R E P O R T

[컴퓨터 네트워크]

학 과	컴퓨터공학부 컴퓨터공학전공
교수님	서경룡 교수님
학 번	201911608
이 름	김지환
제출일	2023.10.10

부경대학교

INDEX

I. Ubuntu 환경 구축

II. HTTP 서버 설치

III. SMTP 서버 설치

IV. DNS 적용

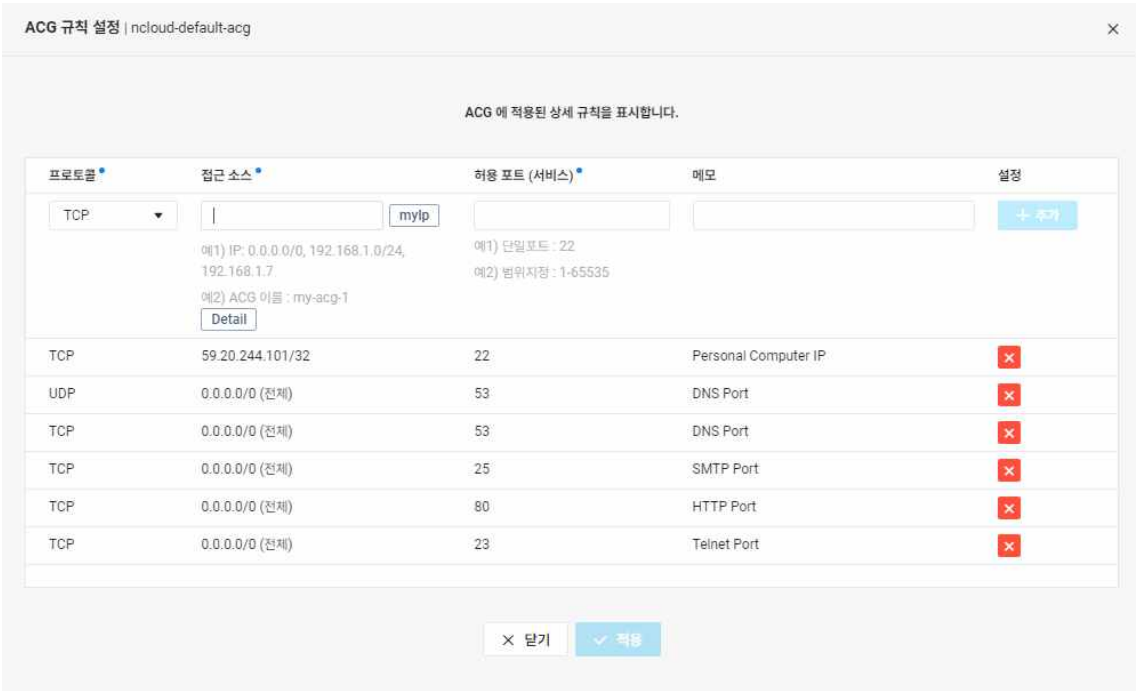
V. 마무리

I. Ubuntu 환경 구축

1. Naver Cloud

OS - Ubuntu 18.04 LTS
PlatForm - Naver Cloud
1년간 무료로 사용할 수 있는 서비스입니다.
서버를 백그라운드에서도 실행하고 싶어서 클라우드를 사용하게 되었습니다.

2. ACG 규칙 설정

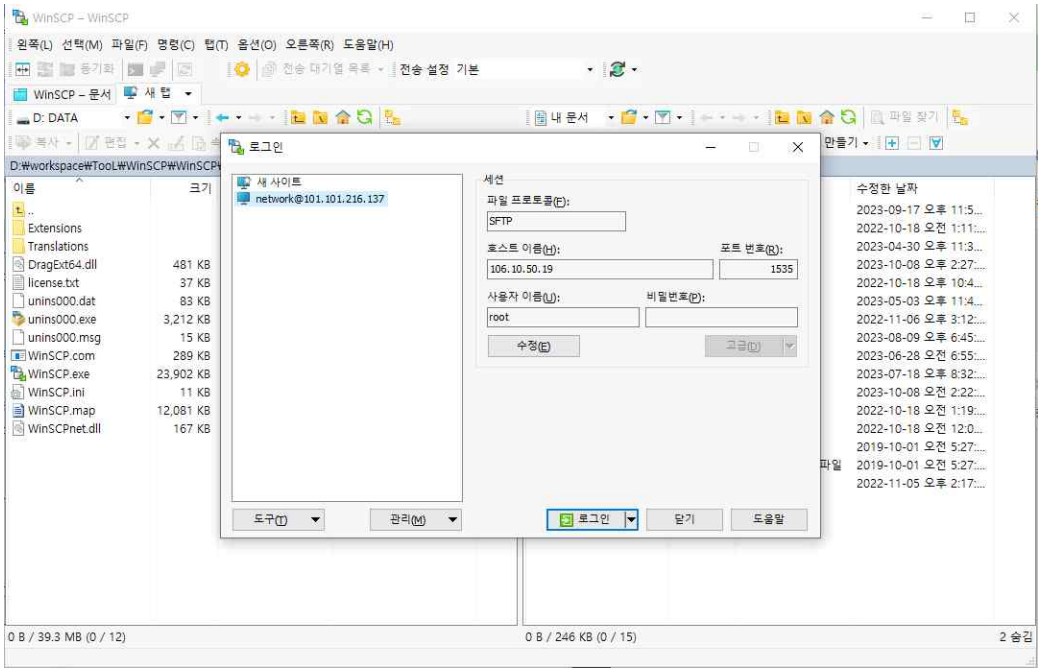


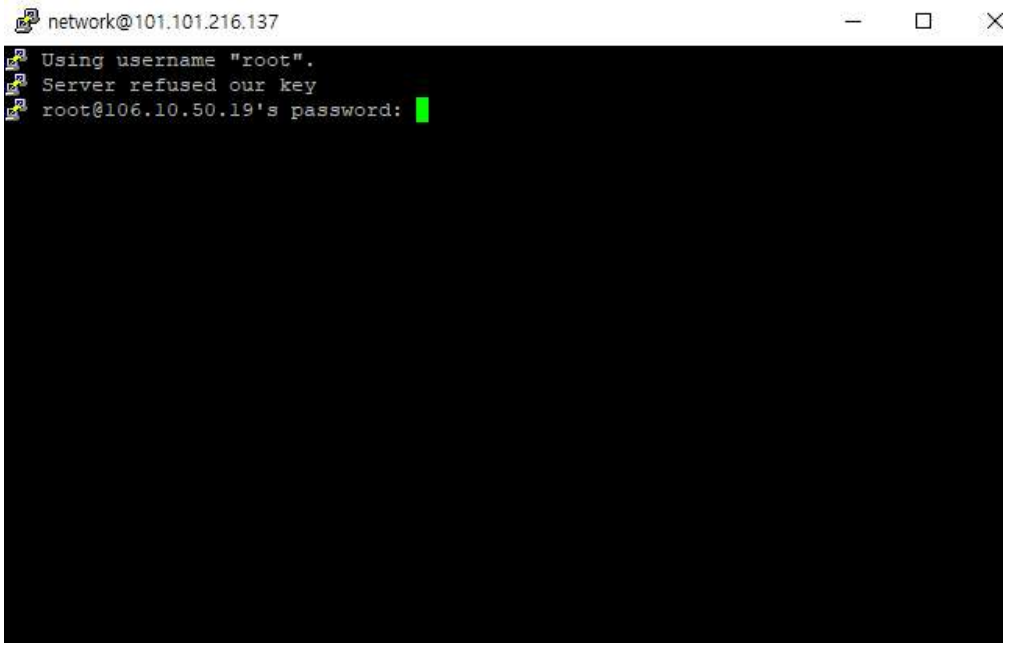
ACG(Access Control Group)을 사용해서 우분투 내 uwf 명령어 대신 간편하게 방화벽을 관리했습니다.

허용 포트 : http(tcp), smtp(tcp), dns(udp/tcp), telnet(tcp), ssh(tcp)

각 서버들을 설치하고 클라이언트에서 요청할 수 있도록 허용합니다.

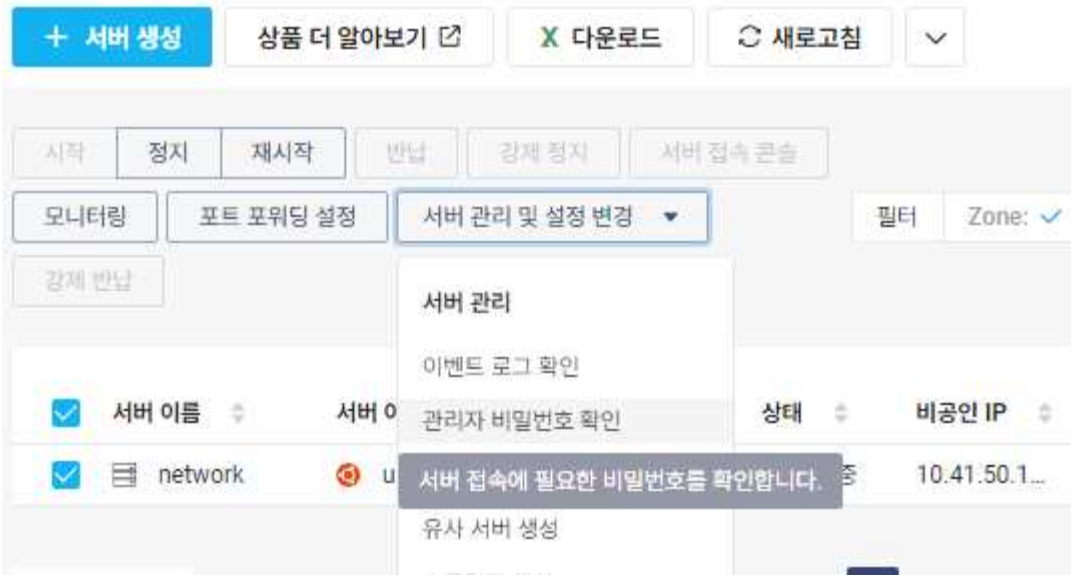
3. WinSCP



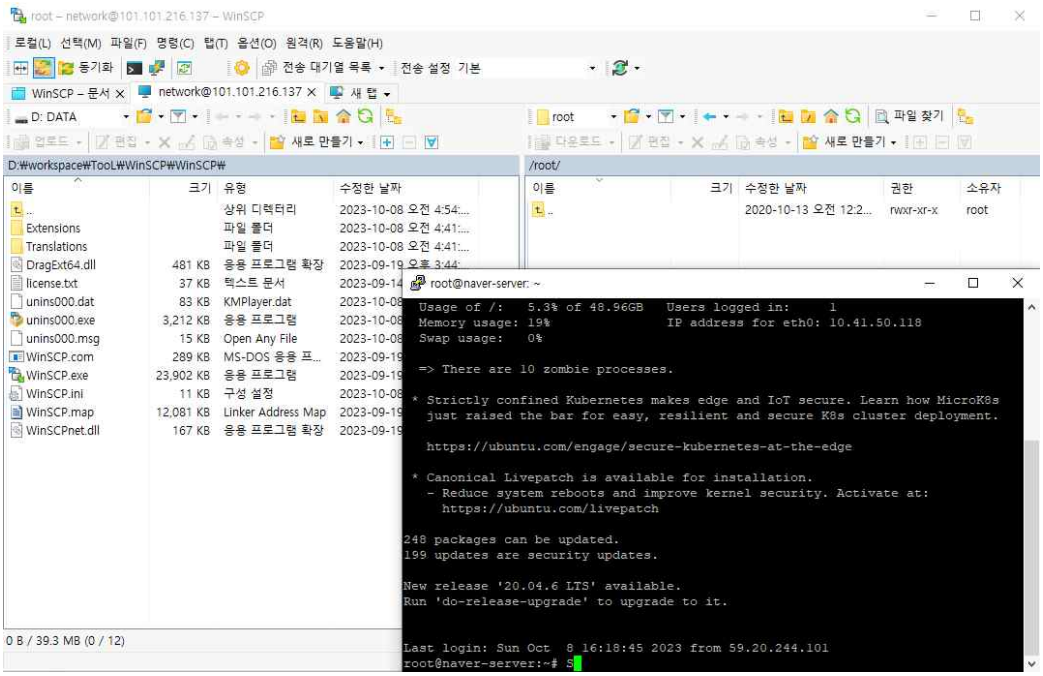


WinSCP는 PuTTY (원격 SSH 접속)과 로컬에서 서버 디렉토리를 관리할 수 있는 툴입니다.
서버 파일을 손 쉽게 컨트롤 할 수 있는 장점으로 사용하게 되었습니다.
PuTTY로 NaverCloud에서 포트포워딩한 서버 접속용 공인 IP와 Naver Cloud에서 설정한 관리자 비밀번호로 Ubuntu OS 접속합니다.
Naver Cloud에서 설정한 관리자 비밀번호는 Ubuntu Login에서 설명

4. Ubuntu Login



[사진 1]



[사진 2]

NaverCloud에서 발급받은 관리자 비밀번호로 우분투 환경에 로그인합니다.
[사진 1]에서는 미리 발급받은 상태라서 관리자 비밀번호 확인이 식별됩니다.
처음 Cloud에서 OS설치 시 해당 메뉴 콤보박스 하단에 초기 관리자 비밀번호 설정을 할 수 있도록 되어있습니다.
root 계정에 Login하면 WinSCP에서 /root 디렉토리로 Ubuntu 서버에 성공적으로 로그인 했음을 확인할 수 있습니다.

II. HTTP 서버 설치

1. Apache Tomcat 설치

- Ubuntu OS Terminal

$ sudo apt install apache2

➜ Ubuntu OS에서는 apache2로 install합니다. ... 설치 완료

$ netstat -tuln

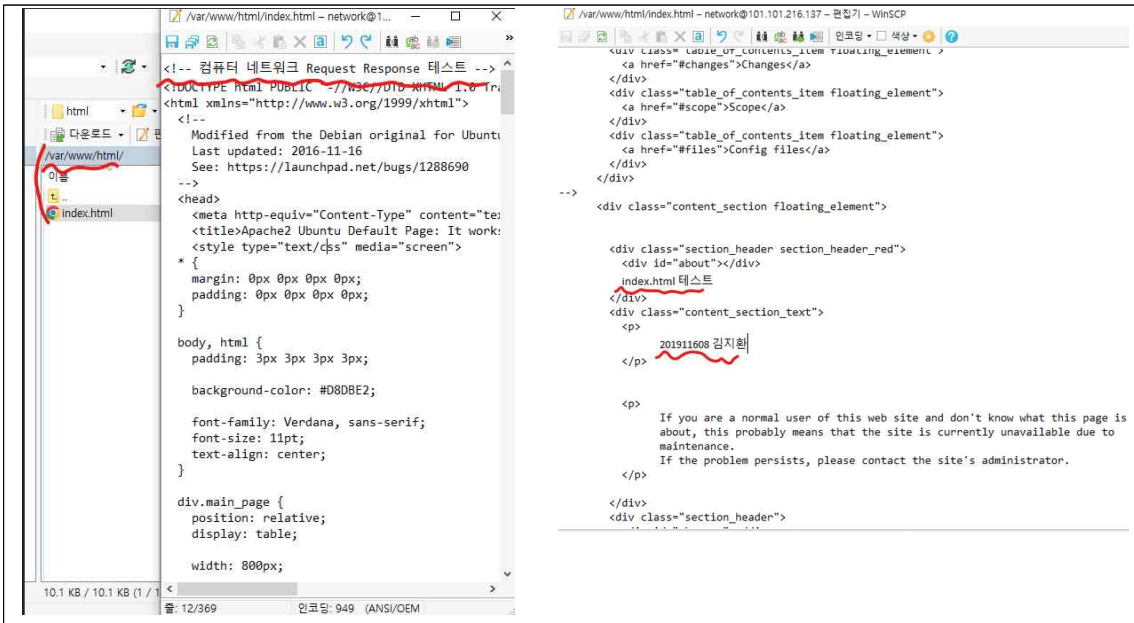
```
root@naver-server:/var/www/html# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
```

➜ -t : tcp, -u : udp, -l ; linstening port, -n : host:port

➜ port : 22(ssh, 접속), 53(DNS 쿼리 처리용), 80(http)

➜ Cloud상 http 서버에 필요한 기본적인 포트가 모두 활성화 되어있는 것을 확인했습니다.

2. index.html



➜ apache web server가 제대로 설치되어 Ubuntu OS의 /var/www/html/ 경로에 index.html이 작성되었습니다.

➜ Server가 제대로 동작되고 있는지 Telnet과 같은 Request/Response로 확인합니다.

➜ 서버의 요청/응답을 확인하기 전 해당 서버에 내가 설정한 index.html이 제대로 동작하는지 확인하기 위해 내가 작성한 index.html이라고 알아볼 수 있게 수정합니다.

3. Client Test - Local

▪ Telnet Install

$sudo apt install telnet telnetd

➜ Ubuntu OS에서 telnet과 telnet demon을 설치합니다.

➜ 설치완료

$netstat -tuln

```
root@naver-server:~# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
```

➜ 1. apache tomcat 설치 할 때와 다르게 23번 포트도 허용된 것을 확인했습니다.

➜ 성공적으로 telnet을 설치했습니다.

▪ Local Test

$sudo telnet localhost 80

```
root@naver-server:/# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.1
host: localhost

HTTP/1.1 200 OK
Date: Sun, 08 Oct 2023 09:36:55 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Sun, 08 Oct 2023 09:35:25 GMT
ETag: "28bc-6073131787ac2"
Accept-Ranges: bytes
Content-Length: 10428
Vary: Accept-Encoding
Content-Type: text/html; charset=utf-8

<!--컴퓨터 네트워크 Request Response 테스트 -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.or
g/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

index

➜ localhost 80 포트로 접속해서 통신합니다.

➜ GET /index.html HTTP/1.1 명령으로 http프로토콜로 index.html을 GET합니다.

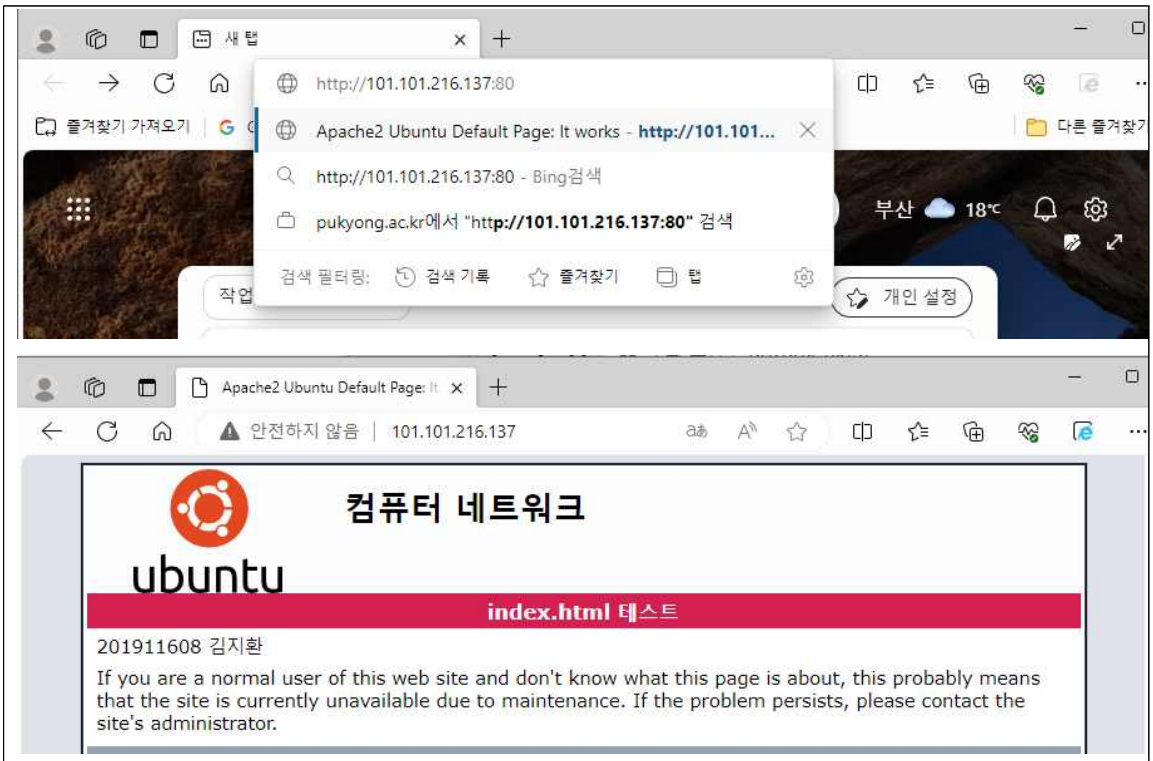
➜ 호스트는 localhost로 작성

➜ 응답이 200 OK로 Success인 것을 확인하고 작성한 index.html이 맞는지 확인합니다.

➜ response 받은 html파일의 comment로 우리가 작성한 내용이 맞으므로 성공적으로 http 서버가 설치되었습니다.

4. Client Test - Global

- http://101.101.216.137:80/



➜ Ubuntu HTTP Server에서 tomcat으로 /var/www/html/index.html을 배포하고 있으므로 ipv4:port로 접속 시 우리가 작성한 index.html을 볼 수 있습니다.

➜ global client에서도 설치한 HTTP Server가 제대로 동작합니다.

5. WireShark

- **tshark Install**

$sudo apt tshark

➜ Naver Cloud 1년 무료버전은 GUI 환경이 지원되지 않고 Cloud상의 Server라서 Wireshark로 확인할 수 없는 이슈가 발생했습니다.
(해당 이슈에 대한 내용은 마지막에 정리하겠습니다.)

➜ tshark는 WireShark의 CLI 버전으로 Command Line을 통해서 Packet을 분석할 수 있는 툴입니다. tshark로 Packet을 캡처 후 WireShark에서 분석하겠습니다.

➜ tshark가 설치되었습니다.

➜ tshark 실행 법 $sudo tshark -i [network interface] -f “port [protocol]”

▪ network interface 확인

$ ip link show

```
root@jihwan:~# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
 group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
 DEFAULT group default qlen 1000
    link/ether f2:20:cd:80:38:81 brd ff:ff:ff:ff:ff:ff
```

➜ lo는 loopback 이므로 network interface로 eth0을 사용하면 됩니다. (우분투 기본)

▪ Test

➜ $ sudo tshark -i eth0 -f "port 80"

```
    1 0.000000000 59.20.244.101 → 10.41.50.118 TCP 66 5163 → 25 [SYN] Seq=0 Win=
64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
    2 0.000044719 10.41.50.118 → 59.20.244.101 TCP 66 25 → 5163 [SYN, ACK] Seq=0
 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
    3 0.020162267 59.20.244.101 → 10.41.50.118 TCP 60 5163 → 25 [ACK] Seq=1 Ack=
1 Win=131328 Len=0
    4 0.031665795 10.41.50.118 → 59.20.244.101 SMTP 93 S: 220 jihwan.com ESMTP P
ostfix (Ubuntu)
    5 0.073968094 59.20.244.101 → 10.41.50.118 SMTP 75 C: EHLO DESKTOPJ4EB3BD
    6 0.073982731 10.41.50.118 → 59.20.244.101 TCP 54 25 → 5163 [ACK] Seq=40 Ack
=22 Win=64256 Len=0
    7 0.074070780 10.41.50.118 → 59.20.244.101 SMTP 231 S: 250-jihwan.com | 250-
PIPELINING | 250-SIZE 10240000 | 250-VRFY | 250-ETRN | 250-AUTH PLAIN LOGIN | 25
0-AUTH=PLAIN LOGIN | 250-ENHANCEDSTATUSCODES | 250-8BITMIME | 250-DSN | 250 SMTP
UTF8
    8 0.084325737 59.20.244.101 → 10.41.50.118 SMTP 66 C: AUTH LOGIN
    9 0.084461601 10.41.50.118 → 59.20.244.101 SMTP 72 S: 334 VXNlcm5hbWU6
   10 0.094641661 59.20.244.101 → 10.41.50.118 SMTP 80 C: User: bmV0d29ya0BqaWh3
YW4uY29t
   11 0.094724249 10.41.50.118 → 59.20.244.101 SMTP 72 S: 334 UGFzc3dvcmQ6
   12 0.105910382 59.20.244.101 → 10.41.50.118 SMTP 64 C: Pass: MTUzNQ==
```

➜ Test시 위와 같이 Packet이 오고 감을 볼 수 있습니다.

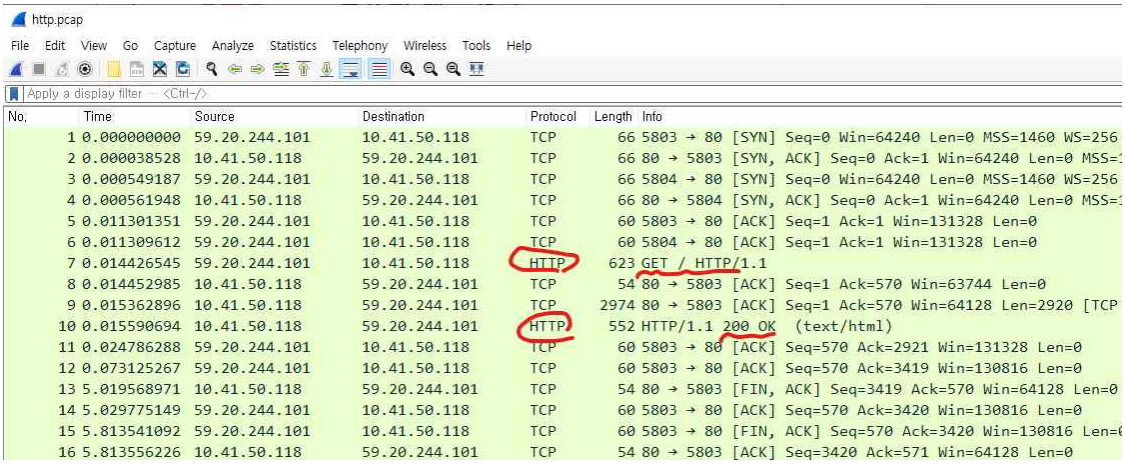
➜ WireShark와 동일하며 해당 기능을 이용해 통신하는 Packet들을 capture해서 WireShark에서 Open할 수 있습니다.

➜ 캡처하는 명령어는 $ sudo tshark -i eth0 -f "port 80" 뒤에 -w [filename].pcap

▪ Packet Capture

```
$ sudo tshark -i eth0 -f "port 80" -w http.pcap
```

- ➜ HTTP 서버에서 packet을 캡처하고 http.pcap이라는 파일로 저장합니다.
- ➜ 명령을 실행 후 웹 브라우저에서 HTTP 서버로 접근합니다.
- ➜ HTTP 서버에 접근 시 패킷 전송량만큼 숫자가 변합니다.
- ➜ 패킷 통신의 녹화를 중단하고 싶으면 Ctrl + c를 눌러서 종료하면 됩니다.
- ➜ 녹화 중단시 녹화 중 송, 수신 한 패킷들의 내역이 http.pcap로 저장됩니다.
- ➜ http.pcap의 저장 경로는 해당 명령을 실행한 디렉토리 위치에서 저장됩니다.



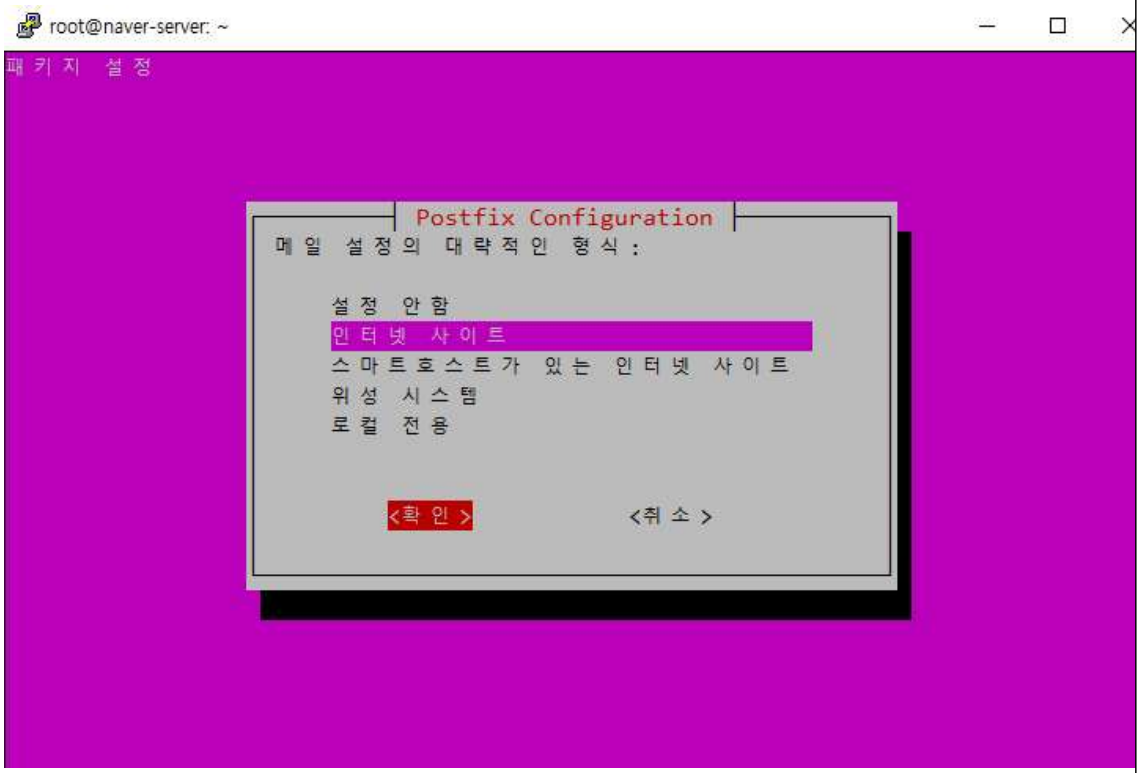
- ➜ TCP로 패킷을 송수신하며 서버와 클라이언트가 연결합니다.
- ➜ 연결 후 클라이언트에서 GET Method를 Request합니다.
- ➜ 서버가 TCP로 요청받은 내용을 처리 중 에러가 발생하지 않으면 200 OK를 Response합니다. 하지만 에러가 발생시 300대, 400대 등의 Status를 Response합니다.
- ➜ tShark에서 캡처한 내용을 WireShark에서 분석했습니다.

III. SMTP 서버 설치

1. Postfix 설치

$ sudo apt install postfix

➜ smtp 서버를 설치하기 위해 postfix를 사용합니다. sendmail보다 쉬운 설치, 보안, 송수신 속도 등의 장점과 2021년부터 적극적으로 개발되고 있는 postfix를 채택 했습니다.



➜ Postfix는 GUI 형태로 설치할 수 있습니다.

➜ SMTP 서버로 메일 송, 수신을 테스트하기 위해서 인터넷 사이트로 설정하겠습니다.

➜ Naver Cloud는 이후 뒤에 GUI는 제공되지 않습니다.

2. POP3 설치

$ sudo apt install dovecot-core dovecot-imapd dovecot-pop3d

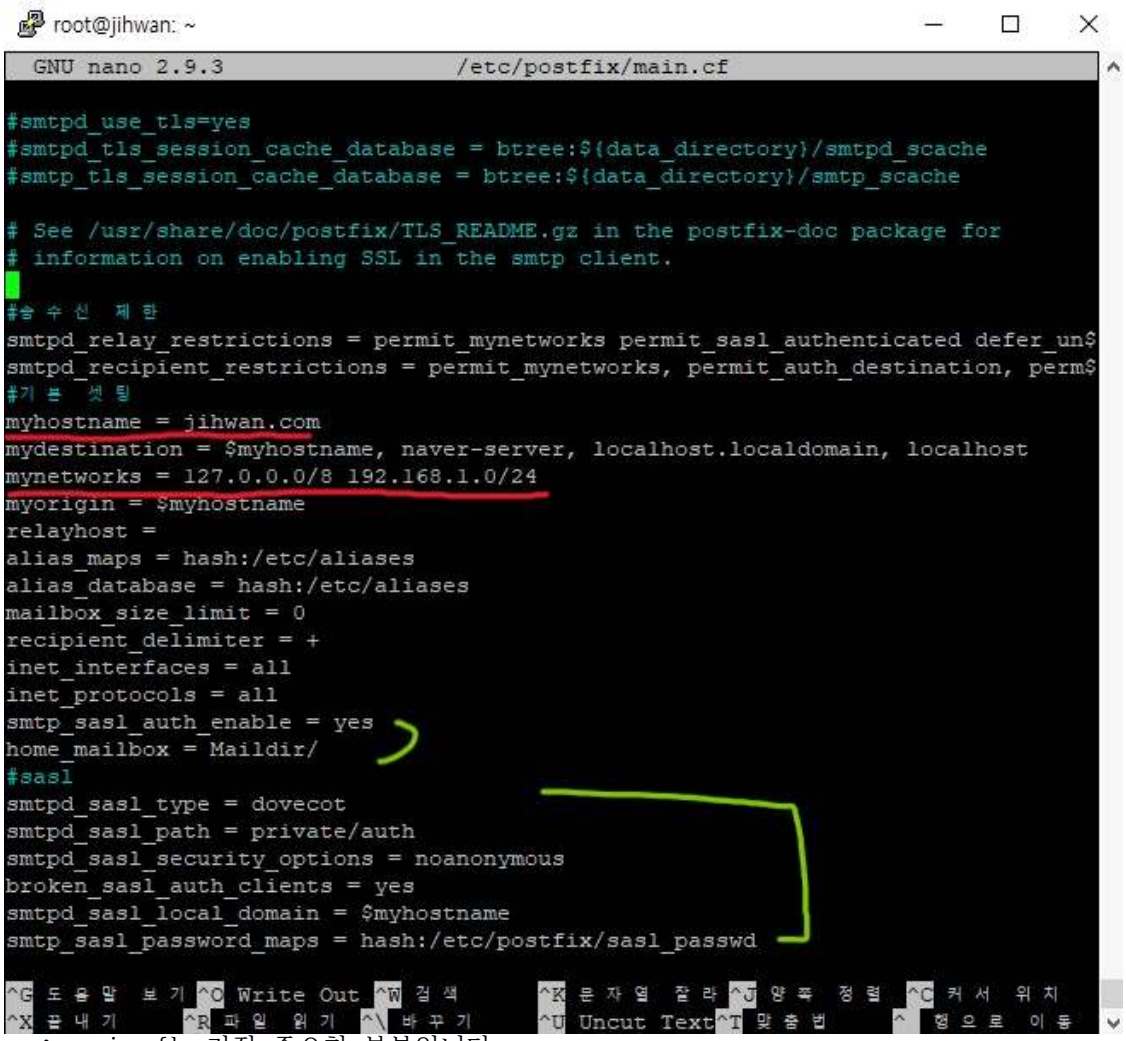
➜ 메일 송, 수신 시 OutLook을 이용해서 테스트하기 위해 POP3 서버도 설치합니다.

➜ dovecot는 이메일 관리에 유용한 패키지입니다. imap은 필요한 라이브러리가 있을 수 있으므로 같이 다운로드 하겠습니다.

3. SMTP 서버 셋팅

- /etc/postfic/main.cf

$ vim/etc/postfix/main.cf



➜ main.cf는 가장 중요한 부분입니다.

➜ 빨간색은 수정할 부분으로,
hostname - ubuntu hostname에 맞게
mynetworks - 네트워크 대역을 설정합니다. 로컬 네트워크를 사용하므로 127.0.0.0/8 192.168.1.0/24로 설정합니다.
relayhost - 중개 서버역할이므로 google등 서버를 사용하지 않으면 비워야합니다.

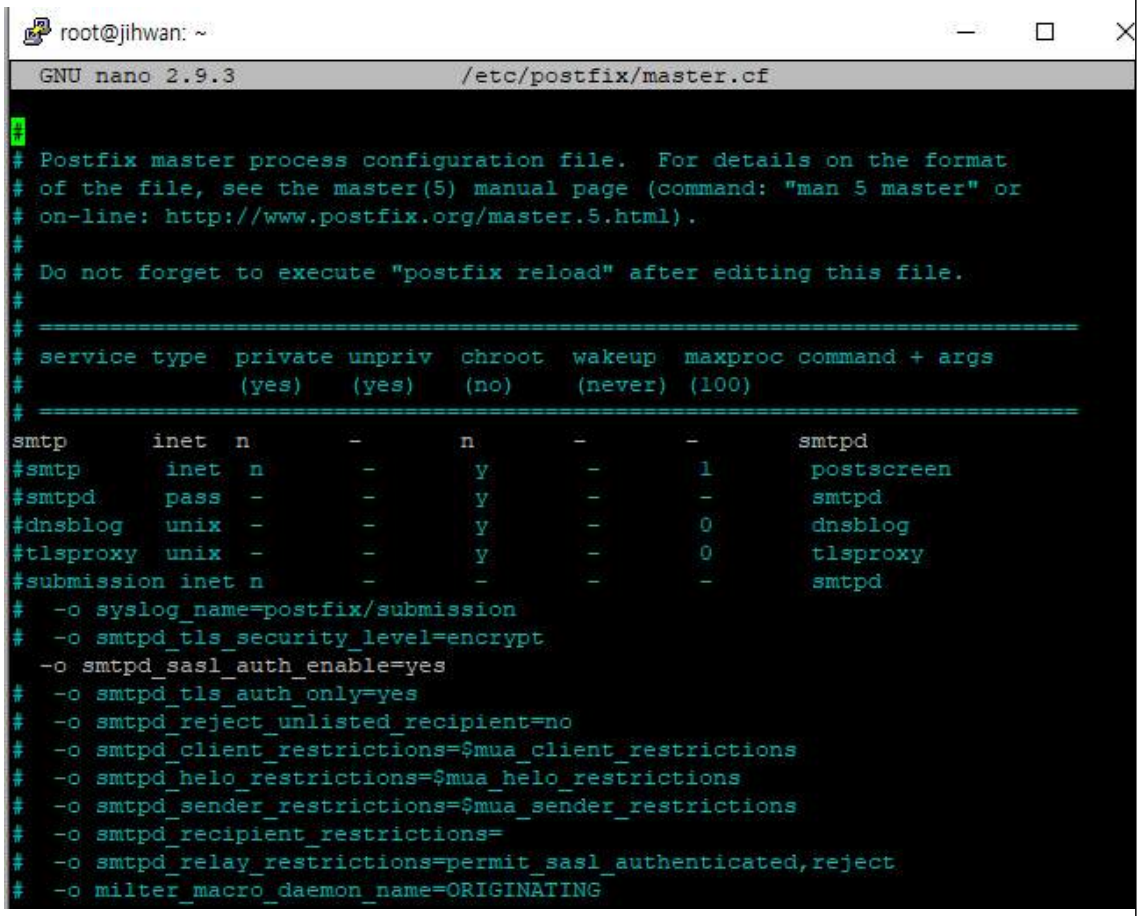
➜ 초록색은 추가할 부분으로 sasl 인증을 위해 작성합니다.

➜ Postfix는 송, 수신 간 SASL 인증으로 클라이언트와 연결하고 DoveCot과 연결되므로 SASL 설정을 합니다.

➜ Type은 dovecot로 smpt_ssal_password_maps로 사용자 인증을 위한 작업을 하는게 중요합니다.

▪ etc/postfix/master.cf

$ vim /etc/postfix/master.cf



➜ smtp의 chroot 때문에 보안 설정 등 번거로워질 수 있으니 n으로 수정합니다.

➜ -o smtpd_sasl_auth_enable=yes의 주석을 해제해서 sasl 인증을 허용합니다.

▪ POP3 및 서버 상태 확인

➜ SMTP 서버 설치에 대한 과제이므로 POP3 셋팅은 생략하겠습니다.

```
root@jihwan:~# netstat -tlun
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
```

➜ 25(smtp), 110(pop3), 143(imap) 포트가 허용된 것을 확인했습니다.

➜ smtp와 pop3 서버가 제대로 설치되었습니다.

4. Client Test - Local

$ telnet localhost 25

➜ MAIL FROM에서 @를 생략 시 Local Client라서 hostname이 자동으로 등록됩니다.

➜ aa1535@부경대 제 메일로 메일을 보내어 SMTP 서버를 확인하겠습니다.

➜ 제 구글 메일로 스팸 처리되어 전달되었습니다.

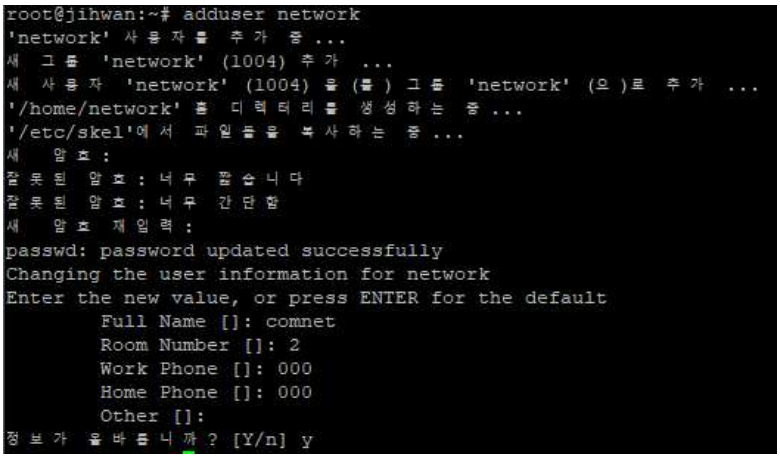
➜ DNS를 등록하지도 않았고 허가받은 DNS가 아니라서 스팸 처리되었습니다.

➜ SMTP 서버가 제대로 설치되었습니다.

5. Client Test - Global

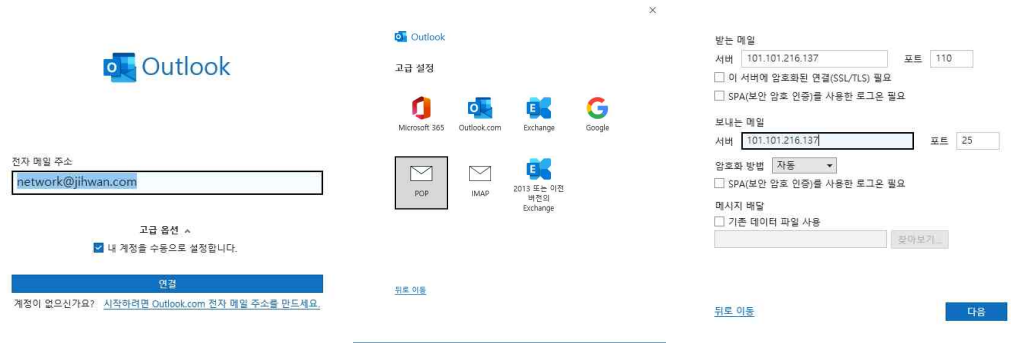
- OutLook

$ adduser <userID>



➜ OutLook에서 사용할 mail user를 생성합니다.

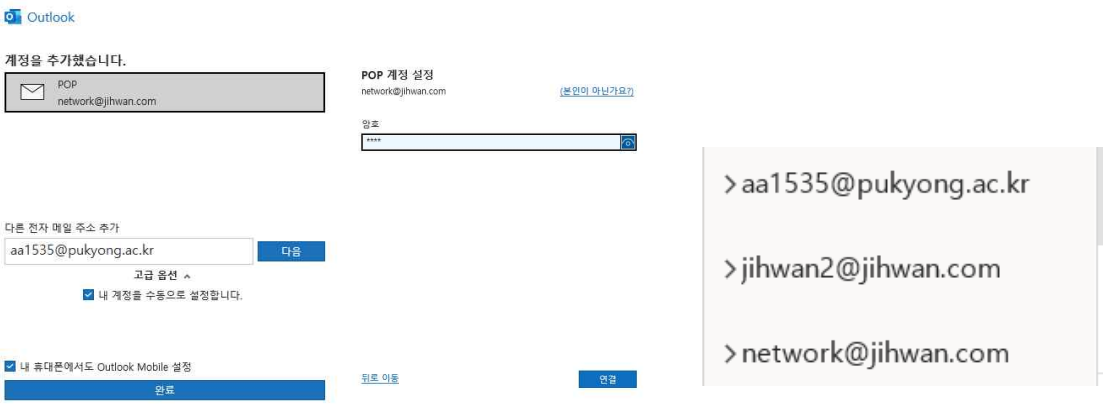
➜ 암호 입력 후 상세 내용을 작성 후 저장합니다.



➜ 생성한 <userID>@호스트 또는 도메인명을 작성 후 POP으로 설정합니다.

➜ 받는 메일과 보내는 메일 모두 서버 공인 IP로 작성 후 알맞은 포트를 작성합니다.

➜ 받는 메일(POP3) - Port : 110, 보내는 메일 (SMTP) - Port : 25



➜ OutLook 계정이 제대로 생성되었습니다.

▪ Global Test

TOOL : OutLook
Protocol : SMTP
Port : 25
Body : Test Mail

➜ OutLook에 SMTP 서버에 연결해서 생성한 계정으로 제 메일 주소로 메일을 전송하여 테스트 해보겠습니다.

➜ Global Client에서도 SMTP 서버 동작이 잘되었음을 확인했습니다.

6. WireShark

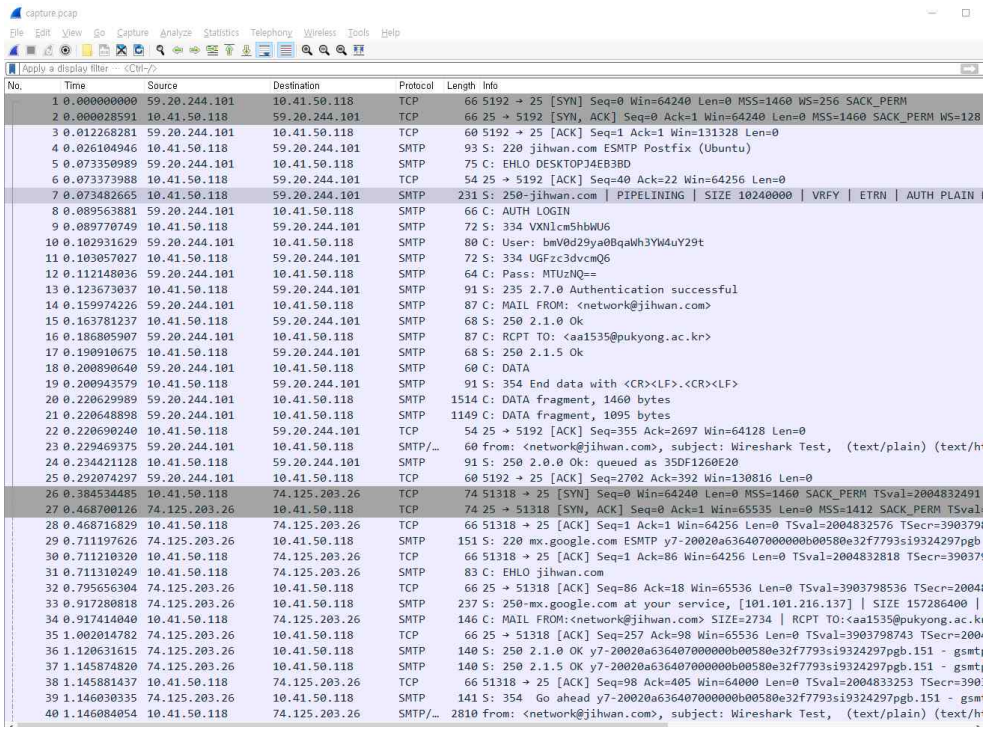
$ sudo tshark -i eth0 -f “port 25” -w smtp.pcap

```
root@jihwan:~# sudo tshark -i eth0 -f "port 25" -w capture.pcap
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
96 ^C
root@jihwan:~# ls
ls: 'thinclient_drives'에 접근할 수 없습니다: 전송 종료지점이 연결되어 있지 않습
니다
CookieRunFont_TTF       __MACOSX      server.key          thinclient_drives
CookieRunFont_TTF.zip   capture.pcap  server.key.secure
```

➜ Ubuntu OS Terminal에서 tshark를 실행하고 패킷의 송수신을 녹화합니다.

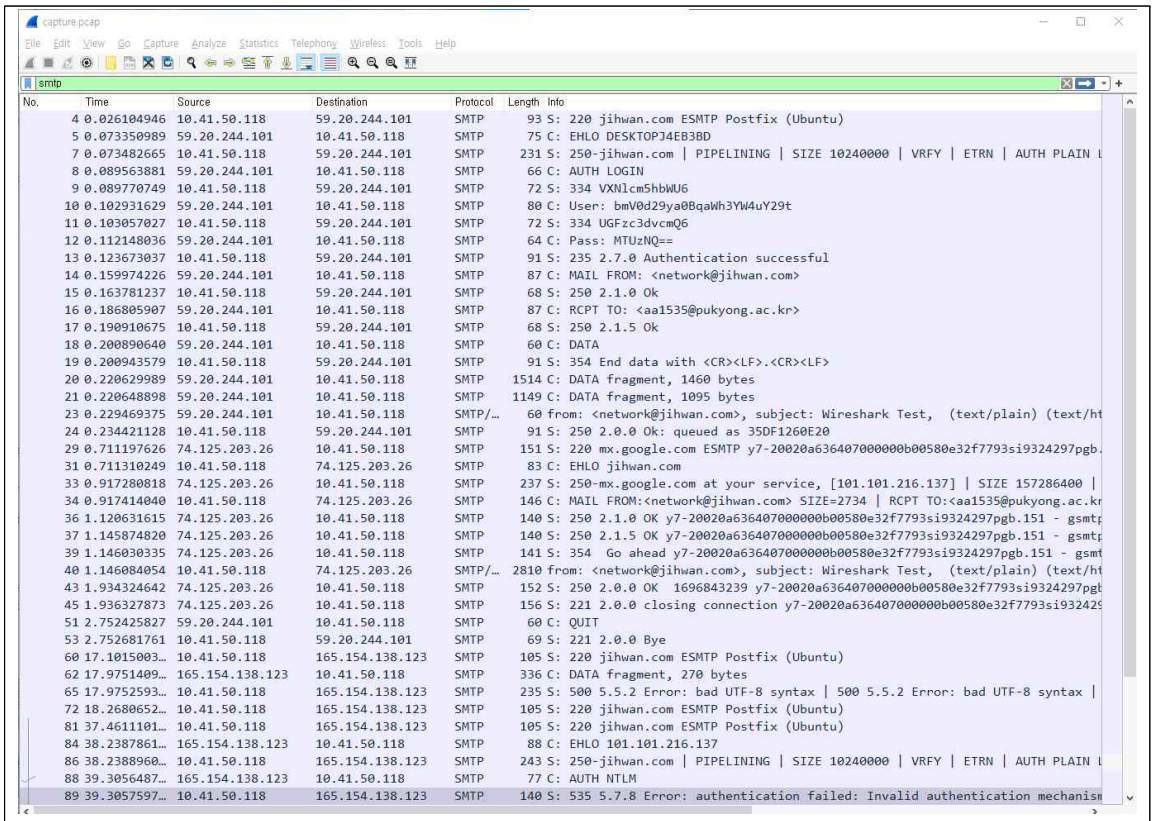
➜ 녹화하는 동안 SMTP 서버를 사용해 mail을 발신합니다.

➜ mail 발신 동안 총 96개의 패킷이 송수신했고 tshark를 종료하면서 capture.pcap 파일을 저장합니다. 해당 pcap 파일을 WireShark에서 Open 하겠습니다.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	59.20.244.101	10.41.50.118	TCP	66	5192 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.000028591	10.41.50.118	59.20.244.101	TCP	66	25 → 5192 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.012268281	59.20.244.101	10.41.50.118	TCP	60	5192 → 25 [ACK] Seq=1 Ack=1 Win=131328 Len=0
4	0.026104946	10.41.50.118	59.20.244.101	SMTP	93	S: 220 jihwan.com ESMTP Postfix (Ubuntu)
5	0.073350989	59.20.244.101	10.41.50.118	SMTP	75	C: EHLO DESKTOPJ4EB3BD
6	0.073373988	10.41.50.118	59.20.244.101	TCP	54	25 → 5192 [ACK] Seq=40 Ack=22 Win=64256 Len=0
7	0.073482665	10.41.50.118	59.20.244.101	SMTP	231	S: 250-jihwan.com | PIPELINING | SIZE 10240000 | VRFY | ETRN | AUTH PLAIN
8	0.089563881	59.20.244.101	10.41.50.118	SMTP	66	C: AUTH LOGIN
9	0.089770749	10.41.50.118	59.20.244.101	SMTP	72	S: 334 VXNlcm5hbWU6
10	0.102931629	59.20.244.101	10.41.50.118	SMTP	80	C: User: bmV0d29ya0BqaWh3YW4uY29t
11	0.103057027	10.41.50.118	59.20.244.101	SMTP	72	S: 334 UGFzc3dvcmQ6
12	0.112148036	59.20.244.101	10.41.50.118	SMTP	64	C: Pass: MTUzNQ==
13	0.123673037	10.41.50.118	59.20.244.101	SMTP	91	S: 235 2.7.0 Authentication successful
14	0.159974226	59.20.244.101	10.41.50.118	SMTP	87	C: MAIL FROM: <network@jihwan.com>
15	0.163781237	10.41.50.118	59.20.244.101	SMTP	68	S: 250 2.1.0 Ok
16	0.186805907	59.20.244.101	10.41.50.118	SMTP	87	C: RCPT TO: <aa1535@pukyong.ac.kr>
17	0.190910675	10.41.50.118	59.20.244.101	SMTP	68	S: 250 2.1.5 Ok
18	0.200890640	59.20.244.101	10.41.50.118	SMTP	60	C: DATA
19	0.200943579	10.41.50.118	59.20.244.101	SMTP	91	S: 354 End data with <CR><LF>.<CR><LF>
20	0.220629989	59.20.244.101	10.41.50.118	SMTP	1514	C: DATA fragment, 1460 bytes
21	0.220648898	59.20.244.101	10.41.50.118	SMTP	1149	C: DATA fragment, 1095 bytes
22	0.220690240	10.41.50.118	59.20.244.101	TCP	54	25 → 5192 [ACK] Seq=355 Ack=2697 Win=64128 Len=0
23	0.229469375	59.20.244.101	10.41.50.118	SMTP/…	60	from: <network@jihwan.com>, subject: Wireshark Test, (text/plain) (text/h
24	0.234421128	10.41.50.118	59.20.244.101	SMTP	91	S: 250 2.0.0 Ok: queued as 35DF1260E20
25	0.292074297	59.20.244.101	10.41.50.118	TCP	60	5192 → 25 [ACK] Seq=2702 Ack=392 Win=130816 Len=0
26	0.384534485	10.41.50.118	74.125.203.26	TCP	74	51318 → 25 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2004832491
27	0.468700126	74.125.203.26	10.41.50.118	TCP	74	25 → 51318 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM TSval
28	0.468716829	10.41.50.118	74.125.203.26	TCP	66	51318 → 25 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2004832576 TSecr=390379
29	0.711197626	74.125.203.26	10.41.50.118	SMTP	151	S: 220 mx.google.com ESMTP y7-20020a636407000000b00580e32f7793si9324297pgb
30	0.711210320	10.41.50.118	74.125.203.26	TCP	66	51318 → 25 [ACK] Seq=1 Ack=86 Win=64256 Len=0 TSval=2004832818 TSecr=39037
31	0.711310249	10.41.50.118	74.125.203.26	SMTP	83	C: EHLO jihwan.com
32	0.795656304	74.125.203.26	10.41.50.118	TCP	66	25 → 51318 [ACK] Seq=86 Ack=18 Win=65536 Len=0 TSval=3903798536 TSecr=2004
33	0.917280818	74.125.203.26	10.41.50.118	SMTP	237	S: 250-mx.google.com at your service, [101.101.216.137] | SIZE 157286400 |
34	0.917414040	10.41.50.118	74.125.203.26	SMTP	146	C: MAIL FROM:<network@jihwan.com> SIZE=2734 | RCPT TO:<aa1535@pukyong.ac.k
35	1.002014782	74.125.203.26	10.41.50.118	TCP	66	25 → 51318 [ACK] Seq=257 Ack=98 Win=65536 Len=0 TSval=3903798743 TSecr=200
36	1.120631615	74.125.203.26	10.41.50.118	SMTP	140	S: 250 2.1.0 OK y7-20020a636407000000b00580e32f7793si9324297pgb.151 - gsmt
37	1.145874820	74.125.203.26	10.41.50.118	SMTP	140	S: 250 2.1.5 OK y7-20020a636407000000b00580e32f7793si9324297pgb.151 - gsmt
38	1.145881437	10.41.50.118	74.125.203.26	TCP	66	51318 → 25 [ACK] Seq=98 Ack=405 Win=64000 Len=0 TSval=2004833253 TSecr=390
39	1.146030335	74.125.203.26	10.41.50.118	SMTP	141	S: 354 Go ahead y7-20020a636407000000b00580e32f7793si9324297pgb.151 - gsm
40	1.146084054	10.41.50.118	74.125.203.26	SMTP/…	2810	from: <network@jihwan.com>, subject: Wireshark Test, (text/plain) (text/h

[사진1 - WireShark]



[사진2 – WireShark]

➜ [사진1] 에서는 TCP의 결과도 출력하고 있습니다.
TCP의 내용은 먼저 클라이언트와 서버와 연결하려고 패킷을 송수신하고 있습니다.
메일이 발신되자 SMTP를 처리하고 SMTP 처리 후 발신지가 구글이므로 구글 서버에 연결해서 메일을 발신하고 있습니다.

➜ SMTP에서는 TELNET에서 수행하던 과정이 패킷에 저장되어 그대로 송수신되고 있습니다. 발신지인 구글로 메일 발신과 함께 QUIT와 221 2.0.0 Bye를 응답받는 것을 확인할 수 있습니다.

IV. DNS 서버 설치

1. bind9 install

$ sudo apt install bind9

➜ 로컬네트워크를 구축하기 위해 우분투의 IP에 DNS를 적용해서 확인합니다.

➜ 오픈소스 DNS 패키지로 DNS 적용을 하기 위해 꼭 설치해야합니다.

➜ 설치가 완료되었습니다.

$ netstat -tuln

```
root@jihwan:~# netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:110             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:143             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 10.41.50.118:53         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
```

➜ 기존 DNS 쿼리를 처리하는 127.0.0.53 외에도 53포트를 가진 ipv4가 두 개 더 추가되었습니다. 해당 ip들도 bind9 적용으로 DNS 쿼리를 처리할 수 있게 됐습니다.

➜ 953번 포트는 Bind에서 제공하는 RNDC(Remote Name Daemon Control) 서비스를 사용할 수 있도록 하는 포트입니다. DNS 서버를 위해 bind설치 시 사용하는 포트입니다.

➜ Bind9을 사용하기 위한 포트들이 제대로 실행되고 있습니다.

2. bind9 Setting

$ vim /etc/bind/named.conf.local

```
 GNU nano 2.9.3                /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not u
// organization
//include "/etc/bind/zones.rfc1918";

zone "jihwan.com" {
        type master;
        file "/etc/bind/db.jihwan.com";
};
```

➜ ssh 환경이라 보기 좋게 nano editor를 사용했습니다.

➜ local domain name에 대한 zone을 설정합니다.

➜ 종속되어 있지 않은 도메인이므로 master type으로 사용합니다.

$ vim /etc/bind/db.jihwan.com

```
 GNU nano 2.9.3                    /etc/bind/db.jihwan.com

;
; BIND data file for local loopback interface

; 1일 동안  유지되는  TTL (초)
$TTL    86400
@       IN      SOA     localhost. root.localhost. (
                              2         ; 일련  번호
                          86400         ; 새로  고침  (1일)
                           7200         ; 재시도  (2시간)
                         604800         ; 만료  (1주일)
                          86400 )       ; 최소  TTL (1일)
;
@       IN      NS      jihwan.com.
@       IN      A       101.101.216.137
www     IN      A       101.101.216.137
smtp    IN      A       101.101.216.137
pop3    IN      A       101.101.216.137
```

➜ named.conf.local에서 지정한 zone 파일을 작성해줍니다.

➜ 사용자@jihwan.com 또는 subdomain.jihwan.com이 허용됩니다.

➜ subdomain은 www, smtp, pop3를 작성했으며 NS는 nameserver를 의미하고 A address를 의미합니다.

$ vim/etc/bind/named.conf.options

```
 GNU nano 2.9.3                /etc/bind/named.conf.options

options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameserv
        // to talk to, you may need to fix the firewall to
        // ports to talk.  See http://www.kb.cert.org/vuls

        // If your ISP provided one or more IP addresses f
        // nameservers, you probably want to use them as f
        // Uncomment the following block, and insert the a
        // the all-0's placeholder.

        forwarders {
                8.8.8.8;
                8.8.4.4;
        };

        allow-query { any; };
```

➜ DNS 서버 주소를 사용해서 DNS 서버에 접속하게 될 경우, DNS 서버에서 허용하는 DomainName 외에는 접속할 수 없는 오류가 발생합니다.

➜ forwarders의 내용은 구글 DNS 서버를 포워딩하므로 DNS 서버에서 정의하지 않은 도메인 정보를 query 요청시에도 누구든지 처리할 수 있도록 해줍니다.

$ sudo apt install resolvconf

➜ 네임 서버를 관리해주는 패키지입니다.

➜ 설치가 완료되었습니다.

$ vim /etc/resolvconf/resolv.conf.d/head

```
nameserver 101.101.216.137
```

➜ head 파일에 editor로 nameserver를 지정해줍니다.

$ sudo reboot

$ cat /etc/resolv.conf

```
root@jihwan:~# cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for gli
#      DO NOT EDIT THIS FILE BY HAND -
# 127.0.0.53 is the systemd-resolved
# run "systemd-resolve --status" to s

nameserver 101.101.216.137
nameserver 127.0.0.53
search ncloud.com
root@jihwan:~#
```

➜ 재부팅 후 resolv.conf에 nameserver로 등록된 것을 확인합니다.

$ nslookup

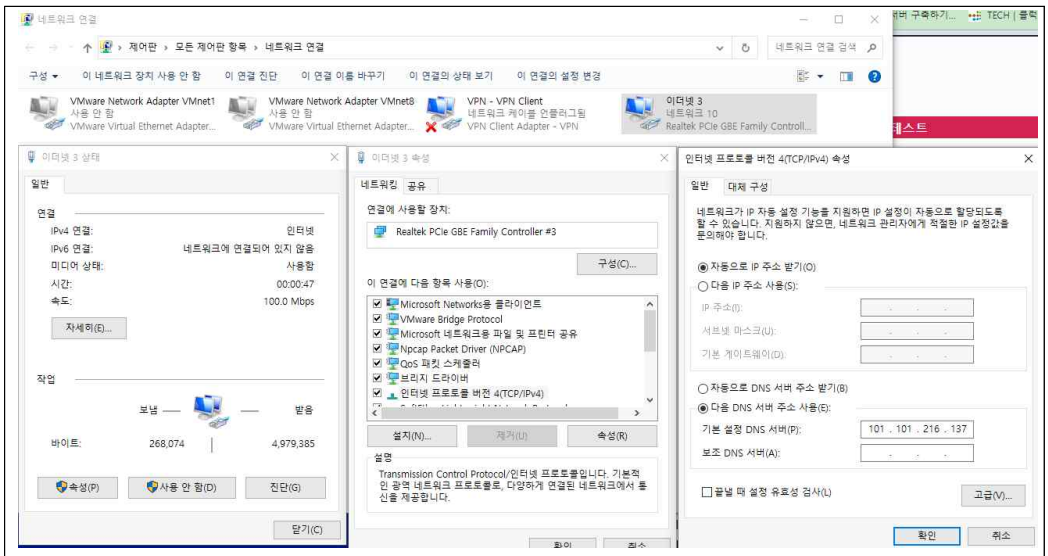
```
root@jihwan:~# nslookup
> www.jihwan.com
Server:         101.101.216.137
Address:        101.101.216.137#53

Name:   www.jihwan.com
Address: 101.101.216.137
> exit
```

➜ nslookup이 실행되면 www.jihwan.com 도메인 주소로 테스트해봅니다.

➜ Zone에서 설정한 서브도메인도 제대로 설정된 것을 확인 할 수 있습니다.

➜ Bind9 셋팅이 끝났습니다.

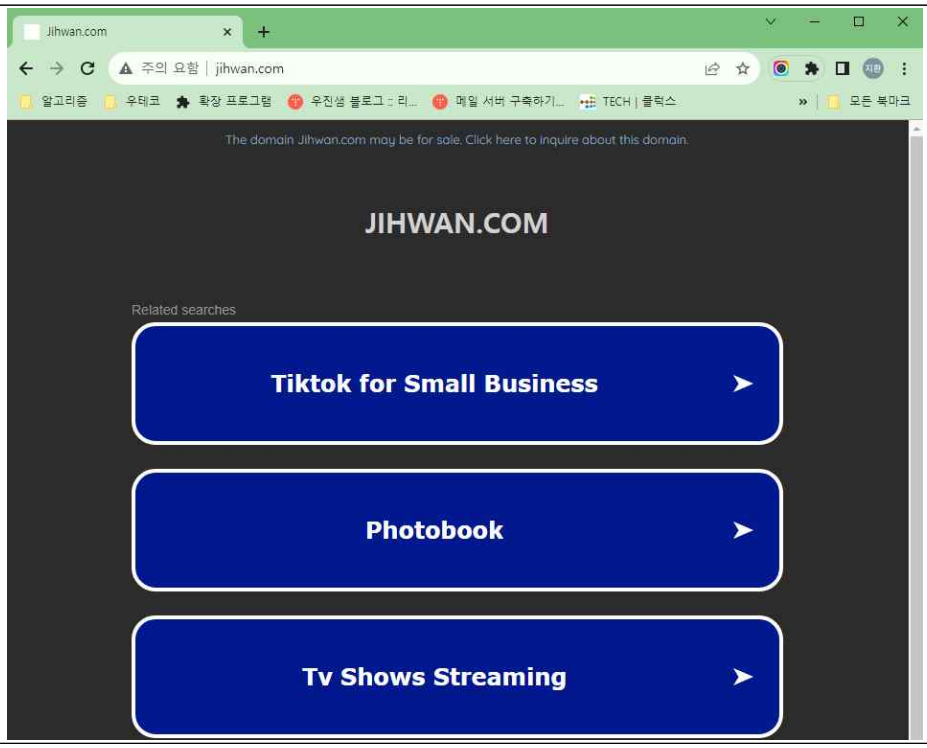
3. 로컬 네트워크 설정



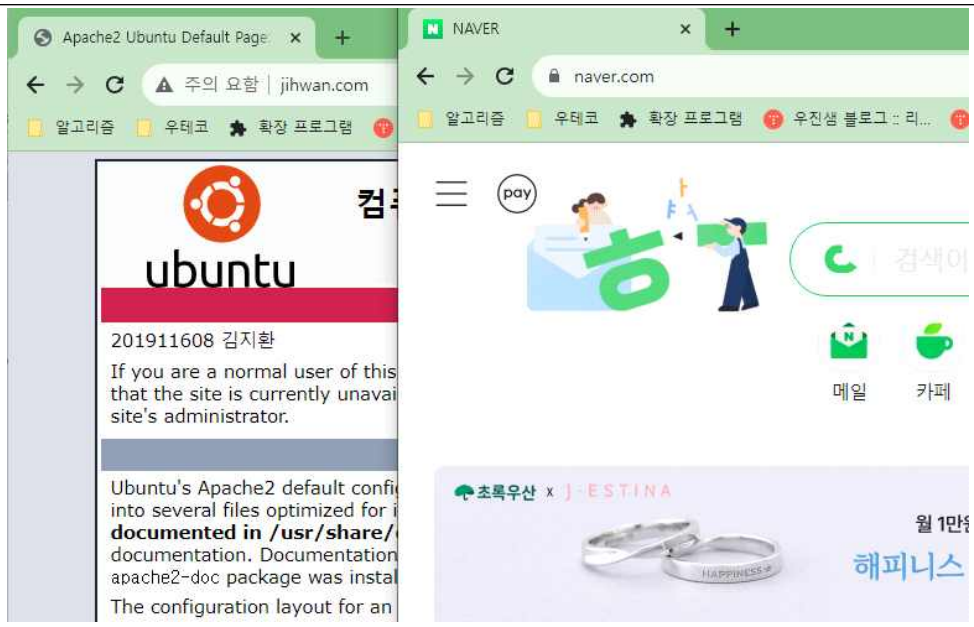
➜ IPv4 설정에서 DNS 서버 주소를 DNS 서버 공인 IP로 설정합니다.

4. 로컬 네트워크 테스트 - HTTP

▪ DNS 적용 전



▪ DNS 적용 후



➜ www.jihwan.com 도메인 주소가 로컬네트워크로 연결 돼 우분투 서버의 DNS로 접속되었습니다.

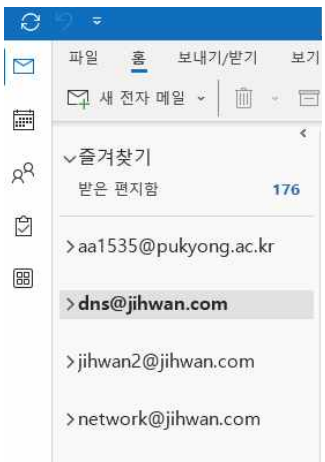
➜ 네이버는 구글 DNS로 포워딩해서 로컬네트워크여도 정상작동되고 있습니다.

5. 로컬 네트워크 테스트 - SMTP

$ addusr dns

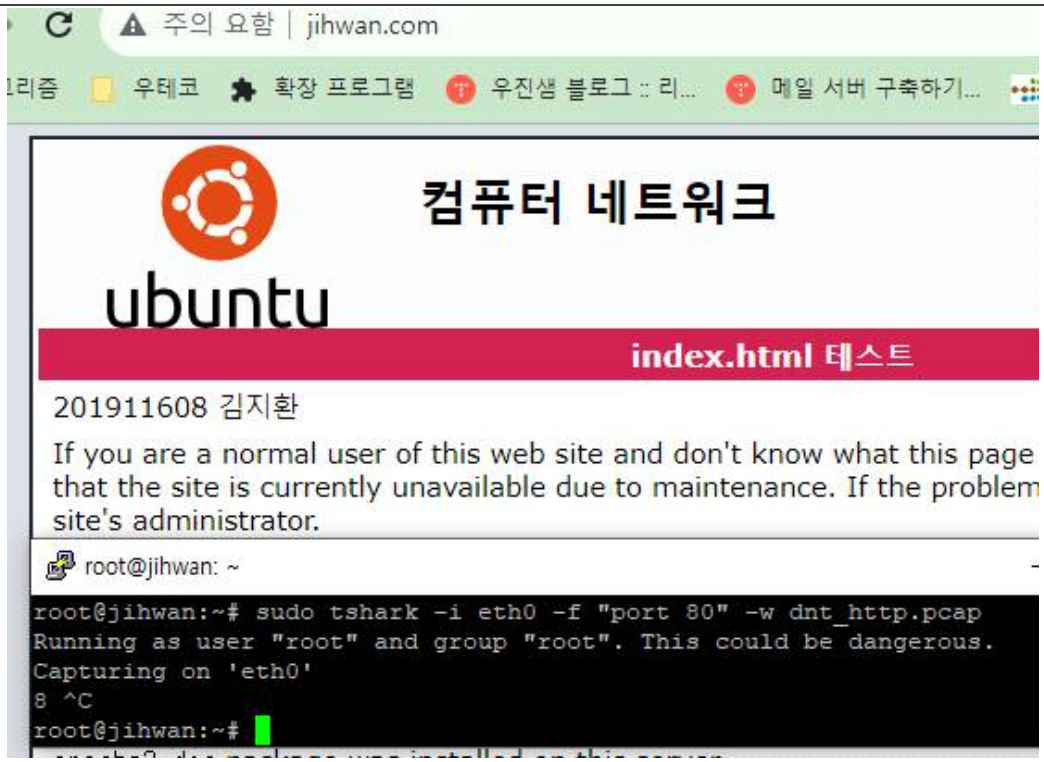
```
root@jihwan:~# adduser dns
'dns' 사용자를 추가 중 ...
새 그룹 'dns' (1005) 추가 ...
새 사용자 'dns' (1005) 을 (를) 그룹 'dns' (으)로 추가 ...
'/home/dns' 홈 디렉터리를 생성하는 중 ...
'/etc/skel'에서 파일들을 복사하는 중 ...
새 암호 :
잘못된 암호 : 너무 짧습니다
잘못된 암호 : 너무 간단함
새 암호 재입력 :
passwd: password updated successfully
Changing the user information for dns
Enter the new value, or press ENTER for the default
        Full Name []: dnstest
        Room Number []: 3
        Work Phone []: 3
        Home Phone []: 3
        Other []: 3
정보가 올바릅니까? [Y/n] y
root@jihwan:~#
```

➜ user dns를 추가합니다.



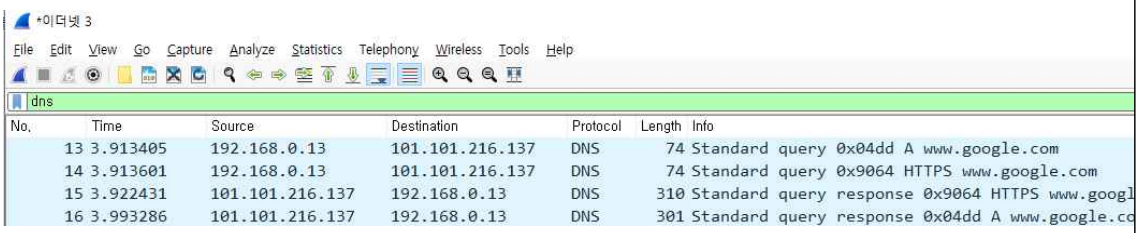
➜ smpt.jihwan.com, pop3.jihwan.com으로 서브 도메인 주소로 SMTP서버 또한 로컬네트워크에서 성공적으로 메일을 추가했습니다.

6. WireShark



➜ HTTP 서버에서도 패킷이 제대로 송수신되고 있습니다.

➜ SMTP 서버에서도 패킷이 제대로 송수신되고 있습니다.



➜ Client에서 Wireshark를 실행후 dns를 필터링하면 DNS서버 공인 IP로 제대로 리다이렉션되고 있습니다.

V. 마무리

1. 진행하는 동안 발생한 이슈

➜ VMWare 같은 가상환경을 사용하지 않아서 WireShark 사용의 문제가 있었습니다.
저는 이슈를 해결하기 위해서 WireShark 배포 이전 tShark라는 것을 알게되었고 tshark로 문제를 해결했습니다.

➜ OutLook을 사용해 본 적이 없어서 SMTP 하나만 연결하려는데 안돼서 어려움이 있었습니다. POP3 또는 IMAP과 같이 사용해서 하나의 메일 서버로 동작해야한다는 점을 깨달았습니다.

➜ 인터넷을 참고해서 SMTP 서버를 설치하는데 인터넷에서는 거의 다 security가 설정된 SMTPS나 POP3S에 대한 내용만 작성되어 있었습니다.
SMTP, POP3와 SMPTS, POP3S의 구성 차이에 대한 생각을 많이 하게 되었습니다.

2. 느낀점

➜ Security를 설정하지 않은 http와 smtp만해도 어려웠는데 https와 smtps 서버의 관리자가 된다면 많이 복잡할 것 같습니다.

➜ 이번 과제를 통해서 저만의 메일 서버를 만들 수 있게 됐습니다.

➜ 컴퓨터 네트워크는 보안이 정말 중요하겠구나를 깨달았습니다. 만약, ftp와 같은 파일 전송 프로토콜에서 보안이 적으면 강제로 DNS 서버 주소를 변경한다면 해킹의 우려가 정말 심각할 것 같다고 느꼈습니다.

3. 참고자료

SMTP 서버 설치 https://terianp.tistory.com/6
DNS 서버 설치 https://tech.ktcloud.com/66