2007년 정보보호 졸업작품 보고서

리눅스기반 NAT 방화벽 구현 및 각종 서버 운용

팀명 :Big Bang

팀원 :김창배

이관우

권용호

신영택

2007. 11

중부대학교 정보보호학과

목 차

그 림 목 차

요 약

국내뿐만 아니라 세계적으로 인터넷 사용이 급증하면서 상대적으로 제한되어 있는 IP주소가 부족한 현상이 발생하고 있다. 부족한 IP주소를 보완하기 위해서 기업이나 국가기관에서는 NAT(Network Address Translation)라는 네트워크주소 변환 기능을사용 하고 있다.

NAT는 IPv4 환경에서 모자라는 공인 IP주소를 이용하여 사설망을 구성한다. 이때 사설IP로는 인터넷을 사용할 수 없는 사설IP주소(10.0.0.0/255.0.0.0, 172.16.0.0/255.240.0.0, 192.168.0.0/255.255.0.0)를 사용한다. 사설망에 있는 호스트에서 인터넷에 접속을 하거나 인터넷망에서 사설망의 호스트에 접속하기 위해서 NAT기능이 필요하다.

NAT는 특정한 IP 주소를 한 그룹에서 다른 그룹으로 매핑하는 기능이다. 주소를 N-to-N 형태로 매핑하는 경우를 정적 NAT라 하고 M-to-N(M>N)를 동적 NAT라고 한다. 네트워크 주소 포트 변환은 기본 NAT의 확장기능으로 여러가지 네트워크 주소와 TCP/UDP 포트를 단일의 네트워크 주소와 TCP/UDP 포트로 변환한다.
N-to-1 매핑이라고 하며 리눅스에서 IP 마스커 레이딩도 이러한 방식을 이용한다. 이러한 기능을 이용하여 많은 중소기업 및 학과 등 각처에서 여러가지 용도로 리눅스서버가 활용되고 있다.

이번 졸업작품에서 우리는 리눅스의 IP 마스커레이딩을 이용하여 부족한 IP주소를 보완 할 수 있는 NAT서버를 구축, 간단한 망을 구축하여 웹서버, FTP서버 등의 기능을 활용할 계획이다. 또한 이러한 웹서버, FTP서버 등을 NAT서버망 안에서 사용할 때에 리눅스 서버의 보안성을 위해서는 미리 제공 되는 네트워크 툴들로 보안성을 검사하는 것이 필요하다. 우리는 여기서 IPtables을 이용하여 간단하게 기본적인 보안 상태를 체크하고 방화벽을 구축하여 서버들의 보안성을 확보할 계획이다.

1. 서 론

1.1 NAT의 필요성

폭발적인 TCP/IP 네트워킹의 증가와 이에 따른 TCP/IP네트워상의 모든 호스트들에 할당된 고유 주소체계의 지속적인 할당으로 인하여 가용한 IP 주소 공간은 빠른 고갈을 초래하였다. 실제 가용한 TCP/IP 주소는 조만간 동이나고 말 것이다.

글로벌 인터넷으로의 연결을 위하여 고유의 주소가 요구되기 때문에 이러한 문제점들은 새로운 대형 네트워크의 신규 연결에 있어서 심각한 문제가 아닐 수 없다. 현재 IPng(IP:next generation)가 장기적인 해결책으로 고려되어지고 있긴 하나 신표준 안의 선택 과정에 이르기까지는 아마 수년이 걸릴 것으로 예상되고 있다.

이에 따라 전략적 방안이 현재의 IP 표준의 수명을 최대한 연장하기 위해 NAT가 출현하였다. NAT는 인터넷 주소의 고갈로 인해 발생되는 단점을 극복하기 위해 동적 및 정적 주소 할당 메카니즘을 이용하여 사설망(Private Network)을 글로벌망(Global Network)으로 사용할 수 있도록 해주는 주소 변환 장치인 NAT를 사용한다.

1.2 내부망의 각 서버 보안성

내부망의 각 서버들은 가설 IP를 쓰고 있지만 외부 서비스를 위해서 공인IP로 주소를 변경하여 서비스하게 된다. 이 때 외부에서 침입할 경우 공인IP로 접근을 하여도 실질적으로 각 서버들의 IP는 가설 IP이므로 접근 할 수가 없게 되어 보안성에서 안정적이다.

1.3 본 졸업작품의 목표 및 추진방법

1.3.1 목표

- IP주소 고갈문제를 NAT 기술 방법으로 해결
- NAT 내부망 각 서버들의 보안성 및 관리 편의성 추구
- 리눅스 각종 서버 구축 및 운영기술 습득
- 웹서버를 이용한 동영상 서비스

1.3.2 추진 방법

리눅스 기반으로 iptables를 이용하여 NAT 망을 구축하고, 망 내에 웹서버, FTP서버, 내부사용자 PC를 구성한다. 웹서버에는 동영상 관련 서비스를 제공하고 FTP 서버는 내·외부 사용자들에게 자료 공유를 제공한다. 위와 같은 각 서버들의 패킷을 통제하기 위해 NAT 망을 두어 패킷을 통제 한다. 또한 외부 사용자는 내부망의 각 서비스를 사용 가능케 한다.

2. 기반기술

2.1 리눅스

2.1.1 정 의

리눅스(Linux)는 유닉스(Unix)에 상응하는 강력한 운영체제로서 인텔(Intel) 386 호환 컴퓨터와 매킨토시(Power PC), 썬 스팍(Sun SPARC), 디지탈 알파(DEC ALPHA)등의 시스템에서 동작하며 진정한 다중처리, 가상 메모리, 공유 라이브러리, 요구 메모리 적재, 뛰어난 메모리 관리 시스템, 그리고 강력한 TCP/IP 네트워킹을 지원한다.

리눅스는 현재 운영체제의 인터페이스 표준인 POSIX 운영체계 규격에 따라 유닉스의 두 가계인 SYSV와 BSD 확장을 덧붙여 완전히 새로 만든 독립적인 창작품(Independent implementation)이다. 소유권 문제가 있는 코드는 전혀 들어 있지 않으며, 리눅스는 "GNU(GNU's Not Unix) 공개 라이센스(GPL: GNU Public License)" 정신에 의거해 자유롭게 배포될 수 있다.

2.1.2 특 징

○ 진정한 다중 사용자, 다중 처리 시스템

다수의 사용자들이 각각 하나 이상의 여러 개의 애플리케이션을 동작시킬 수 있으며, 각각의 응용 프로그램은 다른 응용 프로그램에 보호적으로 동작하여 하나의 응용 프로그램 오작동이 전체 시스템을 다운시키는 것을 완벽히 보호한다. 리눅스는 하드웨어 드라이버 구동을 시스템 차원에서 철저히 분리하여 관리함과 동시에 응용 프로그램의 하드웨어 드라이버 접근을 철저히 시스템의 감시 하에 둠으로서 시스템을 완벽하게 보호하며 시스템의 안정성을 보장한다.

○ 뛰어난 신뢰성, 동급 최고의 성능

기업환경에서 강력한 서버 제품군에 주로 사용되는 유닉스의 기본 설계에 따라 더욱 효율적인 설계방식을 추가하여 시스템의 자원을 아주 효율적으로 사용하여, 가장 많이 사용되는 PC 서버에서도 엔터프라이즈급의 성능과 안정성을 발휘할 수 있다.

○ 폭 넓은 하드웨어 장치 지원

일반적인 유닉스 기반 운영체제는 제작사의 하드웨어에서만 동작하도록 만들어져 있어서 지원되는 주변장치나 하드웨어들이 극히 적다. PC를 비롯한 다양한 하드웨어에서 돌아가는 유닉스의 경우에도 역시 지원되는 하드웨어 수가 그리 많지 않은 것이 현실이다. 리눅스의 경우 유닉스 기반의 운영체제로는 가장 많은 수의 하드웨어를 지원한다. 운영체제 커널의 소스가 공개가 되어 있기 때문에 하드웨어 지원이 필요

할 경우,다양한 채널을 통해 하드웨어 드라이버가 제작, 추가되어 현재에 이르렀는데 일반적인 PC 수준의 널리 알려진 하드웨어의 경우 대부분이 지원되고 있다.

윈도우즈 NT 뿐만 아니라 서버용 OS중에서 하드웨어 장치 지원 부분에서는 비교할 수 없을 정도로 우위에 있다.

○ 뛰어난 안정성과 보안성

유닉스의 대용품으로 사용되는 윈도우즈 NT와 SCO OpenServer, UnixWare, BSDi의 BSD/OS보다 확실히 안정성이 높고 보안성이 뛰어나다. 리눅스에서의 버그와 보안 결점은 단 몇 시간 안에 보고되지만, 그 외의 상용 OS는 패치나 버그 수정을 발표하는 데 걸리는 시간만 해도 몇 달이 소요된다. 대표적인 예로, PING 보안 헛점에 의한 버그의 경우, 리눅스는 단 4시간만에 보고되어 수정되었지만, 마이크로소프트 윈도우즈 NT의 경우 자사의 서버가 이 버그로 인해 서비스를 못하게 되어서 몇 주 지나서야 이러한 문제를 시인했다. 아울러, 리눅스는 세계에서 가장 효율적인 네트워크 OS를 만들며 네트워크 코드를 향상시키는 작업을 상시적 으로 하고 있는 엄청난 수의 전문 프로그래머들이 지원을 아끼지 않고 있다.

○ 다양한 업무 환경을 만족시키는 다양한 배포판의 존재

리눅스에는 서로 다른 여러 리눅스 배포본이 있으며, 기본적인 내용은 동일하다. 그 중에서 가장 많이 사용되고, 완벽하며, 안정적인 배포판은 레드햇(RedHat), 데비안(Debian), 그리고 슬랙웨어(Slackware) 등이며 이러한 배포판들의 차이는 시스템 관리방식 및 구성되는 응용 프로그램의 종류나 범위 등에 따라 차이가 나게 된다. 따라서 고객이 필요한 분야에 적합한 배포판을 구해 설치함으로써 자신에게 필요한 응용 프로그램을 인터넷에서 다운 받아 추가적으로 설치해야 하는 번거로움을 줄일 수가 있다. 레드햇과 데비안 배포본의 경우 일반적인 유닉스 시스템에서는 지원하지 않는 새로운 패키지 관리 시스템을 도입하여 응용 프로그램의 업그레이드로 인한 시스템 구성 요소의 의도하지 않았던 삭제나 변형을 방지하기 때문에 시스템의 안전한 업그레이드를 보장하며 새로운 소프트웨어의 설치를 쉽게 할 수 있도록 도와준다.

○ 다양하고 완벽한 네트워킹 기능

하드웨어 수준에서 가장 널리 쓰이는 이더넷(Ethernet), IBM사의 ARCnet, FDDI(광 케이블 인터페이스) ISDN, 심지어 아마추어 HAM 라디오를 이용한 네트워크를 구성하는 데 필요한 AX.25를 위한 드라이버를 지원하며, 최근에는 네트워크을 이용한 병렬처리 컴퓨터에 리눅스가 사용되면서 여기에 부산물로 Gigabyte 이더넷 드라이버와 ATM 드라이버가 나오기 시작하고 있어 리눅스 2.2버전에서부터 지원되고 있다. 현재 최다 프로토콜의 지원과 가장 많은 네트워킹 서비스와 기능을 제공한다. 하드웨어 계층의 바로 윗 계층인 전송 층에서 현재 인터넷 비지니스에서 널리 사용되고 있는 TCP/IP 프로토콜은

물론, 노벨(Novell)사의

넷웨어(Netware)를 위한 IPX 프로토콜, IBM의 SNA 프로토콜, 애플(Apple)사의 AppleTalk 프로토콜, 마이크로소프트 윈도우즈 워크그룹에 사용되는 SMB 프로토콜 등을 사용하여 특정 프로토콜만을 지원하는 시스템간의 브리지 서버(Bridge Server) 역할을 할 수 있다. 부가적으로 저렴한 모뎀을 이용한 PPP프로토콜을 지원하여 PPP 서버로서의 기능을 수행 할 수 있다.

이렇게 가장 활용성이 높은 서비스를 운영체제 수준에서 기본적으로 제공하고 여기에 이러한 프로토콜을 이용하여 많은 부가적인 서버로서의 기능을 수행하게 된다. 기본적으로 웹 서버, FTP 서버, NFS 서버 등을 제공하며, 디렉토리 서비스를 제공하는 파일 서버와 프린터 서버, 팩스 서버 및 뉴스서버 등의 역할을 아주 훌륭히 수행하고 있다.

○ 다양하고 완벽한 네트워킹 기능

일반적으로 운영체제는 자기 고유의 파일 시스템만을 지원하는 것이 보통이다. 이것은 다른 시스템에서 만들어진 자료 기록을 읽을 수 없게 만든다. 파일 시스템상의 자료 중 대부분은 네트웍을 통해 전송이 되기도 하지만 그렇지 못한 경우 다른 시스템에서 만들어진 자료를 읽을 방법이 없다. 리눅스를 위해 만들어져 경이적인 성능을 발휘하는 EXT2 파일 시스템은 기술적으로나 실용적으로나 여타의 파일 시스템에 비해 압도적인 퍼포먼스와 안정성을 자랑한다. 리눅스는 자신의 파일 시스템 외에도 공통적으로 사용되는 다양한 파일 시스템을 지원한다. 윈도우즈 NT의 NTFS를 비롯하여 윈도우즈 95의 VFAT, DOS의 FAT 파일 시스템, 그리고 노벨 넷웨어를 이용한 원격 파일 시스템을 비롯한 공통적으로 많이 사용되는 것들과 CD-ROM에서 사용하는 ISO 9660 파일 시스템, OS/2의 HPFS과 SCO, Coherent 같은 상용 유닉스 파일 시스템, 아미가(Amiga) 컴퓨터에서 사용되는 FFS, Sun OS, Free BSD, Net BSD, NextStep의 파일 시스템인 UFS 및 교육용 유닉스인 미닉스 파일 시스템의 기본적인 지원으로 해당 시스템에서 만들어진 자료의 기록 매체로부터 데이터를 직접 읽어들일 수 있음으로써, 가히 현존하는 서버용 OS 중에서 가장 높은 유연성을 보유하고 있다.

○ 풍부한 응용프로그램의 제공

리눅스에는 컴퓨터 하드웨어와 직접 교신하는 소프트웨어인 커널과 표준 응용 프로그램 모음이 포함되어 있다. 표준 리눅스 설치는 기본 운영체제와 X윈도우 시스템(전체 그래픽 사용자 인터페이스 포함), 네트워크 도구(FTP, WWW, IRC, 그리고 NEWS 등의 서버와 클라이언트)와 같은 많은 응용 프로그램, 마이크로소프트 윈도우 응용 프로그램 실행기(WINE)와 도스 에뮬레이터(DOSEMU), TeX와 같은

조판 시스템, 편집기(Emacs, Joe, Jed, Vi, 그리고 Pico), 개발 도구(그래픽 전위 프로그래밍을 가능하게 해주는 GTK+와 Tcl/Tk, 자바 환경을 제공하는 JDK, GNU C/C++의 컴파일러인 gcc 및 g++, 소스코드 수준의 디버거인 gdb, 유닉스 make의 GNU 버전인 gnumake, 유닉스의 yacc와 호환되는 파서 생성기인 bison, 베이직, Python, Perl, 어셈블러, 포트란, 그리고 파스칼 등), 게임과 그 이상을포함하고 있다.

○ 강력한 SMP(대칭형 다중 처리) 아키텍처 지원

리눅스 2.0버전에서부터 SMP 지원이 강화되었으며, 개발 버전인 2.1버전에서부터 인텔 CPU 아키텍처에 대한 SMP 지원이 지속적으로 이루어지고 있다. 현재 인텔 프로세서에 대해 최대 16 프로세서까지 지원되고 있다.

○ 다양한 사용자 지원 체계

여타의 서버 운영체제가 자체의 제작회사 위주의 서비스를 펼치고 있지만 리눅스는 필요로 하는 사용자 및 고객에 대한 다양한 수준과 형태의 서비스를 전개하고 있다. 리눅스 서버 운영체제와 관련하여 전세계적 범위로 전개되는 서비스의 종류에는 크게 상용 서비스와 비상용 서비스로 구분할 수 있다.

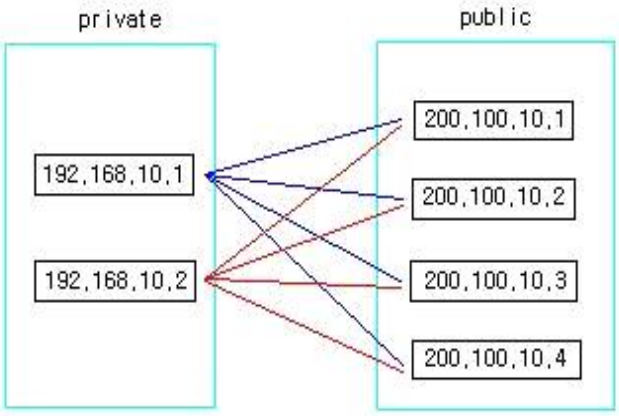
상용 리눅스 버전 구입 고객은 상용 서비스와 공개 서비스를 모두 받을수 있으며, 비상용 리눅스 사용자도 질 높은 공개 서비스를 받을 수 있다. 상용 리눅스 버전 구입 고객은 해당 제품을 구입한 벤더, 가령 리눅스코리아에서 제품을 구입한 고객은 리눅스코리아를 통해, 레드햇 소프트웨어 사에서 제품을 구입한 고객은 레드햇 소프트웨어 사에서 제품에 대해 제공하는 서비스를 받을 수 있다. 그리고 OS 자체 이외에 대 고객 서비스 기능 지원, 서버 유지보수, 네트웍 유지보수 및 OS와 관련된 심화된 유료 서비스를 별도로 신청할 수 있다. 미국 인포월드(InfoWorld)는 1997년 한해동안 최고의 기술지원을 전개한 회사 및 단체를 선정했는데 , 여기에 리눅스 공동체가 1위에 올랐다. 이것은 어지간한 상용 회사에서 제공하는 수백 수십만원 하는 유료 서비스보다 리눅스 공동체에서 제공하는 무료 상호 서비스가 훨씬 뛰어나다는 것을 의미한다.

리눅스 공동체에서 제공하는 상호 기술지원 서비스로는 다양한 뉴스그룹을 통한 신속한 지원, 전자우편을 통한 메일링 리스트 서비스, 다양한 형태와 분야를 가지는 공식, 비공식 웹을 통한 서비스 등이 있으며, 한국에는 추가적으로 국내 대형 BBS에 구성되어 있는 리눅스 동호회 모임과 한국 리눅스 사용자 모임 및 각지역별 리눅스 사용자 모임이 구성되어 있어 활발하고 자발적이고 질높은 사용자지원 활동을 벌이고 있다.

2.2 NAT 기술

2.2.1 dynamic NAT

내부의 사설IP가 라우터 또는 NAT 소프트웨어에 의해서 미리 정해진 공인IP로 랜덤하게 매핑.

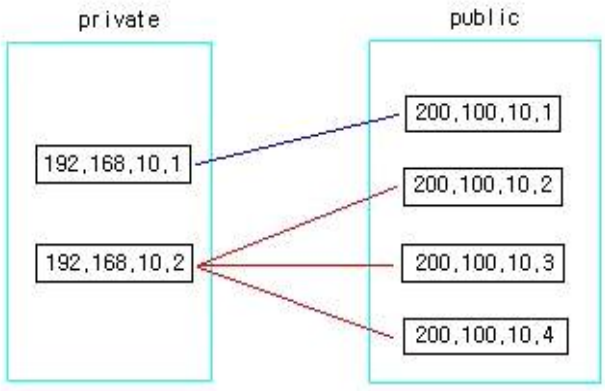


<그림 2-2-1>

위에서 192.168.10.1은 200.100.10.1 ~ 4의 어떤 공인IP와도 매핑이 가능합니다. 192.168.10.2도 192.168.10.1과 같게 매핑이 가능합니다. 그러나 dynamic NAT의 경우 정해진 공인IP가 이미 다 사용중일 경우엔 나머지 사설IP는 공인IP를 사용할 수가 없습니다. 예를 들어서 192.168.10.1~4가 200.100.10.1~4를 각각 사용중 일때 192.168.10.5는 자리가 빌 때까지 대기를 해야합니다. 이러한 점을 보완하는 방법으로 NAT overload(또는 PAT)가 있다.

2.2.2 static NAT

특정 사설IP가 특정 공인IP만 사용하도록 관리자가 미리 정해놓는 방식



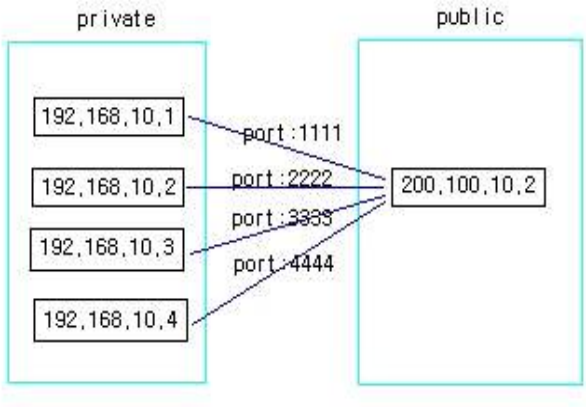
<그림 2-2-2>

위에서 192.168.10.1은 200.100.10.1로만 무조건 매핑이 가능합니다.

그리고 192.168.10.2는 나머지 200.100.10.2 ~ 4 중에서 하나를 사용하게 된다.

2.2.3 PAT(Port Address Translation)

포트 변환을 통해서 NAT를 실행합니다. static NAT나 dynamic NAT의 경우 사용할 수 있는 공인IP보다 사설IP의 수가 많다면 모자라는 만큼 외부로 나갈 수 없는 사설IP가 많아집니다. 하지만 PAT는 포트 변환을 하기때문에 공인IP가 하나만 있어도 많은 수의 사설IP가 외부로 나갈 수 있습니다.



<그림 2-2-3>

위 경우 사용할 수 있는 공인IP는 한 개이지만 내부의 사설IP들은 각각 포트를 달리함으로써 개별적으로 외부로 나가는게 가능하게 된다.

2.3 IPtables 운영 기술

2.3.1 사용 명령어

○ 체인 제어 명령

사용법 iptables [-NXPLFZ] [chain명]

IPTABLES 옵션

설 명

-N : 새로운 chain 생성 (--new-chain)

-P : 체인의 정책 변경 (--policy)

-L : 체인에 적용된 정책 리스트 출력 (--list)

-F : 체인에 적용된 정책 지우기 (--flush)

-X : 비어있는 체인 제거 (--delete-chain)

-E : 체인의 이름을 변경 (--rename-chain)

-Z : 체인내의 패킷과 바이트의 카운트를 0으로 리셋 (--zero)

-A : 체인에 새로운 필터링 규칙을 추가하기 (--append)

-I : 체인의 특정 지점에 필터링 규칙 삽입 (--Insert)

-R : 체인의 특정 지점의 필터링 규칙을 교환 (--replace)

-D : 체인의 특정 지점의 필터링 규칙을 제거 (--delete)

○ 파라미터

IPTABLES 파라미터

설 명

-p : (--protocol) 프로토콜을 지정
tcp, udp, icmp or all(모든 프로토콜)을 지정
!(not)를 통해서 역의 결과를 만들 수도 있다.
-p tcp (tcp 프로토콜을 사용하는)
-p ! tcp (tcp 프로토콜을 사용하지 않는)

-s : (--source) 패킷을 발생시키는 발생지
!(not)을 통해서 역의 결과 도출 가능
-s 211.239.151.21 (211.239.151.21을 패킷 발생지로 하는)
-s ! 211.239.151.21 (패킷 발생지를 211.239.151.21으로 하는것을 제외하고)

-sport : (--source-port) 패킷을 발생시킨 발생지에서 접속해온 포트 ! 사용 가능
--sport 21 (21번 포트에서 발생한 패킷)

-d : (--destination) 패킷이 도착하는 지점. 사용법은 souece와 동일
--dport : (--destination-port) 패킷의 도착지 포트 , 사용법은 sport와 동일
포트가 연속적으로 여러 개일 경우 '-' 로 표현할 수 있다
(20-22 → 20, 21, 22를 의미)

-j : (--jump) 필터링 규칙에 의해 적용되는 패킷을 target으로 보낸다.

-i : (--in-interface) 패킷이 들어올 때 경로가 되는 인터페이스를 지정
!(not) 사용가능
interface = lo (localhost), eth0 (랜카드 한개일 경우),
-i eth0 (설정된 랜카드를 통해서 들어오는)

-o : (--out-interface) 패킷이 나갈 때 경로가 되는 인터페이스를 지정
! 사용가능
-o eth0 (랜카드로 나가는 패킷을 지정)
-o ! eth0 (랜카드로 나가는 패킷을 제외하고)

-f : (--fragment) 패킷이 한번에 전달되지 못할만큼 크기가 클 때 패킷을 여러 개로 나누어서 (분절) 여러개의 패킷으로 전달할 때 사용 . 이들 패킷은 목적지에 도착해서 재구성되어 전체 패킷이 된다.
!(not) 가능

-c : (--set-counters) INSERT(-I), APPEND(-A), REPLACE(-R) 명령을 사용하는 동안 규칙의 패킷과 바이트의 카운터를 초기화 시킨다 .

<사용법>

iptables -[ADC] chain rule-specification [options]
iptables -[RI] chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LFZ] [chain] [options]

```
iptables -[NX] chain
iptables -P chain target [options]
iptables -E old-chain-name new-chain-name

설명
# chain - INPUT, OUTPUT, FORWARD or 직접 생성한 chains
# rulenum - 규칙이 생성된 순서 (순서대로 1,2... )
# rule-specification - 파라미터를 사용하여 만들어진 규칙 (필터링 규칙)
# target - ACCEPT, DROP, QUEUE, RETURN
```

2.4 웹서버 운영 기술

2.4.1 웹서버란?

웹 브라우저를 사용하여 World Wide Web을 사용하는 클라이언트에게 미리 저장된 하이퍼텍스트를 제공하는 서버를 말한다. 주로 사용되는 서버에는 마이크로 소프트 윈도우 기반의 iis 서버와 유닉스 기반의 아파치 서버가 있다. 우리는 이번 졸업작품에서 아파치 서버를 사용하였다.

2.4.2 아파치서버란?

○ 정의

NCSA의 httpd 1.3 버전을 기반으로 탄생하였고, 이름은 기존 NCSA httpd 1.3에 패치들을 제공했던 사람들이 모여 구성한 그룹인 " A PAtCH server"에서 유래

○ 특징

- 새로 컴파일 하지 않아도 기능을 추가하거나 삭제 가능
- 지속적인 패치 파일을 제공 -> 안전성 확보
- 리눅스(Linux), 솔라리스(Solaris) 등의 다양한 플랫폼을 지원
- Open Source : 소스가 공개되어 있다.
- 무료로 제공

2.5 FTP서버 운영 기술

2.5.1 리눅스 FTP(File Transfer Protocol)

FTP는 인터넷의 대표적인 응용 서비스로서 인터넷 사용자가 Network상의 다른 Host에 있는 파일을 전송하거나 접근할 때 사용하는 Protocol이다. 1971년경부터 인터넷에서 사용되었으며 현재까지 큰 변화가 없는 반면에 Client와 Server는 꾸준히 향상되어 왔다.Telnet과는 달리 FTP는 단순히 파일의 위치변경과 송수신에 관련된 기능만을 수행할 수 있다. FTP의 기능 중에는 또한 Directory 변경, 파일목록보기 및 파일수신과 같은 기능들이 있다.

2.5.2 FTP동작원리

○ FTP 동작 순서

FTP 클라이언트는 자신의 소스 포트로 1024 보다 큰 임의의 번호를 사용하여 FTP 서버에 접속을 시도한다.(FTP 서버의 포트는 대부분 FTP 표준에서 정의된 21번 포트를 사용한다.)

○ 로그인

연결이 성공하고 나면 클라이언트는 로그인을 시도하며 로그인에 성공하고 나면 FTP서버의 디렉토리를 탐색할 명령어를 보낼 수 있다.

○ 포트확보

클라이언트가 파일 전송 요청을 하면 서버는 높은 번호를 갖는 다른 임의의 포트를 새로 확보한 후, 그 곳에서 연결을 기다린다.

○ 포트결정

클라이언트는 이 새 포트에 대해서는 모르고 있을 것이므로, 서버는 이미 설정되어 있는 연결을 통해 FTP 클라이언트에게 이 새 포트 번호를 보낸다.

○ 포트연결

새 포트 번호를 받은 FTP 클라이언트는 임의의 새로운 포트를(기존에 클라이언트가 사용하던 포트번호보다 1이 큰 포트)확보한 후 서버가 보내준 포트로 새 연결을 만들고, 원래의 연결은 부가적인 메시지를 서로 주고받기 위해 열어둔다.

2.5.3 방화벽과 FTP

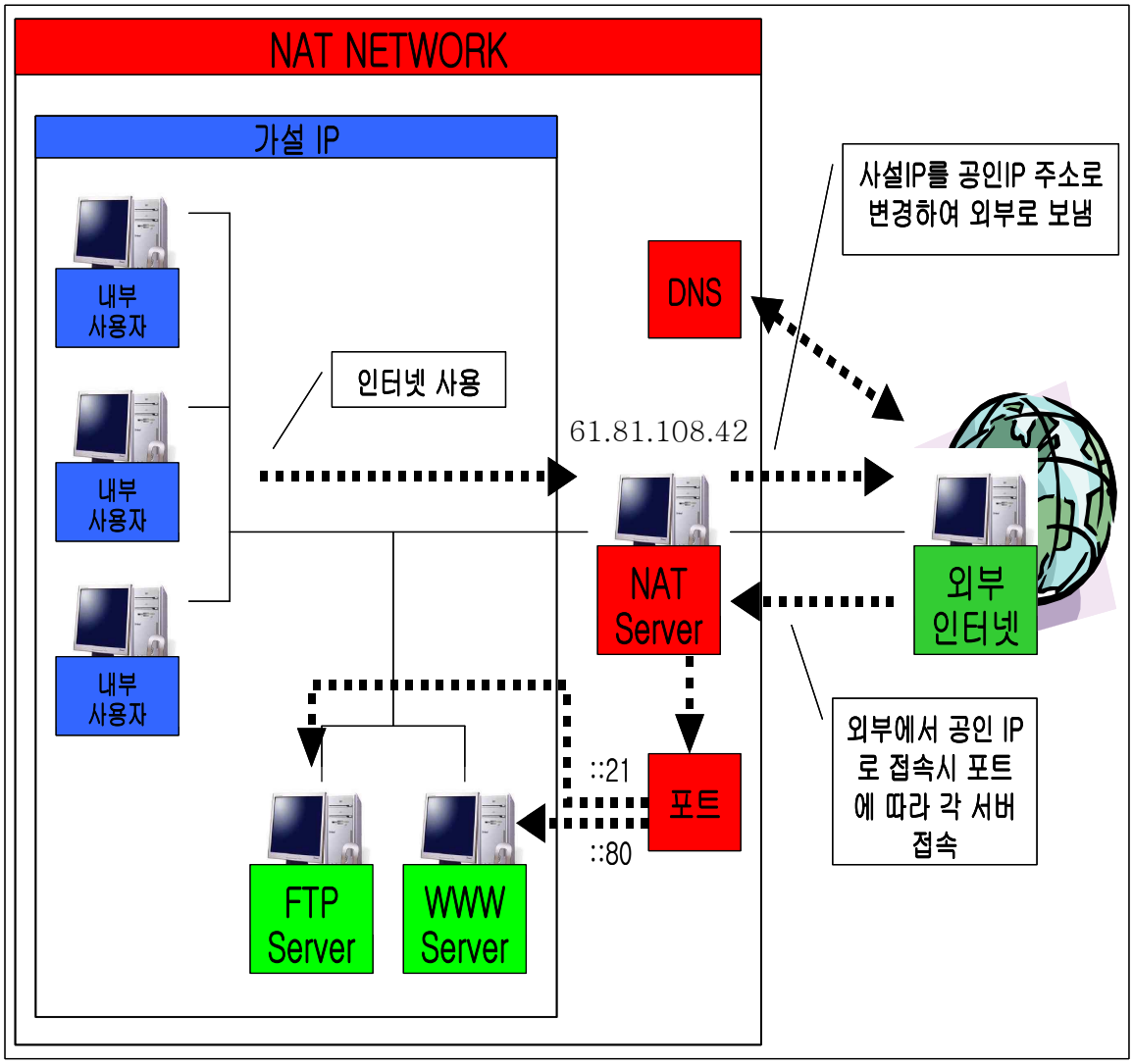
방화벽은 인터넷의 높은 포트로부터 내부 네트워크의 높은 포트로 들어오는 임의의 연결을 좋지 않은 것으로 간주한다. 그러므로 FTP 패킷들이 방화벽을 통과하려면 방화벽 시스템에 FTP를 위한 응용 수준의 프락시(proxy)를 구현해두어야 하며 FTP프록시는 FTP요청을 감시하다가 원격지로부터 데이터를 받을 필요가 생기면 높은 포트를 사용할 수 있도록 허용해 주는 역할을 한다.FTP가 일반 패킷 필터링 방화벽을 통과할 때 문제점은 대부분 자신을 통과하는 데이터의 의미에 대해서는 이해하지 못하므로 임의의 높은 포트로 전송되는 데이터는 안 좋다는 사실만 알고 있다. 따라서 이런 방화벽은 FTP를 끊어 버리게 된다.

2.5.4 passive transfer 방식

서버가 아닌 클라이언트가 파일 전송의 주체가 되는 방식으로서 클라이언트가 파일 전송을 위한 연결 요청을 초기화하기 때문에 내부로 들어오는 연결의 허용에 대한 복잡한 규을 만들 필요가 없다.

3. 구축 및 운영

3.1 설계도



<그림 3-1>

NAT망을 구성하기 위해 먼저 NAT서버에 랜카드 2개를 설치한다. 첫 번째 랜카드는 공인IP선을 이용하여 외부와 연결을 하고, 두 번째 랜카드는 스위치허브에 연결하여 내부 망을 구성한다. 그리고 Web서버, FTP서버, 내부사용자도 마찬가지로 스위치허브에 연결하여 내부 망을 구성한다.

NAT서버 내부 망에 FTP서버, Web서버를 두어서 내·외부사용자가 사용할 수 있게 구현하고, 내부사용자는 NAT서버를 이용하여 인터넷을 사용하는 동시에 내부서버도 사용한다. NAT서버를 통해 주소변환해서 나가기 때문에 보안상 손쉽게 관리 할 수 있으며, NAT서버는 DNS 서비스 및 방화벽을 담당하게 된다.

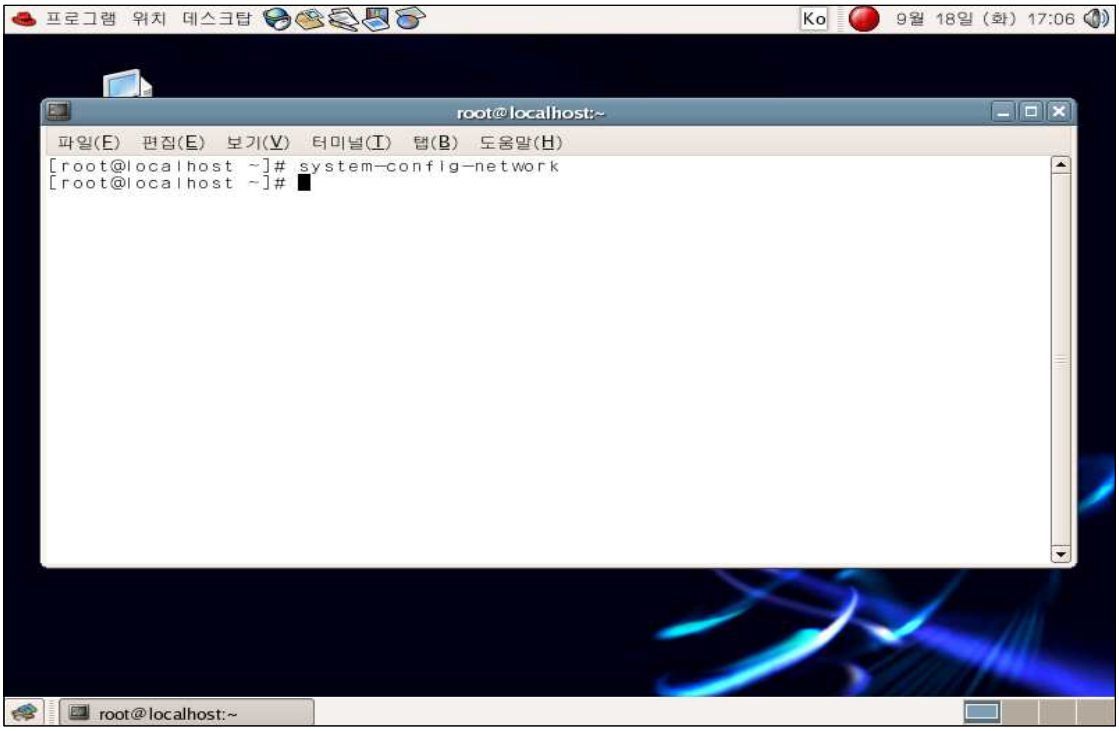
3.2 구축환경 설정

3.2.1 NAT서버 PC설정

네트워크 구성 - NIC 2개

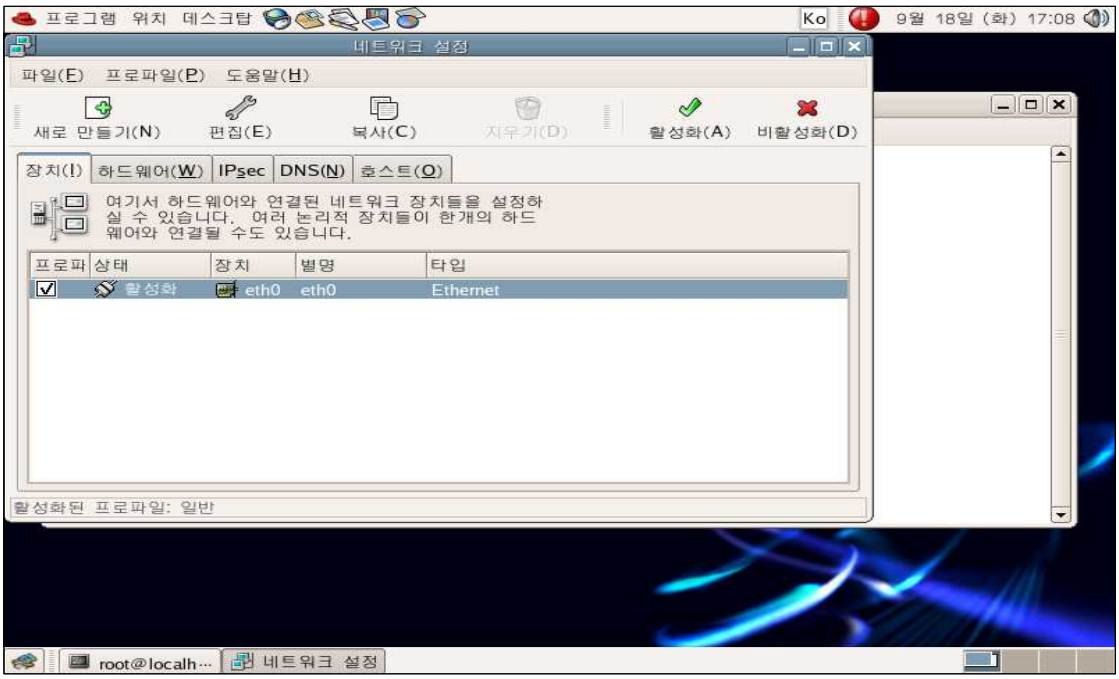
새로운 NIC 마운트

system-config-network



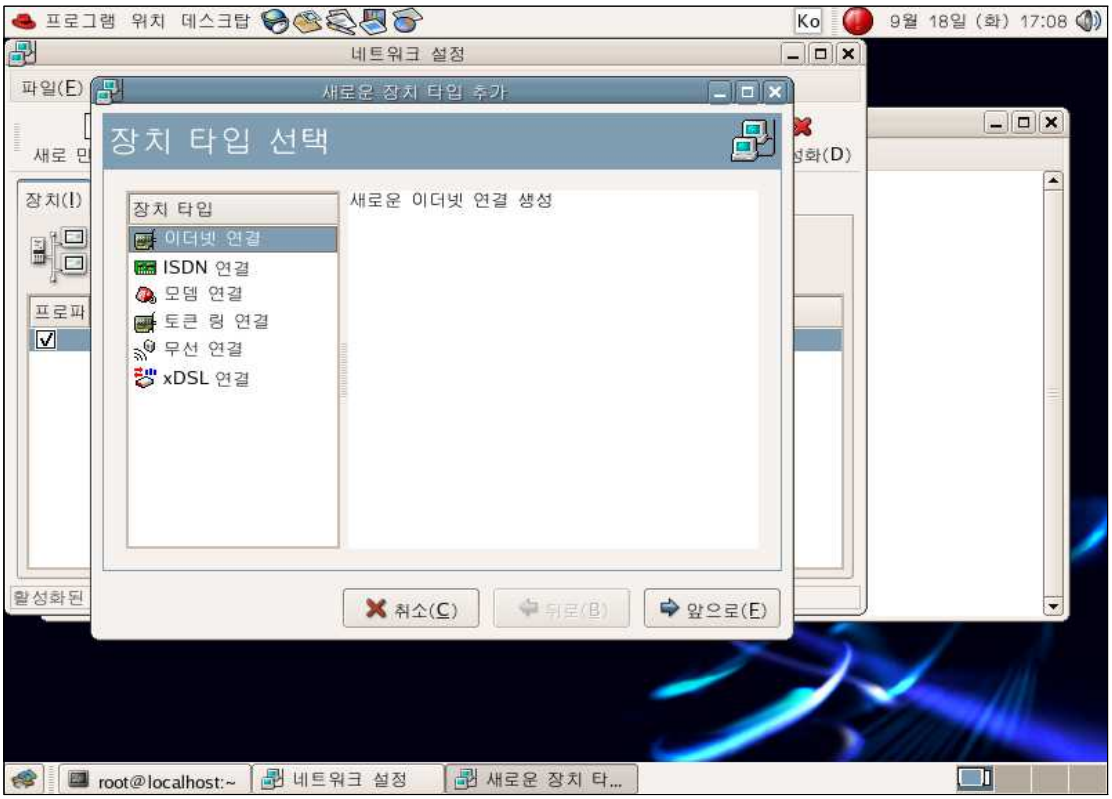
<그림 3-2-1>

현재의 네트워크 설정을 보여준다. 새로만들기를 클릭한다.



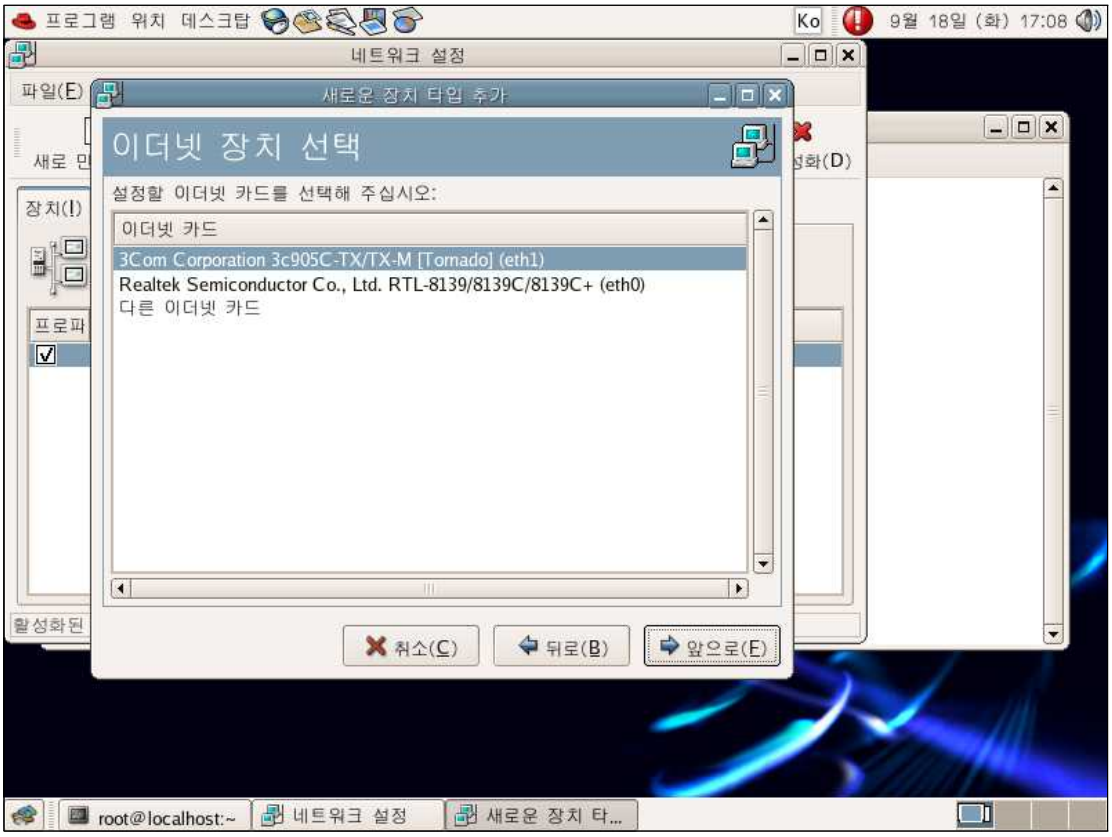
<그림 3-2-2>

장치타입에서 Ethernet 연결을 선택



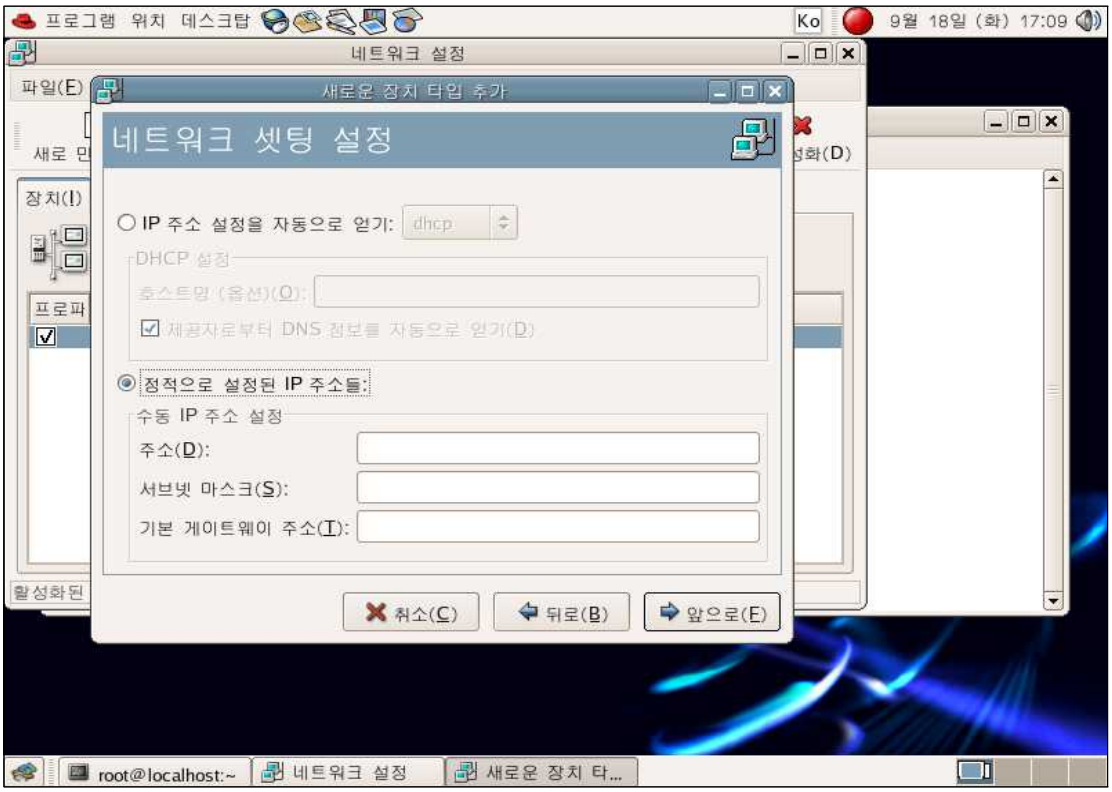
<그림 3-2-3>

장치에 맞는 모델명 선택



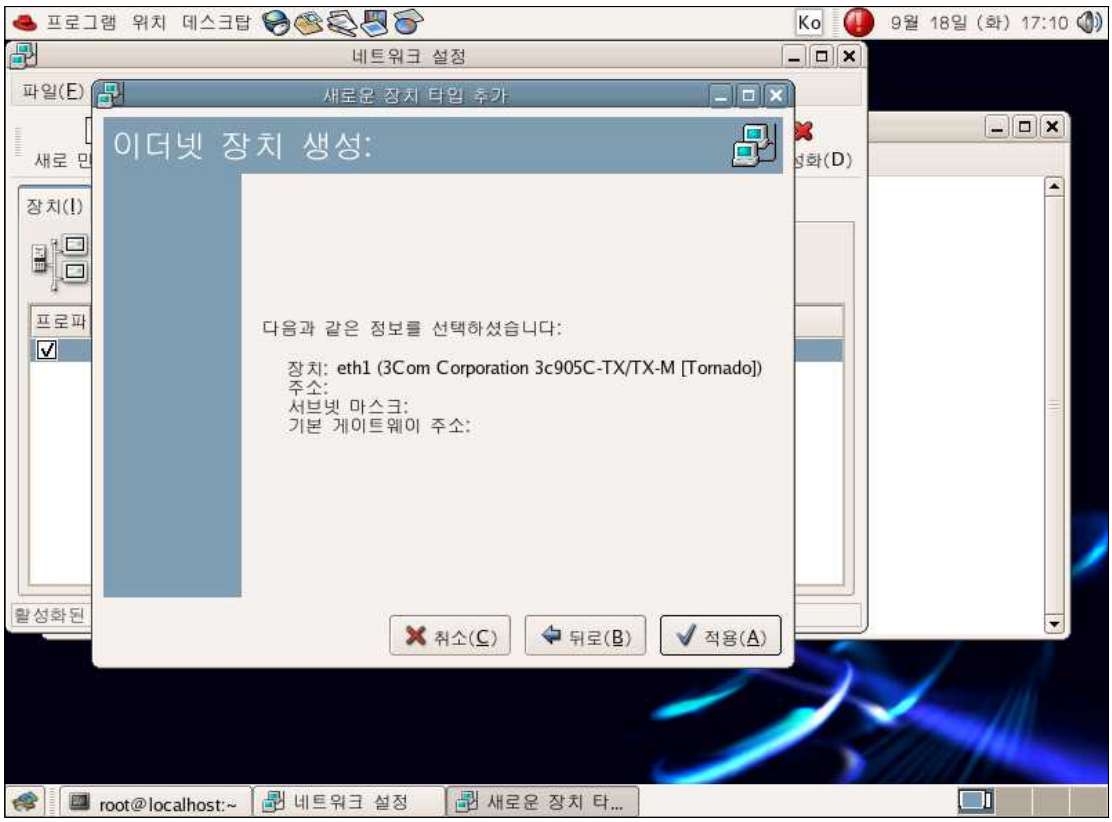
<그림 3-2-4>

정적 IP설정을 위한 화면



<그림 3-2-5>

새로운 Ethernet 장비 추가 완료



<그림 3-2-6>

NIC 장치 환경설정

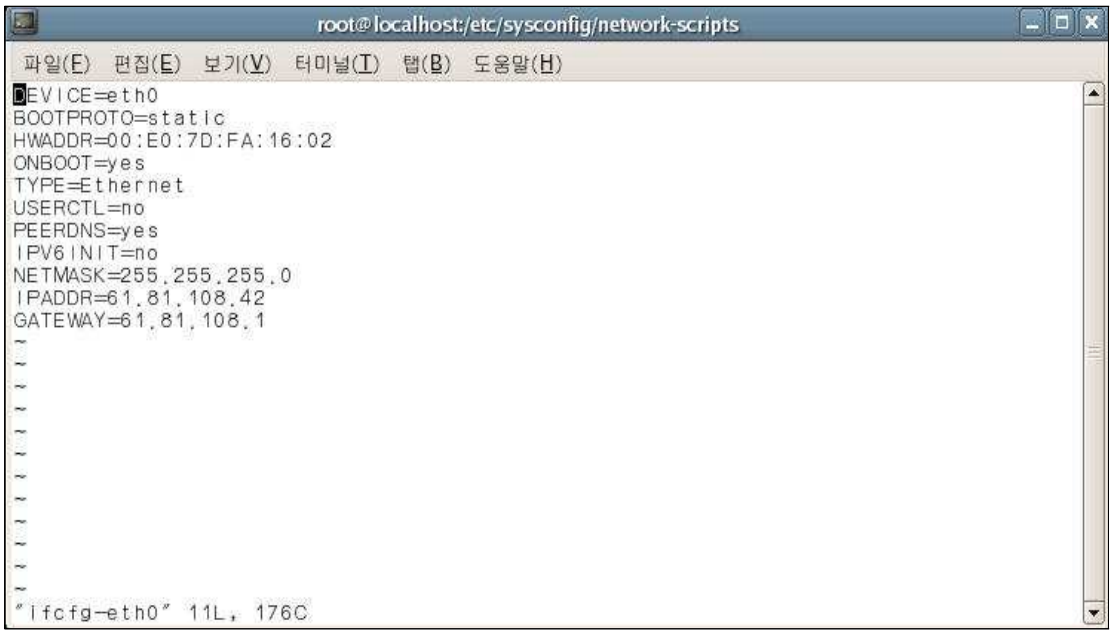
NIC장치 이름 - eth0,eth1

설정파일 위치 - /etc/sysconfig/network-scripts

설정파일 이름 - ifcfg-eth0, ifcfg-eth1

eth0설정 - 61.81.108.42

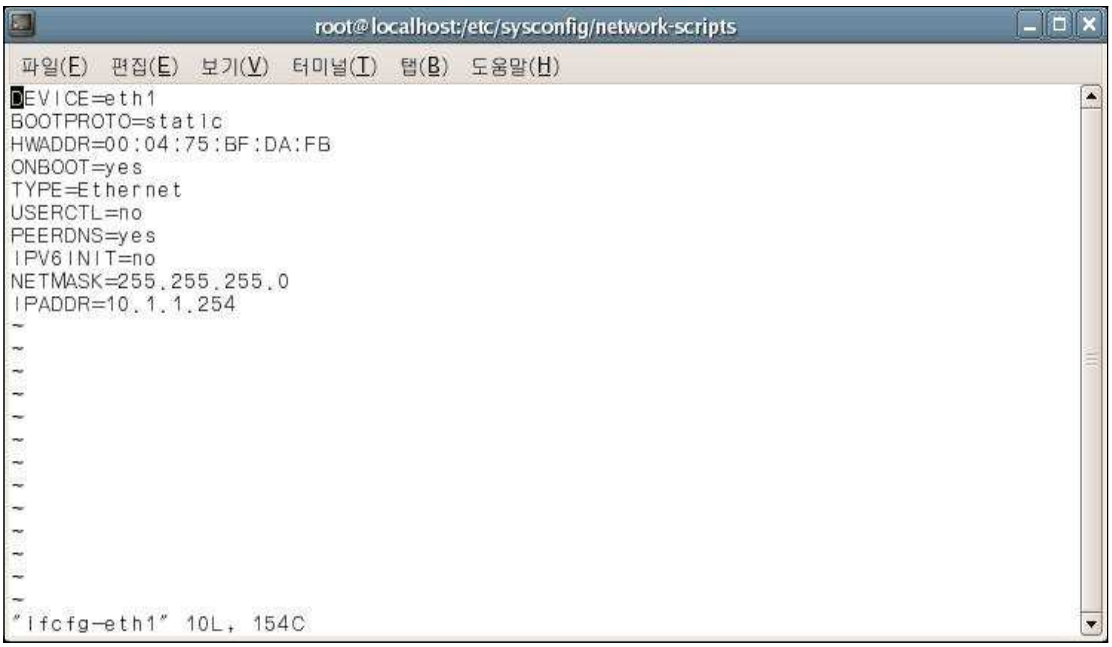
vi ifcfg-eth0



<그림 3-2-7>

eth1설정 - 10.1.1.254

vi ifcfg-eth1



<그림 3-2-8>

3.2.2 웹서버 PC 설정

네트워크 구성 - 랜카드 1개

eth0 - 10.1.1.10:80

설정파일 위치 - /etc/sysconfig/network-scripts

설정파일 이름 - ifcfg-eth0

vi ifcfg-eth0

<그림 3-2-9>

3.2.3 ftp서버 PC 설정

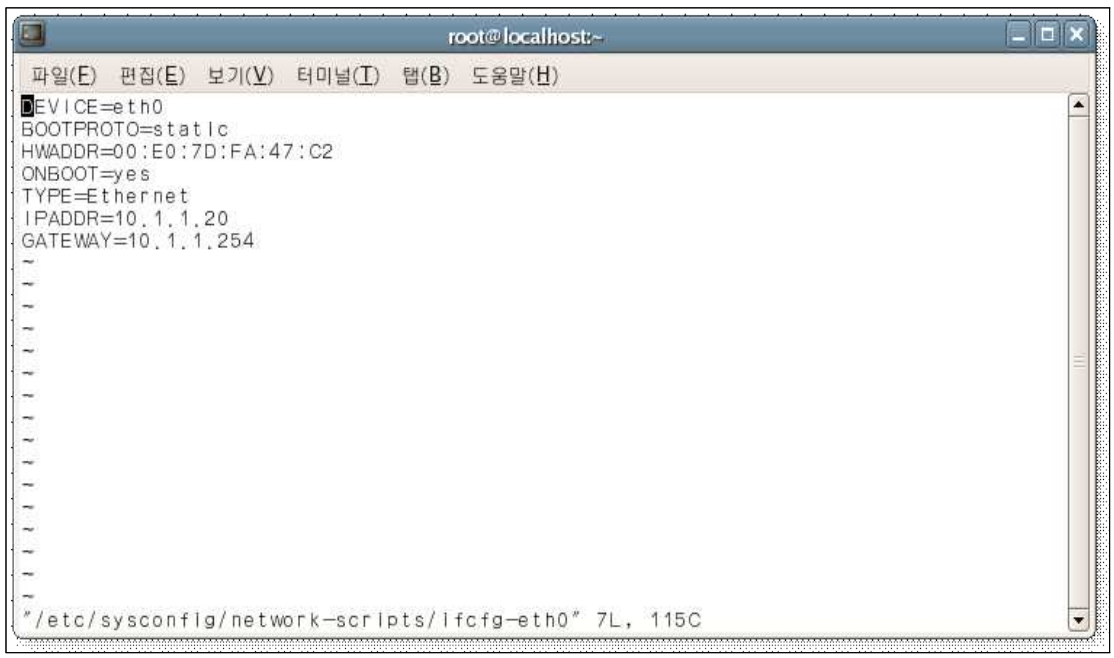
네트워크 구성 - 랜카드 1개

eth0 - 10.1.1.20:21

설정파일 위치 - /etc/sysconfig/network-scripts

설정파일 이름 - ifcfg-eth0

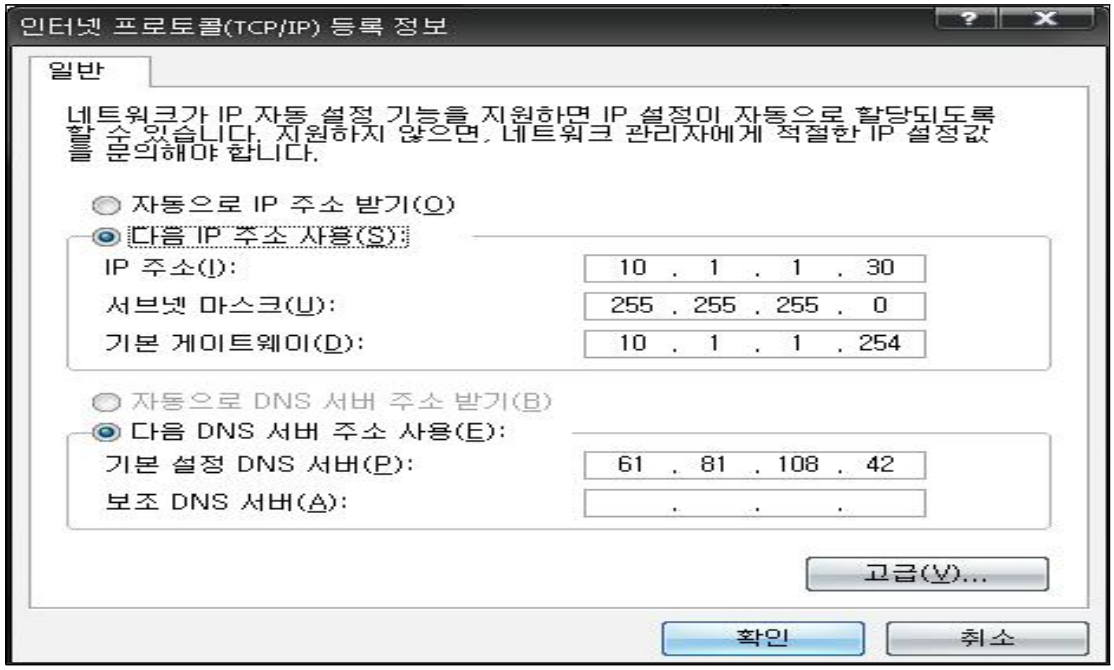
vi ifcfg-eth0



<그림 3-2-10>

3.2.4 내부사용자 PC 설정

IP주소 - 10.1.1.30



<그림 3-2-11>

3.3 NAT를 이용한 IP주소 변환

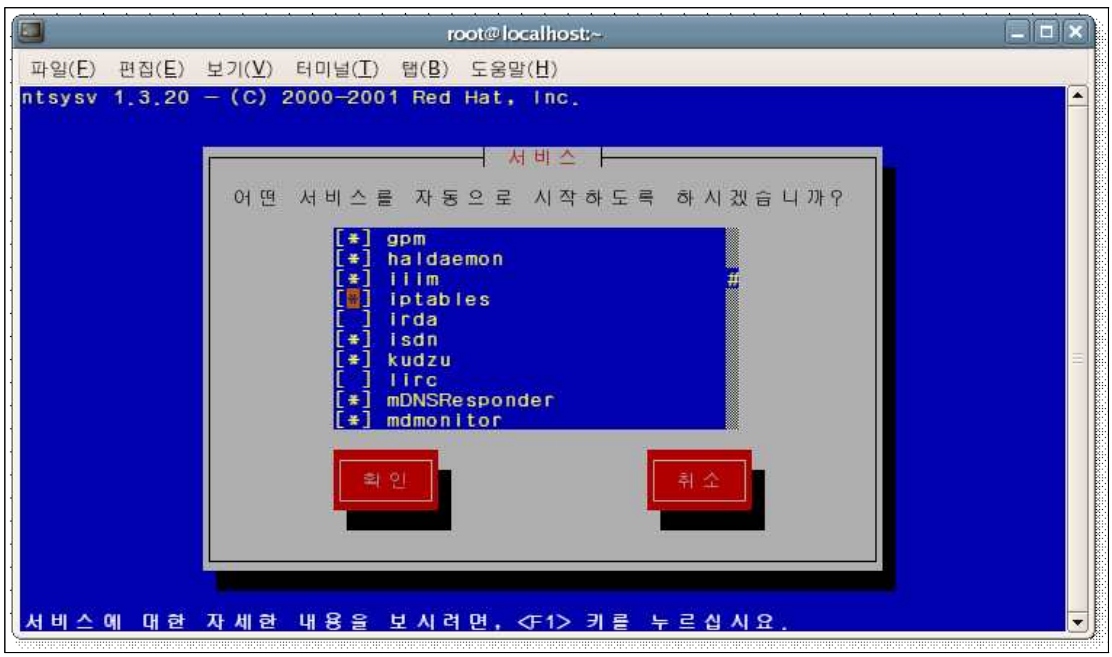
3.3.1 iptables 초기화

NAT서버 컴 - iptables 초기화

iptables -F

iptables -t nat -F

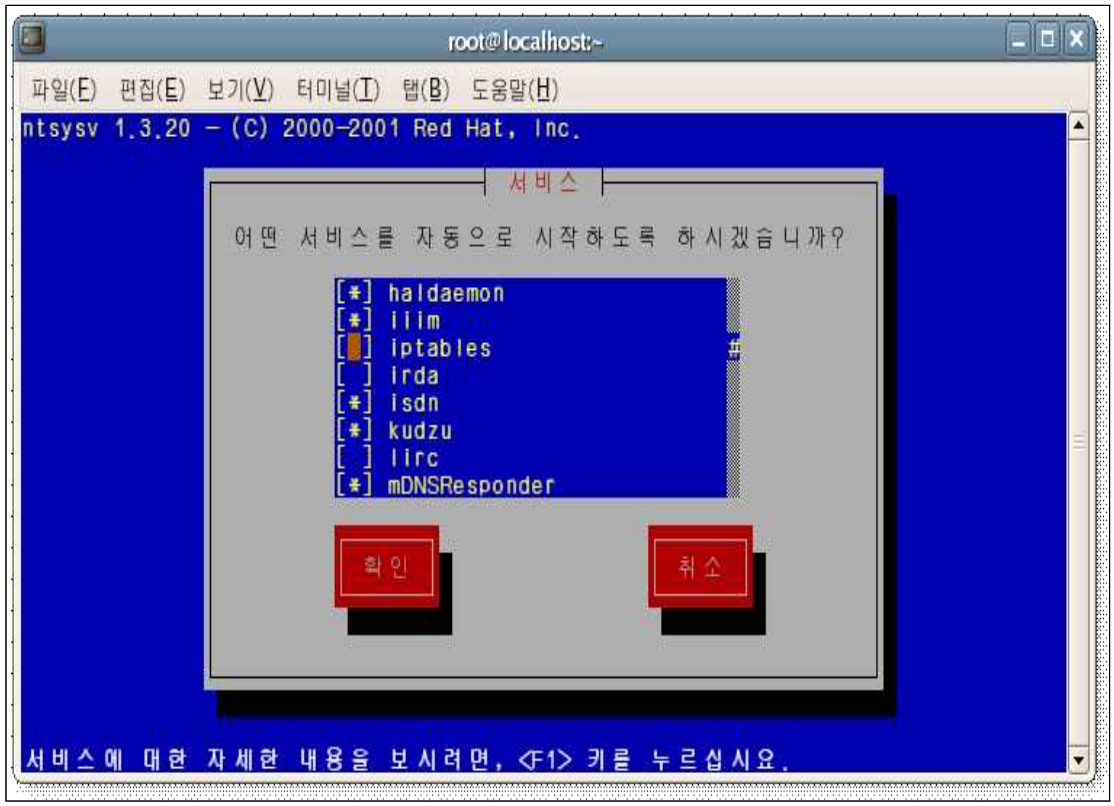
iptables 활성화



<그림 3-3-1>

ftp서버 컴, 웹서버 컴 - iptables 비 활성화

ntsysv



<그림 3-3-2>

3.3.2 iptalbles FORWARD 설정

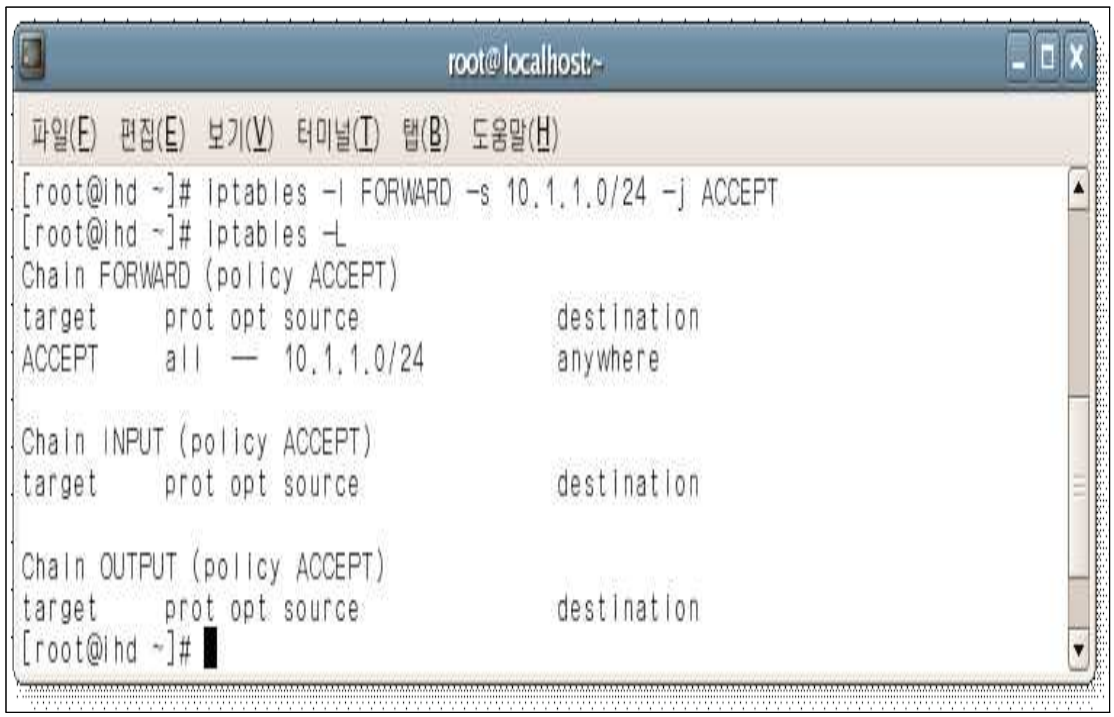
iptables -L

iptables에 현재 설정된 룰을 확인한다.

<그림 3-3-3>

iptables -I FORWARD -s 10.1.1.0/24 -j ACCEPT

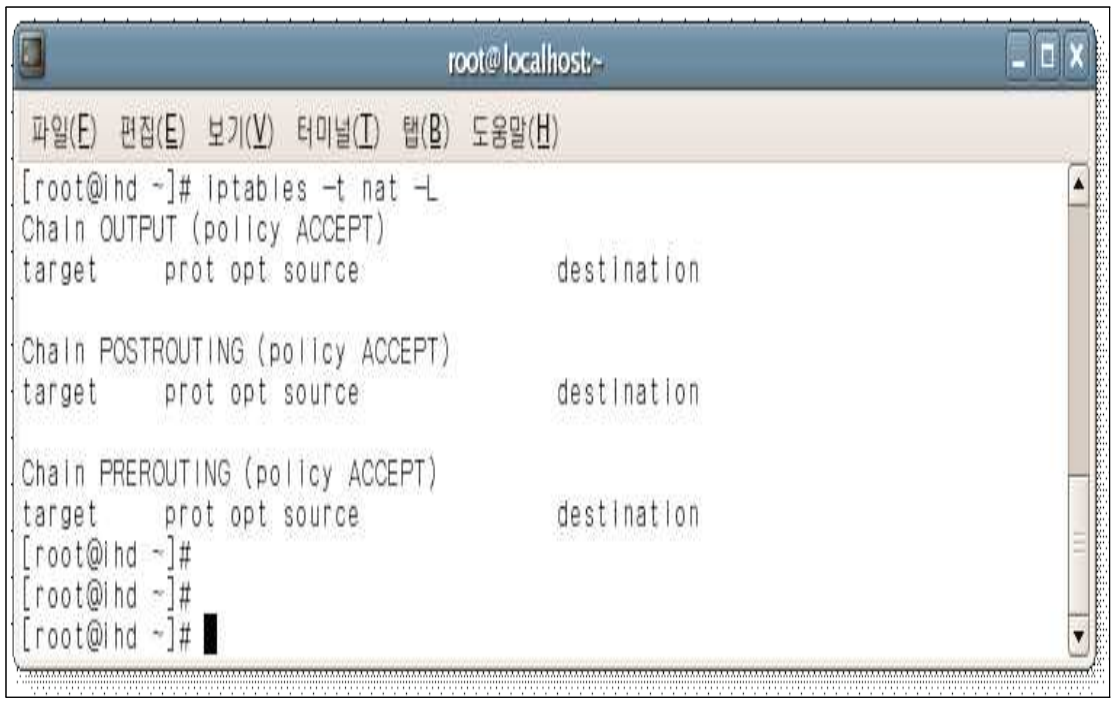
iptables을 이용하여 10.1.1.0/24의 가설 ip의 통로를 허용해준다.



<그림 3-3-4>

iptables -t nat -L

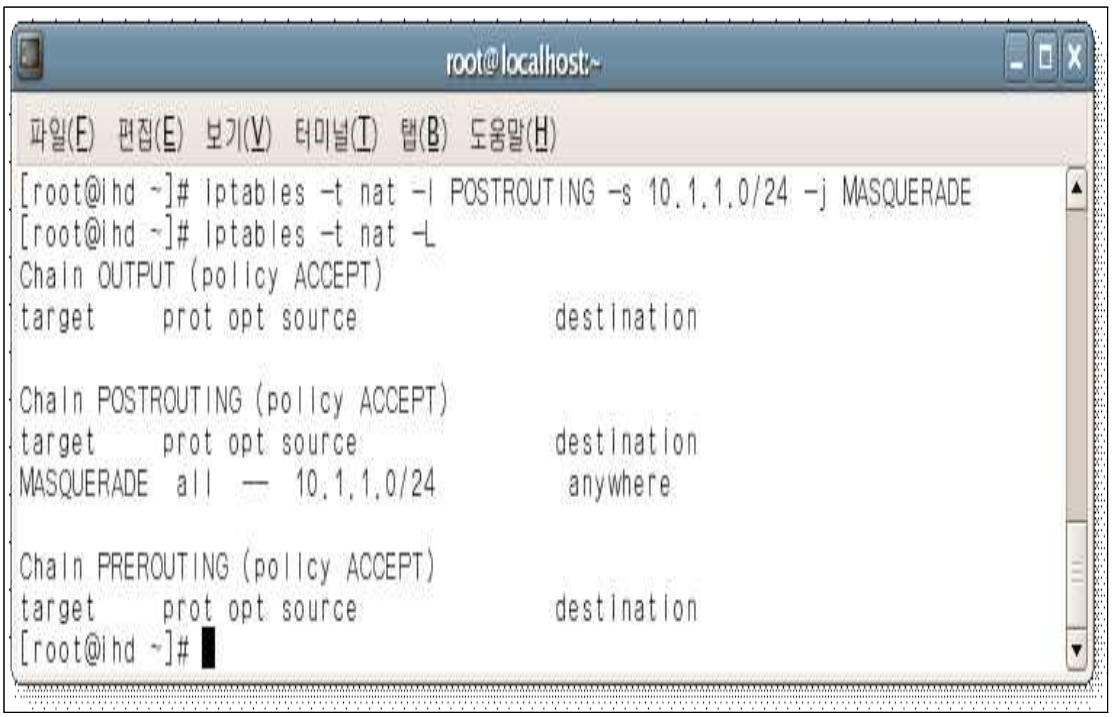
iptables NAT룰 부분의 현재 설정을 확인한다.



<그림 3-3-5>

iptables -t nat -I POSTROUTING -s 10.1.1.0/24 -j MASQUERADE

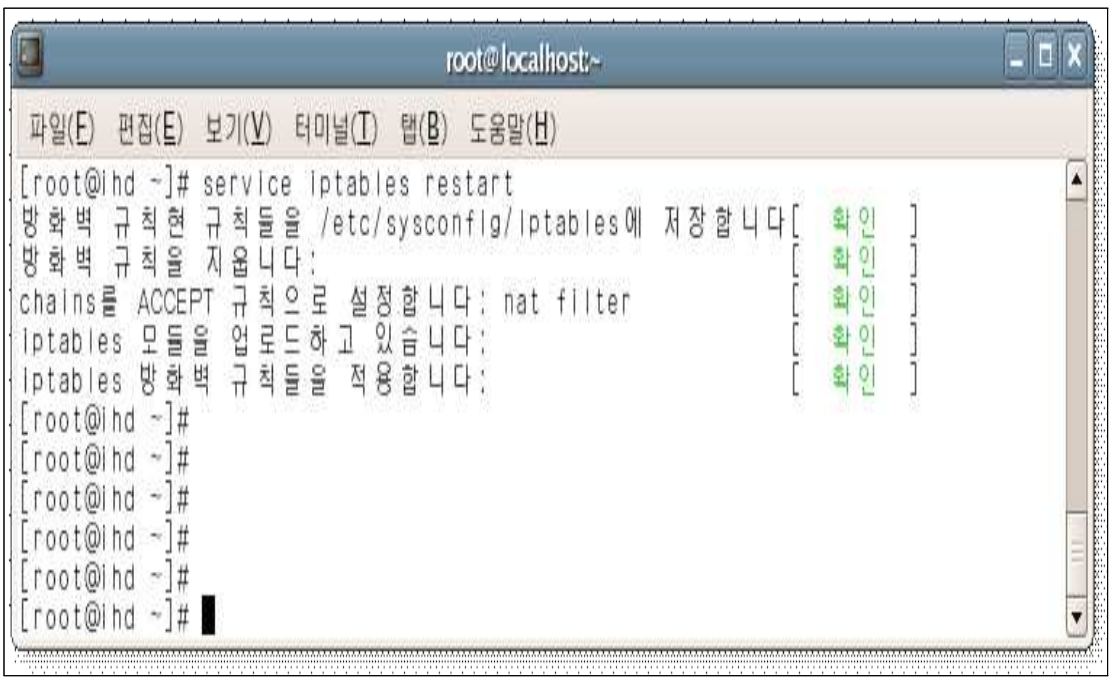
iptables을 이용하여 외부로 나가는 10.1.1.0/24의 가설 ip를 현재 NAT서버의 ip를 이용하여 주소변환을 한후 외부로 패킷을 내보낸다.



<그림 3-3-6>

service iptables restart

iptables를 재시작 한다.



<그림 3-3-7>

3.4 ftp서버 구축

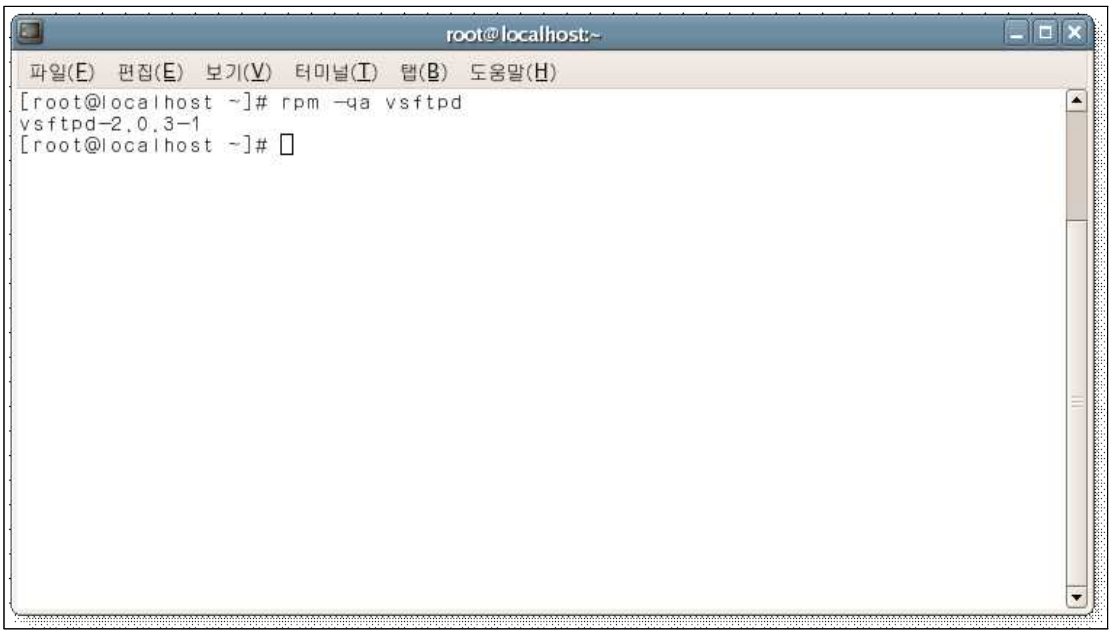
3.4.1 설치

ftp 서버는 vsftpd 를 사용

yum -y install vsftpd

yum을 이용하여 vsftpd의 최신버전(vsftpd-2.0.3-1)을 설치

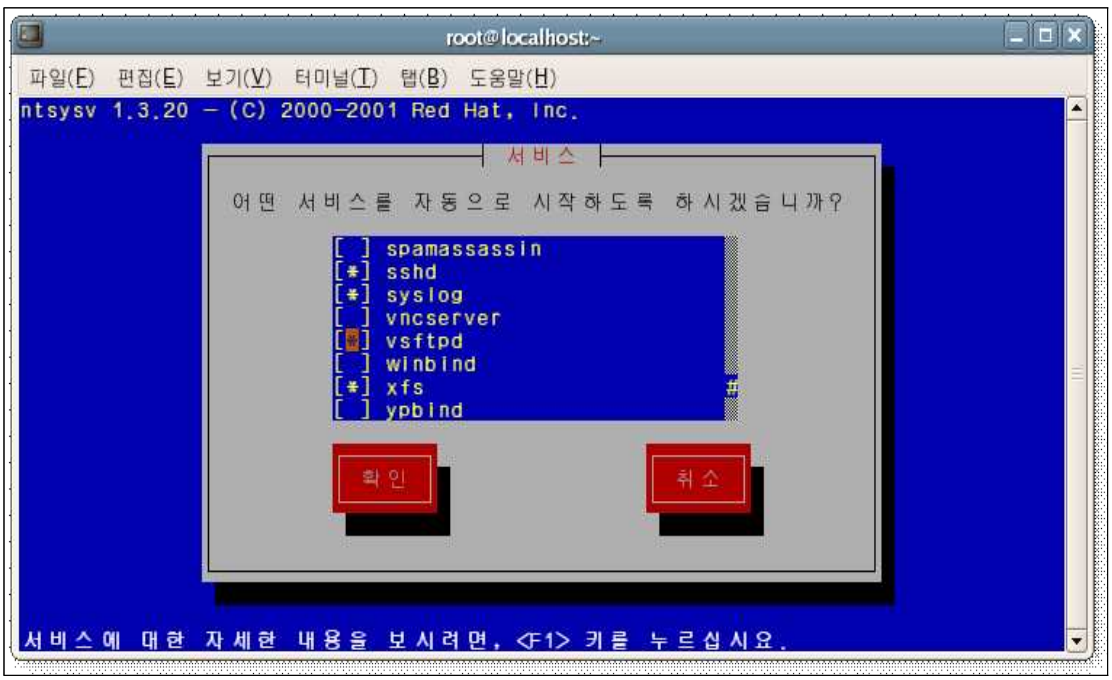
방화벽 설정은 NAT서버에서 구축하였으므로 사용하지 않는다.



<그림 3-4-1>

ntsysv

시스템 부팅시 ftp서비스를 자동시작할수 있게 등록한다.



<그림 3-4-2>

3.4.2 ftp 환경설정

vi /etc/vsftpd/vsftpd.conf

vsftpd의 환경설정을 한다.

<그림 3-4-3>

27행 - anon_upload_enable=YES 의 주석 부분을 제거한다.

익명사용자의 업로드 권한을 허용한다.

31행 - anon_mkdir_write_enable=YES 의 주석 부분을 제거한다.

익명사용자의 폴더만들기 권한을 허용한다.

85행 - ftpd_banner=Welcome to bigbang FTP service.

ftp 서버 접속시 환영메세지를 설정 한다.

3.5 웹 서버 구축

3.5.1 httpd mysql, php설치설치

웹 서버는 httpd 를 사용

yum -y install httpd

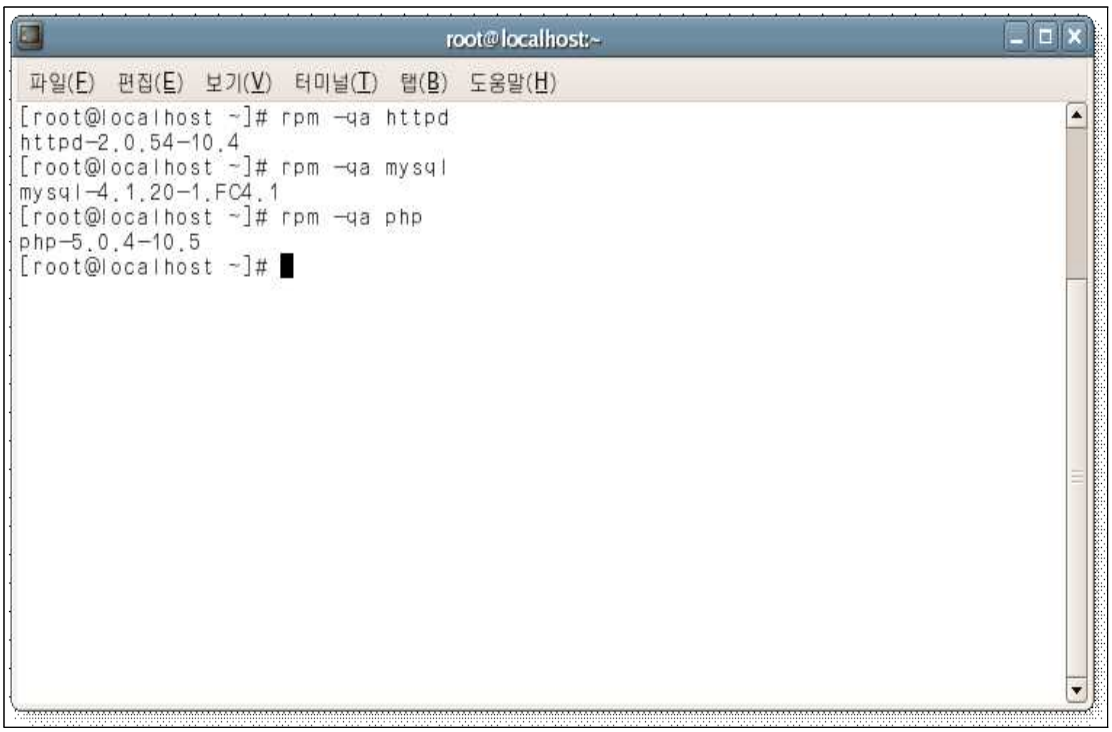
mysql, php 설치

yum -y install mysql

yum -y install php

yum을 이용하여 httpd(httpd-2.0.54-10.4), mysql(mysql-4.1.20-1.FC4.1)

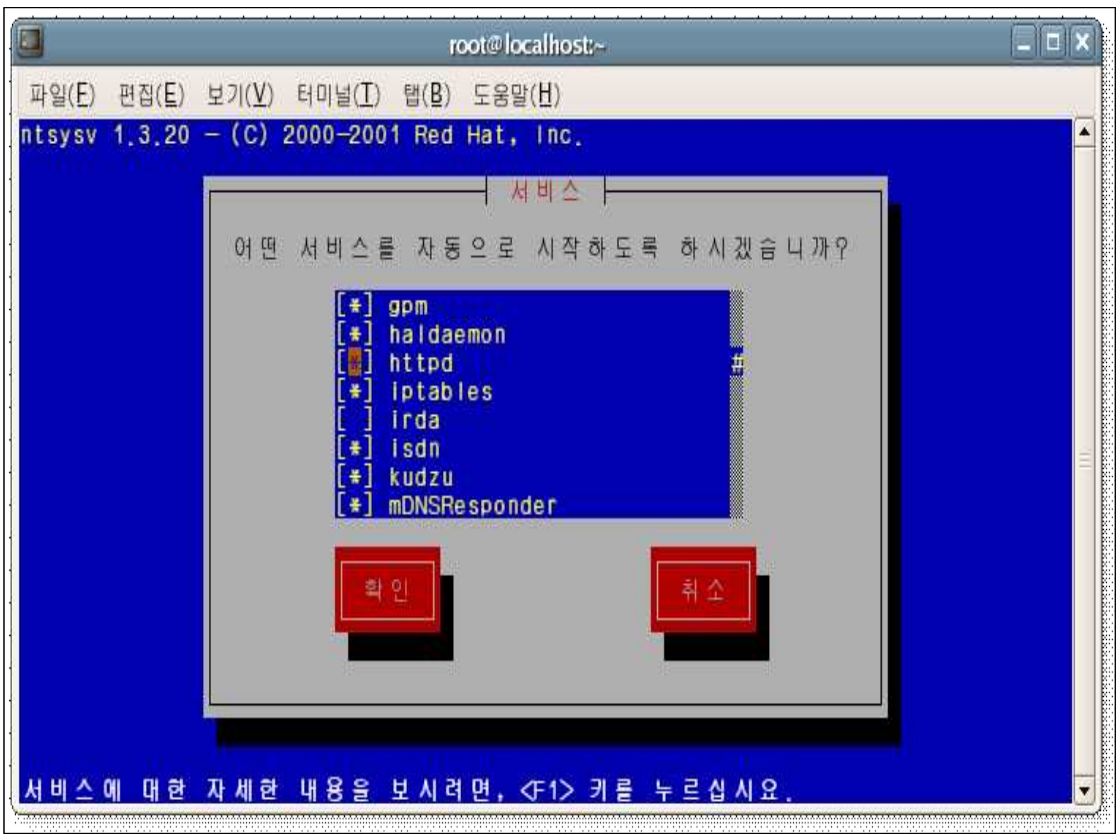
php(php-5.0.4.-10.5)의 최신버전을 설치



<그림 3-5-1>

ntsysv

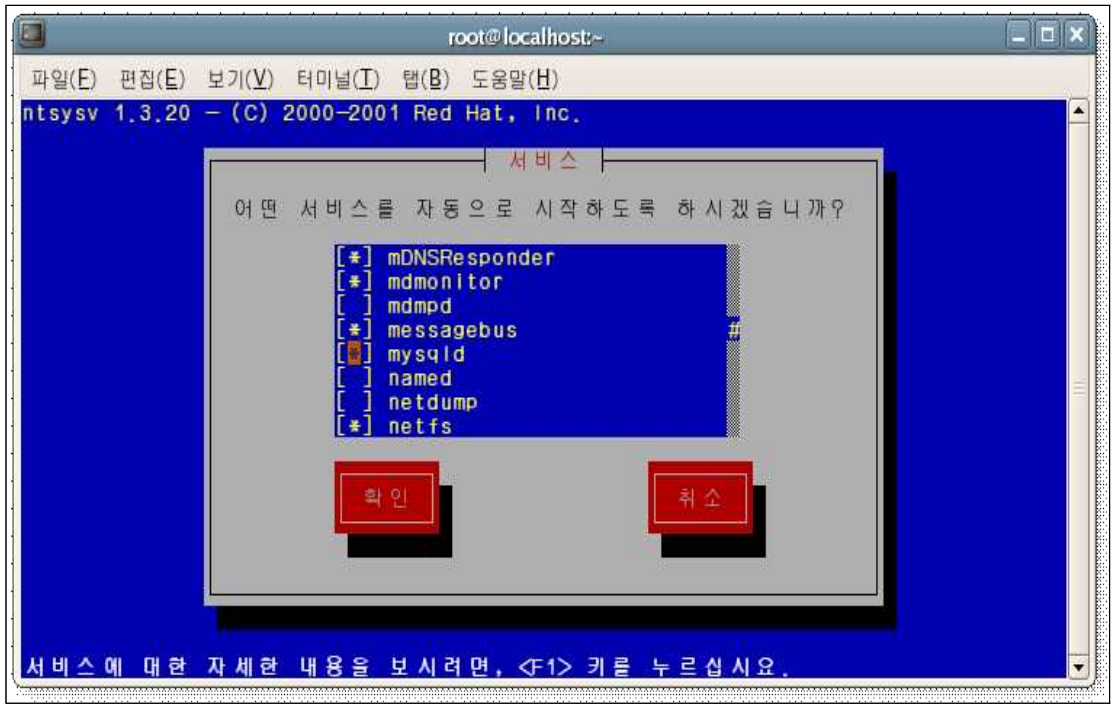
시스템 부팅시 httpd 서비스를 자동 시작할수 있게 등록한다.



<그림 3-5-2>

ntsysv

시스템 부팅시 mysqld 서비스를 자동 시작할수 있게 등록한다.



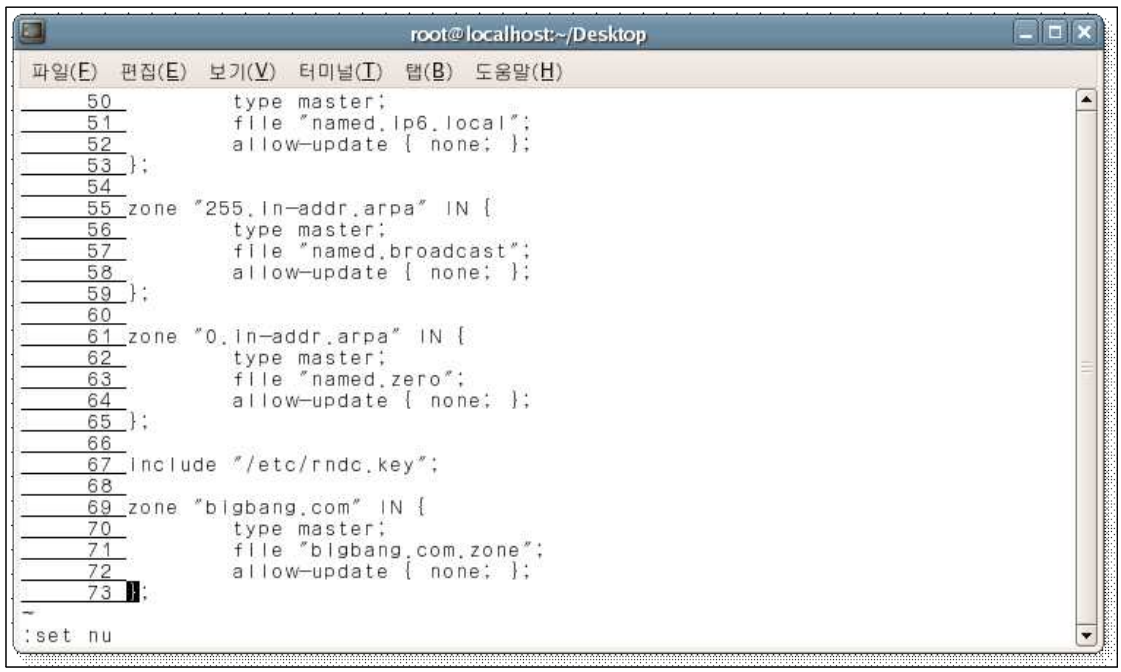
<그림 3-5-3>

차후 자세한 설정은 침입탐지시스템(IDS) 부분에서 자세히 설명합니다.

3.6 DNS 설정

3.6.1 NAT 서버 내 DNS 설정

vi /etc/named.conf

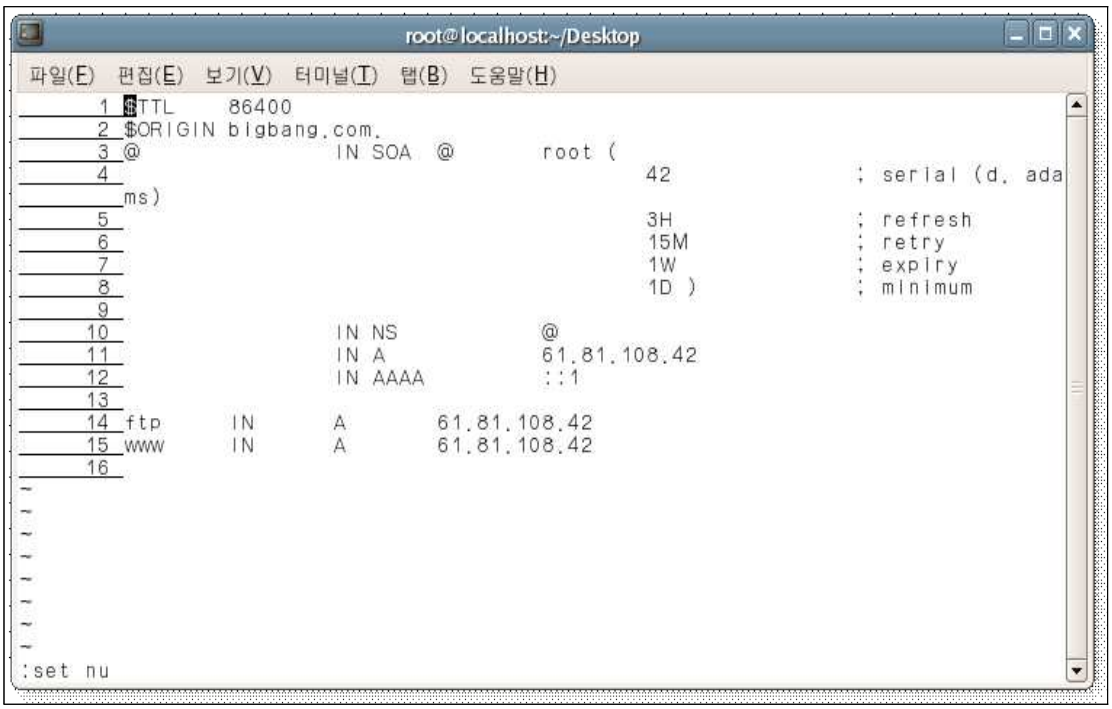


<그림 3-6-1>

```
zone "bigbang.com" IN {
 type master;
 file "bigbang.com.zone";
 allow-update { none; };
};
```

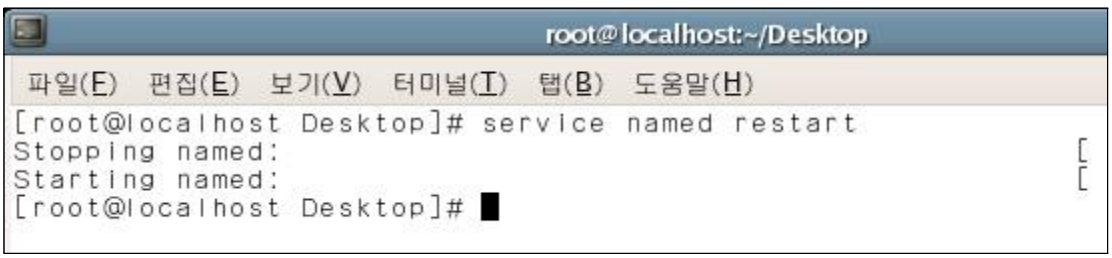
위와 같이 DNS이름과 zone 파일의 이름을 설정해준다.
zone 파일을 포함하는 폴더 - /var/named/
zone 파일 이름 - bigbang.com.zone

vi bigbang.com.zone



<그림 3-6-2>

2번행 - $ORIGIN bigbang.com. 으로 DNS이름을 설정. 마지막 (.)을 꼭 표기
11번행 - IN A 61.81.108.42 -> 61.81.108.42는 NAT서버의 주소
14번행 - ftp IN A 61.81.108.42 -> ftp(21번포트)로 접근하는 DNS는
61.81.108.42를 변환15번행 - www IN A 61.81.108.42 ->
www(80포트)로 접근하는 DNS는 61.81.108.42로 변환



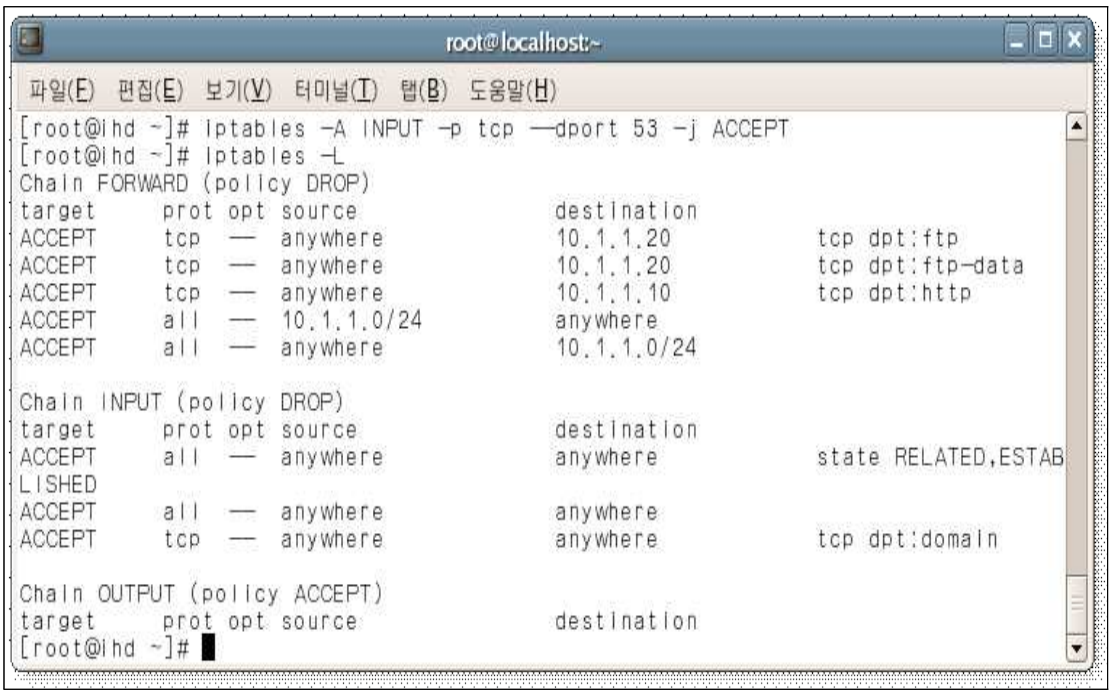
<그림 3-6-3>

service named restart

3.6.2 iptables를 이용하여 NAT 서버 DNS포트 설정

iptables -A INPUT -p tcp --dport 53 -j ACCEPT

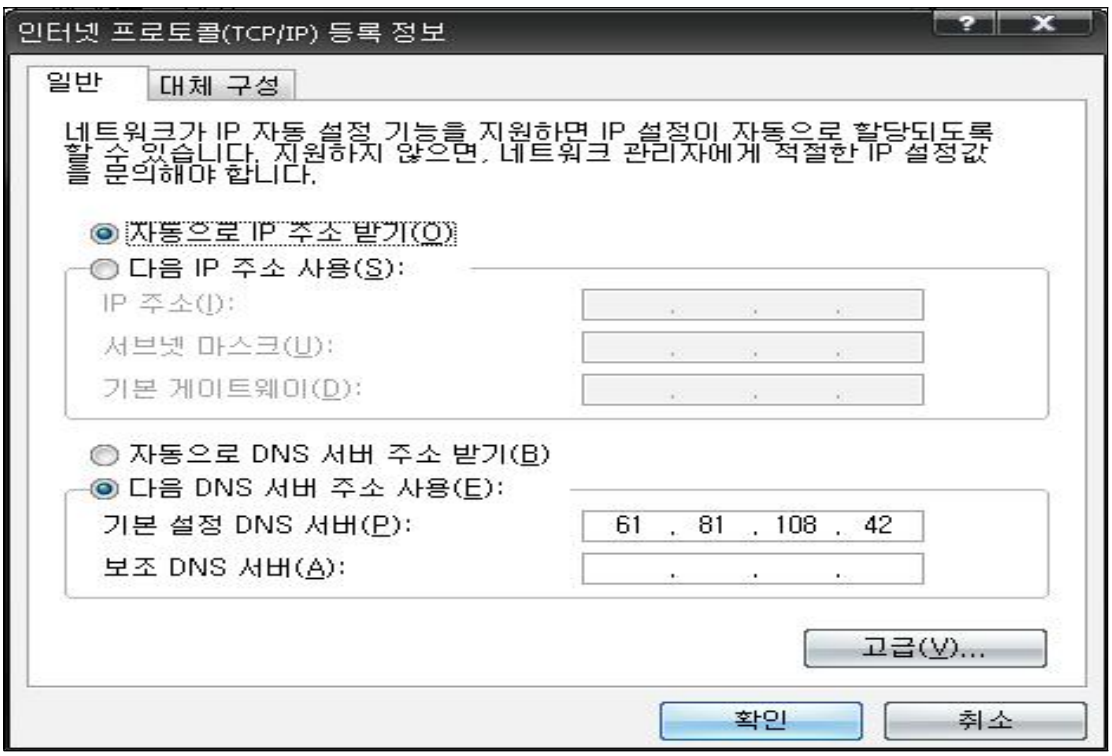
DNS사용을 위한 DNS포트(53번)의 접근을 허용



<그림 3-6-4>

3.6.3 외부사용자 DNS 설정

DNS주소를 61.81.108.42(NAT서버 컴 주소)로 설정



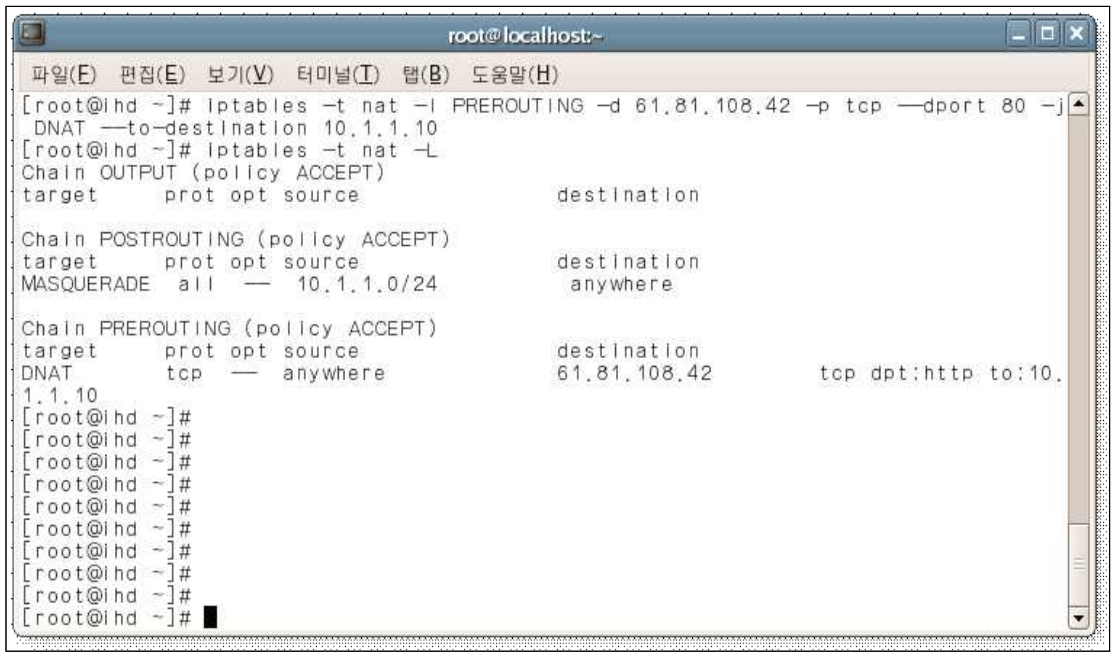
<그림 3-6-5>

3.7 각 Port를 이용한 외부사용자의 NAT 접속

3.7.1 웹 서버

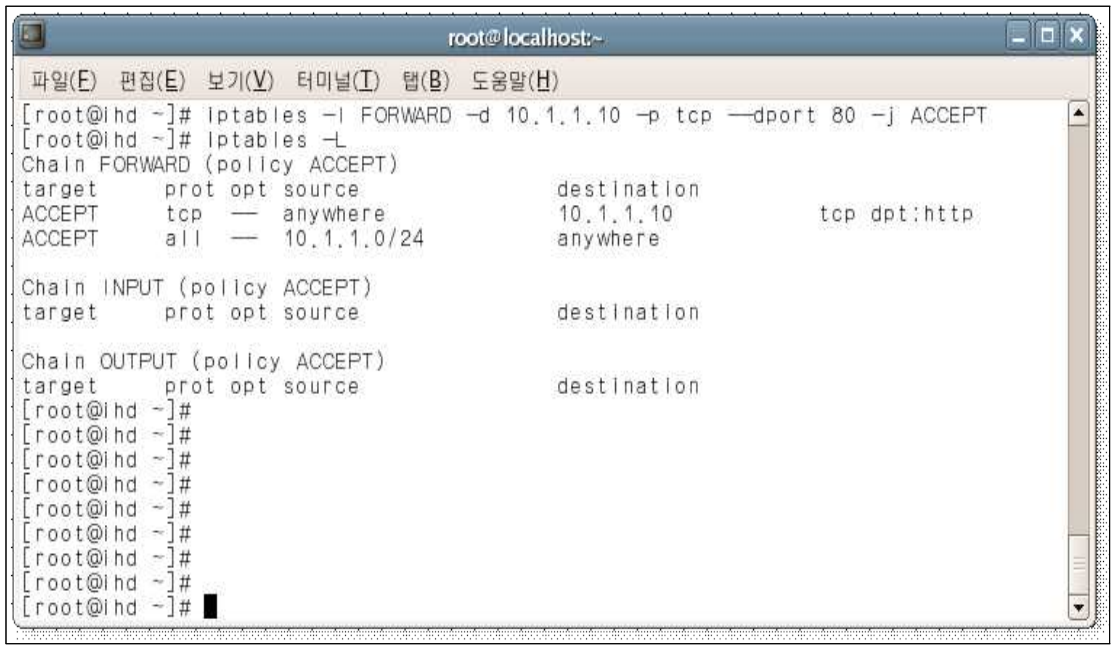
iptables -t nat -I PREROUTING -d 61.81.108.42 -p tcp --dport 80
-j DNAT --to-destination 10.1.1.10

외부망에서 NAT 서버주소(61.81.108.42)로 접속시 80번 포트로 접속을 시도하면
NAT망 내의 10.1.1.10으로 경로를 변경한다.



<그림 3-7-1>

iptables -I FORWARD -d 10.1.1.10 -p tcp --dport 80 -j ACCEPT
위에서 변경한 경로를 따라 10.1.1.10으로 갈수있게 80번 포트를 허용한다.



<그림 3-7-2>

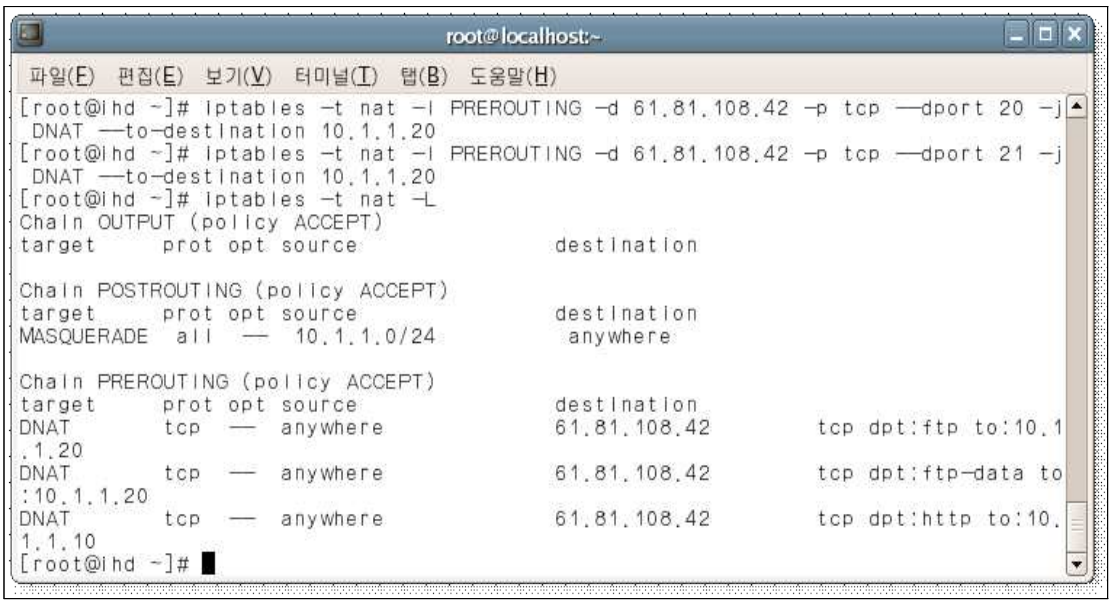
3.7.2 ftp 서버

iptables -t nat -I PREROUTINFG -d 61.81.108.42 -p tcp --dport 20 -j DNAT --to-destination 10.1.1.20

외부망에서 NAT 서버주소(61.81.108.42)로 접속시 20번 포트로 접속을 시도하면 NAT망 내의 10.1.1.20으로 경로를 변경한다.

iptables -t nat -I PREROUTINFG -d 61.81.108.42 -p tcp --dport 21 -j DNAT --to-destination 10.1.1.20

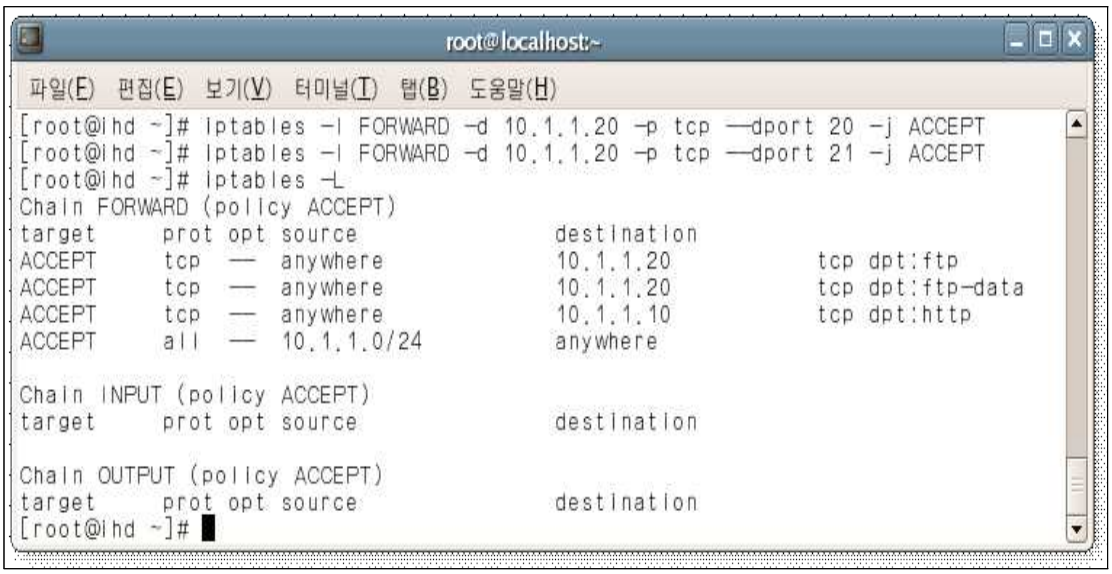
외부망에서 NAT 서버주소(61.81.108.42)로 접속시 21번 포트로 접속을 시도하면 NAT망 내의 10.1.1.20으로 경로를 변경한다.



<그림 3-7-3>

iptables -I FORWARD -d 10.1.1.20 -p tcp --dport 20 -j ACCEPT 위에서 변경한 경로를 따라 10.1.1.20으로 갈수있게 20번 포트를 허용한다.

iptables -I FORWARD -d 10.1.1.20 -p tcp --dport 21 -j ACCEPT 위에서 변경한 경로를 따라 10.1.1.20으로 갈수있게 21번 포트를 허용한다.



<그림 3-7-4>

service iptables restart

<그림 3-7-5>

3.8 NAT서버 IDS(침임탐지시스템) 구축

3.8.1 pcre 설치

snort 가 설치되기 위해서는 아래의 pcre 와 pcre-devel 이 있어야 한다.

rpm -qa |grep pcre <- rpm 설치되어 있는지 확인한다.

yum -y install pcre-devel <- yum를 이용한 rpm 설치

<그림 3-8-1>

3.8.2 libpcap 설치

snort는 libpcap 기반으로 한 스니퍼이기에 libpcap을 먼저 설치한다.

cd /usr/local/src <- 폴더 이동하여 이곳에 다운로드한다.

wget http://www.tcpdump.org/release/libpcap-0.9.7.tar.gz <- 다운로드한다

tar xvfz libpcap-0.9.7.tar.gz <- 압축풀기

cd libpcap-0.9.7 <- 압축 해제한 폴더이동

./configure

echo $? <- 메시지 0이외는 에러 메시지

make && make install

echo $?

3.8.3 snort 설치

Snort란 일종의 침입탐지시스템(IDS:Intrusion Detection System)으로 실시간 트래픽 분석, 프로토콜 분석, 내용검색/매칭, 침입탐지 Rule에 의거하여 오버플로우, 포트스캔, CGI공격, OS확인 시도 등의 다양한 공격과 스캔을 탐지할 수 있다.
침입탐지 Rule은 보안 커뮤니티를 통해 지속적으로 업데이트되고 또한 사용자가 직접 Rule을 작성하여 추가할 수 있도록 설계되어 최신공격에 대한 적응에 빨리 대처할 수 있다.

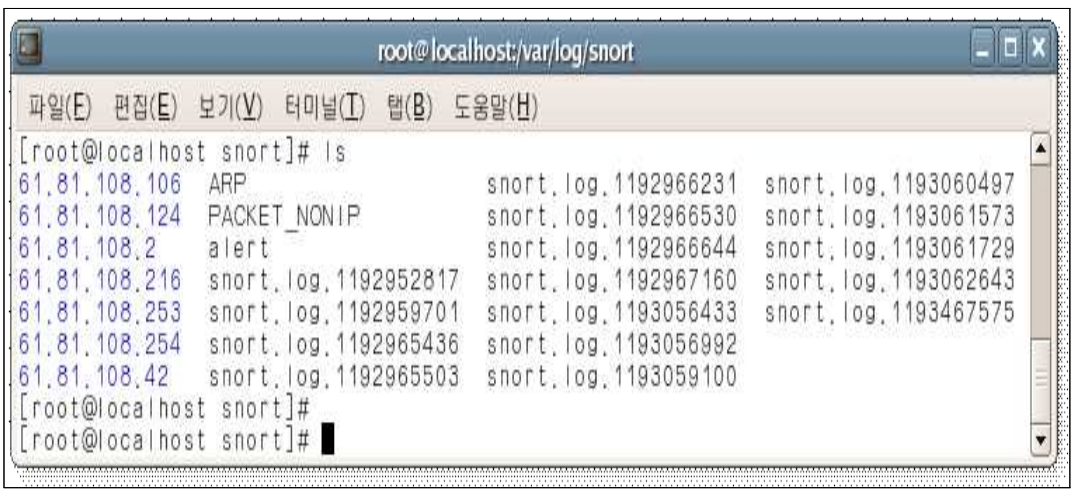
```
# cd /usr/local/src
# wget http://www.snort.org/dl/current/snort-2.8.0.tar.gz
# tar  xvzf  snort-2.8.0.tar.gz
# cd  snort-2.8.0
# ./configure  &&  make  &&  make install
# echo $?
```

○ Sniff mode

1~3 계층에서 활동
```
# snort -v
```
3~7 계층까지 활동.
```
# snort  -vd
```
3~7 계층까지 활동하면서 MAC ADDRESS 까지 봄
```
# snort  -vde
```

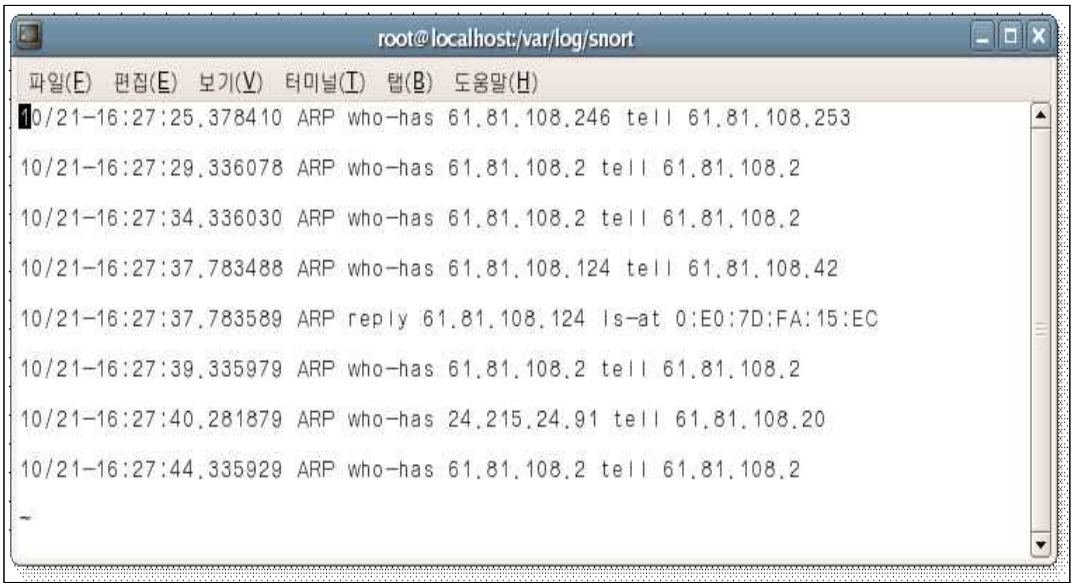
○ Logging mode

sniff mode 에서는 화면상에 뿌려주기만 하지만 logging mode 에서는 파일로 저장이 가능하다.
유형 : ascii(문자형태로 저장), pcap(2진형태로 저장), none(사용않함)
```
# snort -vde -K ascii
```
아래의 경로로 가보면 해당 아이피에 대한 각각의 폴더 안에 로그들이 저장된다.
```
# ls  /var/log/snort/
```



<그림 3-8-2>

cat /var/log/snort/ARP



<그림 3-8-3>

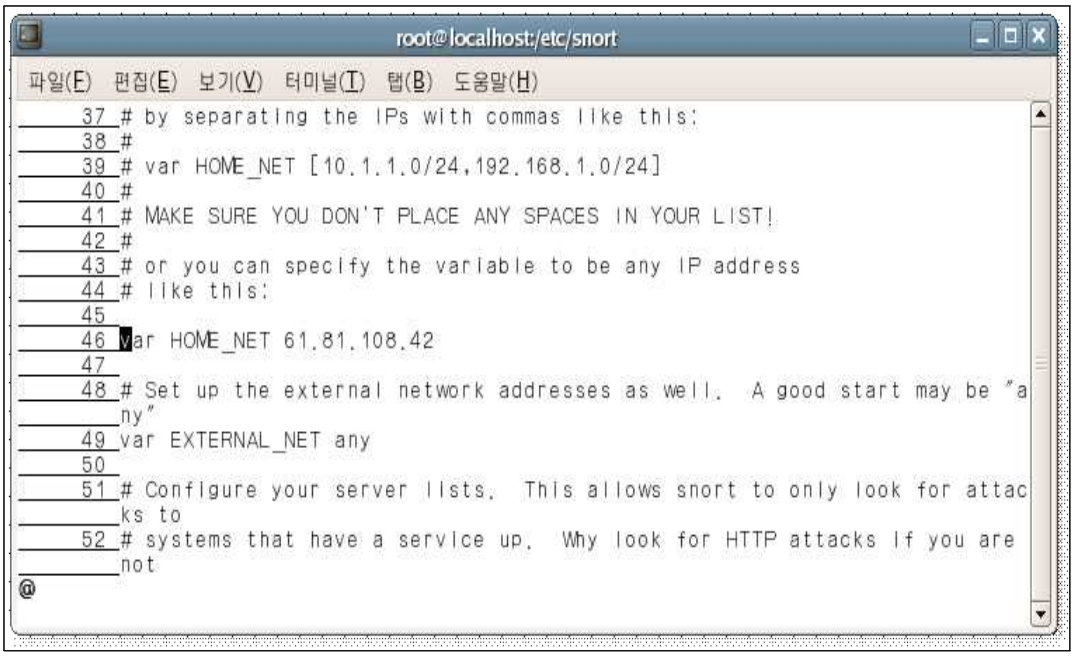
○ 환경설정

snort 환경설정 파일이 압축을 푼 디렉토리 안에 존재 한다.
cd /usr/local/src/snort-2.8.0/etc

폴더를 하나 만들어 그쪽으로 복사해서 관리한다
mkdir /etc/snort
cp ./* /etc/snort && cd /etc/snort

환경설정 파일을 수정을 한다.
vi snort.conf

홈네트워크 즉 감시할 네트워크 및 IP 설정
-> 46행 var HOME_NET 61.81.108.42

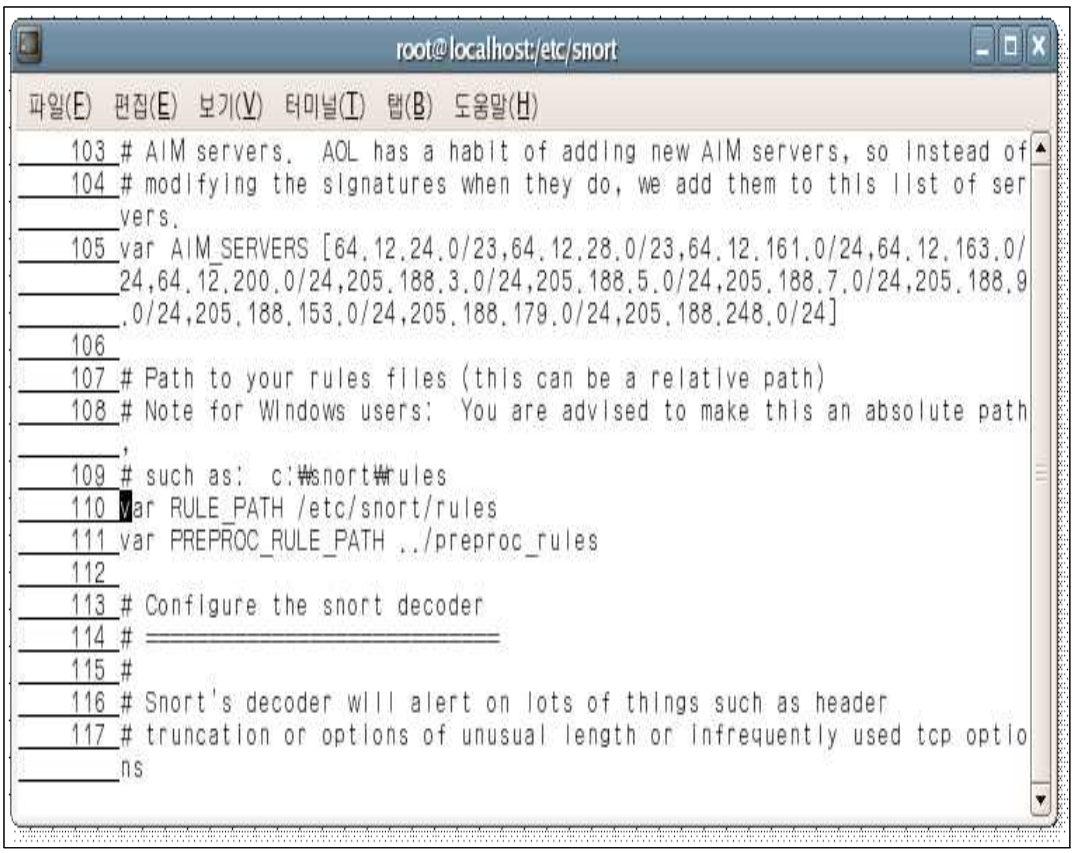


<그림 3-8-4>

snort 는 해커들의 공격방법을 룰로 저장하여 그 정보대로 검색후 차단을 하기에 룰이 중요하다

유료는 룰이 많지만 무료 룰을 사용한다

-> 110행 var RULE_PATH /etc/snort/rules



<그림 3-8-5>

이렇게 설정을 하고 나왔지만 정작 룰이 없기에 에러가 난다. 밑에서 다운을 받자

snort -c snort.conf <- rule 이 없기에 error 남

○ 룰 설치

무료 룰 다운로드

cd /etc/snort/

#wget http://www.snort.org/pub-bin/downloads.cgi/Download/vrt_pr/
snortrules-pr-2.4.tar.gz

룰을 다운받아 해당 경로에 붙여넣고 다시한번 snort 를 실행 이제 에러 없이 잘 작동한다

tar xvzf snortrules-pr-2.4.tar.gz

snort -c snort.conf

vi /etc/snort/rules/web-misc.rules

97행, 98행, 452행 #(주석)처리

<그림 3-8-6>

snort -c snort.conf <- 동작이 잘 되는지 확인후 정지

snort 가 동작되면 포그라운드에서 작업을 할수 없기에 백그라운드 모드에서 실행

snort -c snort.conf -D -> 백그라운드로 동작

ps -ef | grep snort -> 백그라운드로 동작 확인

직접 도스 공격이라든가 알려진 해킹공격을 시도하면 그때는 로그에 저장이 됨을 알수 있다.

ls /var/log/snort

3.8.4 snort 와 mysql DB 연동

○ NAT 서버

server컴 DB연동 관련 패키지 설치 - mysql-devel, perl-DBD-MySQL

yum -y install mysql-devel perl-DBD-MySQL

snort 재컴파일

cd /usr/local/src/snort-2.8.0/doc

more INSTALL

cd ..

make clean <- 기존 컴파일 된 파일 삭제

./configure --with-mysql && make && make install

echo $?

snort.conf 설정

vi /etc/snort/snort.conf

snort.conf 파일에 DB설정을 해준다.

DBuser=sdb

DBpassword=sdb123

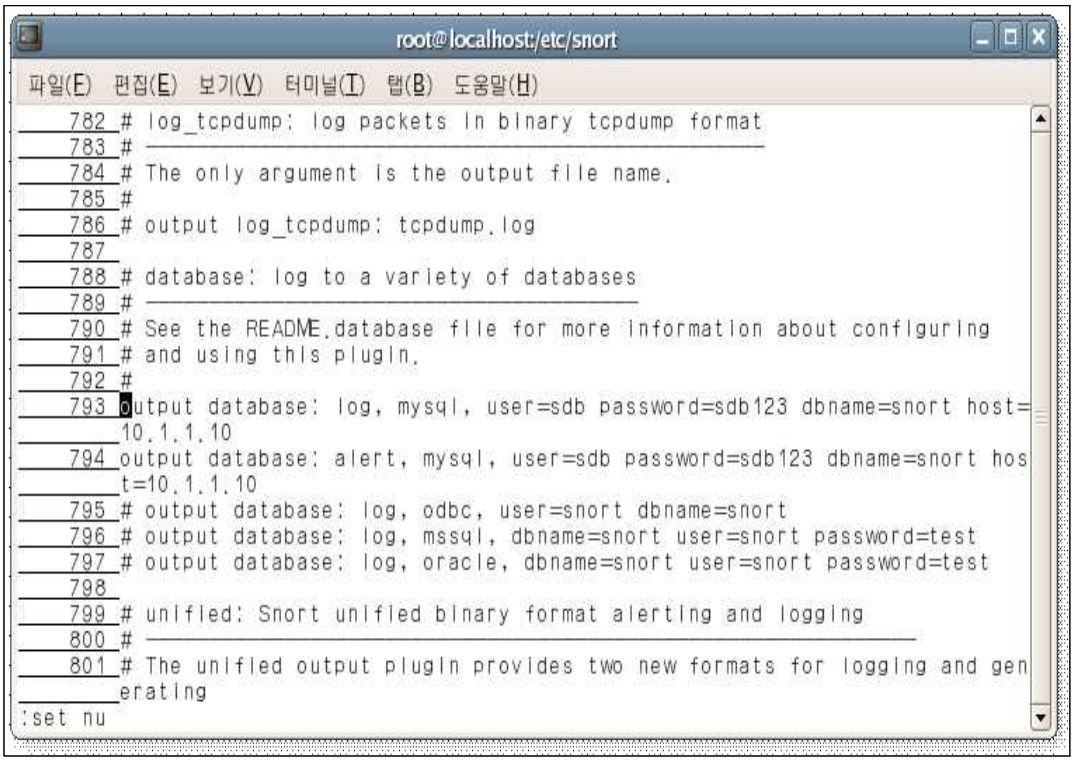
dbname=snort

DBhost=10.1.1.10

796행 주석제거 후 수정

output database: log, mysql, user=sdb password=sdb123 dbname=snort host=10.1.1.10

output database: alert, mysql, user=sdb password=sdb123 dbname=snort host=10.1.1.10



<그림 3-8-7>

○ www 서버

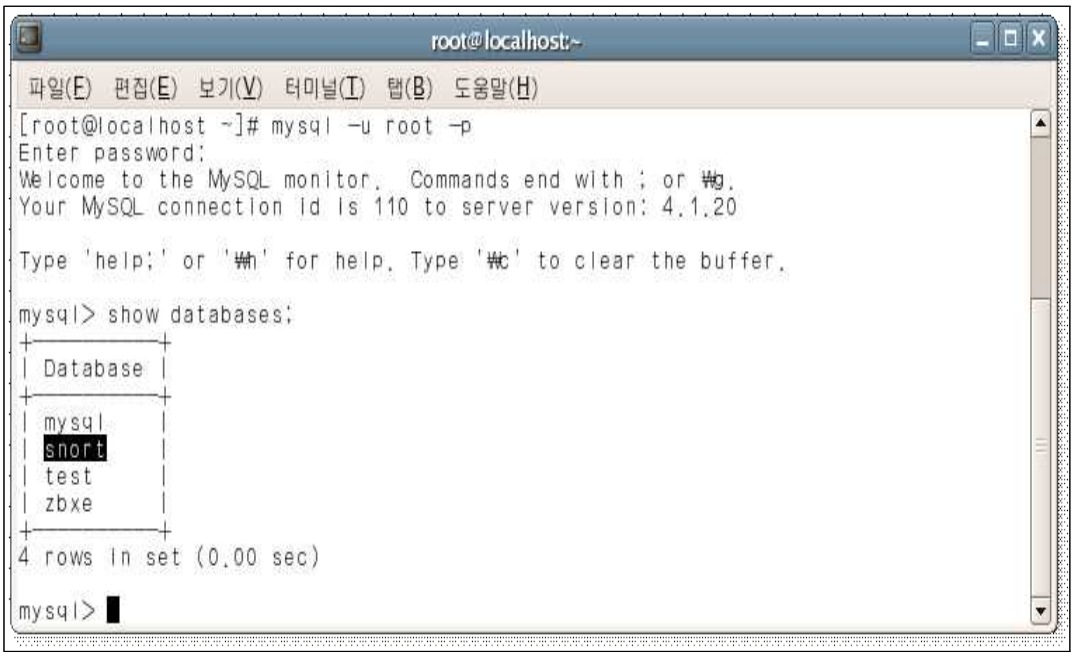
mysql-server 설치

yum -y install mysql-server

<그림 3-8-8>

mysqld 시작
service mysqld start
mysql의 root 암호 설정
mysql -u root
mysqladmin -u root password 1234

DB생성
mysql -u root -p
> show databases; <- mysql DB리스트를 보여줌
> create database snort; <- snort DB 생성
> show databases;



<그림 3-8-9>

> use snort; <- snort DB 접근
> show tables; <- snort의 tables 리스트 보여줌
NAT 서버에서 create_mysql 파일을 www서버로 보내준다
cd /usr/local/src/snort-2.7.0.1/schemas
scp ./create_mysql root@10.1.1.10

snort DB에 create_mysql 파일을 내용을 보내줌으로 snort에 대한 tables를 생성한다.
cd /root
mysql -u root -p snort < create_mysql
mysql -u root -p
> use snort;
> show tables;
아래와 같이 tables의 검색을 하면 아무 내용이 없을 것이다
> select * from data; <- data tables 검색
> select * from signature;

DBuser 및 DB암호 생성

> use mysql;

> show tables;

> select host,user,password from user; <- DBuser 검색

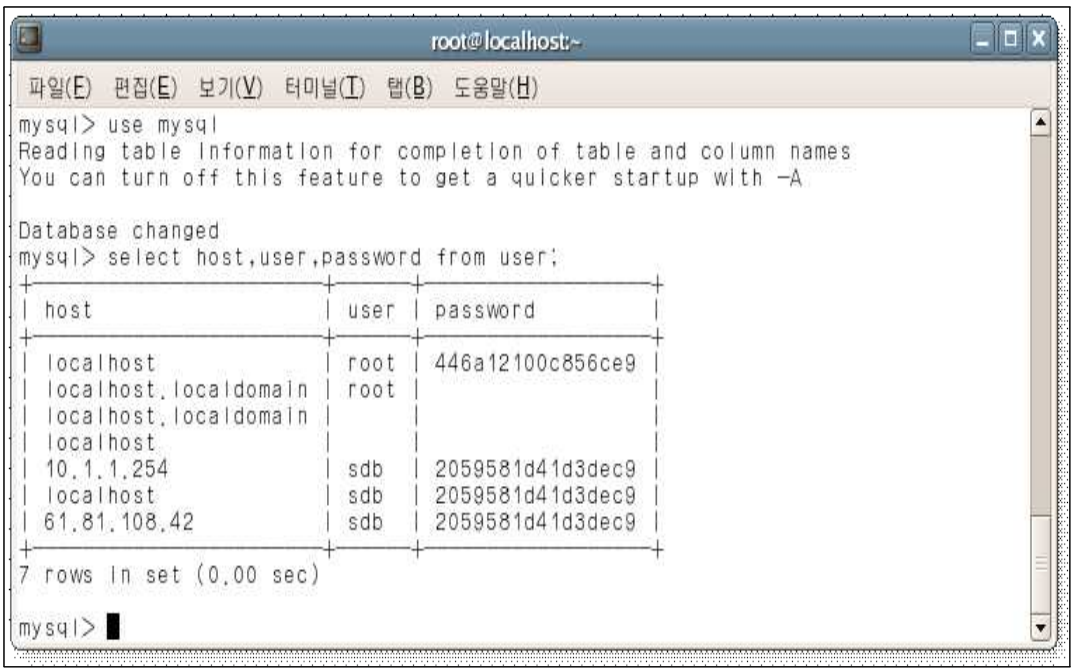
> grant CREATE,DELETE,SELECT,UPDATE,INSERT on snort.* to sdb@61.81.108.42;

> grant CREATE,DELETE,SELECT,UPDATE,INSERT on snort.* to sdb@localhost; <- DBuser 생성 및 관한 설정

> set password for sdb@61.81.108.42=PASSWORD('sdb123'); <- DBuser에 대한 암호 생성

> set password for sdb@localhost=PASSWORD('sdb123');

> select host,user,password from user;

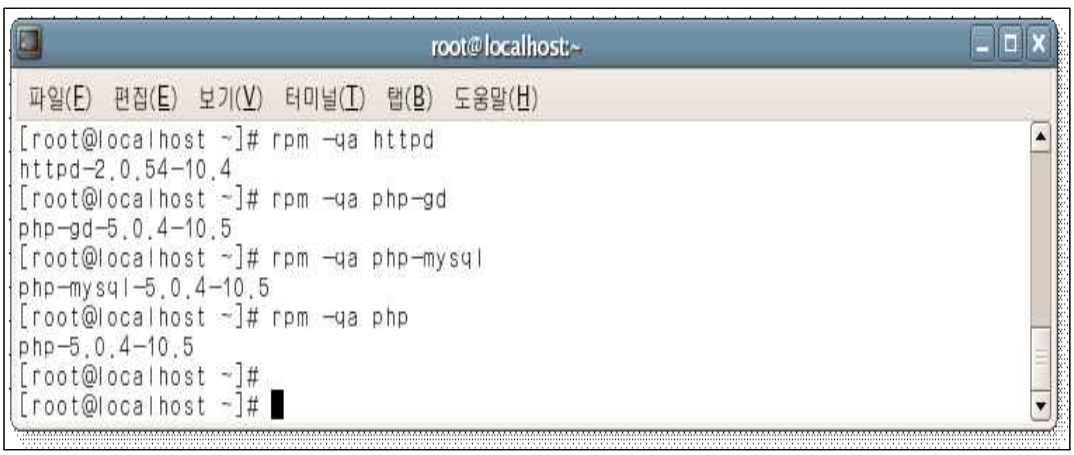


<그림 3-8-10>

3.8.5 www 서버 base, apm 연동

apm 패키지(httpd, php-gd, php-mysql, php)설치

yum -y install httpd php-gd php-mysql php



<그림 3-8-11>

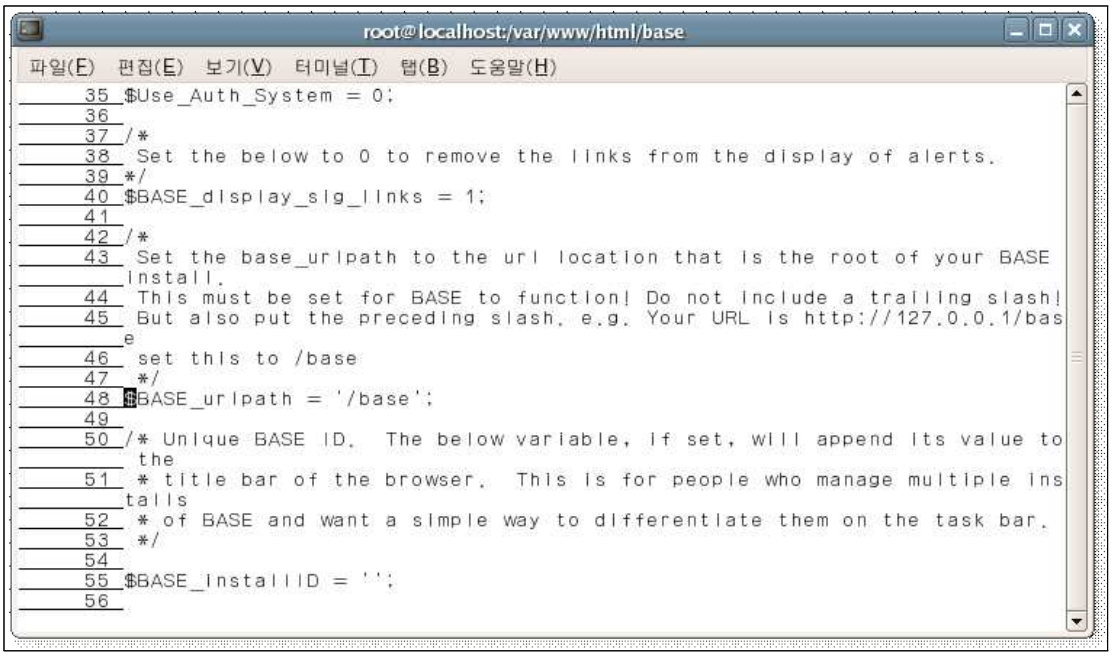
adodb, base source 설치

```
# cd  /usr/local/src
# wget  http://easynews.dl.sourceforge.net/sourceforge/adodb/adodb495a.tgz
# wget  http://easynews.dl.sourceforge.net/sourceforge/secureideas/base-1.3.8.tar.gz
# tar xvzf adodb495a.tgz  -C  /var/www
# tar xvzf base-1.3.8.tar.gz  -C  /var/www/html
# cd /var/www/html
# mv  base-1.3.8  base
# cd  base
# cp  base_conf.php.dist base_conf.php
```

base source 환경설정

```
# vi  base_conf.php
```

48행 수정 $BASE_urlpath = '/base';
70행 수정 $DBlib_path = '/var/www/adodb';
92행 수정 $alert_dbname = 'snort';
95행 수정 $alert_user = 'sdb';
96행 수정 $alert_password = 'sdb123';



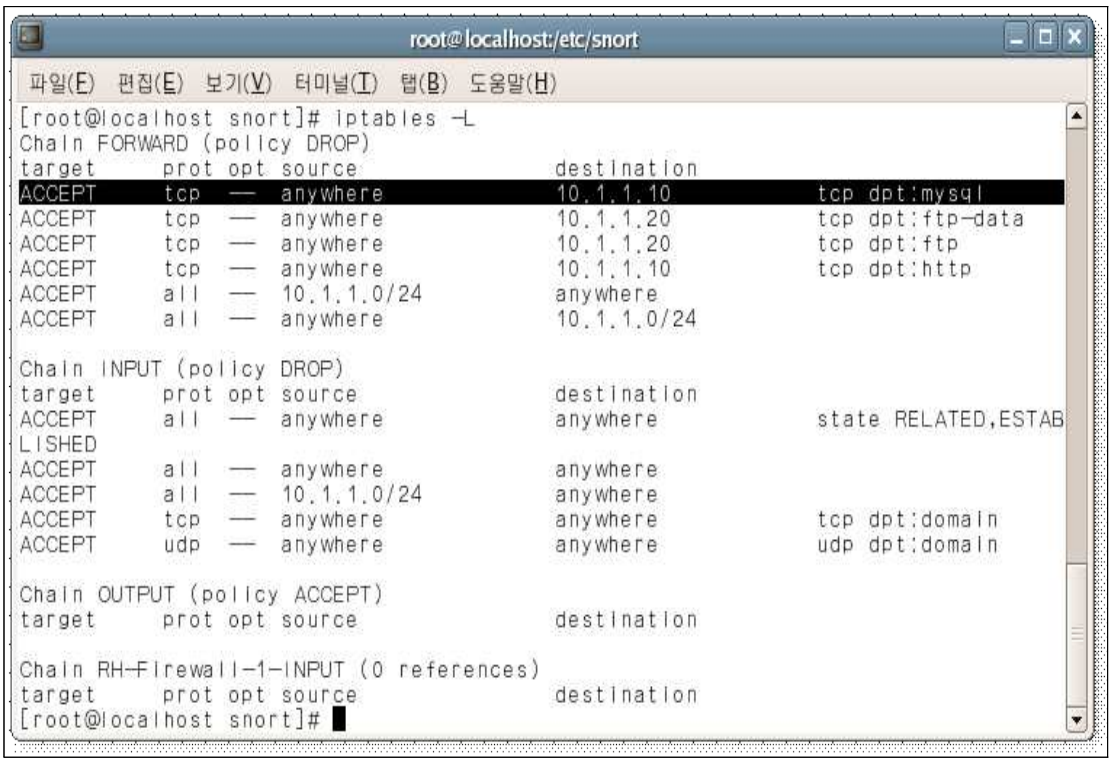
<그림 3-8-12>

```
service  mysqld  restart
service  httpd  restart
```

<그림 3-8-13>

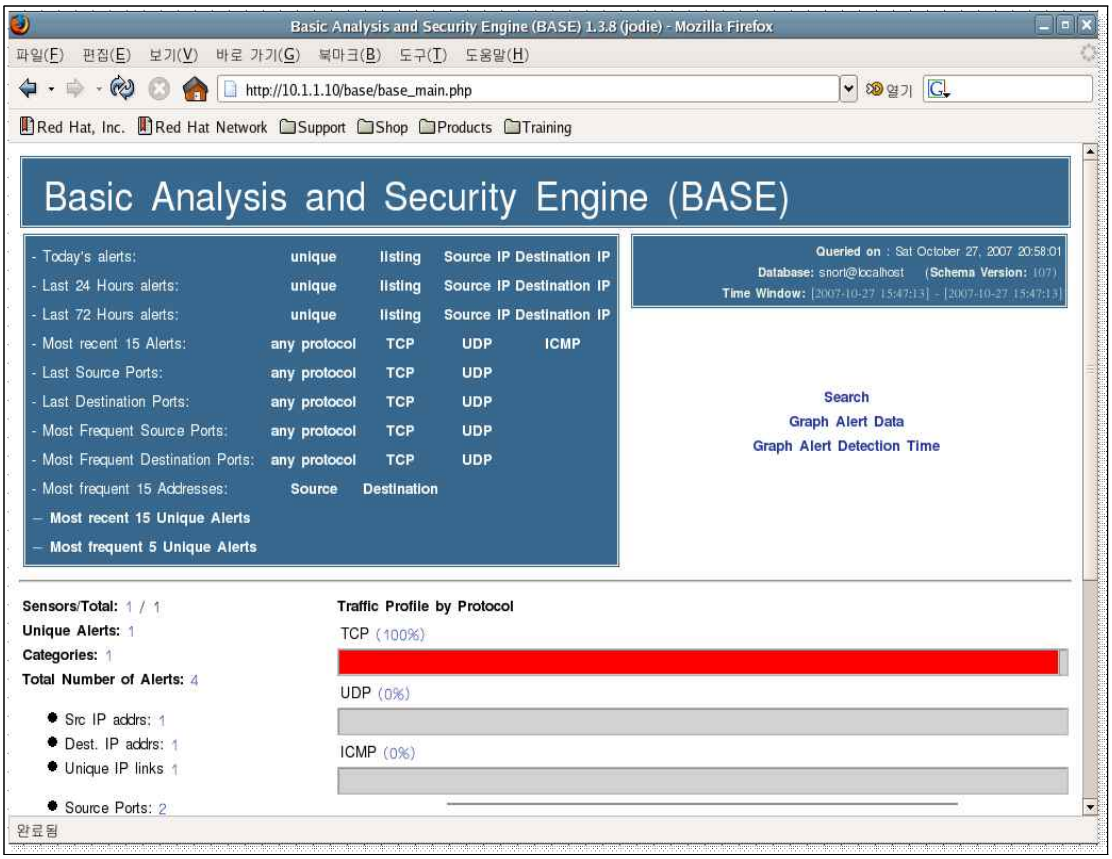
NAT서버에서 www서버로 mysql포트 통로 허용한다.

iptables -I FORWAED -d 10.1.1.10 -p tcp --dport 3306 -j ACCEPT



<그림 3-8-14>

웹브라우저에서 http://10.1.1.10/base 접속



<그림 3-8-15>

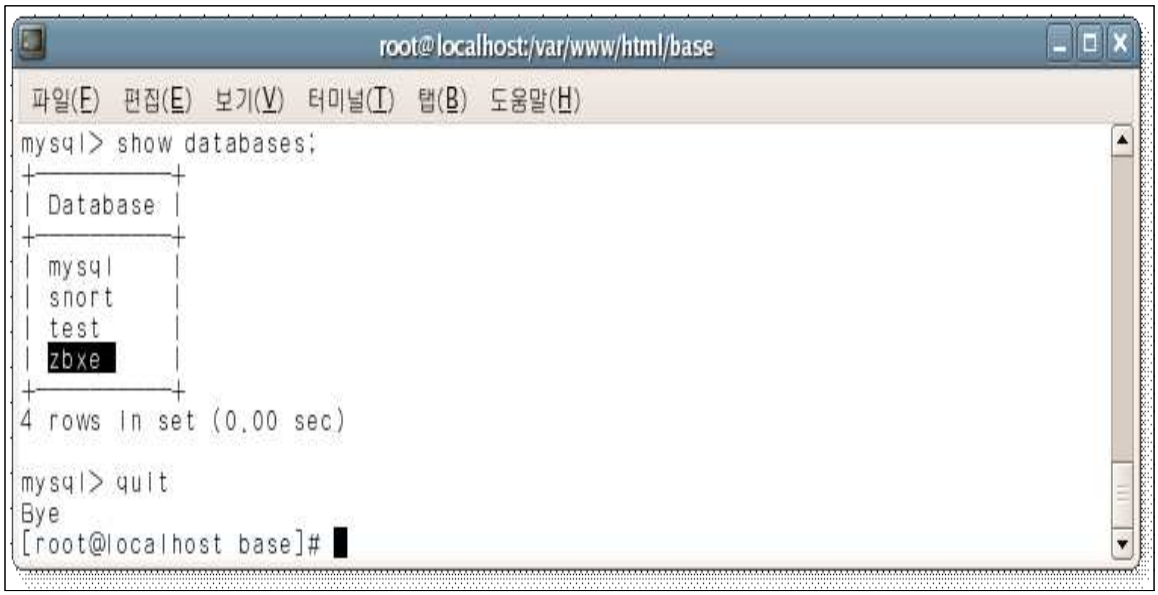
3.9 제로보드XE 설치

제로보드를 설치하기 전에 www서버에 DB생성을 해준다.

mysql -u root -p

> show databases;

> create dtatbase zbxe;



<그림 3-9-1>

설치 및 작동 환경

PHP 4.x 또는 PHP 5.x (단 PHP 5.2.2는 PHP의 버그로 인하여 설치가 불가능하다)

- XML 라이브러리 (필수)
- ICONV (선택. 다만 특정 기능에서 이상현상을 보일 수 있다)
- GD (필수. 이미지 변환 기능을 위해 필수이다)

DATABASE

- MySQL 4.1 이상 : UTF-8을 사용하기 위해 MySQL은 4.1 이상만 지원합니다. (4.x 불가능)
- Sqlite2 or Sqlite3 : Sqlite를 사용하기 위해서는 PHP에 Sqlite extension이 설치되어 있어야 한다.
- Cubrid

제로보드xe 다운로드 및 설치

cd /var/www/html

wget http://downloads.sourceforge.net/zbxe/zbxe.beta.0.2.3.tgz

tar xvfz zbxe.beta.0.2.3.tgz

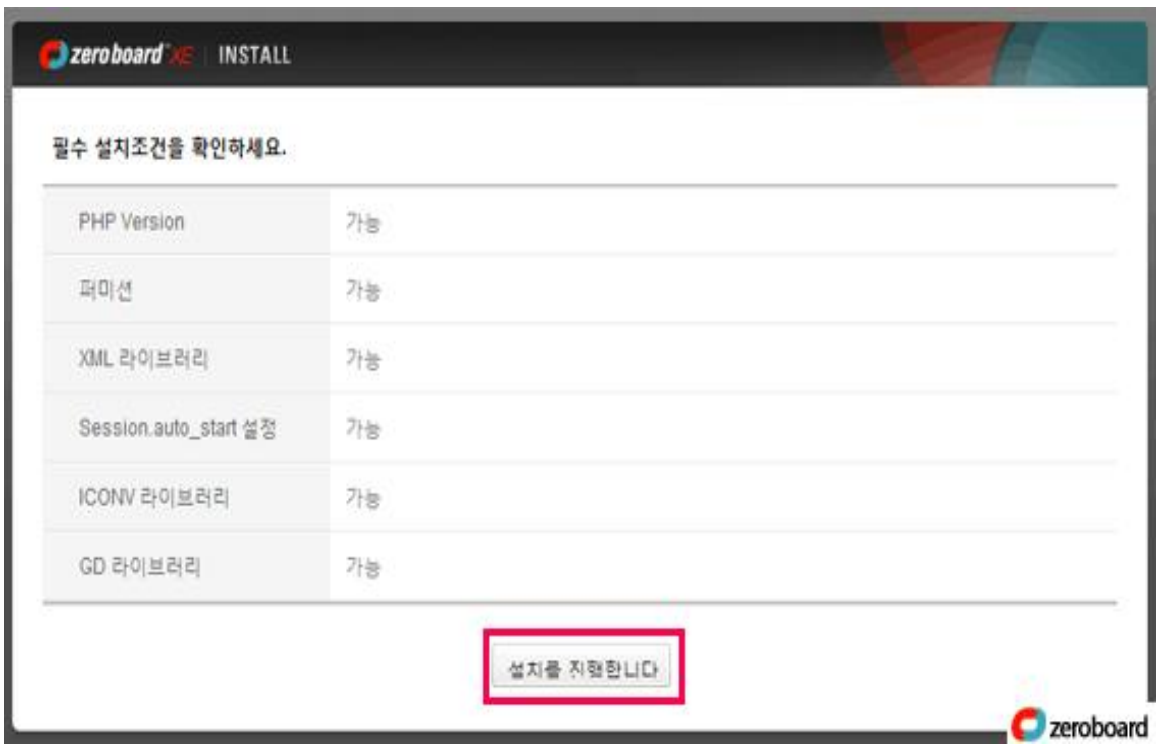
mv zbxe.beta.0.2.3 zbxe

1. 웹 브라우저를 실행하여 주소창에 아래와 같이 입력한다.
   http://www.abc.com/zbxe

2. 홈페이지에서 사용하실 1)언어를 선택하시고 2)라이센스에 동의 한다.
제로보드 XE에서는 한국어,영어,중국어,프랑스어,스페인어,일어 를 지원하고 있다.

<그림 3-9-2>

3. 설치 가능 여부를 확인한 후 "설치를 진행 합니다" 버튼을 클릭한다.



<그림 3-9-3>

4. 사용하시려는 DB를 선택하신 후 "설치를 진행합니다" 버튼을 클릭한다.

1) mysql : 대부분 많이 사용하는 Database 입니다. 이 경우 트랜잭션*은 이루어 지지 않는다.

2) mysql_innodb : mysql에서 트랜잭션을 지원한다.

*트랜잭션 : 다중 쿼리를 하나의 쿼리로 볼 때 실행 쿼리 중 실패하는 것이 있을 때 원 상태로 복원이 가능하게 된다.

3) sqlite3_pdo : 별도의 DBMS를 사용하지 않고, Database를 온라인상에 올린다.

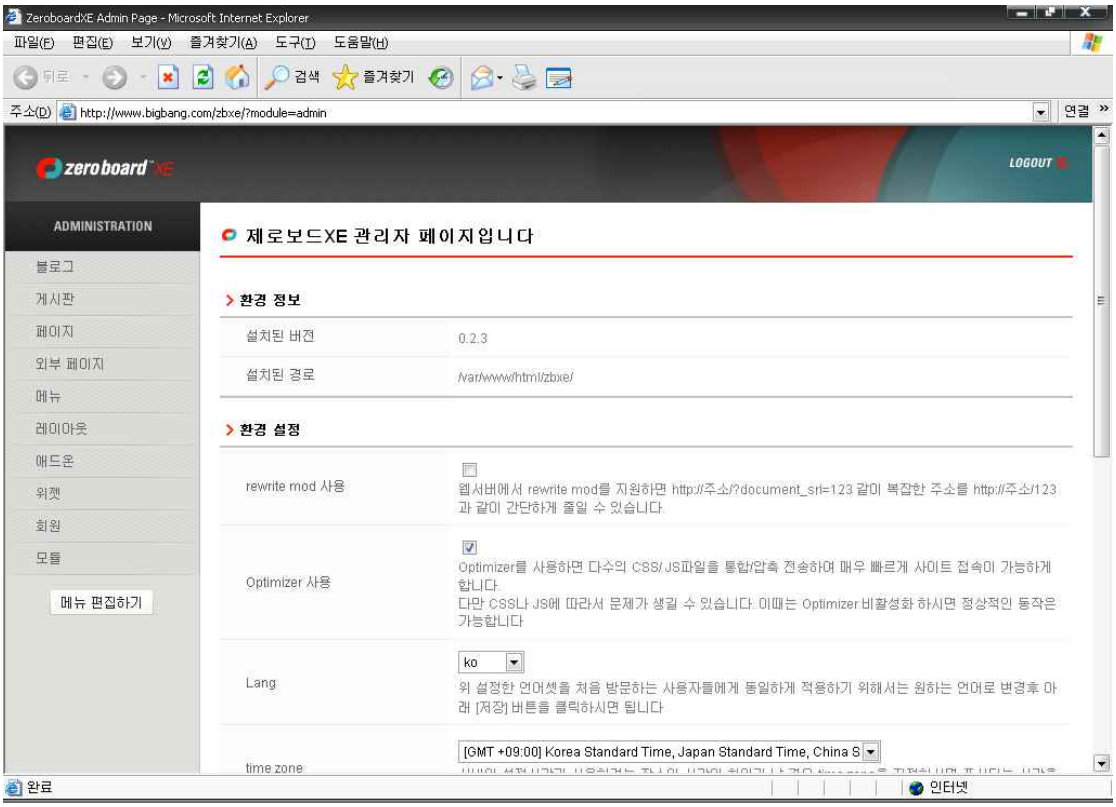
<그림 3-9-4>

5. 밑줄 그어진 부분은 필수적으로 확인하고 기입하셔야 할 부분이며, *(별)표시는 꼭 하지 않으셔도 된다.

* DB 호스트네임 = localhost
* DB아이디 및 암호 = root(1234)
* Database = zbxe <-- 위에서 생성한 Database

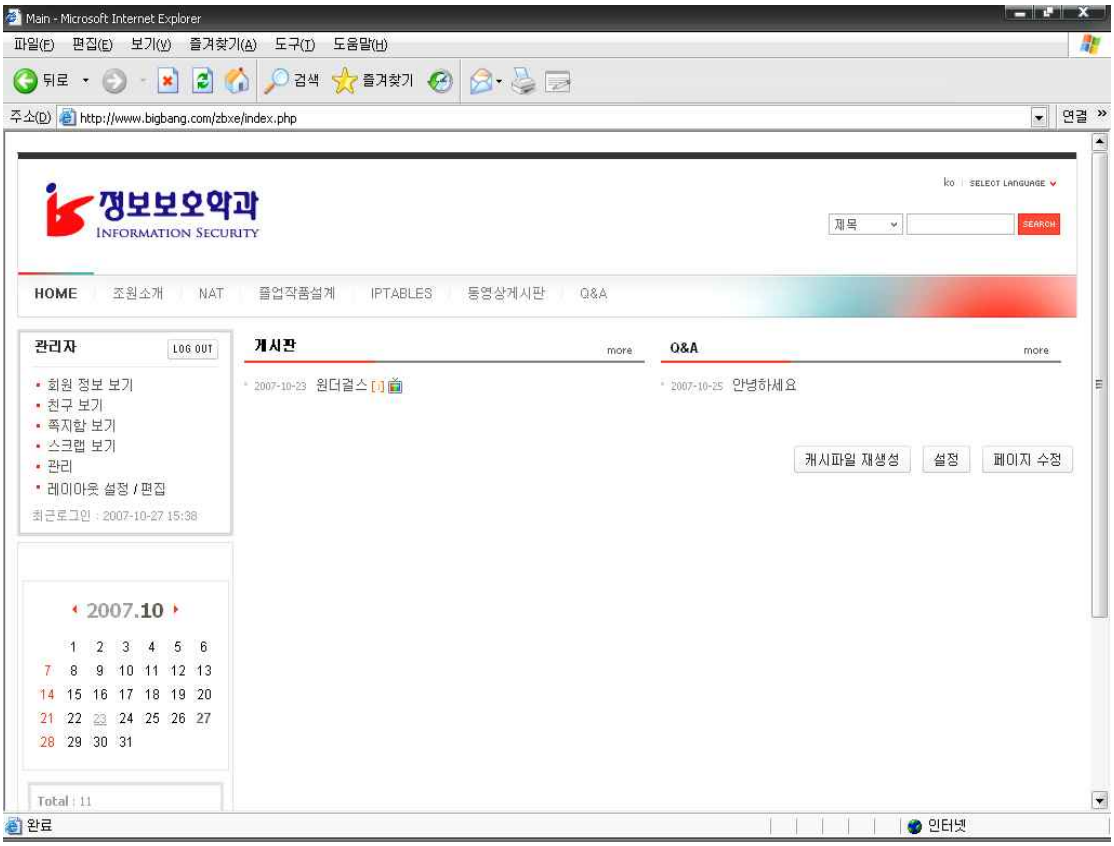
<그림 3-9-5>

* 관리자 페이지



<그림 3-9-6>

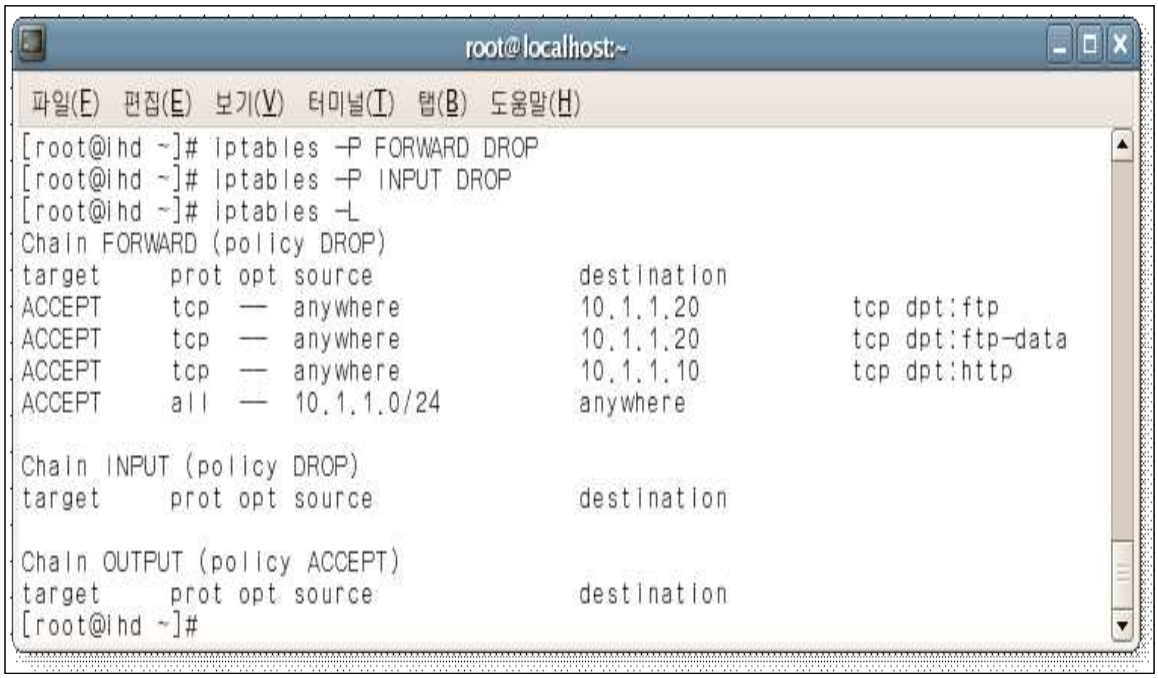
* 외부사용자에서 www서버 홈페이지 접속 화면



<그림 3-9-7>

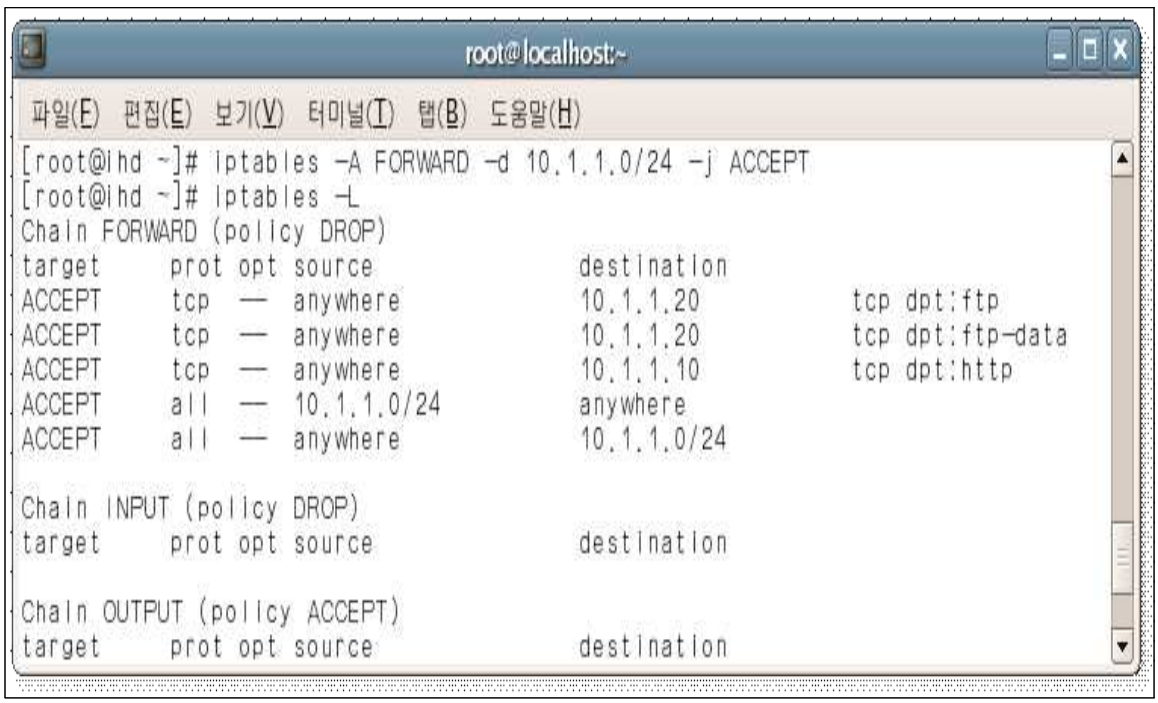
3.10 기타 방화벽 설정

iptables -P FORWARD DROP
iptables의 FORWARD 룰을 차단 한다.
iptables -P INPUT DROP
iptables의 INPUT 룰을 차단 한다.



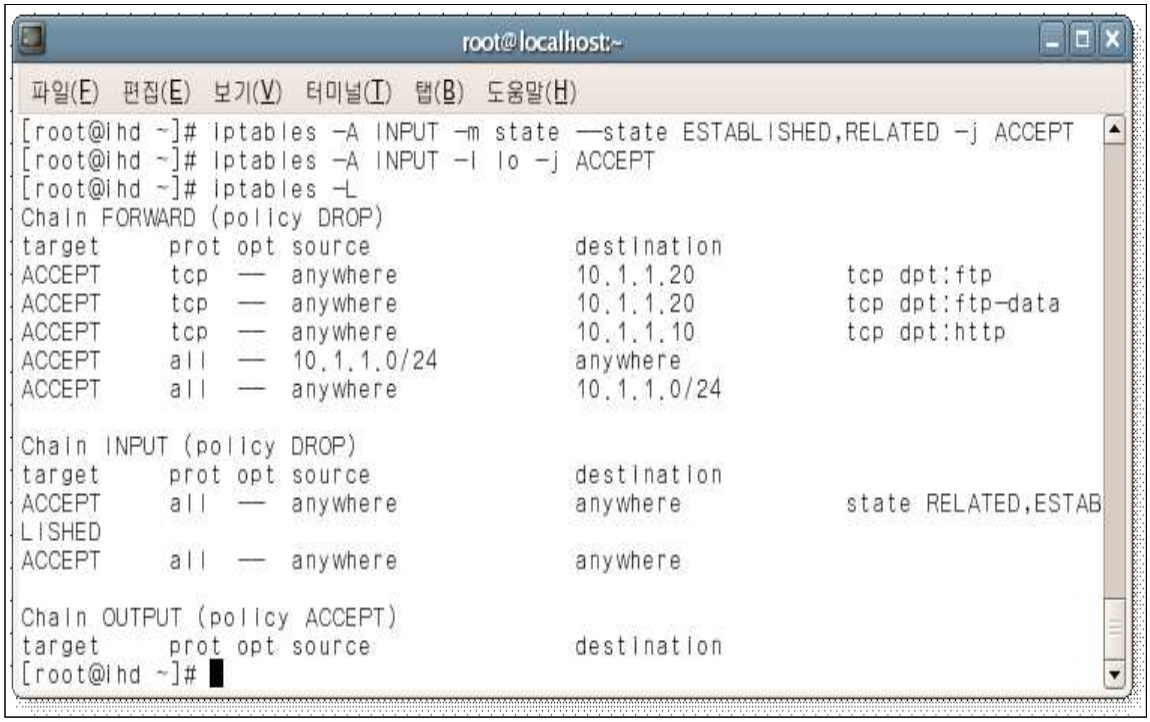
<그림 3-10-1>

iptables -A FORWARD -d 10.1.1.0/24 -j ACCEPT
10.1.1.0/24로 갈수있게 모든 통로를 허용한다.



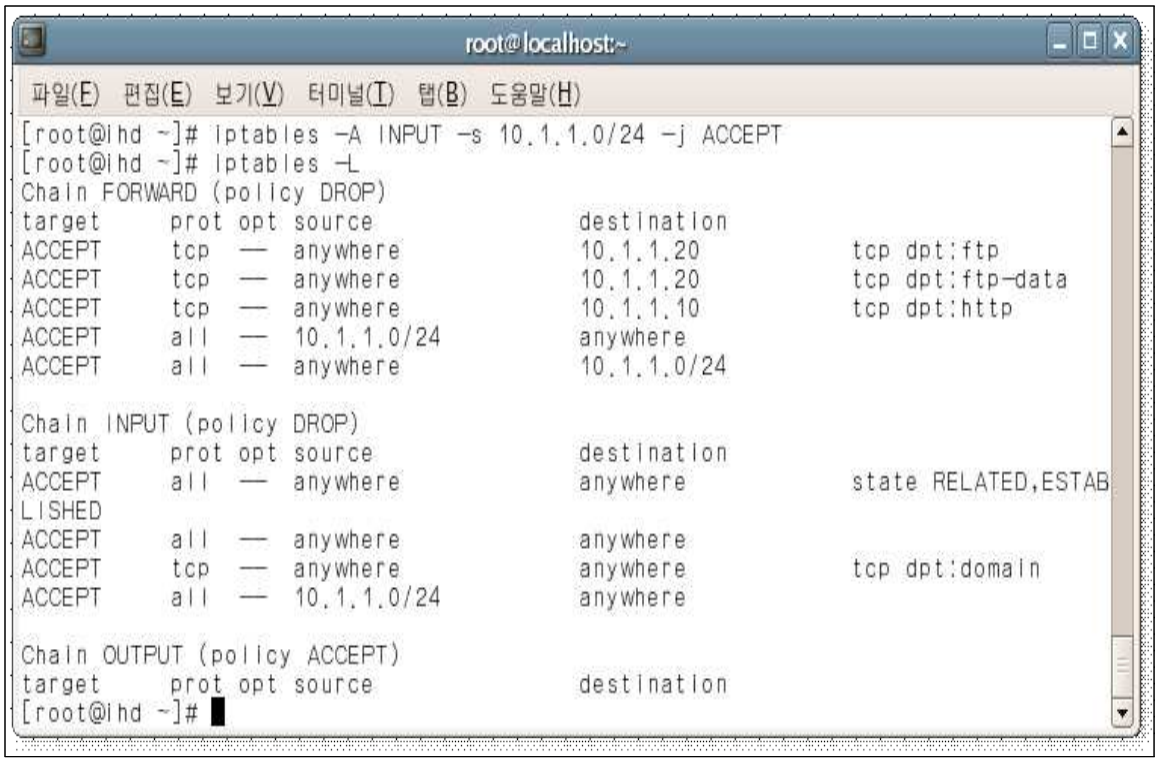
<그림 3-10-2>

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT 접속하는 모든 ESTABLISHED(통신패킷)과 RELATED(ftp데이터패킷)을 허용한다.
iptables -A INPUT -i lo -j ACCEPT
local과 자기자신을 허용한다.



<그림 3-10-3>

iptables -A INPUT -s 10.1.1.0/24 -j ACCEPT
위에서 INPUT룰을 차단함으로 10.1.1.0/24에서 오는 모든 패킷을 허용해줘야 한다.

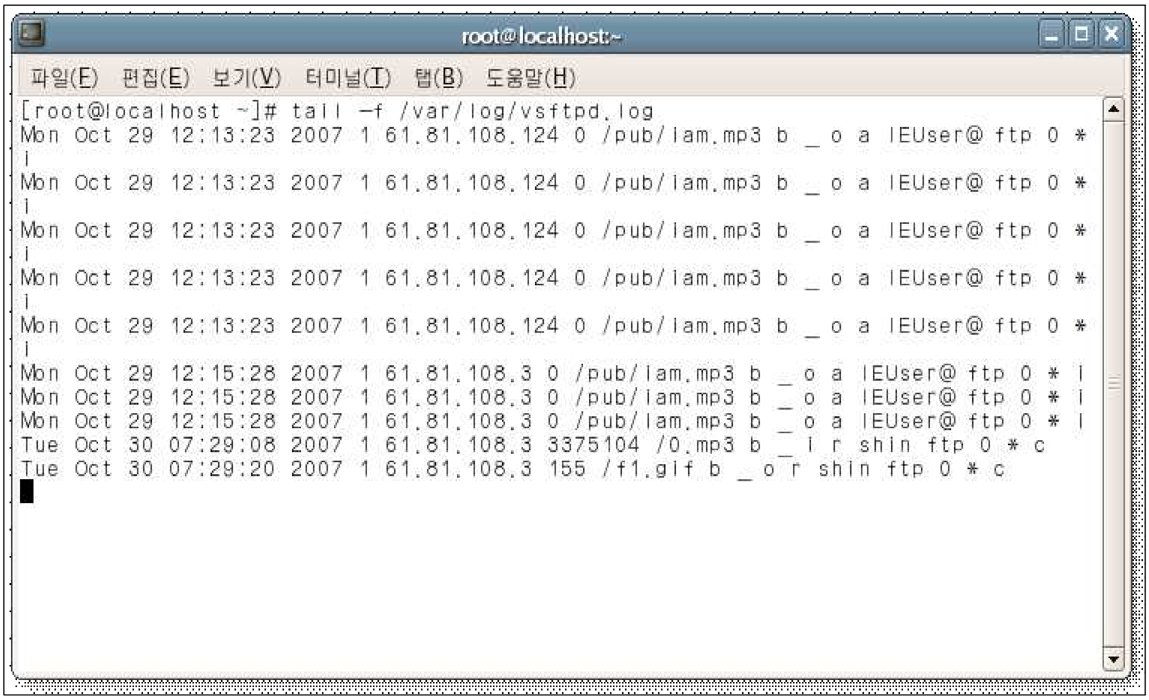


<그림 3-10-4>

4. 분 석

4.1 ftp서버 로그 분석

4.1.1 ftp파일 사용 로그



<그림 4-1-1>

tail -f /var/log/vsftpd.log

실시간으로 ftp의 log를 보기위해 명령어를 실행한다.

위 그림에서 마지막줄을 보면

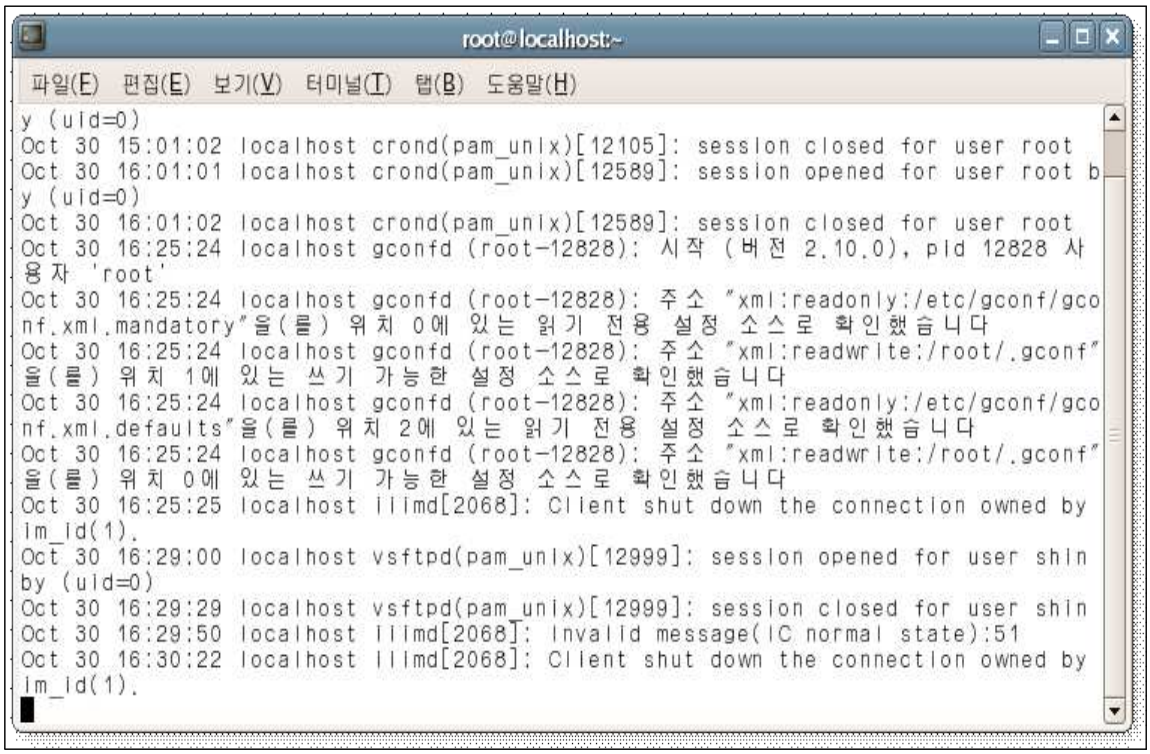
Tue Oct 30 07:29:08 2007 1 61.81.108.3 3375104 /0.mp3 b _ i r shin ftp 0 * c

- Tue Oct 30 07:29:08 2007 : 현재의 연월일 시간을 표기
- 61.81.108.3 : 접속한 사용자의 IP주소
- 3375104 /0.mp3 : ftp서버에 업로드한 파일의 용량 과 이름
- b _ i : b는 바이너리 모드로 업로드 한것이고 i는 업로드 했다는 표시
- shin : ftp에 접속한 사용자ID

Tue Oct 30 07:29:20 2007 1 61.81.108.3 155 /f1.gif b _ o r shin ftp 0 * c

- Tue Oct 30 07:29:08 2007 : 현재의 연월일 시간을 표기
- 61.81.108.3 : 접속한 사용자의 IP주소
- 155 /f1.gif : ftp서버에 업로드한 파일의 용량 과 이름
- b _ o : b는 바이너리 모드로 다운로드 한것이고 o는 업로드 했다는 표시
- shin : ftp에 접속한 사용자ID

4.1.2 ftp접속 로그



<그림 4-1-2>

tail -f /var/log/message

Oct 30 16:29:00 localhost vsftpd(pam_unix)[12999] : session opened for user shin by (uid=0)

- Oct 30 16:29:00 : 현재의 연월일 시간을 표기
- localhost : 서버의 호스트이름
- vsftpd(pam_unix)[12999] : 서버에 설치된 vsftpd 데몬
- session opened for user shin by (uid=0) : 사용자ID가 shin이라는 사용자가 접속 성공

Oct 30 16:29:29 localhost vsftpd(pam_unix)[12999] : session closed for user shin

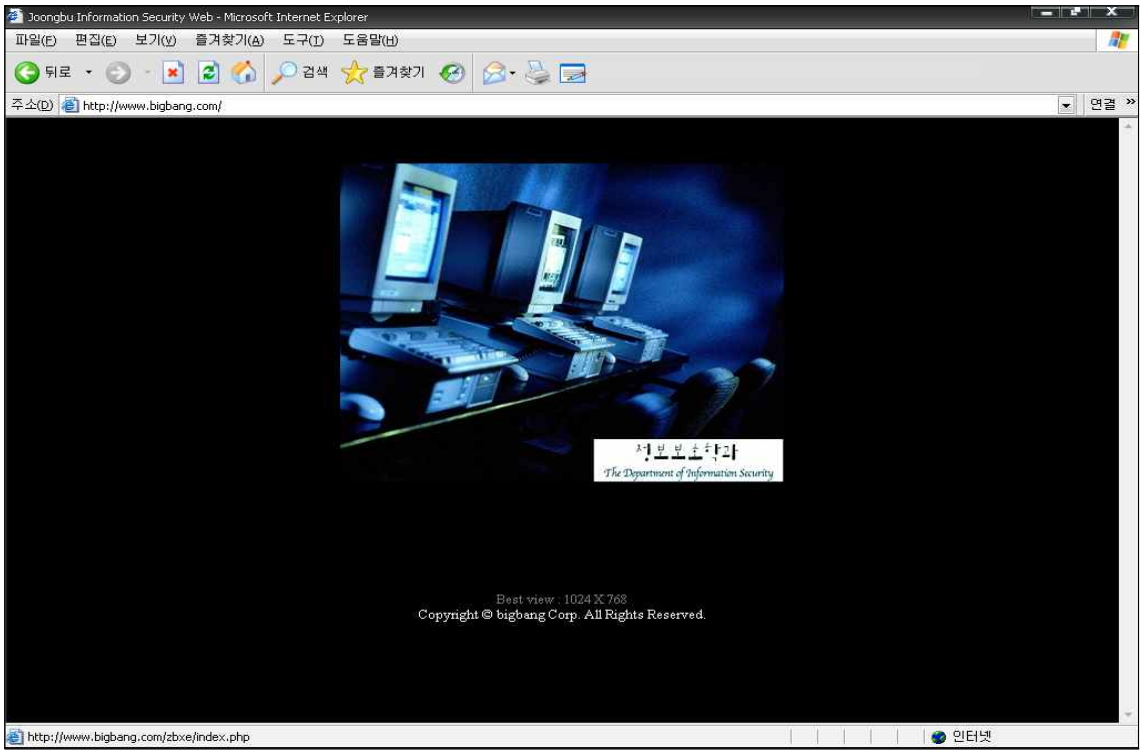
- Oct 30 16:29:00 : 현재의 연월일 시간을 표기
- localhost : 서버의 호스트이름
- vsftpd(pam_unix)[12999] : 서버에 설치된 vsftpd 데몬
- session opened for user shin by (uid=0) : 사용자ID가 shin이라는 사용자가 접속 끊음

4.2 IDS를 이용한 접속 기록

외부나 내부에서 들어오거나 나갈 때는 NAT서버를 통해서 이동함으로 NAT서버의 IDS에 모든 기록이 남게 된다. 관리자는 내부망의 접근을 IDS 기록을 확인하여서 관리 감독을 할수 있다.

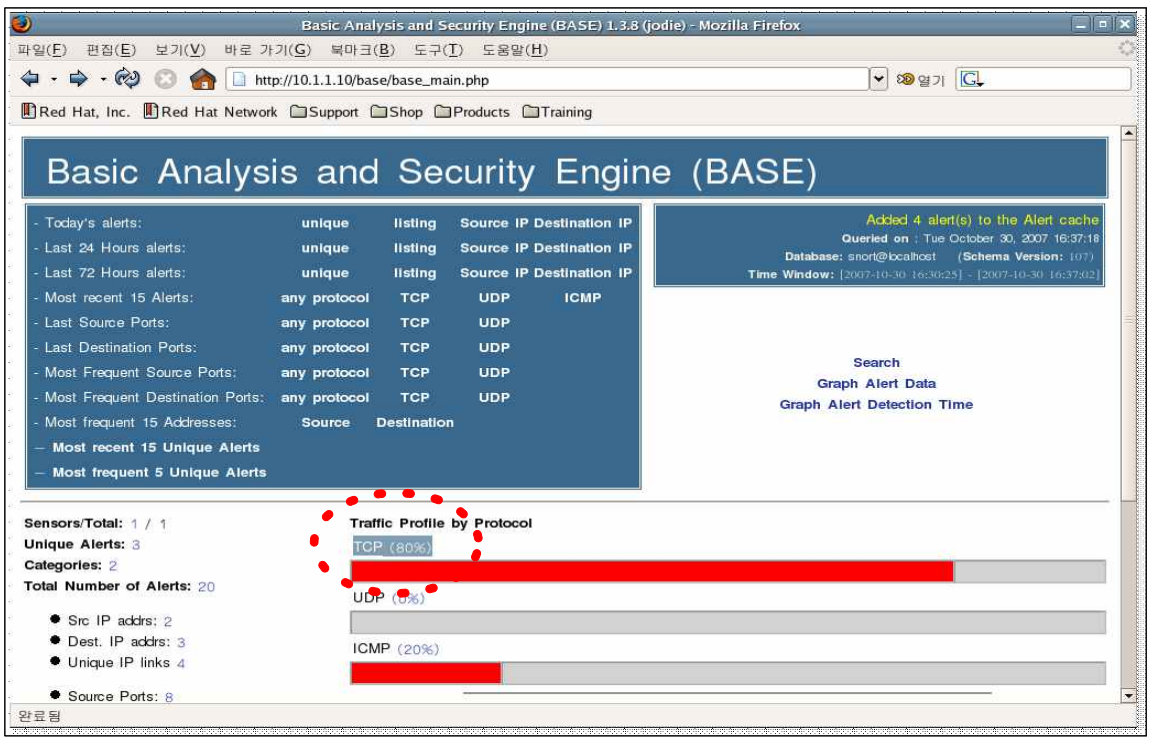
4.2.1 WEB접속시 IDS 접속 기록

주소 입력창에 Web주소인 http://www.bigbang.com를 입력하게 되면 Web서버의 접속이 된다. 이때 Web서버로의 접속 기록을 IDS로 확인할수 있다.



<그림 4-2-1>

IDS base 메인페이지에서 보면 TCP에 기록이 남게 된다.



<그림 4-2-2>

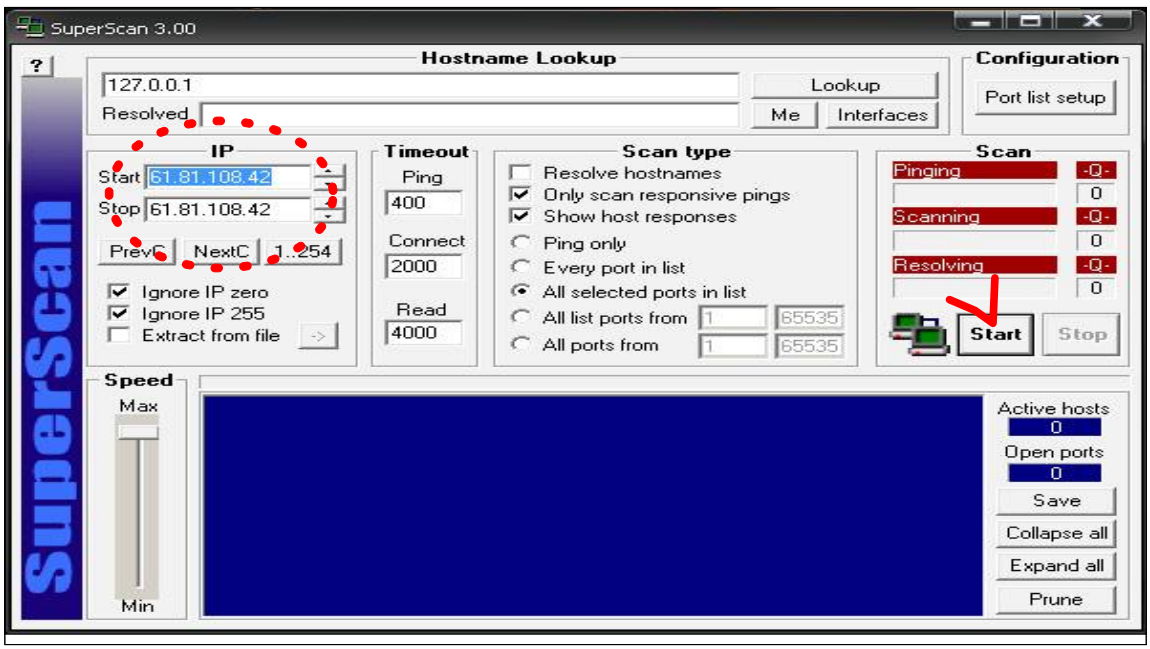
아래 그림에서 TCP에 대한 접속 기록을 확인결과
출발지 주소 = 61.81.108.124:3194
목적지 주소 = 61.81.108.42:80
IP가 61.81.108.124, 포트번호가 3194인 PC에서 61.81.108.42(NAT 서버)로 접속을 하였고, 여기서 포트번호 80으로 들어 왔다는 것은 Web으로 접근을 하였다는 것이다.

ID	Signature	Timestamp	Source Address	Dest. Address	Layer 4 Proto
#0-(6-32)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:08	61.81.108.124:3194	61.81.108.42:80	TCP
#1-(6-31)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:08	61.81.108.124:3194	61.81.108.42:80	TCP
#2-(6-30)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:08	61.81.108.124:3192	61.81.108.42:80	TCP
#3-(6-29)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:08	61.81.108.124:3192	61.81.108.42:80	TCP
#4-(6-25)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:07	61.81.108.124:3158	61.81.108.42:80	TCP
#5-(6-26)	[local] [snort] WEB-CGI calendar access	2007-10-30 16:37:07	61.81.108.124:3158	61.81.108.42:80	TCP

<그림 4-2-3>

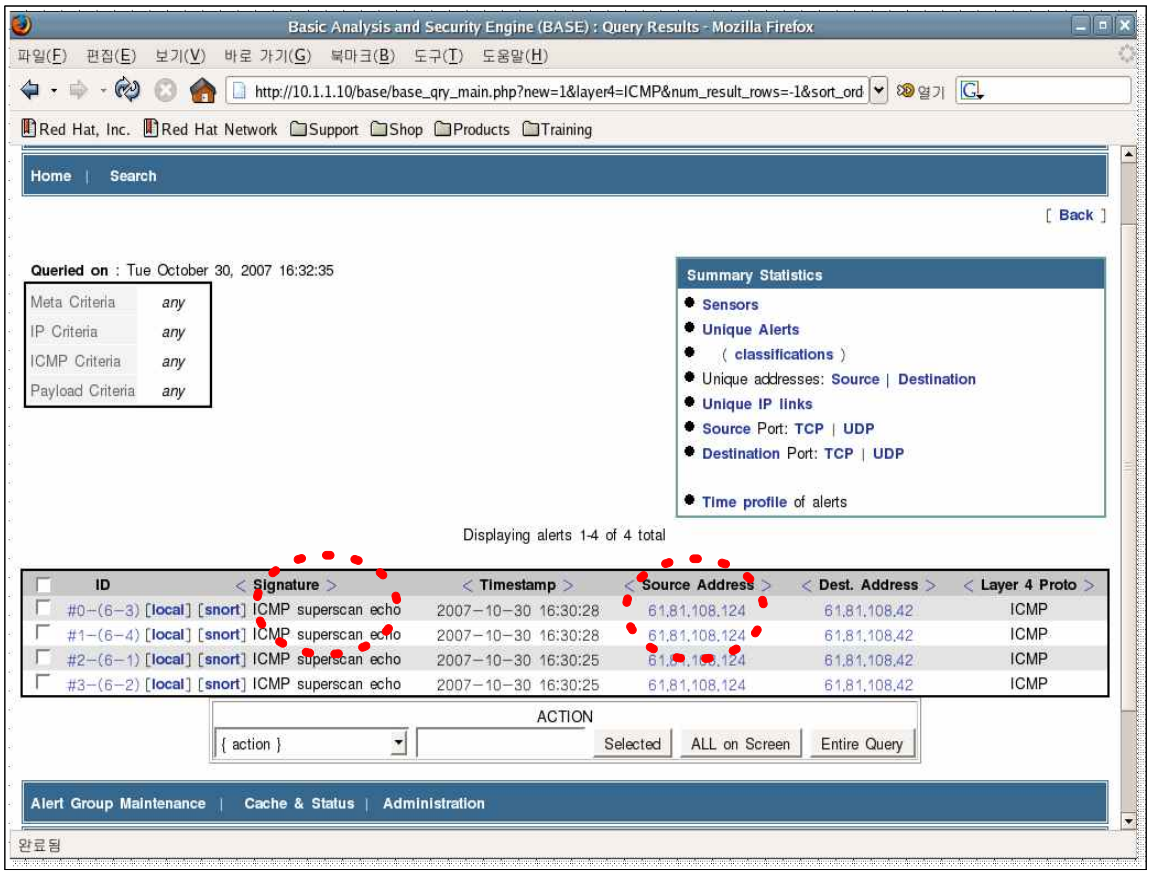
4.2.2 portscan시 IDS 접근 기록

외부 PC에서 NAT서버의 포트번호를 확인 하기 위해서 portscan프로그램을 사용한다.
IP 주소 입력창에 대상 IP를 입력한다.
즉, NAT서버의 IP주소를 입력 하고 시작 버튼을 누르게 되면 portscan이 시작되지만 NAT서버의 방화벽으로 인해 포트번호를 확인 할 수가 없게 된다.



<그림 4-2-4>

아래 그림과 같이 NAT서버에서는 IDS를 이용한 portscan 시도를 알수 있다.
superscan이라는 메시지가 남고 61.81.108.124에서 시도했다는 기록이 남게 된다.



<그림 4-2-5>

만약 위와 같은 포토스캔 뿐만 아니라 다른 악의적 목적으로 접근시에는 IDS의 접근 기록 확인함으로써 관리자는 그 특정 IP 주소에 대한 접근거부 및 보안을 할 수 있다.

5. 결 론

현재 IP Internet에 있어서 두 가지 직면한 문제가 있다. IP의 고갈과 ROUTING에서의 스케일이 계속 커지고 있다는 점이다. IP의 고갈을 해결할 수 있는 방법은 IP주소를 재사용하는 것이다. 즉 도메인 안에 있는 매우 적은 수의 호스트만이 도메인 바깥과 통신을 하고, 다른 호스트들은 그 호스트를 이용해서 통신을 하는 것이다. 이와 같이 우리는 이번 졸업작품을 통하여 부족한 IP주소의 해결 방안으로 NAT망을 구축하여 가설 네트워크를 만들고 많은 컴퓨터들이 이 가설 네트워크 내부 망에서 네트워크를 사용할 수 있게 구현하였다. NAT는 외부 네트워크에 알려진 것과 다른 IP 주소를 사용하는 내부 네트워크에서, IP 주소를 변환하는 것이다.

여러 주소에 대한 저비용의 대안으로 한 네트워크상의 여러 컴퓨터들이 하나의 IP주소를 공유하도록 허용하는 기술이 NAT 기술이다. NAT는 저비용 인터넷 서비스의 장점과 여러 컴퓨터들에 대한 연결성의 장점을 혼합한다.

실제로 대부분의 기업내에서 가설 IP를 사용하여 네트워크망을 형성한다. 우리는 이번 연구에서 기업내에서 사용하는 네트워크망을 실제로 구현하기 위하여 네트워크망 내에 NAT서버를 하나 두어서 라우터의 기능을 대신 하여 라우터의 기능을 NAT서버가 대신하도록 함으로서 내부망의 원활한 라우팅이 가능하게 하였다.

내부에는 Web서버, FTP서버를 구현하여 내•외부망의 모든 사용자들이 사용할 수 있도록 구현 하였다. 네트워크망 내의 보안 설정은 모든 패킷들은 NAT 서버를 통하여 이동하기 때문에 실질적으로 서버보안강화를 위해 NAT서버 내에 iptables를 이용하여 라우팅설정 및 방화벽을 설정 하였다. NAT서버의 IDS 구현으로 계속적인 탐지를 함으로써 보다 내부 망에 대한 보안을 강화 하였다.

이런 과정에서 우리는 NAT 에 관해 연구를 하며 NAT의 문제점을 알수 있었다. NAT 는 두 개의 연결된 네트워크에서 서로 다른 IP주소변환처리 주소영역을 사용할 수 있도록 해주는 아주 유용한 주소변환 기술이다. 하지만 NAT는 IP레이어와 TCP레이어상에서 출발지, 목적지 주소를 일일이 설정해줘야 한다는 것이다. 이런 수동적인 방법은 여러 문제 대해서 일일이 수정해야한다는 불편한 점이 많아진다. 이 문제를 해결하기 위해 능동적인 방법이 필요 할 것이다.

참고문헌

※ 문 헌

리눅스 서버관리 실무 바이블 - Linux Server Admin Bible v2.0
(출판사 : SU슈퍼유저코리아 / 저자 : 대한민국서버관리자그룹)

리눅스 네트워크 관리자 가이드(개정판) - Linux network administrator's guide
(출판사 : 한빛미디어 / 저자 : 올라프 커치 , 장윤식 역)

Fedora 리눅스 네트워크 & 웹 서버 무작정 따라하기
(출판사 : 길벗 / 저자 : 신재훈)

리눅스 서버 보안관리 실무
(출판사 : 슈퍼유저코리아 / 저자 : 홍석범)

※ 사이트

제로보드 : http://www.zeroboard.com/

리눅스포털 : http://www.superuser.co.kr/

리눅스스쿨 : http://www.linuxschool.net/

KLDP | Open Source, Geek, IT : http://kldp.org

부 록

1. DNS 포워드 zone파일 문법 요약

- ;(세미콜론) : 주석을 의미한다.
- $TTL : Time To Live 의 약어로 www.bigbang.com 을 질의 해 갔을때, 질의해간 다른 네임서버가 해당 IP주소를 캐시에 저장하는 기간(기본은 초단위)
- @ : /etc/named.conf 에 정의된 bigbang.com을 의미함다. (bigbang.com 으로 고쳐써도됨)도메인 다음에 (.)주의
- IN : 클래스의 이름으로 인터넷을 의미
- SOA : Start Of Authority 약어로 권한의 시작을 뜻함
- NS : Name Sever의 약어로, 설정된 도메인의 네임서버 역할을 하는 컴퓨터를 지정
- MX : Mail Exchanger 의 약어로, 메일서버 컴퓨터를 설정
- A : 호스트 이름에 상응하는 IP 주소를 지정
- CNAME : 호스트 이름에 대한 별칭을 부여할 때 사용

2. vsftpd 설정

2.1. vsftpd.conf 파일에서 자주 사용하는 기타 옵션들

- anonymous_enable : 익명 사용자의 접속허가 설정
- local_enable : 로컬 사용자의 접속허가 설정
- write_enable : 로컬 사용자의 저장, 삭제, 디렉토리 생성같은 명령 수행 허가 설정(익명 사용자는 해당없음)
- anon_upload_enable : 익명사용자의 파일 업로드 허가 설정
- anon_mkdir_write_enable : 익명사용자의 디렉토리 생성 허가 설정
- dirlist_enable : 다운로드 허가 설정
- listen_port : FTP 서비스의 포트 번호 설정(기본21번)
- deny_file : 업로드를 금지할 파일을 지정 (예:hide_file{*.gif, *.jpg, *.avi})
- max_clients : FTP 서버의 초대 동시 접속자수 지정
- max_per_ip : 한 pc가 동시에 접속할수 있는 접속수 지정

2.2. FTP Client 명령어

- help.? : 도움말
- dir<폴더명> : 디렉토리의 리스트를 알아본다. ls 명령어와 동일
- put <file> : 로컬 시스템의 파일을 원격 FTP Sever 로 전송
- size <file> : 원격 FTP Sever에 있는 파일의 크기를 출력한다
- mkdir <폴더명> : 원격 FTP Sever의 새로운 디렉토리를 생성한다
- pwd : 현재위치를 출력한다
- status : 현재의 상태를 출력한다
- exit : 원격 FTP Sever를 빠져 나온다
- qite : 원격 FTP Sever를 빠져 나온다
- mput <file1 file2> : 로컬시스템의 여러개의 파일을 동시에 보낼때 사용
- cd <Dir> : Dir 디렉토리로 이동
- nlist <Dir> : 원격 FTP Sever의 DIR 디렉토리의 리스트를 출력한다
- rename <name1 name2> : 원격 FTP Sever의 파일이름을 name1에서 name2 로 변경한다
- chmod : 원격 FTP Sever의 퍼미션을 조정한다
- open <FTP host> : 원격 FTP Host로 접속한다
- rmdir :<DIR> :DIR 디렉토리를 삭제한다.
- delete <FILE> : 파일을 삭제한다
- mdelete < file1 file2> : 여러개의 파일을 동시에 삭제한다.

3 httpd 설정

3.1. 웹서버 시스템의 위치선정

파일 시스템	파일명	폴더 위치
설정파일 관련 폴더	*.conf	/etc/httpd/conf
실행파일 관련 폴더	httpd	/usr/sbin
웹서버 기본 폴더	-	/home/httpd
웹문서 저장 폴더	*.html	/home/httpd/html
CGI 스크립트 폴더	*.cgi, *.pl	/home/httpd/cgi-bin
아이콘 폴더	*.gif, *.jpg	/home/httpd/icons
로그파일 저장 폴더	*.log	/etc/httpd/logs

3.2. 웹서버 설치

○ 매우 중요한 3개의 conf 파일의 값을 설정한다.

httpd.conf : 웹서버 설정 관련 파일

srm.conf : 웹서버 자원 관련 파일

access.conf : 웹서버 보안 접근제어 관련 파일

○ httpd.conf

내 용	예약어	설정값
서버 실행방법	Servertype	standalone
포트번호	Port	80
Client machine name	HostnameLookups	off
보안을 위한 웹서버 실행자 변경	User, Group	nobody
서버관리자의 전자우편 주소	ServerAdimin	root@localhost
서버의 루트 디렉토리	ServerRoot	/etc/httpd
각종기록파일 위치지정 (서버의 루트디렉토리를 기준으로 한다)	Errorlog Transferlog Refererlog Agentlog	logs/error-log logs/access-log logs/referer-log logs/agent-log
서버의 프로세스 ID기록 파일	PidFile	/var/run/httpd.pid
서버 내부상황 보고	ScoreBoardFile	/var/run/apache-status
서버명(일반적으로 설정안함. DNS에 등록된 경우에 정의)	#ServerName	www.tiger.co.kr
Timeout 시간설정	TimeOut	400
서버 프로세스의 갯수설정	MinSpareServers MaxSpareServers StarServer	5 10 5
동시에 서비스 할 클라이언트의 최대 갯수	MaxClient	150
프로세스당 처리건수	MaxRequestsPerChild	30
프록시 서버 기능 여부 (프록시를 설정했을때 on)	#ProxyRequests	on

○ access.conf

access.conf : 전체적인 기본설정
.htaccess : 각 디렉토리 별로 자세히 결정

가. access.conf 설정

내 용	예약어 및 설정값
문서 디렉토리에 대한 설정	<Directory /home/httpd/html> Options Index Include ExecCGI AllowOverride None Order allow, deny Allow from all </Directory> (주) 보통 인트라넷인 경우에는 Order deny, allow Deny from all Allow kgi, admin, webmaster로 기술해 주어 접근제어에 중점을 둔다.
CGI 디렉토리에 대한 설정	<Directory /home/httpd/cgi-bin> AllowOverride none Options none </Directory>
인증 매카니즘 설정하기	<Directory /home/httpd/html> AuthType Basic AuthName Yoo hyun woo AuthUserFile /etc/httpd/conf/.htpasswd AuthGroupFile /etc/httpd/conf/.htgroup Order allow, deny Allow from all Require user admin Require group admin </Directory>

나. 암호파일 만들기

(1) passwd

#htpasswd -c /etc/httpd/conf/.htpasswd admin

(주) -c (create) : 맨처음 .htpasswd를 생성할 때만 사용, 두번째 등록시는 생략한다.

#htpasswd /etc/httpd/conf/.htpasswd webmaster

(2) group

vi 에디터를 사용해서 .htgroup 파일을 만든다.

.htgroup	admin:admin webmaster

<표 6-3-4>

4. Log files

파일명	파일의 역할
access_log	웹서버가 어떤 파일을 읽어 클라이언트에게 전달했는지 보여준다.
agent_log	사용자가 어떤 부라우저를 통해 웹서버에 접속했는지 그이름을 나열
error_log	에러상황 기록
referer_log	각 페이지들이 어떻게 무엇을 참조했는지 기록

<표 6-4>

5. 실시간 웹서버 상황보기

가. access.conf에 다음내용 추가

내 용	예약어 및 설정값
실시간 웹서버 상황보기 관련	<Location /status> SetHandler server-status Order deny, allow Deny from all allow from your_admin_host </Location>

<표 6-5>

나. 웹브라우저에서 보기

URL에 다음과 같이 넣고 실행한다.

your.server.name/status

your.server.name/status?refresh=n

(주) n : 갱신주기 초단위

6. 아파치 서버 기동 및 종료

가. 스크립트 작성 : /usr/sbin에 기동/종료를 쉽게 할 수 있는 스크립트를 작성하여 사용한다.

내 용	스크립트
w_start.sh	/usr/sbin/httpd -f /etc/httpd/conf/httpd.conf
w_stop.sh	kill -TERM `cat /var/run/httpd.pid
w_restart.sh	kill -HUP `cat /var/run/httpd.pid

<표 6-6>

start 웹서버

#. start.sh

위 스크립트 실행 후 프롬프트가 즉시 떨어진다면 아무 에러 없이 웹서버 데몬이 실행되었다는 것이다. 웹브라우저를 실행한후 URL 입력창에 localhost를 입력한 후에 잠시 기다리면 기다리고 기다리던 아파치 로고가 나올 것이다. 여기까지 되면 당신은 성공한 것이다. 서버가 작동하지 않았다면 logs/error_log를 보고 문제를 해결하길 바란다. 이제 부터는 여러분 몫이다. 성공하신 분은 계속되는 연구가 필요하며 실패하신 분은 게시판이나 메일을 이용하여 해결하는 수밖에 없다

stop 웹서버

#. stop.sh