2007년 정보보호 졸업작품 보고서

허니넷을 이용한 로그서버보안 및 침입대응방안

팀명 :IMPACT

팀원 :임연호

이성민

차주상

이승우

이재필

2007. 11

중부대학교 정보보호학과

목 차

<그림 차례>

요 약

최근 IT가 급속하게 발전함에 따라 웜이나 바이러스, 불법적인 해킹기술들이 날이 갈수록 다양해지고 점점 고도화 되어 가고 있다. 2004년 3970건이던 해킹 건수는 2005년 4549건, 지난해 4286건, 올해에는 9월까지만 5881건으로 계속 증가하는 추세다. 올해 해킹의 유형을 살펴보면 웜·바이러스 침투가 4929건이었으며 경유지 악용(582건), 홈페이지 변조(187건), 자료 유출(134건) 등이었다. 정부 포털, 범정부 민원센터, 지역 종합 민원실, 부처 웹사이트가 사이버 공격 대상의 80%를 차지했다. 반면 보안 관련 정부 기관 예산은 지난해 918억 원에서 올해 1010억 원으로 증가했다.

해킹이 이렇게 심각한 데도 많은 정부기관 홈페이지는 홈페이지 내에서 개인정보 유출을 방지하는 필터링 프로그램을 갖추지 않은 것으로 나타났다. 올해 4월까지 전체 29개 중앙행정기관 중 8개 기관, 480개 지방자치단체 중 73.5%인 353개 기관이 필터링 장치를 구축하지 않았다.

그래서 우리 조는 어떻게 하면 조금이나마 해킹의 위협에서 안전하게 시스템을 운영할 수 있을까 생각하고 고민하던 중 아직 우리에게 생소한 Honeynet이란 걸 접하게 되었다. 허니넷(Honeynet)이란 외부 공격을 유인해 현재 벌어지고 있는 해킹 상황을 확인할 수 있도록 구성한 가상 네트워크이다. 마치 꿀로 벌을 유인하는 것과 같이 해커를 가상의 네트워크로 유인해 최신 해킹 경향을 파악할 수 있도록 하는 것이 목적이다. 침입자를 잡기 위한 것이 아니라 그들의 움직임을 감시하고 배우기 위한 네트워크이다. 해커나 공격자들이 허니넷에 들어가면 모든 활동이 통제되고 모니터링 된다. 하지만 공격자들은 이런 사실을 인지하지 못한다. 이 네트워크는 기업의 서버 시스템과 유사하게 설치되며 수많은 가짜 File과 Directory, 진짜처럼 보일 수 있는 다른 정보를 저장하고 있다. 또 하나의 시스템이 아니라 여러 대로 구성되며 내외부의 트래픽이 모두 통제된다. 허니넷을 통해 수집된 정보를 분석하면 최신 사이버 공격 도구와 기술, 동기 등을 알 수 있다.

이를 바탕으로 침해사고대응센터에서는 사이버 공격에 대한 경보를 내린다. 또한 한국정보보호진흥원(KISA)에서는 개인정보를 유출하려는 다양한 공격 기법을 수집·분석할 수 있는 '개인정보 허니팟(HoneyPot)'을 구축해 07년 12월부터 가동할 예정이라고 하였고 인터넷침해사고대응지원센터(KISC)에서 운영하고 있는 허니넷(Honeynet) 네트워크를 사용하여 개인정보침해공격과 웹서버 해킹 등을 검출하여 공격자의 정보 수집을 상세히 분석할 것이라 하였다. 이렇듯 개인정보보호와 웜·바이러스의 침입대응에 만전을 기하고자 우리조원들은 Honeynet 개념을 도입하여 학과와 학교를 지키고 나아가 국가를 지킬 수 있는 환경조성에 앞장서도록 하고자 하였다.

1. 서론

1.1 Honeypot과 Honeynet

1990년에 처음 비춰진 허니팟(Honeypot)은 꿀 딴지라는 호기심을 가질 만한 자원을 이용하여 더 많은 사용자를 끌어들이는 것처럼 시스템에 취약한 부분이 보인다면 더 많은 공격자를 끌어 들 일 수 있다는 말이다. 자칫 비정상적인 사용자를 유인하는 것으로 오해 할 수 있다 하지만 허니팟(Honeypot)의 원래 목적과는 다르다고 말 할 수 있다.

허니팟(Honeypot)의 경우 다양한 목적과 다양한 방식으로 구현이 가능하다. 예를 들면 윈도우 NT계열 서버 안의 리눅스 혹은 윈도우 서버가 이에 해당이 된다. 이러한 허니팟(Honeypot)은 네트워크의 시스템 등에 침입을 당해 피해를 받는 중 비정상적인 사용자의 행동, 공격기법을 분석하여 인터넷상에 존재하는 보안 위협을 줄일 수 있다.

그리고 하나의 시스템이 아닌 다수의 시스템으로 이루어진 네트워크 구조를 허니넷(Honeynet)이라고 하며 구성 형태로 보면 보안솔루션, 로그서버, 허니팟(Honeypot) 시스템으로 등으로 이루어져 있고 비정상적인 사용자는 보안솔루션 또는 허니팟 서버에 자유롭게 들어 갈 수 있도록 설정이 되어 있으며 인터넷에서 대두 될 만한 리스크를 줄이고자 하는 연구가 목적이며 이를 바탕으로 자료를 토대로 분석 하고 대응할 수 있는 기술 정보를 제공하며 정보수집 능력과 조기에 알려줄 수 있는 경보능력을 가지고 있어 보안의 시스템 및 응용프로그램에 적용 될 수 있다.

1.1.1 Honeypot

■ 장점

Honeypot은 하루 평균 1MB 정도의 정보를 수집한다. 반면에 IDS는 분당 1M 정도의 정보를 수집한다. 그리고 Honeypot을 공개서버에 설치하지 않으므로 일반적으로 트래픽이 거의 없는 시스템이므로 이러한 정보는 비정상적인 으로 접근한 행위에 의한 것이라 판단이 되며 아주 유용한 자료가 된다. 시스템 자원의 경우 방화벽이나 침임 탐지 시스템처럼 그 성능이 시스템 자원의 영향을 받지 않으며 Honeypot은 단지 자신에게 보내진 패킷만을 수집하기 때문에 데이터를 처리하기 위해 고성능의 시스템을 사용할 필요가 없고 저사양의 시스템으로도 운영이 가능 하며 쉽게 말하면 Honeypot의 개념은 단순하다고 말할 수 있다. 또한 설치 및 적용 그리고 운영이 용이하며 Honeypot은 공격을 당하는 즉시 그 가치를 발휘하게 되며 추적 및 대응이 가능 하다가 볼 수 있다.

■ 단점

Honeypot은 Honeynet의 기능에 비해 자신에게 오는 트래픽만을 감시할 수 있다는 것과 Honeypot은 Mirroring 포트에 설치되는 IDS와 달리 일반 포트에 설치되므로 자신에게 오는 패킷만을 수집 하므로, 네트워크 전반에 대한 침입정보를 분석 할 수 없다는 것과 공격자가 Honeypot 시스템임을 쉽게 노출 시킬 가능성이 있으며 공격자가 Honeypot 시스템을 발견한 후에는 이 시스템의 접근을 피하거나 마비시킬 수 있는 공격을 할 수도 있다.

1.1.2 Honeynet

■ Honeynet의 필요성

세계적으로 사이버 범죄는 갈수록 지능·고도화 돼가고 있고 실제 인터넷 환경에서 유포되는 웜·바이러스와 인터넷에 관련된 데이터를 수집해야 할 목적으로 가상의 서비스를 운용하고 해커 및 웜 바이러스의 접근을 유도함으로써 효과적인 분석체계를 갖춰야 필요가 있다. 이는 지피지기백전불태[知彼知己百戰不殆]라는 말과 같이 상대를 알고 나를 알면 백 번 싸워도 위태롭지 않다는 뜻으로 상대편과 나의 약점과 강점을 충분히 알고 승산이 있을 때 싸움에 임하면이길 수 있다는 말로써 현재 인터넷상의 호스트를 공격하는 많은 사용자들의 침입행위를 수집 하고 분석 하여 후에 같은 방법으로 공격하려 한다면 관리자 또는 공격자에게 경보 또는 경고를 하여 네트워크의 전반적인 리스크를 줄여 보고자 하는 차원에서 이러한 Honeynet 구축 및 운영을 제시하고자 한다. 또한 무작정 공격자를 기다리는 것이 아니라 Real Target 을 모방하여 유명사이트 또는 대학 병원등 공격자의 흥미를 유발하는데 목적이 있다. 하지만 위에의 글을 읽고 오해에 소지가 있다고 생각하겠지만 Honeynet은 다수의 Honeypot으로 구성된 네트워크로 해커들의 행동과 방법을 파악하여 다양한 해킹에 대처하기 위한 시스템이라는 것을 전제하여야 한다. 실제로 Honeynet은 국제 사이버범죄에 한 몫을 기여하는 등 다각적인 노력을 기울이고 있다.

■ 장점

Honeynet은 일반적으로 일반 호스트 안에 여러 대의 시스템을 이루고 있으며 그 구성요소로는 보안 솔루션 과 같은 Firewall 또는 침임 탐지 시스템으로 구성이 되며 어느 시스템에 적용을 하여도 손색이 없으며 Honeynet을 이용하여 뛰어난 데이터 수집 능력을 할 뿐만 아니라 비정상적인 사용자의 침입 시 경고 및 관리자에게 경보 역할을 하여 시스템의 전반적인 리스크를 줄일 수 있고 다양한 시스템과 응용 프로그램에 적용을 할 수 있다.

■ 단점

Honeynet 시스템을 접하게 되면 우선 Honeynet의 설정 및 구축의 복잡성에 많은 부분에 공감이 될 것이며 구축을 하고 운영을 하는데 있어서 전문적인 인력이 필요하고 Honeynet의 시스템에서 공격자와의 상호작용에 의한 보안에 위험성이 노출 되어 있다고 볼 수 있다.

1.2 본 과제의 목표 및 내용

1.2.1 목표

Honeynet 구축환경 요소 중 보안솔루션인 Firewall Service, IDS, LogServer, Honeypot Server 등을 차례로 구현한 후 각종 웜·바이러스 해커들의 침입패턴과 기술을 분석하고 IDS 로그 분석을 통해 Honeynet을 통한 해킹시도 및 웜·바이러스의 조기 경보와 로그서버를 안전하게 보호하는 방법을 제시한다.

1.2.2. 추진방법

Linux 기반으로 Firewall, IDS, LogAnalysis Server를 구축하고 Windows 기반으로는 Honeypot을 구축하여 현재 보안이 취약한 홈페이지를 대상으로 비정상적인사용자와 각종 웜·바이러스 등의 자료를 수집하여 침입패턴과 기술을 분석하여 그에 맞는 침임 대응 방안을 모색한다.

2. 주요 기반 기술

2.1 Firewall

방화벽의 원리는 허가된 사용자 외에는 접근 자체를 차단하는 것이다. 방화벽이 현재까지의 보안 대책으로서 가장 효과적인 이유는, 다양한 컴퓨터 시스템들이 각기 다른 운영체제에서 실행되고, 각 시스템이 안고 있는 보안의 문제점도 서로 다르기 때문에 호스트 컴퓨터마다 일정한 수준의 보안 능력을 부여하기 어렵기 때문이다.

2.1.1 방화벽의 종류

가. 패킷 필터링 방식

접속제어방식

- OSI 7 layer 중 네트워크층(IP) 과 전송 층(TCP)에서 접속제어
- 패킷 필터링을 통해 Source IP Addr, Destination IP Addr, TCP Port, TCP Sync Bit 필터링

■ 장점

- 처리속도가 빠르다
- 사용자에게 투명성을 제고하고 서비스에 대한 유연성이 좋다

■ 단점

- TCP/IP 패킷 헤더의 조작이 쉬워 변조 시 차단불가
- 전송 데이터에 대한 분석이 불가
- 접속제어 수량 및 방식에 따라 방화벽에 부하 가중

나. 어플리케이션 프록시 방식

접속제어방식

- OSI 7 Layer의 어플리케이션 계층에 방화벽 기능이 들어있다.
- 어플리케이션 게이트웨이가 서비스별 Proxy를 이용하여 접근제어

■ 장점

- 망사이의 연결이 Proxy를 통해서만 가능하므로 경계선 방어가 우수하고 IP 주소의 노출이 없다.
- Logging 및 Audit기능 제공 / 인증기능 우수

■ 단점

- 서비스별 Proxy 데몬 사용으로 서비스에 유연성이 적다.

2.1.2 Bridge Firewall

Bridge Firewall 은 이런 개념으로 컴퓨터에 랜카드를 2개를 꼽아서 랜카드1 로 들어와서 랜카드2 로 나가는 방식을 지원한다. 리눅스에서는 커널레벨에서 Bridge 를 지원한다.

허브가 ip 가없듯 이 Bridge Firewall 역시 ip 가없다. (단, 관리 목적으로 IP를 가지기도 한다.) Bridge 는 투명하다. traceroute 로 해도 나타나지 않는다. 마치 허브처럼 연결을 해주며 이러한 중간에서 패킷을 제어할 수 있는 것이 Bridge Firewall 인 것이다.

일반적인 방화벽은 Router 형식으로 동작하는 것이 보통이다. Firewall을 설치하고자 하는 Network에서는 Network을 재구성하는 불편함이 있을 수도 있다. 하지만 Birdge Firewall은 일반적인 hub처럼 Network에 아무런 설정이나 변화 없이도 설치가 가능하다. 또한 Bridge Firewall은 Routing을 하지 않으므로 처리속도 면과 안정성 면에서도 일반 Firewall에 비해 뛰어나다고 할 수 있다.

2.1.3 iptables

iptables은 TABLE, CHAIN, TARGET의 요소를 가지고 있다.

가. TABLE 분석

- mangle, nat , filter 3개의 테이블이 있으며, 테이블을 명시하지 않고 사용할 경우에는 filter가 기본 값이 된다.
- 3개의 테이블은 고유한 특성을 가지고 있으며, 정리하면 다음과 같다.

* mangle : 패킷이 맨 처음 들어왔을 경우에 제어가 가능하며, 패킷의 차단과 허용을 포함하고 라우팅 전, 후에 규칙을 적용할 수 있다.
* nat : 패킷의 헤더를 검사하여 소스와 목적지의 아이피 변환을 목적으로 한다.
* filter : mangle과 비슷하나, nat로 나가는 패킷을 제외한 것의 차단과 허용을 목적으로 한다.

나. CHAIN 분석

- 각 테이블마다 체인이 구성되어 있으며, 기본적으로 INPUT, FORWARD, OUTPUT이 있고, nat, mangle에는 PREROUTING, POSTROUTING의 체인이 추가되어 있다.
- iptables -L명령어로 체인 리스트를 확인할 수 있으며, 괄호 안에 기본 정책이 명시 되어 있다.
- 각 체인은 한 줄로 한 가지씩 규칙을 가지고 있으며, 규칙은 무조건적으로 정할 수 있는 것이 아니라 각 체인의 특성에 따라 정할 수 있다.
- 체인의 흐름을 보면
  PREROUTING -> INPUT(FORWARD) -> OUTPUT -> POSTROUTING의 순서로 패킷이 이동하면서 규칙에 적용된다.
- FORWARD 체인을 중심으로 왼쪽은 들어오는 패킷을 제어할 수 있으며, 오른쪽은 나가는 패킷을 제어한다. 만약에 PREROUTING 체인에서 나가는 패킷을 제어하고자 규칙을 작성한다고 해도 규칙이 적용되지 않는다.
  또한, 체인은 -N옵션을 이용하여 새로 만들고 -X옵션을 이용하여 삭제할 수 있다.

다. TARGET 분석

- 각체인의 규칙은 target을 어떻게 정하는 가가 기본이며, 각 타깃의 특성을 보면

ACCEPT : 패킷을 허용한다.
DROP : 패킷을 차단한다.
REJECT : 패킷을 거부한다.
RETURN : 패킷을 맨 아래 규칙으로 내린다.
MARK : 패킷에 마크를 표시한다.
LOG : 로그를 남긴다.(/var/log/messages)
MASQUERADE : 패킷의 출발지를 나가는 장치의 아이피로 바꾼다.
SNAT : 패킷의 출발지를 지정한 아이피로 바꾼다.
DNAT : 패킷의 도착지를 지정한 아이피로 바꾼다.

- 타겟은 사용할 수 있는 테이블이 다르며, 테이블에 맞게 사용하여야 한다.

- 기타
prot : ip 프로토콜(-p) tcp, udp
opt : 장치 (-o 나가는 장치, -i 들어오는 장치) eth0, ppp0
source : 출발지 주소 (-s)
destination : 도착지 주소 (-d)

2.2 IDS

네트워크 시스템에서 정당한 사용 권한이 없는 사용자의 침입을 감시하고 강제로 접속을 끊는 등 필요한 조치를 취하는 시스템을 의미한다.
침입 탐지 시스템은 보통 방화벽과 함께 운용되는데 방화벽은 외부의 침입을 감지하는 시스템이고, 침입 탐지 시스템은 방화벽 내부에서 침입자의 행동을 감시하는 시스템이다.
침입 탐지 시스템은 방화벽으로 막을 수 없는 CGI를 이용한 공격이나 버퍼오버플로우(Buffer Over Flow)도 탐지할 수 있을 뿐만 아니라 LAN 내부의 구성원이 저지를 수 있는 부정한 행위까지 감시할 수 있다.

침입 탐지 시스템(IDS)은 침입 탐지를 위한 기준 데이터에 따라 네트워크 기반 IDS (Network Based IDS), 호스트 기반 IDS (Host Based IDS), 하이브리드 IDS (Hybrid IDS)로 분류한다. 또한 IDS는 탐지 방식에 따라 비정상행위 탐지 방식 IDS (Anomaly IDS)과 오용 탐지 방식 IDS (Misuse IDS)으로 구분하고, 대응 방식에 따라 액티브 IDS (Active IDS)와 패시브 IDS (Passive IDS)로 구분한다.

네트워크 기반 IDS는 네트워크 패킷을 감시하는 방식으로 하나의 IDS를 이용하여 내부와 외부를 모두 감시할 수 있지만 ID를 도용한 사용자를 탐지할 수 없다는 단점이 있다.
호스트 기반 IDS는 로그 파일과 감사(Audit) 자료를 이용하여 침입을 탐지하는 방식으로 ID를 도용한 사용자까지도 탐지할 수 있지만 시스템 자원을 많이 사용한다.
그래서 보통 침입 탐지 시스템은 호스트 기반과 네트워크 기반 IDS를 복합한 하이브리드 기반의 IDS를 사용한다.

오용 탐지 방식 IDS(Misuse IDS)는 사용자마다 미리 정의된 데이터 사용 규칙을 만들어 놓고, 이 범위를 벗어나는 경우 침입한 것으로 판단하는 것이다. 오용 탐지 방식 IDS는 오판 율이 낮은 대신 새로운 공격 유형에 대해서는 대응하지 못하는 단점이 있다.

비정상행위 탐지 방식 IDS(Anomaly IDS)는 로그 파일과 사용자의 활동, 시스템 호출 등을 감시하여 침입을 탐지하는 방식으로 새로운 공격 유형을 탐지할 수 있지만 오판 율이 높다는 것이 단점이다. 실제로 미 공군에 대한 20,000건의 침입 탐지 사건 중 실제 침입은 2건에 불과했다는 보고가 있다.

액티브 IDS(Active IDS)는 침입자의 세션을 강제로 종료하고 이후 접속하지 못하도록 차단하는 방식으로 방화벽과 함께 동작한다. 패시브 IDS(Passive IDS)는 침입자가 있다는 것을 메신저나 메일을 통해 알려주는 방식이다.

2.2.1 Libpcap

그림 2-1 Libpcap

Snort 가 패킷을 스니핑 하려면 패킷캡처 library인 libpcap을 기반으로 동작하므로 snort를 설치하기 전에 먼저 설치해야 한다.

Libpcap이란 "Portable Packet Capturing Library"의 줄임말이며, 간단하게 패킷을

Capture하기 위한 함수 라이브러리이다. 패킷을 Capture하기 위한 도구로는 BPF(Berkeley Packet Filter), DLPI, NIT, SNOOP, SNIT, SOCK_PACKET, LSF(Linux Socket Filter), drain등 각 운영체제별로 다양한 도구가 있지만 각각의 사용자가 취향에 맞는 운영체제별로 패킷을 Capture하기 위한 코드를 별도로 구성해야 한다면 많은 어려움이 있을 것이다 그래서 이러한 도구들을 수용하는 Portable한 API가 있는데 이것이 바로 libpcap이다. libpcap 를 이용한 가장 대표적인 프로그램이 tcpdump 와 SAINT 와 같은 프로그램들이 있다..

2.2.1.1 Libpcap Setting

```
소스 경로로 이동
cd /var/tmp/libpcap-0.9.8

config 설정
./configure ₩
--prefix=/usr ₩
--bindir=/usr/bin ₩
--sbindir=/usr/sbin ₩
--libexecdir=/usr/libexec ₩
--sysconfdir=/etc/snort ₩
--localstatedir=/var ₩
--libdir=/usr/lib ₩
--includedir=/usr/include ₩
--mandir=/usr/share/man

컴파일
make

find /*  > /root/LIBPCAP1

컴파일 후 설치
make install

find /*  > /root/LIBPCAP2

diff  /root/LIBPCAP1  /root/LIBPCAP2  >
/root/Installed/libpcap-0.9.8-installed
```

2.2.2 pcre

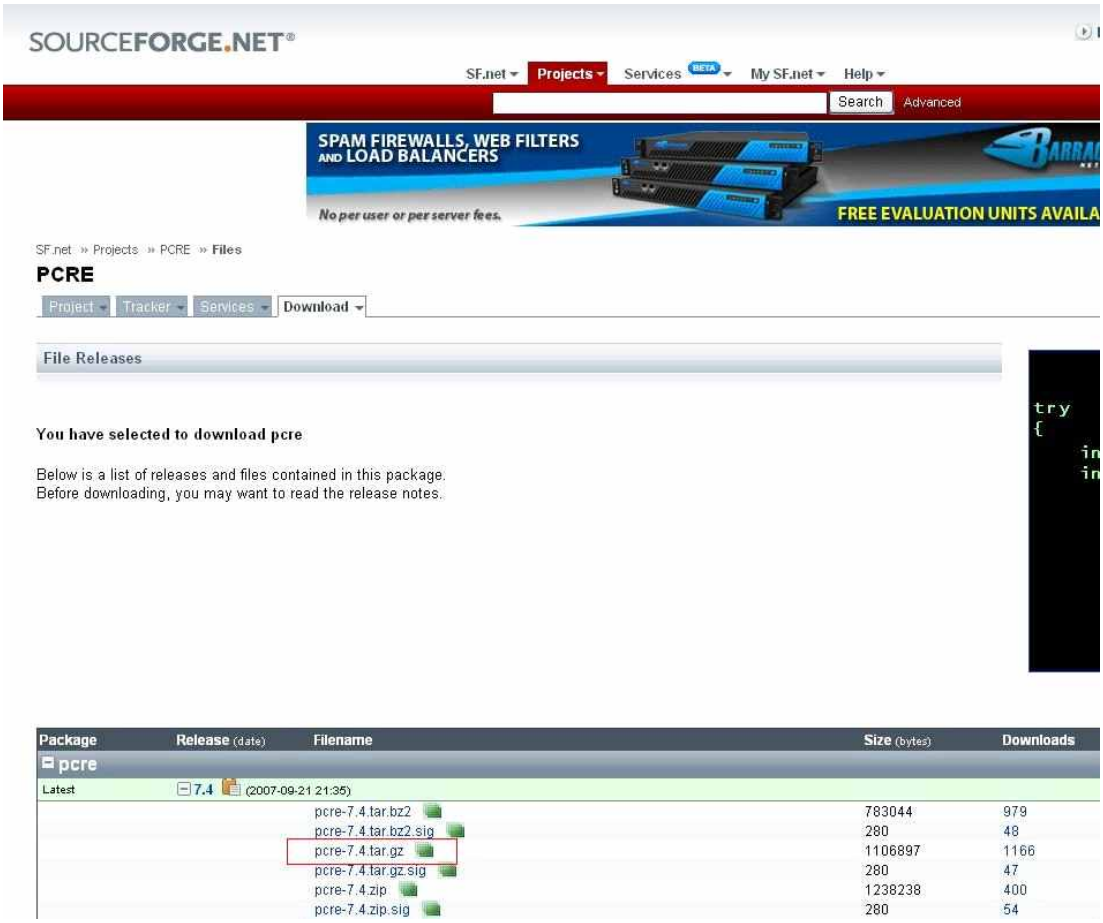


그림 2-2 pcre

```
[pcre install] pcre-7.4.tar.gz

# tar zxf pcre-7.4.tar.gz
# cd pcre-7.4
# ./configure && make && make install
# vi /etc/ld.so.conf
  /usr/local/lib 추가
# ldconfig -v
```

2.2.3 snort

2.2.3.1 Snort 란 무엇인가?

Snort는 “sniffer and more"라는 말에서 유래되었는데, 처음 공개되었을 때는 코드도 얼마 되지 않는 단순한 패킷 스니퍼 프로그램이었다. 그러나 이후 현재의 IDS와 같이 rule을 이용한 분석 기능이 추가되고, 커뮤니티를 통하여 계속적인 기능 보완과 향상을 통해 지금과 같이 다양한 기능과 탁월한 성능을 갖춘 프로그램이 되었다. snort는 공식 홈페이지인 http://www.snort.org를 통해 지속적인 업그레이드를 제공하고 있다.

그림 2-3 snort

snort의 기능은 간단히 다음과 같이 분류할 수 있다.

패킷 스니퍼(Sniffer)	네트워크의 패킷을 읽어 보여주는 기능
패킷 로거(Logger)	모니터링 한 패킷을 저장하고 로그에 남기는 기능
Network IDS	네트워크 트래픽을 분석하여 공격을 탐지하는 기능으로 buffer overflow나 port scan, IP scan 등 대부분의 공격을 탐지할 수 있다.

. snort는 오픈 소스로 개발 중인 패킷 캡처 라이브러리인 libpcap을 사용하여 패킷을 캡처하고, 수집된 패킷이 사전에 정의된 snort 공격 룰과 비교하여 만약 매칭 되었을 경우 syslog를 통해 로그를 남기거나 특정 디렉터리의 특정 파일 또는 database에 남기도록 할 수 있다.

2.2.3.2 Snort의 구조

snort프로그램은 몇 가지 구성 요소들이 플러그인 형태로 이루어져 있어 쉽게 각자의 환경에 따라 변경하고 수정할 수 있도록 되어 있는데, 기본적으로 다음과 같은4가지 구성요소로 이루어져 있다.

① 스니퍼(sniffer)
② preprocessor
③ 탐지엔진
④ 로깅(출력)

snort는 먼저 스니퍼를 통해 snort IDS를 통과하는 모든 패킷을 수집하게 된다. 여기에서 수집된 데이터는 바로 룰 기반의 탐지 엔진을 거치지 않고 그 전에 preprocessor를 통해 보다 효율적인 공격 탐지를 위해 HTTP 인코딩플러그인이나 포트스캔 등 몇 가지 플러그인을 먼저 거치면서 매칭이 되는지 확인하게 된다. 물론 preprocessor 역시 모듈화 되어 있어 각자의 환경에 불필요하다면 disable 할 수 있다. 이를테면 RPC 트래픽에 대해 탐지할 필요가 없다면 RPC 관련 preprocessor를 주석처리하면 된다.
그리고 preprocessor를 통과한 패킷은 snort IDS의 핵심이라 할 수 있는 룰 기반의 탐지엔진을 거치면서 사전에 정의된 탐지 룰과 매칭 되는지 확인하게 된다. 만약 룰에 매칭되었을 경우에는 사전에 정의된 정책에 따라 로그에 남게 되고, 그렇지 않은 경우 통과를 하게 된다.

2.2.4 Mysql

2.2.4.1 MySQL 이란 무엇인가?

MySQL 은 진정한 멀티유저, 멀티 쓰레드 SQL(Structured Query Language)데이터베이스 서버입니다. SQL은 가장 널리 쓰이는 DB질의어 이구요. MySQL 은 DB서버 데몬인 mysqld 와 여러 가지 이용자 프로그램 그리고 라이브러리로 구성되어 있습니다.
MySQL의 주목표는 속도, 뛰어난 수행능력 그리고 사용의 편리함입니다. 애초에 MySQL은 TcX 내부에서 자체적으로 사용할 목적으로 만들어 졌습니다. 그 당시 어떤 제조업체가 제공하는 SQL 서버 보다 강력한 DB서버가 필요했기 때문이지요. TcX에서는 MySQL을 1996부터 사용해 왔는데 현재 700만 레코드 이상 되는 500개 이상의 테이블이 (100 기가 바이트 이상) 주요 업무에 사용되고 있다고 합니다.
MySQL은 높은 수행능력을 발휘하는 것을 목표로 수년 전 개발이 시작되었습니다. 여전히 개발이 진행 중이지만 매우 풍부하고 쓸 만한 함수들을 제공 합니다
(PHP3와 최적으로 연동됩니다)

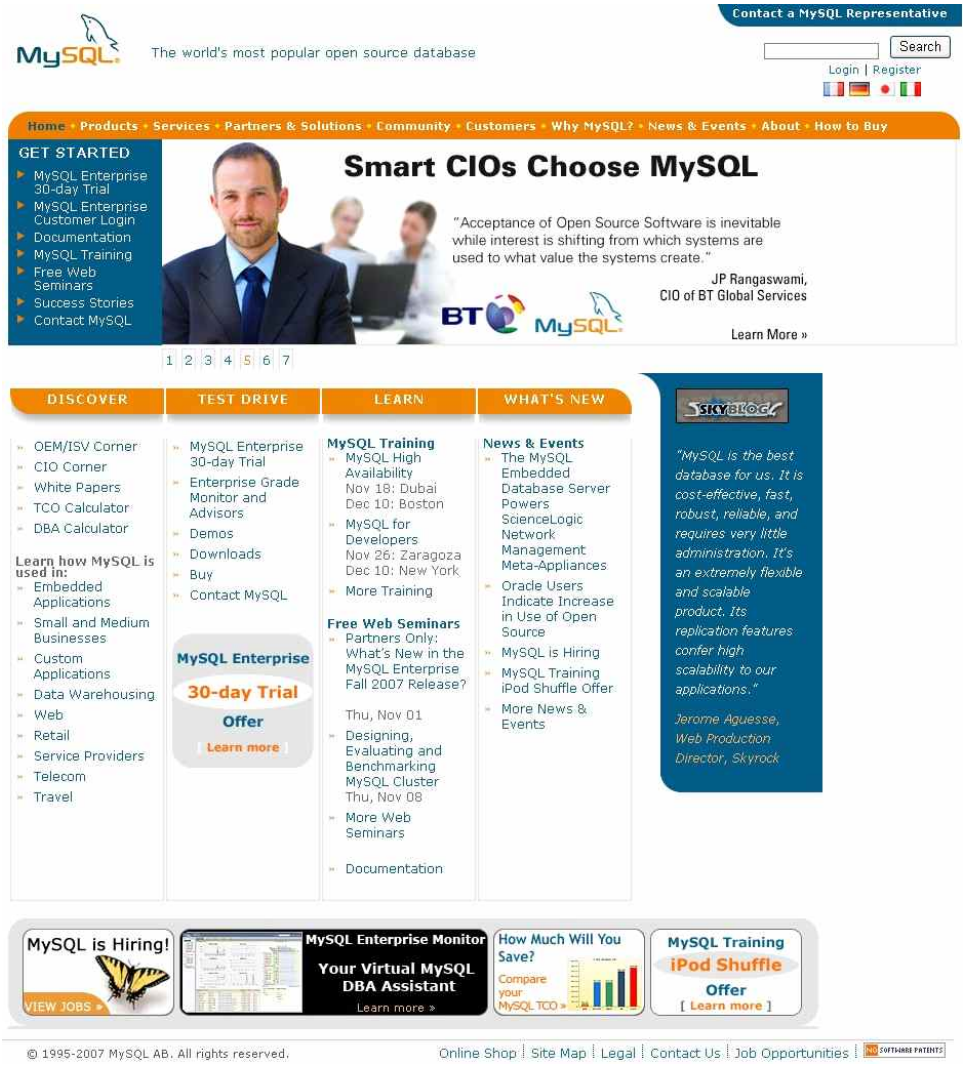


그림 2-4 Mysql

2.2.4.2 MySQL 주요 기능

- kernel threads를 이용, 완벽한 multi-threaded를 지원합니다. (복수 CPU 지원 가능)
- C, C++, Java, Perl, Python ,TCL 의 API를 지원합니다.
- 수많은 운영체제에서 안정적으로 동작합니다.
- 여러 가지 매우 편리한 column types을 지원합니다.
- signed/unsigned integers long, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, DATETIME, TIMESTAMP, YEAR, SET, ENUM 매우 빠르게 joins 을 수행합니다.

- 모든 연산자와 함수를 SELECT 와 WHERE 문에서 지원합니다.

```
mysql> SELECT CONCAT (first_name, " ", last_name) from tbl_name
            WHERE income/dependents > 10000 AND age > 30;
```

- SQL GROUP BY 와 ORDER BY 문을 완벽하게 지원 합니다.
- group functions (COUNT() , AVG() , STD() , SUM() , MAX() and MIN())등
- ANSI SQL 과 ODBC syntax에서 LEFT OUTER JOIN 을 지원 합니다.

- 서로 다른 DB 내의 테이블들의 Join 이 가능합니다.
- 사용자 권한(privilege)을 유연하게 관리 할 수 있습니다
- ODBC (Open-DataBase-Connectivity) for Windows95 (with source)를 지원합니다.
- 테이블 당 16 개의 인덱스 생성이 가능합니다. 각 인덱스는 1~ 15 의 칼럼으로 구성 되며 최대 256 bytes 크기 입니다

- 규모가 매우 큰 테이블을 사용할 수 있습니다.(50,000,000 records 이상)
- C 와 C++을 이용해 작성되었으며 많은 compilers에서 테스트 되었습니다.
- ISO-8859-1 Latin1 character set을 지원합니다.
- SQL92표준에 따라 tables 과 columns 에 별명(alias)을 사용할 수 있습니다.
- DELETE , INSERT , REPLACE , UPDATE 후 결과 행수를 리턴 합니다.
- 함수이름과 동일한 테이블 명, 컬럼 명을 사용 할 수 있습니다.
- 에러메시지의 다국어 지원이 가능합니다.
- TCP/IP socket을 이용 원격 MySQL서버에 접속할 수 있습니다.

2.2.5 Apache

아파치는 1995 년 그 당시에 가장 인기 있었던 웹 서버중의 하나인 NCSA HTTPD 1.3 버전 을 기반으로 탄생하였다. 그 후 기존의 NCSA 웹 서버에 더욱 향상된 기능들을 탑재하여 Apache 웹 서버를 발표하였으며, 현재는 인터넷 웹 서버 중에서 최고의 인기를 구가하고 있는 이른바 '잘 나가는' 소프트웨어 중의 하나이다. 그 이유를 들자면 지속적으로 패치파일을 제공하고 최고의 퍼포먼스를 내고 있기 때문이다. 물론 무료로 제공된다는 점과 많은 마켓 쉐어의 점유로 인하여 안정성을 인정받았다는 점도 한 이유가 된다. 그리고 Windows NT 4.0 과 95 용으로도 Beta 판이 나와 있으므로 Unix 환경이 갖추어 지지 않은 분들은 한번 사용해 보기 바란다. 아파치는 현재까지 1.3b3 까지 나와 있으며, 1.2 시리즈는 테스트 버전이 아닌 안정된 버전이다. 만약 여러분들이 1.2 베타나 구 버전의 아파치 웹 서버를 사용하고 있다면 안정성이나 Security 면에서 빠른 시일 안에 여러분들의 웹 서버를 업그레이드 또는 패치하기를 권하는 바이다.

2.2.6 PHP

2.2.6.1 PHP란 무엇인가?

PHP는 HTML 문서 내부에 포함 되어 웹서버에서 실행될 수 있는 스크립트 언어이다.

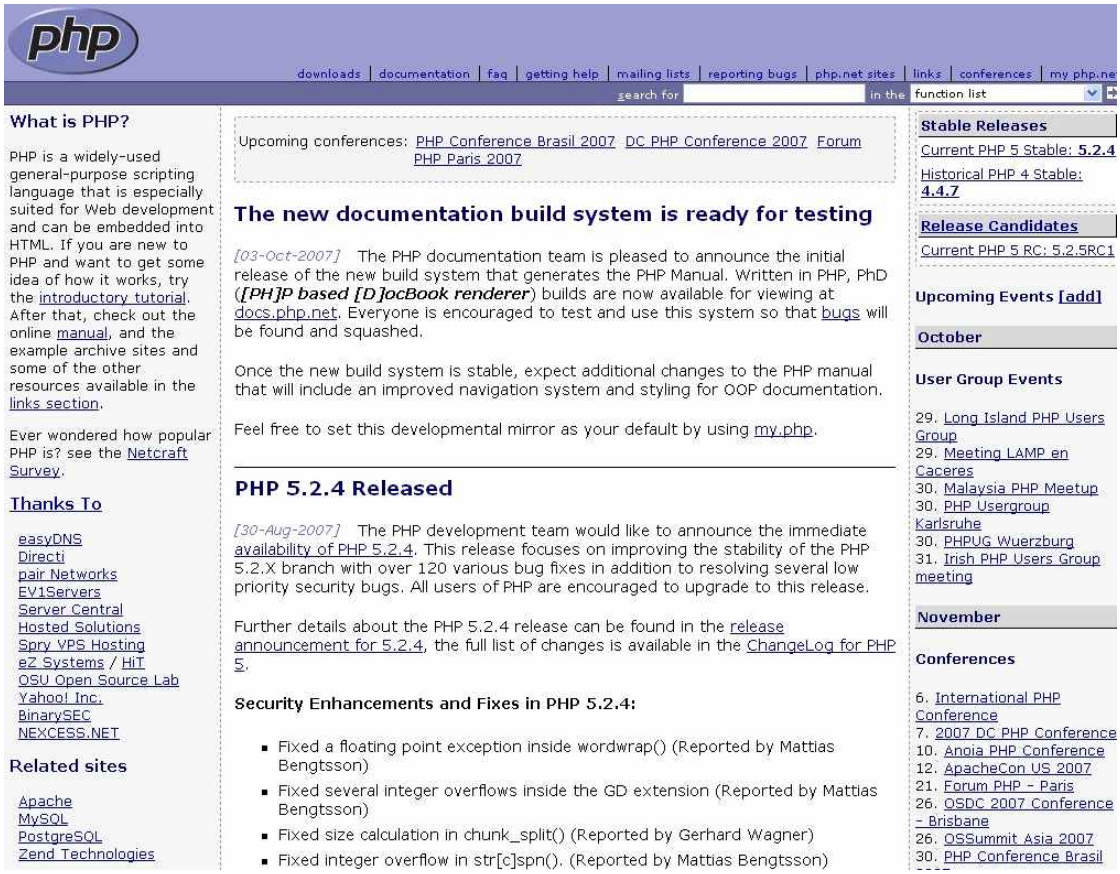


그림 2-5 PHP

간단한 예를 보면

```
<html><head><title>예제1- 1</title></head>
<body>
<?php
    echo "정보보호 IMPACT 팀입니다.";
?>
</body></html>
```

Perl이나 C와 같은 다른 언어와 다르게 HTML을 출력하는데 많은 명령어가 필요 없다. 여러분은 HTML내에 여러분이 하고자 하는 일(위의 경우는 "정보보호 IMPACT 팀입니다." 문자열 출력)에 대한 스크립트를 적어주면 된다. PHP 코드는 특정한 시작/끝('<?php' 와 '?>') 태그 사이에 들어가게 되는데 이 태그는 "PHP 모드"로 들어가거나 나오게 하는 것이다. PHP가 자바스크립트(Java Script)와 같은 클라이언트 측(client side) 스크립트 언어와 구별 되는 가장 큰 특징은 이 코드가 서버에서 실행된다는 것이다.

2.2.6.2 PHP는 무엇을 할 수 있는가?

PHP는 CGI 프로그램에서 할 수 있는 모든 것을 할 수 있다. HTML 폼을 통해 데이터를 가져 오고, 동적인 웹페이지를 만들거나, 쿠키(Cookie)를 보내고 받을 수 도 있고 기존의 C 나 Perl 을 이용한 CGI에서 구현하기 힘들었던 동적인 GIF이미지 의 생성 , HTTP 인증 , 파일업로드 등의 기능도 쉽게 구현 할 수 있다.

아마도 PHP의 가장 강력하고 강력한 부분은 데이터베이스(Database)와의 연동부분일 것이다. PHP를 사용하면 여러분은 데이터베이스를 사용한 동적인 웹 페이지를 간단하게 만들 수 있다.다음에 나오는 DB server들을 현재 사용할 수 있다. :

Adabas D, InterBase, Solid, dBase, mSQL, Sybase, Empress, MySQL,
Velocis, FilePro, Oracle, Unix dbm, Informix, PostgreSQL

PHP는 IMAP나 SNMP, NNTP, POP3, HTTP등의 프로토콜들을 사용해 서 다른 서비스들에 접근하여 데이터를 교환할 수 있다. 심지어는 raw network 소켓을 사용하면, 그 외의 프로토콜들을 사용할 수도 있다.

PHP의 특징을 간단하게 요약하면 다음과 같다.

- 서버에서 해석되는 스크립트 언어이다.
- 데이터베이스 연결을 쉽게 해준다.
- 코드 작성이 쉽다.
- Unix와 Windows 환경 모두에서 사용 가능하다.

2.3 Log Server/Analysis Server

원격 로그서버는 다른 시스템들의 log들을 저장할 하드 드라이브 공간을 제공하도록 미리 설정되어진 시스템일 뿐 그 이상도 이하도 아니다. 이 시스템은 완벽한 보안으로 차단되어 있어야 하며 모든 RPC 데몬 들이나 다른 기타 서비스들도 암호화되지 않고서는 절대 접근이 허락되지 않는다. 데이터들은 오직 UDP/Port 514 번을 통해서만 전송이 허락된다.

O 로그분석이란

웹사이트 방문자의 분석을 통하여 방문자수와 페이지 뷰수, 쿠키값 분 석등을 통한 방문자 정보 분 석등을 통하여 사이트의 현 상황을 면밀히 분석하는 것을 뜻한다.

O 로그파일을 이용한 분석

일반적으로 자체 웹서버를 가지고 있는 경우에 해당하며 IIS 의 경우 ex00000.log 파일이 생성이 되며 Apache 의 경우 access.log 파일이 생성이 된다.
각 웹서버를 통하여 생성된 로그파일에는 사용자의 IP와 사용한 페이지, 사용시각, 쿠키값들이 저장되게 된다. 이렇게 생성된 로그파일의 경우 굉장히 방대한 분량의 데이터가 생성이 되며 이 로그파일들을 직접 분석하기는 사실 거의 불가능하며 로그분석 솔루션을 이용하

여 로그파일을 분석해 주어야 한다.

O 로그 분석 툴
web trends 와 wise log enterprise 가 있으며 각 솔루션의 가격은 web trends 의 경우 약 1천만원 선이며 wise logenterprise 의 경우 약 500만원정도의 가격대를 형성하고 있다.

각 로그분석툴에서 추출해 낼 수 있는 일반적인 정보는 아래와 같다.
- 총 페이지 뷰수, 일평균 페이지 뷰수, 기본 페이지 뷰수, 방문당 페이지뷰수
- 총 히트수, 일평균 히트수, 방문당 히트수
- 방문수, 일평균 방문수, 순 방문자, 일평균 방문자, 1회 방문자, 2회이상 방문자, 평균 이용시간등의 정보들을 추출해 낼수 있다.

2.4 Honeynet/Honeypot

2.4.1 Honeynet의 종류

2.4.1.1 Generation 1

1999년도에 처음 구축이 되었으며, Layer 3 인 네트워크 계층인 방화벽에 의한 접근통제(access control)를 수행 하며 방화벽 같은 경우 모든 Inbound Traffic 을 모두 허용하지만 반대로 Outbound Traffic에 대해서는 지정한 수 이상의 연결이 되면 일정 수 이상에 한해서 Traffic을 차단하도록 되어 있다. 이와 같은 방식으로 비정상사용자가 외부로 Scanning 공격을 하거나, DDos 공격을 하는 것을 막는데 효과가 있었으며, 공격자에게는 Honeynet이 쉽게 노출 되는 상황도 발생하게 되었다. 데이터의 수집은 보통 네트워크 레벨인 3계층에서 이루어지며, FTP Service 또는 Telnet Service 사용정보를 쉽게 수집 할 수 있으며 로그서버를 이용하여 syslogd를 이용하여 로그를 수집하였다.

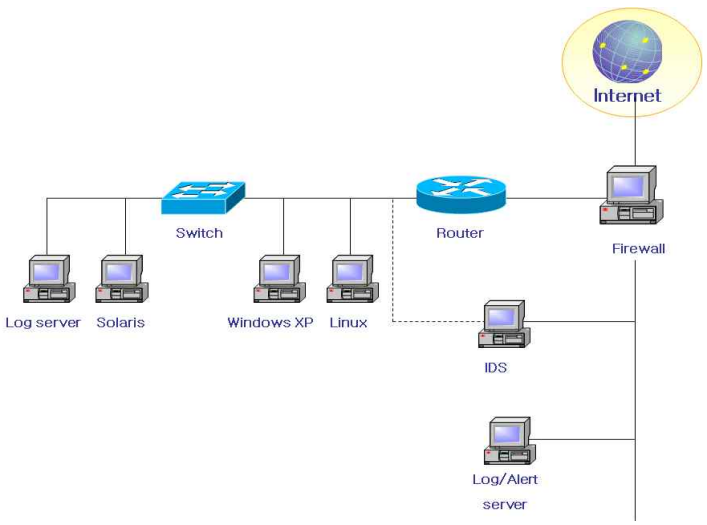


그림 2-6 1세대 Honeynet

2.4.1.2 Generation 2

2002년도에 구축이 되었으며, Layer 2 인 데이터계층에서 Data Control을 하며, signature 기반의 접근통제(access control)를 수행하며 네트워크의 Routing topology의 변경 또는 TTL의 값이 감소가 없어서 공격자는 접근통제시스템의 여부를 쉽게 알 수 없다. 2세대의 특징으로 기술적으로 발전한 해커의 공격 행위를 수집하는데 유용하고 외부로 나가는 패킷을 Layer 2 Level과 비교하여 해당하는 패킷이 존재할 때, 이 패킷을 수정하여 공격이 이루어지지 않도록 한다. 따라서 공격자는 Honeynet 내에서 파일 다운로드 등 원하는 모든 작업을 수행할 수 있으며 외부 시스템에 대한 공격도 시도할 수 있으나 공격이 성공하지는 않는다는 것을 알 수 있다. 현재 공격자의 행동을 탐지, 수집하기 위한 다양한 시스템 및 kernel module의 개발도 진행되고 있다.

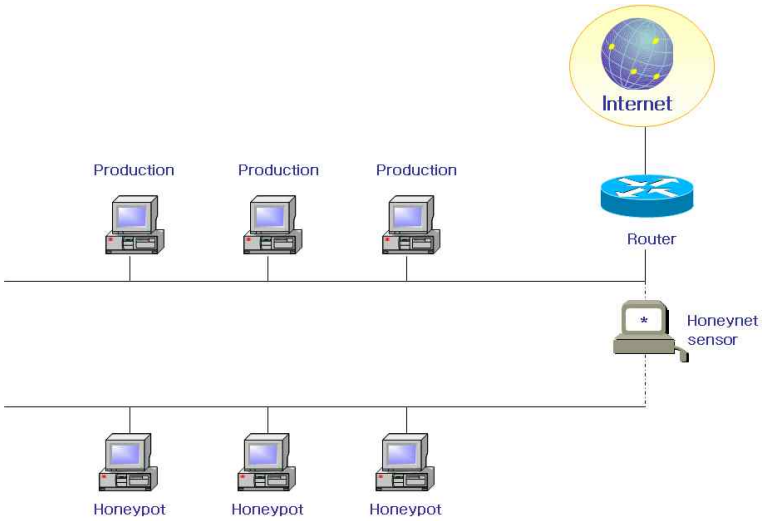


그림 2-7 2세대 Honeynet

2.4.2 Honeynet의 기본 구성

Honeynet은 하나의 시스템이 아닌 다수의 시스템으로 들어오고 나가는 트래픽이 통제되어야 하며 이러한 데이터를 수집하여 정보를 분석 하고 비정상적인 사용자의 공격 패턴, 공격 기술 등을 파악하여 대처 할 수 있으며, 수집한 정보는 비정상적인 사용자가 공격할시 조기에 관리자에게 알려줄 수 있으며 시스템 보안에 만전을 기여할 수 있다.

2.4.2.1 Data Control

Data Control의 개념은 Inbound 되는 Traffic은 허용 하고, Outbound 되는 Traffic은 제안하는 것으로 비정상적인 사용자의 행위를 Honeynet 내로 제안을 두어 외부 시스템에 대해 피해를 줄 수 있는 위험을 방지하는 것으로 이때 비정상적인 사용자가 Traffic이 통제되는 사실을 알지 못하도록 하는 것이 중요 하며 공격을 당한 후 에도 비정상적인 사용자가 외부의 시스템을 공격하지 못하도록 하는 방법이다.

2.4.2.2 Data Capture

Data Control의 개념은 비정상적인사용자가 컴퓨터에 접속하여 행동하는 모든 정보를 수집해야한다는 의미로 비정상적인사용자의 공격 정보를 모두 수집해야만 Honeynet의 의미라고 볼 수 없을 뿐만 아니라 침해를 당하여 Forensic 기법 사용 시 로그기록 및 증거 자료로 사용할 수 없기 때문이다. 여기서 정보수집 방법으로는 3계층인 네트워크 계층 과 7계층인 응용 프로그램, 시스템 레벨, 또는 사용자 행위 등 모든 레벨에서 수집하는 것이 가장 좋다. 이렇게 수집을 할 경우 자세한 데이터를 수집할 수 있다.

2.4.2.3 Data Collection

Data Collection 개념은 하나의 Honeynet이 아닌 다수의 Honeynet를 이용하여 정보를 수집하는 것으로 여러 Honeynet의 데이터를 효율적으로 분석할 수 있도록 수집하는 것이며 분산되어 있는 Honeynet 을 하나로 통합하는 것이다.

2.4.3 Honeypot의 형태

2.4.3.1 Production Honeypot

현재 Production Honeypot은 상용 목적으로 운영이 되고 있으며 조직 또는 특정 환경의 보안을 강화하고 위험을 감소시키는데 목적이 있다 현재는 Honeypot의 개념으로 상용된 Honeypot 시스템이다. 일반적으로 설치 및 시스템에 적용하기 쉬운점이 있으나 구입을 해야하는 하는 것이 특징이다. 쉬운 예로 Hot Zone 이 있다.

2.4.3.2 Research Honeypot

현재 Research Honeypot은 연구 목적으로 운영이 되고 있으며 Hacker community에 대한 정보를 얻어서 연구를 하기 위한 목적을 갖고 있으며 일반적으로 시스템에 설치 및 적용 그리고 유지하는데 복잡하다는 특징이 있다. 쉬운 예로 Honeynet Project 가 있다.

2.4.4 Honeypot 도구들

2.4.4.1 Deception Toolkit(DTK)

최초의 Open Source Honeypot solution으로 Fred Cohen이 개발하였다. Perl scripts와 Ccode로 이루어져 있으며, 다양한 종류의 서비스를 emulate 한다. 하지만, DTK는 1998년에 발표되고, 1999년까지 update 되었으나, 그 이후에는 더 이상 update 되지 않았으며,documentation이 부족하고, 수정 및 사용이 어려운 점이 있다.

2.4.4.2 CyberCop Sting

1998년에 Alfred Huger가 개발하였으며, NAI사 가 판매한 최초의 상용 Honeypot 툴이다. 동시에 여러 가지 다른 시스템을 application level 및 IP stack level에서 emulate 할 수 있다. 사용이 편리하게 만들어졌으나, 현재는 더 이상 판매되지 않고 있다.

2.4.4.3 NetFacade

1998년에 Marty Roesch가 개발하였으며, 동시에 여러 가지 다른 시스템을 emulate 할

수있다. 중요한 특징으로는 NetFacade 한대를 이용하여 하나의 C class 네트워크 전체를 emulate 할 수 있다는 것이다. 상용 툴로서 현재 판매되고 있으며, 주로 정부나 군사기관에서 사용되고 있다. Marty Roesch가 개발한 공개용 침입탐지시스템인 snort는 NetFacade를 troubleshoot 하면서 만들어졌다고 한다.

2.4.4.4 BackOfficer Friendly

1998년에 Marcus Ranum이 개발하였으며, 최초의 desktop기반의 Honeypot이다.

2.4.4.5 Hot Zone

2002년 4월 Marcus Ranum이 개발한 공개 Honeypot이다. 다양한 서비스와 취약점을 emulate 하며, 주요 특징은 centr alized logging 및 자동 업데이트가 가능하다는 점이다.

2.4.5 공격자와 허니팟의 상호작용(Interaction)

공격자가 시스템에 침입해서 자신이 원하는 행동을 어느 정도까지 할 수 있느냐는 허니팟의 보안, 데이터의 질과 양에 영향을 준다. 허니팟은 저마다 다른 목적을 갖고 있다. 이러한 목적에 따라 허니팟은 다양한 레벨의 기능을 갖게 되며, 공격자에게 제공하는 기능도 이에 따라 달라진다. 허니팟이 제공하는 상호 작용레벨에 따라 장단점이 있으므로, 어떤 형태의 허니팟을 구축할 것인지 결정하는 것 또한 중요하다.

2.4.5.1 Low level

보통 Telnet 또는 FTP와 같은 서비스와 해당 취약점을 emulate 한 것으로, 공격자는 로그인시도 및 매우 제한적인 상호작용만이 가능하다. 이를 통해 획득할 수 있는 정보는 적다.

2.4.5.2 Medium level

제한된 서비스 환경을 제공하는 것이며, 제한된 상호작용을 할 수 있도록 한다. 공격자는 원래의 시스템에 대해 영향을 줄 수 없으나, chroot 환경에 대한 공격이나, 설정의 잘못에 의해 전체 시스템에 대한 권한을 획득할 수 있는 위험성을 갖고 있다. 실제 사용되는 경우는 드물다.

2.4.5.3 High level

실제 OS 시스템 자체로 이루어진 허니팟이다. 어떠한 서비스도 emulate 하지 않고 실제 제공하며, 주로 연구 및 대응을 위해 사용된다. 공격자가 다른 시스템 또는 조직에 피해를 줄 수 있는 위험성도 크다.

3. Honeynet 구축

3.1 Bridgie Firewall 구축 과정

브리지 방화벽은 NAT와 같이 이더넷 카드가 2장 있어야 하는데, 이 두 장의 카드는 0.0.0.0 으로 설정하고 이 두 장의 이더넷 카드를 합쳐서 하나의 가상 인터페이스로 설정하게 된다.
따라서 방화벽을 기준으로 내부와 외부 케이블이 eth0 이나 eth1등 어떤 인터페이스에 연결되어도 관계없다. 참고로, NAT 의 경우 외부 케이블은 공인 ip가 할당된 인터페이스에, 내부 케이블은 사설 ip 가 할당된 인터페이스에 연결하여야 했다.

이렇듯 리눅스 시스템을 브리지 형태로 구성하려면 2.4 버전에서는 커널패치를 해준 뒤 커널 메뉴에서 브리지 지원 메뉴를 별도로 선택해 주고, 2.6 이후버전에서는 커널 자체에서 지원함으로 별도의 커널패치를 하지 않아도 된다.

.

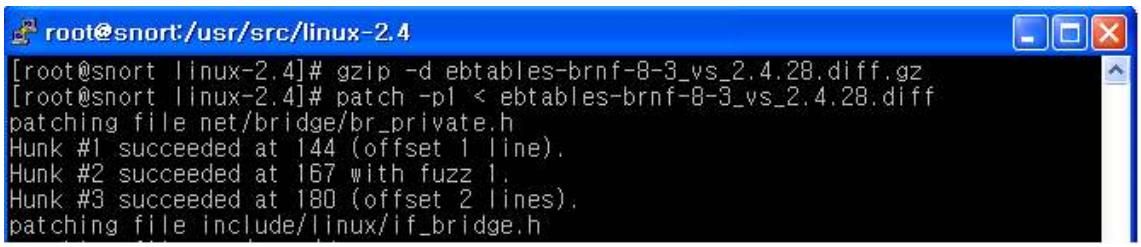


그림 3-1 패치 파일 다운

2.4 커널의 패치를 위해서는 커널 소스 디렉토리인 /usr/src/linux/ 디렉토리로 이동한 후 http://ebtables.sourceforge.net 에서 자신의 커널 버전에 맞는 패치 파일을 다운로드하도록 한다. 이후 이 파일을 gzip -d 로 압축해제 후 패치하면 된다. 이후 make menuconfig 를 실행한 후 Networking options에서 확인해 보면 아래와 같이 802.1d Ethernet Bridging 를 선택하면 Bridge: ebtables 라는 것이 추가된 것을 알 수 있으며 이 메뉴를 선택하면 된다.

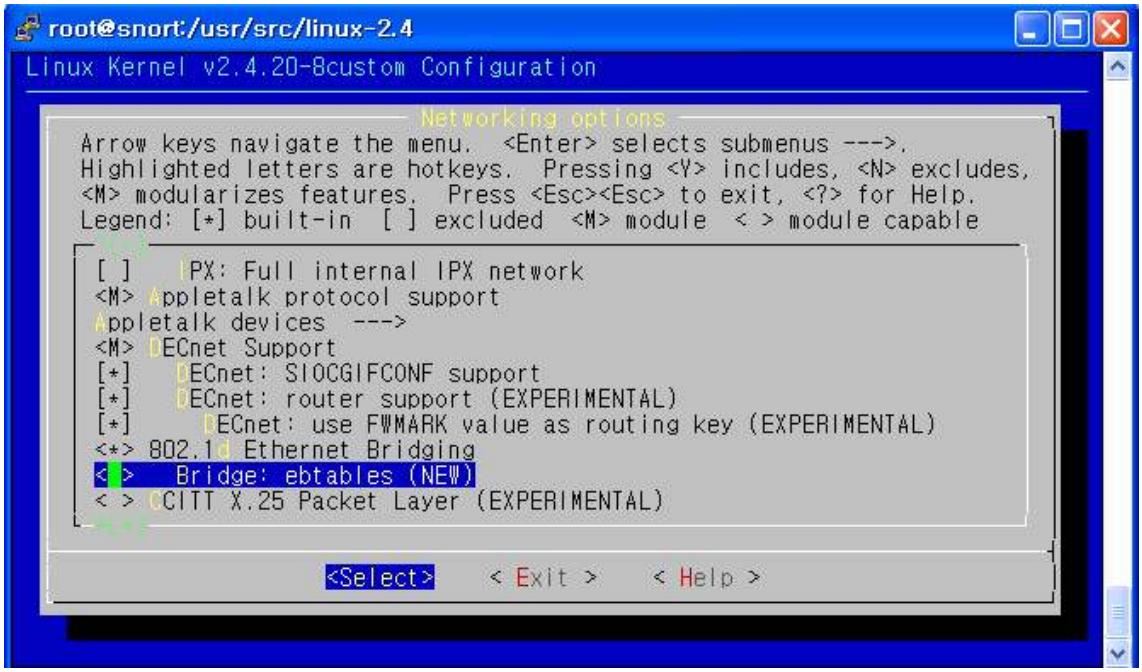


그림 3-2 커널 설정 화면

이후 커널컴파일 및 재부팅하면 사용할 준비가 되었다. 다음으로는 브리지를 설정하기 위한 관리 프로그램인 bridge-utils를 설치하도록 한다.

bridge-utils 다운로드 : http://bridge.sourceforge.net

이후 아래와 같이 압축 해제 후 ./configure; make 로 컴파일만 하면 된다. 이후 생성된 brctl 실행파일을 /sbin/으로 옮기도록 한다.

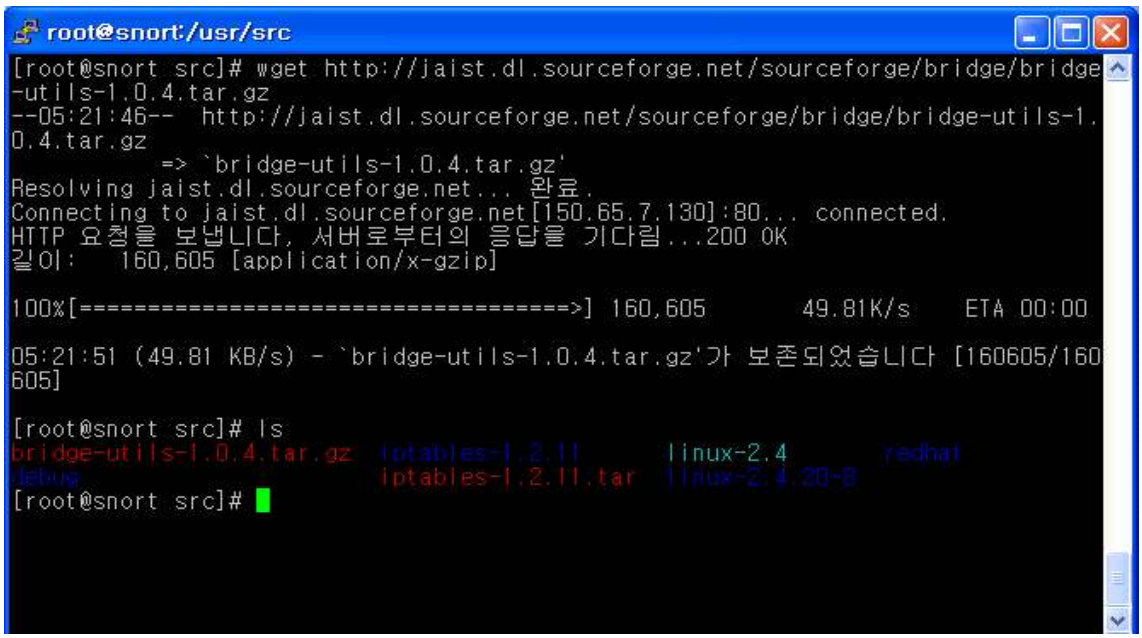


그림 3-3 브리지 유틸리티 압축 해제

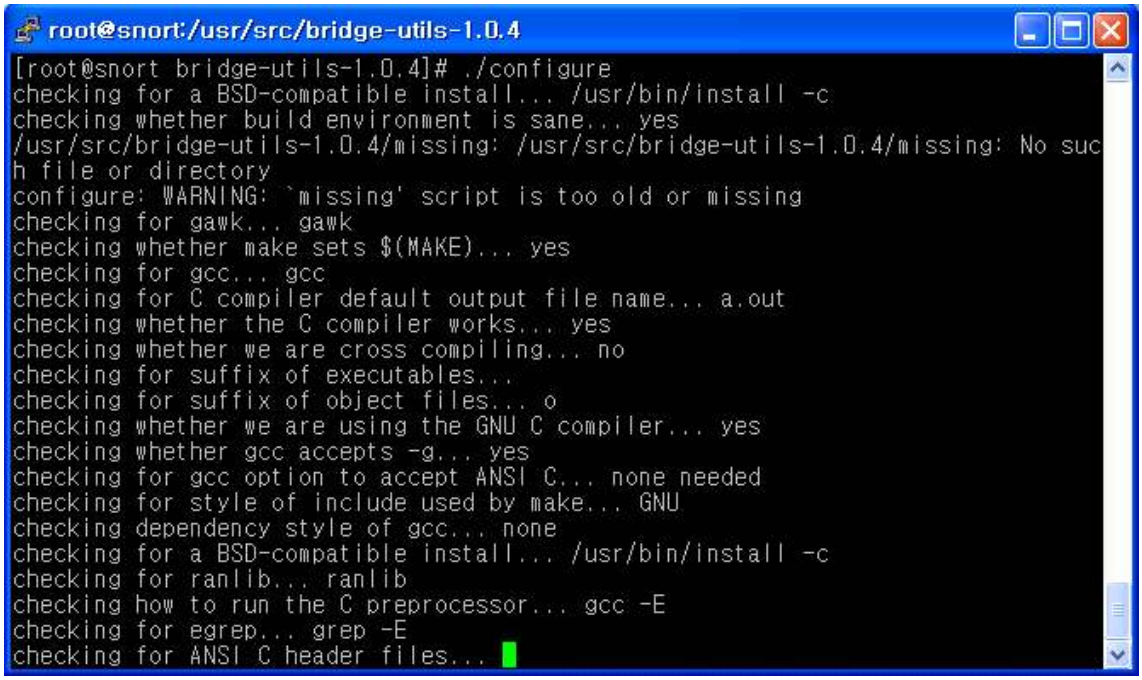


그림 3-4 브리지 유틸리티 환경 설정

```
[root@firewall bridge-utils-x.x.x]# make
[root@firewall bridge-utils-x.x.x]# mv brctl/brctl /sbin/
```

3.1.1 brctl 설정하기

bridge-utils인 brctl을 설치한 후 실행하면 사용할 수 있는 옵션이 많이 있다. 브리징이나 스패닝트리(stp)등이 있으며 실제 브리지 방화벽을 운영할 때는 아래와 같은 옵션만 알고 있으면 된다. 실제 아래와 같은 스크립트를 만들어 br.sh와 같은 파일로 작성 후 부팅할 때 자동으로 실행하도록 설정한다.

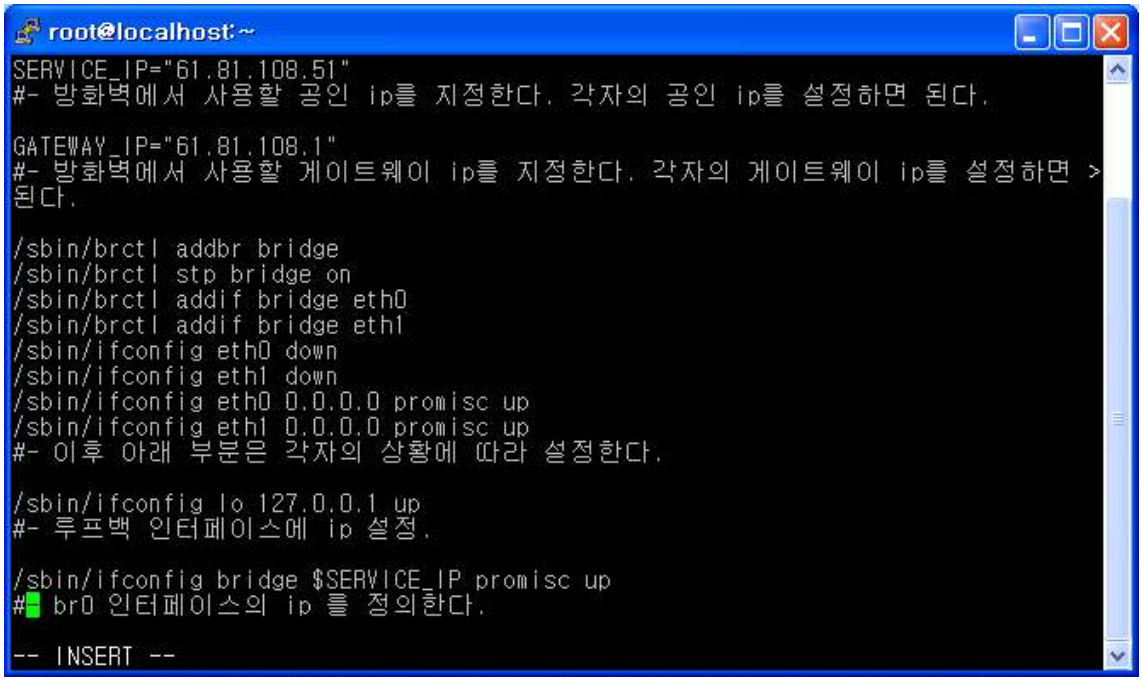


그림 3-5 자동 실행 br.sh 작성

위와 같이 설정 후 ifconfig를 실행하면 다음과 같이 보일 것이다.

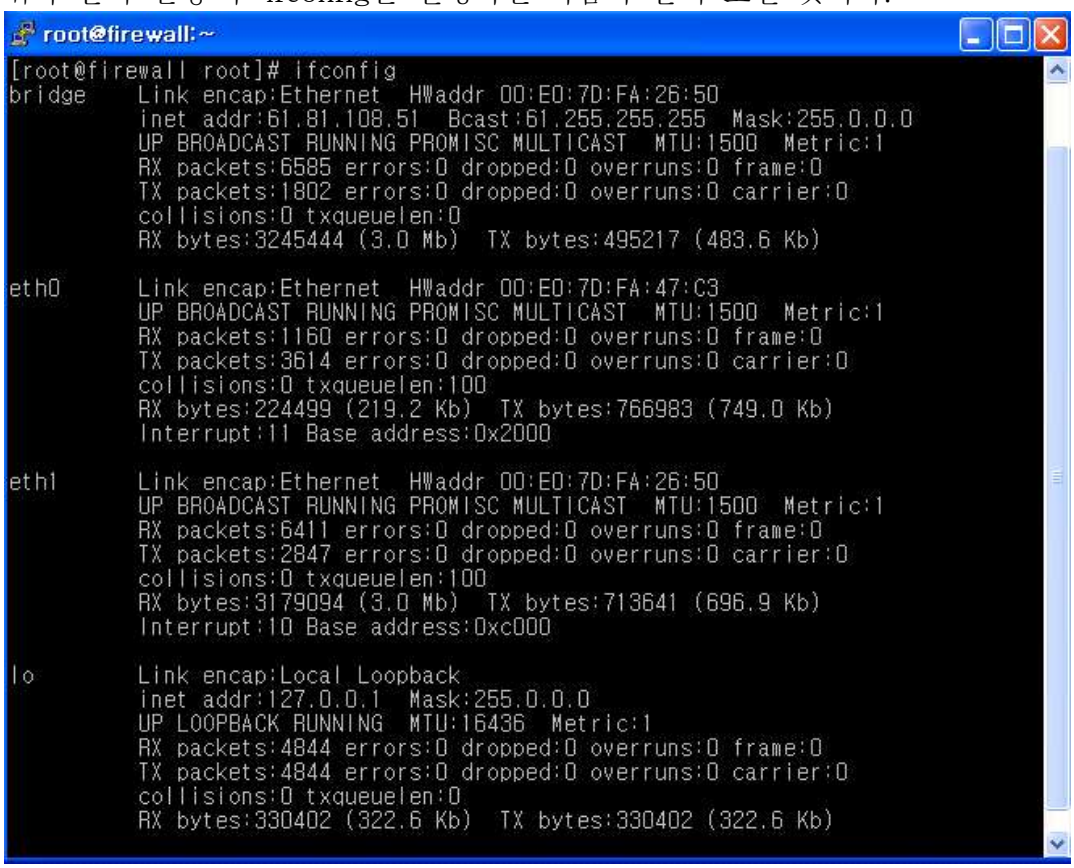


그림 3-6 ifconfig

route -n 이나 netstat -r 을 실행하면 아래와 같이 bridge 인터페이스가 보일 것이다.

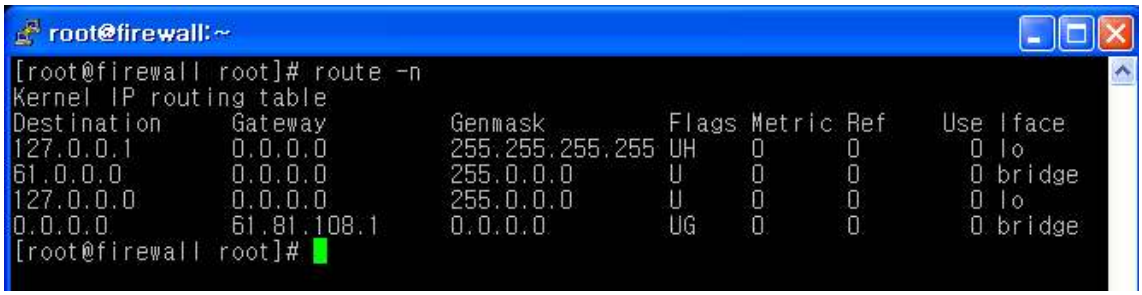


그림 3-7 route -n

현재 브리지 상태를 조회하려면 다음과 같이 brctl show를 실행하면 된다.

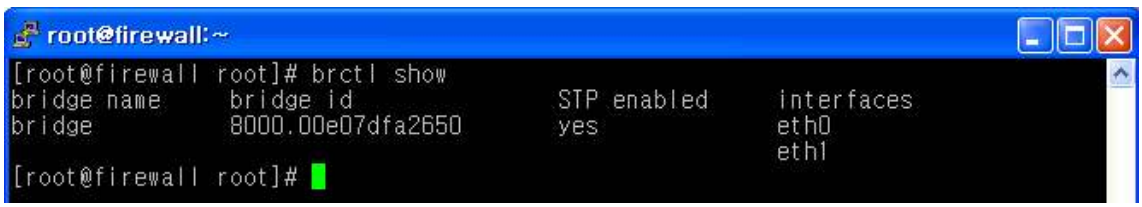


그림 3-8 brctl show

여기에서 만약 브리지를 해지하고 재설정하려면 먼저 해당 브리지에 설정되어 있는 인터페이스를 삭제한 후 브리지를 삭제하면 된다. 물론 원격접속을 한 상태라면 네트워크가 끊기게 되므로 주의하여야 한다. 아래와 같이 스크립트 파일을 생성한 뒤 원격접속이 아닌 리눅스 콘솔상에서 실행한다.

```
#!/bin/sh
ifconfig bridge down
brctl stp bridge off
brctl delif bridge eth0
brctl delif bridge eth1
brctl delbr bridge
ifconfig eth0 down
ifconfig eth1 down
ifconfig eth0 61.81.108.51 promisc up
ifconfig eth1 192.168.1.1 promisc up
```

3.2 iptables 구축 과정

3.2.1. iptables 설치 및 리눅스 커널 설정하기

각자 위와 같은 구조에 따라 서버자체 또는 NAT나 브리지 방식을 적절히 사용하면 될 것이다. 이제 리눅스 방화벽의 기본이 되는 netfilter 기반의 iptables에 대해 알아보고 이는 어떤 장점과 기능을 제공하는지 알아보자 iptables는 다른 상용 방화벽이 제공하는 기능을 대부분 가지고 있는데, 그 중에서 가장 대표적인 기능 또는 이전 버전에 비해 향상된 기능은 다음과 같다.

3.2.1.1 상태추적 기능 제공

최근의 방화벽에서 제공하는 고급기능중 하나인 상태추적은 방화벽을 통과하는 모든 패킷에 대한 연결 상태를 추적(tracking)하여 이 정보를 메모리에 기억하고 있다가 기존의 연결을 가장하여 접근할 경우 메모리에 저장된 상태 목록과 비교하여 적합하면 통과하고 그렇지 않으면 거부하는 기능으로서 지능화된 공격시도를 차단할 수 있는 기능중 하나이다. 이를테면 단순히 포트나 tcp flag 등을 매칭하여 필터링한 구 버전의 방화벽에서 기존의 연결 없이 단지 ack flag를 설정한 tcp 패킷이 들어온다면 이미 연결중인 트래픽으로 판단하여 허용하지만 상태 추적이 제공될 경우에는 아무리 ack flag를 달고 들어온다 하더라도 이전의 접속 목록에 syn 및 syn/ack 와 관련된 정보가 없기 때문에 비정상 트래픽으로 간주하여 필터링하게 되는 것이다. iptables 이전 버전인 ipchains 와 같이 상태추적을 제공하지 않는 방화벽은 stateless 라고 하며 지금부터 알아볼 iptables 와 같이 상태추적이 제공되는 방화벽은 stateful 이라고 한다.

3.2.1.2 향상된 매칭 기능 제공

iptables는 방화벽에서 기본적으로 제공하는 매칭 정보인 패킷의 소스 ip, 목적지 ip 및 소스 포트, 목적지 포트 번호 뿐만 아니라 추가적으로 다양한 매칭 기능을 제공하고 있다. 이를테면 상태 추적을 통한 현재의 연결 상태나 하드웨어 MAC 주소, 패킷 발신자의 유저나 그룹 프로세스, ip 헤더의 TOS(Type Of Service)등 여러 가지 조건을 이용하여 세부적이고 복잡한 매칭 및 필터링이 가능하다. 물론 일부 기능은 추가적으로 커널 패치를 통해 구현할 수 있다.

3.2.1.3 포트 포워딩(port forwarding) 기능 포함 제공

이전 버전인 ipfw이나 ipchains를 사용할 때는 NAT를 이용하기 위해서 별도로 분리되어 있던 툴인 ipmasqadm을 사용하여야 했으나 iptables에는 NAT 기능이 자체적으로 포함되어 있어 NAT를 위해 별도의 프로그램을 이용할 필요가 없이 커널 메뉴에서 추가로 지정해서 컴파일 해 주면 iptables만으로도 바로 사용할 수 있다. 2.6버전이 아닌 2.4버전의 커널 사용시에는 아래와 같은 커널 설정확인을 거쳐야한다. STEP 1. KERNEL 설정하기 (버전 2.4 기준) 커널소스 디렉토리에서 make config나 make menuconfig를 실행하여 커널 메뉴로 들어가면 된다. 나머지는 각자 환경에 따라 적용하기 바라며 커널 2.4 버전의 경우 아래와 같이 보이게 된다. 이 중에서 방화벽 관련 설정은 “Networking options” 부분에 정의되어 있으므로 이 메뉴를 선택하도록 한다.

매우 많은 선택 메뉴들이 있는데, 가장 기본적인 메뉴만 위와 같이 선택하도록 한다. 만약 모듈(module)로 선택하였을 경우에는 이후에 해당 모듈을 로드(load) 해주어야 한다. 그리고 선택하려는 메뉴가 어떤 기능과 역할을 하는지 모른다면 해당메뉴에서 우측의 Help를 클릭하면 상세한 안내를 볼 수 있으니 참고하면 된다. 그럼 다음과 같은 순서로 KERNEL을 설정하도록 해보자.

3.2.2 iptables 설치하기

① Kernel 메뉴설정

make config (또는 make menuconfig)

입력 → Networking options 선택

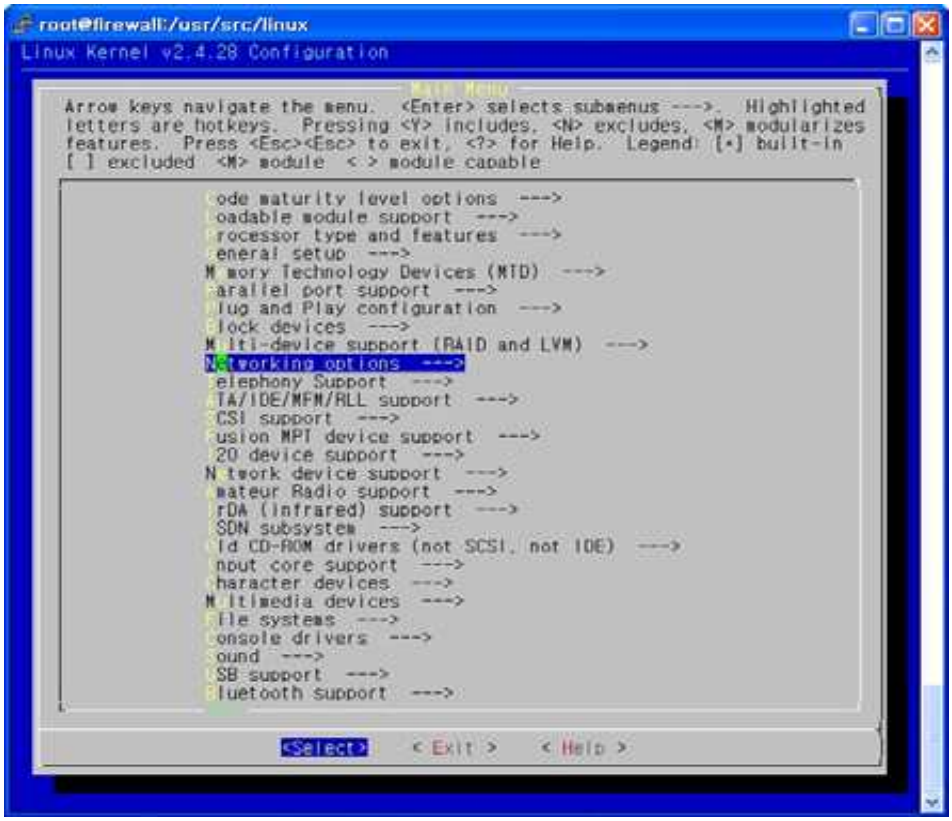


그림 3-9 Kernel 메뉴설정

② netfilter 설정확인

IP Netfilter Configuration 선택

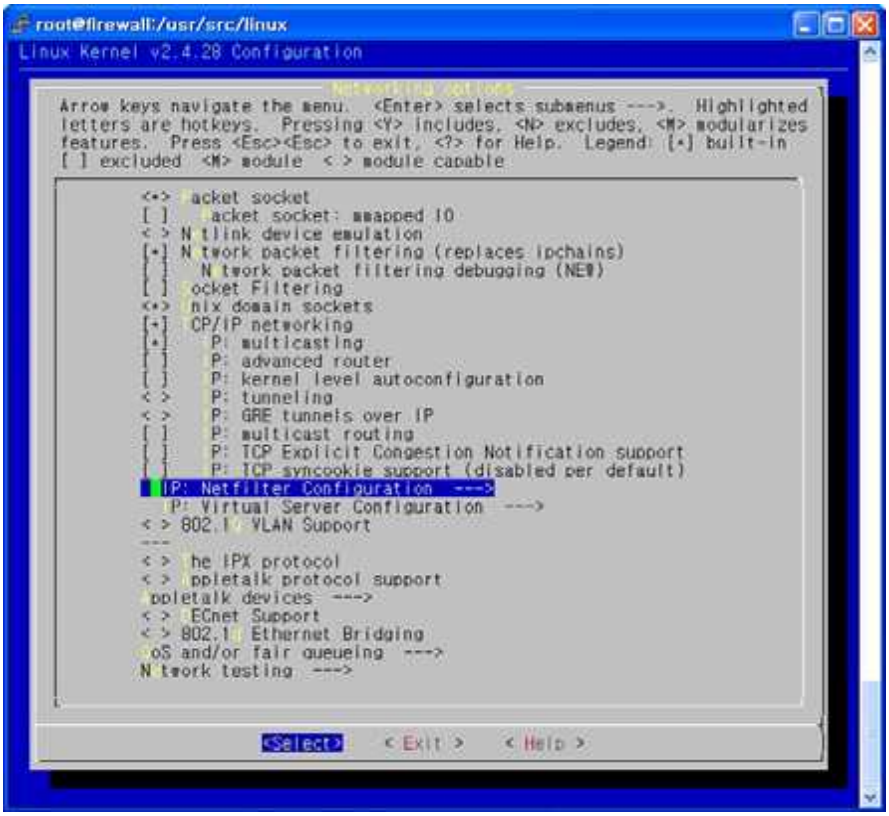


그림 3-10 netfilter 설정확인

③ netfilter 설정확인

Connection tracking 선택

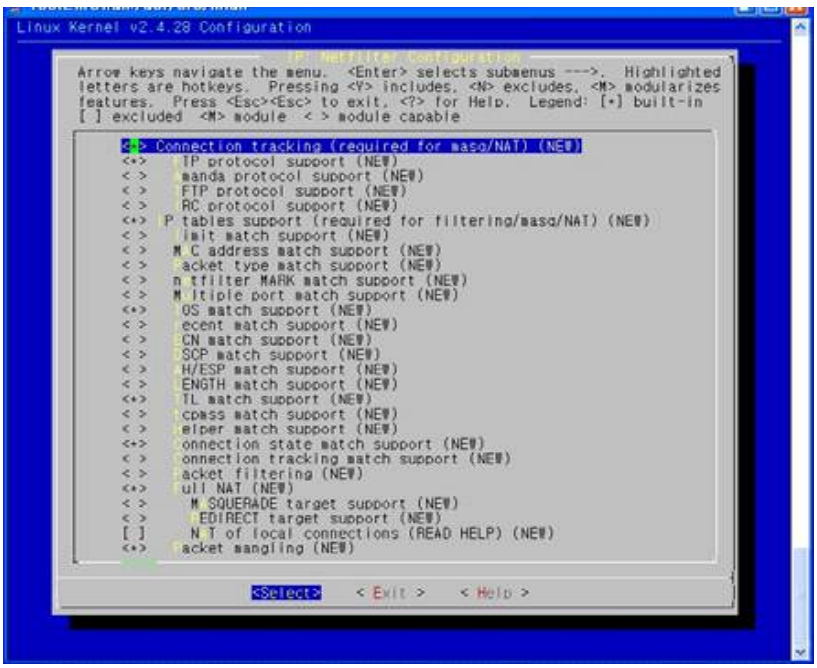


그림 3-11 netfilter 설정확인

④ netfilter 설정확인

LOG target support 선택

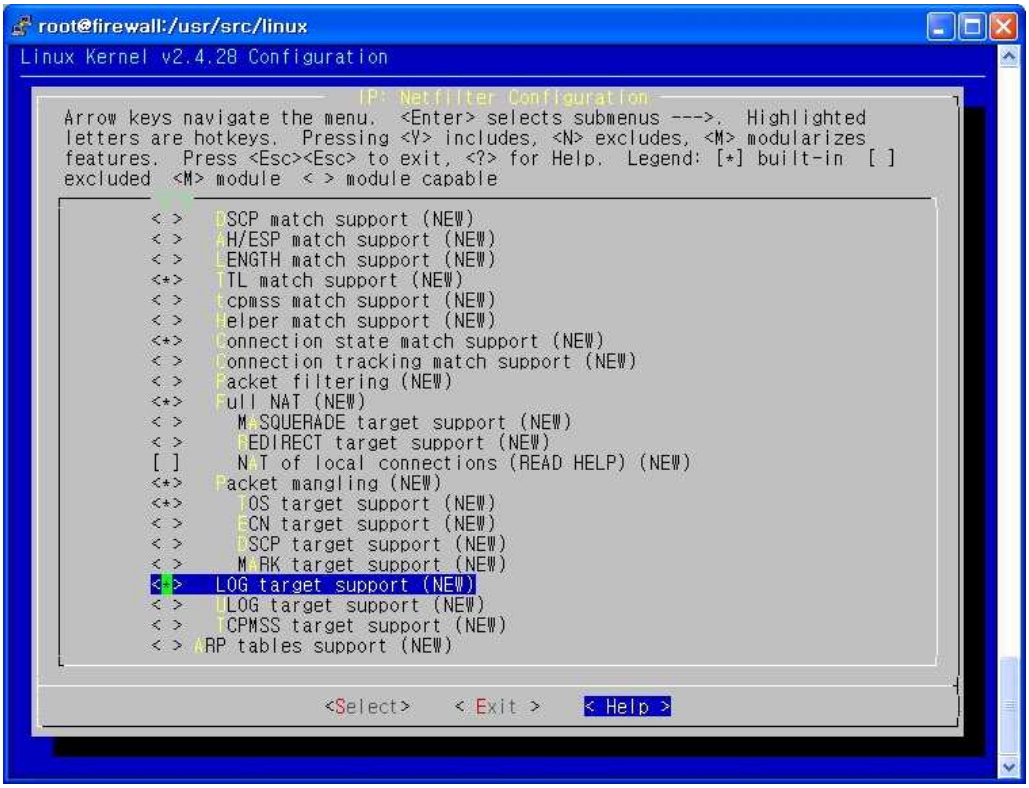


그림 3-12 netfilter 설정확인

최근의 리눅스 배포판에서는 별도로 커널 컴파일을 하지 않고 CD로 설치된 상태에서도 기본적으로 iptables 방화벽 기능이 모듈로 작동하고 있지만 가급적 커널 컴파일을 통해 불필요한 기능은 끄고, 꼭 필요한 기능만 사용하도록 하는 것을 권장한다.

iptables 방화벽은 시스템의 응용 프로그램 수준이 아니라 커널 수준에서 동하기 때문에 먼저 커널에서 방화벽이 작동할 수 있도록 설정하여야 한다. 이를 위해서는 커널소스를 다운받아 커널 컴파일을 하여야 하는데, 소스 파일은 아래 URL에서 다운로드 할 수 있다. 그럼 다음과 같은 순서로 설치하도록 하자.

① iptables rpm 제거

그림 3-13 iptables rpm 제거

rpm -q iptables : 먼저 RPM 명령어를 이용하여 기존의 iptables 버전을 확인한다.

rpm -e iptables --nodeps : 최신 버전이 아니라면 기존의 iptables 패키지를 삭제하여 새로 설치 할 준비를 한다.

② iptables 소스 다운받기

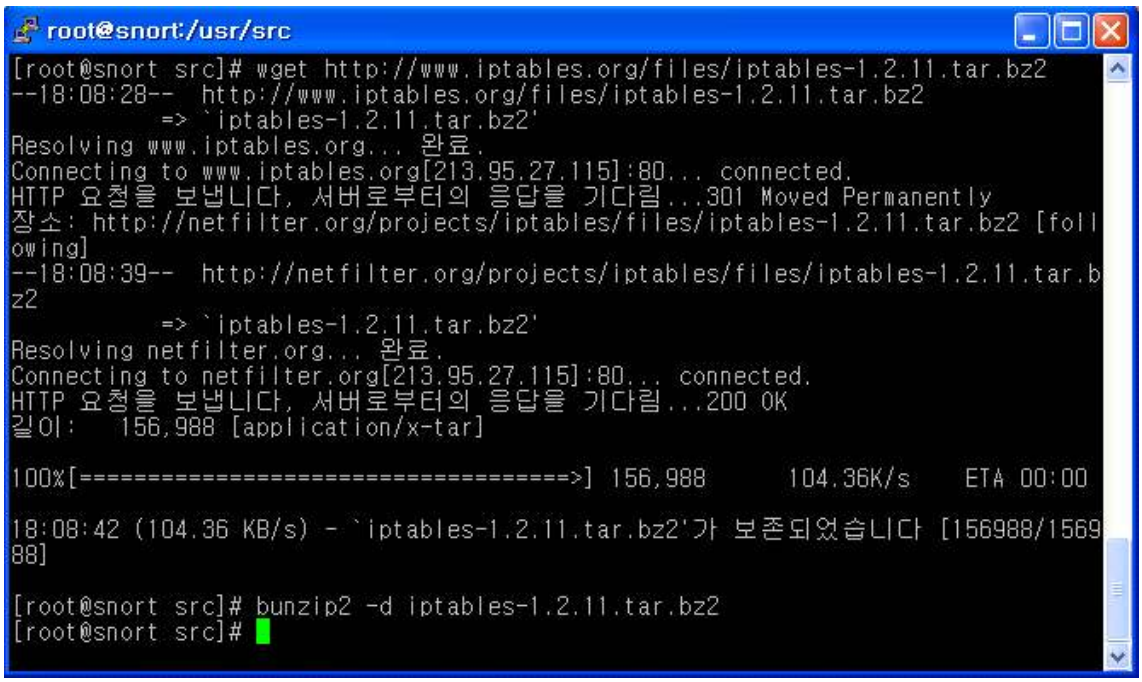


그림 3-14 iptables 소스 다운받기

wget http://www.iptables.org/files/iptables-1.2.11.tar.bz2 : wget이나 lynx를 이용하여 iptables 최신 소스 다운을 받는다.

bunzip2 -d iptables-1.2.11.tar.bz2 : 다운 받은 파일의 압축을 푼다.

③ 소스 압축풀기

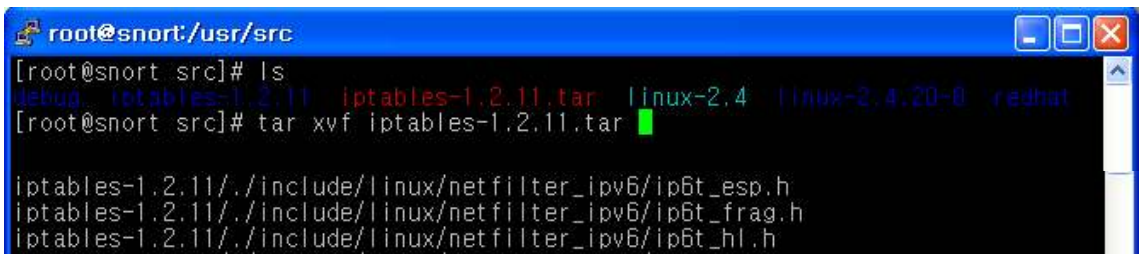


그림 3-15 소스 압축풀기

rm -f linux : 링크를 확인 후 존재한다면, 삭제한다. linux 디렉토리가 다른 디렉토리로 링크되어 있는지 확인한다.

tar xvf iptables-1.2.11.tar : tar 압축된 파일을 푼다.

④ iptables 디렉토리로 이동

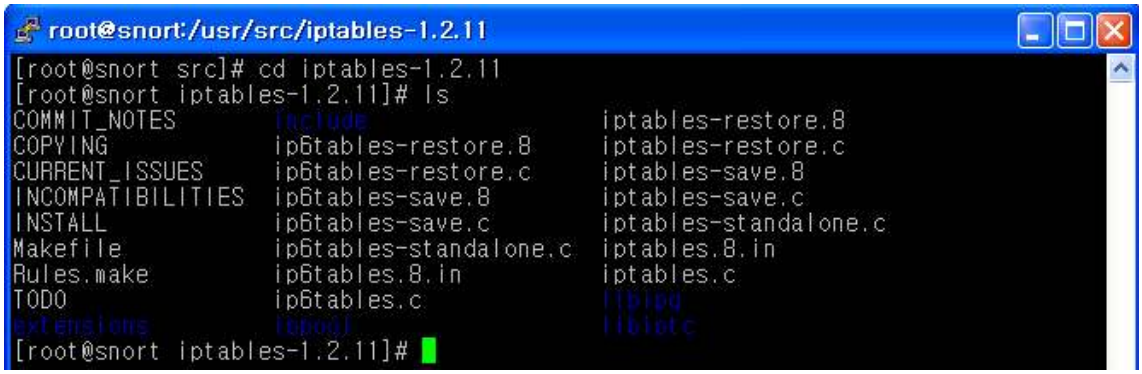


그림 3-16 iptables 디렉토리로 이동

cd iptables-1.2.11 : 압축을 푼 디렉토리로 이동한다.

⑤ ipatbles 컴파일

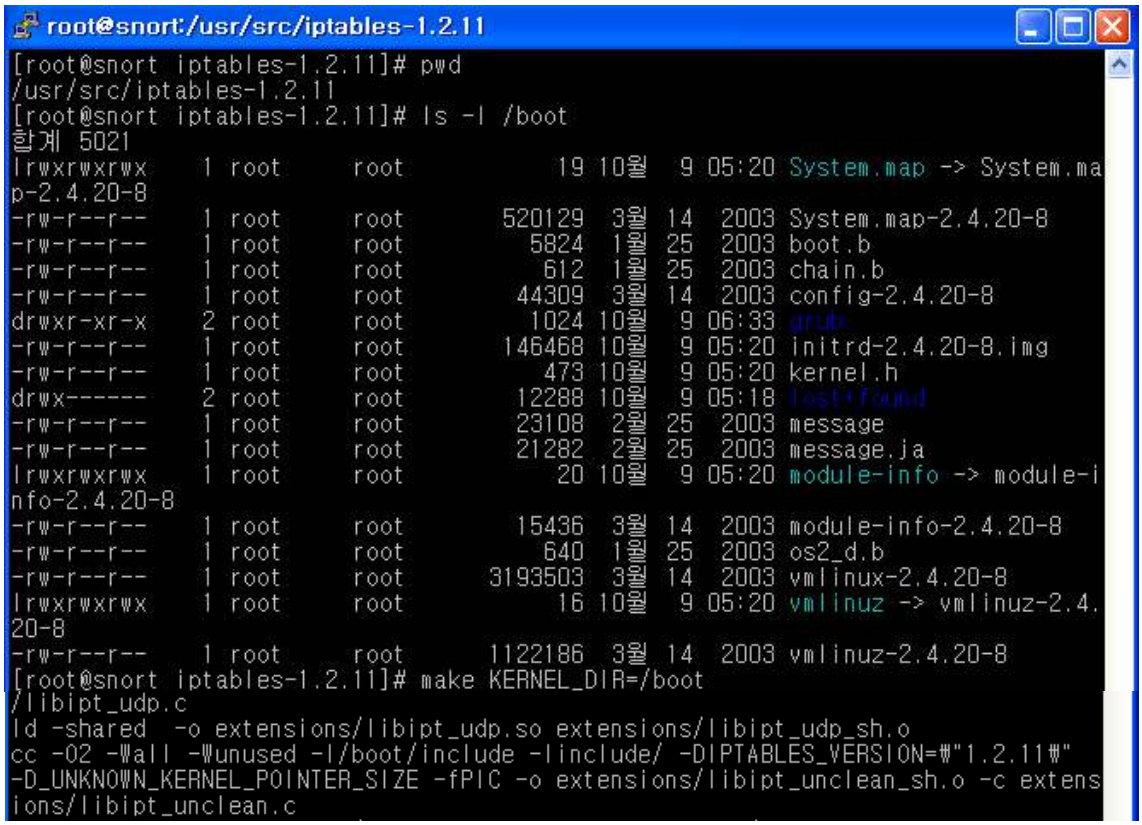


그림 3-17 ipatbles 컴파일

make KERNEL_DIR=/boot : make를 이용하여 컴파일 한다. KERNEL_DIR을 커널디렉토리로 지정해준다.

⑥ iptables install하기

그림 3-18 iptables install하기

make install : 컴파일 후 인스톨을 수행한다. KERNEL_DIR을 커널디렉토리로 지정해준다.

⑦ 설치 확인하기

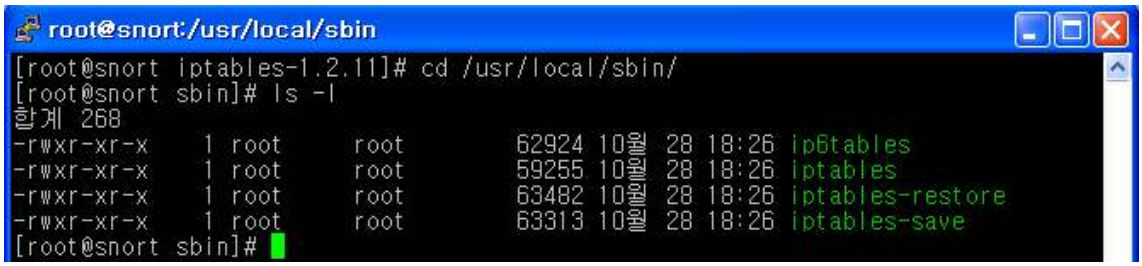


그림 3-19 설치 확인하기

cd /usr/local/sbin/ : 설치된 파일을 확인하기 위해 설치 폴더로 이동하여 파일을 확인한다. 상기와 같은 파일이 존재한다면 설치는 이상없이 완료되었음을 확인할 수 있다.

3.2.3 iptables를 활용한 보안 룰 설정하기

① 룰 초기화 하기

iptables -L : 현재 설정된 룰을 확인하다.
iptables -t filter -F : filter 룰을 삭제한다.
iptables -t nat -F : nat 룰을 삭제한다.
iptables -t mangle -F : mangle 룰을 삭제한다.

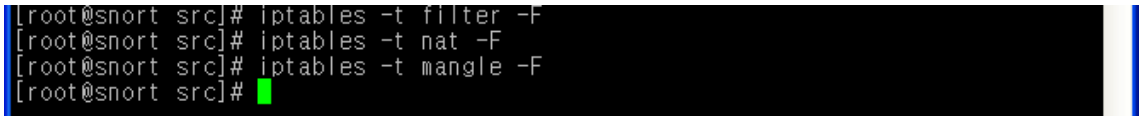


그림 3-20 룰 초기화

iptables를 이용하여 방화벽 룰을 설정할 때 제일 먼저 하여야 할일 혹 이미 존재할지 모르는 기존의 룰을 모두 삭제하는 것이다. iptables -L을 실행하여 현재 설정된 룰을 확인하여 하나하나 삭제해도 되지만 기존에 존재하는 모든 룰을 한꺼번에 삭제할 수 있는데, 이러한 것을 플러싱(flushing)한다고 한다. 다음과 같이 특정한 chain을 지정하지 않을 경우에는 INPUT이나 OUTPUT등 모든 테이블의 룰을 동시에 초기화한다.

② Loopback 트래픽 허용하기

iptables -A INPUT -i lo -j ACCEPT : Loopback에 대해 INPUT을 허용한다.
iptables -A OUTPUT -o lo -j ACCEPT : Loopback에 대해 OUTPUT을 허용 한다.

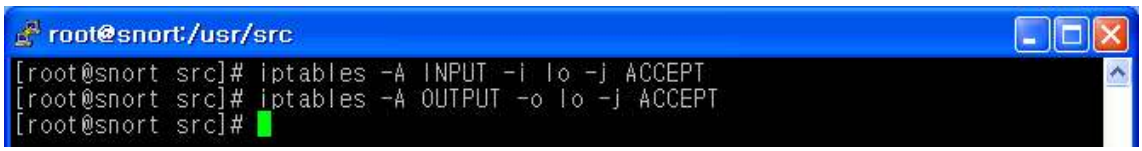


그림 3-21 Loopback 트래픽 허용하기

루프백 인터페이스는 시스템 내부의 가상 인터페이스이므로 루프백과 관련된 모든 트래픽은 허용하는 것이 좋다. 현재 자신의 시스템에서 루프백 트래픽이 사용되는지 알아보려면 간단히 tcpdump -i lo 로 확인해 보기 바란다. 여기에서 -i 는 interface 의 의미이다. 위 그림은 lo 인터페이스를 통해 들어오는 패킷과 lo 인터페이스를 통해 나가는 패킷을 허용 설정한 예이다. 이 룰에서의 -i 은 in 의 의미이고, -o 은 out 의 의미이다.

③ 기본정책(default policy) 설정

iptables -P INPUT DROP : INPUT 정책의 기본은 DROP으로 설정한다.
iptables -P FORWARD DROP : FORWARD 정책의 기본은 DROP으로 설정한다.
iptables -P OUTPUT ACCEPT : OUTPUT 정책의 기본은 ACCEPT으로 설정한다.

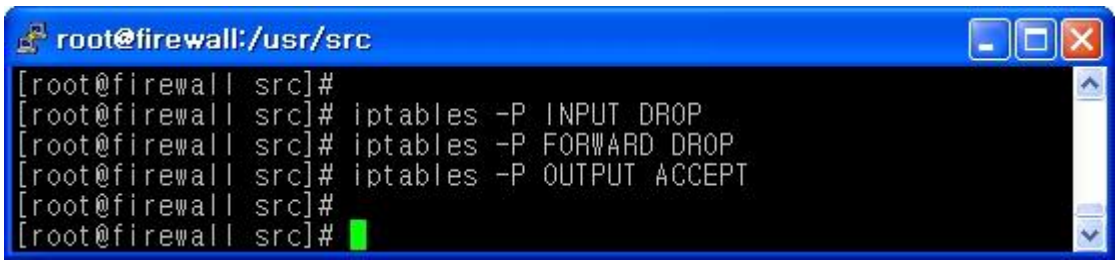


그림 3-22 기본정책(default policy) 설정

기본 정책은 앞에서 설명한 바와 같이 지정한 모든 룰에 매칭되지 않을 때 최종적으로 매칭되는 것으로 대문자인 -P 로 표현하며 **ACCEPT, DROP 둘 중에 하나가 사용된다**. 기본 정책으로 REJECT는 사용하지 않으며 -j DROP 과 같이 -j 를 사용하지 않는다는 점을 주의하기 바란다. 대부분 기본 정책으로 DROP을 설정하는데, 특히 원격에서 설정시에는 사전에 허용 정책이 없으면 접속 자체가 끊겨 버리므로 주의하여야 한다.

상기의 내용은 **INPUT, FORWARD, OUTPUT 에 대해 기본 정책을 설정한 예인데**, 자체 방화벽에서 FORWARD 는 사용하지 않기 때문에 DROP을 설정하였고, 들어오는 패킷만 제대로 처리 하면 되므로 OUTPUT은 ACCEPT를 설정하고 INPUT은 DROP을 하였다. 여기에서 OUTPUT을 ACCEPT 로 설정한 이유는 어차피 INPUT 과 OUTPUT이 동시에 ACCEPT 되어야 통신이 될 수 있는데, OUTPUT 만 ACCEPT 해 두면 외부에서 내부로 들어오는 패킷에 대해서는 엄격하게 통제하되 내부에서 외부로 나가는 트래픽에 대해서는 허용하는 것이 좋기 때문이다. 이는 뒤에서 살펴볼 상태추적을 통해 이미 연결을 맺어 연결이 성립된 ESTABLISHED 와 RELATED 상태는 모두 허용하도록 하였기 때문에 가능하다.

④ 상태추적 설정

iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT :
INPUT 룰에 의해 허용된 트래픽을 지속적으로 허용한다.
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT :
OUTPUT룰에 의해 허용된 트래픽을 지속적으로 허용한다.

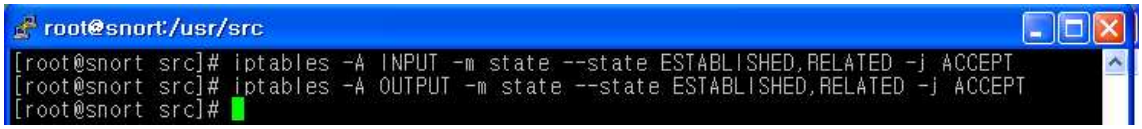


그림 3-23 상태추적 설정

상태추적이 제공되지 않는 stateless 구형 방화벽의 경우 매번 들어오고 나가는 패킷마다 패킷을 허용할 것인지 혹은 차단할 것인지 여부를 체크하여야 했지만 상태 추적을 이용 하면 이미 룰에서 허용이 된 트래픽의 경우 뒤이어 전송되는 모든 패킷을 다시 첫 번째 룰부터 검사할 필요 없이 바로 허용할 수 있다. 이것이 바로 상태 추적의 가장 큰 장점 중 하나이다. 따라서 아래 두 줄을 방화벽 룰 선정 시 가능한 먼저 설정해 주면 룰이 단순해지고 더욱 효율적이라 할 수 있다. 아래에서 ESTABLISHED 와 RELATED 는 이미 NEW 를 통해 트래픽이 허용된 후 이와 관련되어 통신하는 패킷에 대한 허용 설정이므로 NEW 에서만 제대로 설정해 주면 된다.

⑤ 사설 IP 주소로 필터링

iptables -A INPUT -s 10.0.0.0/8 -j DROP : 소스 IP가 10.0.0.0/8인 패킷을 차단한다.
iptables -A INPUT -s 172.16.0.0/12 -j DROP : 172.16.0.0/12인 모든 패킷을 차단한다.
iptables -A INPUT -s 192.168.0.0/16 -j DROP : 192.168.0.0/16인 모든 패킷을 차단한다.
iptables -A INPUT -s 224.0.0.0/4 -j DROP : 소스 IP가 224.0.0.0/4인 모든 패킷을 차단한다.
iptables -A INPUT -s 240.0.0.0/5 -j DROP : 소스 IP가 240.0.0.0/5인 모든 패킷을 차단한다.

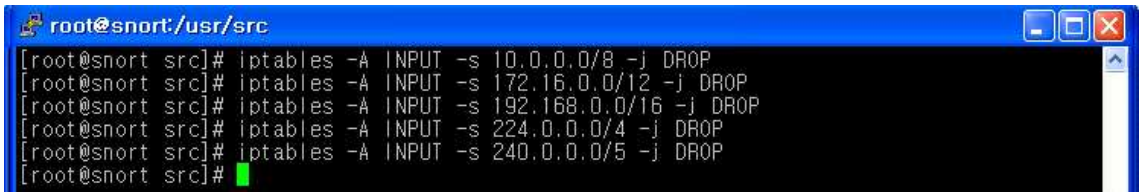


그림 3-24 사설 IP 주소로 필터링

IANA(http://www.iana.org/)에서 특별한 목적으로 사용하기 위해 예약해 둔 사설 ip 대역이 있다. 이러한 ip 주소는 RFC1918(www.ripe.net/db/rfc1918.html)에 명시되어 있는데, 특별한 목적으로 사용될 뿐 공인 네트워크인 인터넷에서는 라우팅될 수 없기 때문에 다음과 같이 사설 ip 주소를 소스로 하여 들어오는 트래픽은 위조된 트래픽이므로 차단하여야한다.

⑥ 루프백(Loopback) IP 주소 차단

iptables -A INPUT -i eth0 -s 127.0.0.0/8 -j DROP : 인터페이스 eth0를 통해 들어오는 127.0.0.0/8 인 IP 주소를 차단한다.

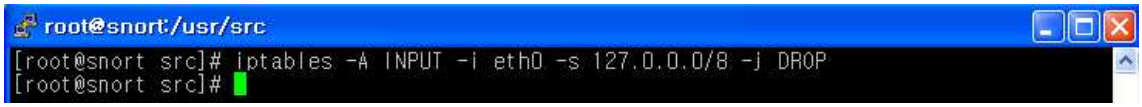


그림 3-25 루프백(Loopback) IP 주소 차단

루프백 ip 주소는 내부의 루프백 인터페이스(lo)를 통해서만 통신하기 때문에 루프백 주소를 소스로 해서 eth0과 같이 외부 인터페이스를 통해 들어오는 트래픽은 위조된 트래픽일 가능성이 높으므로, 보안상 차단하는 것이 좋다.

⑦ 예약된 IP 주소 차단

iptables -A INPUT -i eth0 -s 0.0.0.0/8 -j DROP :
인터페이스 eth0를 통한 IP주소가 .0.0.0/8 인 트래픽을 차단한다.
iptables -A INPUT -i eth0 -s 169.254.0.0/16 -j DROP :
인터페이스 eth0를 통한 IP주소가 0.0.0.0/8 인 트래픽을 차단한다.
iptables -A INPUT -i eth0 -s 192.0.2.0/24 -j DROP :
인터페이스 eth0를 통한 IP주소가 0.0.0.0/8 인 트래픽을 차단한다.
iptables -A INPUT -i eth0 -s 248.0.0.0/5 -j DROP :
인터페이스 eth0를 통한 IP주소가 0.0.0.0/8 인 트래픽을 차단한다.

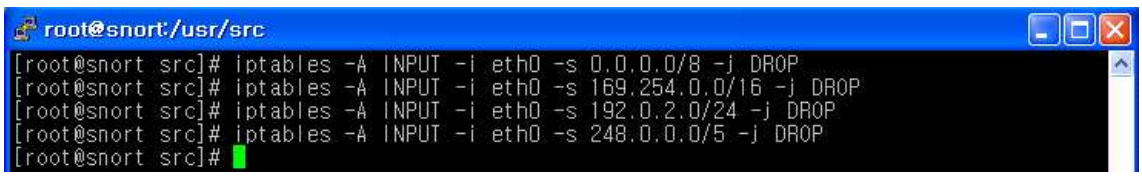


그림 3-26 예약된 IP 주소 차단

0.0.0.0/8 과 248.0.0.0/5는 예약된 ip 대역이며 169.254.0.0/16 은 DHCP등에서 임시로 사용하는 대역이며 192.0.2.0/24 는 TEST-NET 대역이다. 기타 차단하고 싶은 IP에 대해 상기와 같은 방법으로 차단하도록 한다.

⑧ SSH 서비스 허용하기

iptables -A INPUT -p TCP -s 192.168.10.11 --sport 1024:65535 --dport 22 -m state --state NEW -j ACCEPT : 소스 IP가 192.168.10.11에서 22번으로 들어오는 TCP 프로토콜을이용한 트래픽에 대해 허용하며, 상태 추적 기능을 이용한다.

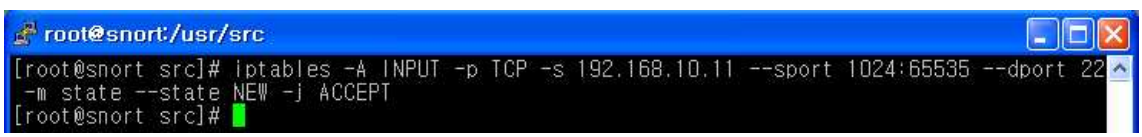


그림 3-27 SSH 서비스 허용하기

서버에 대한 원격 접속을 위해 최근에는 telnet에 대한 대안으로 세션을 암호화한 ssh가 많이 사용되고 있는 추세이다. ssh는 22/tcp 포트를 사용하므로 목적지 포트 22번에 대한 룰을 설정해 주면 된다. ssh 서버는 22번을, 클라이언트는 1024 이후의 임의의 포트를 사용하므로, 방화벽에서 각각의 포트를 허용해 주면 된다. 위 그림은 192.168.10.11에서 방화벽서버의 22/tcp 로의 접근을 허용하는 룰이다. 이후 방화벽에서 192.168.10.11 로 응답하는 패킷은 앞에서 허용한 상태추적 룰에 따라 허용되게 된다. 만약 상태 추적에서 허용하지 않았더라도 OUTPUT의 기본 정책이 ACCEPT이므로 OUTPUT에 대해서는 별도로 생각해 주지 않아도 된다. 만약 모든 소스 IP 에 대해 ssh를 허용하려면 192.168.10.11 부분에 0/0이나 any를 지정하거나 -s 부분을 빼면 된다.

iptables -A INPUT -p TCP -s 192.168.10.11 --sport 1024:65535 --dport 22 -j ACCEPT :

소스 IP가 192.168.10.11에서 22번으로 들어오는 TCP 프로토콜을 이용한 트래픽을 허용한다.

그림 3-28 22번이용 트래픽 허용

만약 상태추적을 사용하지 않는다면 위와 같이 -m state 이하 부분을 생략하고 포트번호만 명시해도 된다. 물론 OUTPUT의 기본정책이 ACCEPT이므로 OUTPUT 은 언급하지 않아도 된다.

iptables -A INPUT -p tcp --destination-port 22 -m mac --mac-source ! 00-AA-BB-04-CC-B2(관리자의 MAC 주소) -j DROP : MAC 주소가 00-AA-BB-04-CC-B2에서만 22번으로 접근 가능, 이외의 모든 MAC 주소를 가진 접근은 차단한다.

그림 3-29 MAC 주소 접근 설정

관리자의 IP 주소가 일정하지 않을 때는 관리자의 MAC 주소를 이용하여 위와 같이 접근 제한을 할 수 있다.

⑨ 메일 서비스 허용 (SMTP/POP3)

iptables -A INPUT -p TCP ! --sport 0:1024 -dport 25 -m state --state NEW -j ACCEPT : 출발지 port가 25인 TCP 프로토콜을 이용하는 트래픽에 대해 허용하고, 상태 추적 기능을 이용한다.

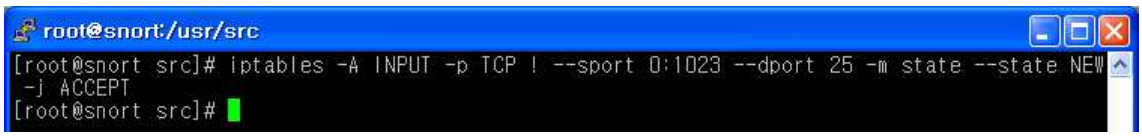


그림 3-30 메일 서비스 허용 (SMTP/POP3)

SMTP 는 25/tcp 를 통해 서비스가 제공되는데, 만약 원격지에서 보내는 메일서버(SMTP) 용도로 허용하여 메일을 보낼 경우에는 외부에서 서버의 25/tcp 로 향하는 트래픽을 허용해 주어야 한다. 이때 메일을 받은 서버는 임시로 큐에 저장했다가 외부로 메일을 발송하게 되는데, 이 경우 내부에서 외부로의 접속이므로 이는 별도의 룰 설정 없이도 허용하게 된다. 위 그림은 외부에서 오는 메일을 받을 수 있도록 25/tcp를 허용한 예인데, 소스포트에서 1024:65535 대신 ! --sport 0:1023 으로 하였는데, 이는 0부터 1023까지가 아니므로 1024:65535 와 동일한 의미가 되는 것이다.

iptables -A INPUT -p TCP --sport 1024: --dport 110 -m state --state NEW -j ACCEPT : Destination port가 110인 TCP 프로토콜을 이용한 트래픽을 허용하고 상태 추적 기능을 사용한다.

그림 3-31 110번 포트 트래픽 허용

pop3 는 110/tcp를 통해 서비스를 제공하는데, 만약 pop3 서비스를 제공한다면 외부에서의 110/tcp 접속도 허용하여야 할 것이다. 위 룰은 소스 ip 에 대한 언급은 없었으므로 모든 ip에 대한 허용이 되고 소스포트는 1204: 이후의 포트이고 목적지 포트는 110/tcp인 패킷을 허용하는 룰이다. 물론 요청에 대한 응답은 상태추적 및 OUPUT 의 기본 정책이 허용이므로 별도로 언급하지 않아도 된다. 만약 특정한 ip 또는 ip 대역에서만 pop3/tcp를 허용한다면 --sport 앞에 "-s 192.168.1.0/24" 와 같이 언급하면 된다.

⑩ FTP 서비스 허용

iptables -A INPUT -p TCP --sport 1024: --dport 21 -m state --state NEW -j ACCEPT : 목적지 포트가 21인 TCP 프로토콜을 사용하는 트래픽에 대해 허용하고, 상태 추적 기능을 이용한다.

그림 3-32 FTP 서비스 허용

FTP 서비스는 다른 서비스와 달리 2가지 모드를 사용하고 2가지 포트를 사용한다. 흔히 FTP 서비스가 사용하는 포트는 21/tcp라고 알고 있으나 21/tcp 외에 추가적으로 다른 포

트도 사용한다. 이때 사용하는 포트는 어떤 모드를 사용하는가에 따라 다른데, 이를테면 Active 모드의 경우 20/tcp를 사용하고, Passive 모드의 경우 1024 이후의 임의의 포트가 사용된다. 따라서 각각의 모드에 따라 룰을 따로따로 설정해 보면 매우 복잡한데, 통합적으로 아래와 같은 하나의 룰로 FTP 서비스에 대한 제어를 할 수 있다. 즉, 두 번째 사용되는 포트는 상태추적과 OUTPUT 의 기본정책이 허용됨에 따라 별도로 설정하지 않아도 되는 것이다.

3.3 Log Server 구축 과정

3.3.1로그를 원격지로 보내기 위한 클라이언트 구축

#vi /etc/sysconfig/network

먼저 hostname을 변경해 준다. 변경을 하지 않을 경우 원격지로 로그를 보내지 못한다.

그림 3-33 hostname을 변경

#vi /etc/hosts

원격지 IP주소와 hostname을 입력한다.

그림 3-34 원격지 IP주소와 hostname을 입력

#vi /etc/syslog.conf

예) kern.* /dev/console

kern.* @logserver.ac.kr /원격지hostname

사용자가 원하는 로그를 위 예처럼 설정해 주면 된다. 만약 모든 로그들을 보내고 싶으면 맨 아래 마지막 줄에 *.* @logserver.ac.kr 을 입력하면 된다.

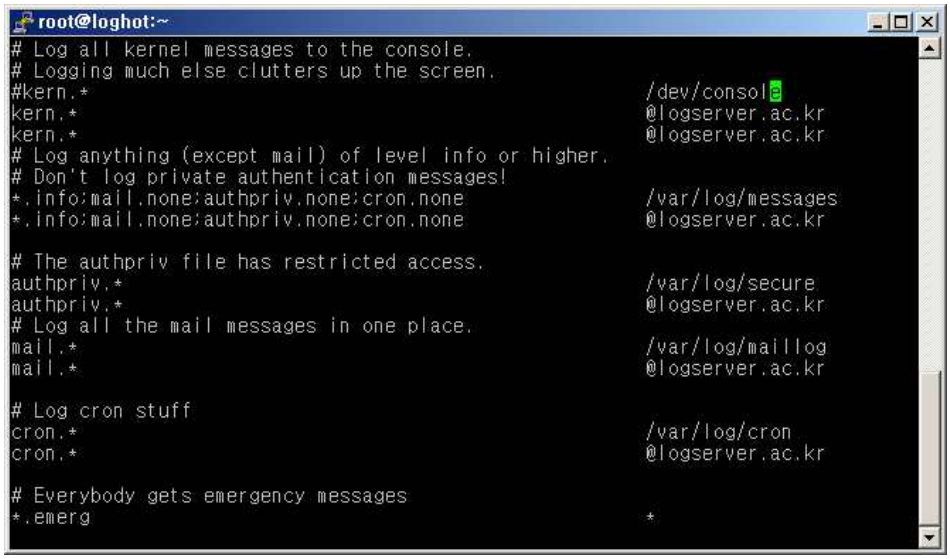


그림 3-35 모든 로그 전송

#ps -ef | grep syslogd　　/ 프로세스 확인
#kill -9 pid　　/ 실행되고 있는 프로세스를 정지
#/etc/rc.d/init.d/syslog restart　　/ syslogd 재시작
syslog.conf 설정파일은 수정한 뒤 꼭 재시작을 하도록 합니다.

만약 실패가 뜰 경우 한 번 더 재시작을 실행 주도록 합니다.

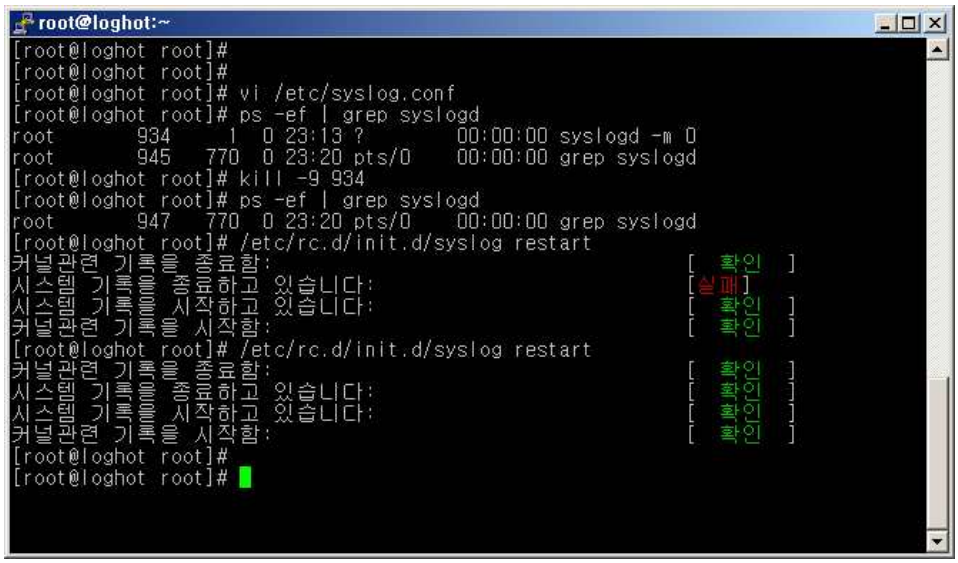


그림 3-36 restart

3.3.2 로그를 받기 위한 원격로그서버 구축

#vi /etc/sysconfig/network
클라이언트와 마찬가지로 hostname을 변경해준다.

그림 3-37 hostname 변경

#ps -ef | grep syslogd
#kill -9 pid
#vi /etc/sysconfig/syslog
원격로그의 허용 을 하기위해서 아래 그림 항목 SYSLOD_OPTIONS=" -m 0 " 이면
“ -r -m 0 "으로 바꿔준 뒤 데몬을 재시작 해준다.

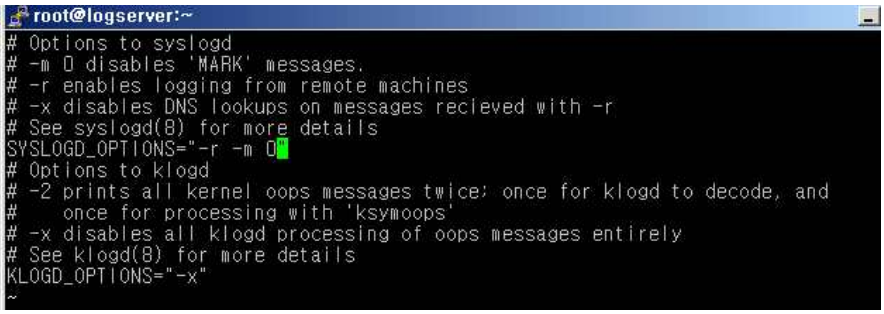


그림 3-38 데몬 재시작

로그 원격 실시간 저장 확인
클라이언트 IP 61.81.108.52 원격접속유저 IP 61.81.108.47 로 클라이언트에 접속한 로그들이 원격 로그 서버에 전송된 내용들입니다.

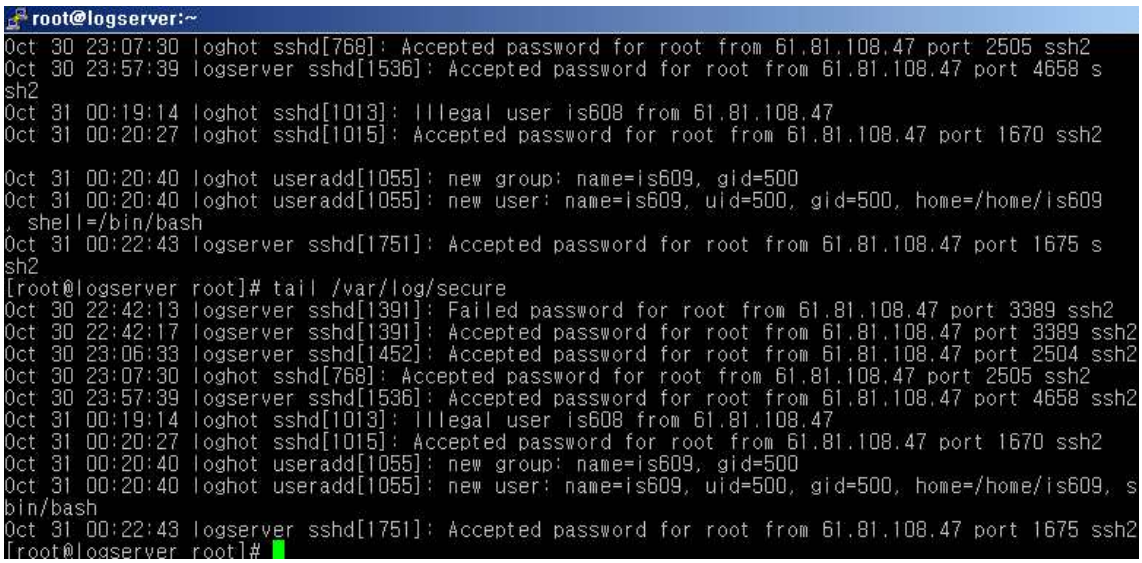


그림 3-39 원격 로그 서버에 전송된 내용

3.4 IDS 구축과정

snort를 설치하기 전에 snort를 웹페이지에서 볼 수 있도록 A.P.M 즉 웹서버를 설치한다. 이 부분은 snort의 로그들을 database에 저장하기 위해서 DB를 설치과정이다. snort가 지원하는 DB종류는 많지만 가장 많이 사용되는 MySQL을 사용해 DB를 사용하였다. RPM 패키지 버전도 있지만 우리는 직접 소스를 다운받아 설치를 하였다. 물론 설치가 복잡하더라도 사용자가 원하는 옵션에 맞게 컴파일 할 수 있다.

3.4.1 MySQL

3.4.1.1 MySQL 다운받기

http://www.mysql.com

http://www.superuser.co.kr

3.4.1.2 MySQL 설치 과정

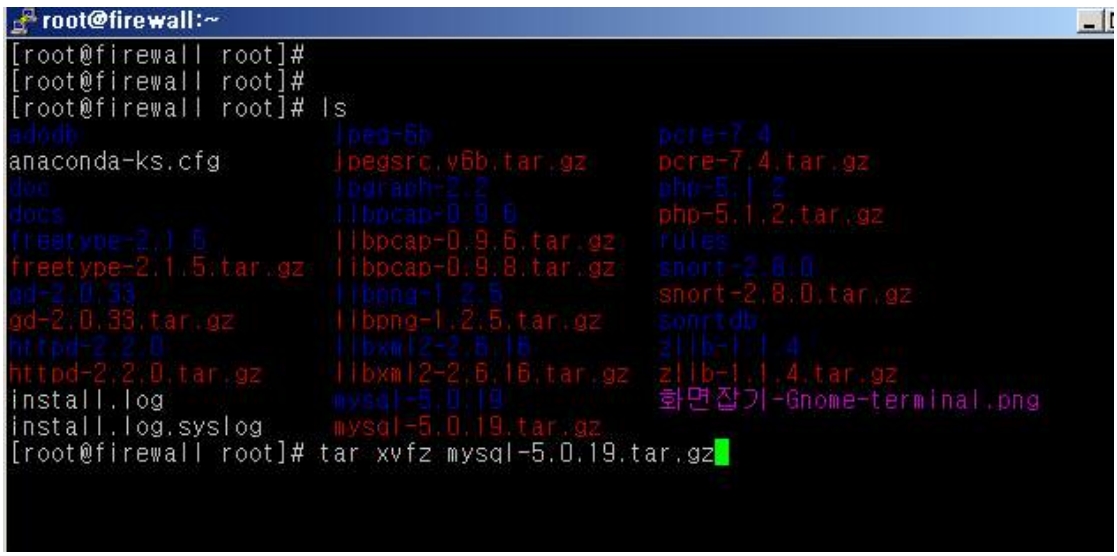


그림 3-40 MySQL 설치과정

#tar xvfz mysql-5.0.19.tar.gz 입력해 압축을 해제한다.

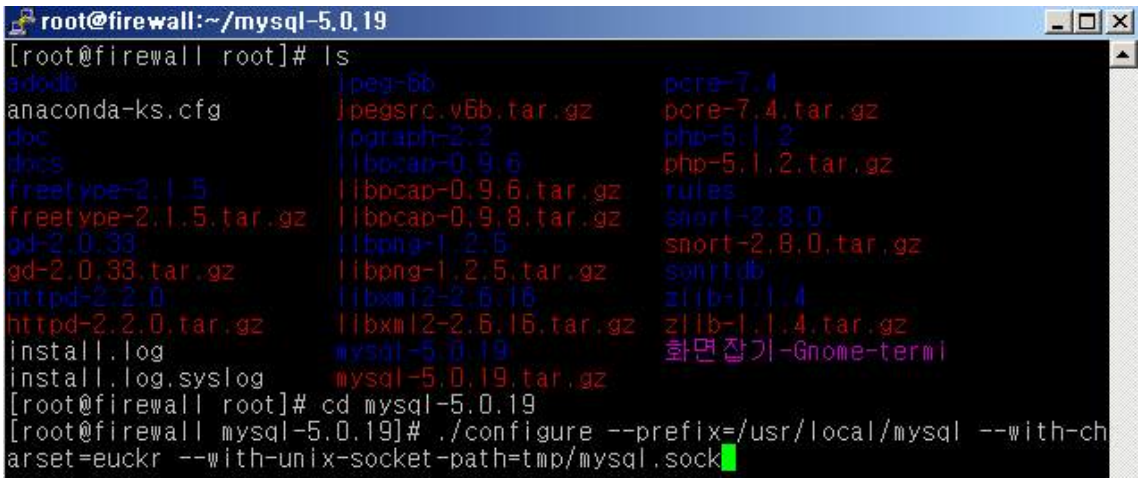


그림 3-41 압축 해제

위 그림 처럼 아래의 명령어를 입력한 뒤 컴파일 하도록 한다.

#cd mysql-5.0.19

#./configure --prefix=/usr/local/mysql --with-charset=euckr
--with-unix-socket-path=/tmp/mysql.sock

configure에 사용된 옵션

--prefix=/usr/local/mysql : MySQL이 설치될 경로

--with-charset=euckr : DB에서의 한글데이터 정렬을 위해 한글 문자를 사용하도록 설정

--with-unix-socket-path=/tmp/mysql.sock : Socket과 관련된 Path를 재설정

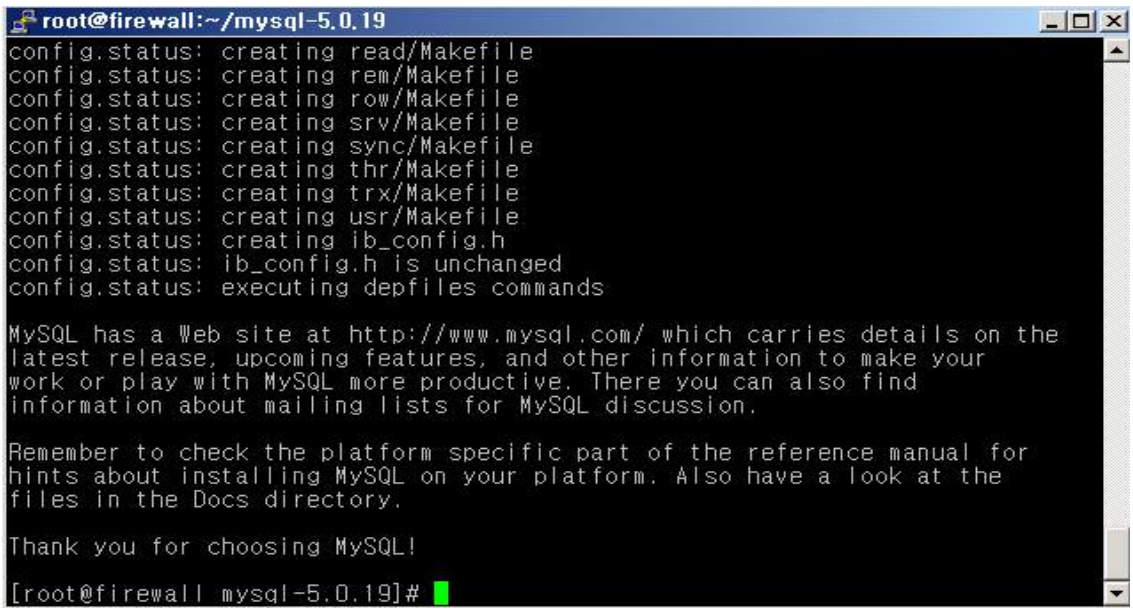


그림 3-42 make && make install

configure가 끝났으면 make && make install을 실행 한다.

이제 컴파일이 끝났으면 MySQL 초기 DB를 생성해 주도록 하자. 우선 아래 그림 과 같이 vi명령어를 이용해서 그림 4-5에 처럼 /usr/local/mysql/lib/mysql을 입력한 다음 저장 하고 나온뒤 ldconfig를 입력해 저장한 내용을 새롭게 읽게 한다.

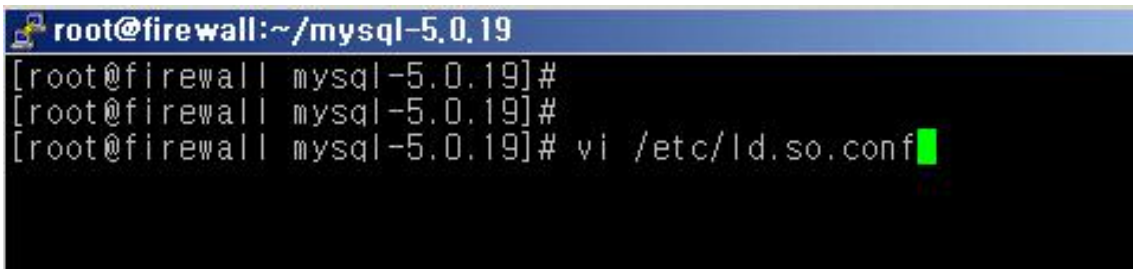


그림 3-43 라이브러리

그림 3-44 ldconfig

아까 저장한 내용을 #cat /etc/.ld.so.conf를 입력해 확인 후 ldconfig 을 입력해 저장된 내용들을 읽어 들입니다.

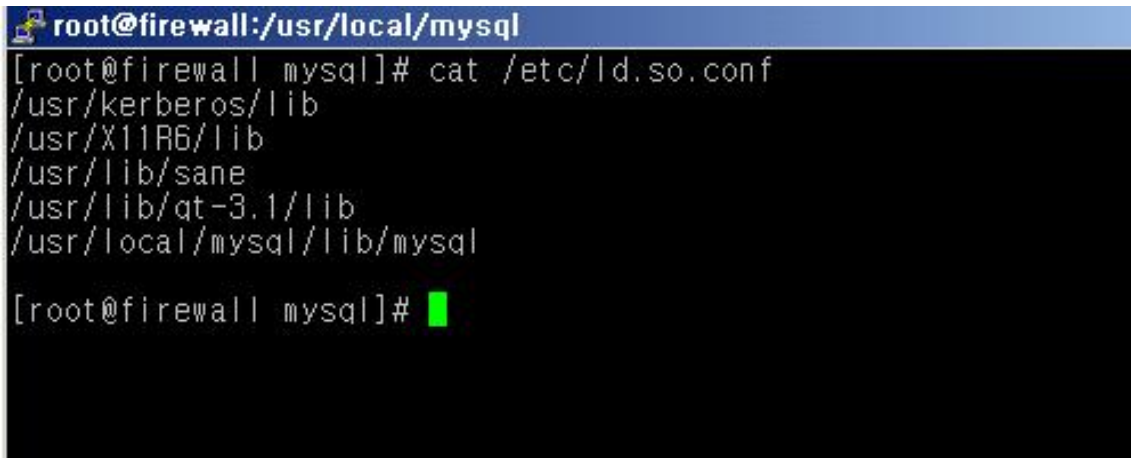


그림 3-45 입력 확인

다음 아래 명령어들을 입력해 초기화 스크립트 복사 및 초기화 DB를 생성해 주도록 한다.
여기서 주의할점 mysql_install_db 스크립트는 반드시 한번만 실행하도록 해야 합니다.
왜냐하면 여러번 실행할 경우 서버가 정상적으로 실행되지 않기 때문입니다.

#cd /usr/local/mysql/share/mysql
#cp mysql.sever /etc/init.d/mysql //:초기화 스크립트 복사
#/usr/local/mysql/bin/mysql_install_db //초기DB 생성

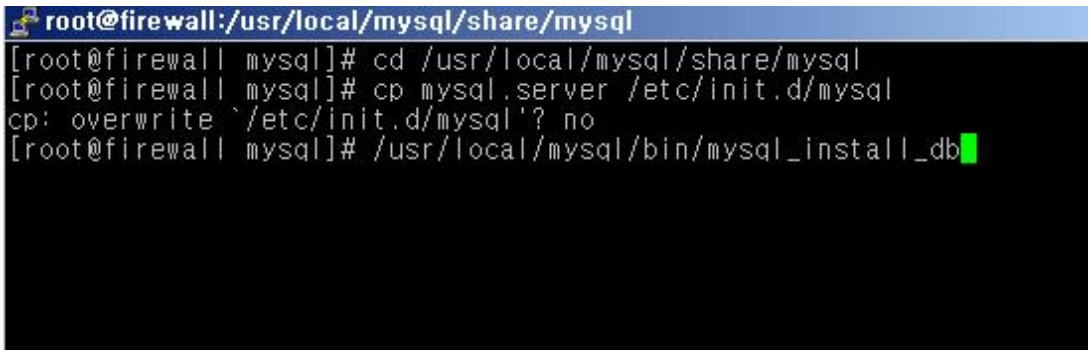


그림 3-46 DB를 생성

이제 아래 명어들을 입력해 MySQL을 구동시킨뒤 ps 명령어로 프로세스를 확입합니다.

#chkconfig mysql on /MySQL을 부팅과 함께 실행
#/etc/init.d/mysql on /초기화 스크립트로 MySLQ 구동
#./bin/mysqld_safe & /MySQL 서버를 백그라운드로 실행
#ps -ef | grep mysqld /MySQL 프로세스 확인

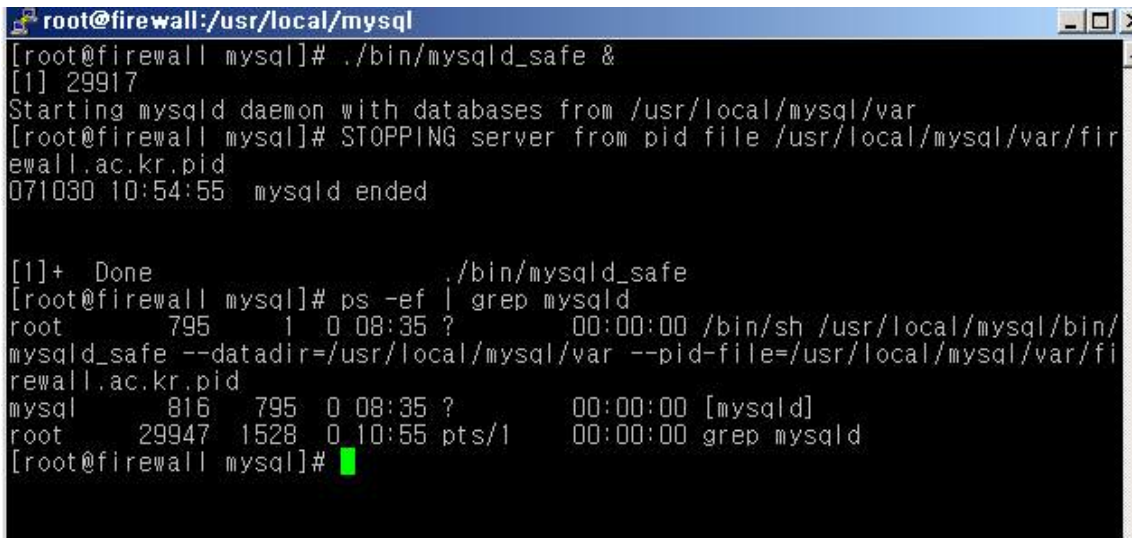


그림 3-47 프로세스 확인

3.4.2 Apache

3.4.2.1 Apache 다운받기

http://www.apache.org

http://www.superuser.ac.kr

3.4.2.2 Apache 설치과정

#tar xvfz httpd-2.2.0.tar.gz를 입력해 압축을 해제한다.

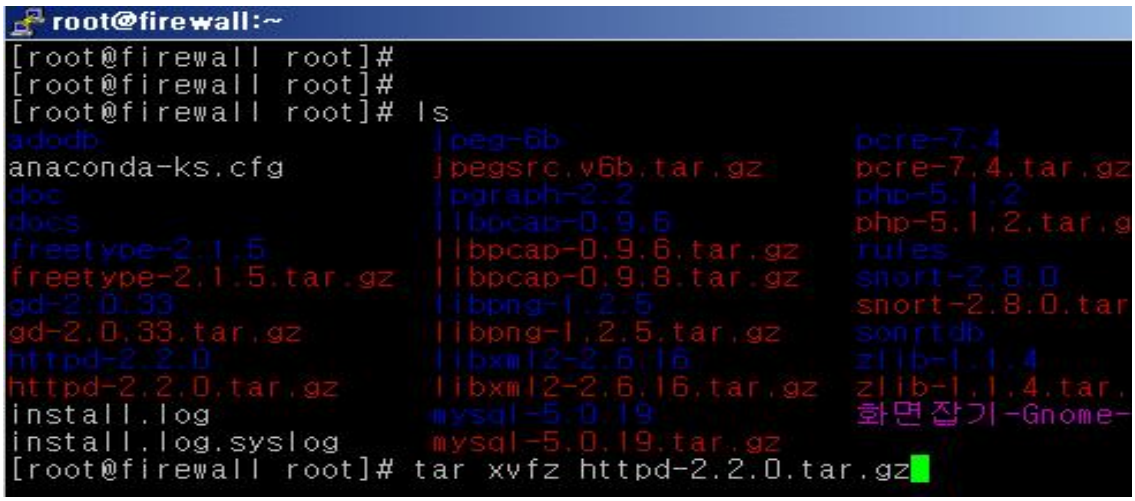


그림 3-48 압축 해제

아래 명령어를 입력하여 컴파일 하도록 한다.

#cd httpd-2.2.0

#./configure --prefix=/usr/local/httpd2 --enable-modules=so --enable-mods-shared=all --with-suexec-caller=nobody --enable-ssl --with-ssl

configure에 사용된 옵션

--prefix=/usr/local/httpd2 /Apache가 설치된 디렉토리

--enable-modules=so /Apache가 사용할 모듈 선택 및 DSO(Dynamic Shared object) 지원하기위한 설정

--enable-mods-shared=all /선택한 모듈로 처리 가능한 모듈은 모두 SO(Shared Obsect)로 처리

--with-suexec-caller=nobody /Apache 실행권한은 대부분 nobody이므로 꼭 설정해야 한다.

--enable-ssl --with-ssl

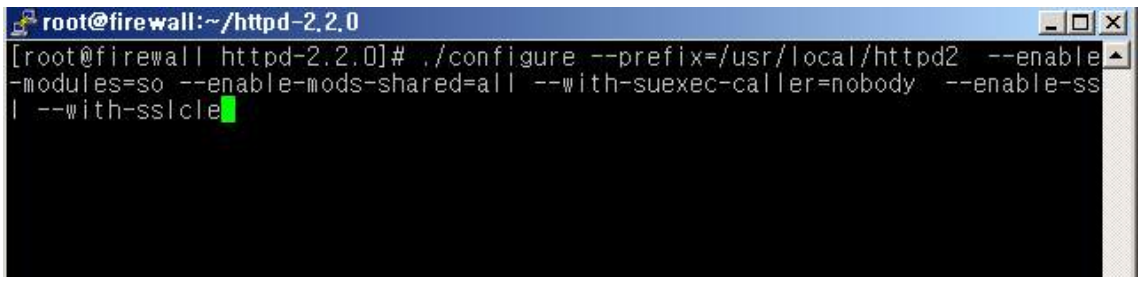


그림 3-49 configure에 사용된 옵션

configure를 실행한 다음 make && make install을 실행한다.

vi 를 이용하여 /etc/ld.so.conf 에 /home/httpd2/lib 와 /home/httpd2/modules 를 추가한 다음 그림 4-12처럼 cat /etc/ld.so.conf 명령어로 확인 ldconfig로 저장된 내용을 읽어 들이고 초기화 스크립트를 복사 합니다.

그림 3-50 /etc/ld.so.conf에 추가

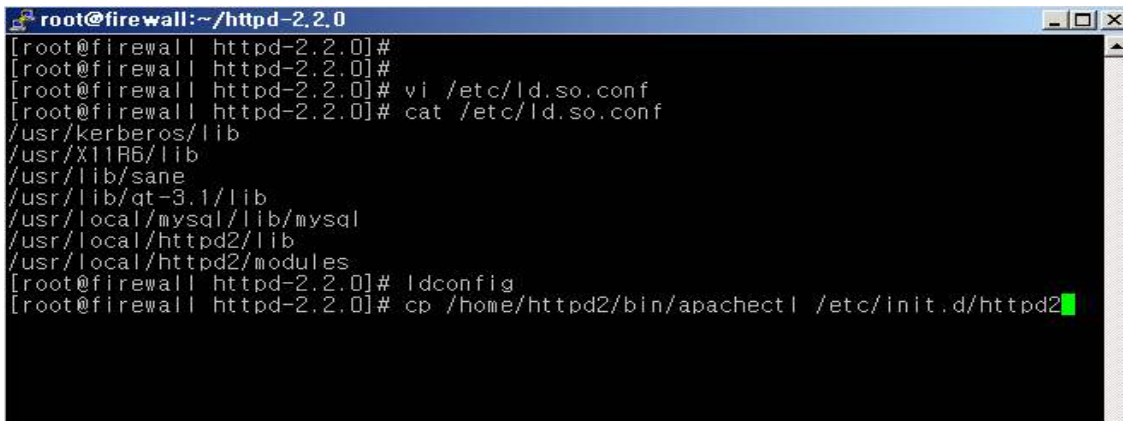


그림 3-51 ldconfig로 저장 및 초기화 스크립트 복사

이제 Apache의 모든 설치는 다 끝났습니다. Apache와 MySQL를 실행을 위해서 vi /etc/rc.d/rc.local를 입력하고 아래 그림 4-13처럼 명령어를 입력, 저장을 합니다. 이것은 부팅시 자동 실행을 위한 설정이고 수동 실행 명령어는 /etc/rc.d/init.d/httpd start 를 입력하고 실행하면 됩니다.

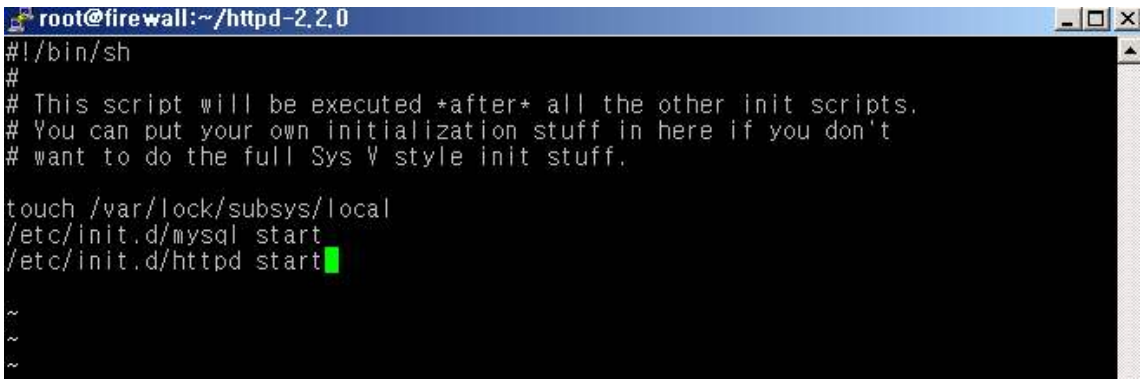


그림 3-52 /etc/rc.d/init.d/httpd start 입력하고 실행

3.4.3 PHP

3.4.3.1 PHP 다운받기

http://www.php.net

http://www.superuser.ac.kr

3.4.3.2 PHP 설치

#tar xvfz php-5.1.2.tar.gz를 입력해 압축을 해제합니다.

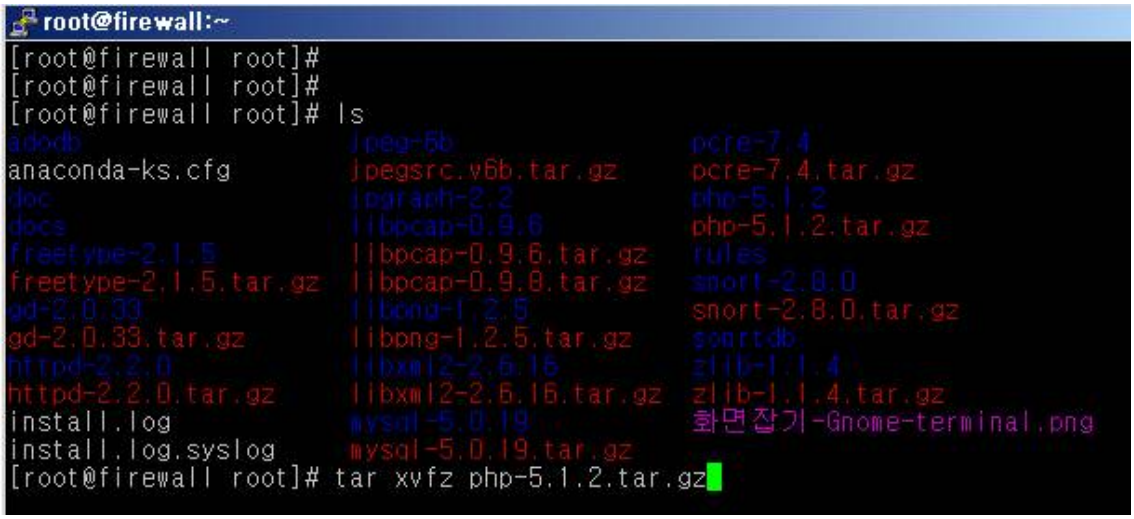


그림 3-53 압축 해제

아래 명령어를 입력하여 컴파일 하도록 한다.

#cd php-5.1.2

./configure --prefix=/usr/localphp --with-apxs2=/usr/local/php
--with-apxs2=/usr/local/httpd2/bin/apxs --with-mysql=/usr/local/mysql
--with-config-file-path=/usr/local/httpd2/conf
--with-exec-dir=/usr/local/httpd2/bin --with-zlib --with-gd --with-png
--with-jpeg-dir --with-xml --with-mod-charset --with-language=korea
--with-openssl-dir

configure에 사용된 옵션

--prefix=/usr/localphp	/PHP가 설치될 디렉토리
--with-apxs2=/usr/local/php	/--with-apxs2는 Apache daemon에 DSO 모듈로 로딩
--with-apxs2=/usr/local/httpd2/bin/apxs	되도록 하는 설정
--with-mysql=/usr/local/mysql	/MySQl과 연동하기 위한 설정
--with-config-file-path=/usr/local/httpd2/conf	/PHP.ini 경로의 위치를 설정
--with-exec-dir=/usr/local/httpd2/bin	/Apache daemon의 실행 경로를 설정
--with-zlib --with-gd --with-png	/ 이 부분은 웹페이지의 실행을 위해 지원되는 각종
--with-jpeg-dir --with-xml	라이브러리 설정 부분
--with-openssl-dir	/ openssl을 사용하기 위한 설정

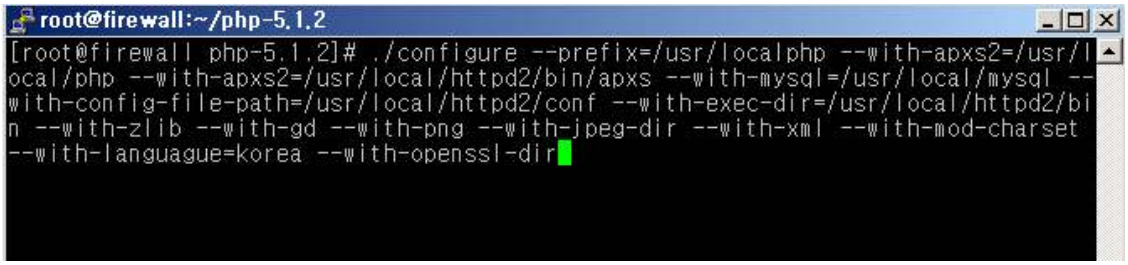


그림 3-54 configure에 사용된 옵션

configure를 실행 한 다음 make &&make install을 실행 한다.
컴파일이 다 완료가 되면 configure 설정 중 --with-config-file-path부분에서 지정한 디렉토리에 cp php.ini-dist /usr/local/httpd2/conf/php.ini 설정파일을 복사합니다.

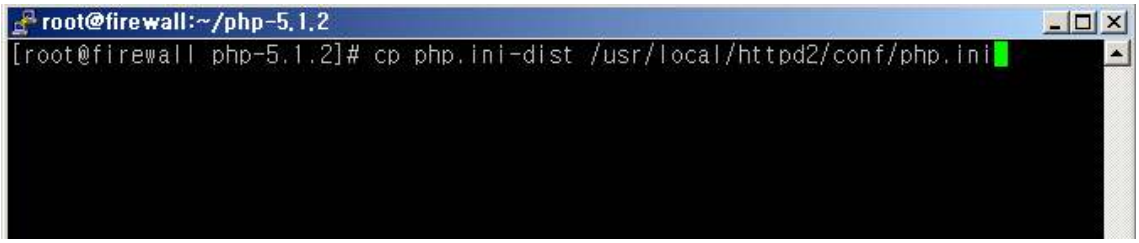


그림 3-55 설정파일 복사

웹서버 구동시 웹페이지에 소스코드만 나오는 오류가 있다 이 오류는 httpd.conf에서 358행에 아래 밑줄 친 명령어를 추가입력 하면 된다.

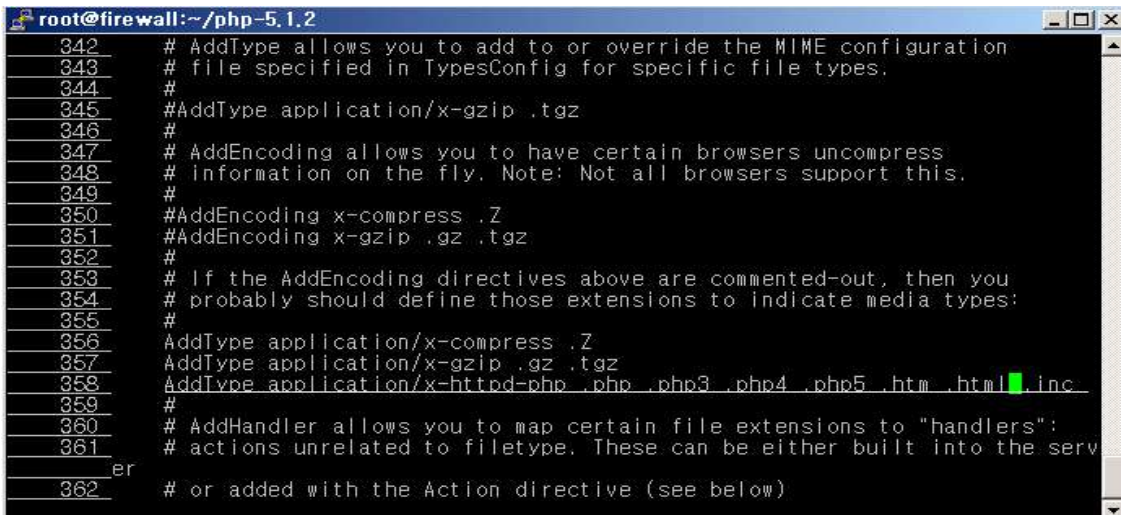


그림 3-56 명령어 추가 입력

3.4.4 Libpcap

3.4.4.1 Libpcap 다운받기

http://tcpdump.org

3.4.4.2 LIbpcap 설치

libpcap은 configure에서 설정 할 것이 없으므로 아래 명령어를 입력하고 실행한다.

#tar xvfz libpcap-0.9.8.tar.gz

#cd libpcap-0.9.8

#./configure --prefix=/usr/local/libpcap && make && make install

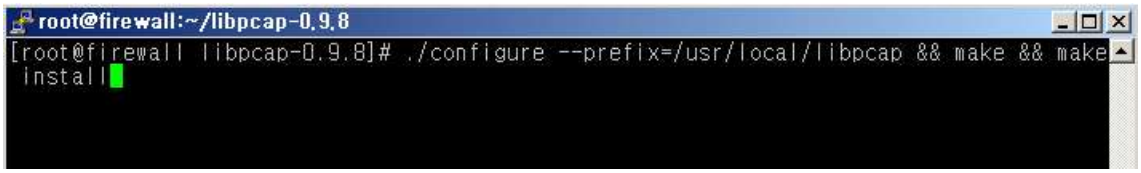


그림 3-57 명령어 입력 실행

3.4.5 pcre

3.4.5.1 pcre 다운

wgethttp://downloads.sourceforge.net/pcre/pcre-7.4.tar.gz?modtime=1190411691&big_mirror=0

3.4.5.2 pcre 설치

pcre는 configure에 설정 할 것이 없으므로 아래 명령어를 입력하고 실행한다.

#tar xvfz pcre-7.4.tar.gz

#./configure && make && make install

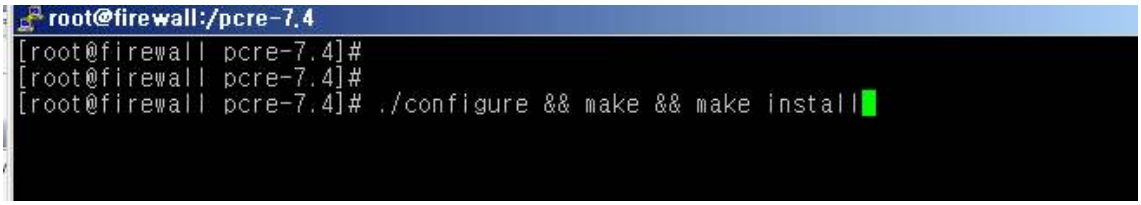


그림 3-58 명령어입력 실행

3.4.6 snort

3.4.6.1 snort 다운

http://www.snort.org

3.4.6.2 snort 설치

snort 현재 날짜로 최신 snort-2.8.0 버전을 다운 받고 압축을 풀고 configure을 실행하기 전에 --with-mysql=/usr/local/mysql 옵션을 반드시 설정해야 한다. 왜냐하면 snort와 DB연동 시 오류를 방지하기 위해서다.

#tar xvfz snort-2.8.0.tar.gz

#./configure --prefix=/usr/local/snort --with-mysql=/usr/local/mysql && make && make install

그림 3-59 옵션 설정

아래 명령어를 입력 하여 snort을 로그를 저장할 디렉토리를 생성한다.

#mkdir /var/log/snort

현재 snort를 실행하지 않았으므로 디렉토리를 확인해보면 생성된 로그가 없다.

그림 3-60 디렉터리 확인

snort를 실행해 보도록 한다

snort는 3가지 모드를 지원 한다

- sniff mode
- loging mode
- nids mode

기본적인 sniff mode를 실행해 보자

#snort -v

실행 결과 아래와 같이 TCP/IP

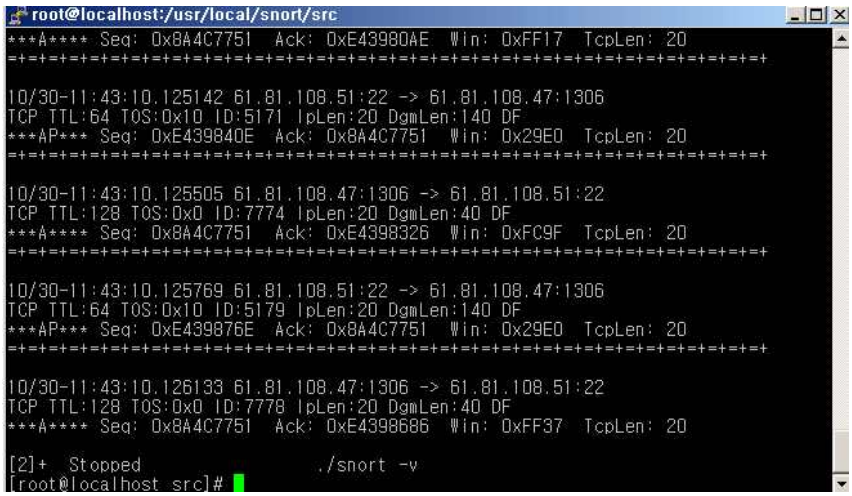


그림 3-61 snort -v

snort.conf DB 설정

output database: log, mysql, user=snort password=tmshxm1 dbname=snortdb host=localhost

output database: alert, mysql, user=snort password=tmshxm1 dbname=snortdb host=localhost

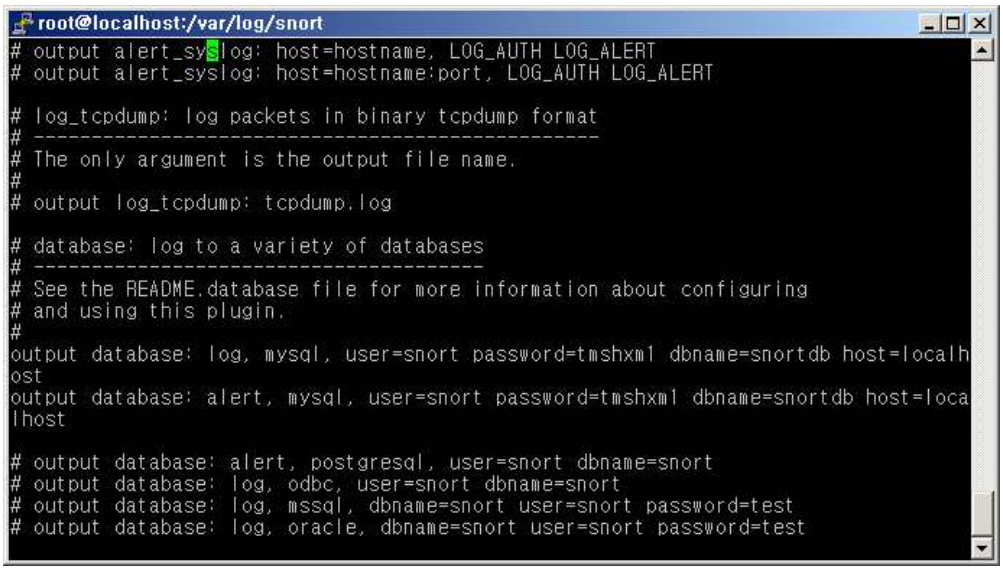


그림 3-62 snort.conf DB 설정

snort와db 연동

/usr/local/mysql/bin 디렉토리로 이동

./mysql -u root -p mysql 로 접속후 use mysql입력 mysqlDB로 이동

처음 접속을 할 경우 password가 없으므로 password를 설정해준다

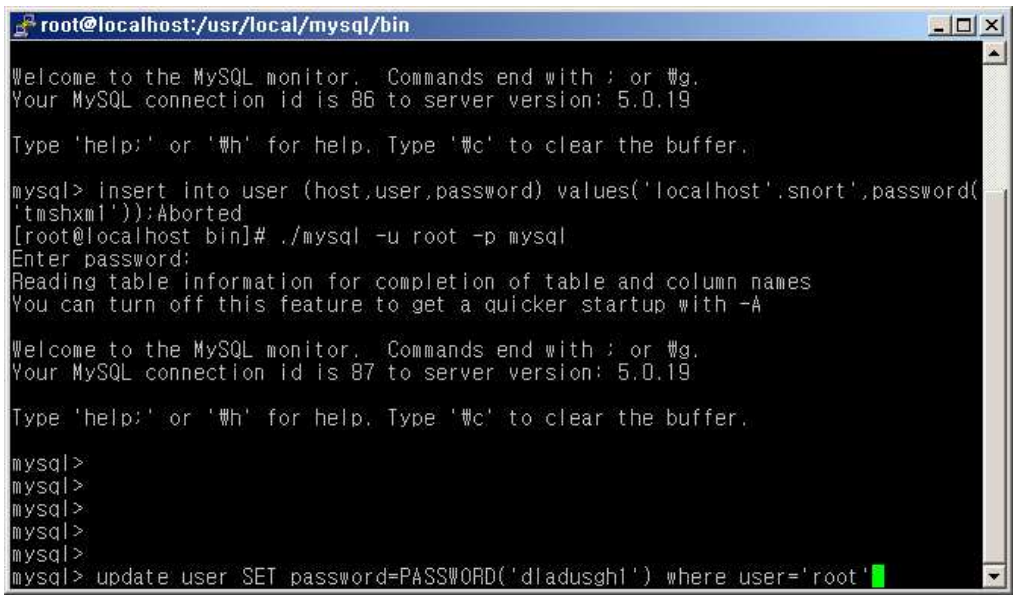


그림 3-63 snort와db 연동

snort 라는 DB생성

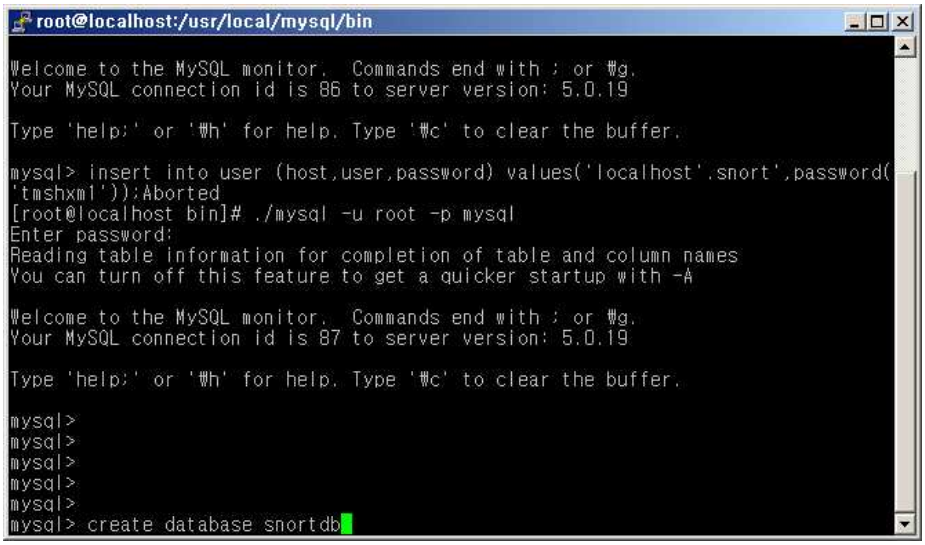


그림 3-64 DB생성

snort라는 사용자를 생성
주의할점 명령어들을 실행한 후 꼭 flush privileges; 를 입력해야 저장이 된다
use mysql DB에 이동후 아래 명령어를 실행
insert into user (host,user,password) values('localhost'.snort',password('tmshxm1'));

그림 3-65 사용자를 생성

유저와 password가 잘 생성 되었는지 확인해보자
select host,user,password from user를 실행 하면 아래와 같이 출력이 된다.

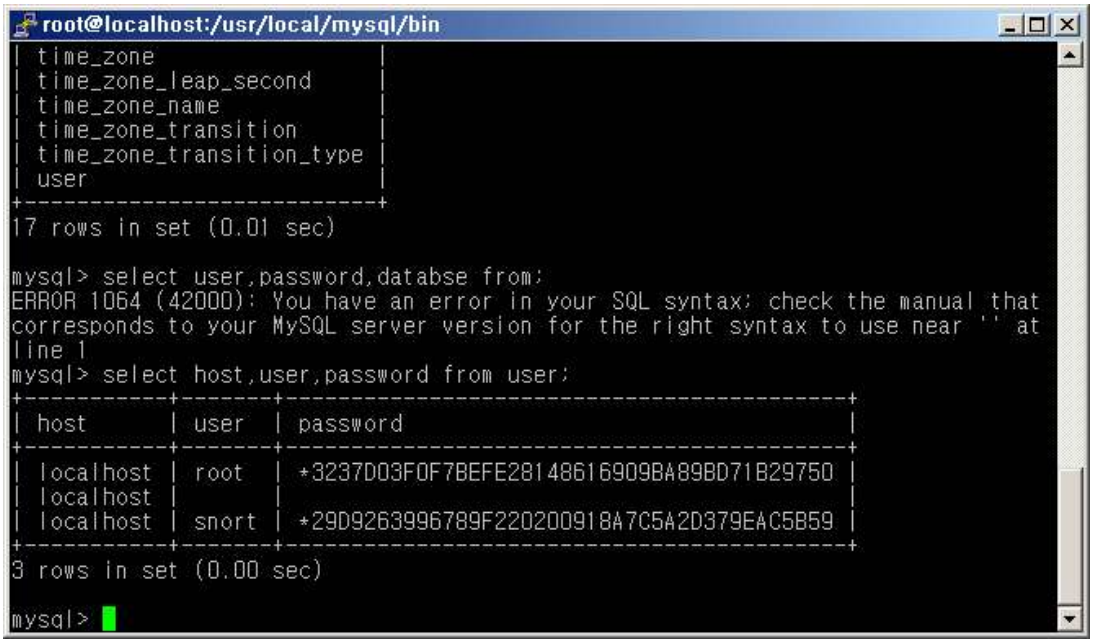


그림 3-66 password 생성 확인

root로 mysql에 접속 use mysql DB로 이동 아래명령어를 실행한다
grant SELECT,INSERT,DELETE,UPDATE,ALTER on snortdb.* to snort@localhost
snort라는 유저를 snortdb에 권한을 부여한다.

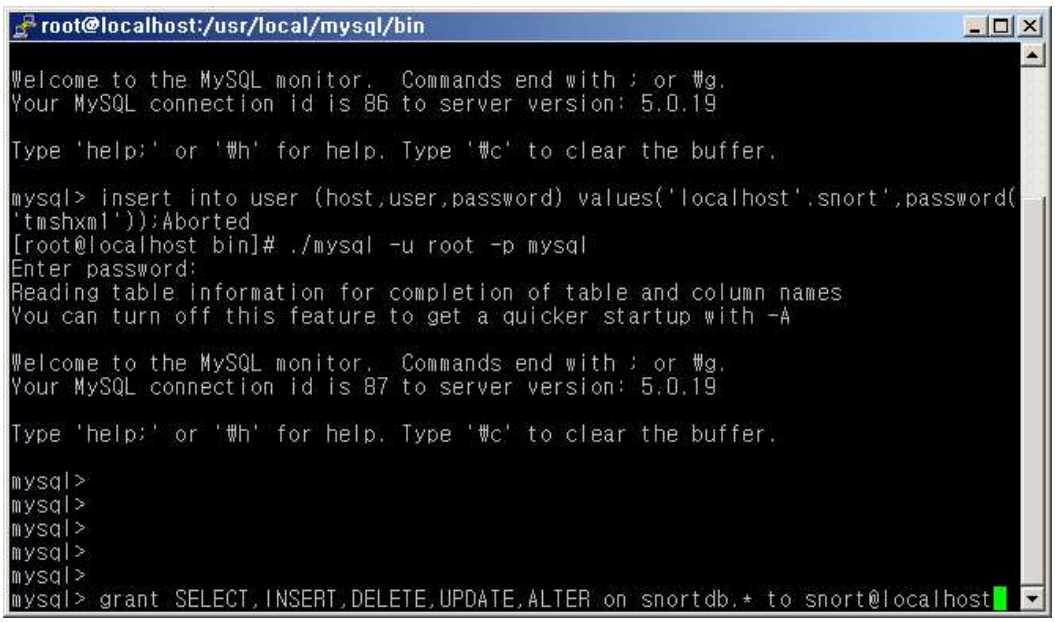


그림 3-67 유저를 snortdb에 권한을 부여

snort 유저의 snortdb에 필요한 table을 생성한다.
만약 이곳에 create_mysql 파일이 없을 경우 find / -name create_mysql 실행하여 절대 경로를 입력해준다.

그림 3-68 snotdb에 필요한 table 생성

다음과 같이 show table을 실행해 아래와 같은 테이블이 보이는지 확인한다.

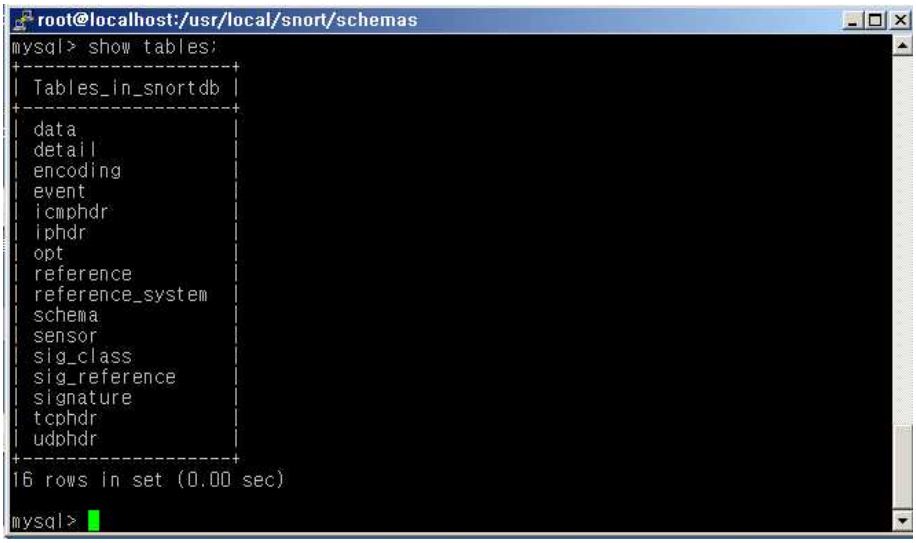


그림 3-69 table 생성 확인

./mysql -u snort -p snortdb </usr/local/httpd2/htdocs/acid/create_acid_tbls_mysql.sql 실행해서 aicd에서 제공하는 acid table을 추가한다.

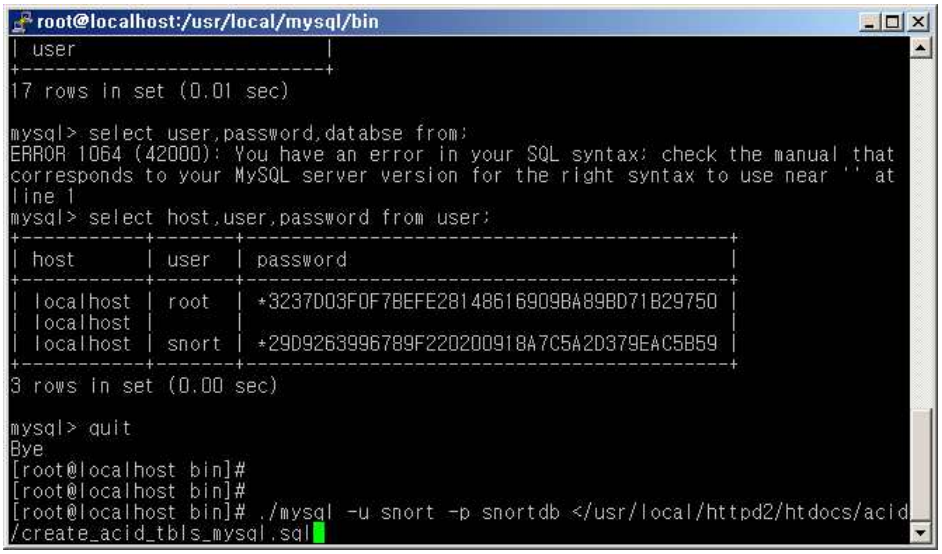


그림 3-70 acid table 추가

아래와 같이 테이블이 생성된 것을 확인할 수 있다.

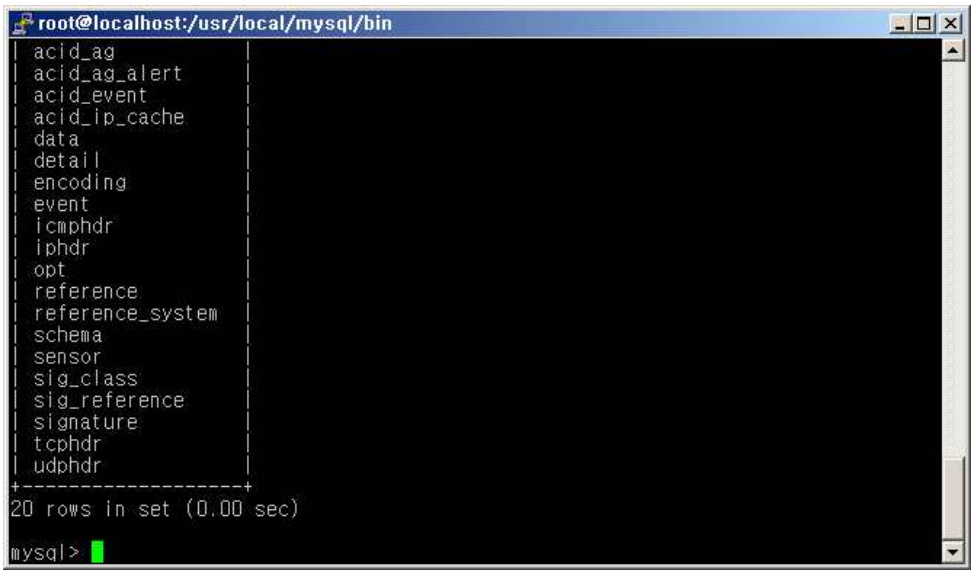


그림 3-71 table 생성 확인

acid 설치

Zlip ,libpcap MyqSQL Apache PHP이 것을 윗부분에서 설정 및 설치를 했다.

acid

http://acidlab.sourceforge.net.acid-9.6b23.tar.gz 에서 다운

/usr/local/httpd2/htdocs 에 압축을 해제 하도록 한다.

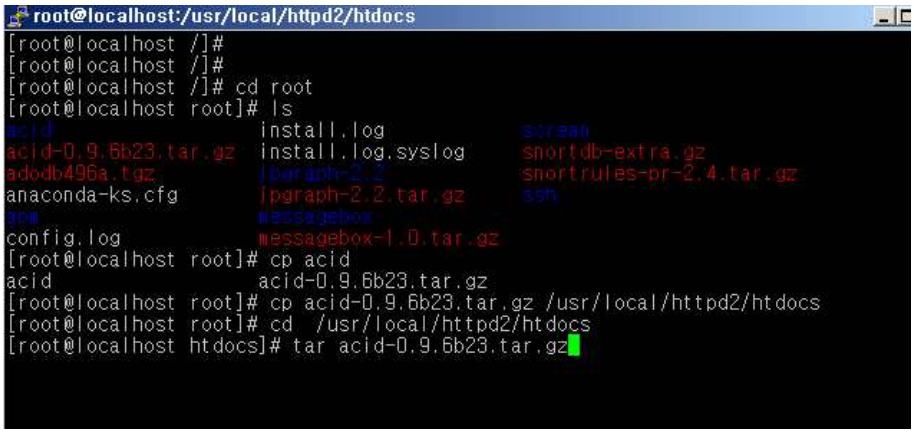


그림 3-72 acid 설치

ADODB

http://easynews.dl.sourceforge.net/sourceforge/adodb/adodb495a.tgz 다운

/usr/local/httpd2/htdocs 에 압축을 해제 하도록 한다.

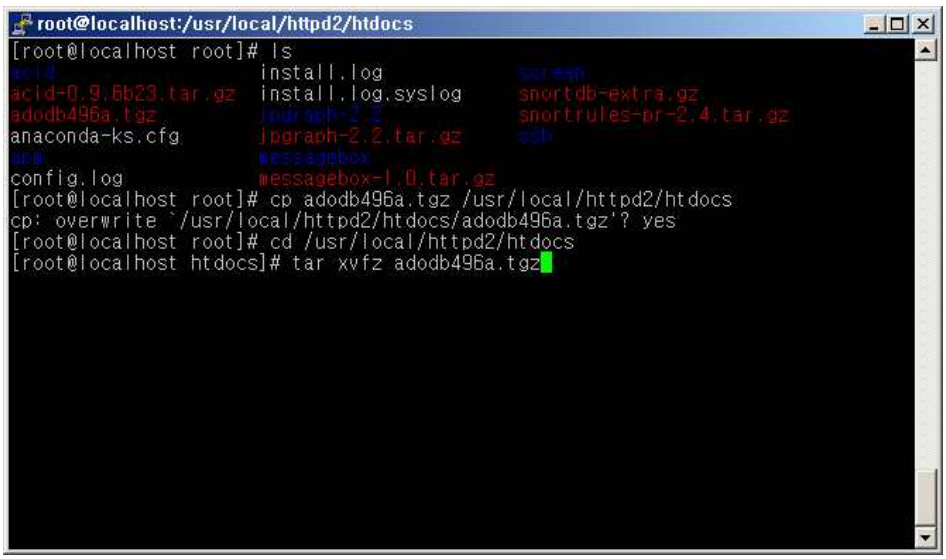


그림 3-73 압축을 해제

이제 acid 환경설정 파일을 수정해준다 /usr/local/httpd2/htdocs/acid/acid_conf.php

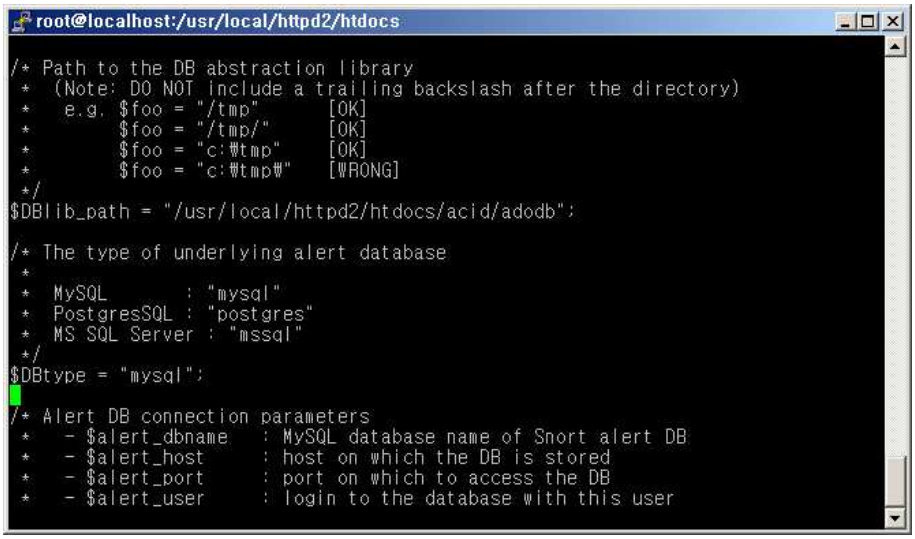


그림 3-74 acid 파일 수정

DBlib_path ="/usr/local/httpd2/htdocsacid/adodb" 추가 입력

빨간색 부분에 자신의 DB이름 DB가 사용하는 포트번호 유저 password 등을 입력한다.

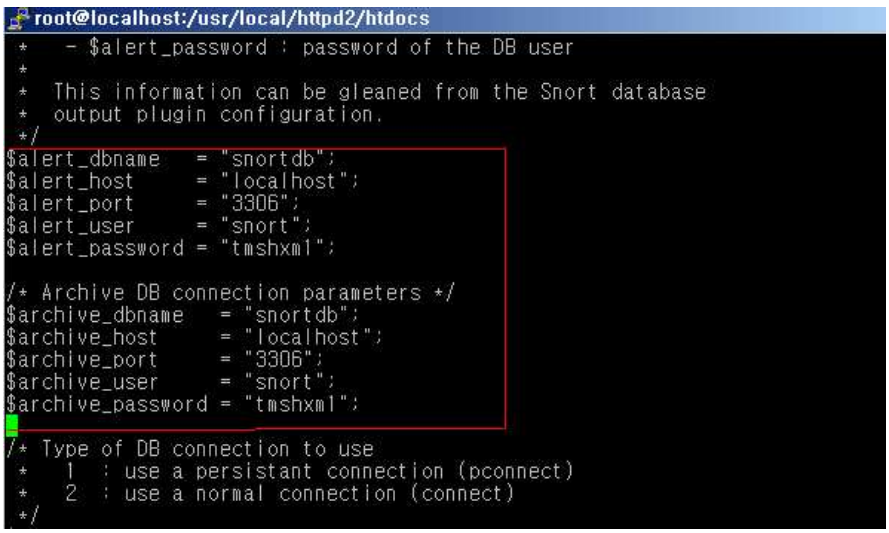


그림 3-75 DB 이름 password등 입력

이제 모든 설치는 끝났다.

http://61.81.108.51/acid/를 주소창에 입력한다.

아래와 같이 센서가 작동 하는 것을 볼 수 있다.

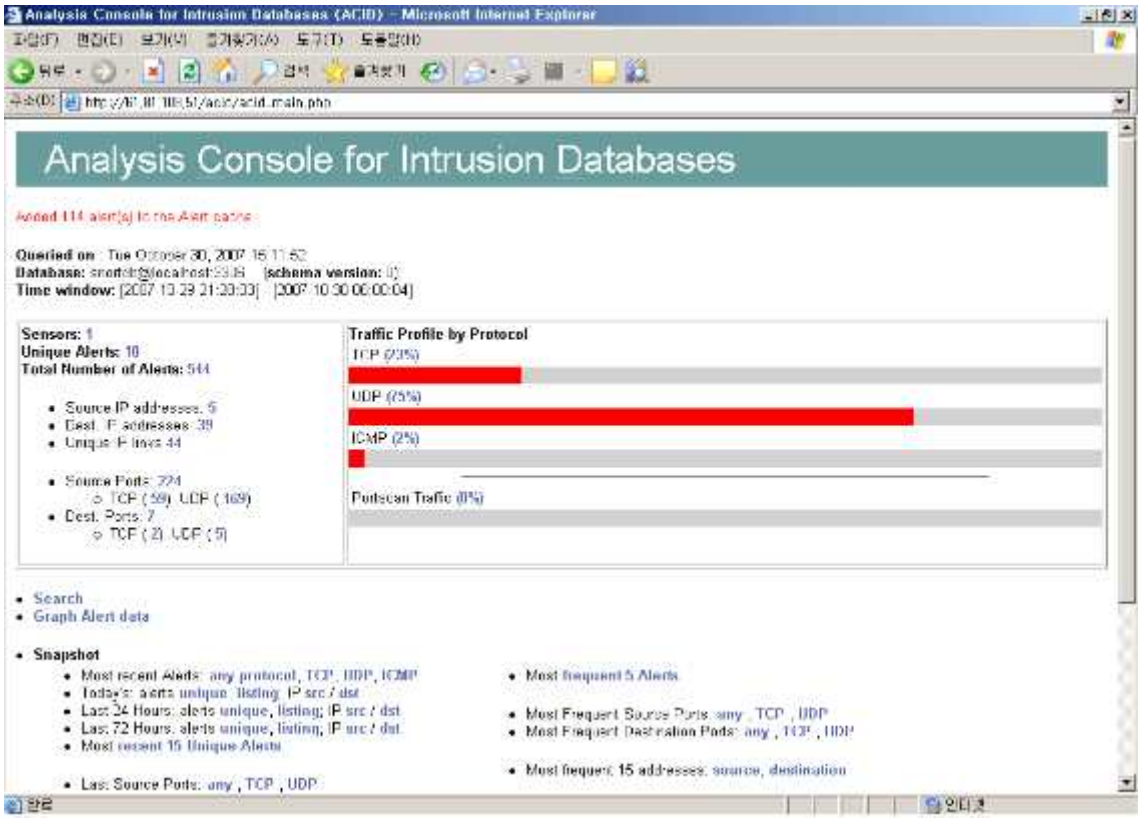


그림 3-76 센서 작동 확인

3.5 Honeypot Server 구축 과정

http://www.Microsoft.com
Virtual PC를 다운 받아 설치한다.

그림 3-77 Virtual PC 설치

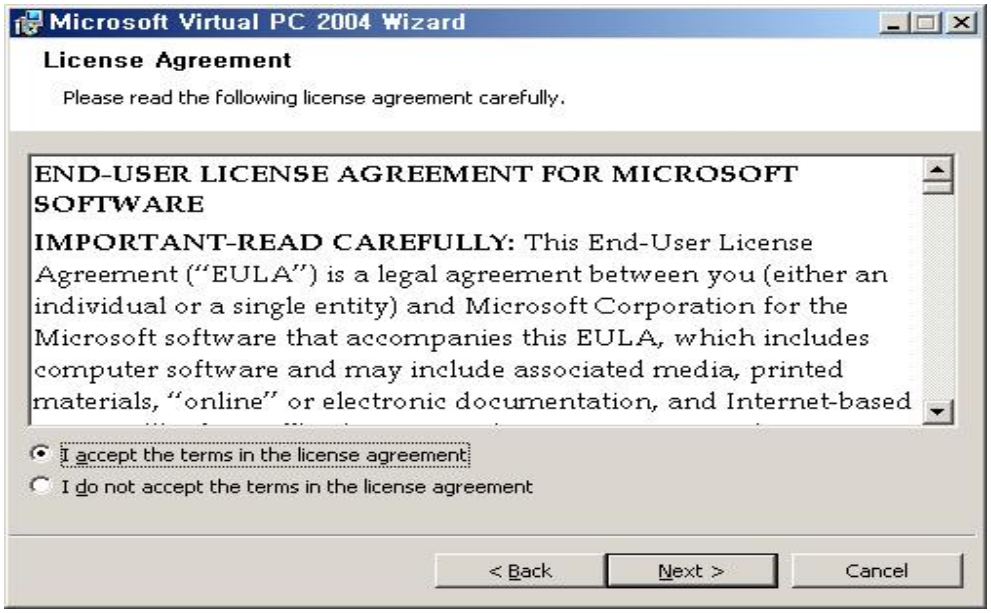


그림 3-78 Virtual PC 설치 과정

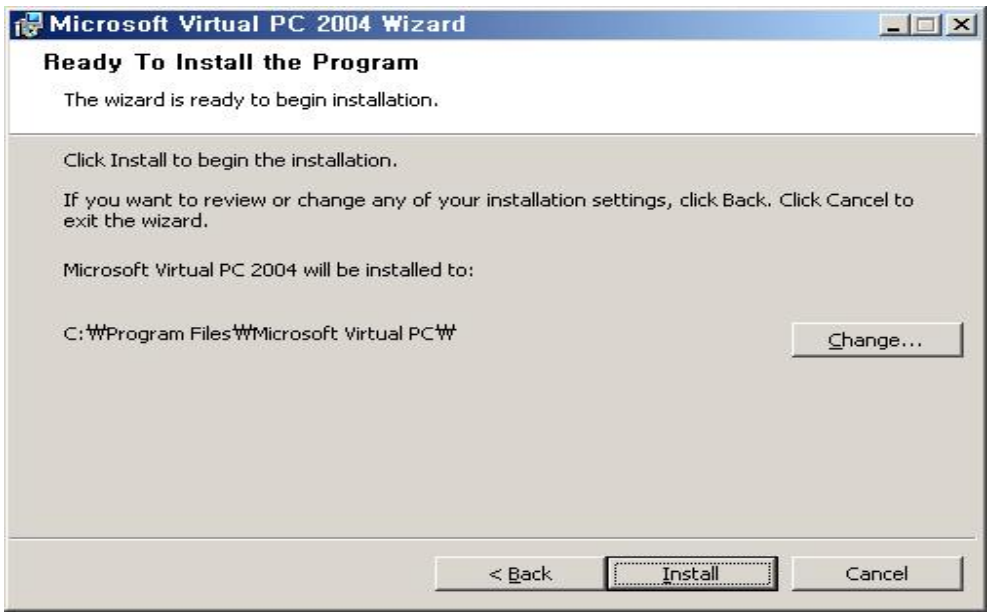


그림 3-79 install

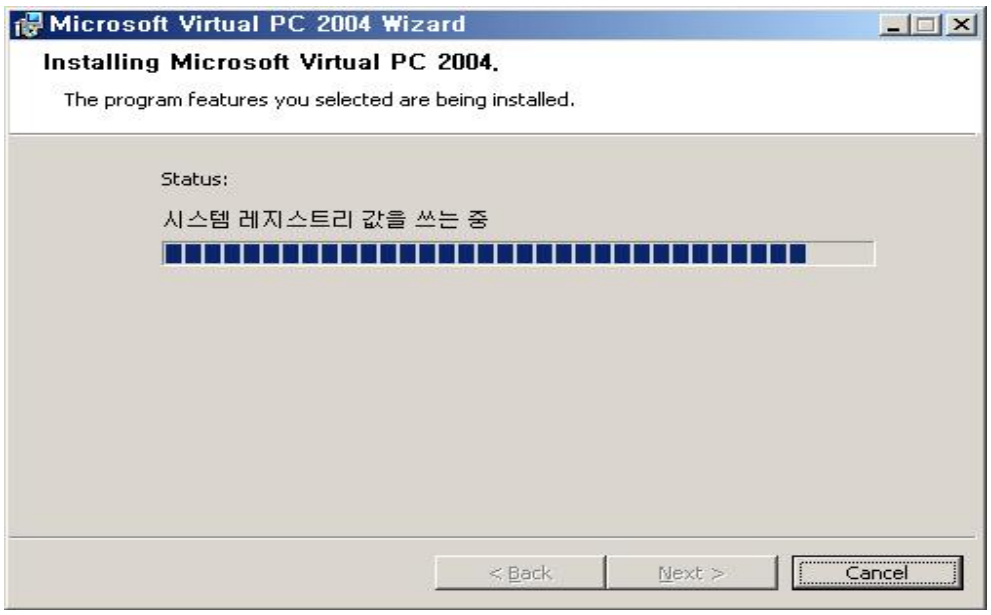


그림 3-80 Virtual PC 설치중

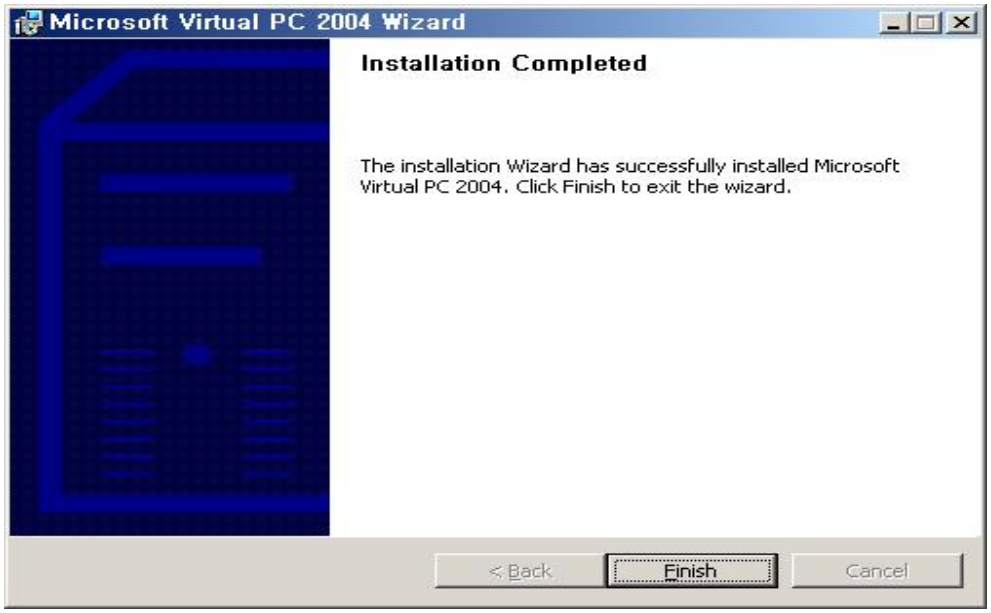


그림 3-81 Finish

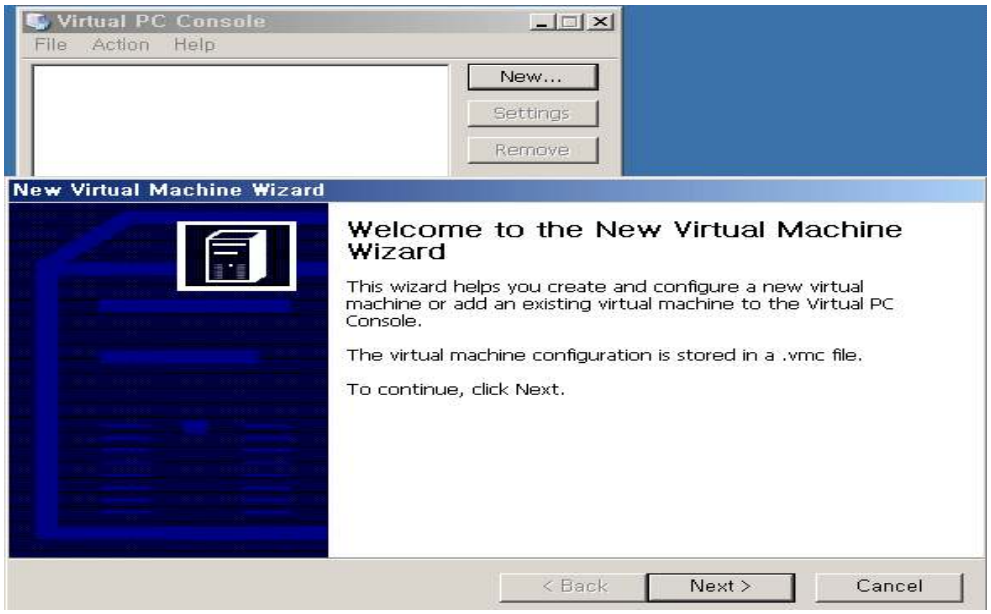


그림 3-82 Next

새로운 virtual machine을 선택 후 next

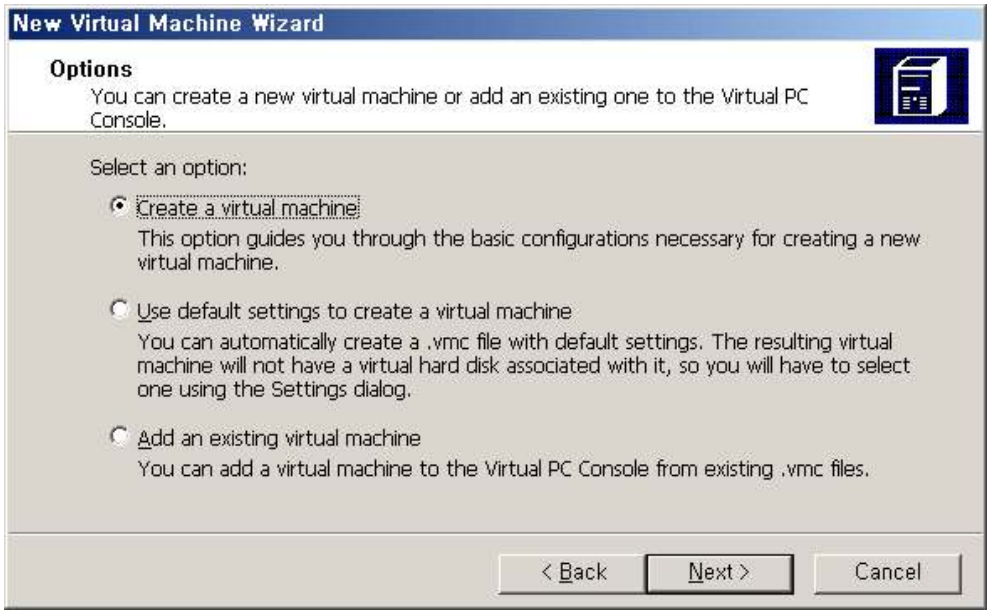


그림 3-83 Option 선택 후 Next

자신이 설치할 운영체제이름으로 가상머신 이름을 설정해주도록 한다.

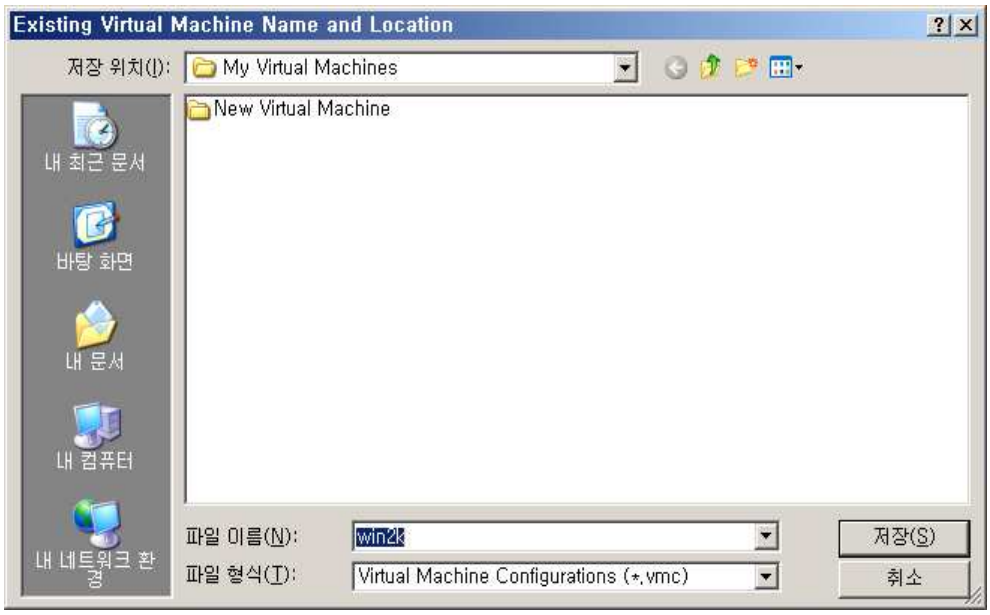


그림 3-84 운영체제 이름

설치할 운영체제를 선택한다. 만약 Unix 계열의 운영체제를 설치하길 원하면 other를 선택한다.

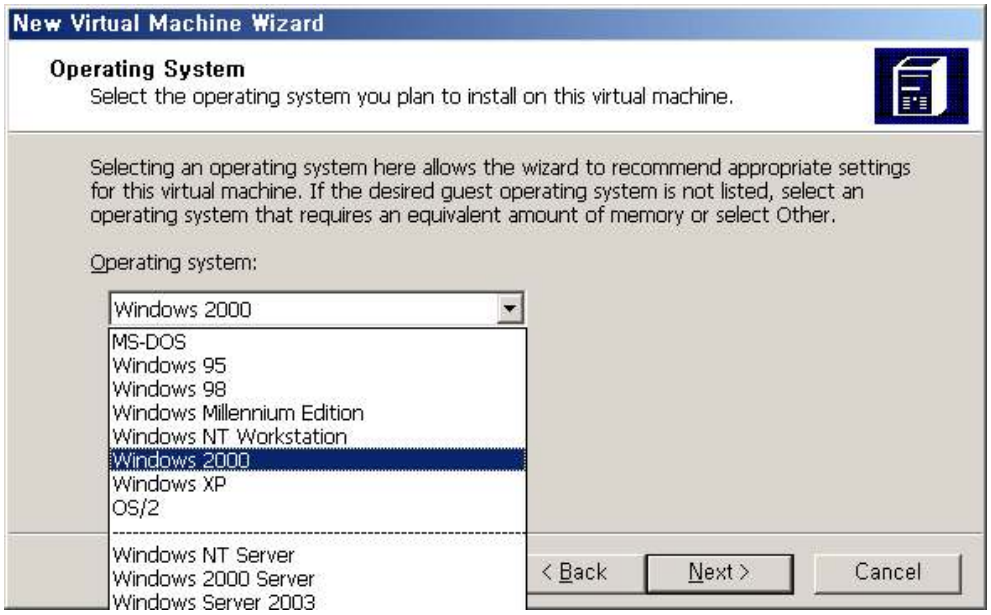


그림 3-85 설치할 운영체제 선택

메모리 설정 부분이다. PC의 메모리는 제한되어 있으므로 128MB 정도만 사용 하도록 한다.

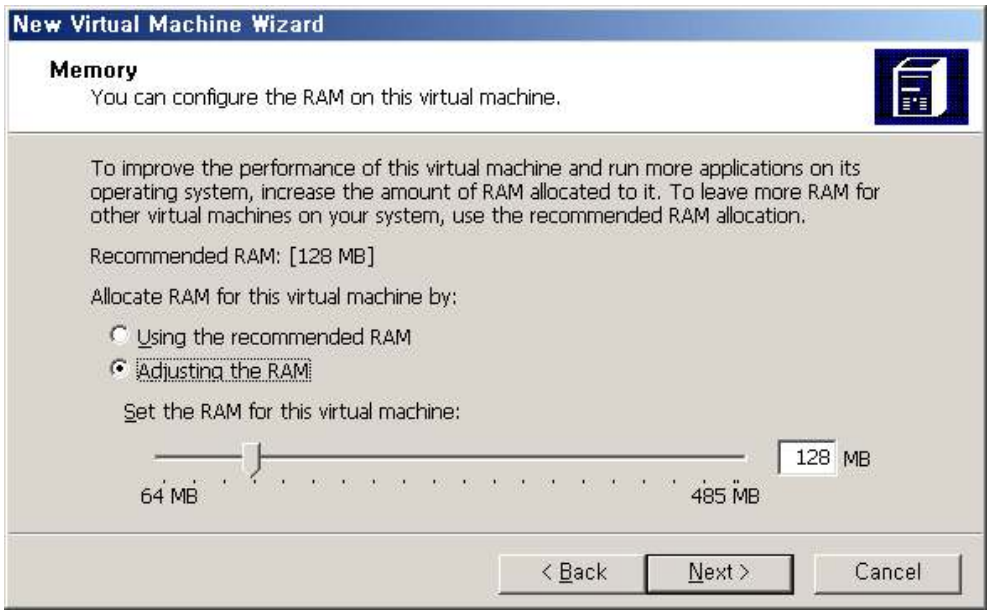


그림 3-86 메모리 설정

새로운 Virtual hard disk를 선택한다.

그림 3-87 Virtual hard disk 선택

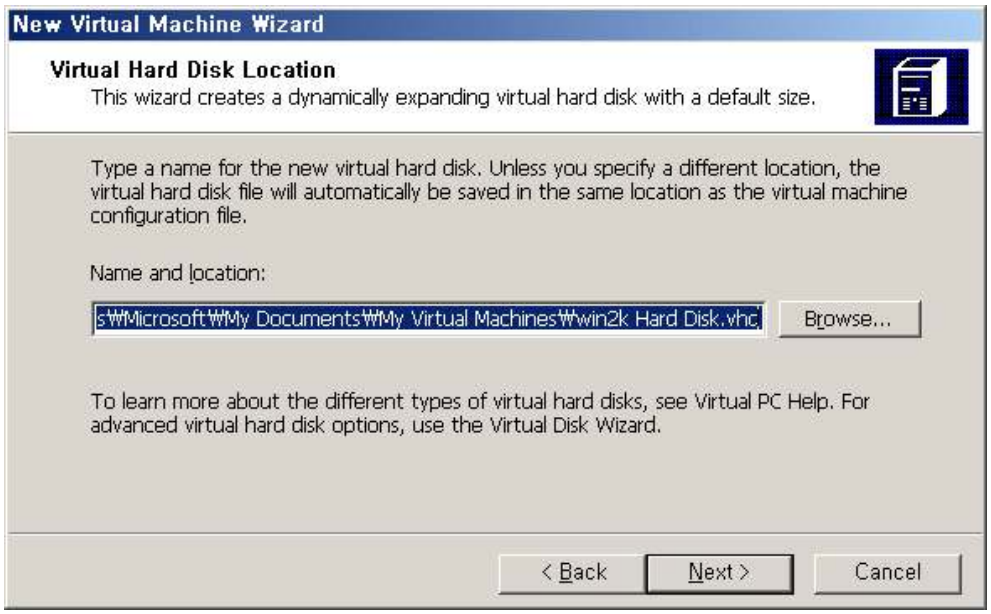


그림 3-88 Next

설치를 모두 끝마쳤다.

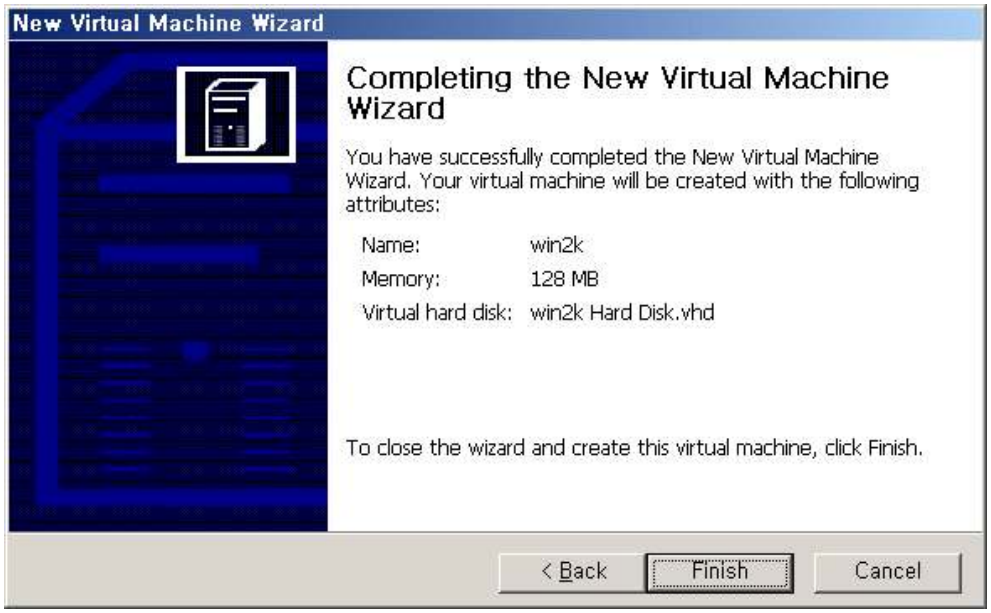


그림 3-89 Finish

Virtual PC를 실행하여 Start를 하도록 하자.

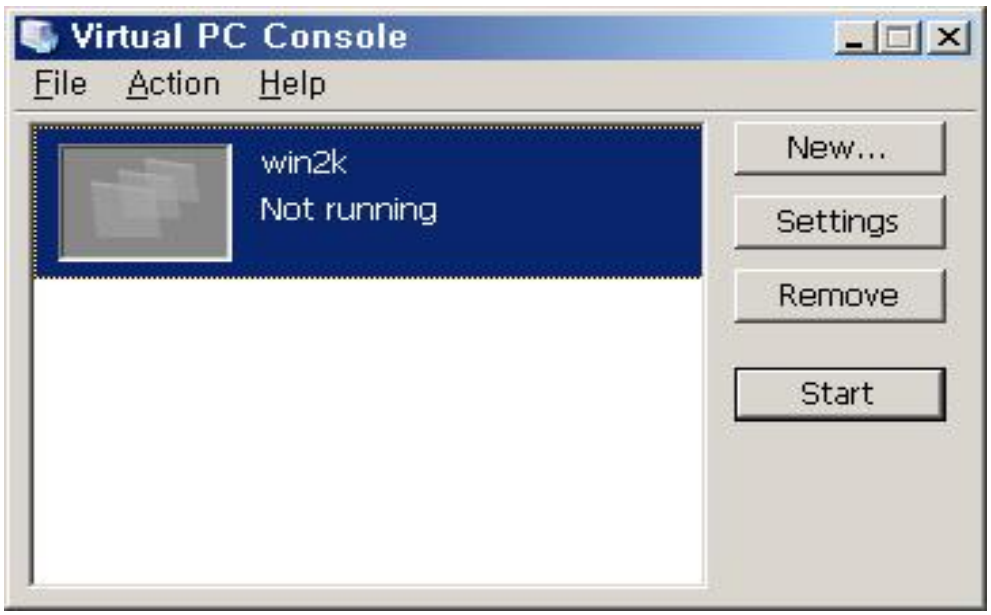


그림 3-90 Virtual PC 실행

Start를 시작하면 아래와 같은 화면이 출력 됩니다.
운영체제를 설치하기위해 Virtual PC CD메뉴에서 drive 설치 또는 이미지 설치를 선택해 설치한다.

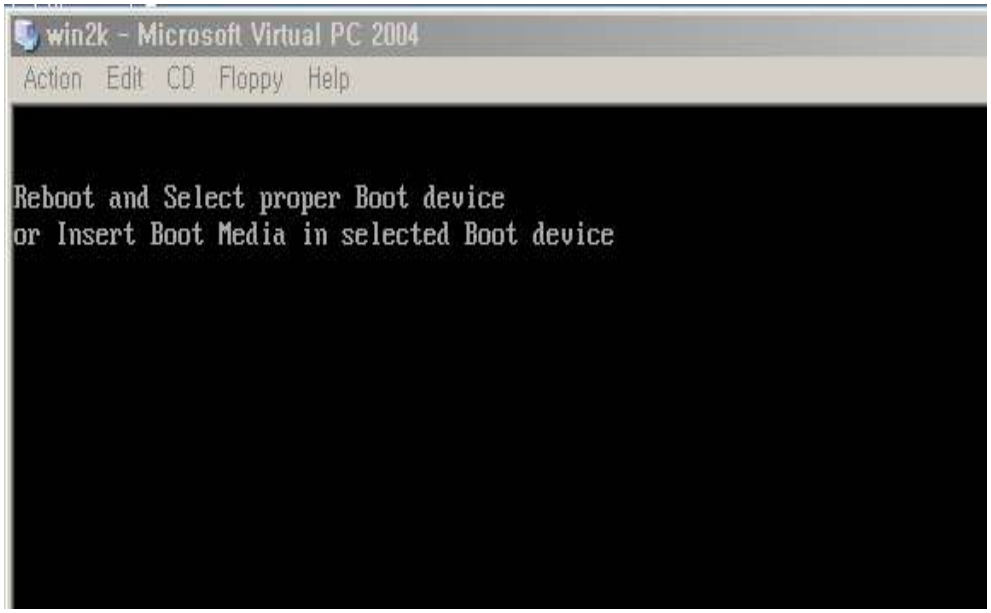


그림 3-91 이미지 선택 설치

위 와 같은 방법으로 아래그림처럼 Linux Winxp Win2000을 설치하였다.

그림 3-92 설치 완료

win2000 server 구동 화면

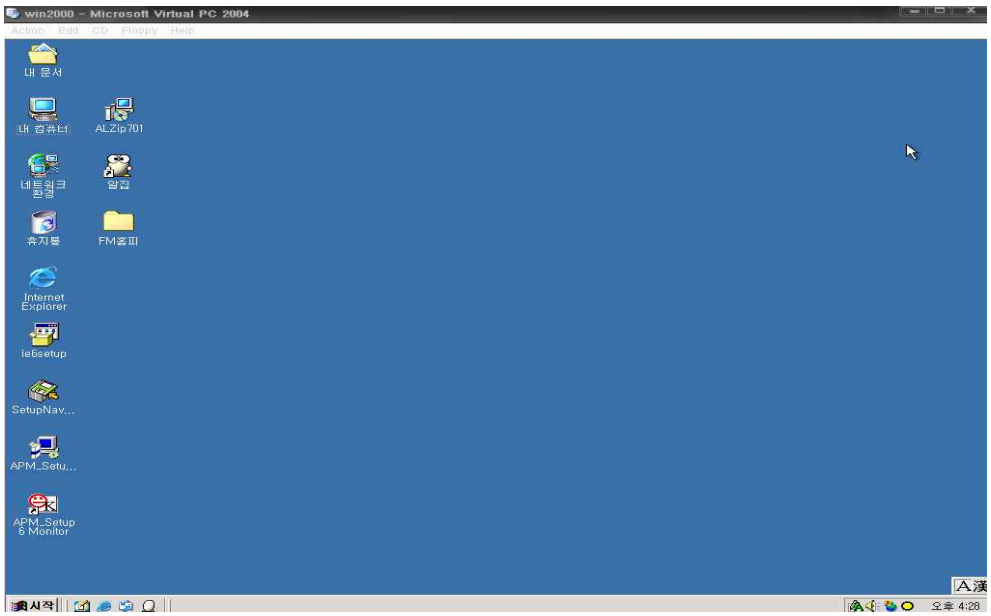


그림 3-93 구동 화면

전체적인 pot 구성 및 설치는 끝났다
이제 hacker들이 침입을 할 수 있도록 보안이 취약한 웹서버를 구동 시키도록 한다.

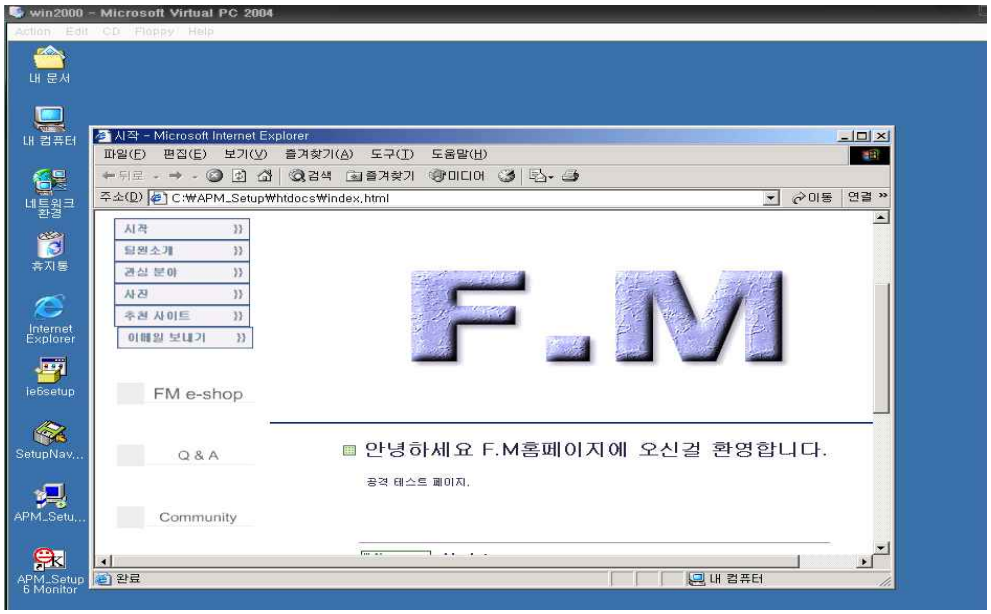
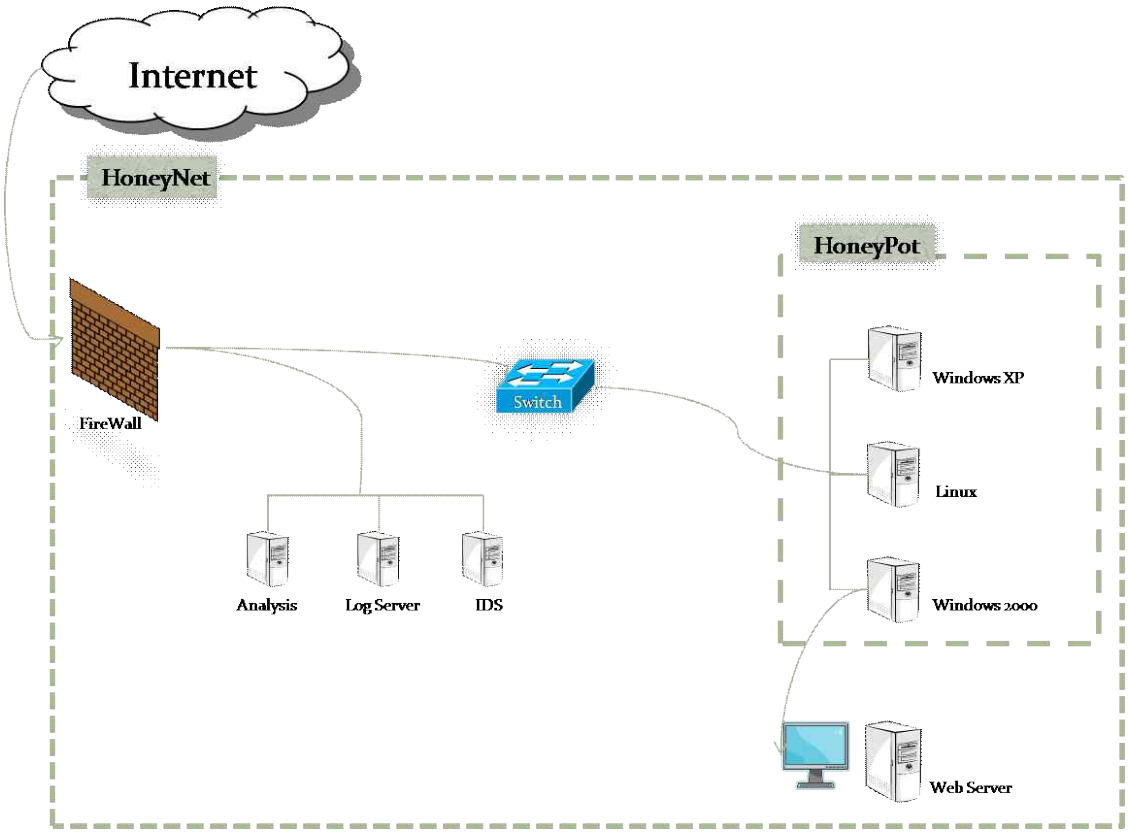


그림 3-94 웹서비스 구동화면

3.6 Honeynet 망 모식도

다음은 Honeynet의 전체적인 망을 설계한 그림이다.



3.6.1 시스템 환경

환경 구축을 위한 구성 요소

Server	CPU	RAM	HDD	LAN CARD	ETC
Firewall	Pentium 4 2.6GHz	512MB	60G	RealtekRt8139 RealtekRt8139	3COM SWITCH 10/100M
Log	Pentium 4 2.6GHz	512MB	60G	RealtekRt8139	
Honeypot	Pentium 4 2.6GHz	512MB	60G	RealtekRt8139	

4. 결과

4.1 Honeynet 시스템의 동작

Honeypot내에 있는 웹서버에 TrapServer1beta를 설치 하였다.
그림 4-94의 웹서버에 불법사용자가 접근 하였더니 그사용자의 ip주소와 로그가 기록 되었다.

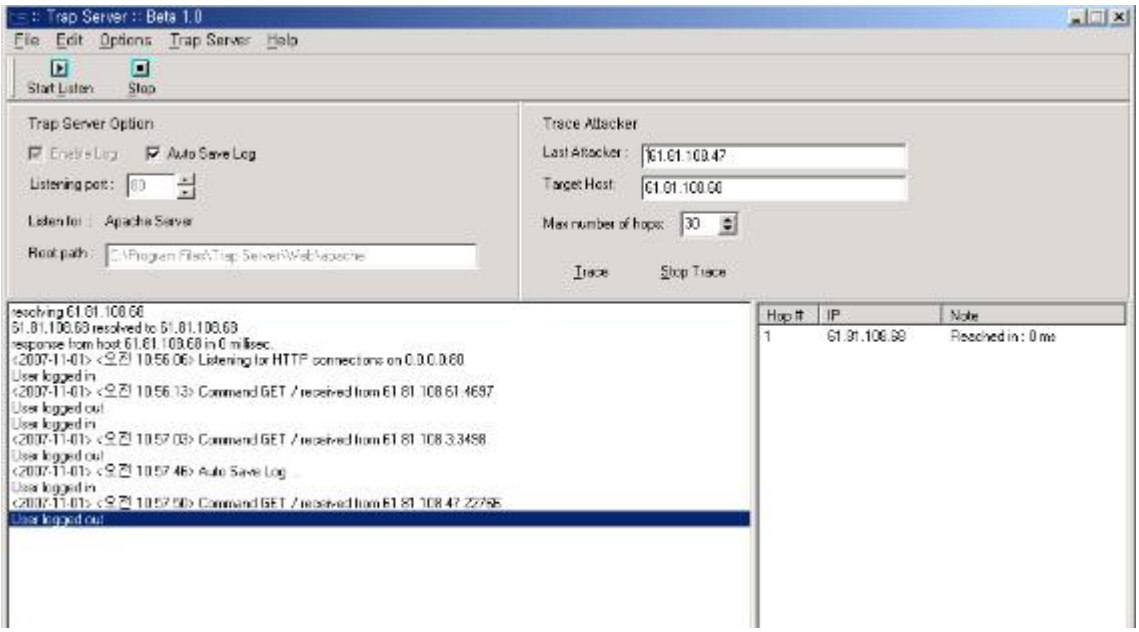


그림 4-1 TrapServer1beta 실행

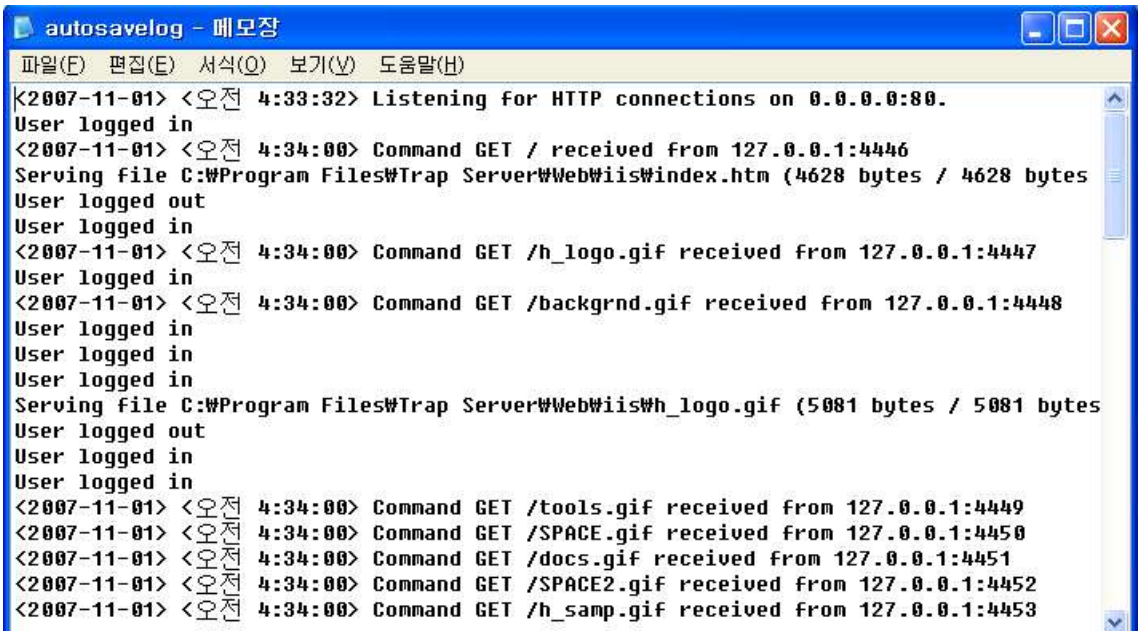


그림 4-2 autosavelog

4.2 Log 기록

로그 서버에서 cd /var/log로 이동해서 ls 명령어를 입력하면 Log message 목록이 보인다.

그림 4-3 Log message 목록

로그 목록 중 하나를 선택해 들어가면 아래와 같은 Log message가 보입니다.

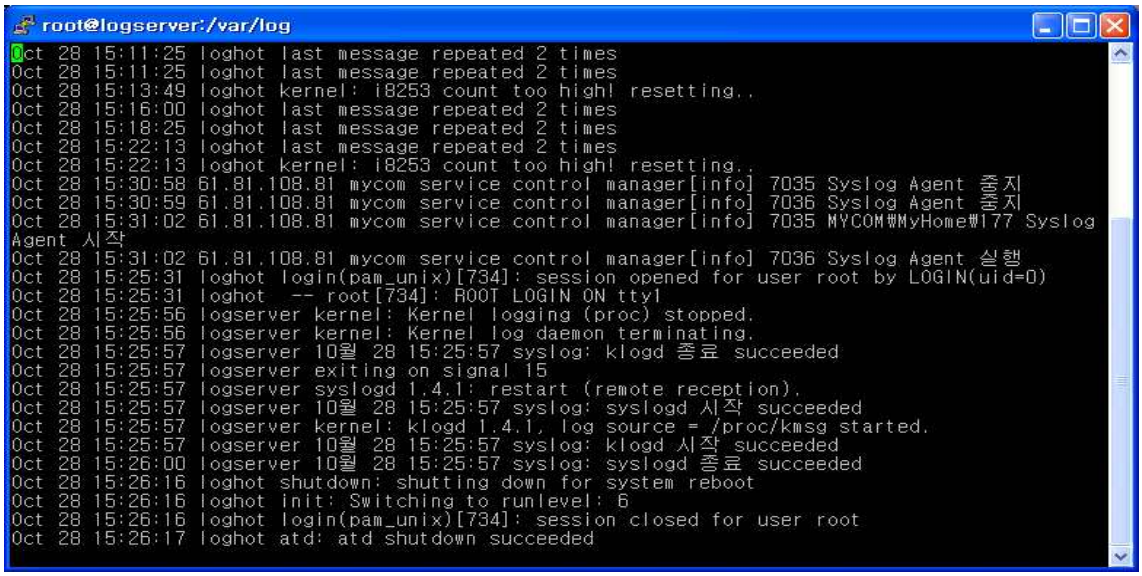


그림 4-4 Log message

30분마다 로그를 전송하라는 설정을 합니다.

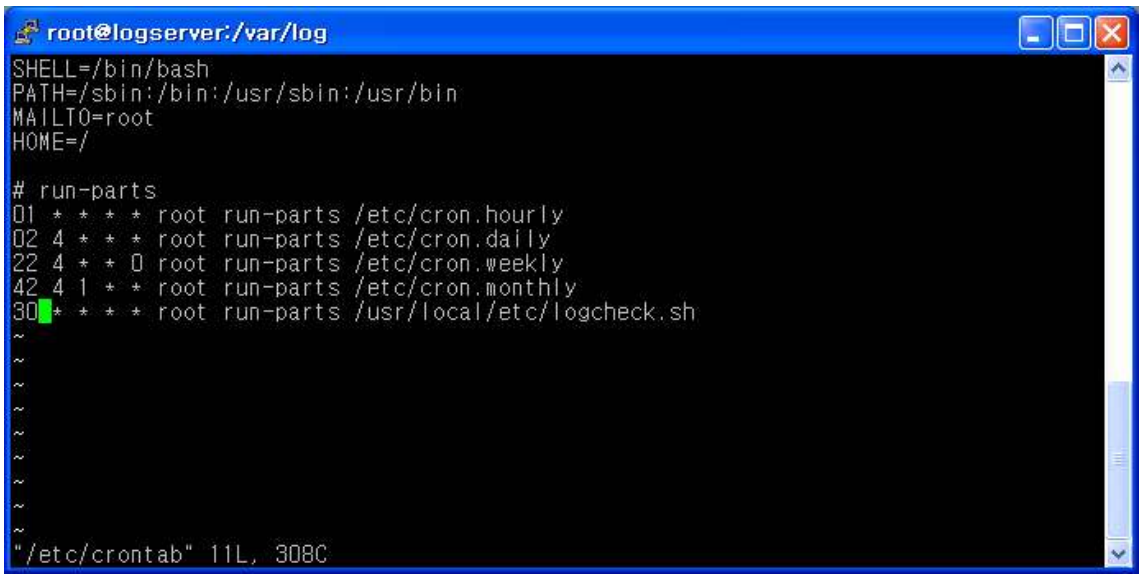


그림 4-5 Log message 자동 전송 시간 설정

저장된 로그가 관리자에게 보내지도록 관리자의 메일 주소를 설정해 줍니다.

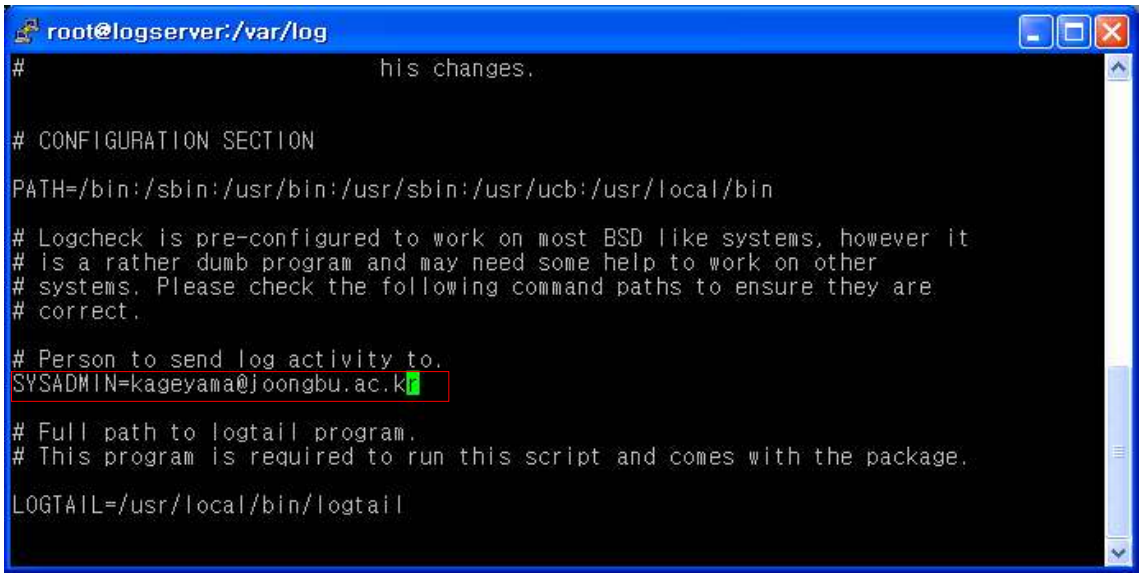


그림 4-6 Log message가 전달될 메일 주소

관리자에게 보내진 메일의 내용입니다.

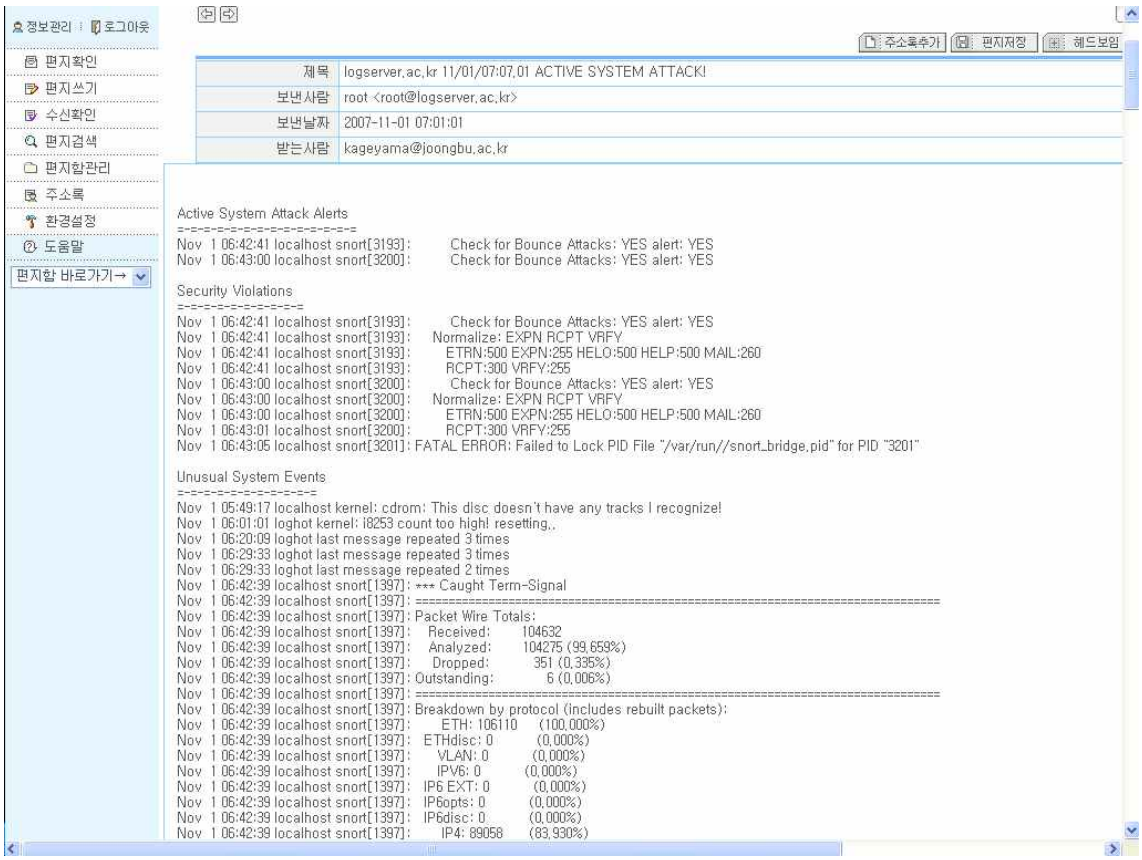


그림 4-7 관리자에게 전송된 로그 기록

4.3 로그서버 보안

그림 4-8 허용 포트 설정

로그 서버 보안을 위해서 로그를 보내는 514번 udp포트와 로그 기록을 메일로 보내는 25번 smtp 포트를 제외한 나머지 포트들은 모두 DROP시킴으로써 로그 서버에 대한 침입을 차단하였다.

5. 결론

Honeynet 구축 환경 요소 중 Firewall, Logserver, Honeypot과 같은 각종 시스템에 서버를 구축하여 응용프로그램을 설치하고 해커의 동기, 해킹경로, 해킹도구 등에 대한 정보를 수집하고 분석하여 모니터링 해본 결과 가상서버 또는 가상네트워크라는 장점을 통하여 여러대의 시스템을 가상 에뮬레이터에서 돌릴 수 있으면서 시스템을 구축하는 비용을 절감할 수 있었다.

보통 시스템을 구성하기 위해서는 많은 시스템이 필요하다. 그러나 가상시스템을 사용한다면 다수의 시스템에 들어가는 비용절감과 공간을 효율적으로 사용할 수 있다. Honeypot의 경우 자신에게 보내진 패킷만을 수집하기 때문에 데이터를 처리하기 위해 고성능의 시스템을 사용할 필요가 없으며 저사양의 시스템만으로도 운영이 가능하다. Honeypot을 이용하여 패킷을 수집하기 보단 Honeynet을 이용하여 패킷수집을 하여 공격 당하는 행위에 대하여 다수의 시스템을 전체적으로 수집 및 대응할 수 있었다.

Honeypot은 일종의 좋은 Tool이라고 할 수 있다. Firewall, IDS 같은 보안 시스템과 함께 사용함으로써 하나의 호스트에 대한 단편적인 부분 을 네트워크 전반에 해당되는 침입 행위에 대한 정보를 수집할 수 있었으며 비정상적인 사용자들의 침입 동기와 침입 방법에 대해 알 수 있었다.

마지막으로 안전한 원격로그서버를 구축하여 시스템 관리 및 현황분석을 하여 가장 기본이 되는 로그(Log)파일을 비정상적인 사용자들이 이 로그(Log)파일을 찾아내거나 유추할 수 없도록 하여 안전하게 로그를 보안하는 방법을 생각하여 적용하였다.

6. 참고자료

■ 도서

리눅스 서버관리 실무 바이블 上, 下
 - 슈퍼유저코리아, 박성수
리눅스 네트워크 관리자 가이드(개정판)
 - 한빛미디어,올라프 커치 , 장윤식 역
Redhat Linux 9.0 Snort Install Manual [Korea].pdf

■ Site

http://www.tcpdump.org/
http://www.snort.org/docs/
http://www.snort.org/docs/setup_guides/snort_base_SSL.pdf
http://www.snort.org/docs/snort-rh7-mysql-ACID-1-5.pdf
http://my.dreamwiz.com/winmil/security/snort.htm
http://www.superuser.co.kr/
http://www.linuxschool.net/
http://www.honeyney.org/
http://hackersnews.org

7. 감사의 글

시작할 때의 설레임이 어느새 끝마치는 아쉬움으로 돌아왔습니다. 이번 졸업작품은 저희들에게 있어서 여러 가지 큰 의미를 지닌 프로젝트였다고 생각합니다.
큰 포부를 가지고 시작한 프로젝트였기에 불타는 의욕을 가지고 시작하였으나 진행이 되면 될수록 불어나는 수많은 의문점과 시행착오들로 인하여 중간 중간 좌절하고 포기하고 싶기도 하였습니다. 하지만 그때마다 저희들에게 큰 힘과 사랑으로 보살펴 주신 교수님들과 학우 여러분들의 은혜에 심심한 감사를 드립니다. 이번 프로젝트를 준비하면서 나름대로 꼼꼼하고 완벽한 프로젝트를 지향하기 위해 노력하였지만 아직도 미비한 점들이 많이 있는 것을 인정하지 않을 수 없습니다.
다음에 이러한 기회가 또다시 주어진다면 미비한 점들을 세세히 살펴 보완하여 좀 더 보람차고 뜻있는 프로젝트가 되도록 최선을 다 할 것입니다.

이번 프로젝트를 끝내고 났을 때 저희팀원들 실력이 더욱더 한층 Up-Grade 되는 것 같아 가슴 한켠 뿌듯한 마음 금할 길이 없습니다. 이렇게 프로젝트를 원만하게 마칠수 있었던 것은 모두 행사에 참여해 주신 정보보호학과 교수님 및 동료학우 여러분들과 또한 관련분야의 모든 분들이 도와주신 덕분이라 생각하며 심심한 감사를 다시 한 번 드립니다.
이제 저희는 이 프로젝트를 끝으로 비록 교정을 떠나지만, 추억에 대한 그리움으로 다시 찾을 때까지 저희들을 기억해주시고, 따뜻하게 맞아주시고, 교수님들의 가슴 속에 소중하고 자랑스러운 제자로 저희들이 기억되어지기를 바랍니다.

그리고 이러한 인연이 사회생활에서의 고난이나 좌절이 생길 때 용기와 희망을 주시는 인생의 선후배로 소중한 인연이 이어지길 바랍니다.