6/27

```
NAT_RTR>en
NAT_RTR#sh accees-list
               ^
% Invalid input detected at '^' marker.

NAT_RTR#sh access-list
Standard IP access list 1
    10 permit 192.168.1.0 0.0.0.255
    20 permit 172.16.2.0 0.0.0.255
    30 permit 10.0.0.0 0.255.255.255
Standard IP access list 10
    10 deny 10.0.0.0 0.255.255.255
    20 deny 172.16.0.0 0.15.255.255
    30 deny 192.168.0.0 0.0.255.255
    40 permit any
    50 deny any

NAT_RTR#
```

아래는 트래픽 필터링 위에는 식별

- nat router에서 acl 적용

  1)

  192.168.1.0/24 네트워크에서 서버 100.100.100.200에 웹 접속만 된다.

  서버로 가는 나머지 트래픽 모두 차단. 그 외 트래픽은 모두 허용

  ip access-list extended permit-outbound

  permit tcp 192.168.1.0 0.0.0.255 host 100.100.100.200 eq 80

  deny ip 192.168.1.0 0.0.0.255 host 100.100.100.200

  permit ip any any

  tcp 192에서 서버 200으로 가는 트래픽 중 웹접속 80 port만 허용

  2)

172. 16.2.0 /24 네트워크에서는 서버 100.100.100.200에 ftp접속만 가능

  permit tcp 172.16.2.0 0.0.0.255 host 100.100.100.200 eq 21

deny ip 172.16.2.0 0.0.0.255 host 100.100.100.200

permit ip any any

둘이 합친다

ip access-list extended permit-outbound

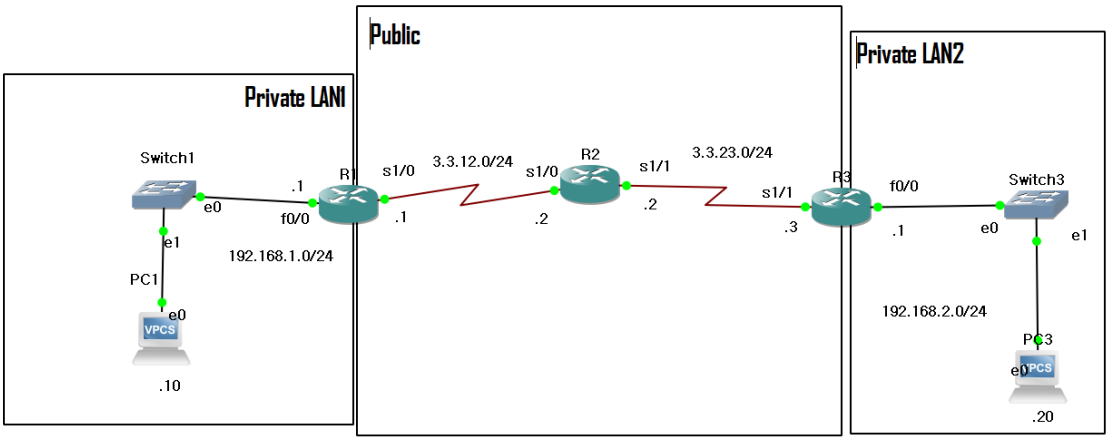
permit tcp 192.168.1.0 0.0.0.255 host 100.100.100.200 eq 80

deny ip 192.168.1.0 0.0.0.255 host 100.100.100.200

permit tcp 172.16.2.0 0.0.0.255 host 100.100.100.200 eq 21

deny ip 172.16.2.0 0.0.0.255 host 100.100.100.200

permit ip any any



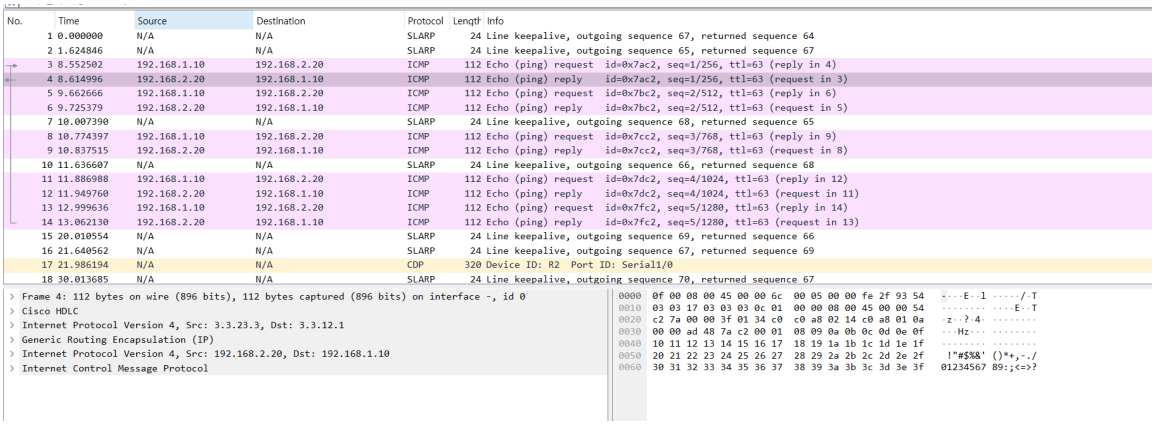
가운데 구간이 public이라 lan1과 lan2가 통신이 안된다.

그것을 터널링을 통해 통신이 되게끔한다. 둘이 연결이 되어있는 것처럼 동작 vpn

가상 사설망

```
R1)
int tun 1
ip unnumbered f0/0
```

```
tunnel source s1/0
tunnel destination 3.3.23.3

ip route 192.168.2.0 255.255.255.0 tunnel 1

R3)
int tun 3
ip unnumbered f0/0
tunnel source s1/1
tunnel destination 3.3.12.1

ip route 192.168.1.0 255.255.255.0 tunnel 3
```



문제가 뭐냐?

안에 내용이 캡쳐할 시 다 보인다. 보안이 전혀 안되는 것

암호화 하는 방법 ip-sec ssl등등

vpn은 결국 터널을 이용해서 통신을 하는데 안을 암호화한다.

원래는 터널링에 암호화가 없지만 vpn에 다 적용되어 있음

ssl ← https 사용

웹 접속시 보안에 사용

- site to site vpn 연결

1) 1단계 정책 설정: 암호화에 사용되는 키 교환을 위한 암호채널 구성

내가 꼭 쓰고 싶은게 있다면 미리 설정해야 함

내가 이렇게 설정했다면 반대도 똑같이 해야 함

```
R1)
crypto isakmp policy 10
    encryption aes //aes 비대칭키
    authentication pre-share // 미리공유하겠다 인증서 방식이 기본설정
    hash sha
    group 2 // 디피 엘만 그룹 키 크기가 2 숫자가 커질수록 암호의 길이가
    exit

crypto isakmp key cisco address 3.3.23.3
```

2) 2단계 정책 설정: 보호트래픽, 데이터를 보호할 정책,

```
ip access-list extended R1R3
    permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

crypto ipsec transform-set r1-policy esp-aes esp-md5-hmac
```

3) 크립토 맵 작성

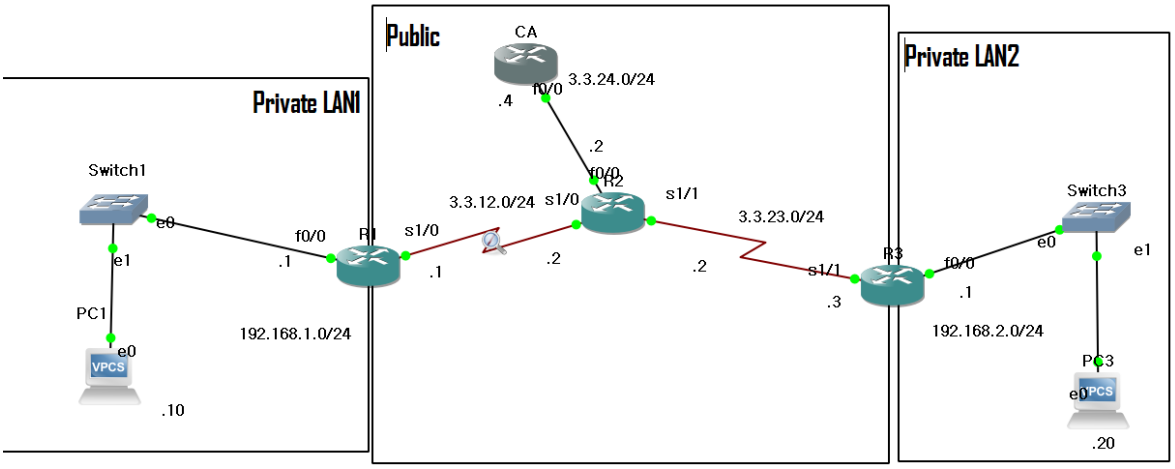
```
crypto map vpn 10 ipsec-isakmp
    match address R1R3
    set peer 3.3.23.3
    set transform-set r1-policy

int s1/0
crypto map vpn
```

💡 덮어쓰기가 안되기에 잘못 썼으면 no로 지워야 한다

PKI(Public Key Infrastructure)란 디지털 인증서 사용을 위한 기반구조를 가리킨다.
즉, 디지털 인증서의 생성, 관리, 배포, 사용, 저장 및 해제를 위하여 필요한 하드웨어,소프트웨어, 정책 및 절차를 총칭하는 용어이다.

디지털 인증서란 인증기관(CA, certificate authority)이 소유자의 신분과 그의 공개키 정보를 보증하기 위해 발급하는 전자문서이다.

즉, CA는 사용자의 공개키 인증서를 전자서명함으로써, 사용자 인증서가 진짜라는 것과 내용이 변조되지 않았다는 것을 보증한다.



r1과 r3가 ca한테 인증서를 받고 서로 보내고 받아온 거를 ca한테 이것이 맞냐고 물어봄

1. NTP 설정 (Network Time Protocol)

   시간동기화

```
CA)
    clock set 12:00:00 jun 27 2024
    conf t
    clock timezone KOR 9
    ntp master

  R1)
    conf t
    ntp server 3.3.24.4
```

```
R3)
 conf t
 ntp server 3.3.24.4
```

2. 디지털 인증 서버 설정

```
CA)
conf t
ip http server

crypto pki server CertSVR
issuer-name cn=CertServer, l=Incheon, c=KR
grant auto
no sh
패스워드 설정(7자리 이상)
```

3. 공개키/개인키 생성

   1) RSA 키 생성

   ip domain-name cloudwave.com

   crypto key generate rsa general-keys modulus 2048

> 💡 공개키 확인 코드는 show crypto mypubkey rsa

4. CA인증서 다운로드

   CA와 사용자 사이에 인증서를 등록하고 CRL(certificate revocation list)를 요청할 때 SCEP(Simple certificate enrollment protocol)을 사용.

SCEP는 인증서 등록시에는 http를 사용하고, CRL 확인 시에는 http 또는 ldap을 이용함.

```
R1)
-CA 지정
crypto pki trustpoint CA
enrollment url http://3.3.24.4
subject-name cn=R3, l=Busan, c=KR
exit

-CA인증서 다운 및 인증
crypto pki authenticate CA

- 라우터 R1의 인증서 요청
crypto pki enroll CA

-인증서 정보 확인
show crypto pki certificates
```

<인증서 발급 과정>
1.R1에 CA정보 등록
2.CA정보를 이용해 CA의 인증서 요청후 수신(CA의 공개키를 가지고 있음)
3.R1이 자신의 개인키/공개키를 생성하고 공개키를 CA에게 전송. 이때 CA의 공개키로 암호화해서 전송.
4.CA는 R1이 보낸 R1의 공개키를 확인하기 위해 자신의 개인키로 복호화함.
5.CA는 R1의 공개키를 보증하는 인증서를 생성하고 R1에게 인증서를 전송함. 이때 자신의 개인키로 전송(R1은 CA의 공개키로 복호화가능)
6.R1은 수신된 인증서를 저장.

요청 후 승인되면 인증서가 만들어지고 자동으로 라우터로 다운로드 된다.

💡 cmd에서 bash하면 우분투 연결 가능하다

aws 클라이언트 vpn 구성하기

Easy-rsa 이용하여 ca.crt, server.crt, server.key , client.crt, client.key 생성하기
sudo apt-get install easy-rsa
make-cadir $HOME/keys && $HOME/keys
cd $HOME/keys

nano vars

```
$ vi vars
.....
set_var EASYRSA_REQ_COUNTRY    "KR"                # 소속 국가 이름 입력
set_var EASYRSA_REQ_PROVINCE   "Seoul"             # 소속 국가 내 시(市)단위 입력
set_var EASYRSA_REQ_CITY       "Gangnam"           # 소속 시(市) 내 구(區)단위 입력
set_var EASYRSA_REQ_ORG        "love_tolty"        # 소속 조직 명 입력
set_var EASYRSA_REQ_EMAIL      "love_tolty@naver.com" # 소속 이메일 주소 입력
set_var EASYRSA_REQ_OU         "love_tolty"        # 소속 조직 내 세부 부서명 입력
.....
```

- 인증서 관련 정보 초기화
  ./easyrsa init-pki

- 인증기관 생성. 인증서, 개인키
  ./easyrsa build-ca

```
$ ./easyrsa build-ca
.....
Enter New CA Key Passphrase:  ********      # CA 비밀번호 입력
Re-Enter New CA Key Passphrase:  ********   # CA 비밀번호 재입력

-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:     # CA 인증기관 이름
.....
```

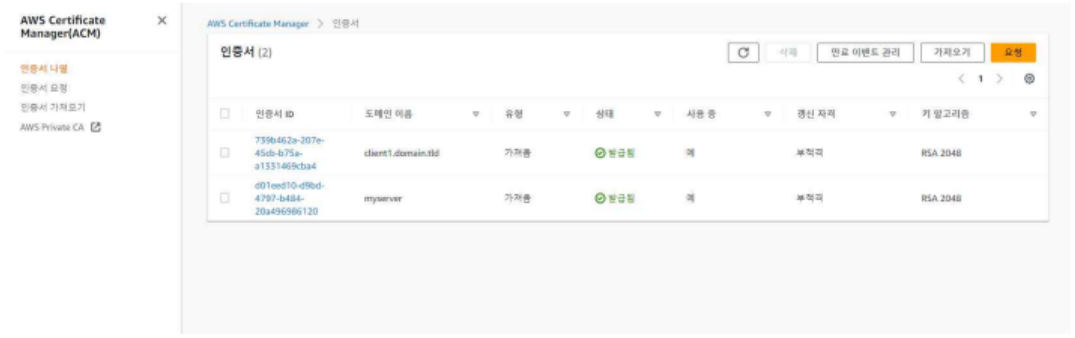
ls $HOME/keys/pki
ls $HOME/keys/pki/private

- 서버/클라이언트 인증서, 개인키 발급
  ./easyrsa build-server-full server nopass
  ./easyrsa build-client-full client1.domain.tld nopass

```
leejeuk@DESKTOP-KM62UF8:~/keys$ ls pki/issued/
client1.leejeuk.tld.crt  server.crt
leejeuk@DESKTOP-KM62UF8:~/keys$ ls pki/private/
ca.key  client1.leejeuk.tld.key  server.key
leejeuk@DESKTOP-KM62UF8:~/keys$
```

Certificate Manager에서 인증서 가져오기(서버, 클라이언트)

>> Certificate Manager에서 인증서 가져오기(서버, 클라이언트)



1.

keys/pki/issued 폴더 안에 존재

cat jeuk.server.crt

인증서 본문에 집어넣는다

2.

keys/pki/private 폴더 안에 존재

cat jeuk.server.key

인증서 프라이빗 키에 집어넣는다.

3.

keys/pki 폴더 안에 존재

cat ca.crt

인증서 체인에 집어 넣는다.

가져온 최종 인증서이다.

우분투가 ca 역할을 하는 것임

인증서를 해준다.

키를 만드는 방법은 여러 개다.

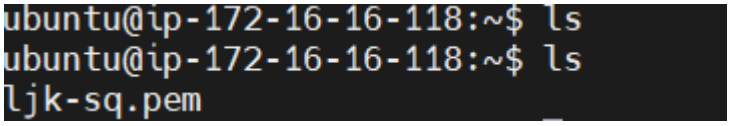
우분투에서 만들어도 윈도우에서 만들어도 똑같다. 짝만 맞춰주면 된다.

서비스 세션 메인을 이용하면 프라이빗 서버에 들어갈 수 있다.

보통은 퍼블릭 서버를 통해서 프라이빗 서버에 들어간다.

mobaxterm에서 연동해서 실행하기

cd/Downloads 에서 scp -i .\ljk-keypair.pem ljk-keypair.pem ubuntu@퍼블릭 ip:/home/ubuntu을 통해 우분투 환경으로 옮겨야 한다.



chmod 400 ljk-keypair.pem

ssh -i ljk-keypair.pem ubuntu@private key