□특집□

전산망 보호를 위한 PC방화벽 구축

백 석 칠†

◈ 목 차 ◈

1. 서론

인터넷의 가장 큰 특징은 전 세계 네트웍을 하나의 공통 프로토콜로 연결하고 있다는 점일 것이다. 이 때문에 세계 각처에서 전자메일, TELNET, FTP, WWW 등을 통한 활발한 정보 교환이 가능하다. 이와 같은 인터넷의 편리함을 이용하기 위하여 국내에서도 많은 연구소, 기업, 대학 등이 인터넷에 연결되어 있거나 연결을 서두르고 있는 실정이다. 그러나 요즈음에 이르러 해커들의 침입 사례가 빈번하게 매스컴을 통하여 보도되면서, 인터넷과의 연결을 꺼리는 분위기가 생기게 되었다. 이러한 해커들의 침입에 대한 보고는 주로 미국을 중심으로 한 구미 선진국들에 관한 내용들이었으나 작금에 이르러 국내에서도 해커의 침입 사례가 빈번하게 발생, 보고되고 있는 처지이다. 이러한 국내·외 여건에 대처하기 위하여 기업, 연구소, 공공기관 및 학계가 서로 협력하여 체계적이고 지속적인 전산망 보안 기술 개발을 하여야 할 것이다. 물론, 본 논문도 이러한 해커들의 침입으로 부터 전산망을 보호하기 위한 기술의 조사 및 연구의 결과를 정리하기 위한 목적으로 작성된 것이다. 본 논문에서 기술할 주된 내용은 전산망 관련 보안 문제에 대하여 대략적으로 소개하고 이들에 대한 대책들 중에서 전산망 보호 기술의 핵심이라고 할 수 있는 방화벽(Firewall) 시스템의 기능 및 종류에 대하여 서술한다. 끝으로 실제로 현장에서 전산망 보호를 위하여 사용할 수 있는 방화벽을 PC를 이용하여 구축하는 방법에 대하여 소개하기로 한다.

† 정회원 : 한국통신 멀티미디어연구소 보안과제 책임자

2. 인터넷 보안관련 문제점

인터넷은 요즈음 보안문제에 있어 심각한 상태에 직면하고 있다. 이러한 보안 관련 문제점들을 무시한 사이트들은 해커들로 부터 심각한 위협을 받을

수 있을 뿐만 아니라 이러한 네트웍 사이트가 침입당한 후에는 해커들이 다른 네트웍 사이트를 침입할 수 있게 하는 전초기지로 이용될 수도 있다. 보안에 신경을 쓰고 있는 사이트들도 이러한 해커들의 계속적인 침입시도 및 네트웍 소프트웨어 상에서 발견될 수 있는 새로운 결함들로 인하여 안심할 수는 없을 것이다. 인터넷 보안 상의 문제점들은 응용서비스 자체가 갖고 있는 허약함 또는 그러한 서비스를 지원하는 프로토콜의 보안결함 등에 기인할 수 있다. 다른 중요한 원인들로는 호스트 configuration 상의 실수및 access control 의 허술함 또는 너무 복잡하게 함으로 인한 실수 등을 들 수 있을 것이다. 이러한 기술적인 면 이외에 보안관련 업무를 매니저가 가볍게 생각함으로 인하여 인력 및 업무시간 투입에 인색함으로 인하여 보안관련 사고가 발생할 수 있다. 이러한 보안 상의 사고는 기업체, 연구소, 공공기관, 학계등이 인터넷의 사용율이 급속도로 증가되므로서 더욱 심각한 재산 및 정보 상의 피해를 발생시키게 되었다. 특히, 요즈음에는 인터넷을 통한 상거래가 빈번히 이루어지면서 인터넷에서 보안문제가 핵심 이슈가 되고 있다. 그러므로 본절에서는 인터넷 상에서 발생할 수 있는 보안상의 문제점들과 그 원인들에 대하여 상세히 언급하고자 한다.

2.1 인터넷 상의 보안사고 사례

인터넷 상의 보안 관련 사고들을 언급할 때, 반드시 언급되는 것은 아마도 UNIX sendmail 프로그램에 관련된 사항들 일 것이다. 이 프로그램의 취약성은 인터넷에서 공개적으로 많이 언급되어왔다. 그리고 문제가 있는 버전들에 대한 보완이 계속되고 있으나 sendmail 프로그램과 네트웍 관련 소프트웨어들의 복잡성 때문에 아직도 중요한 보안상의 취약점이 발견되고 있다. 그러므로 방화벽을 이용하여 sendmail에 대한 접근을 제어할 수 있도록 할 필요가 있다. 한편, 오픈 도메인에 있는 FTP 서버 경우에 사용자에게 서버 접근에 있어서 특권을 부여하는 트로이 목마를 갖고 있는 것이 발견되고 있어 경각심을 불러 일으키고 있다. 이러한 버전의 FTP 를 사용하는 사용자들 뿐만 아니라 기존의 FTP 서버 사용자들도 보안상의 문제점이 발생하지 않도록 필요한 조치를 하여야 한다.

마지막으로 CERT (Computer Emergency Response Team)와 CIAC(Computer Incident Advisory Capability) 에서 보고한 sniffer 프로그램에 의한 폐해를 들 수 있다. 해커들은 인터넷 상의 많은 호스트들로 침입을 시도하며 때로는 게이트웨이가 compromise 당하는 경우도 빈번하다. 이 경우에 해커들은 sniffer 프로그램을 이 곳에 심어서 네트웍 트래픽을 모니터링한다. 이 때, 유저들이 이 곳을 경유하여 다른 네트웍에 있는 시스템들을 TELNET 이나 FTP 로 액세스하는 경우, logun 이름과 password 가 sniffer 프로그램에 의해 해커의 손에 넘어가게 된다. 이 때, 불법적으로 입수한 계정은 해커들이 또 다른 시스템으로 침입할 때, 사용된다. 이러한 실례는 고정되어 있거나 재사용되는 패스워드(static or reusable password)의 문제점을 실감있게 부각시키는 사례라 할 수 있다. 그러므로 별도의 보안 조치없이 인터넷을 통하여 다른 네트웍 상에 있는 시스템을 액세스하는 행위는 고양이에게 생선을 맡기는 일과 같다고 할 수 있겠다. 다음 섹션들에서는 보안상의 문제를 야기시킬 수 있는 주된 원인들에 대하여 고찰하기로 한다.

2.2 허술한 인증(Weak Authentication)

인터넷 사고 처리팀들의 경험에 의하면 많은 보안관련 사고들은 주로 취약한 패스워드나 정적인 패

스워드의 사용에 기인한다고 한다. 인터넷 상의 패스워드들은 여러가지 방법들에 의해 풀릴 수 있지만 주로 다음 두가지 방법이 사용된다. 첫째 방법으로 CRACK과 같은 프로그램을 이용하여 불법적으로 입수된 PASSWORD 화일을 해독하는 방법을 들 수 있고 두번째로는 앞서 언급한 sniffer와 같은 프로그램을 통하여 패스워드를 가로채는 것이다. 유닉스의 경우 패스워드들을 암호화한 후, 보통의 유저들도 그 내용을 읽을 수 있는 화일에 넣어둔다. 이 화일은 단순히 복사를 하거나 여러가지 침입방법을 이용하여 획득할 수 있다. 이렇게 손에 넣은 패스워드화일은 이미 널리 알려진 CRACK과 같은 프로그램을 이용하여 해독할 수 있다. 그러나, 모든 패스워드가 풀리는 것은 아니다. 패스워드가 8자보다 작거나 영어로 된 단어 등은 쉽게 해독되는 반면 여러가지 특수 문자나 숫자를 조합하여 만들어진 패스워드는 잘 풀리지 않는다.

또 다른 인증과 관련된 문제로 현재 사용되고 있는 TCP나 UDP가 인증 작업을 호스트 주소로만 할 수 있다는 것이다. 그러므로 호스트 사용자들 자체를 인증할 수 없는 문제가 야기된다. 실례로 NFS 경우 NFS 서버 관리자가 어떤 특정한 호스트 사용자에게 NFS 서버 사용권한을 부여하고자 할 경우 그 유저가 이용하고 있는 호스트 자체에게 권한을 부여하여야만 한다. 그 결과로 그 호스트를 액세스할 수 있는 다른 유저들도 이 NFS 서버를 액세스할 수 있게 되므로 이는 보안상의 치명적인 약점이 될 수 밖에 없다.

2.3 Spying 과 모니터링 용이

TELNET이나 FTP를 이용한 다른 네트웍 호스트들에 대한 액세스 시, 패스워드는 일반적으로 암호화되지 않은 상태로 네트웍 상에서 전송된다. 이러한 상황에서 앞서 언급한 sniffer 같은 프로그램으로 이 패스워드들을 가로챌 수 있으며 이러한 계정 중에 관리자 계정이 포함되어있을 경우 심각한 보안 문제가 발생하게 된다. 이러한 종류의 사례가 가장 빈번하게 발생하고 있다는 점에 유의하여야 할 것이다. 전자우편의 경우도 모니터링을 쉽게 당할 수 있으며 그 속에 담겨있는 중요 정보들은 관련 사이트들에 관한 것들일 수 있으며 이것이 해커들의 침입에 악용될 수 있다. 물론, 전자우편의 경우도 모니터링을 쉽게 당할 수 있으며 그 속에 담겨있는 중요 정보들은 관련 사이트들에 관한 것들일 수 있으며 이것이 해커들의 침입에 악용될 수 있다. 전자우편 내용이 기업의 기밀 사항인 경우 그 내용이 공개되는 것만으로도 심각한 타격을 입을 수 있을 것이다. 그러므로 전자우편의 보안 취약성은 간과해서는 안될 사항이다.

한편, X윈도우 시스템은 유닉스 사용자들에게 없어서는 안될 응용프로그램이다. X윈도우는 하나의 웍스테이션에 많은 윈도우를 띄울 수 있는 기능이 있다. 또한, 자기자신의 시스템에서 다른 시스템 화면에 자기자신의 윈도우를 띄울 수도 있다. 그러므로, 해커들은 이러한 X윈도우의 기능을 이용하여 다른 시스템에 자기 자신의 윈도우를 띄우고 그 시스템의 사용자가 입력하는 중요정보(패스워드, 기타 기밀)들을 읽어들여 가로채기도 한다.

2.4 Spoofing 용이

TCP와 UDP는 호스트의 IP 주소만을 이용하여 인증작업을 할 수 있으므로 해커들은 이를 이용하여 자기자신의 호스트를 트러스트 호스트로 위장하는 것은 매우 쉬운 일이다. 이러한 일은 IP 소스 라우팅이라는 기능을 이용하여 가능하게 된다. 이 기능은 목적지 호스트까지 경로를 지정할 수 있으며 물론

돌아오는 패스도 지정할 수 있다. 그러므로 해커들이 자기자신의 호스트를 트러스트 호스트로 위장하는 다음과 같은 방법을 사용할 수 있다.

(1) 해커 자신의 호스트 IP 주소를 트러스트 호스트의 것으로 바꾼다.

(2) 해커는 공격하고자 하는 서버까지의 경로를 IP소스 라우팅을 이용하여 설정한다. 이 경우에 공격대상 서버 바로 전의 호스트는 또 다른 트러스트 호스트가 되게 경로를 설정한다.

(3) 해커는 서버에게 이 경로를 이용하여 클라이언트 요구사항을 보낸다.

(4) 서버는 이러한 요구사항이 트러스트 호스트에서 직접 온 것으로 간주하고 응답을 트러스트 호스트로 보낸다.

(5) 이러한 서버의 응답을 받은 트러스트 호스트는 이미 정해져 있는 경로를 따라 서버에게서 받은 응답을 해커 호스트로 보낸다.

많은 유닉스 호스트들이 이렇게 소스 라우팅된 패킷들을 받아줄 뿐만 아니라 보내주기도 한다. 대부분의 라우터들도 이와 같이 작동한다. 그러나 몇몇 라우터들은 소스라우팅된 패킷을 막아주는 기능을 보유하고 있다. 한 예로 조직 내부에서 해킹하기 위하여 주로 사용되는 방법을 소개해보기로 하자. 우선 다음과 같은 모양을 갖는 조직을 생각해보자.

즉, 유저들은 주로 PC를 사용하고 서버로는 유닉스 호스트들을 이용하며 이들을 TCP/IP LAN으로 연동시키고 있는 경우를 상상해보기로 하자. 이러한 상태에서 해커 자신은 UNIX 서버 상에서 운용되고 있는 NFS 서버를 액세스할 수 있는 권한이 없다고 할 때, 해커는 다른 유저(UNIX 서버 상의 NFS 서버에 접근 가능한 자)이 자기자신의 PC 파워를 끌 때까지 기다린 후 자기자신의 시스템 IP 주소를 전원이 나간 PC의 주소로 설정하여 공격목표인 NFS 서버 시스템에 쉽게 들어갈 수 있다. 그러므로 이러한 방법은 주로 조직 내부에 존재하는 해커들이 즐겨 사용하는 핵킹 방법이다.

전자우편은 spoofing 하기가 훨씬 용이하다. 인터넷 상에서 서로 다른 호스트 사이에서 E-mail을 주고 받을 때 ASCII 문자의 명령어로 구성된 프로토콜을 사용한다. 그러므로 침입자(Intruder)는 TELNET을 통하여 SMTP 포트에 직접적으로 이러한 명령어를 입력시킬 수가 있다. 그리고 이러한 명령어에 따라 작동하는 SMTP 서버는 침입자가 입력하는 모든 내용을 사실로 받아들이므로 조작하여 입력하는 거짓 주소나 그 밖의 정보들을 그대로 인정하게 되므로 별다른 특권이 없는 해커들이 전자우편을 거짓으로 작성하여 보내는 일은 어렵지 않다. 그 밖에 서비스들 가령 DNS 같은 것들도 spoofing을 당할 수 있다. 그러나 이러한 서비스들을 spoofing 하는 것은 전자우편의 경우보다 어렵다.

2.5 LAN 서비스 및 트러스트 호스트

시스템들의 관리를 용이하게 하고 LAN의 성능을 향상시키기 위하여 많은 네트웍 사이트들이 NIS(Network Information Services)와 NFS 서버들을 사용하고 있다. 이러한 서비스들은 네트웍 상의 많은 호스트들을 관리하기 쉽게 하며 화일이나 데이타들을 서로 쉽게 주고 받을 수 있게 하는 반면, 보안에 있어서는 자체적인 많은 결함들을 갖고 있다. 즉, 이러한 네트웍 상황에서는 중심이 되는 서버 시스템이 해커들에 의하여 점령되면 이 서버 시스템을 믿고 있는 다른 모든 호스트들은 훨씬 쉽게 침입을 당할 수 있다. rlogin과 같은 서비스들의 경우 사용자 이용 편의를 위하여 트러스트 호스트라는 개념을 도입하여 이들 호스트간에 맺어지는 세션들에서는 패스워드를 요구하지 않는다. 이러한 기능은 네트웍 상에 패스워드가 흘러다니지 않게 하여 보안을 향상

시킨다고도 할 수 있지만, 이 시스템들 중에서 한 시스템이 해커에 의해 침입 당한 경우, 다른 시스템에 대한 액세스 권한은 자동으로 해커의 수중에 넘어가게되는 보안문제가 발생할 수 있다. 이러한 연유로 요즈음에는 상호 신뢰 호스트 개념의 사용은 권장되고있지 않다.

2.6 복잡한 구성과 제어

호스트 시스템들의 액세스 제어를 구성하고 또 이들이 정확하게 되었는지 테스트하는 것은 간단한 문제가 아니다. 결과적으로 잘못된 구성은 해커들의 침입 경로로 사용될 수 있다. 아직도 어떤 유닉스 시스템들의 경우에 액세스 제어가 거의 되어있지 않은 상태에서 공급되고 있다.

지금까지 많은 보안 사고들은 해커(물론 일반 유저들 그리고 사고처리팀 및 시스템 공급 업체들도 포함) 들이 발견한 취약점들에 기인한다. 또한 대부분의 유닉스 시스템들이 널리 알려진 BSD 네트웍 코드들로 부터 발전되어 왔기 때문에 이들 코드들을 연구한 해커들은 자신들이 발견한 보안상의 결함들을 이용하여 시스템들에 침입할 수 있을 것이다. 이러한 코드 상의 버그들이 존재하는 이유로는 소프트웨어 자체의 복잡함과 모든 환경에서 테스트 될 수 될 수 없는 점 등을 들 수 있을 것이다.

2.7 호스트 베이스 보안 소홀

호스트 베이스 보안은 해당 네트웍 사이트들의 호스트 수가 증가함에 따라 더욱 소홀히 취급될 가능성이 높아진다. 한 호스트에 대한 보안을 철저히 한다는 것은 다른 많은 시스템들의 보안을 소홀히 하는 결과를 낳을 수 있기 때문이다. 그리고 시스템 관리자의 역할이 자주 바뀌는 경우가 있어 서둘러 작업을 하는 경우가 자주 발생할 수 있다. 따라서 각 시스템 간의 보안 상태가 고르지 않게 설정되는 결과를 초래할 수 있다. 이 경우에 보안 상태가 취약한 시스템이 해커의 공격대상이 될 가능성이 높아진다.

네트웍 소프트웨어의 결함이 발견되면 방화벽과 같은 시스템으로 보호를 받지 못하고 있는 호스트들은 신속하게 보안 헛점을 제거하여야 한다. 그러나, 단 기간 내에 많은 시스템들을 패칭하는 것은 쉬운 일이 아니다. 더우기 사용하고 있는 OS 의 버전이 다를 때, 이러한 작업이 불가능하게 될 수도 있다. 이러한 네트웍 사이트는 해커들의 활동에 속수무책인 처지(sitting duck)가 되고 말 것이다.

이 상의 여러 섹션에서 언급한 것 처럼 TCP와 UDP는 보안상의 결함에도 불구하고 인터넷 환경에서 널리 쓰이고 있다. 정부, 기업, 연구소, 대학 등 뿐만 아니라 많은 일반 사용자들도 폭발적으로 인터넷을 사용하고 있는 상황에서 이러한 보안관련 사고들이 얼마나 빈번하게 발생하는가에 대한 정확한 통계자료는 없다. 그러나 CERT/CC(Computer Emergency Response Team/Coordination Center)는 1988년 부터 조사한 기초통계자료를 갖고 있다. 이들 데이타에 따르면 매년 보안관련사고들은 급격히 증가하고 있는 실정이며 물론, 인터넷의 규모도 매우 빠른 속도로 확장되고 있다고 한다.

한편 NIST 는 인터넷이 매우 유용한 전산망이지만 해커들의 공격에 상당히 취약하다는 점을 강조하고 있으며 다음과 같은 사항들이 보안사고 위험 수준에 관계가 있다고 주장하고 있다.

- 현 네트웍 사이트에 있는 시스템들의 수
- 현재 제공하고 있는 서비스의 종류
- 인터넷에 연결되어있는 방식
- 현 사이트의 외부 인식정도
- 보안관련사고에 대처하는 방식

많은 시스템들이 서로 연결되어있을 때, 보안을

위한 제어가 더욱 어려워진다. 그리고 인터넷에 연결되어있는 포인트들이 많은 네트웍 사이트는 단일 게이트웨이를 통하여 연결되어있는 사이트 보다 해커의 공격을 받을 수 있는 가능성이 높아진다. 또한, 지명도가 높고 잘 알려진 사이트일 수록 공격목표대상이 되기 쉬울 것이다. NIST는 네트웍 사이트의 보안수준을 높이고 인터넷을 보다 안전하게 이용하기 위한 방법으로 일회성 패스워드(One-time password)를 사용하는 방화벽 시스템의 구축을 강력히 추천하고 있다. 다음 장에서는 주로 방화벽 시스템의 구조 및 보안기능 등을 중점적으로 서술하기로 한다.

3. 방화벽의 기능 및 종류

앞 장에서 언급한 인터넷과 연관된 보안 문제들은 기존의 잘 알려진 보안기술이나 호스트 액세스 제어 등을 통하여 상당 수준 해결되거나 또는 이들의 위험성이 많이 완화될 수 있다. 그 중에서도 방화벽은 자신의 네트웍 사이트들을 인터넷을 통하여 침입하는 해커들로 부터 효과적으로 방어할 수 있을 뿐만 아니라 인터넷 상에 존재하는 유용한 서비스의 이용 및 정보 획득을 용이하게 해줄 수 있어 가장 각광 받는 보안기술로 인정받고 있다. 그러므로 본 장에서는 주로 방화벽 구축을 위한 기술에 관하여 언급하기로 한다.

한편, 일반적으로 방화벽들은 다음 2가지 사항들을 설계 정책으로 채택한다.

1) 거절이 명시되지 않은 서비스는 허락한다.
(Permit any service unless it is expressly denide.)

2) 허락이 명시되지 않은 서비스는 거절한다.
(Deny any service unless it is expressly permitted.)

3.1 방화벽 기능

방화벽이라 하면 네트웍에 보안기능을 제공할 수 있는 라우터, 호스트 시스템 또는 이러한 시스템들의 집합 등만을 의미하는 것은 아니다. 차라리 방화벽이라고 하는 것은 보안이라는 목표에 다가가기 위한 접근방법이라고 할 수 있다. 방화벽은 승낙할 서비스들이나 액세스를 정의하는 보안정책을 구현할 수 있게 할 뿐만 아니라 네트웍의 구조, 라우터 및 호스트들, 그 외에 보안기법(정적인 패스워드를 대체할 수 있는 진보된 인증 등) 등을 이용한 보안정책의 구현을 의미한다. 방화벽 시스템의 주된 목표는 자신의 네트웍으로 부터 다른 네트웍들의 액세스를 제어하거나 외부로 부터 액세스를 규제하는 것이다. 모든 연결행위를 방화벽 시스템을 통해서만 가능케 함으로서 이러한 연결행위를 모니터링할 수 있게 된다. 방화벽 시스템은 라우터, PC, UNIX 호스트 또는 이들의 집합체가 될 수도 있다. 이들은 네트웍 사이트나 서브넷을 외부의 호스트들에 의해서 자주 사용되는 프로토콜 및 서비스들을 이용한 해커들의침입으로 부터 보호할 목적으로 구성될 수 있다. 이 방화벽 시스템은 주로 네트웍 사이트를 보호하기 위하여 인터넷과 연결하는 길목에 위치시키거나 이보다는 규모가 작은 서브넷들이나 호스트들을 보호하기 위하여 이들의 게이트웨이에 설치하기도 한다.

방화벽을 설치하는 이유 중에서 첫번째로 곱을 수 있는 것은 네트웍 사이트의 호스트들이 사용하고 잇는 서비스들 중에 보안 취약점이 있는 NFS나 NIS 같은 편리한 서비스를 계속적으로 안전하게 이용하면서 동시에 외부로 부터 침입을 봉쇄하기 위함일 것이다. 사실 이러한 방화벽이 설치되어있지 않을 경우 해당 사이트의 호스트들은 전적으로 소스트 단위 보안에 의존하여야 하며 모든 호스트들을 상당 수준의 보안 상태를 유지하게 하여야 할 것이다. 그

러나 앞서 언급한 것처럼 호스트의 수가 많아지면 이러한 작업은 용이치 않게 되므로서 보안상의 헛점을 발생시키게 될 것이다. 방화벽은 이러한 상황에서 해당 사이트 내에 존재하는 많은 호스트들의 보안상태를 전반적으로 향상시켜줄 방법이 될 것이다.

다음은 방화벽을 사용하여 얻을 수있는 보안상의 주요 잇점들이다.

- 보안상의 취약점을 갖고 있는 서비스들의 보호
- 해당 사이트 시스템들에 대한 액세스 제어
- 집중화된 보안
- 강화된 프라이버시
- 네트웍 사용 또는 오용(misuse)에 관한 자료축적
- 네트웍 액세스 정책 강화

이상에서 언급한 사항들이 방화벽을 구축함으로 얻을 수 있는 보안상의 잇점들이다. 그러나 완벽한 보안이란 불가능하다. 방화벽에 의한 보안도 예외가 될 수는 없다. 아래 열거하는 사항들이 방화벽을 설치하므로서 발생할 수 있는 유저들이 느낄 수 있는 사용상의 불편과 보안상의 문제점들이다.

- 유용한 서비스들에 대한 제한된 액세스
- 백도어(Back Doors)들에 대한 무방비
- 내부 해커에 대한 무방비
- 기타 문제들(WWW, gopher, MBONE, Viruses, Through-put, all eggs in single basket)

3.2 방화벽의 종류

앞 절에서는 주로 방화벽 구성요소들에 대하여 언급하였다. 본절에는 이러한 구성요소들을 이용하여 구축할 수 있는 방화벽의 타잎들에 대하여 구체적으로 서술코저 한다. 방화벽의 유형에는 대체로 다음 4 종류가 있다고 할 수 있다.

- Packet Filtering Firewall
- Dual-homed Gateway Firewall
- Screened Host Firewall
- Screened Subnet Firewall

위에서 언급한 방화벽 유형외에도 다른 여러가지 종류가 있을 수 있다. 그러나 이러한 것들은 현재 언급한 것들을 좀더 확장한 것이 대부분이므로 지면관계상 더 이상 언급하지 않기로 한다. 한편, 요즈음에는 모뎀들이 고속화됨에 따라 SLIP 이나 PPP 를 통하여 인터넷에 연결하는 행위가 빈번히 발생하고 있다. 그러므로 모뎀과 방화벽을 효과적으로 연동할 수 있는 방법에 대해서도 서술하기로 한다.

3.2.1 Packet Filtering Firewall

패킷 필터링 방화벽은 복잡하지 않은 소규모 사이트들을 보호하기 위하여 사용되며 구축하기도 용이하다. 그러나, 방화벽으로서 여러가지 결점들과 바람직하지 못한 특성들을 내포하고 있다. 본절에서 소개하는 4가지 타잎의 방화벽 시스템들 중에서 가장 많은 결점들을 갖고 있다. 그림 3-1에 예시되어 있는 것처럼 패킷 필터링을 기능을 갖고 있는 라우터를 인터넷 게이트웨이에 위치시킨 후, 특정 서비스 프로토콜이나 인터넷 주소들을 막거나 필터링하기 위하여 패킷 필터링 룰(rule)을 라우터에 셋팅한다. 이렇게 구성된 방화벽은 일반적으로 해당 사이트 시스템들로 부터 인터넷의 접속은 모두 허용하는 반면, 인터넷으로 부터 내부 시스템들이나 서비스들의 액세스는 모두 막거나 극히 일부만 허용한다. 물론, 이러한 내부 시스템들이나 서비스들에 대한 액세스는 보안정책에 따라 결정되어야한다. 그러나 본질적으로 보안상에 결함이 있는 NIS, NFS, X 윈도우 같은 서비스들에 대해서는 예외없이 접근을 허용치 않는 것이 상례이다. 사실 패킷 필터링 방화벽은 구성요소인 패킷 필터링 라우터가 갖는 것과 같은 불편함을 갖는다. 더우기 이러한 불편함은 보안을

요하는 사이트들의 구조가 복잡해지고 보안정책이 엄격하게 됨에 따라 더욱 심화된다. 이러한 요소들로 다음과 같은 것들이 있다.

- 기록(log)을 남기는 기능이 미약하거나 심지어 없는 경우도 있어 라우터가 이미 설득(compromise)당했는지 또는 현재 공격당하고 있는지 등에 대한 상황을 파악하기가 용이치 않다.
- 패킷 필터링 룰은 테스트 자체가 쉽지 않기 때문에 테스트가 되지 않은 취약점들로 인하여 사이트 전체가 위험에 노출될 수도 있다.
- 너무 복잡한 필터링 룰이 요구되는 경우, 이를 다루는 자체가 불가능해질 수도 있다.
- 인터넷으로 부터 액세스를 허용하고 있는 호스트들은 자체적인 수단을 확보하여야 한다.

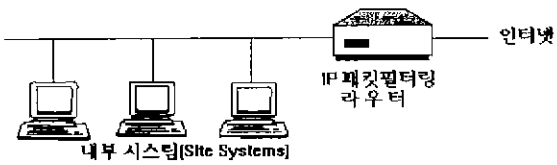

그림 3-1 패킷 필터링 방화벽

특히, 유의하여야 할점은 패킷 필터링 라우터가 소스 포트나 내부 또는 외부로 향하는 패킷들을 필터링하지 못하면 방화벽의 두번째 정책인 "허락이 명시되지 않은 서비스는 거절한다."를 구현하는 것이 더욱 힘들게 될 수도 있다. 그러므로 두번째 정책을 실현하기 위해서는 필터링 기능이 좀더 뛰어난 라우터를 선택하는 것이 바람직하다

3.2.2. Dual-homed Gateway Firewall

Dual-homed 게이트웨이는 패킷 필터링 방화벽 시스템보다 좀더 높은 보안성능을 실현할 수 있다. 이 게이트웨이는 두개의 네트웍 인터페이스를 갖고 있는 호스트 시스템으로 구성된다. 이 시스템은 IP 패킷 포워딩 기능을 할 수 없게 세팅되어야 한다. 이 밖에 추가적인 보안을 위하여 패킷 필터링 라우터를 인터넷과 연결하는 길목에 위치시키기도 한다. 이러한 라우터의 추가는 그림 3-2 에서 알 수 있는 것처럼 Dual-homed 게이트웨이와 라우터 사이에 screened subnet 설치를 가능케 한다. 이 screened subnet 내에는 인포메이션 서버(웹, FTP, Gopher 등) 또는 모뎀 풀들을 위치시킬 수 있다. Dual-homed 게이트웨이는 패킷 필터링 라우터와는 달리 인터넷과 내부 사이트를 오고 가는 IP 트래픽에 대한 완벽한 블록이 될 수 있다. 서비스들과 액세스는 이 게이트웨이에 설치되는 Proxy 서버들에 의해 가능케 된다. 이러한 기능은 간단하지만 아직까지 매우 안전한 방화벽 기능을 하는 것으로 알려져있다.

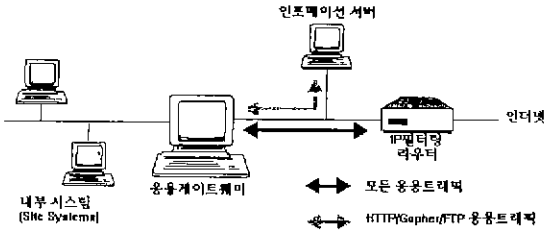

그림 3-2 Dual-homed Gateway Firewall with Router

그리고 앞서 언급한 방화벽 정책 중 두번째를 만족시킬 수 있다. 왜냐하면 결코 어떠한 서비스들도 게이트웨이 내에 해당 서비스의 프록시 서버를 설치하지 않고는 이 게이트웨이를 통과할 수 없기 때문이다. 더우기 Source-routed 된 패킷들을 받을 수 있는 기능을 작동하지 못하게 할 수 있으므로 어떠한 패킷들도 이 게이트웨이를 우회하여 보호받고 있는 사이트 시스템들에게 접근할 수 없다. 이러한 방화벽이 구축된 상태에서는 내부 사이트의 시스템들에 대한 정보가 단지 방화벽에만 알려지게 된다. 즉, 내부 사이트에 대한 DNS 정보가 방화벽 시스템에게만

알려지게 되므로 내부 사이트에 존재하는 시스템들의 이름이나 IP 주소들은 외부에 알려지지 않게 되므로 해커들의 침입으로 부터 내부 사이트 시스템들을 보호할 수 있게 된다.

이러한 방화벽의 가장 간단한 구성은 TELNET, FTP, 그리고 집중화된 전자메일 서비스등에 대한 프록시 서버를 Dual-homed 게이트웨이로 사용되는 시스템에 설치하는 것이다. 방화벽으로 호스트 시스템을 사용하므로 유저들을 인증할 수 있는 소프트웨어나 그 밖의 다른 인증 수단을 방화벽 시스템에 장착하여 보안 수준을 높일 수 있을 뿐만 아니라 방화벽에 대한 액세스, 액세스 시도 및 침입을 위한 목적으로 보이는 시스템에 대한 조사(Probe)등의 행위에 관한 기록을 저장할 수 있다. 그림 3-2에서 보는 바와 같이 Dual-homed 게이트웨이 방화벽 시스템은 인포메이션 서버와 관련된 트래픽을 다른 트래픽으로 부터 분리할 수 있는 기능도 갖고 있다. 즉, 게이트웨이가 인포메이션 서버들을 위한 프록시 서버들을 장착하고 있는 경우, 라우터는 인터넷으로 부터 인포메이션 서버들에 대한 직접적인 액세스들은 막고 대신 해당 프록시 서버가 설치된 방화벽 시스템(Dual-homed 게이트웨이)을 통하여 접근하도록 할 수 있다. 이러한 기능은 인포메이션 서버들에 대한 외부로 부터 직접적인 액세스로 인하여 서버들의 이름이나 IP 주소들이 DNS에 의해 외부에 알려지는 것을 방지할 수 있다. 라우터와 Dual-homed 게이트웨이 사이에 있는 서브넷(일명, Screened 서브넷 또는 DMZ)에 이러한 인포메이션 서버들을 위치시키는 이유는 다음과 같다. 즉, 그림 3-2 에서 알 수 있는 것처럼 해커들이 인포메이션 서버들을 성공적으로 침입한 후에도 내부 사이트 시스템들을 침입하기 위해서는 반드시 Dual-homed 게이트웨이를 통과해야 하기 때문이다.

한편, Dual-homed 게이트웨이 방화벽 시스템의 단점으로는 프록시가 존재하는 서비스들만을 액세스할 수 있다는 것이다. 그러므로 외부에 허용하고자 하는 서비스들은 게이트웨이의 인터넷 사이트(외부)에 위치시켜야 한다. 그러나 라우터를 사용하는 경우 screened subnet 에 위치시킬 수 있다. 물론, Dual-homed 게이트웨이 방화벽 시스템에 사용되는 호스트 시스템은 매우 안전하게 관리되어져야 한다. 즉, 보안상 취약점이 있는 서비스들이나 기술들은 철저히 배제하여야 한다.

3.2.3 Screened Host Firewall

Screened Host 방화벽은 Dual-homed 게이트웨이 방화벽보다 융통성이 있다. 그러나 이러한 융통성은 보안에 있어서 얼마 간의 희생을 요구한다. 그러므로 Screened Host 방화벽은 Dual-homed 게이트웨이 방화벽에 의한 보안상태보다 좀더 융통성을 요하는 사이트에 적당하다. Screened Host 방화벽은 패킷 필터링 라우터와 이에 의해 보호되는 서브넷에 위치한 응용게이트웨이(Application Gateway)로 구성되어 있다. 응용게이트웨이는 단지 하나의 네트웍 인터페이스를 요한다. 응용게이트웨이는 응용게이트웨이에 프록시 서버가 존재하는 TELNET, FTP 및 기타 서비스들을 사이트 시스템들에게 보낸다.

한편, 라우터는 보안상의 취약점이 있는 프로토콜들이 응용-게이트웨이나 사이트 시스템들에 접근하는 것을 막거나 선별적으로 통과시킨다. 이러한 작업은 다음과 같은 룰에 따라 수행된다.

- 인터넷으로 부터 응용게이트웨이로의 트래픽은 허용한다.
- 인터넷으로 부터 오는 다른 모든 트래픽은 거절한다.
- 내부로 부터 오는 트래픽 중에서 응용게이트웨이를 통과하지 않은 트래픽은 거절한다.

Dual-homed 게이트웨이와는 다르게 응용게이트웨이는 단지 하나의 네트웍 인터페이스를 필요로 하며 응용게이트웨이와 라우터 사이에 별도의 서브넷을 요구하지 않는다. 이러한 점은 방화벽을 좀더 융통성이 있게 하지만 라우터가 신뢰하는 서비스들(Trusted Services)을 응용게이트웨이를 통과시키지 않고 직접 해당 사이트 시스템에 보내게 되어 보안상의 문제를 일으킬 수 있는 가능성이 높아진다. 여기서 신뢰받는 서비스들은 프록시 서버들이 서포트되지 않는 서비스들이거나 보안상의 심각한 문제가 없다고 판단되는 것들로서 위험성을 감수할 만한 서비스들을 의미한다. 예로 NTP 나 DNS 를 들 수 있다. DNS 경우는 사이트 시스템들이 인터넷 시스템들에 대한 DNS를 액세스하기를 원할 경우 서비스를 허락한다.

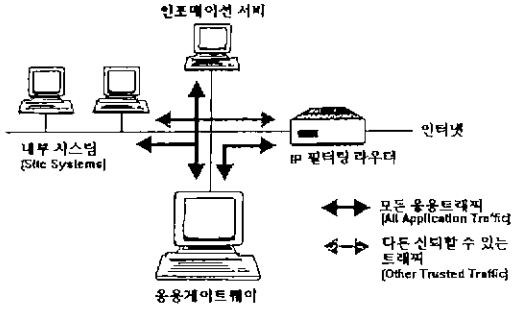

그림 3-3 Screened Host Firewall

Screened host 방화벽의 부가적인 융통성은 다음 두가지 우려 사항을 발생시킨다. 첫번째, 패킷 필터링 라우터에 입력되는 룰은 복잡하고 테스트가 용이치 않은 연유로 라우터를 통한 security hole 이 발생하기 쉽다. 그러나 라우터는 응용서비스들의 트래픽을 응용게이트웨이로 국한시키기 때문에 룰셋(rule set)은 패킷 필터링 게이트웨이 방화벽 시스템에 사용되는 라우터 경우보다는 간단하다. 두번째 문제점은 Screened host 방화벽이 갖고 있는 융통성이 보안정책을 위반하게 되는 가능성을 열어놓게 될 수 있기 때문이다. 이러한 위험성은 Dual-homed 게이트웨이 방화벽의 경우에는 문제가 되지 않는다. 여하튼, 보안상의 문제를 야기시키지 않기 위해서는 강력한 보안정책이 필수적이라고 할 수 있겠다.

3.2.4 Screened Subnet Firewall

Screened Subnet 방화벽은 Dual-homed 게이트웨이 방화벽과 Screened Host 방화벽을 융합한 구조를 갖고 있다. 이러한 구조는 방화벽의 각 구성요소들을 서로 다른 시스템들에 위치시킬 수 있어, 구성이 복잡성을 유발할 수 있지만, 전체적인 성능의 향상을 가져올 수 있다. 이러한 각 방화벽의 구성요소가 되는 시스템들은 특정 동작만을 취하게 인스톨되므로 자체 컨피규레이션은 오히려 단순해진다. 그림 3-4 에 예시되어있는 것처럼 2대의 라우터는 스크린드 서브넷(Screened Subnet)을 구축하는데 사용된다. 이 서브넷은 일명 DMZ 이라고도 하며 이 곳에 응용게이트웨이, 인포메이션 서버들, 모뎀 풀(Modem Pool) 또는 매우 조심스러운 액세스 콘트롤을 요하는 기타 시스템들을 위치 시킬 수 있다. 인터넷에 연결되어있는 라우터는 다음과 같은 룰을 따라 트래픽을 라우팅한다.

- 응용게이트웨이로 부터 인터넷 시스템들로 향하는 트래픽을 라우팅한다.
- 전자메일서버로 부터 인터넷 사이트들로 향하는 전자메일 트래픽을 라우팅한다.
- 인터넷 사이트에서 응용게이트웨이로 향하는 응용트래픽을 라우팅한다.
- 인터넷 사이트에서 전자메일서버로 향하는 전자메일 트래픽을 라우팅한다.
- 인터넷 사이트에서 인포메이션 서버로 향하는 FTP, Gopher 등의 트래픽을 라우팅한다.
- 그밖에 다른 모든 트래픽은 거절한다.

이와 같이 인터넷 쪽에 연결되어있는 라우터는 인터넷으로 부터 Screened 서브넷 위에 있는 특정 시스템들에 대한 액세스를 제한하거나 인터넷으로 연결되지 말아야하는 모뎀 풀, 인포메이션 서버들, 사이트 시스템 등으로 부터 오는 트래픽을 막는 역할을 수행한다. Screened 서브넷 상에 존재하는 호스트 "로 부터 기인한(From)" 또는 "으로 향하는(To)" 필요없는, 보안상의 취약점을 갖고 있는, NFS, NIS 및 기타 프로토콜들의 패킷들도 이 라우터에 의해 걸러진다. 내부 서브넷에 연결되어있는 라우터는 Screened 서브넷 위에 있는 특정 시스템들"로 부터"(From) 또는 "로 향하는"(To) 트래픽을 다음과 같은 룰들에 의해 제어한다.

- 응용게이트웨이로 부터 인터넷 시스템들로 향하는 트래픽을 라우팅한다.
- 전자메일서버로 부터 내부시스템들로 향하는 전자메일 트래픽을 라우팅한다.
- 내부 시스템에서 응용게이트웨이로 향하는 응용트래픽을 라우팅한다.
- 내부 시스템에서 전자메일서버로 향하는 전자메일 트래픽을 라우팅한다.
- 내부 시스템에서 인포메이션 서버로 향하는 FTP, Gopher 등의 트래픽을 라우팅한다.
- 그밖에 다른 모든 트래픽은 거절한다.

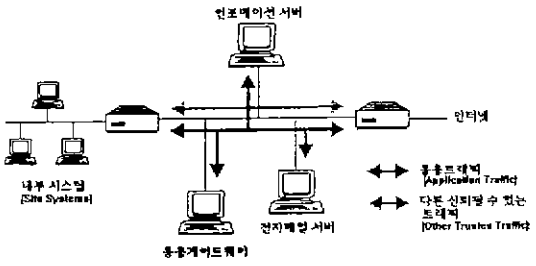

그림 3-4 Screened Subnet Firewall

그러므로 Screeened 서브넷 방화벽이 설치가 되면 내부 시스템(Site Systems)들은 인터넷으로 부터 직접적인 접촉은 불가능하며 그 반대 경우도 마찬가지 상황이 된다. 이점은 Dual-homed 게이트웨이 방화벽 경우와 동일하다. 그렇지만, 라우터들이 특정 시스템들로 트래픽을 향하게 함으로서 응용게이트웨이가 Dual homed 네트웍 인터페이스를 갖지 않아도 된다. 특히, 내부 네트웍을 위한 게이트웨이로 라우터가 사용되므로 향상된 throughput 을 얻을 수 있다. 이러한 연유로 Screened 서브넷 방화벽은 많은 트래픽이 있는 또는 매우 빠른 트래픽 처리를 요하는 사이트에 설치하는 것이 바람직하다. 그리고 해커들이 사이트 시스템들에 침입하기 위해서는 두 대의 라우터를 통과해야 하기 때문에 보안이 강화된 효과를 얻을 수 있다. 응용게이트웨이, 전자메일서버, 인포메이션 서버들을 제외한 나머지 다른 시스템들은 인터넷 사이트에 알려질 필요가 없다. 응용게이트웨이는 모든 inbound connection 들을 인증할 수 있는 인증 소프트웨어를 장착할 수도 있다. 이러한 인증 소프트웨어를 응용게이트웨이에 설치함에 다라 구성이 복잡해지는 결과를 낳을 수 있지만 응용게이트웨이와 패킷 필터링을 위한 시스템들을 서로 분리시켜 구성하게 되므로 결과적으로는 컨피규레이션이 더 간단해지고 관리하기가 용이해진다.

한편, Screened 서브넷 방화벽은 Screened Host 방화벽처럼 신뢰할 수 있는 서비스들(Trusted Services)을 사이트 시스템들로 라우팅할 수 있는 유연성을 갖고 있다. 그러나 이러한 유연성이 방화벽의 보안 성능을 약화시키는 요인이 될 수 있음을 유의하여야 한다. 이러한 점을 고려하면 역시 Dual-homed 게이트웨이 타잎의 방화벽 시스템이 보안을 위해서는 더 바람직한 시스템이라고 할 수 있다. 왜냐하면 Dual-homed 게이트웨이에 프록시 서버가 없는 서비스들은 결코 내부 사이트로 통과시킬 수 없기 때문이다. 일반적으로 거의 모든 사이트는 X 윈도우나 NFS 서비스의 트래픽이 내부 서브넷과 인터넷사이

에 존재하지 않기를 바라나 부득이 하게 이를 허용하여야 하는 경우가 발생하기도 한다. 이러한 때, Screened 서브넷 내에 이러한 서비스를 허용하여야 하는 시스템들을 위치시키면 보안정책 훼손을 최소화시킬 수 있을 것이다.

Screened 서브넷 방화벽이 갖고 있는 다른 결점은 보안정책을 실현하기 위하여 라우터 사용에 치중하는 경향이 있다. 패킷 필터링 라우터는 configuration 하기가 해당 사이트에 따라 매우 복잡해지는 경우가 있고 이때 발생할 수 있는 configuration 상에 실수는 내부 시스템 전부를 보안사고 위험에 처하게 할 소지가 있다는 것이다. 이러한 점을 종합하면 Screened 서브넷 방화벽은 throughput 과 유연성을 동시에 요하는 사이트에 적합하다는 결론을 내릴 수 있다.

4. PC를 이용한 방화벽 구축

최근 인터넷 사용자 층 및 인원의 폭발적인 증가와 함께, 컴퓨터 네트웍분야는 지난 5년간 엄청난 발전을 하였다. 한편, 이러한 고도 성장의 그늘에서 자라난 독버섯(해커)들에 의한 폐해가 근래에 이르러 심각한 문제가 되고 있다. 이러한 시점에서 해커들의 침입을 효과적으로 막을 수 있는 방화벽(Firewall) 시스템이 전산망 보안 툴의 핵심으로 떠오르고 있다. 이러한 상황에 발맞춰 많은 상용 방화벽 시스템들이 소개되고 있다. 그 중에서 몇 가지 예를 들어 보자. 대표적인 선(SUN)소프트사의 Firewall-1(6천만원 정도), TIS(Trusted Information Systems, Inc.) Gauntlet(PC586 베이스의 방화벽 시스템으로 가격은 15,000$)등을 예로 들 수 있다. 한마디로 상당히 고가(high price)라고 아니할 수 없다. 그러나 인터넷에는 손쉽게 구할 수 있는 프리웨어 프록시 서버 프로그램(SOCKS, TIS Firewall Toolkit)들이 있다. 그러므로, 본논문에서는 이들 방화벽 프록시 서버 프로그램들 중 하나인 SOCKS를 이용하여 PC방화벽(Dual-honed Gateway 타잎)을 구축하는 방법에 관하여 서술하기로 한다. 물론, SOCKS 를 이용하여 Screened Host 타잎 방화벽도 구축할 수있으나 본 논문에서는 다루지 않기로 한다.

4.1 PC 방화벽 구축을 위한 준비

4.1.1 하드웨어 및 소프트웨어

하드웨어 (H/W)	PC	○CPU : i486 이상 ○Memory : 16Mbates 이상 ○HDD 540Mbytes 이상
	LAN 카드	○3C509, NE2000, SMC-Elite 등과 같이 리눅스 시스템에서 드라이버를 제공하는 LAN카드 2개를, 될 수 있으면 같은 제품으로 준비한다.
	허브 (Hub)	○각 부나 팀 단위 시스템의 수량에 맞는 것으로 준비하면 된다. (한 예로, 10대 미만의 시스템을 보유하고 있을 경우, 12port(10base-T) 커넥터를 보유한 것으로 준비하면 적당) ○물론, thin cable을 이용할 경우, 허브는 불필요
소프트웨어 (S/W)	OS	○Linux Slackware 2.3.x(Kernel 버전 1.2.8) 이상
	Proxy Server	○SOCKS 4.2(다운로드 장소 : "http : //www.socks.nec.com")

4.1.2 서브넷 설치

현재, 각 팀이나 부에서 LAN을 통하여 인터넷 상에 직접 연결되어 있는 PC나 Workstation(W/S)들은 해커들의 직접적인 공격 대상이 될 수 있다. 이러한 시스템들을 보호하기 위해서는 팀이나 부 단위로 자체 서브넷(Subnet)을 구축한 후, 이 서브넷에 기존에 사용하고 있는 시스템들을 연결하여 사용하는 것이 안전하다. 물론, 이 경우에 방화벽 시스템(PC running Linux)에 부착된 2개의 네트웍 인터페이스(Lancard)들 중, 한 개는 I/P 주소를 획득하여 LAN에 직접 연결한다. 나머지 한 개의 네트웍 인터페이

스는 임의의 C 클라스 인터넷 주소(한 예로 192.168.2.1)로 셋팅한 후, 내부 네트웍을 구성하기 위한 허브에 연결하면 된다. 기존에 사용 중인 PC나 W/S들도 위에서 방화벽이 사용한 192.168.2.x 계열의 C 클라스 IP Address를 임의로 선택하여 Lan 카드를 셋팅한 후, 허브에 연결하는 것으로 서브넷의 구성은 완료된다. 구성 완료된 서브넷의 개념도는 다음과 같다.

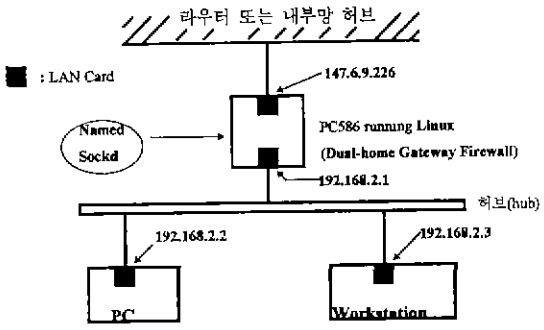

그림 4-1 서브넷 구성 개념도

4.2 방화벽 구축

PC를 방화벽 시스템으로 만들기 위해서는 반드시 PC가 멀티프로세싱을 할 수 있도록 하여야 한다. 그러므로 기존의 PC용 OS인 MS-DOS나 Windows 3.1 시스템은 방화벽 구현에 적합하지 않다. Windows 95가 멀티프로세싱을 하지만 본 논문에서 방화벽을 구축하기 위해서 사용할 프록시 서버 SOCKS가 Window 95를 지원하지 않기 때문에 사용할 수 없다.

이러한 상황에서 가장 접근하기 쉬운 OS로는 리눅스(Linux)를 들 수 있다. 리눅스 시스템은 i386 계열의 PC를 위한 OS로 인터넷 상에서 탄생했다. 초기에는 많은 버그와 불안정으로 인하여 대중화되는데 약간의 시간이 지체되었다. 그러나 인터넷 상에 존재하는 진정한 의미의 해커들에 의하여 지속적으로 수정, 보완된 결과, SLACKWARE 2.3.x에 이르러서는 상당히 안정되어 거의 다운되는 일이 발생하지 않고 있다.

그외에도 퍼블릭 도메인상에 존재하는 유닉스 관련 S/W 소스(Source)들은 거의 모두 리눅스를 지원하고 있다. 물론, SOCKS도 리눅스를 지원한다. 그러므로 "전산망 보호를 위한 PC 방화벽의 구축"에서는 리눅스를 사용하기로 한다.

4.2.1 리눅스 시스템의 설치

리눅스 시스템 설치 방법에 관해서는 Linux Installation HOW-TO라는 메뉴얼을 참조하면 된다. (다운로드 장소 : "http://sunsite.unc edu/mdw/linu.html")

그러나, 방화벽을 위하여 설치할 경우, 기존의 리눅스 시스템 설치와는 다른 다음 몇가지 고려해야 할 사항들이 있다. 즉, 리눅스 시스템을 PC에 설치한 후, 커널을 다음과 같이 다시 컴파일하여야 한다. 커널 컴파일을 하기 위하여 행해야 하는 "make config" 명령 수행 시에 다음과 같은 조건을 주어야 한다.

(1) Turn on Networking Support
(2) Turn on TCP/IP Networking
(3) Turn off IP Forwarding(CONFIG_IP_FORWARD)
(4) Turn on IP Firewalling
(5) Probably turn on IP accounting
(6) Turn on Networking Device Support
(7) Turn on Ethernet support(이때, 설치된 랜카드에 맞는 드라이버를 선택한다.)

그 후에 커널 컴파일 작업은 기존의 방법을 따르면 된다. 이렇게 해서 생성된 커널을 "/vmlinuz"에 카피하여 사용하면 된다. 그런데 기존의 리눅스 시스템 커널은 Ethernet 카드 하나만을 자동감지(auto-detect) 할 수 있게 되어있으므로 "/etc/lilo.conf"

화일에 다음과 같은 명령을 추가하여야 한다.

```
# LILO configuration file
# To detect two ethernet cards
append="ether=5,0x300,eth0 ether=11,0x280,eth1"
```

위에서 ether 변수의 첫번째 항목은 각 ethernet card 의 IRQ 넘버이며, 두번째 항목은 IO Base Address 를 나타낸다. 즉, 첫번째, 두번째 ethernet 카드인 eth0,eth1 는 각각 IRQ 넘버가 5,11 및 IO Base Address 0x300 과 0x280 을 사용하고 있다.

4.2.2. Proxy Server(SOCKS) 설치

리눅스 시스템 위에 Proxy Server(SOCKS) 를 설치하기 위해서는 수퍼유저로 되어야 한다. "http://www.socks.nec.com" 에서 갖고 온 SOCKS 시스템의 압축을 풀어 설치 한다. 그런데 sockd는 inetd 나 stand-alone 모드로 작동시킬 수 있다.

(a) inted 에 의한 작동

SOCKS 가 설치된 디렉토리 내에 "./include/socks.h" 화일을 자신의 네트웍 환경에 맞도록 다음과 같이 수정하여야 한다.

```
#define SOCKS_DEFAULT_SERVER
"firewall.sec_lab.swrc.ktrc" /* 서브넷과 연결되는 LAN 카드 IP 주소 */
#define SOCKS_DEFAULT_NS
"192.168.2.1" /* 서브넷과 연결되는 LAN 카드 IP 주소 */
#define SOCKS_DEFAULT_DNAME
"sec_lab.swrc.ktrc" /* 서브넷의 도메인 이름 */
#define_MULTIHOMED_SERVER
/* 2개 이상의 랜카드 사용시 정의하여야 한다 */
#define SOCKD_ROUTE_FILE
"/etc/sockd.route"
/* #define NOT_THROUGH_INETD */
/* inetd 를 이용하여 sockd 를 작동할때는 정의하지 않는다. */
/*#define MAX_CLIENTS    5 */
/* inetd 를 이용할 때는 정의하지 않는다. */
```

(1) "make install.server" 명령어를 수행한다.

(2) "/etc/services" 화일에 다음 줄의 내용을 추가한다.

```
socks    1080/tcp    #Socks gateway services
```

(3) "./ sockd" 디렉토리에 생성된 sockd를 "/etc"디렉토리에 복사한다.

(4) "/etc/sockd.conf" 화일을 다음과 같이 작성한다.

```
permit      192.168.2.0255.255.255.0
deny        0.0.0.0.0.0.0.0 : /usr/ucb/finger
@%A\ /usr/ucb/mail-s 'SOCKS : rejected
%u@%A to %Z(%s)' root@%A
```

(5) "/etc/inetd.conf" 화일에 다음 줄의 내용을 추가한다.

```
socks  stream tcp  nowait nobody /etc/sockd
sockd
```

(6) "kill - HUP<pid of inetd>" 명령어를 이용하여 inetd 를 재가동시킨다.

b) metd 에 의한 작동

위에서 SOCKS 를 설치하는 방법은 inetd 로 구동되는 Sockd 를 설치하는 방법에 대해서 언급한 것이다. 사실 inetd에 의한 구동은 socks 에 대한 요구사항이 발생했을 때(즉, 서브넷 안에 있는 호스트에서 방화벽을 통하여 인터넷과 교류하고자 할 경우). inetd가 sockd를 wake-up 하여 서비스 하게 함으로서 평상시 방화벽 시스템 자체 자원을 효과적으로 사용하는 방법이지만 서비스해야 할 호스트들이 서브넷 상에 많을 경우 서비스 속도가 현저히 지연될 가능성이다. 이런 경우에는 sockd 를 stand-alone 으

로 사용하는 것이 효과적이다. 그런데 sockd 를 stand-alone 으로 사용하기 위해서는 "./include/socks.h" 화일의 내용중에서 다음 사항을 정의한 후, 다시 컴파일하여야 한다.

```
#define SOCKS_DEFAULT_SERVER
"firewall.sec_lab.swrc.ktrc" /* 서브넷과 연결되는
LAN 카드 IP 주소 */
#define SOCKS_DEFAULT_NS
"192.168.2.1" /* 서브넷과 연결되는 LAN 카드 IP
주소 */
#define SOCKS_DEFAULT_DNAME
"sec_lab.swrc.ktrc" /* 서브넷의 도메인 이름 */
#define_MULTIHOMED_SERVER
/* 2개 이상의 랜카드 사용시 정의하여야 한다 */
#define SOCKD_ROUTE_FILE
"/etc/sockd.route"
#define NOT_THROUGH_INETD
/* sockd 를 작동할때는 stand-alone 으로 작동시
킬 때 정의한다. */
#define MAX_CLIENTS    5
```

이렇게 컴파일된 sockd는 "/etc/rc.d/rc.inet2" 화일에 다음과 같은 shell 명령어들을 추가시켜 시스템 부팅시 자동으로 동작하게 한다.

```
NET = "/usr/sbin"
if [-f$(NET)/sockd] ; then
        echo - n "sockd"
        $(NET)/sockd
fi
```

4.2.3 Name Server 설치

Dual-homed Gateway 방화벽 뒷단에서 보호를 받고 있는 시스템들은 방화벽 역할을 하는 Linux 시스템(PC586)이 패킷들을 Kernel 레벨에서 포워딩하지 않는 이유로 인하여 Domain Name Service 를 받을 수 없다. DNS 는 TCP 와 UDP프로토콜을 동시에 사용하므로 sockd가 이를 중개해줄 수가 없다. 이러한 문제점을 해결하기 위해서 방화벽 시스템 위에 Name Server 를 설치하는 것이 바람직하다. 이 Name Server 가 방화벽 뒷단에서 오는 호스트 이름에 대한 IP 주소 쿼리를 resolve 해 준다. 본 논문에서는 Name 서버 147.6.1.2 로 DNS 쿼리를 포워딩하는 Name Server 를 방화벽에 설치하였다.

4.2.4 서브넷 상 시스템들의 환경 설정

서브넷에 연결되어있는 PC나 W/S 들이 위에서 설치된 리눅스 방화벽 시스템을 통과하여 외부 네트웍(Internet)과 통신하기 위해서는 다음과 같은 조치를 하면 된다. 단, SOCKS 4.2 는 TCP 프로토콜 만을 지원하고 있어서 NFS 등과 같은 UDP 서비스를 받을 수가 없다.

(a) PC(MS 윈도우 3.1과 '95)

기존의 윈도우 3.1에서 작동하는 application 들 중에 socks 를 지원하는 것(Netscape, win_tel, win_ftp)들이 있으나 이들은 각각 자신들의 옵션들을 셋팅하여야 하는 번거로움을 갖고 있다. 그러나 socks 를 지원하는 Trumpet Winsock에서 socks를 사용할 수 있게 하는 옵션을 active 시키면 위에서 언급한 application 들의 옵션들을 셋팅할 필요없이 그대로 사용할 수 있다. 즉, socks를 지원하지 않는 응용프로그램도 방화벽 뒤에서 사용할 수 있다.

아래 그림과 같이 Trumpet Winsock의 File 메뉴에서 Firewall Setup 항목을 선택한 후, Enable Firewall을 선택한 다음, Firewall Host IP Address에 리눅스 방화벽 시스템의 서브넷 상에 연결되어있는 Lan 카드의 IP Address(192.168.2.1)를 설정한다.

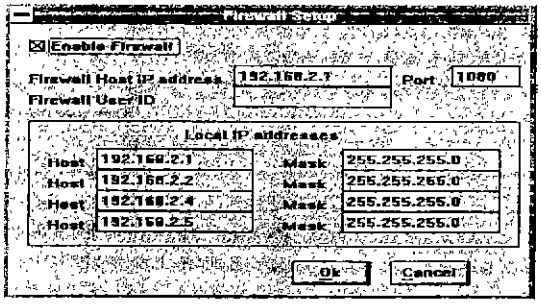

그림 4-2 Trumpet Winsock 에서 Firewall Setup

Win '95 의 경우에는 자체적으로 Winsock을 갖고 있으나 이들은 SOCKS 를 지원하지 않는다. 그러므로 네트웍 관련 응용프로그램들이 방화벽을 통과할 수 가 없다. 그러나 Win '95용 Netscape 경우 SOCKS 를 지원하므로 Netscape 의 Options 메뉴에 있는 Network Preferences 항목을 선택한 다음, 프록시 항목에서 VIEW 버튼을 누르면 다음과 같은 윈도우가 나타난다. 여기서 항목에 아래 그림과 같이 입력하면 Netscape 가 sockd 가 설치된 방화벽을 통과하여 인터넷으로 나갈수 있게 된다

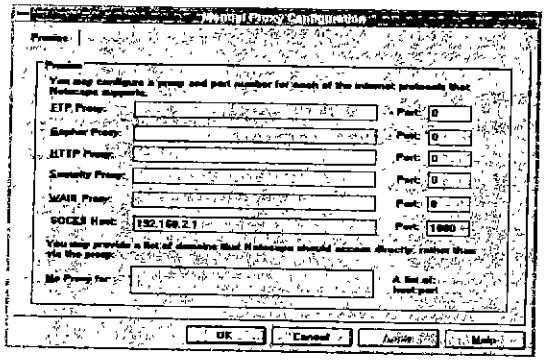

그림 4-3 Netscape 의 Socks Host 지정 윈도우

그림 4-3의 SOCKS Host 항목에서 192.168.2.1은 sockd 가 설치된 방화벽의 랜카드 중에서 내부 서브넷에 연결된 IP 주소를 지칭하며 Port 넘버 1080은 sockd 가 사용하고 있는 Port 넘버를 나타내고 있다.

그러나 Netscape 를 제외한 다른 응용프로그램은 방화벽을 통과할 수 없다. 그러므로 본 논문에서 구축한 방화벽 뒷단에서 네트웍 관련 응용프로그램을 사용하는 방법을 Netscape 와 같이 socks를 지원하는 application 등을 사용하거나 Win '95용 Trumpet Winsock를 사용하여야 한다.

(b) W/S(Linux PC포함)

SOCKS 패키지 속에 포함되어있는 rtelnet, rftp, rfinger 를 기존의 telnet, ftp, finger 대신에 사용하면 된다. 웹서비스를 받기 위해서는 Netscape 의 Option 항목에 있는 Proxies 항목의 SOCKS Host 필드에 Linux 방화벽 시스템의 IP 주소를 입력하면 된다. 단, 아래의 내용을 "/etc/socks.conf" 화일에 입력하여야 한다.

direct 192.168.2.0 255.255.255.0

sockd @ = 192.168.2.1 0.0.0.0 0.0.0.0

위 내용 중에 direct로 시작되는 첫번째 줄은 내부 서브넷에 연결되어있는 192.168.2로 시작하는 IP 주소를 갖는 호스트들은 sockd를 경유하지 않고 상호간에 직접 연결하게 한다는 것을 뜻한다. 그리고 두번째 줄은 목적지가 내부(192.168.2.x)가 아닌 외부(인터넷)로 나갈 때는 sockd 가 설치된 방화벽을 경유하게 한다는 것을 명시한 것이다.

5. 결론

본 논문에서는 인터넷 상에서 존재하는 보안상의 문제점들에 관하여 간략하게 언급하였으며, 이러한 문제점들을 극복할 수 있는 방안 중의 하나인 방화벽 시스템에 관하여 주로 기능 및 종류에 대하여 비교적 상세하게 서술하였다. 끝으로 방화벽을 구축, 사용할 수 있도록 방화벽 프록시 서버 프로그램인 Socks 4.2 이용하여 PC 방화벽을 구축하여 실전에 투입하는 방법에 대하여 상세히 기술하였다. 이러한

방법으로 구축된 방화벽은 내부 서브넷에 존재하는 기존의 PC나 웍스테이션들을 인터넷 상에 노출시키지 않을 수 있어, 외부의 해커들로 부터 시스템들을 보호할 수 있게 된다. 일반적으로 방화벽 시스템을 구축하지 않고 Gateway 만을 외부에 연결한 다음 서브넷을 구성했을 경우, 서브넷 내에 연결되어있는 시스템들이 인터넷으로 나가고자 할 때는 반드시 Gateway 에 login 한 다음, 그 곳에서 인터넷과 접속하여야만 했기 때문에 불편했다. 그러나 앞서 서술한 방화벽을 설치한 후에는 외부로 부터의 침입을 효과적으로 막을 수 있을 뿐만 아니라 서브넷 안에 있는 시스템에서 가보고자 하는 인터넷 사이트 이름을 직접 입력하므로서 바로 연결될 수 있기 때문에 사용하기 편리할 것이다. SOCKS 버전 4의 경우 UDP 를 지원하고 있지 않아 불편한 점이 없진 않지만 일반적으로 인터넷 사용자들이 자주 사용하는 Telnet, ftp, finger, Netcape 등이 TCP 프로토콜을 채택하고 있기 때문에 사용 상의 불편은 별로 없을 것이다. 더우기 UDP패킷을 차단하는 것은 보안에 있어서 상식으로 간주되고 있다. Socks 버전 5에서는 UDP 도 지원하지만 아직 베타 버전에 머물고 있다. 물론, 앞서 언급한 SOCKS 는 Linux 에 설치하는 방법과 동일하게 웍스테이션에도 설치하여 방화벽으로 사용할 수 있다. 이러한 방화벽을 위한 플랫폼의 결정은 보호해야될 내부 서브넷의 규모나 내부 사용자들의 인터넷 사용 빈도에 따라 결정되어야 될 사항이다.

요즈음 많은 보안관련 상품들이, 특히 방화벽 시스템들이, 소개되고 있는데 이들의 가격들이 상당히 고가(대체로 수천만원대)로 전부가 수입품들이다. 더우기 이들은 전부가 바이러리(binary) 형태로 제공되고 있기 때문에 트로얀 호스(Trojan Horse)의 존재 가능성을 배제할 수 없다. 그러나 보안에 관한 옳바른 인식의 부족으로 인하여 이러한 제품들을 구입하여 설치하면 보안 문제가 해결되는 것으로 알고 있는 사이트도 많은 것 같아 안타까움을 금할 수 없다. 사실, 보안을 위한 투자도 다른 종류의 투자와 마찬가지로 최소 비용으로 최대의 효과를 거두어야 한다. 이러한 점을 감안할 때, 본 논문에서 구축하고자 하는 방화벽은 같은 기관내에서 정보의 흐름을 통제하는데 또는 소규모 내부망을 인터넷에 연결하고자 하는 사이트들에게 유용하게 쓰일 수 있을 것으로 생각된다. 그리고 아직 개선의 여지가 많음을 인정하지 않을 수 없다. 앞으로 개선 및 추가되어야 할 기능으로 다음과 같은 것 들이 있다. 즉, 사용자의 이용 편리를 위한 GUI 개발, 일회용 패스워드를 이용한 인증기법, 방화벽과 방화벽 사이에서 전송되는 데이타 보호를 위한 가상 사설망(Virtual Private Network)및 실시간 침입감지 기능 장착 등을 언급할 수 있을 것이다.

참 고 문 헌

1. Steven M. Bellovin. Security Problems in the TCP/IP Protocol Suite. Computer Communications Review, 9(2) : 32-48, April 1989.
2. Computer Emergency Response Team/Coordination Center. CA-94 : 01, Ongoing Network Monitoring Attacks. Available from FIRST.ORG, file/pub/alerts/cert9401.txt, Feb. 1994.
3. D. Brent Chapman. Network(In) Seurity Through IP Packet Filtering. In USENIX Security Symposium III Proceedings, pages 63-76. USENIX Association, September 14-16 1992.
4. William R. Cheswick and Steven M. Bellovin. Firewall and Interet Security. Addison-Wesley , Reading, MA, 1994.

5. Computer Incident Advisory Capability. Number e-07, unix sendmail vulnerabilities update. Available from FIRST. ORG, file /pub/alerts/e-07. txt, January 1994.
6. Computer Incident Advisory Capability. Number e-14, wuarchive ftpd trojan horse. Available from FIRST. ORG, file /pub/alerts/e-14. txt, February 1994.
7. David Koblas and Michelle R. Koblas. SOCKS.
8. Dan Farmer and Wietse Venema. Improving the security of your site by breaking into it. Available from FTP. WIN. TUE. NL, file / pub/alerts /e-14. txt, Ferbruary 1994.
9. Simpson Garfinkel and Gene Spafford. Practical UNIX Security. O' Reilly and Associates, Inc., Sebastopol, CA, 1992.
10. John P. Wack and Lisa J. Carnahan. Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls, NIST Special Publication 800-10.
11. Marcus J. Ranum. Thinking About Firewalls. In SANS-II Conference, April 1993.
12. Marcus J. Ranum and Federick M. Avolio. TIS Toolkit and Methods for Internet Trusted Information Systems. Inc.
13. Larry J. Hughes. Jr. Actually Useful Internet Security Techniques, New Riders Pub., 1995.
14. Wietse Venema TCP WRAPPER Networking monitoring, access control. and booby traps.

백 석 철

1982년 2월 서울대학교 물리교육학과 졸업(학사)
1985년 2월 한국과학기술원 물리학과 졸업(이학 석사)
1985년 4월 한국통신 사업지원단 입사(전임 연구원)
1985~1987 한국통신 업무자동보고시스템(ARS) 개발
1988~1990 한국통신 연구개발본부 IBM 대형시스템 MVS 시스템 매니저
1995년 8월 한국과학기술원 물리학과 박사과정 졸업(이학 박사)
1996년 현재 한국통신 연구개발본부 멀티미디어 연구소 인터넷 시스템 개발팀 보안 과제 책임자

저서 : - U(1)xU(1) 및 SU(2)xU(1)xU(1) 장이 중력에 미치는 영향(석사논문)
- 지구자기꼬리(Geomagnetotail) 내에 존재하는 이온(Ion)의 비선형 동력학(Non-linear Dynamaics)에 관한 연구(박사학위 논문)
- ARS 시스템 개발 보고서, MVS 시스템 운영 지침서, 전산망 보안시스템 개발 선행연구 보고서 등 다수