있는데, 플래그를 보면 Unicast로 되어있는 점입니다.

0.0.0.0 -> 255.255.255.255 브로드캐스트인 패킷인데 왜 플래그는 Unicast일까요?

먼저 IP의 src, dest가 저렇다면 무조건 broadcast이므로

아마도 디폴트 플래그는 원래부터 0 이고, 서버에서 클라이언트에 보내는 Offer나 ACK같은 메시지에서만 플래그에 따라서 Unicast로 할지, broadcast로 할지 결정을 하는 것이라고 추측했습니다.

따라서 Discover 메시지에서는 큰 신경을 쓰지 않아도 될 것 같습니다.

아래를 계속해서 살펴보면, 정보가 들어있는 것은 클라이언트의 MAC주소 뿐입니다. DHCP서버를

찾는 Discover메시지 이기에 아무런 정보가 들어있지 않은 것입니다.

이후 여러 옵션들을 찾을 수 있었고

```
Option: (55) Parameter Request List
    Length: 14
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (31) Perform Router Discover
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (43) Vendor-Specific Information
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
    Parameter Request List Item: (252) Private/Proxy autodiscovery
```

그 중에서는 파라미터 정보가 가장 중요하다고 할 수 있습니다. 앞으로 서버에서 클라이언트에게 Offer, ACK를 보낼 때 정보를 여기에 담아서 줄 것입니다.

(2) Offer

```
Dynamic Host Configuration Protocol (Offer)
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xa759ba56
  Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
    0... .... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 0.0.0.0
  Your (client) IP address: 192.168.45.252
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type (Offer)
  Option: (54) DHCP Server Identifier (192.168.45.1)
  Option: (51) IP Address Lease Time
  Option: (1) Subnet Mask (255.255.255.0)
  Option: (3) Router
  Option: (6) Domain Name Server
  Option: (255) End
  Padding: 000000000000000000000000000000000000000000000000000000000000000000000000…
```

Message Type이 Reply로 바뀌고,

src:192.168.45.1 dest:192.168.45.252 인 Unicast로 패킷을 보냅니다. 사실 dest IP주소는 클라이언트에게 할당이 되기 전인 주소지만 아마 효율을 위해서 할당이 될 것이라 가정하고 Broadcast가 아니라 Unicast로 패킷을 보내는 것이라고 추측할 수 있습니다.

또 바뀐 점은 클라이언트에게 해당 IP 주소를 사용할 수 있다는 것을 알려주기 위해서

Your (client) IP address: 192.168.45.252 를 설정해서 보내줍니다. 이 다음 중요한 부분은

```
▾ Option: (1) Subnet Mask (255.255.255.0)
     Length: 4
     Subnet Mask: 255.255.255.0
▾ Option: (3) Router
     Length: 4
     Router: 192.168.45.1
▾ Option: (6) Domain Name Server
     Length: 8
     Domain Name Server: 210.220.163.82
     Domain Name Server: 219.250.36.130
```

이렇게 서브넷 마스크, 게이트웨이 주소, DNS 서버 주소 까지 알려주며, 주요 파라미터 4개를 알려주는 모습을 볼 수 있었습니다.

(3) Request

```
▾ Dynamic Host Configuration Protocol (Request)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xa759ba56
    Seconds elapsed: 0
  ▾ Bootp flags: 0x0000 (Unicast)
       0... .... .... .... = Broadcast flag: Unicast
       .000 0000 0000 0000 = Reserved flags: 0x0000
    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: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
    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: (61) Client identifier
  > Option: (50) Requested IP Address (192.168.45.252)
  > Option: (54) DHCP Server Identifier (192.168.45.1)
  > Option: (12) Host Name
  > Option: (81) Client Fully Qualified Domain Name
  > Option: (60) Vendor class identifier
  > Option: (55) Parameter Request List
  ▾ Option: (255) End
       Option End: 255
```

이제 Offer의 정보들을 확인하고 그에 맞춰서 임대를 요청합니다.

Discover와 매우 유사해 보이지만 Request에서는 옵션 부분에 특정 DHCP 서버에 대한 Identifier가 존재해 해당 DHCP에서 요청을 받아들이는 것이 가능합니다. 이제

```
Option: (55) Parameter Request List
    Length: 14
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (31) Perform Router Discover
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (43) Vendor-Specific Information
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
    Parameter Request List Item: (252) Private/Proxy autodiscovery
```

이 부분을 실제로 요청해주고 IP 주소를 임대해줍니다.

(4) ACK

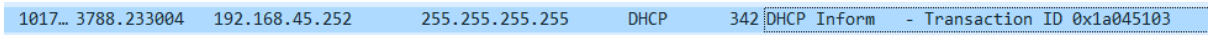
```
Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xa759ba56
    Seconds elapsed: 0
  Bootp flags: 0x0000 (Unicast)
      0... .... .... .... = Broadcast flag: Unicast
      .000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.45.252
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  Option: (53) DHCP Message Type (ACK)
  Option: (54) DHCP Server Identifier (192.168.45.1)
  Option: (51) IP Address Lease Time
  Option: (1) Subnet Mask (255.255.255.0)
  Option: (3) Router
  Option: (6) Domain Name Server
  Option: (255) End
    Padding: 0000000000000000000000000000000000000000000000000000000000000000000000000000…
```

Message Type이 Reply 이고 저의 IP 주소가 192.168.45.252임을 확정시켜줍니다. 그와 동시에

옵션. 특히 서브넷 마스크, 게이트웨이 주소, DNS 서버 주소 등 4개(IP 포함)의 파라미터 역시 확정시켜줍니다.

```
Option: (1) Subnet Mask (255.255.255.0)
    Length: 4
    Subnet Mask: 255.255.255.0
Option: (3) Router
    Length: 4
    Router: 192.168.45.1
Option: (6) Domain Name Server
    Length: 8
    Domain Name Server: 210.220.163.82
    Domain Name Server: 219.250.36.130
```

(5) Inform



이 외에도 패킷 중에서 Inform 패킷을 받을 수가 있었는데, 해당 패킷은 IP 주소는 이미 가지고 있고 다른 로컬 파라미터만을 요구하는 DHCP 메시지 이기에 클라이언트 -> 서버로 보내는 패킷임에도 불구하고 Unicast로 보내는 것을 확인할 수 있었습니다.

```
Bootp flags: 0x0000 (Unicast)
    0... .... .... .... = Broadcast flag: Unicast
    .000 0000 0000 0000 = Reserved flags: 0x0000
  Client IP address: 192.168.45.252
  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: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
  Client hardware address padding: 00000000000000000000
  Server host name not given
```

실제로 이미 IP가 존재하기에 위 부분을 보면 Client IP address에 값이 들어가 있는 것을 확인할 수 있고,

```
Option: (55) Parameter Request List
    Length: 14
    Parameter Request List Item: (1) Subnet Mask
    Parameter Request List Item: (3) Router
    Parameter Request List Item: (6) Domain Name Server
    Parameter Request List Item: (15) Domain Name
    Parameter Request List Item: (31) Perform Router Discover
    Parameter Request List Item: (33) Static Route
    Parameter Request List Item: (43) Vendor-Specific Information
    Parameter Request List Item: (44) NetBIOS over TCP/IP Name Server
    Parameter Request List Item: (46) NetBIOS over TCP/IP Node Type
    Parameter Request List Item: (47) NetBIOS over TCP/IP Scope
    Parameter Request List Item: (119) Domain Search
    Parameter Request List Item: (121) Classless Static Route
    Parameter Request List Item: (249) Private/Classless Static Route (Microsoft)
    Parameter Request List Item: (252) Private/Proxy autodiscovery
```

해당 파라미터들을 요청하고 있었습니다.

```
1017… 3788.275230   192.168.45.1        192.168.45.252      DHCP      590 DHCP ACK      - T
```

그것을 ACK를 쳐주고

```
1019… 3793.646251   192.168.45.252      192.168.45.1        DHCP      342 DHCP Request  - Transaction ID 0xde979899
1019… 3793.685064   192.168.45.1        192.168.45.252      DHCP      590 DHCP ACK      - Transaction ID 0xde979899
1389… 5604.011069   192.168.45.252      192.168.45.1        DHCP      342 DHCP Request  - Transaction ID 0x74b09fc3
1389… 5604.064411   192.168.45.1        192.168.45.252      DHCP      590 DHCP ACK      - Transaction ID 0x74b09fc3
```

이후에는 반복적으로 임대가 만료되기 임박할 때 갱신 요청을 해주는 패킷 역시 볼 수 있었습니다. 두번의 Request가 아니라 한번의 Request로 ACK가 되었던 것을 보아하니 T1에서 바로 갱신이 되고 T2까지 가지는 않았던 걸로 보입니다. 해당 경우에도 이미 IP 주소는 존재하므로

```
Client IP address: 192.168.45.252
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: HP_8f:ef:15 (a8:b1:3b:8f:ef:15)
Client hardware address padding: 00000000000000000000
Server host name not given
Boot file name not given
```

이곳을 보면 Client IP address가 이미 존재하는 것을 확인할 수 있었습니다.

5. 느낀점

과제를 진행하면 서 느낀점은 생각보다 예상과는 달라보이는 패킷들이 많았다는 점이었습니다. 이전 강의에서의 예제나 이론과는 달리 Unicast로 패킷들이 보내진다는 점이나, 헤더의 정보에서 패딩의 유무나 쿠키의 존재 역시 그랬습니다. 그 과정에서 검색과 추측을 통해 많은 것을 배워갔다고 자부할 수 있었습니다.