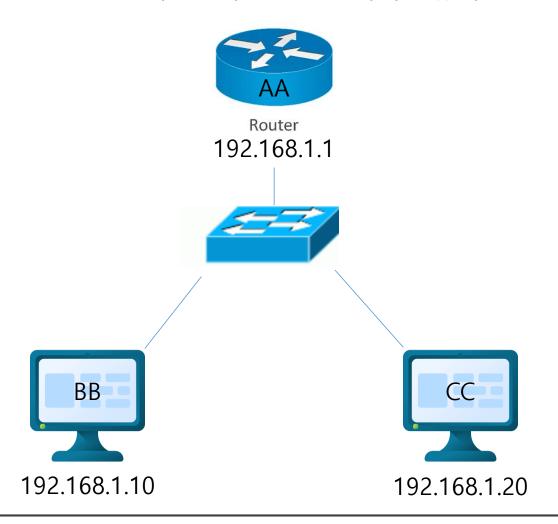
## ARP Table 정적으로

## 고정시키기

2020.08.03 김현진

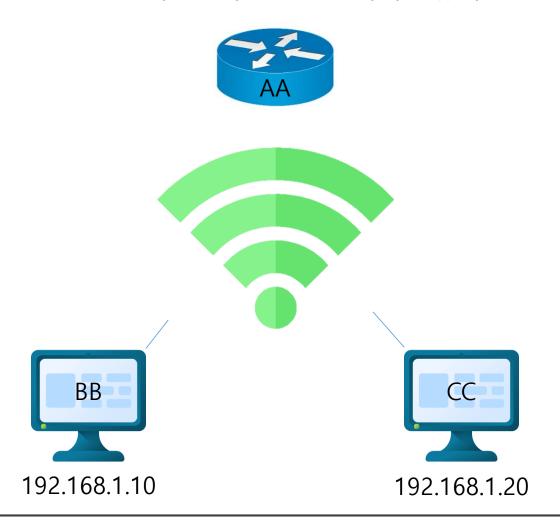
#### ARP Table

IP-Mac의 1:1 매칭 정보를 가지고 있다.



#### ARP Table

IP-Mac의 1:1 매칭 정보를 가지고 있다.



# ARP Spoofing

## ARP Table 정적으로 설정하기

ARP 명령어

netsh 명령어

#### ARP 명령어

arp -s [IP주소] [MAC주소]

C:\WINDOWS\system32>arp -s 192.168.35.197 40-5b-d8-ad-4a-c4

#### ARP 명령어

```
C:\WINDOWS\system32>arp -s 192.168.35.197 40-5b-d8-ad-4a-c4
                   192.168.35.105
                                          f0-6e-0b-7c-cf-67
                   192.168.35.145
                                          50-77-05-62-51-8f
                       168 35
                   192.168.35.197
                   194,100,33,433
                                                                 <sup>식</sup>적적적적적적적적적

정정정정정정정정정정
                   224.0.0.2
                   224.0.0.22
                                          01-00-5e-00-00-16
                   224.0.0.251
                   224.0.0.252
                                          01-00-5e-00-00-fc
                   224.0.0.253
                                          <u>01-00-5e-00-00-fd</u>
                   239.255.255.250
                                          01-00-5e-7f-ff-fa
                   255.255.255.255
                                          ff-ff-ff-ff-ff
                 인터페이스: 192.168.72.1
                                           --- 0x12
                                                                 요혀
                                                                 192 . 168 . 35 . 197
                   194,168,72,254
                   192.168.72.255
```

#### netsh 명령어

네트워크 설정 변경 명령어

netsh interface ip add neighbors "interface" <ip> <MAC>

C:#WINDOWS#system32<mark>>netsh interface ip add neighbors "wlan0" 192.168.35.197</mark> 40-5b-d8-ad-4a-c3

#### netsh 명령어

네트워크 설정 변경 명령어

C:\WINDOWS\system32>netsh interface ip add neighbors "wlan0" 192.168.35.197 40-5b-d8-ad-4a-c3

C:#WINDOWS#system32>arp	-a	
인터페이스: 192.168.255 인터넷 주소 224.0.0.22 239.255.255.250	물리적 주소	유형 정적 정적
인터페이스: 192.168.35. 인터넷 주소 192.168.35.1 192.168.35.197 224.0.0.22 239.255.255.250	물리적 주소 00-23-ee-de-99-91	60저 유도) 정 정 정 정 정 정 정
인터페이스: 192.168.72. 인터넷 주소 224.0.0.22 239.255.255.250	물리적 주소	유형 정적 정

## ARP Spoofing

인터페이스: 192.168.35 인터넷 주소 192.168.35.1 192.168.35.2 192.168.35.3 192.168.35.13 192.168.35.105 192.168.35.145 192.168.35.145 192.168.35.145 192.168.35.171 192.168.35.255 224.0.0.2 224.0.0.2 224.0.0.252 224.0.0.253 239.255.255.255	5.203 0x11 물리적 주소 00-23-aa-de-99-91 50-77-05-66-84-8f 00-24-d6-d1-61-7c d0-94-66-f6-d5-ba b8-5a-73-1b-42-25 f0-6e-0b-7c-cf-67 50-77-05-62-51-8f 64 7b ce e6 42 5c 00-0c-29-85-76-5c ff-ff-ff-ff-ff-ff 01-00-5e-00-00-16 01-00-5e-00-00-fb 01-00-5e-00-00-fc 01-00-5e-00-00-fd 01-00-5e-7f-ff-fa ff-ff-ff-ff-ff-ff	60저 저 저 저 저 저 저 저 저 저 저 저 저 저 저 저 저 저 저
인터페이스: 192.168.72 인터넷 주소 192.168.35.197 192.168.72.254 192.168.72.255	.1 0x12 물리적 주소 40-5b-d8-ad-4a-c4 UU-5U-5b-t8-3a-U4 ff-ff-ff-ff-ff	요 50 50 50 50 50 50 50 50 50 50 50 50 50
224.0.0.2 224.0.0.22 224.0.0.251 224.0.0.252 224.0.0.253 239.255.255.250	01-00-5e-00-00-02 01-00-5e-00-00-16 01-00-5e-00-00-fb 01-00-5e-00-00-fc 01-00-5e-7f-ff-fa	( 저 저 저 저 저 저 저 저 저

C:#WINDOWS#system32>arp -a		
		유형 정적 정적
192.168.35.197 40-5 224.0.0.22 01-0	적 주소 3-aa-de-99-91	유명 유도 저지 저지 저지 저지 저지
224.0.0.22 01-0	0x12 적 주소 0-5e-00-00-16 0-5e-7f-ff-fa	유형 정적 정적

#### K-XARP



```
* [ 9 ARP ] *
- 응답(Reply) 패킷 --
 보내는 MAC (Sender MAC): 00:0c:29:85:76:5c
 보내는 IP (Sender IP): 192.168.35.197
 받는 MAC(Target MAC): 00:24:d6:d1:61:7c
 받는 IP (Target IP): 192.168.35.203
!!!! ARP Spoofing Attack !!!!
   Attacker MAC is [ 00-0c-29-85-76-5c ]
***********************************
      ARP Table 정적으로 설정 [시작]
```

```
Request_List = []
ARP_Table={}
def Right Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
  if sIP in ARP_Table: # ip가 ARP_table에 있는지 확인한다.
    Mac_Check(sIP,sMAC)
  else :
    ARP Table[sIP] = sMAC
    #print(A
              tlP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3]))
              #print(tIP)
def Mac Check
  if ARP_Table
    print("
              if tIP in Request_List:
  else :
    print("¥
                  print("") # 이미 존재하는 Request IP
    print("#
    print("#
              else:
                                                                                                                  Request
    print("#
                  Request_List.append(tIP)
    print("#
    print("#
    print("#
                                            #")
    print("#
    print("#################")
    print("#
                                            #")
    print("#
               ARP Table 정적으로 설정 [시작]
    print("#
    Right_Mac = ARP_Table.get(sIP)
    Static_Setting(sIP,Right_Mac)
```

```
Request_List = []
ARP Table={}
def Right_Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
 if sIP in ARP Table: # ip가 ARP table에 있는지 확인한다.
   Mac Check(sIP,sMAC)
  else :
         if sIP in Request_List :# request 리스트 안에 있는 지 sIP 조회 하는 문구
   #print(A
             Right_Reply(sIP,sMAC) # 올바른 reply패킷이므로 ARP_Table에 조회하는 함수로 넘어간다.
def Mac Checl
 if ARP_Tabl
         else :
    print("
             print("₩n!!!! request 리스트엔 없는 reply패킷 !!! ARP 스푸핑이 의심 됩니다 !!!!₩n")
  else :
   print("\
             if sIP in ARP_Table: # ARP_Table 에 Reply sender IP가 있는지 체크한다.
   print("#
   print("
                Mac_Check(sIP,sMAC) # Mac 주소가 중복 되었는지 체크한다.
                                                                                                  Reply
   print("#
   print("#
   print("#
                                     #")
   print("#
                                     #")
   print("#
   print("#################")
   print("#
             ARP Table 정적으로 설정 [시작]
   print("#
   print("#
    Right_Mac = ARP_Table.get(sIP)
```

Static\_Setting(sIP,Right\_Mac)

```
Request_List = []
ARP_Table={}
 ef Right_Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
  ıf sIP ın ARP_Table:# ıp가 ARP_table에 있는지 확인한다.
    Mac_Check(sIP,sMAC)
  else :
    ARP\_Table[sIP] = sMAC
    #print(ARP_Table)
def Mac_Check(sIP,sMAC): # Mac 주소가 중보되었는지 체크하는 함수
  if ARP_Table.get(sIP) == sMAC :
    print(" ******* 정상적인 응답(Reply) 패킷입니다. *******")
  else :
    print("\n##############")
    print("#
                                               #")
    print("#
                  !!!! ARP Spoofing Attack !!!!
    print("#
    print("#
                                                #")
    print("#
              Attacker MAC is [",sMAC,"]!
    print("#
                                                #")
                                                #")
    print("#
    print("################")
    print("#
                                                #")
    print("#
                ARP Table 정적으로 설정 [시작]
                                                #")
    print("#
    Right_Mac = ARP_Table.get(sIP)
    Static_Setting(sIP,Right_Mac)
```

```
Request_List = []
ARP_Table={}
 ef Right_Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
  ıf sIP ın ARP Table: # ıp가 ARP table에 있는지 확인한다.
    Mac_Check(sIP,sMAC)
  else :
    ARP Table[sIP] = s
    #print(ARP_Table)
                 def Right_Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
                    if sIP in ARP_Table: # ip가 ARP_table에 있는지 확인한다.
def Mac Check(sIP,sMAC
 if ARP Table.get(sIP)
                         Mac_Check(sIP,sMAC)
    print(" ****** 정상
  else :
                     else:
    print("\n#######
    print("#
                         ARP_Table[sIP] = sMAC
    print("#
    print("#
                         #print(ARP_Table)
    print("#
    print("#
            Attack
    print("#
                                          #")
    print("#
    print("#
                                          #")
              ARP Table 정적으로 설정 [시작]
    print("#
    print("#
                                          #")
    Right_Mac = ARP_Table.get(sIP)
    Static_Setting(sIP,Right_Mac)
```

```
Request_List = []
ARP_Table={}
def Right_Reply(sIP,sMAC): # ARP 테이블에 IP가 존재하는지 체크하는 함수
  if sIP in ARP_Table: # ip가 ARP_table에 있는지 확인한다.
    Mac_Check(sIP,sMAC)
  else :
    ARP\_Table[sIP] = sMAC
    #print(ARP_Table)
 ef Mac_Check(sIP,sMAC): # Mac 주소가 중보되었는지 체크하는 함수
  if ARP Table.get(sIP) == sMAC :
    print(" ******* 정상적인 응답(Reply) 패킷입니다. *******")
  else :
    print("\n##############")
    print("#
                                                #")
    print("#
                  !!!! ARP Spoofing Attack !!!!
    print("#
    print("#
                                                #")
    print("#
              Attacker MAC is [",sMAC,"]!
    print("#
                                                #")
                                                #")
    print("#
    print("################")
    print("#
                                                #")
    print("#
                ARP Table 정적으로 설정 [시작]
                                               #")
                                                #")
    print("#
    Right_Mac = ARP_Table.get(sIP)
    Static_Setting(sIP,Right_Mac)
```

```
def Mac_Check(sIP,sMAC): # Mac 주소가 중보되었는지 체크하는 함수
Request List = []
                           if ARP_Table.get(sIP) == sMAC :
ARP Table={}
                              print(" ****** 정상적인 응답(Reply) 패킷입니다. *******")
def Right_Reply(sIP,sMAC): # ARP 테이블(
 if sIP in ARP Table : # ip가 ARP table
                           else:
   Mac_Check(sIP,sMAC)
                              print("\n################")
  else :
   ARP Table[sIP] = sMAC
                              print("#
   #print(ARP Table)
                              print("#
                              print("#
                                              !!!! ARP Spoofing Attack !!!!
  Mac Check(sIP,sMAC): # Mac 주소기
                              print("#
  if ARP Table.get(sIP) == sMAC :
   print(" ******* 정상적인 응답(Reply)
                              print("#
                                         Attacker MAC is [",sMAC,"]!
  else :
                              print("#
   print("#
                              print("#
   print("#
                              print("###############")
              !!!! ARP Spoofing
   print("#
   print("#
                              print("#
           Attacker MAC is [",sMA
   print("#
                                           ARP Table 정적으로 설정 [시작]
                              print("#
   print("#
   print("#
                              print("#
   print("#
            ARP Table 정적으로
   print("#
                              Right_Mac = ARP_Table.get(sIP)
   print("#
                              Static_Setting(sIP,Right_Mac)
   Right_Mac = ARP_Table.get(sIP)
                              # ARP reply 패킷을 다시 보내서 다른 맥주소가 오는지 확인 해볼
   Static_Setting(sIP,Right_Mac)
                              # static으로 설정하는 부분이 있어야함.
```

```
def Static_Setting(sIP,Right_Mac):
    output = os.popen('netsh interface ip add neighbors "wlan0" ' + sIP + " " + Right_Mac)
    print(output.read())
    print("###### ARP Table 정적으로 설정 [완료] ######\n")
    output = os.popen('arp -a')
    print(output.read())
```

```
###### ARP Table 정적으로 설정 [완료] ######
인터페이스: 192.168.255.1 --- 0x5
                                          유형
정적
정적
 인터넷 주소
                      물리적 주소
                      01-00-5e-00-00-16
 224.0.0.22
 239.255.255.250
                      01-00-5e-7f-ff-fa
인터페이스: 192.168.35.203 --- 0x11
  인터넷 주소
                      물리적 주소
                                          유형
                                          불처
 192 168 35 1
                      00-23-22-42-99-91
                                          정적
  192.168.35.197
                      40-5b-d8-ad-4a-c3
                                          낑끽
 224.0.0.22
                      01-00-5e-00-00-16
 239.255.255.250
                      01-00-5e-7f-ff-fa
인터페이스: 192.168.72.1 --- 0x12
                                          유형
정적
정적
 인터넷 주소
                      물리적 주소
                      01-00-5e-00-00-16
 224.0.0.22
 239.255.255.250
                      01-00-5e-7f-ff-fa
```

## Q & A

Thank you ☺