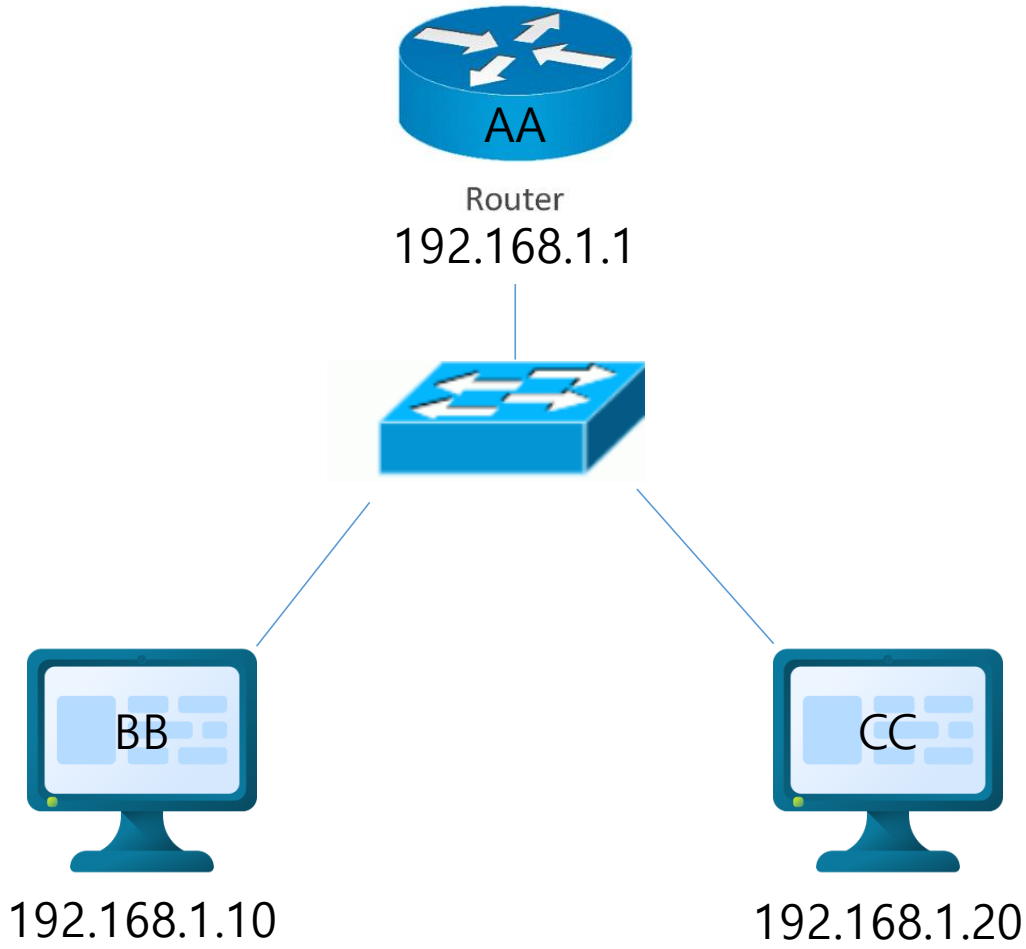
ARP Table 정적으로 고정시키기

2020.08.03 김현진

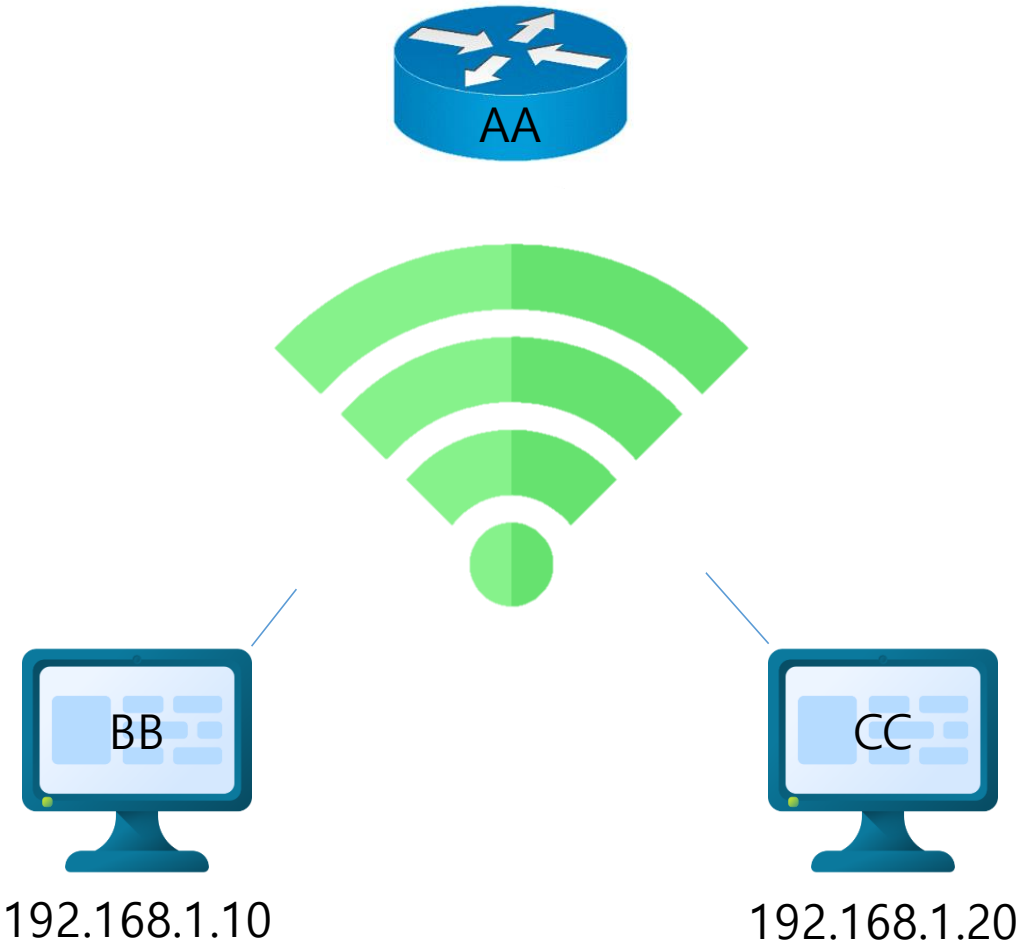
ARP Table

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]

ARP 명령어

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

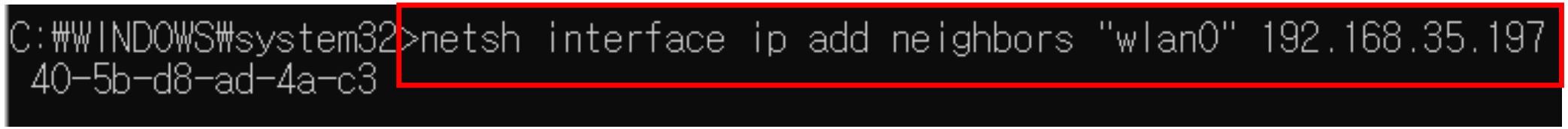
```
  192.168.35.105         f0-6e-0b-7c-cf-67     동적
  192.168.35.145         50-77-05-62-51-8f     동적
  192.168.35.171         64-7b-ce-e6-42-5c     동적
  192.168.35.197         40-5b-d8-ad-4a-c3     동적
  192.168.35.255         ff-ff-ff-ff-ff-ff     정적
  224.0.0.2              01-00-5e-00-00-02     정적
  224.0.0.22             01-00-5e-00-00-16     정적
  224.0.0.251            01-00-5e-00-00-fb     정적
  224.0.0.252            01-00-5e-00-00-fc     정적
  224.0.0.253            01-00-5e-00-00-fd     정적
  239.255.255.250        01-00-5e-7f-ff-fa     정적
  255.255.255.255        ff-ff-ff-ff-ff-ff     정적

인터페이스: 192.168.72.1 --- 0x12
  인터넷 주소             물리적 주소           유형
  192.168.35.197         40-5b-d8-ad-4a-c4     동적
  192.168.72.254         00-50-56-f8-3a-04     동적
  192.168.72.255         ff-ff-ff-ff-ff-ff     정적
  224.0.0.2              01-00-5e-00-00-02     정적
  224.0.0.22             01-00-5e-00-00-16     정적
```

netsh 명령어

네트워크 설정 변경 명령어

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



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: 192.168.255.1 --- 0x5
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.35.203 --- 0x11
  인터넷 주소           물리적 주소           유형
  192.168.35.1          00-23-aa-de-99-91     동적
  192.168.35.197        40-5b-d8-ad-4a-c3     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.72.1 --- 0x12
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
```

ARP Spoofing

```
인터페이스: 192.168.35.203 --- 0x11
  인터넷 주소           물리적 주소           유형
  192.168.35.1          00-23-aa-de-99-91     동적
  192.168.35.2          50-77-05-66-84-8f     동적
  192.168.35.3          00-24-d6-d1-61-7c     동적
  192.168.35.13         d0-94-66-f6-d5-ba     동적
  192.168.35.39         b8-5a-73-1b-42-25     동적
  192.168.35.105        f0-6e-0b-7c-cf-67     동적
  192.168.35.145        50-77-05-62-51-8f     동적
  192.168.35.171        64-7b-ce-c6-42-5c     동적
  192.168.35.197        00-0c-29-85-76-5c     동적
  192.168.35.255        ff-ff-ff-ff-ff-ff     정적
  224.0.0.2             01-00-5e-00-00-02     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  224.0.0.251           01-00-5e-00-00-fb     정적
  224.0.0.252           01-00-5e-00-00-fc     정적
  224.0.0.253           01-00-5e-00-00-fd     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
  255.255.255.255       ff-ff-ff-ff-ff-ff     정적

인터페이스: 192.168.72.1 --- 0x12
  인터넷 주소           물리적 주소           유형
  192.168.35.197        40-5b-d8-ad-4a-c4     정적
  192.168.72.254        00-50-5b-f8-3a-04     동적
  192.168.72.255        ff-ff-ff-ff-ff-ff     정적
  224.0.0.2             01-00-5e-00-00-02     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  224.0.0.251           01-00-5e-00-00-fb     정적
  224.0.0.252           01-00-5e-00-00-fc     정적
  224.0.0.253           01-00-5e-00-00-fd     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
```

```
C:₩WINDOWS₩system32>arp -a

인터페이스: 192.168.255.1 --- 0x5
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.35.203 --- 0x11
  인터넷 주소           물리적 주소           유형
  192.168.35.1          00-23-aa-de-99-91     동적
  192.168.35.197        40-5b-d8-ad-4a-c3     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.72.1 --- 0x12
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
```

K-XARP

Python

```
* [ 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]        #
#                                              #
################################################
```

Python

```
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
```

```
tIP = str(int(arp.tpa[0]))+"."+str(int(arp.tpa[1]))+"."+str(int(arp.tpa[2]))+"."+str(int(arp.tpa[3]))
#print(tIP)

if tIP in Request_List :
    print("") # 이미 존재하는 Request IP
else :
    Request_List.append(tIP)
```

Request

```
def Mac_Check
    if ARP_Tabl
        print(" *
    else :
        print("W
        print("#
        print("#
        print("#
        print("#
        print("#
        print("#      Attacker MAC is [ ',sMAC, ']!      # ")
        print("#                                                #")
        print("#                                                #")
        print("##########################################")
        print("#                                                #")
        print("#          ARP Table 정적으로 설정 [시작]          #")
        print("#                                                #")
        print("##########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```

Python

```
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_Tab
        #print(A


def Mac_Checl
    if ARP_Tabl
        print(" *
    else :
        print("₩
        print("#
        print("#
        print("#
        print("#
        print("#
        print("#      Attacker MAC is [',sMAC,']!       # ")
        print("#                                                #")
        print("#                                                #")
        print("##########################################")
        print("#                                                #")
        print("#          ARP Table 정적으로 설정 [시작]          #")
        print("#                                                #")
        print("##########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```

```
if sIP in Request_List :# request 리스트 안에 있는 지 sIP 조회 하는 문구
    Right_Reply(sIP,sMAC) # 올바른 reply패킷이므로 ARP_Table에 조회하는 함수로 넘어간다.


else :
    print("₩n!!!! request 리스트엔 없는 reply패킷 !!! ARP 스푸핑이 의심 됩니다 !!!!₩n")
    if sIP in ARP_Table :  # ARP_Table 에 Reply sender IP가 있는지 체크한다.
        Mac_Check(sIP,sMAC)  # Mac 주소가 중복 되었는지 체크한다.
```

Reply

Python

```
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)


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("#                                        #")
        print("#            !!!! ARP Spoofing Attack !!!!            #")
        print("#                                        #")
        print("#      Attacker MAC is [",sMAC,"]!      #")
        print("#                                        #")
        print("#                                        #")
        print("########################################")
        print("#                                        #")
        print("#          ARP Table 정적으로 설정 [시작]          #")
        print("#                                        #")
        print("########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```

Python

```
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] = s
        #print(ARP_Table)


def Mac_Check(sIP,sMAC
    if ARP_Table.get(sIP)
        print(" ******* 정상
    else :
        print("₩n#######
        print("#
        print("#
        print("#
        print("#
        print("#       Attack
        print("#                                                    #")
        print("#                                                    #")
        print("##########################################")
        print("#                                                    #")
        print("#          ARP Table 정적으로 설정 [시작]            #")
        print("#                                                    #")
        print("##########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```


```
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)
```

Python

```
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)


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("#                                        #")
        print("#            !!!! ARP Spoofing Attack !!!!            #")
        print("#                                        #")
        print("#      Attacker MAC is [",sMAC,"]!      #")
        print("#                                        #")
        print("#                                        #")
        print("########################################")
        print("#                                        #")
        print("#          ARP Table 정적으로 설정 [시작]          #")
        print("#                                        #")
        print("########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```

Python

```
Request_List = []
ARP_Table={}

def Right_Reply(sIP,sMAC): # ARP 테이블
    if sIP in ARP_Table :  # ip가 ARP_table
        Mac_Check(sIP,sMAC)
    else :
        ARP_Table[sIP] = sMAC
        #print(ARP_Table)


def Mac_Check(sIP,sMAC):  # Mac 주소
    if ARP_Table.get(sIP) == sMAC :
        print(" ******* 정상적인 응답(Reply)
    else :
        print("₩n##################
        print("#
        print("#
        print("#             !!!! ARP Spoofing
        print("#
        print("#      Attacker MAC is [",sM
        print("#
        print("#
        print("####################
        print("#
        print("#          ARP Table 정적으로
        print("#
        print("####################
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
```

```
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("#                                      #")
        print("#            !!!! ARP Spoofing Attack !!!!            #")
        print("#                                      #")
        print("#      Attacker MAC is [",sMAC,"]!      #")
        print("#                                      #")
        print("#                                      #")
        print("########################################")
        print("#                                      #")
        print("#          ARP Table 정적으로 설정 [시작]          #")
        print("#                                      #")
        print("########################################₩n₩n")
        Right_Mac = ARP_Table.get(sIP)
        Static_Setting(sIP,Right_Mac)
        # ARP reply 패킷을 다시 보내서 다른 맥주소가 오는지 확인 해볼
        # static으로 설정하는 부분이 있어야함.
```

Python

```
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())
```

Python

```
######## ARP Table 정적으로 설정 [완료] ########


인터페이스: 192.168.255.1 --- 0x5
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.35.203 --- 0x11
  인터넷 주소           물리적 주소           유형
  192.168.35.1          00-23-aa-de-99-91     동적
  192.168.35.197        40-5b-d8-ad-4a-c3     정적
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적

인터페이스: 192.168.72.1 --- 0x12
  인터넷 주소           물리적 주소           유형
  224.0.0.22            01-00-5e-00-00-16     정적
  239.255.255.250       01-00-5e-7f-ff-fa     정적
```

Q & A

Thank you ☺