우분투 리눅스

시스템 & 네트워크

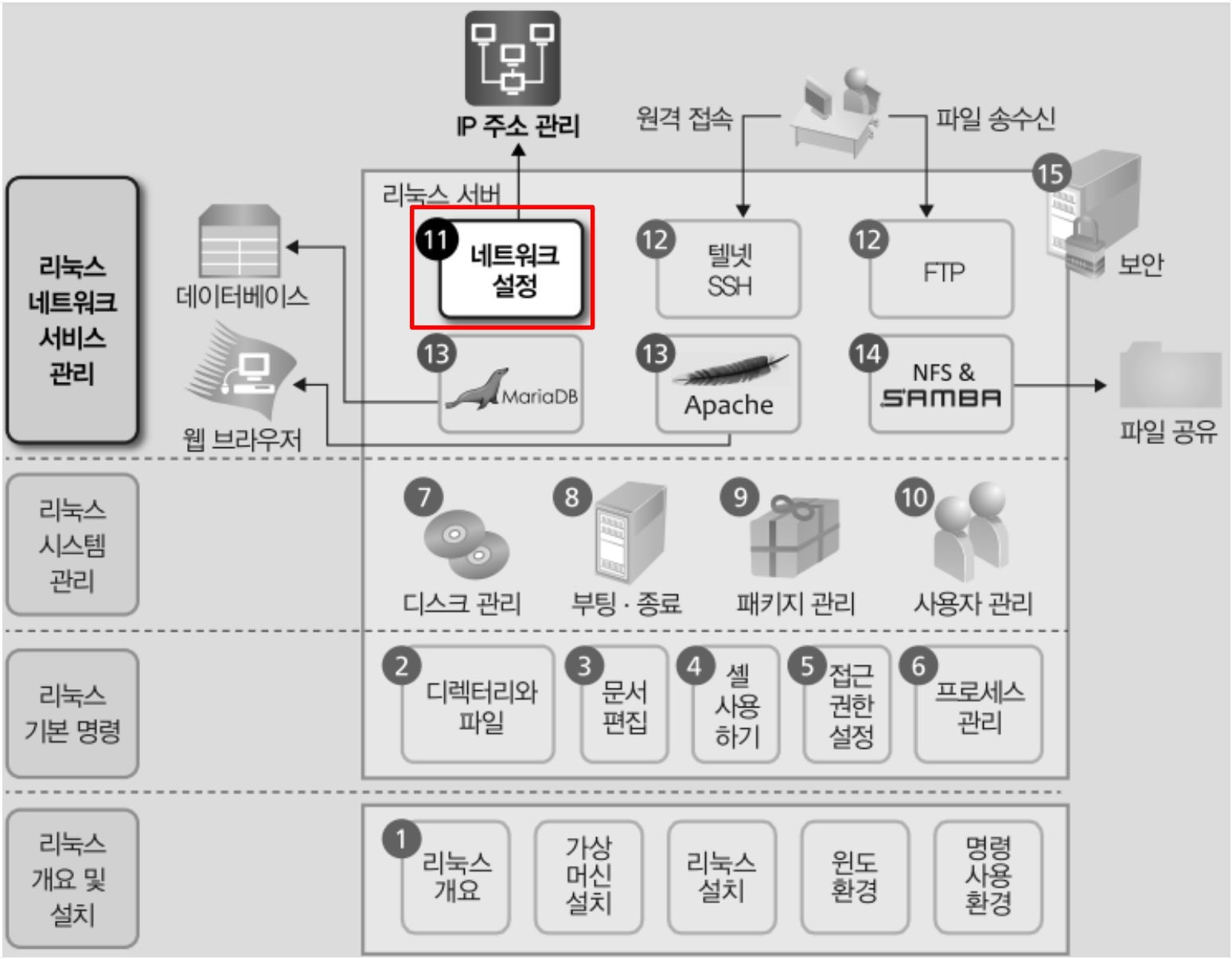
Chapter 11. 네트워크 관리

목차

학습목표

- TCP/IP 프로토콜의 계층 구조를 설명할 수 있다.
- MAC 주소와 IP 주소의 차이를 설명할 수 있다.
- 네트워크 인터페이스를 설정할 수 있다.
- 라우팅 테이블을 확인하고 기본 게이트웨이를 설정할 수 있다.
- DNS 설정을 확인하고 질의를 수행할 수 있다.
- ping과 traceroute 명령을 사용하여 통신이 가능한지 확인할 수 있다.
- 네트워크와 관련된 통계 정보를 확인할 수 있다.
- 같은 네트워크에 연결된 시스템의 MAC 주소와 IP 주소를 확인할 수 있다.
- 패킷을 캡처하여 저장하고 내용을 분석할 수 있다.

리눅스 실습 스터디 맵


IP 주소 관리
원격 접속
파일 송수신
리눅스 서버
15
보안
리눅스 네트워크 서비스 관리
데이터베이스
11 네트워크 설정
12 텔넷 SSH
12 FTP
웹 브라우저
13 MariaDB
13 Apache
14 NFS & SAMBA
파일 공유
리눅스 시스템 관리
7 디스크 관리
8 부팅 · 종료
9 패키지 관리
10 사용자 관리
리눅스 기본 명령
2 디렉터리와 파일
3 문서 편집
4 셸 사용하기
5 접근 권한 설정
6 프로세스 관리
리눅스 개요 및 설치
1 리눅스 개요
가상 머신 설치
리눅스 설치
윈도 환경
명령 사용 환경

00 개요

[그림 11-1] 11장의 내용 구성

01 네트워크의 기초

TCP/IP 프로토콜

- 프로토콜이란 컴퓨터와 컴퓨터 사이에서 데이터를 어떻게 주고받을 것인지를 정의한 통신 규약
- 인터넷이라고 부르는 네트워크는 TCP/IP라는 프로토콜에 따라 데이터를 주고받음
- TCP/IP 프로토콜은 [그림 11-2]와 같이 5계층으로 구성
- 전송 계층의 TCP와 네트워크 계층의 IP로 전체 프로토콜을 대표하여 TCP/IP 프로토콜이라고 함

[표 11-1] TCP/IP 프로토콜 모델의 계층별 역할과 대표 프로토콜

계층	기능	프로토콜	전송 단위
응용 계층	서비스 제공 응용 프로그램	DNS, FTP, SSH, HTTP, Telnet	메시지
전송 계층	응용 프로그램으로 데이터를 전달, 데이터 흐름 제어 및 전송 신뢰성 담당	TCP, UDP	세그먼트
네트워크 계층	주소 관리 및 경로 탐색	IP, ICMP	패킷
링크 계층	네트워크 장치 드라이버	ARP	프레임
물리 계층	케이블 등 전송 매체	구리선, 광케이블, 무선	비트

[그림 11-2] TCP/IP 프로토콜 모델

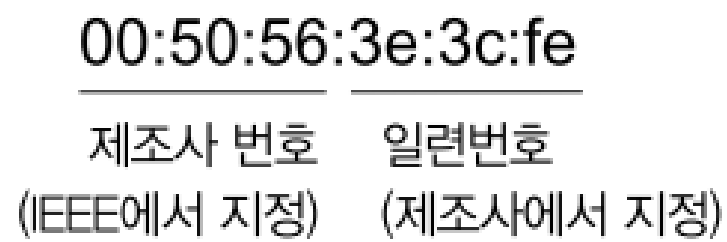
01 네트워크의 기초

주소

- 일반적으로 네트워크 인터페이스는 거의 대부분 이더넷(ethernet) 방식을 사용
- 컴퓨터의 주소는 MAC 주소, IP 주소, 호스트명이 있음

MAC 주소

- MAC는 'media access control'의 약자
- MAC 주소는 하드웨어를 위한 주소이며 다른 말로 이더넷 주소, 하드웨어 주소, 물리 주소라고도 함
- MAC 주소는 네트워크 인터페이스 카드(다른 말로 랜 카드)에 저장된 주소라고 생각하면 됨
- MAC 주소는 쌍점(:)이나 붙임표(-)로 구분되는 여섯 개의 16진수로 구성되며, 총 48비트
  - 앞의 세 자리는 제조사 번호, 뒤의 세 자리는 일련번호

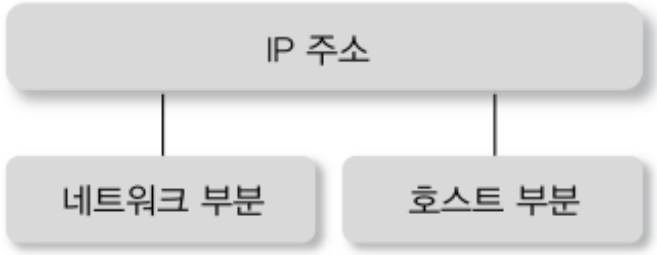


[그림 11-3] MAC 주소의 예

01 네트워크의 기초

IP 주소

- 우리가 보통 인터넷 주소라고 부르는 것이 IP(internet protocol) 주소
- IP 주소는 인터넷으로 연결된 네트워크에서 각 컴퓨터를 구분하기 위해 사용
- IP 주소는 1바이트의 크기를 가진 네 자리 숫자로 구성되므로 총 4바이트
  - 예를 들어 192.168.100.5와 같이 숫자 네 가지와 마침표(.)로 구성
- IP 주소는 네트워크를 구분하는 네트워크 주소 부분과, 해당 네트워크 안에서 특정 컴퓨터를 식별하는 호스트 주소로 구분
- IP 주소는 총 32비트(4바이트) 중 몇 비트를 네트워크 부분으로 사용하고 나머지 몇 비트를 호스트 부분으로 사용하는지에 따라 A 클래스, B 클래스, C 클래스로 구분
- C 클래스의 구조
  - 앞의 3바이트가 네트워크 부분
  - 뒤의 1바이트만 호스트 부분으로 사용
  - 호스트 부분으로 사용할 수 있는 숫자 1~254
  - 예: 192.168.100.5는 C 클래스이므로 네트워크 부분은 앞의 세 자리인 192.168.100이고, 뒤의 5는 호스트 부분

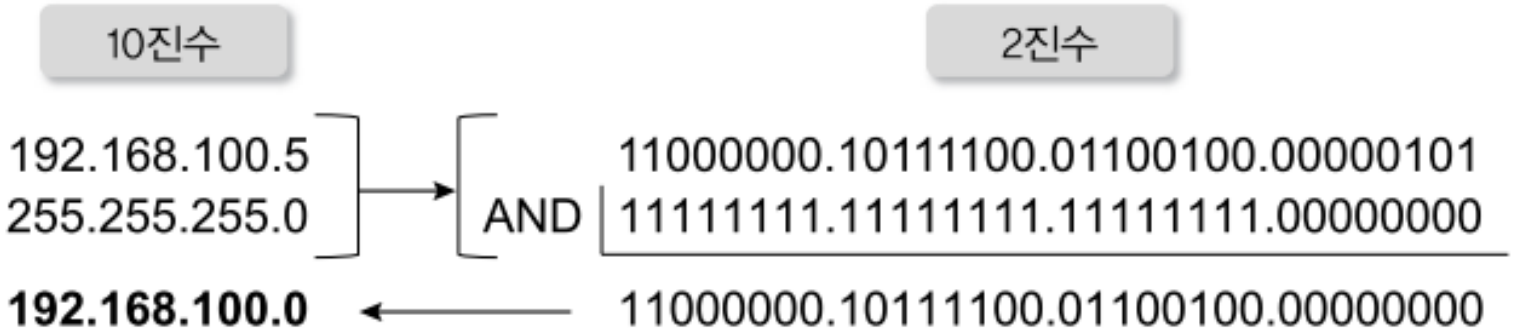


[그림 11-5] IP 주소의 구성

01 네트워크의 기초

넷마스크와 브로드캐스트 주소

- 넷마스크: IP 주소에서 네트워크 부분을 알려주는 역할
- 넷마스크는 하나의 네트워크를 다시 작은 네트워크(서브넷)로 분리할 때도 사용하는데, 그래서 서브넷 마스크라고 부르기도 함
- 넷마스크 예



[그림 11-6] 넷 마스크 계산의 예

  - IP 주소와 넷마스크를 10진수에서 2진수로 바꾼 다음, 두 값을 가지고 AND 연산을 수행
  - AND연산을 하면 네트워크 부분만 남고 호스트 부분은 0이 됨
  - 넷마스크는 IP 주소와 AND 연산을 수행하여 네트워크 부분만 남기는 역할
- 브로드캐스트 주소: 같은 네트워크에 있는 모든 컴퓨터에 메시지를 보낼 때 사용하는 것
  - 호스트 부분을 모두 1로 설정
  - 예: IP 주소에서 네트워크 부분이 192.168.100.0이면 브로드캐스트 주소는 192.168.100.255

01 네트워크의 기초

호스트 이름

- 사람은 숫자보다는 이름으로 된 것을 더 잘 기억한다. 그래서 나온 것이 호스트 이름.
- 호스트 이름도 IP 주소처럼 두 부분으로 구성
  - 예: 네이버를 예로 든다면 naver.com이 네트워크 부분, www가 호스트 부분에 해당

포트 번호

- 각 서비스를 구분하는 번호
- 포트 번호는 TCP/IP 프로토콜의 4계층인 전송 계층에서 사용하는 번호
- /etc/services 파일에 포트 번호 저장

```
user1@myubuntu:~$ cat /etc/services
(생략)
tcpmux    1/tcp                                 # TCP port service multiplexer
echo              7/tcp
echo              7/udp
discard   9/tcp           sink null
discard     9/udp         sink null
systat      11/tcp        users
(생략)
user1@myubuntu:~$
```

02 네트워크 설정

네트워크를 설정하려면

- IP 주소, 넷마스크와 브로드캐스트 주소, 게이트웨이(라우터) 주소, DNS 주소

호스트 이름 설정하기

- 호스트 이름 확인하기 : hostname, uname -n

uname

기능	시스템 정보를 출력한다.
형식	uname [옵션]
옵션	-m : 하드웨어 종류를 출력한다. -n : 호스트 이름을 출력한다. -r : 운영체제의 릴리즈 정보를 출력한다. -s : 운영체제의 이름을 출력한다. -v : 운영체제의 버전을 출력한다. -a : 위의 모든 정보를 출력한다.
사용 예	uname -n uname -a

```
user1@myubuntu:~$ uname -n
myubuntu
user1@myubuntu:~$
```

02 네트워크 설정

■ 호스트 이름 설정하기

- hostname 명령

hostname

기능	호스트 이름을 출력하거나 설정한다.
형식	hostname [호스트 이름]
사용 예	hostname　　hostname mail.han.server

- hostname 명령으로 호스트 이름을 검색

```
user1@myubuntu:~$ hostname
myubuntu
user1@myubuntu:~$
```

- hostname 명령으로 호스트 이름을 설정: root 권한 필요

```
[user1@myubuntu:~$ sudo hostname mail.han.server
[sudo] password for user1:
user1@myubuntu:~$ hostname
mail.han.server
user1@myubuntu:~$
```

02 네트워크 설정

호스트 이름 설정 파일

- 재시작해도 호스트 이름이 바뀐 상태를 유지하려면 호스트 이름을 설정하는 파일 자체를 수정해야 함
- 우분투에서 호스트 이름을 저장하는 파일은 /etc/hostname

```
user1@myubuntu:~$ cat /etc/hostname
myubuntu
user1@myubuntu:~$
```

02 네트워크 설정

■ 네트워크 인터페이스 설정하기

ifconfig

기능	네트워크 인터페이스의 IP 주소를 설정한다.
형식	ifconfig [인터페이스명] [옵션] [값]
옵션	-a : 시스템의 전체 인터페이스에 대한 정보를 출력한다. up/down : 인터페이스를 활성화 · 비활성화한다. netmask 주소 : 넷마스크 주소를 설정한다. broadcast 주소 : 브로드캐스트 주소를 설정한다.

- 현재 설치된 네트워크 인터페이스 설정 확인하기 : ifconfig

```
user1@myubuntu:~$ ifconfig
eth0  Link encap:Ethernet  HWaddr 00:0c:29:1c:d4:22
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1c:d422/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:53115 errors:1 dropped:0 overruns:0 frame:0
          TX packets:48387 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4535465 (4.5 MB)  TX bytes:5644687 (5.6 MB)
          Interrupt:19 Base address:0x2000
lo   Link encap:Local Loopback
(생략)
```

02 네트워크 설정

현재 설치된 네트워크 인터페이스 설정 확인하기 : ifconfig

- eth0의 설정 내용
  - MAC 주소(ether) : 00:0c:29:1c:d4:22
  - IP 주소(inet) : 192.168.0.2
  - 넷마스크(netmask) : 255.255.255.0
  - 브로드캐스트 주소(broadcast) : 192.168.0.255
  - IPv6 주소(inet6) : fe80::20c:29ff:fe1c:d422

특정 네트워크 인터페이스 설정 확인하기

```
user1@myubuntu:~$ ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:0c:29:1c:d4:22
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1c:d422/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:53247 errors:1 dropped:0 overruns:0 frame:0
          TX packets:48416 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4545848 (4.5 MB)  TX bytes:5648231 (5.6 MB)
          Interrupt:19 Base address:0x2000
user1@myubuntu:~$
```

02 네트워크 설정

네트워크 인터페이스 사용 해제하기 : down 옵션

```
user1@myubuntu:~$ sudo ifconfig eth0 down
user1@myubuntu:~$ sudo ifconfig eth0
eth0:    Link encap:Ethernet  HWaddr 00:0c:29:1c:d4:22
           BROADCAST MULTICAST  MTU:1500  Metric:1
           RX packets:53442 errors:1 dropped:0 overruns:0 frame:0
           TX packets:48494 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:4560904 (4.5 MB)  TX bytes:5656740 (5.6 MB)
           Interrupt:19 Base address:0x2000
user1@myubuntu:~$
```

- 인터페이스가 다운되었을 때 flags를 보면 UP과 RUNNING이 없어지고,
- 둘째 줄과 셋째 줄에 있던 IP 주소 관련 부분인 inet과 inet6이 출력되지 않음

네트워크 인터페이스 활성화하기 : up 옵션

```
user1@myubuntu:~$ sudo ifconfig eth0 up
user1@myubuntu:~$ ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:0c:29:1c:d4:22
          inet6 addr: fe80::20c:29ff:fe1c:d422/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:53448 errors:1 dropped:0 overruns:0 frame:0
          TX packets:48502 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4562658 (4.5 MB)  TX bytes:5658224 (5.6 MB)
          Interrupt:19 Base address:0x2000
user1@myubuntu:~$
```

02 네트워크 설정

네트워크 인터페이스 수동으로 설정하기

```
ifconfig 인터페이스명 IP 주소 netmask 넷마스크 주소 broadcast 브로드캐스트 주소
```

- 예: IP 주소를 기존의 192.168.0.14로 바꿔서 설정

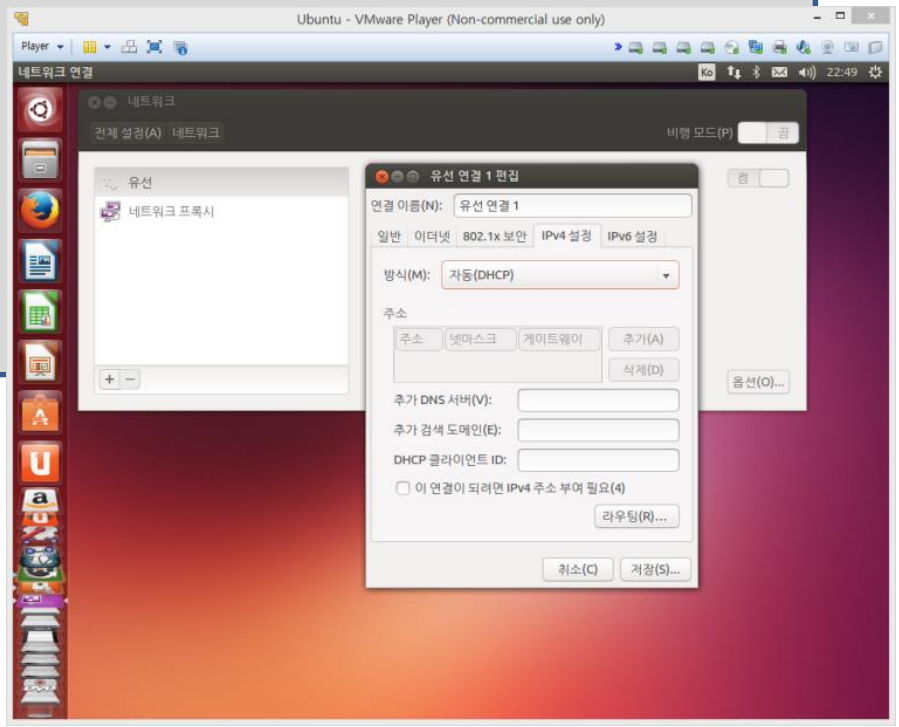
```
user1@myubuntu:~$ sudo ifconfig eth0 192.168.0.2 netmask 255.255.255.0
broadcast 192.168.0.255
user1@myubuntu:~$ ifconfig eth0
eth0  Link encap:Ethernet  HWaddr 00:0c:29:1c:d4:22
          inet addr:192.168.0.2  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe1c:d422/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:53665 errors:1 dropped:0 overruns:0 frame:0
          TX packets:48561 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4583513 (4.5 MB)  TX bytes:5665386 (5.6 MB)
          Interrupt:19 Base address:0x2000
user1@myubuntu:~$
```

02 네트워크 설정

네트워크 인터페이스 설정을 파일에 저장하기

- 부팅할 때 네트워크가 설정되게 하려면 /etc/network/interfaces 파일에 설정

```
user1@myubuntu:~$ sudo vi /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
iface eth0 inet static
address 192.168.0.10
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
dns-nameservers 168.126.63.1 168.126.63.2
~
:wq
user1@myubuntu:~$
```



- 네트워크 매니저 사용 가능

02 네트워크 설정

게이트웨이 설정하기

- 인터넷은 네트워크와 네트워크를 연결한 것
- 네트워크를 다른 네트워크와 연결할 때 연결점이 되는 장치가 게이트웨이
- 게이트웨이의 설정과 확인은 route 명령으로 수행

route

기능	라우팅 테이블을 편집하고 출력한다.
형식	route 명령
명령	add : 라우팅 경로나 기본 게이트웨이를 추가한다. del : 라우팅 경로나 기본 게이트웨이를 삭제한다.
사용 예	route route add default gw 192.168.0.1 dev eth0

라우팅 테이블 보기 : route

```
user1@myubuntu:~$ route
Kernel IP routing table
Destination  Gateway  Genmask   Flags   Metric   Ref  Use  Iface
default          192.168.0.1  0.0.0.0    UG      0        0  0  eth0
192.168.0.0      *                255.255.255.0     U       1        0  0  eth0
user1@myubuntu:~$
```

02 네트워크 설정

route 명령으로 라우팅 테이블 편집하기

[표 11-2] route 명령으로 라우팅 테이블 편집하기

기능	명령 형식과 사용 예
라우팅 경로 추가(네트워크)	route add -net 네트워크 주소 netmask 넷마스크 dev 인터페이스명 route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
라우팅 경로 추가(호스트)	route add -host 호스트 주소 dev 인터페이스명 route add -host 192.168.1.5 dev eth0
라우팅 경로 제거(네트워크)	route del -net 네트워크 주소 netmask 넷마스크 [dev 인터페이스명] route del -net 192.168.1.0 netmask 255.255.255.0
라우팅 경로 제거(호스트)	route del -host 호스트 주소 route del -host 192.168.1.5
기본 게이트웨이 추가	route add default gw 게이트웨이 주소 dev 인터페이스명 route add default gw 192.168.1.1 dev eth0
기본 게이트웨이 제거	route del default gw 게이트웨이 주소 route del default gw 192.168.1.1
루프백(lo) 추가	route add -net 127.0.0.0 netmask 255.0.0.0 dev lo

02 네트워크 설정

■ 라우팅 테이블의 출력 항목

[표 11-3] 라우팅 테이블의 출력 항목

항목	기능
Destination	라우팅 대상 네트워크나 호스트의 주소
Gateway	게이트웨이 주소 또는 설정되어 있지 않으면 *를 출력
Genmask	대상 네트워크의 넷마스크 255.255.255.255 : 대상이 호스트인 경우 0.0.0.0 : 기본(default) 경로
Flags	U : 경로 활성화(UP) H : 대상이 호스트 G : 게이트웨이로 사용 R : 동적 라우팅을 위한 경로 재생성 D : 데몬 또는 리다이렉트에 의해 동적으로 재설치 M : 라우팅 데몬 또는 리다이렉트에 의해 경로 수정 A : addrconf에 의해 설치 C : 캐시 항목 ! : 경로 거부
Metric	대상까지의 거리로 최근 커널에서는 사용되지 않지만 라우팅 데몬에서 사용할 수도 있다.
Ref	해당 경로에 대한 참조 수이지만 리눅스 커널에서는 사용하지 않는다.
Use	경로를 탐색한 수
Iface	패킷이 전달되는 인터페이스 이름

02 네트워크 설정

기본 게이트웨이 설정하기

```
user1@myubuntu:~$ sudo route add default gw 192.168.0.1 dev eth0
user1@myubuntu:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.0.1     0.0.0.0         UG    0      0        0 eth0
192.168.0.0     *               255.255.255.0   U     1      0        0 eth0
user1@myubuntu:~$
```

기본 게이트웨이 삭제하기

```
user1@myubuntu:~$ sudo route del default gw  192.168.0.1
user1@myubuntu:~$ route
Kernel IP routing table
Destination   Gateway   Genmask      Flags   Metric  Ref  Use   Iface
192.168.0.0    *          255.255.255.0  U       1       0  0   eth0
user1@myubuntu:~$
```

02 네트워크 설정

DNS 설정하기

- 호스트명을 IP 주소로 바꾸는 역할을 수행
- 만약 DNS가 설정되어 있지 않으면 이름으로 서버에 접속할 수 없으며 직접 IP 주소를 사용하여 접속 가능

DNS 서버 지정하기

- DNS 서버의 주소를 /etc/resolv.conf 파일에 저장

```
user1@myubuntu:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by
resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
user1@myubuntu:~$
```

- DNS 서버에 질의하기 : nslookup

nslookup

기능	DNS 서버와 대화식으로 질의하고 응답을 받는다.
형식	nslookup [도메인명]
사용 예	nslookup nslookup www.daum.net

```
user1@myubuntu:~$ nslookup
> www.hanb.co.kr
Server:         127.0.1.1
Address:        127.0.1.1#53
Non-authoritative answer:
Name:   www.hanb.co.kr
Address: 218.237.65.4
> exit
user1@myubuntu:~$
```

03 네트워크 상태 확인

통신 확인하기: ping

ping

기능	네트워크 장비에 신호(ECHO_REQUEST)를 보낸다.
형식	ping [옵션] 목적지 주소
옵션	-a : 통신이 되면 소리를 낸다. -q : 테스트 결과를 지속적으로 보여주지 않고 종합 결과만 출력한다. -c 개수 : 보낼 패킷 수를 지정한다.
사용 예	ping 192.168.0.1 ping -a www.naver.com

- 옵션 없이 사용하는 경우: 패킷은 기본적으로 56바이트의 크기 + 8바이트 헤더 정보

```
user1@myubuntu:~$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=1.52 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=1.25 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=1.28 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=1.41 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=64 time=1.40 ms
(생략)
```

03 네트워크 상태 확인

-q 옵션 사용하기

- -q 옵션을 사용하면 아무 메시지도 출력되지 않다가 +C로 종료하면 통계 정보만 출력

```
user1@myubuntu:~$ ping -q 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
^C
--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.193/1.612/2.273/0.408 ms
user1@myubuntu:~$
```

-c 옵션 사용하기

- -c 옵션을 사용하면 보낼 패킷 수를 지정

```
user1@myubuntu:~$ ping -c 3 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=5.77 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=7.68 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=1.50 ms
--- 192.168.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.501/4.987/7.684/2.585 ms
user1@myubuntu:~$
```

03 네트워크 상태 확인

도메인 이름을 사용하는 경우

```
user1@myubuntu:~$ ping www.hanb.co.kr
PING www.hanb.co.kr (218.237.65.4) 56(84) bytes of data.
64 bytes from 218.237.65.4: icmp_seq=1 ttl=55 time=5.99 ms
64 bytes from 218.237.65.4: icmp_seq=2 ttl=55 time=3.47 ms
64 bytes from 218.237.65.4: icmp_seq=3 ttl=55 time=3.57 ms
64 bytes from 218.237.65.4: icmp_seq=4 ttl=55 time=3.33 ms
^C
--- www.hanb.co.kr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 3.331/4.093/5.996/1.102 ms
user1@myubuntu:~$
```

- 시스템에 따라서 보안을 강화하기 위해 ping 패킷이 왔을 때 응답하지 않도록 설정하는 경우도 있으므로, ping으로 연결되지 않는다고 해서 무조건 해당 시스템이 동작하지 않는다는 것은 아니다

03 네트워크 상태 확인

■ 통신 경로 확인하기

traceroute

기능	목적지까지 패킷이 거치는 경로를 출력한다.
형식	traceroute 목적지 주소
사용 예	traceroute 192.168.0.1 traceroute www.naver.com

■ 정상으로 경로가 확인되는 경우

```
user1@myubuntu:~$ traceroute www.hanb.co.kr
traceroute to www.hanb.co.kr (218.237.65.4), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  2.062 ms  1.994 ms  3.048 ms
 2  175.116.52.1 (175.116.52.1)  6.285 ms  6.176 ms  6.084 ms
 3  58.234.17.21 (58.234.17.21)  4.296 ms  4.092 ms  3.985 ms
 4  58.225.77.21 (58.225.77.21)  3.885 ms  3.599 ms  3.478 ms
 5  58.229.119.29 (58.229.119.29)  5.485 ms 221.139.233.229 (221.139.233.229)
11.180ms 1.255.23.57 (1.255.23.57)  10.946 ms
 (생략)
 9  222.239.220.46 (222.239.220.46)  4.019 ms  4.824 ms  4.890 ms
10  218.237.65.4 (218.237.65.4)  4.805 ms  4.832 ms  9.233 ms
user1@myubuntu:~$
```

03 네트워크 상태 확인

중간 노드의 기관 확인: whois 명령 사용

- 예: 5번에 출력된 58.229.11.125가 어느 기관의 것인지?

```
user1@myubuntu:~$ whois 58.229.119.29
% [whois.apnic.net]
% Whois data copyright terms    http://www.apnic.net/db/dbcopyright.html
% Information related to '58.224.0.0 - 58.239.255.255'
inetnum:        58.224.0.0 - 58.239.255.255
netname:        HANANET
country:        KR
descr:           Hanaro Telecom, Inc.
(생략)
% Information related to '58.224.0.0 - 58.239.255.255'
inetnum:        58.224.0.0 - 58.239.255.255
netname:        broadNnet-KR
descr:           SK Broadband Co Ltd
(생략)
user1@myubuntu:~$
```

03 네트워크 상태 확인

■ 정상으로 경로가 확인되지 않는 경우

- 네트워크가 연결되지 않는 구간은 *가 출력: 통신에 장애가 있거나 traceroute 명령을 거부

```
user1@myubuntu:~$ traceroute www.naver.com
traceroute to www.naver.com (202.131.30.11), 30 hops max, 60 byte packets
 1  192.168.0.1 (192.168.0.1)  14.625 ms  14.354 ms  14.268 ms
 2  175.116.52.1 (175.116.52.1)  15.087 ms  14.892 ms  14.891 ms
 3  58.234.17.21 (58.234.17.21)  13.548 ms  13.471 ms  13.245 ms
 4  58.225.77.13 (58.225.77.13)  12.912 ms  13.152 ms  12.953 ms
 5  58.229.119.153 (58.229.119.153)  13.722 ms 1.255.23.49 (1.255.23.49)  13.521 ms
1.255.23.45 (1.255.23.45)  13.341 ms
 6  1.255.26.98 (1.255.26.98)  13.149 ms 58.229.12.230 (58.229.12.230)  8.739 ms
8.526 ms
 7  118.221.4.90 (118.221.4.90)  40.367 ms 219.254.67.158 (219.254.67.158)  40.158
ms
118.221.4.90 (118.221.4.90)  39.919 ms
 8  218.237.28.254 (218.237.28.254)  7.614 ms 202.179.176.10 (202.179.176.10)
39.389
ms  12.299 ms
 9  202.179.176.6 (202.179.176.6)  12.027 ms * *
10  * * *
11  * * *
12  * * *
(생략)
```

03 네트워크 상태 확인

네트워크 상태 정보 출력하기

netstat

기능 네트워크의 상태 정보를 출력한다.

형식 netstat [옵션]

옵션
- -a : 모든 소켓 정보를 출력한다.
- -r : 라우팅 정보를 출력한다.
- -n : 호스트명 대신에 IP 주소를 출력한다.
- -i : 모든 네트워크 인터페이스 정보를 출력한다.
- -s : 프로토콜별로 네트워크 통계 정보를 출력한다.
- -p : 해당 소켓과 관련된 프로세스의 이름과 PID를 출력한다.

사용 예
```
netstat -rn
netstat -s
```

03 네트워크 상태 확인

- **라우팅 테이블 확인하기 : -r 옵션**

```
user1@myubuntu:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags  MSS  Window  irtt  Iface
default          192.168.0.1  0.0.0.0      UG     0  0   0  eth0
192.168.0.0      *           255.255.255.0  U      0  0    0  eth0
user1@myubuntu:~$ netstat -rn
Kernel IP routing table
Destination     Gateway     Genmask         Flags  MSS  Window  irtt  Iface
0.0.0.0          192.168.0.1  0.0.0.0      UG  0  0  0  eth0
192.168.0.0      0.0.0.0      255.255.255.0  U  0  0  0  eth0
user1@myubuntu:~$
```

03 네트워크 상태 확인

현재 열려 있는 포트 확인하기

- 현재 통신이 진행 중인 서비스는 해당 서비스 포트가 LISTEN 상태

```
user1@myubuntu:~$ netstat -an | grep LISTEN
tcp     0        0 127.0.1.1:53          0.0.0.0:*          LISTEN
tcp     0        0 0.0.0.0:22            0.0.0.0:*          LISTEN
tcp     0        0 127.0.0.1:631         0.0.0.0:*          LISTEN
tcp     0        0 0.0.0.0:25            0.0.0.0:*          LISTEN
tcp6    0        0 :::22                 :::*               LISTEN
tcp6    0        0 ::1:631               :::*               LISTEN
tcp6    0        0 :::25                 :::*               LISTEN
unix    2      [ ACC ]     STREAM     LISTENING     44715    @user1-com.canonical.
Unity.Master.Scope.music.T11455560759694
(생략)
```

03 네트워크 상태 확인

- **현재 열려 있는 포트를 사용 중인 프로세스 확인하기 : -p 옵션**

```
user1@myubuntu:~$ sudo netstat -p | more
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        28      0 192.168.0.2:44360       productsearch.ubu:https CLOSE_WAIT
8899/gvfsd-http
tcp        28      0 192.168.0.2:40565       3rdpartymedia.ubu:https CLOSE_WAIT
8899/gvfsd-http
tcp        28      0 192.168.0.2:38463       productsearch.ubu:https CLOSE_WAIT
8899/gvfsd-http
tcp        28      0 192.168.0.2:38464       productsearch.ubu:https CLOSE_WAIT
8899/gvfsd-http
(생략)
```

03 네트워크 상태 확인

인터페이스별 네트워크 통계 정보 확인하기 : -i 옵션

- RX-OK, TX-OK는 정상적으로 주고받은 패킷의 개수
- RX-ERR, RX-DRP, RX-OVR, TX-ERR, TX-DRP, TX-OVR는 송수신 중에 오류가 발생한 패킷의 개수

```
user1@myubuntu:~$ netstat -i
Kernel Interface table
Iface   MTU    Met  RX-OK  RX-ERR   RX-DRP   RX-OVR  TX-OK   TX-ERR   TX-DRP   TX-
OVR  Flg
eth0    1500   0  8665  0  0  0  5363  0  0  0  BMRU
lo      65536   0  578  0  0  0  578  0  0  0  LRU
user1@myubuntu:~$
```

03 네트워크 상태 확인

프로토콜별 네트워크 통계 정보 확인하기 : -s 옵션

- 예: IP 프로토콜, ICMP 프로토콜, TCP 프로토콜별로 통계 정보를 출력

```
user1@myubuntu:~$ netstat -s
Ip:
    7426 total packets received
    0 forwarded
    0 incoming packets discarded
    7404 incoming packets delivered
    5866 requests sent out
    4 outgoing packets dropped
    84 dropped because of missing route
Icmp:
    99 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 14
        timeout in transit: 52
        echo replies: 33
    71 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 38
        echo request: 33
IcmpMsg:
        InType0: 33
        InType3: 14
        InType11: 52
        OutType3: 38
        OutType8: 33
(생략)
```

03 네트워크 상태 확인

MAC 주소와 IP 주소 확인하기: arp 명령

arp

기능	ARP 캐시 정보를 관리한다.
형식	arp [IP 주소]
사용 예	arp
	arp 192.168.0.1

- 옵션없이 사용할 경우: 현재 같은 네트워크에 연결되어 있는 시스템의 MAC 주소와 IP 주소를 출력

```
user1@myubuntu:~$ arp
Address                  HWtype  HWaddress           Flags Mask    Iface
192.168.0.17             ether   b4:b6:76:ab:f7:48   C             eth0
192.168.0.1              ether   00:26:66:f4:37:69   C             eth0
user1@myubuntu:~$
```

- 특정 시스템의 MAC 주소를 확인

```
user1@myubuntu:~$ arp 192.168.0.1
Address                  HWtype  HWaddress           Flags Mask    Iface
192.168.0.1              ether   00:26:66:f4:37:69   C             eth0
user1@myubuntu:~$
```

03 네트워크 상태 확인

■ 패킷 캡처하기 : tcpdump

tcpdump

기능 네트워크상의 트래픽을 덤프한다.

형식 tcpdump [옵션]

옵션
-c 패킷 수 : 지정한 패킷 수만큼 덤프 받고 종료한다.
-i 인터페이스명 : 특정 인터페이스를 지정한다.
-n : IP 주소를 호스트명으로 바꾸지 않는다.
-q : 정보를 간단한 형태로 보여준다.
-X : 패킷의 내용을 16진수와 ASCII로 출력한다.
-w 파일명 : 덤프한 내용을 지정한 파일에 저장한다.
-r 파일명 : 덤프를 저장한 파일에서 읽어온다.
host 호스트명 또는 주소 : 해당 호스트가 받거나 보낸 패킷만 덤프한다.
tcp port 번호 : 지정한 포트 번호 패킷만 덤프한다.
ip : IP 패킷만 덤프한다.
arp 192.168.0.1

사용 예
```
tcpdump
tcpdump -i eth0
tcpdump -i eth0 -w DUMP.out
tcpdump tcp port 22 and host 192.168.0.7
```

03 네트워크 상태 확인

옵션 없이 사용하는 경우

- 현재 시스템에서 주고받는 모든 패킷을 캡처하여 패킷의 헤더 부분 정보를 출력

```
user1@myubuntu:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:23:10.781499 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq
703408911:703409027, ack 3151930450, win 270, length 116
23:23:10.781867 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 116:168, ack
1, win 270, length 52
23:23:10.782143 IP 192.168.0.17.9611 > 192.168.0.2.ssh: Flags [.], ack 168, win 252,
length 0
23:23:10.782252 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 168:284, ack
1, win 270, length 116
23:23:10.782498 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 284:336, ack
1, win 270, length 52
(생략)
23:23:10.821886 IP 192.168.0.17.9611 > 192.168.0.2.ssh: Flags [P.], seq 1:53, ack
24532, win 253, length 52
^C
177 packets captured
177 packets received by filter
0 packets dropped by kernel
user1@myubuntu:~$
```

03 네트워크 상태 확인

캡처할 패킷 개수 지정하기 : -c 옵션

- 예: 패킷을 3개만 캡처

```
user1@myubuntu:~$ sudo tcpdump -c 3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:24:02.546272 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq
703435019:703435135, ack 3151931438, win 270, length 116
23:24:02.546670 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 116:168, ack
1,
win 270, length 52
23:24:02.546949 IP 192.168.0.17.9611 > 192.168.0.2.ssh: Flags [.], ack 168, win 252,
length 0
3 packets captured
18 packets received by filter
0 packets dropped by kernel
user1@myubuntu:~$
```

03 네트워크 상태 확인

캡처한 패킷 정보를 파일로 저장하기 : -w 옵션

- 예: 패킷 세 개를 캡처하여 dump.out 파일에 저장

```
user1@myubuntu:~$ sudo tcpdump -c 3 -w dump.out
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
3 packets captured
6 packets received by filter
0 packets dropped by kernel
user1@myubuntu:~$
```

- 패킷을 저장한 파일이 바이너리 파일이기 때문에 그냥 열면 내용을 볼 수 없음

```
user1@myubuntu:~$ file dump.out
dump.out: tcpdump capture file (little-endian) - version 2.4 (Ethernet, capture
length 65535)
user1@myubuntu:~$ cat dump.out
▒▒  ▒▒▒▒▒▒▒▒.▒▒2S▒▒▒▒▒▒jj▒▒▒▒▒▒v▒▒▒▒▒▒H
▒▒▒ISC?▒▒▒▒▒▒K▒▒▒▒▒▒W▒▒▒▒▒▒<▒▒▒▒▒▒f▒▒▒;r▒▒▒?r▒▒▒▒▒▒X▒▒▒▒▒▒c▒▒▒~▒▒▒▒▒▒W▒▒▒sZ.▒▒▒2S▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒v▒▒▒▒▒▒H
J▒▒▒}BI▒▒▒o▒▒▒M:▒▒▒▒▒▒%▒▒▒)헏▒▒▒P▒▒▒_▒▒▒Fob▒▒▒₩
▒▒▒7@Yx▒▒▒h_▒▒▒3_▒▒▒▒▒▒▒▒▒▒"▒▒▒▒▒▒▒▒▒'R▒▒▒d▒▒▒▒▒▒▒▒▒▒D{▒▒▒!L▒▒▒
▒▒▒▒▒▒D▒▒▒B▒▒▒{0▒▒▒▒▒▒▒▒p▒▒▒.▒▒▒2S▒▒▒▒▒▒<<▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒v
▒▒▒▒▒▒▒▒v▒▒▒▒▒▒H▒▒▒▒▒▒user1@myubuntu:~$
```

03 네트워크 상태 확인

캡처한 패킷 파일 읽기 : -r 옵션

- 예: dump.out 파일의 내용 확인

```
user1@myubuntu:~$ sudo tcpdump -r dump.out
reading from file dump.out, link-type EN10MB (Ethernet)
23:24:46.968687 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq
703436635:703436687, ack 3151932166, win 270, length 52
23:24:46.969175 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 52:168, ack
1,
win 270, length 116
23:24:46.969376 ARP, Request who-has 192.168.0.2 tell 192.168.0.17, length 46
user1@myubuntu:~$
```

03 네트워크 상태 확인

특정 포트로 송수신되는 패킷 캡처하기 : tcp port 옵션

- 예: 192.168.0.17과 주고받는 패킷 중에서 22번 포트에 해당하는 패킷 세 개를 캡처

```
user1@myubuntu:~$ sudo tcpdump -c 3 tcp port 22 and host 192.168.0.17
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:26:50.146729 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq
703446183:703446299, ack 3151939030, win 270, length 116
23:26:50.147119 IP 192.168.0.2.ssh > 192.168.0.17.9611: Flags [P.], seq 116:168, ack
1, win 270, length 52
23:26:50.147388 IP 192.168.0.17.9611 > 192.168.0.2.ssh: Flags [.], ack 168, win 255,
length 0
3 packets captured
15 packets received by filter
0 packets dropped by kernel
user1@myubuntu:~$
```

03 네트워크 상태 확인

- **캡처한 내용을 ASCII로 보기 : -X 옵션**

```
user1@myubuntu:~$ sudo tcpdump -Xqr dump.out
reading from file dump.out, link-type EN10MB (Ethernet)
23:24:46.968687 IP 192.168.0.2.ssh > 192.168.0.17.9611: tcp 52
        0x0000:  4510 005c 6bad 4000 4006 4d7b c0a8 0002  E..₩k.@.@.M{....
        0x0010:  c0a8 0011 0016 258b 29ed 975b bbde ab06  ......%.)..[....
        0x0020:  5018 010e f9f4 0000 28f1 2114 c78c d9be  P.......(.!.....
        0x0030:  b246 0db5 6c53 433f 8ec1 4b9d fb57 a5dd  .F..lSC?..K..W..
        0x0040:  3cbe 8866 cc9c d83b 72ca 3f72 cfdd 58ff  <..f...;r.?r..X.
        0x0050:  9163 cf7e eff9 ada7 57f9 735a               .c.~....W.sZ
(생략)
```

우분투 리눅스

시스템 & 네트워크