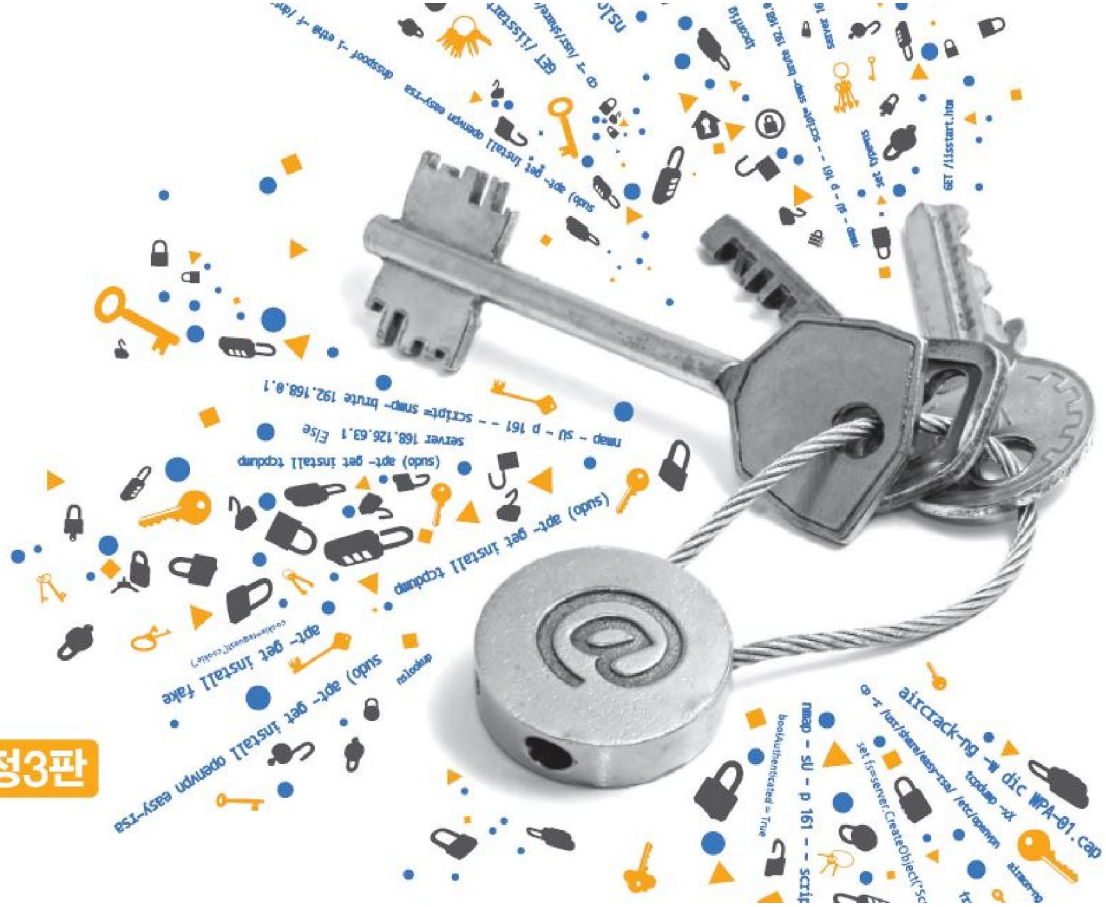
원리를 IT 가르쳐 맛보다

네트워크 해킹과 보안

개정3판

정보 보안 개론과 실습



Chapter 08 터널링

목차

IT COOKBOOK

학습목표

- 터널링을 이해한다.
- VPN을 이해하고 구성할 수 있다.
- 은닉 채널과 암호화 개념을 이해한다.

■ 터널링(Tunneling)

- 인터넷을 사적이고 안전한 네트워크의 일부로 사용하게 하는 기술

■ 캡슐화

- 터널 장비를 지날 때 원래 패킷에 있던 2계층이나 3계층 정보를 벗겨내지 않고 캡슐화 수행

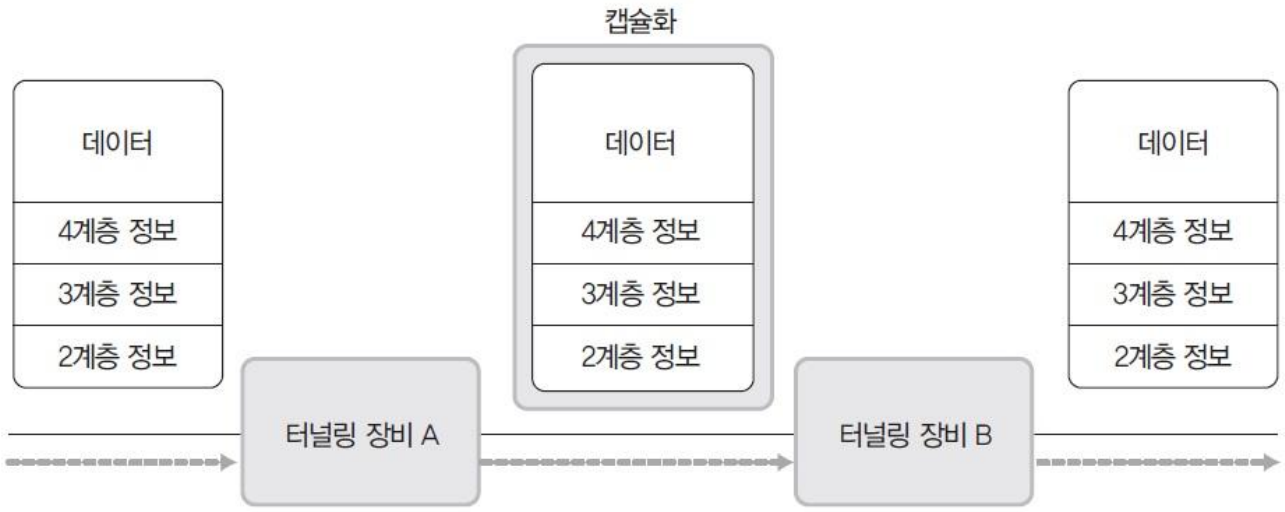


그림 8-1 터널링에서의 패킷 캡슐화

■ VPN(Virtual Private Network)

- 터널링의 대표적인 보안 장비

■ Internal Network

- 기업 내부에서 데이터 통신을 하기 위한 네트워크
- 회사 내의 데이터 통신은 인터넷과 구분된 별도의 임대 회선(Leased Line) 사용
- 가격이 고가인 것이 단점

→ 임대 회선과 비슷한 수준의 기밀성을 제공하려면 VPN 사용 및 암호화 필요
VPN에서 사용하는 암호화 프로토콜에는 PPTP, L2TF, IPSec, SSL 등이 있음.

■ VPN 이용 사례

- 해외 여행을 가서도 국내 게임 서버 이용
- 집에서도 회사 내의 서버에 보안 상태로 접근

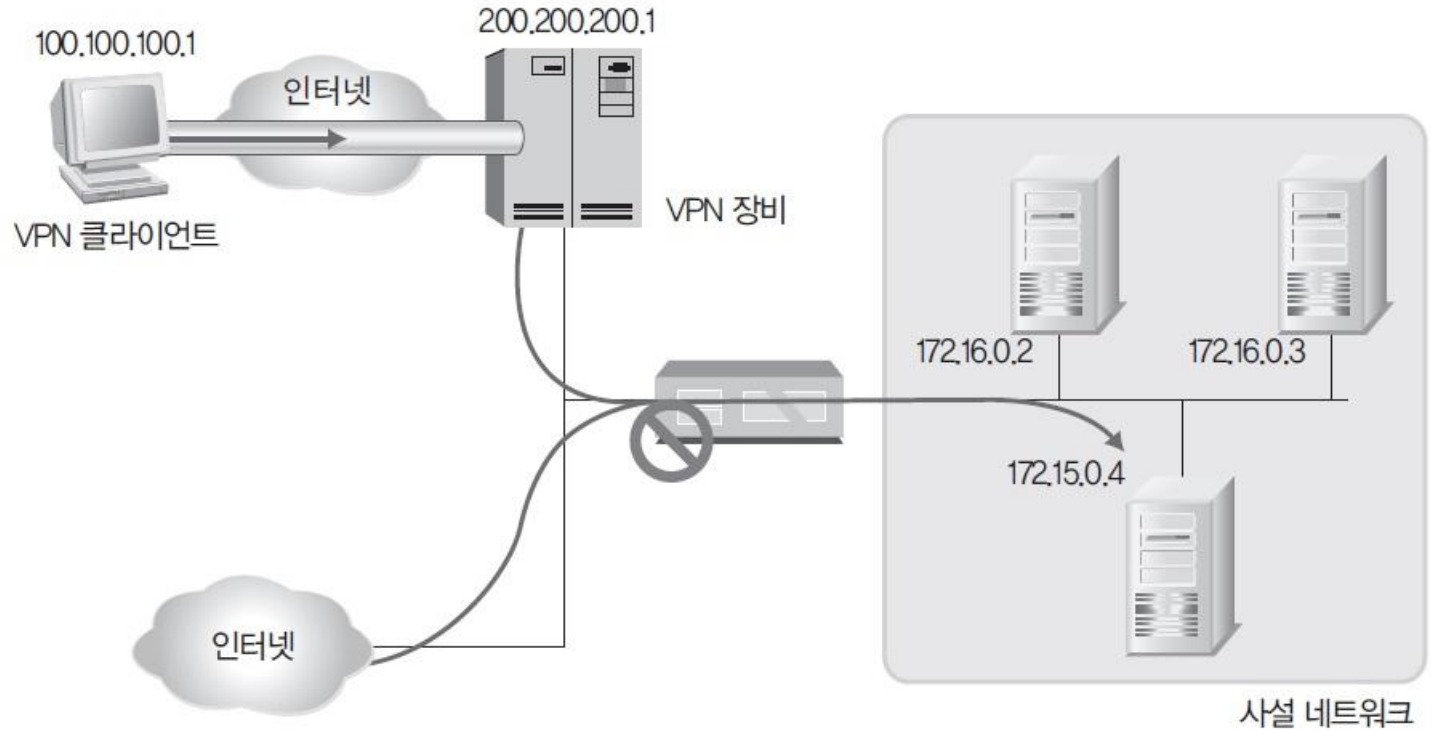


그림 8-2 VPN을 이용한 외부에서의 접근

■ VPN 이용 사례

- 원격의 두 지점을 내부 네트워크처럼 이용

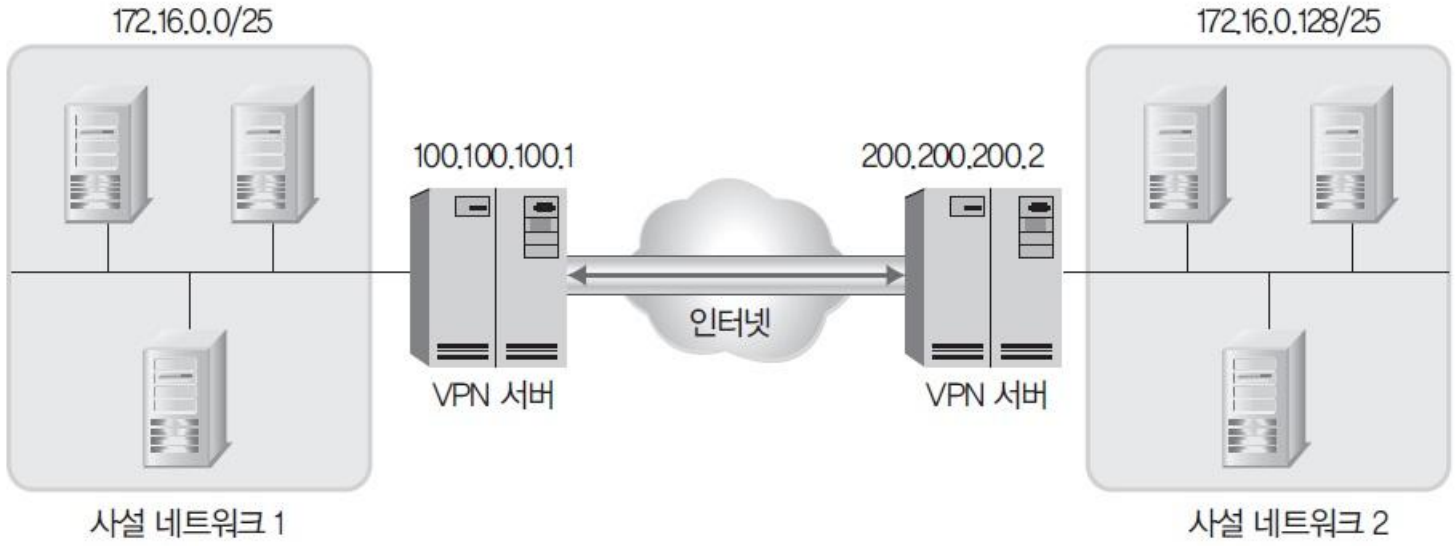


그림 8-3 VPN을 이용한 터널링

실습 8-1 Open VPN 이용하기

실습환경 • VPN 서버 시스템 : 우분투 데스트탑 14
• VPN 클라이언트 시스템 : 윈도우 7

① OpenVPN과 Easy-RSA 설치하기

(sudo) apt- get install openvpn easy-rsa

② OpenVPN 서버 설정 파일 수정하기

▪ 압축을 해제하여 복사('su' 명령을 이용해 root 권한으로 변경한 뒤 작업)

gunzip - c /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz > /etc/openvpn/server.conf

그림 8-5 서버 설정 파일 복사

② OpenVPN 서버 설정 파일 수정하기

- 텍스트 에디터로 '/etc/openvpn/server.conf'를 열어 다음 부분을 수정

변경 사항	변경 전	변경 후
VPN 연결시 암호화에 사용될 RSA 키 길이를 1024 비트에서 2048 비트로 조정	dh dh1024.pem	dh dh2048.pem
클라이언트의 웹 트래픽을 목적지로 전달	;push "redirect-gateway def1 bypass-dhcp"	push "redirect-gateway def1 bypass-dhcp"
DNS 서버 설정(개인별 DNS 주소 사용)	;push "dhcp-option DNS 208.67.222.222" ;push "dhcp-option DNS 208.67.220.220"	push "dhcp-option DNS 8.8.8.8" push "dhcp-option DNS 8.8.4.4"
VPN에 접속한 사용자의 권한을 nobody와 nogroup으로 제한(root 권한으로 VPN을 사용할 때는 그대로 둠)	;user nobody ;group nogroup	user nobody group nogroup

③ 패킷 포워딩 설정하기

- '/etc/sysctl.conf' 값을 다음과 같이 설정

변경 사항	변경 전	변경 후
패킷 포워드 허용	#net.ipv4.ip_forward=1	net.ipv4.ip_forward=1

④ 인증기관(CA)설치 및 서버 인증서 생성하기

- CA를 생성하기 위해 Easy RSA의 스크립트를 복사하고, 키 저장을 위한 디렉토리 생성

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
mkdir /etc/openvpn/easy-rsa/keys
```

```
root@ubuntu-14: /
root@ubuntu-14:/# cp -r /usr/share/easy-rsa/ /etc/openvpn/
root@ubuntu-14:/# mkdir /etc/openvpn/easy-rsa/keys
root@ubuntu-14:/#
```

그림 8-6 Easy-RSA 스크립트 복사 및 키 저장 공간 생성

④ 인증기관(CA)설치 및 서버 인증서 생성하기

- 키 생성을 위한 기본 정보를 입력하기 위해 '/etc/openvpn/easy-rsa/vars' 파일을 설정

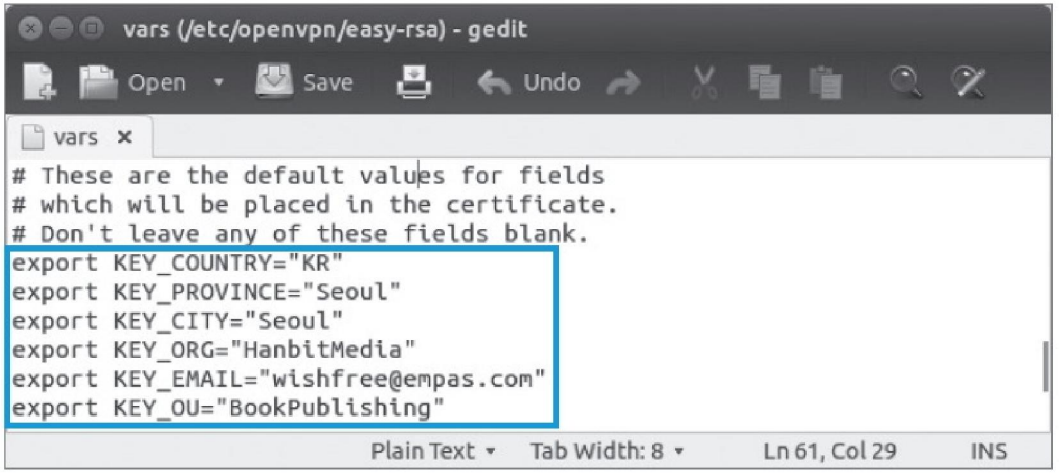


그림 8-7 키 생성을 위한 정보 입력

- '/etc/openvpn/easy-rsa/vars'에서 인증서에 사용할 키(Key) 이름을 바꿈.

변경 사항	변경 전	변경 후
키 이름 변경	export KEY_NAME='EasyRSA'	export KEY_NAME='server'

④ 인증기관(CA)설치 및 서버 인증서 생성하기

- 인증서에 사용할 디피에-헬만 키를 생성

openssl dhparam - out /etc/openvpn/dh2048.pem 2048

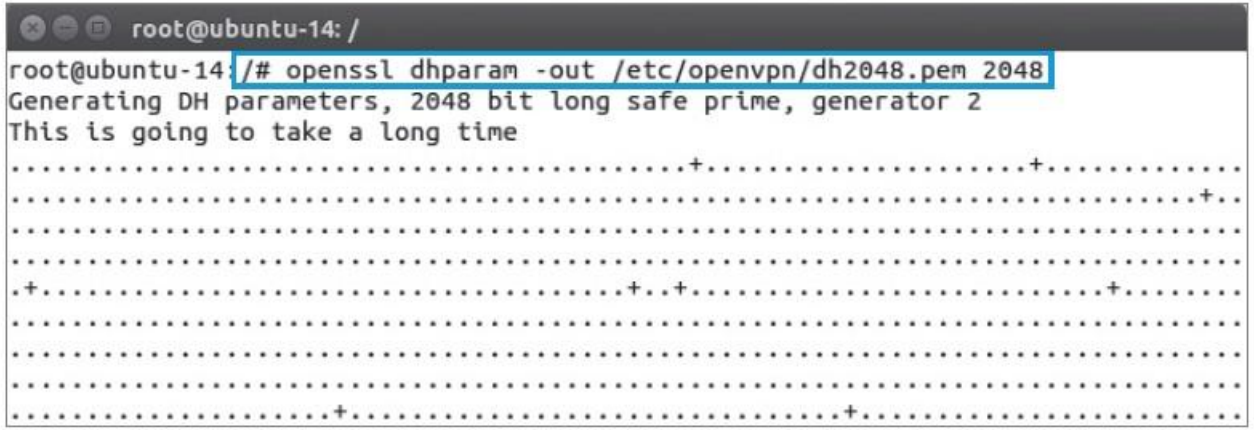


그림 8-8 디피에-헬만 키 생성

⑤ 인증기관(CA)의 인증서 생성하기

- 인증기관 CA를 생성하기 위해 '/etc/openvpn/easy-rsa/' 디렉토리로 이동

```
cd /etc/openvpn/easy-rsa/
```

```
root@ubuntu-14: /etc/openvpn/easy-rsa
root@ubuntu-14:/etc/openvpn/easy-rsa# ls
build-ca          build-key-server  list-crl          sign-req
build-dh          build-req         openssl-0.9.6.cnf  vars
build-inter       build-req-pass    openssl-0.9.8.cnf  vars~
build-key         clean-all         openssl-1.0.0.cnf  whichopensslcnf
build-key-pass    inherit-inter     pkitool
build-key-pkcs12  keys              revoke-full
root@ubuntu-14:/etc/openvpn/easy-rsa#
```

그림 8-9 /etc/openvpn/easy-rsa/ 디렉토리 내용

- 현재 작업할 디렉토리를 '/etc/openvpn/easy-rsa/var'로 설정하기 위해 source 명령 실행

```
source ./vars
```

⑤ 인증기관(CA)의 인증서 생성하기

- 혹시 있을지 모를 미리 생성한 키 값을 삭제하고, 인증서를 생성할 수 있는 인증기관을 만듦.

./clean-all

./build-ca

```
root@ubuntu-14: /etc/openvpn/easy-rsa
root@ubuntu-14:/etc/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/
keys
root@ubuntu-14:/etc/openvpn/easy-rsa# ./clean-all
root@ubuntu-14:/etc/openvpn/easy-rsa# ./build-ca
Generating a 2048 bit RSA private key
.........................+++
.............................................+++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KR]:
State or Province Name (full name) [Seoul]:
Locality Name (eg, city) [Seoul]:
Organization Name (eg, company) [HanbitMedia]:
Organizational Unit Name (eg, section) [BookPublishing]:
Common Name (eg, your name or your server's hostname) [HanbitMedia CA]:
Name [Server]:
Email Address [wishfree@empas.com]:
root@ubuntu-14:/etc/openvpn/easy-rsa#
```

그림 8-10 인증기관(CA) 생성

⑥ 서버 인증서와 키 생성

- 서버 인증서는 'build-key-server' 명령을 통해 수행

./build- key- server server

```
root@ubuntu-14: /etc/openvpn/easy-rsa
root@ubuntu-14:/etc/openvpn/easy-rsa# ./build-key-server server
Generating a 2048 bit RSA private key
...+++
.....................................+++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KR]:
State or Province Name (full name) [Seoul]:
Locality Name (eg, city) [Seoul]:
Organization Name (eg, company) [HanbitMedia]:
Organizational Unit Name (eg, section) [BookPublishing]:
Common Name (eg, your name or your server's hostname) [server]:
```

그림 8-11 서버 인증키 생성 1/2

⑥ 서버 인증서와 키 생성

- 'Country Name'은 기본 값으로 두고, 'challenge password'는 Enter로 넘김.
- 인증서의 기한과 생성 확정을 물어보면 두 곳 모두 'y'를 입력

```
root@ubuntu-14: /etc/openvpn/easy-rsa
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'KR'
stateOrProvinceName   :PRINTABLE:'Seoul'
localityName          :PRINTABLE:'Seoul'
organizationName      :PRINTABLE:'HanbitMedia'
organizationalUnitName:PRINTABLE:'BookPublishing'
commonName            :PRINTABLE:'server'
name                  :PRINTABLE:'Server'
emailAddress          :IA5STRING:'wishfree@empas.com'
Certificate is to be certified until Jul  1 13:55:11 2026 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@ubuntu-14:/etc/openvpn/easy-rsa#
```

그림 8-12 서버 인증키 생성 2/2

⑥ 서버 인증서와 키 생성

- 생성된 인증기관의 인증서(.crt 파일)와 키(.key 파일) 확인

```
root@ubuntu-14: /etc/openvpn/easy-rsa/keys
root@ubuntu-14:/etc/openvpn/easy-rsa/keys# ls -al
total 56
drwx------ 2 root root 4096  7월  3 22:55 .
drwxr-xr-x 3 root root 4096  7월  3 22:30 ..
-rw-r--r-- 1 root root 5637  7월  3 22:55 01.pem
-rw-r--r-- 1 root root 1765  7월  3 22:31 ca.crt
-rw------- 1 root root 1708  7월  3 22:31 ca.key
-rw-r--r-- 1 root root  137  7월  3 22:55 index.txt
-rw-r--r-- 1 root root   21  7월  3 22:55 index.txt.attr
-rw-r--r-- 1 root root    0  7월  3 22:30 index.txt.old
-rw-r--r-- 1 root root    3  7월  3 22:55 serial
-rw-r--r-- 1 root root    3  7월  3 22:30 serial.old
-rw-r--r-- 1 root root 5637  7월  3 22:55 server.crt
-rw-r--r-- 1 root root 1082  7월  3 22:55 server.csr
-rw------- 1 root root 1708  7월  3 22:55 server.key
root@ubuntu-14:/etc/openvpn/easy-rsa/keys#
```

그림 8-13 생성된 인증서와 키 확인

⑦ OpenVPN 서버 구동하기

- 인증기관의 인증서(ca.crt)와 서버 인증서(server.crt), 서버 키(server.key)를 '/etc/openvpn'으로 복사

```
cp /etc/openvpn/easy-rsa/keys/ca.crt} /etc/openvpn
cp /etc/openvpn/easy-rsa/keys/server.crt /etc/openvpn
cp /etc/openvpn/easy-rsa/keys/server.key /etc/openvpn
```

```
root@ubuntu-14: /etc/openvpn
root@ubuntu-14:/etc/openvpn# ls -al
total 56
drwxr-xr-x   3 root root  4096  7월   4 07:49 .
drwxr-xr-x 143 root root 12288  7월   3 18:18 ..
-rw-r--r--   1 root root  1765  7월   4 07:49 ca.crt
-rw-r--r--   1 root root   424  7월   3 22:17 dh2048.pem
drwxr-xr-x   3 root root  4096  7월   3 22:30 easy-rsa
-rw-r--r--   1 root root 10289  7월   3 20:51 server.conf
-rw-r--r--   1 root root  5637  7월   4 07:49 server.crt
-rw-------   1 root root  1708  7월   4 07:49 server.key
-rwxr-xr-x   1 root root  1301  2월   4  2014 update-resolv-conf
root@ubuntu-14:/etc/openvpn#
```

그림 8-14 복사된 인증서와 키 확인

⑦ OpenVPN 서버 구동하기

service openvpn start

service openvpn statuss

```
root@ubuntu-14: /etc/openvpn
root@ubuntu-14:/etc/openvpn# service openvpn start
 * Starting virtual private network daemon(s)...
 *   Autostarting VPN 'server'
root@ubuntu-14:/etc/openvpn# service openvpn status
 * VPN 'server' is running
root@ubuntu-14:/etc/openvpn#
```

그림 8-15 Openvpn 서비스의 시작 및 확인

⑧ 클라이언트 인증서와 키 생성하기

- '/etc/openvpn/easy-rsa' 디렉토리에서 'build-key' 명령을 통해 수행

  ./build- key client01

```
root@ubuntu-14: /etc/openvpn/easy-rsa
root@ubuntu-14:/etc/openvpn/easy-rsa# ./build-key client01
Generating a 2048 bit RSA private key
.........................+++
.......+++
writing new private key to 'client01.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [KR]:
State or Province Name (full name) [Seoul]:
Locality Name (eg, city) [Seoul]:
Organization Name (eg, company) [HanbitMedia]:
Organizational Unit Name (eg, section) [BookPublishing]:
```

그림 8-16 클라이언트 인증서 및 키 생성

⑧ 클라이언트 인증서와 키 생성하기

- '/etc/openvpn/easy-rsa/keys' 디렉토리에서 'client01'의 인증서와 키 확인

```
root@ubuntu-14: /etc/openvpn/easy-rsa/keys
root@ubuntu-14:/etc/openvpn/easy-rsa/keys# ls
01.pem  client01.crt  index.txt.attr      serial.old
02.pem  client01.csr  index.txt.attr.old  server.crt
ca.crt  client01.key  index.txt.old       server.csr
ca.key  index.txt     serial              server.key
root@ubuntu-14:/etc/openvpn/easy-rsa/keys#
```

그림 8-17 생성된 클라이언트 인증서 및 키 확인

⑨ OpenVPN 연결하기

- OpenVPN 클라이언트를 다운로드 받아 설치

  https://openvpn.net/ index.php/open-source/downloads.html

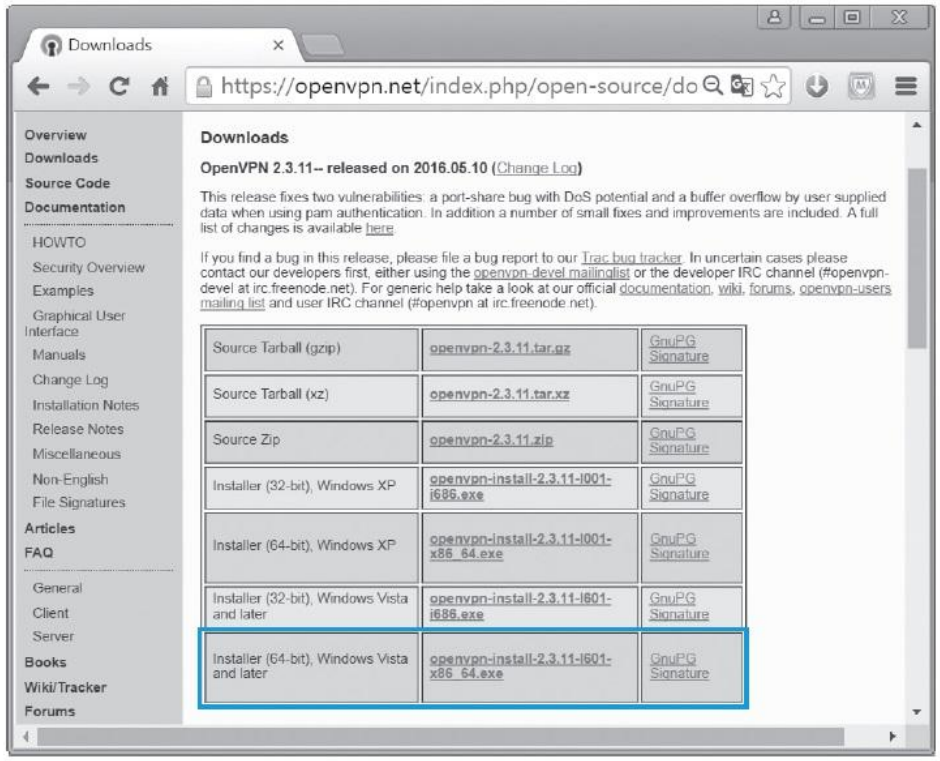


그림 8-18 OpenVPN 클라이언트 다운로드 페이지

⑨ OpenVPN 연결하기

- OpenVPN 설치 경로(C:₩Program Files₩OpenVPN₩sample-config)에서 OpenVPN 클라이언트의 연결 설정 파일인 client.ovpn 파일을 'C:₩Program Files₩OpenVPN₩config'에 복사하고 다음 항목을 수정

변경 사항	변경 전	변경 후
서버의 IP 지정	remote my-server-1 1194	remote 192.168.0.200 1194
인증기관의 인증서, 클라이언트 인증서, 키 파일명을 지정	ca ca.crt cert client.crt key client.key	ca ca.crt cert client01.crt key client01.key

⑨ OpenVPN 연결하기

- ca.crt, client01.crt, client01.key 파일을 각각 다음의 경로에서 복사하여 옮김.

  /etc/openvpn/easy-rsa/keys/client01.crt

  /etc/openvpn/easy-rsa/keys/client01.key

  /etc/openvpn/ca.crt

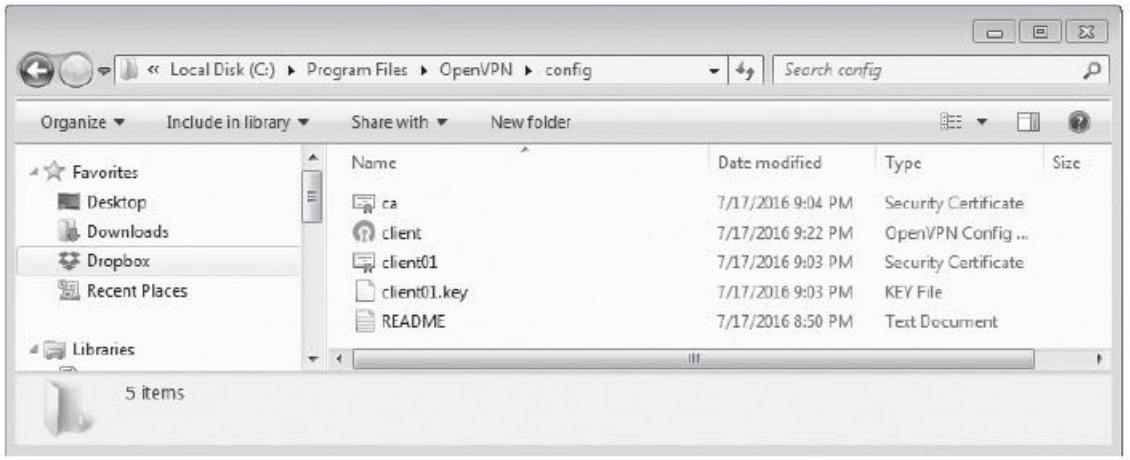


그림 8-19 'C:₩Program Files₩OpenVPN₩config' 폴더

⑨ OpenVPN 연결하기

- 'OpenVPN GUI'를 관리자 권한으로 실행

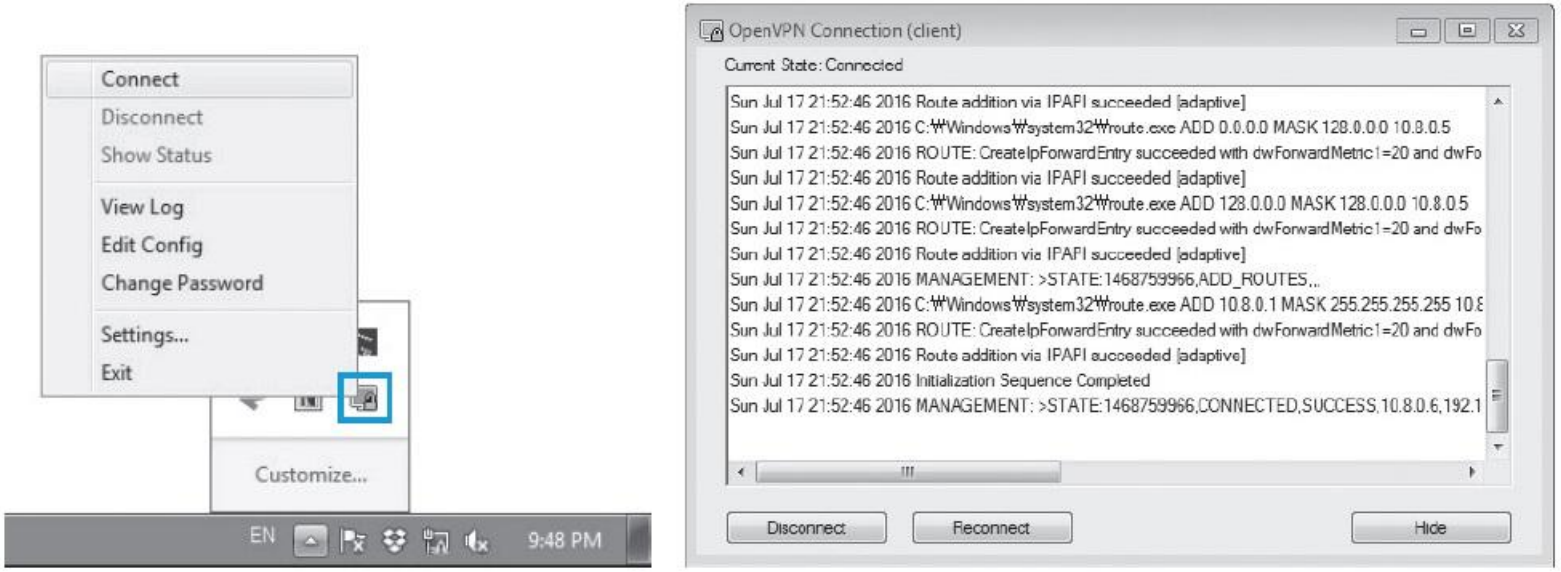


그림 8-20 OpenVPN GUI에서 VPN 연결 실행 및 결과 확인

⑩ OpenVPN 연결하기

- 클라이언트와 서버에 각각 새로운 인터페이스가 생성되고, IP가 할당됨.

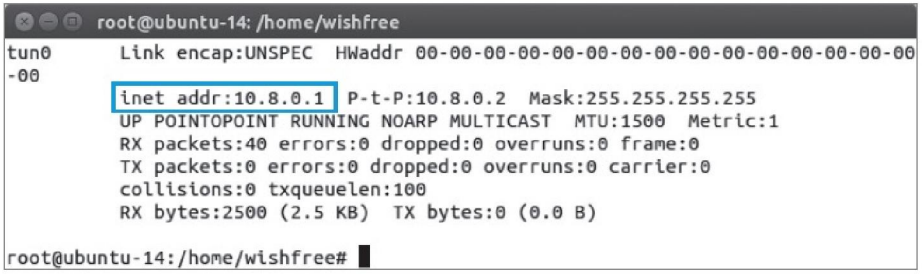


그림 8-21 OpenVPN 서버 인터페이스 확인

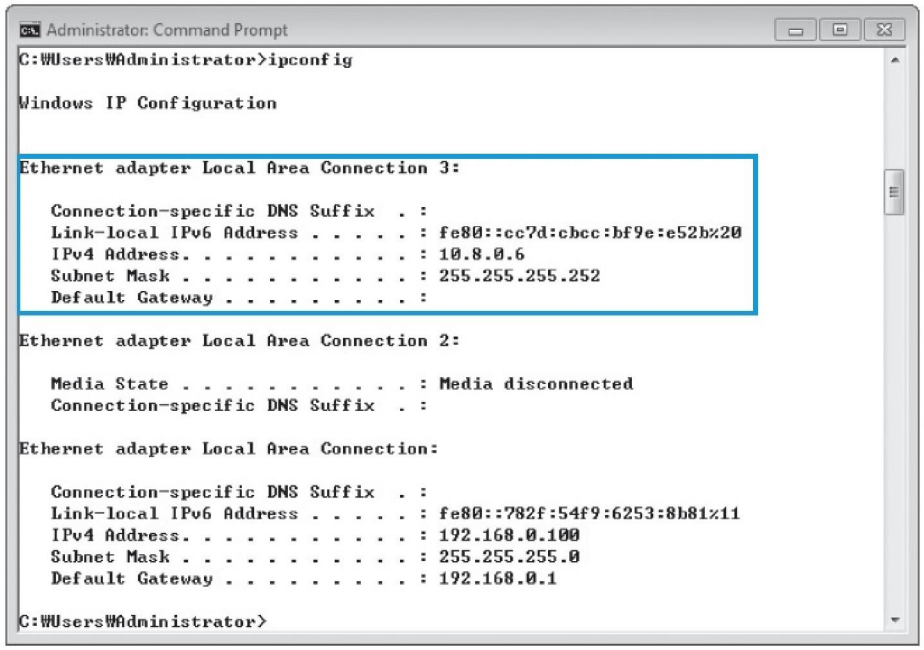


그림 8-22 OpenVPN 클라이언트 인터페이스 확인

⑩ OpenVPN 연결하기

- Wireshark를 실행하여 두 시스템 간의 패킷을 확인

ping 10.8.0.1 -t

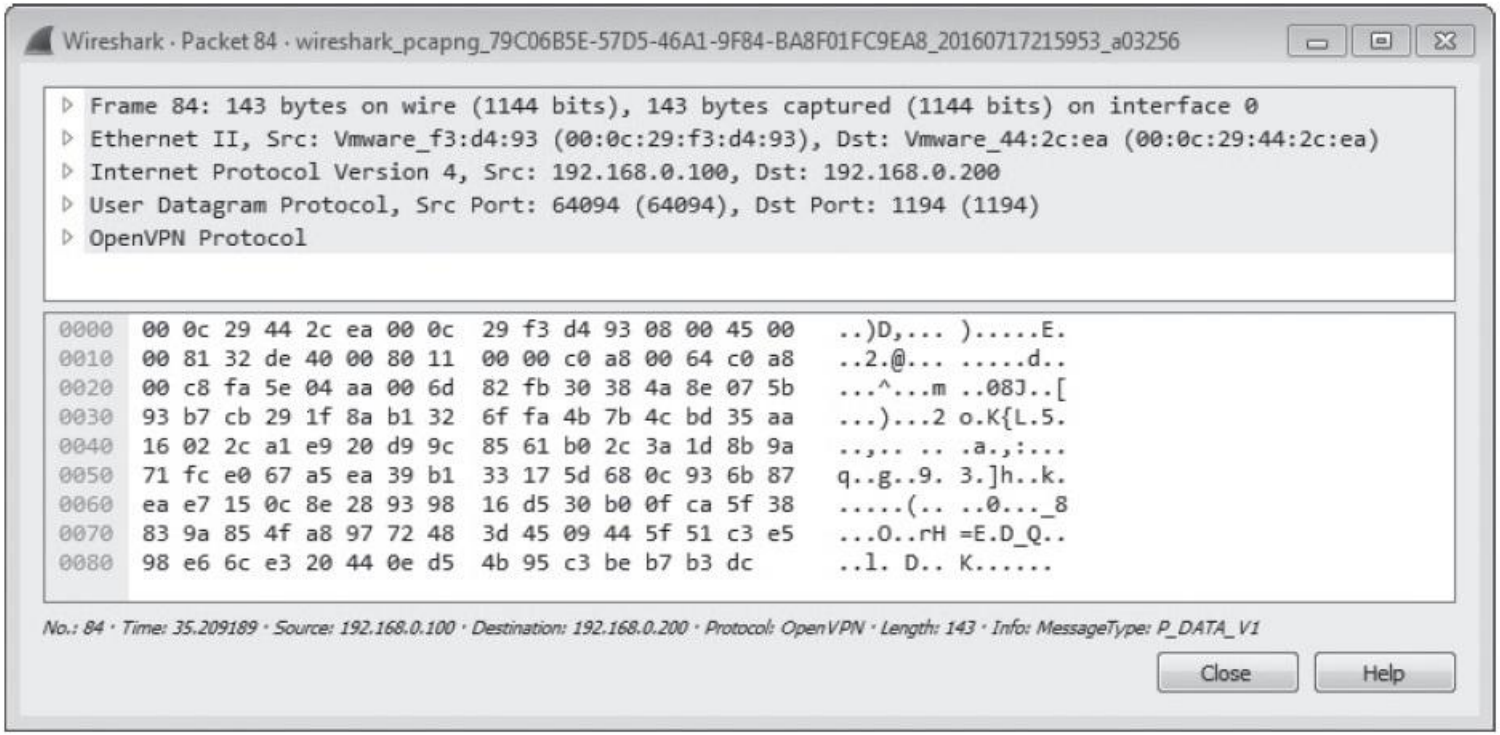


그림 8-23 OpenVPN 패킷 확인

실습 8-2 SSH 터널링하기

실습환경
- SSH 터널링 클라이언트 시스템 : 윈도우 7
- SSH 터널링 서버 시스템 : 우분투 데스크탑 14
- 필요 프로그램 : PuTTY, ssh

① SSH 설치하기

- 데스크탑에 SSH 설치

  (sudo) apt- get install openssh- server

- /usr/sbin/sshd 프로세스가 확인되면 SSHD 서비스가 설치된 것

  (sudo) ps - ef | grep sshd

```
wishfree@ubuntu-14: ~
wishfree@ubuntu-14:~$ ps -ef | grep sshd
root      2613     1  0 16:35 ?        00:00:00 /usr/sbin/sshd -D
root      2827  2613  0 16:40 ?        00:00:00 sshd: root@pts/0
wishfree  3174  2068  0 16:59 pts/1    00:00:00 grep --color=auto sshd
wishfree@ubuntu-14:~$
```

그림 8-24 ssh 서비스 확인

① SSH 설치하기

- root 계정으로 로그인이 필요한 경우 /etc/ssh/sshd-config 파일의 'PermitRootLogin' 값을 'without-password'에서 'yes'로 바꿈

  (sudo) vi /etc/ssh/sshd- config

그림 8-25 ssh 설정 변경

- 설정을 마치면 ssh 서비스를 재시작

  (sudo) /etc/init.d/ssh restart

① SSH 설치하기

- 윈도우 클라이언트에서 PuTTY로 접속하여 서비스 동작 여부 확인

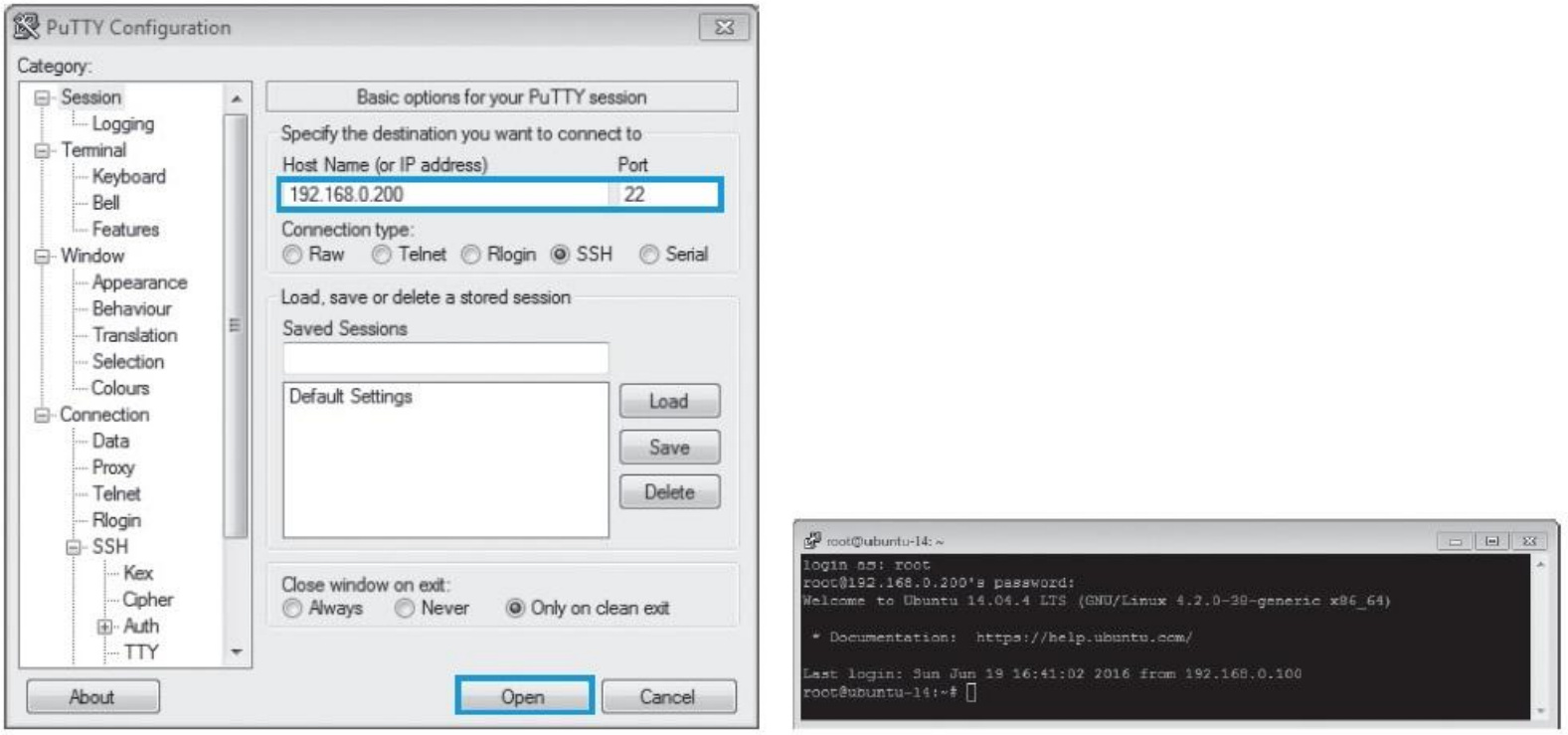


그림 8-26 SSH 서비스의 정상 작동 여부 확인

② SSH를 이용한 터널링 설정하기

- 로컬 주소에서 5000번 포트로 접속을 요청해오는 패킷을 SSH 터널을 이용해 전달해주도록 설정

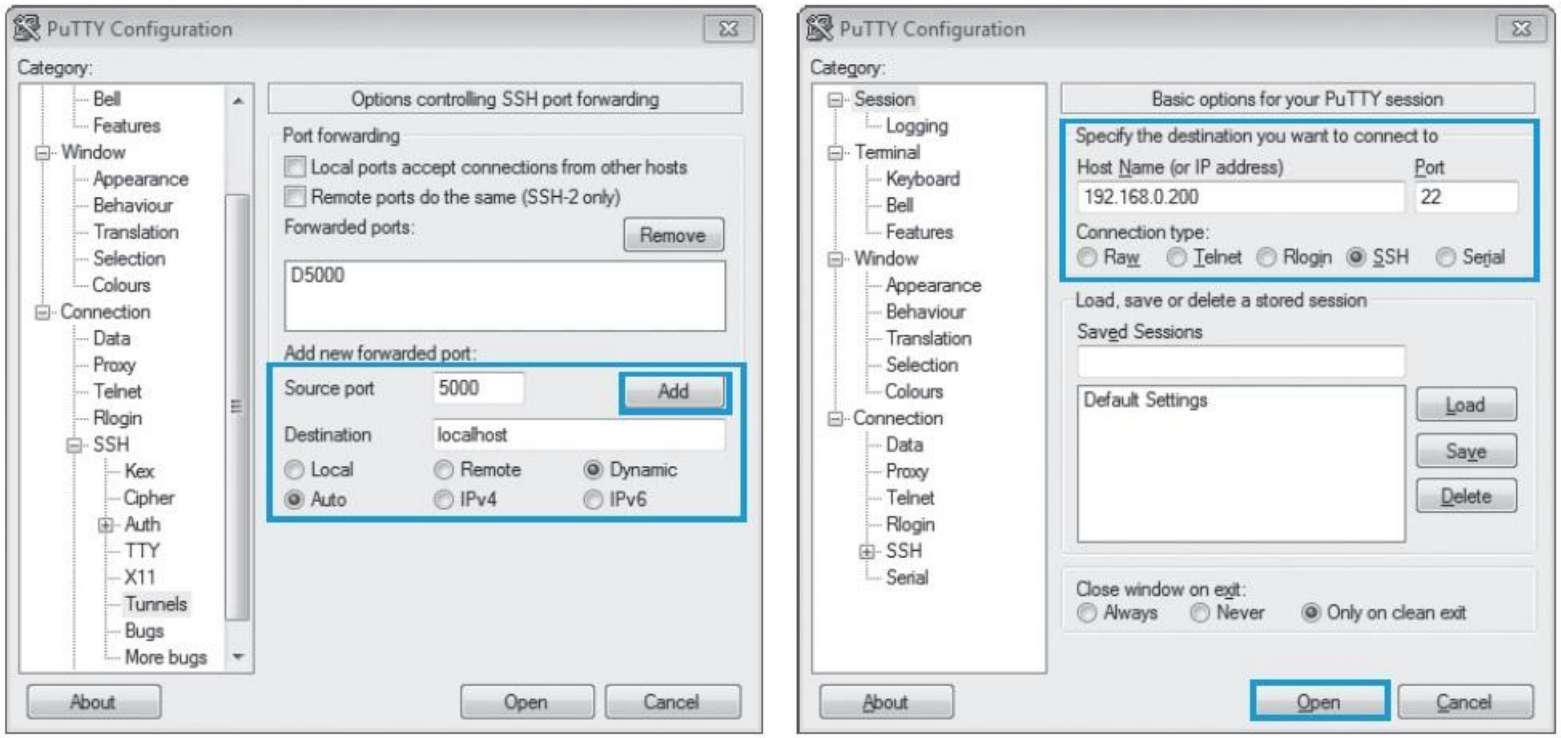


그림 8-27 SSH 터널링 생성

② SSH를 이용한 터널링 설정하기

- PuTTY를 이용한 SSH 터널링 완료 확인

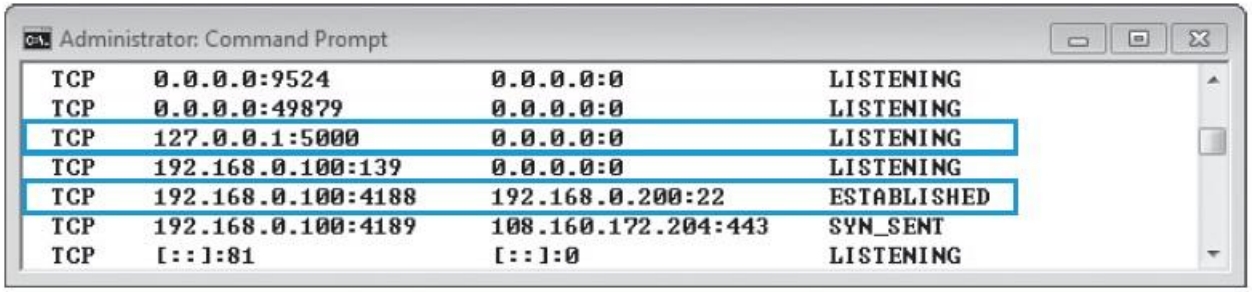


그림 8-28 SSH 터널링 세션 확인

③ 터널링을 이용해 웹 서핑하기

- 인터넷 익스플로러에서 [Tools(도구)]-[Internet Option(인터넷 옵션)]-[Connections(연결)] 탭 선택 후 'Local Area Network(LAN) settings' 항목의 <LAN settings> 버튼 클릭

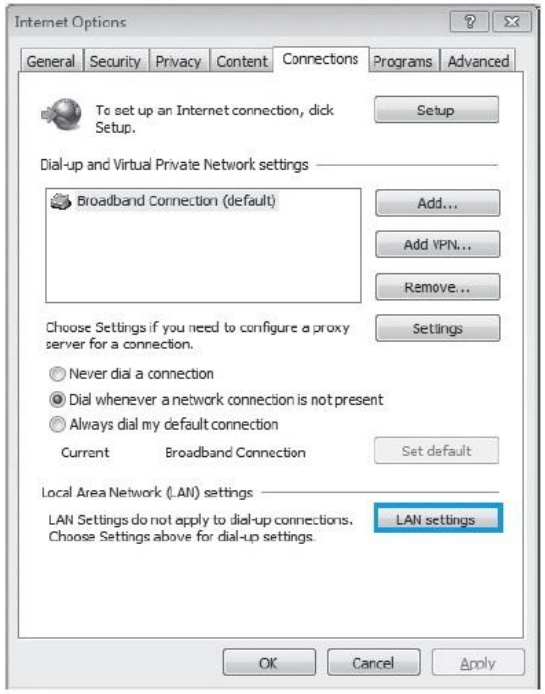


그림 8-29 랜 설정

③ 터널링을 이용해 웹 서핑하기

- Proxy 서버 사용을 선택한 뒤, <Advanced> 버튼 누름
- Socks 라인의 Proxy address to use 항목에 localhost, Port 항목에 5000을 입력

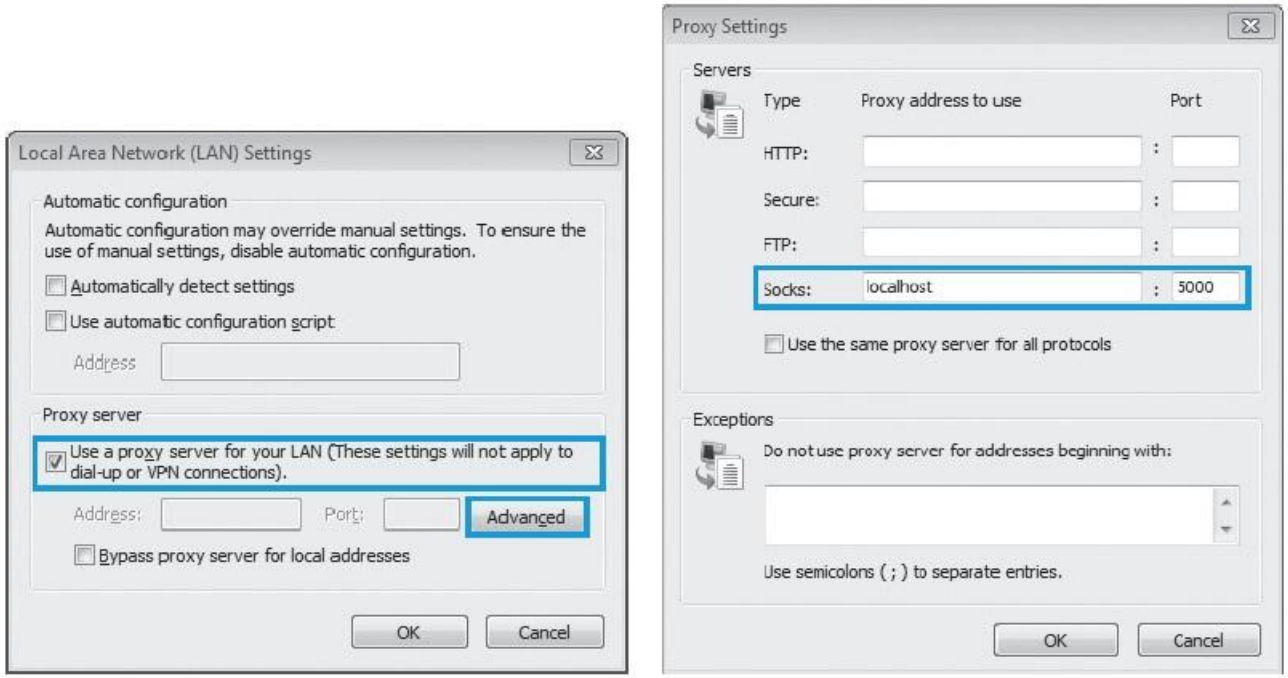


그림 8-30 프록시 설정

④ 터널링 상태 이해하기

- Wireshark를 실행하여 웹 서핑 시 어떤 패킷이 클라이언트에서 생성되는지 확인

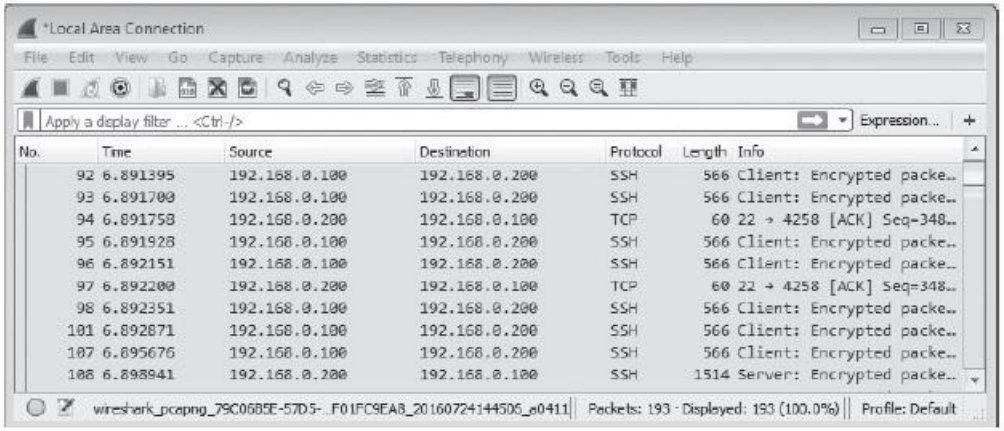


그림 8-32 SSH 터널링 패킷 캡처

④ 터널링 상태 이해하기

- 클라이언트에서 netstat 명령으로 네트워크 연결을 확인

netstat –an

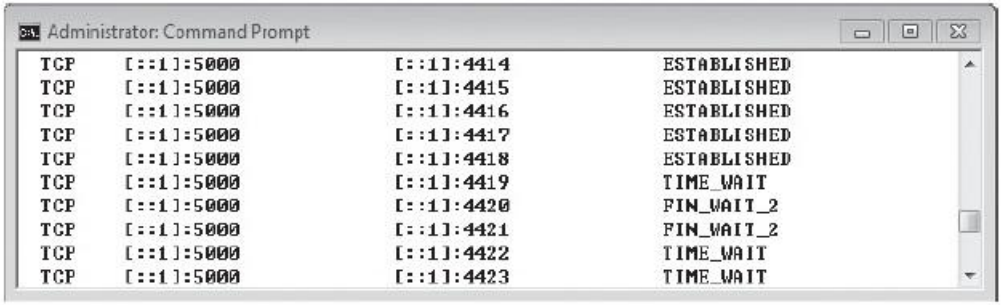


그림 8-33 클라이언트에서 netstat를 실행한 결과

④ 터널링 상태 이해하기

- SSH 서버 확인

  netstat -an | grep 80

```
root@ubuntu-14: /
root@ubuntu-14:/#
root@ubuntu-14:/#
root@ubuntu-14:/# netstat -an | grep 80
tcp        0      0 192.168.0.200:40720     222.122.117.35:80       ESTABLISHED
tcp        0      0 192.168.0.200:51454     182.162.92.32:80        ESTABLISHED
tcp        0      0 192.168.0.200:51424     182.162.92.32:80        ESTABLISHED
tcp        0      0 192.168.0.200:46632     182.162.92.36:80        ESTABLISHED
tcp        0      0 192.168.0.200:51426     182.162.92.32:80        ESTABLISHED
tcp        0      0 192.168.0.200:56006     182.162.92.33:80        ESTABLISHED
tcp        0      0 192.168.0.200:50394     1.255.49.46:80          ESTABLISHED
tcp        0      0 192.168.0.200:37280     125.209.226.239:80      CLOSE_WAIT
tcp        0      0 192.168.0.200:59072     182.162.92.37:80        ESTABLISHED
```

그림 8-34 SSH 서버에서 netstat를 실행한 결과

■ 은닉 채널(Covert Channel)

- 1973년 램프슨(Lampson)에 의해 정의된 용어
- 표면적인 목적 외의 정보나 은닉 메시지를 전송하기 위해 기본 통신 채널에 기생하는 통신 채널

■ ackcmd 툴

- ackcmd 툴은 ACK 패킷만 이용(세션을 성립시키지 않음).
- 실제로는 ACK 패킷 안에 숨겨진 데이터를 주고 받음.

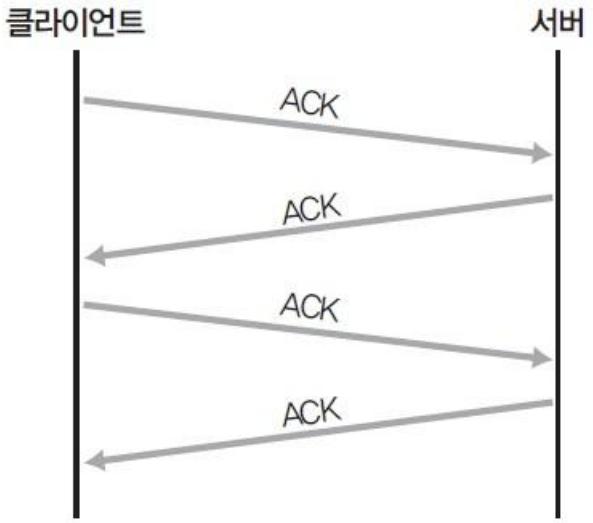


그림 8-35 ackcmd의 패킷 전송 과정

■ 은닉 채널과 방화벽 우회

- ackcmd는 공격자가 공격 대상 서버의 웹 서비스를 이용할 때 발생하는 것과 유사한 형태로 ACK 패킷을 발생시켜 서로 통신을 수행

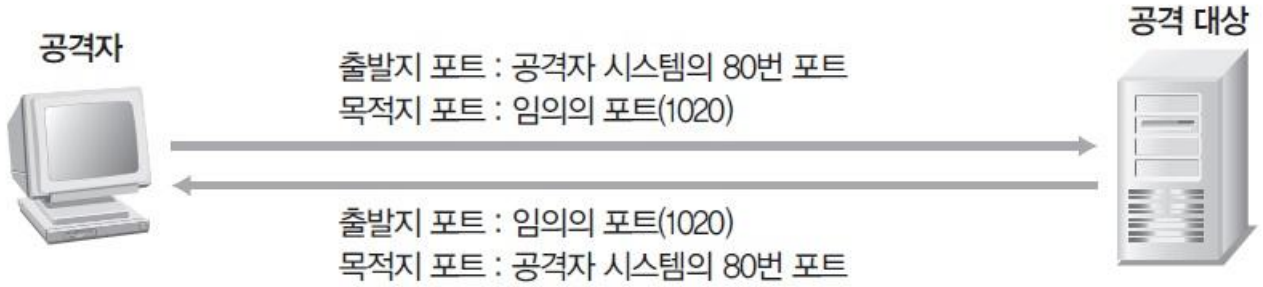


그림 8-36 ackcmd 패킷의 통신 포트

■ 은닉 채널과 방화벽 우회

- ackcmd 툴이 통신하는 패킷을 캡처해보면 ACK 패킷으로 통신을 수행하고 있음을 확인할 수 있음.

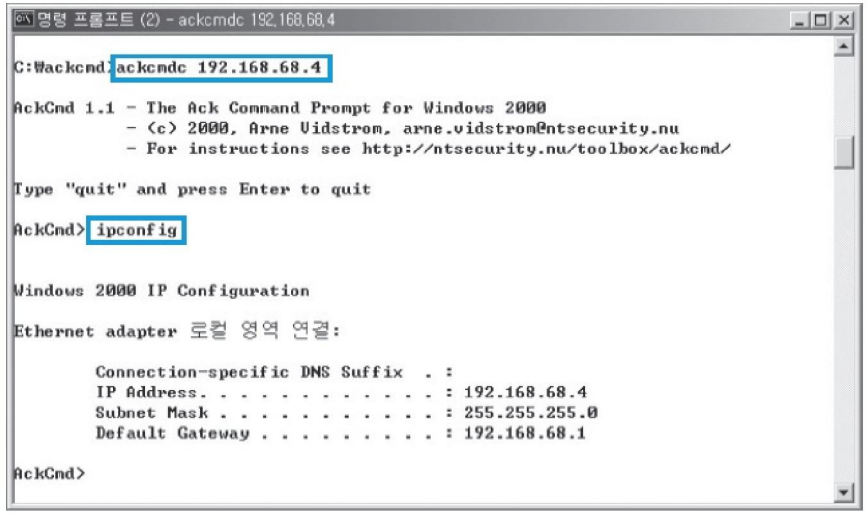


그림 8-37 ackcmd 접속 후 IP 확인 결과

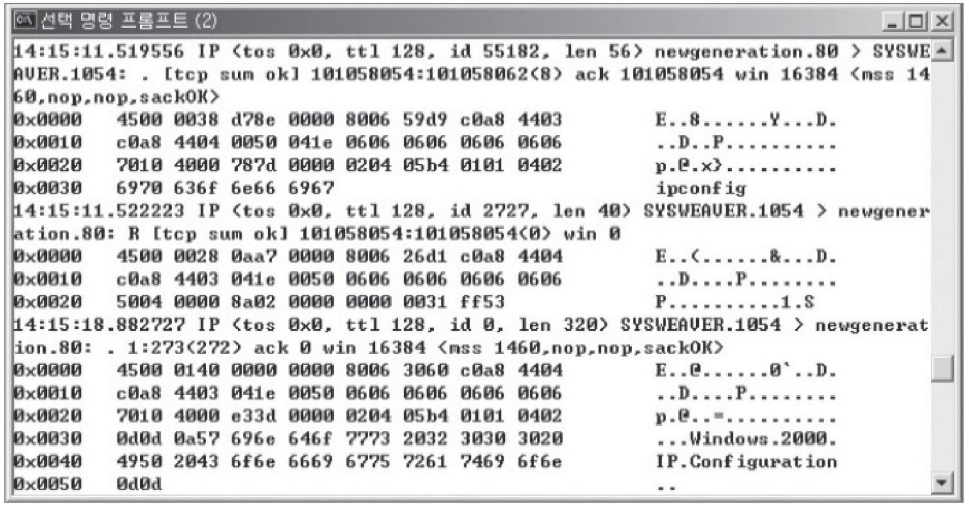


그림 8-38 ackcmd 통신 시 패킷 내용

2. 은닉 채널

실습 8-23 셸 백도어 설치하고 이용하기

실습환경
- DNS2TCP 클라이언트 시스템 : 우분투 데스크탑 14
- DNS2TCP 서버 시스템 : 우분투 서버 16
- 필요 프로그램 : DNS2TCP

① dns2tcp 설치하기

- dns2tcp의 서버와 클라이언트 설치

  (sudo) apt- get install dns2tcp

② dns2tcp 서버 실행하기

- dsn2tcpd_config 파일을 만들어 아래와 같이 설정

(sudo) vi ./dns2tcpd_config

```
root@ubuntu-S-16: /
listen = 0.0.0.0
port = 53
user = nobody
chroot = /home/nobody/
pid_file =/var/run/dns2tcp.pid
domain = dns2tcp.wishfree.com
key = secretkey
resources = ssh:127.0.0.1:22
                                                    1,1           All
```

그림 8-41 dns2tcpd_config 파일의 내용

② dns2tcp 서버 실행하기

- dns2tcp 서버는 dns2tcpd_config를 이용하여 다음과 같이 실행

(sudo) dns2tcpd - d 3 -f ./dns2tcpd_config

```
root@ubuntu-S-16: /
root@ubuntu-S-16:/# dns2tcpd -d 3 -f ./dns2tcpd_config
18:30:13 : Debug options.c:97   Add resource ssh:127.0.0.1 port 22
18:30:13 : Debug socket.c:55    Listening on 0.0.0.0:53 for domain dns2tcp.wishf
ree.com
root@ubuntu-S-16:/#
```

그림 8-42 dns2tcpd 실행

② dns2tcp 서버 실행하기

- netstat 명령을 통해 UDP 53 포트가 dns2tcpd에 의해 열려 있음을 확인
- 53번 포트가 다른 프로그램에 의해 이미 사용되고 있다면 dns2tcp 실행 시 에러 발생

```
root@ubuntu-S-16: /
root@ubuntu-S-16:/# netstat -anp | grep 53
udp        0      0 0.0.0.0:53              0.0.0.0:*
20232/dns2tcpd
unix  2      [ ACC ]     STREAM     LISTENING     12534    1/init              /
run/snapd.socket
unix  2      [ ACC ]     STREAM     LISTENING     12539    1/init              /
run/uuidd/request
unix  3      [ ]         STREAM     CONNECTED     35371    12680/vsftpd
unix  3      [ ]         STREAM     CONNECTED     35372    1/init              /
run/systemd/journal/stdout
root@ubuntu-S-16:/#
```

그림 8-43 dns2tcpd 실행 결과

③ dns2tcp 클라이언트 실행하기

- dns2tcp 클라이언트 실행을 위해 dsn2tcpc_config라는 파일을 다음과 같이 설정

```
root@ubuntu-14: /
domain = dns2tcp.wishfree.com
resource = ssh
local_port = 2222
key = secretkey
debug_level = 3
server = 192.168.0.2
```

그림 8-44 dns2tcpc_config 파일의 내용

- dns2tcpc_config를 이용하여 다음과 같이 실행

  (sudo) dns2tcpc - f ./dns2tcpd_config

```
root@ubuntu-14: /
root@ubuntu-14:/# dns2tcpc -f ./dns2tcpc_config
debug level 3
Debug socket.c:233      Create socket for dns : '192.168.0.2'
Listening on port : 2222
When connected press enter at any time to dump the queue
```

그림 8-45 dns2tcpc 실행

③ dns2tcp 클라이언트 실행하기

- local_port 항목으로 설정한 2222번 포트에 대한 연결 확인

```
root@ubuntu-14: /
root@ubuntu-14:/# netstat -anp | grep 2222
tcp        0      0 127.0.0.1:2222          0.0.0.0:*               LISTEN
28432/dns2tcpc
root@ubuntu-14:/#
```

그림 8-46 dns2tcpc 실행 결과

④ dns2tcp를 이용해 통신 연결하기

ssh wishfree@127.0.0.1 - p 2222 - D 6789

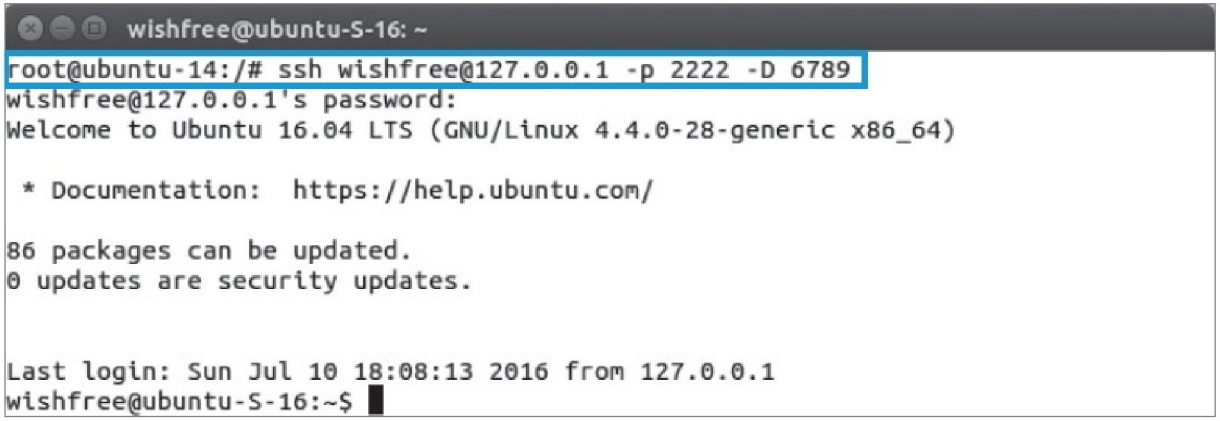


그림 8-47 dns2tcp를 통한 ssh 연결

옵션	내용
wishfree@127.0.0.1	로컬 시스템(127.0.0.1)의 wishfree 계정으로 로그인
-p 2222	목적지 포트는 2222번
-D 6789	출발지 포트는 6789번

④ dns2tcp를 이용해 통신 연결하기

- 연결 후 해당 패킷 확인

```
root@ubuntu-14: /
19:01:30.675572 IP (tos 0x0, ttl 64, id 18293, offset 0, flags [DF], proto UDP (
17), length 102)
    192.168.0.2.domain > 192.168.0.200.47356: 46337* 1/0/0 +FIB1wHvBA.dns2tcp.wi
shfree.com. TXT "A+FIAAAHvEA" "" (74)
        0x0000:  4500 0066 4775 4000 4011 70f7 c0a8 0002  E..fGu@.@.p.....
        0x0010:  c0a8 00c8 0035 b8fc 0052 e2ab b501 8580  .....5...R......
        0x0020:  0001 0001 0000 0000 0a2b 4649 4231 7748  .........+FIB1wH
        0x0030:  7642 4107 646e 7332 7463 7008 7769 7368  vBA.dns2tcp.wish
        0x0040:  6672 6565 0363 6f6d 0000 1000 01c0 0c00  free.com........
        0x0050:  1000 0100 0000 0300 0d0b 412b 4649 4141  ..........A+FIAA
        0x0060:  4148 7645 4100                           AHvEA.
```

그림 8-48 dns2tcp 통신 패킷

감사합니다.

네트워크 해킹과 보안 개정3판

정보 보안 개론과 실습