네트워크상에서 바이러스 차단을 위한 방화벽 시스템의 설계 및 구현

최 준 호† · 김 판 구††

요 약

컴퓨터 시스템의 데이터 파괴 및 시스템 정지 등과 같은 악의적인 일을 수행하는 컴퓨터 바이러스의 종류 및 피해가 증가되고 있으며, 이에 따른 바이러스 검사 및 치료에 있어서 현재보다는 다양한 대응 방법이 요구되고 있다. 기존의 대부분의 바이러스 대처 방법으로는 악성 코드 패턴을 탐색하여 치료하는 백신과 같은 프로그램을 사용하며, 또한 클라이언트용으로 Windows 시스템에서 주로 사용되어 오고 있다. 그리고 Linux와 같은 Unix계열의 시스템에서는 파일 시스템에 파일을 저장하고 난 후 백신으로 치료하는 형태를 가지고 있어 바이러스 확산에 능동적으로 대처하지 못하고 있다. 본 논문에서는 네트워크를 통해 유입되는 데이터에 대해 바이러스 감염 여부를 실시간으로 탐지하여 그 결과를 웹상에 알려주는 방화벽 시스템을 설계 및 구현하였다. 이 시스템은 기존 방화벽 보안 정책을 유지하면서 네트워크에서 유입되는 데이터에 대해 바이러스를 실시간으로 탐지해낸다. 방화벽 시스템에 대해 바이러스 탐지 모듈을 추가한 결과, 기존보다 평균 8%정도의 시간적 지연이 있음을 확인하였다.

Design and Implementation of a Firewall System for Blocking Viruses in the Network

Jun-Ho Choi† · Pan-Koo Kim††

ABSTRACT

Many kinds of viruses have been occurring in the computer systems. These are able to destroy some critical data and even make the computer system halt their function. Consequently, its damage was enormous. So we need a more efficient method that detects the computer viruses, different from the existing one. Most of the existing method to cope with the viruses are to use anti-virus program that probes malicious code patterns and get rid of them from the infected file. This kind of program is usually launched on the Microsoft Windows system for users. And also, for the reason that anti-virus program in the Unix-like operating system scans the malicious code patterns into the infected files after storing them firstly in their file system, it seems that this can not actively deal with virus spreading. In this paper, we have designed and implemented a firewall system that diagnoses the incoming files through the network in real-time and then reports the results by the HTML file on the Web. This system can detect the incoming data over the network keeping the existing firewall security policies and function. As a result of experiment, we found that our proposed system has the 8% of processing time overhead when adding virus probing module to the established firewall system.

키워드 : 방화벽(Firewall), 바이러스(Virus)

1. 서 론

인터넷 사용이 증가함에 따라 전산망에 대한 불법적인 침입이나 바이러스와 같은 시스템의 파괴, 오동작을 일으키는 악성 소프트웨어로 인한 피해가 날로 증가하고 있다. 악성 소프트웨어에는 컴퓨터 바이러스(Computer Virus), 트로이 목마(Trojan horse), 웜(Worm), 논리 폭탄(Logic Bomb), 백 도어(Back door) 등이 있는데 최근 시스템에 많은 피해를 준 '러브레터 바이러스'와 그 변종들은 컴퓨터 바이러스의 대표적인 형태이다[5, 14]. 이러한 바이러스를 퇴치하기 위한 백신의 대부분은 Windows 클라이언트용으로 제작, 배포되는 반면에 Linux나 Unix용 백신 소프트웨어는 고가이면서 그 숫자 또한 극히 드문 형편이다. 또한, 백신 소프트웨어는 파일이 로컬 시스템에 저장된 후 관리자에 의해 검사되는 선 저장-후 검사 방법이므로 바이러스의 침입에 적극적으로 대응을 못하고 있는 실정이다. 따라서 바이러스

† 준 회 원 : 조선대학교 대학원 전자계산학과
†† 정 회 원 : 조선대학교 컴퓨터공학부 교수
논문접수 : 2001년 5월 18일, 심사완료 : 2001년 6월 25일

검사의 문제점을 극복하기 위해서는 로컬 시스템의 검사와 더불어 바이러스 유입의 주요 경로가 되고 있는 네트워크 상에서의 바이러스 검사가 병행되어야 한다.

일반적으로 네트워크에 연결된 호스트의 주요 피해는 허용되지 않은 사용자가 특정 호스트에 침입하여 정보의 노출이나 손상을 시키는 행위를 하고, 시스템 장애를 유발하여, 정상적인 사용자 이용에 지장을 주는 것이다. 시스템 이용이 허용되지 않는 접근을 차단할 수 있는 가장 효과적인 보안 대책은 허가된 네트워크와 사용자들로부터의 접근 요구와 서비스 제공에 대해 해당 사용자의 활동 상황을 감시하고 기록하는 등의 내부 통제도 함께 필요하게 된다. 방화벽은 내부 네트워크와 외부 네트워크의 사이에서 양자간의 연결점 역할을 수행하는 시스템을 의미하는데 내부 네트워크와 외부 네트워크의 모든 통신은 방화벽을 거쳐야 하며, 방화벽은 양자간에 오가는 모든 통신을 감시하여, 허용되지 않는 접근을 막는다. 이로써 내부 네트워크는 불법적인 네트워크 침입이나 해킹으로부터 보호된다. 내·외부간 모든 통신은 방화벽을 반드시 거쳐야만 이루어지게 되므로 방화벽 자체는 중요한 정보를 갖지는 않지만 외부로부터의 접근이 대부분 봉쇄되어 침입의 위험으로부터 보호된다. 따라서 방화벽 시스템에 바이러스 감염 여부를 진단할 수 있는 기능을 도입할 경우 네트워크 보안을 유지하며 네트워크 상에서 전송되는 바이러스의 진단과 감염에 대한 치료 및 감염 파일을 차단함으로써 시스템 및 파일 보호 측면에서 큰 효과가 있을 것으로 예상된다. 이에 본 논문에서는 기존 방화벽 시스템의 설계 방법론을 분석하여 바이러스를 탐지할 수 있는 방화벽 시스템 설계를 위한 요구사항과 모듈을 제시한다.

본 논문의 구성은 다음과 같다. 제2장에서는 방화벽의 구현 모델별 특성을 분석을 기술하고, 제3장에서는 본 논문에서 구현될 시스템의 전체 구성과 방화벽 구축을 위한 툴킷, 그리고 네트워크 접근 제어 규칙에 대해 설명한다. 또한 바이러스 탐지 및 차단 모듈을 설계하고 방화벽과의 연동을 기술한다. 제4장에서는 구현된 시스템의 바이러스 탐지 및 차단 과정을 소개하고, 이에 대한 네트워크상의 성능을 분석하며 제5장에서 결론을 맺는다.

2. 방화벽 구현 모델별 특성 분석

방화벽 시스템은 동작하는 프로토콜 계층에 따라 OSI 계층 3인 네트워크 계층과 계층 4인 트랜스포트 계층에서 패킷필터링 기능을 수행하는 스크리닝 라우터와 응용 계층에서 패킷필터링 기능과 인증 기능 등을 수행하는 응용 계층의 게이트웨이로 분류할 수 있다[3-4].

2.1 패킷필터링 방식

패킷필터링 방식의 방화벽의 장점은 방화벽 기능이 OSI 7 모델에서 제3, 4계층에서 처리되기 때문에 다른 방식에 비해 처리속도가 빠르며, 사용자에게 투명성을 제공한다. 또한 기존에 사용하고 있는 응용 서비스 및 새로운 서비스에 대해서 쉽게 연동할 수 있는 유연성이 있다. TCP/IP 프로토콜의 구조적인 문제 때문에 TCP/IP 패킷의 헤더는 쉽게 조작 가능하다. 따라서 외부침입자가 이러한 패킷의 정보를 조작한다면 내부시스템과 외부시스템이 직접 연결된다. 또한 FTP, Mail에 바이러스가 감염된 파일 전송 시 잠재적으로 위험한 데이터에 대한 분석이 불가능하며 접속제어 규칙의 개수 및 접속제어 규칙 순서에 따라 방화벽에 부하를 많이 줄 수 있다. 그리고 다른 방식에 비해서 강력한 로깅 및 사용자 인증 기능을 제공하지 않는다.

2.2 어플리케이션 방식

어플리케이션 게이트웨이는 OSI 네트워크 모델의 어플리케이션 계층에서 방화벽 기능을 수행한다. 이 는 각 서비스별로 Proxy Daemon이 있어 프록시 게이트웨이 또는 응용게이트웨이라고도 한다. 어플리케이션 게이트웨이는 각 서비스별 프록시를 이용하여 패킷 필터링 방식처럼 IP 주소 및 TCP 포트를 이용하여 네트웍 접근제어를 할 수 있으며 추가적으로 사용자 인증 및 파일 전송시 바이러스 검색기능과 같은 기타 부가적인 서비스를 지원한다. 프록시는 클라이언트와 서버사이에 존재하여 그 접속을 관리하며 이미 접속된 연결에 대해서는 데이터 전달을 위한 전달자로서 기능을 한다. 따라서 클라이언트는 프록시를 통해서만 실제 서버로의 데이터를 주고받을 수 있다. 즉, 클라이언트와 실제 서버간에 직접적인 연결을 허용하지 않는다.

2.3 Circuit Gateway

서킷 게이트웨이는 OSI 네트웍 모델에서 5계층에서 7계층 사이에 존재하며 어플리케이션 게이트웨이와는 달리 각 서비스별로 프록시가 존재하는 것이 아니고, 어느 어플리케이션도 이용할 수 있는 일반적인 프록시가 존재한다.

방화벽을 통해서 내부 시스템으로 접속하기 위해서는 먼저 클라이언트에 서킷 프록시를 인식할 수 있는 수정된 클라이언트 프로그램이 필요하다. 따라서 수정된 클라이언트 프로그램이 설치되어있는 클라이언트만 circuit 형성이 가능하다.

2.4 Hybrid 방화벽

여러 유형의 방화벽들을 경우에 따라 복합적으로 구성할 수 있는 방화벽이다. 이 방화벽은 서비스의 종류에 따라서 사용자의 편의성, 보안성 등을 고려하여 방화벽 기능을 선택적으로 부여할 수 있지만 서비스의 종류에 따라서 다양한 보안정책을 부여함으로써 구축 및 관리하는데 어려움이 따를 수 있다.

3. 네트워크상의 바이러스 탐지 시스템 설계

본 장에서는 제2장에서 제시한 방화벽 시스템의 구현 모델의 특징을 분석하여 네트워크 상에서 바이러스를 탐지하고 차단하는 시스템의 방법론을 제시하고, 이를 적용할 방화벽 시스템의 설계 시 고려해야할 사항에 대하여 기술한다.

3.1 시스템 설계 기본 정책

방화벽 시스템은 네트워크 보안의 수준과 서비스와 상관관계를 가지고 있으므로 이에 따라 방화벽의 설계와 시스템 구축 형태가 달라진다. 방화벽 시스템의 정책에는 크게 두 가지 모델이 있는데, 첫 번째는 서비스 접근 정책에서 서비스를 거부할 서비스로 정의된 서비스를 제외한 모든 서비스를 허용하는 보안 정책이고, 두 번째는 이와 반대로 허용할 서비스로 정의된 서비스를 제외한 모든 서비스를 거부하는 보안정책이며, 일반적으로 정보 보호 분야에서 일반적으로 사용되는 모델이다.

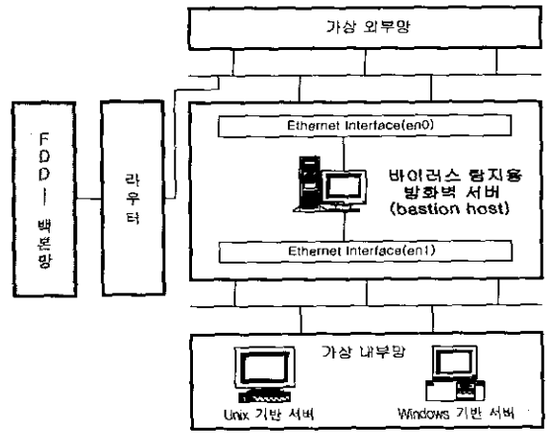
네트워크 상에서 바이러스를 차단하는 시스템은 바이러스를 탐지하기 위해 처리하는 시간을 최소화시켜야 한다. 이를 위해 첫째, 네트워크 부하가 발생하지 않는 방향으로 설계되어야 한다. 시스템의 내부 자원을 보호하기 위하여 허용되지 않는 서비스 이외에는 접근할 수 없도록 고려해야 한다. 둘째, 접근이 허용되는 서비스는 Telnet, FTP, WWW로 제한한다. 내부의 호스트가 외부의 Telnet 또는 FTP 서비스를 사용할 경우에는 먼저 Bastion Host의 Proxy Telnet 또는 FTP Gateway에 접속을 한다. 이때 사용자 인증을 요구하게 된다. 사용자 ID와 패스워드를 입력하여 사용자 확인을 마친 후에 원하는 서비스에 재 접속하는 방식으로 서비스를 사용할 수 있다. 따라서, FTP 서비스를 통하여 파일을 전송 받게 될 경우 파일 전송 후 전송된 파일이 바이러스에 감염된 파일인지를 감시하여 이를 통보하여 주도록 설계하였다. 외부의 호스트가 내부에 호스트에 Telnet이나 FTP를 이용하려고 하는 경우에도 마찬가지로 Bastion Host에 먼저 접속을 한 후에 다시 원하는 Host로 재 접속해야 한다.

3.2 시스템 요구사항 분석

수립된 기본 정책에 따라 본 논문의 바이러스 탐지 모듈 이식 방법론, 시스템의 성능을 정의한다.

3.2.1 바이러스 탐지 모듈 적용

Bastion Host의 Proxy Server는 사용자 단계와 응용 프로토콜 단계에서 액세스 제어를 제공 할 수 있고, 응용 프로그램의 사용에 대한 기록을 유지하고 감시 추적을 위해서도 사용될 수 있다. 응용 게이트웨이는 사용자 단계에서 입출력되는 모든 트래픽에 대한 기록을 관리하고 제어할 수 있으며, 해커 및 불법 침입자를 방어하기 위해서 강력한 인증 기법이 필요하다. 응용 게이트웨이는 사용되는 응용 서비스에 따라 각각 다른 소프트웨어를 구현하여 사용하기 때문에 고수준의 보안을 제공할 수 있다. 이는 네트워크에 첨가되고 보호가 필요한 새로운 응용이 생기면 이를 위해 새로운 특수 목적용 코드를 생성할 수 있다. 이에 본 연구에서의 네트워크로 유입되는 파일의 바이러스 탐지 및 감염 파일 차단 모듈은 Bastion Host 시스템의 응용 게이트웨이 수준에 탑재되어 구현되도록 한다.



(그림 1) 바이러스 탐지 및 차단 방화벽 시스템의 네트워크 구성도

3.2.2 시스템 성능 기준

컴퓨터 보안 표준(TCSEC : Trusted Standard Evaluation Criteria-Orange Book)에서는 물리적 보안, 운영체제의 신뢰성, 서로 다른 사용자의 유형 등을 기술하고 있다.

본 연구에서 설계될 바이러스 탐지 방화벽 시스템은 C2 이상의 보안 등급을 만족할 수 있도록 구성한다. C2 보안 등급은 접근 환경을 제어하는 부가적인 보안 형태를 가지고 있는데, 이는 보안과 관계된 로그 데이터를 저장해 두어서 필요할 때 보안감사를 할 수 있는 보안 등급이다.

바이러스 탐지 방화벽 시스템은 선택적 접근 제어(DAC : Discretionary Access Control) 기능과 감사추적 기능을 포함하고, 다음과 같은 기준을 만족할 수 있도록 설계한다.

- 접근 금지에 대한 설계정책 준수
- 유지보수가 쉬운 간결한 구조로 설계
- 트래픽과 불법적인 접근을 기록하는 기능
- 프록시를 통한 인터넷 서비스
- 사용자 인증에 대한 높은 신뢰도

3.3 시스템 설계

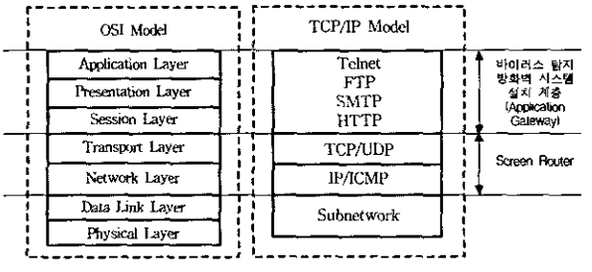
본 논문에서 제안되는 시스템은 바이러스 탐지 모듈에 대한 지속적인 성능 개선과 확장을 위해 기능을 모듈화하여 시스템을 설계하였다. 이는 시스템 설계의 기본 정책과 요구사항을 만족할 수 있도록 응용 게이트웨이 방식으로

설계하였다.

네트워크상의 바이러스 탐지 방화벽 시스템의 시스템 모델과 내부 구조, 처리 흐름은 다음과 같다.

3.3.1 시스템 모델링 정의

본 연구에서 설계된 시스템의 바이러스 탐지 기능과 그에 대한 서비스가 TCP/IP 네트워크 구조의 어느 계층에서 구현되어야 하는지를 다음 (그림 2)와 같이 표현할 수 있다. 이는 OSI 모델의 상위 3계층에 해당하는 TCP/IP 응용층에서 구현되어 인증, 접근제어, 바이러스 탐지 및 차단 기능을 제공하는 응용 게이트웨이 방식으로 설계되었다. 이러한 구성은 사용자 서비스 수준에서 모듈 적용이 용이하므로 쉽게 바이러스 탐지 기능뿐만 아니라 또 다른 보안 모듈을 적용할 수 있다.



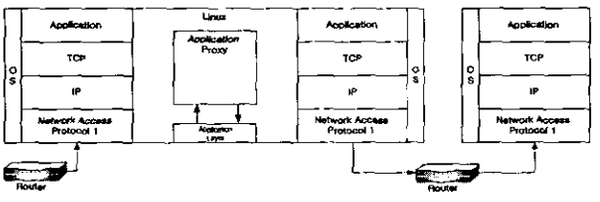
(그림 2) 바이러스 탐지 시스템 계층 모델

3.3.2 시스템 구조

응용 게이트웨이는 사용되는 응용 서비스에 따라 소프트웨어를 구현하여 사용하기 때문에 고수준의 보안을 제공할 수 있다. 응용 레벨 게이트웨이를 사용하기 위해서 사용자는 응용 게이트웨이 장치에 로그인하거나 서비스를 이용할 수 있는 특수한 클라이언트 응용 서비스를 실행해야 한다. 각각 응용에 따라 다르게 사용하는 특수한 게이트웨이는 제각기 내부에 관리 도구와 명령 언어를 가지고 있다.

응용 게이트웨이는 실제 서버의 관점에서 볼 때 클라이언트처럼 동작하며, 클라이언트 관점에서 볼 때는 실제 서버처럼 동작한다. 응용 게이트웨이의 실현 예는 TELNET 게이트웨이, FTP 게이트웨이, Sendmail, NNTP News Forwarder 등이 있다.

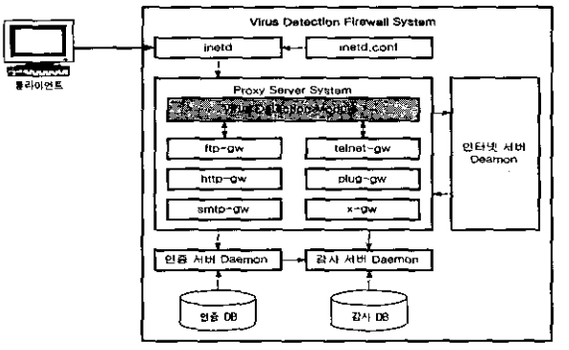
바이러스 탐지 모듈은 응용 게이트웨이에 모듈화되어 탑재되므로 사용자 단계와 응용 프로토콜 단계에서 액세스 제어를 제공 할 수 있고, 응용 프로그램의 사용에 대한 기록(log)을 유지하고 감시 추적을 위해서도 사용될 수 있다.



(그림 3) Application(Proxy) Gateway에서의 바이러스 탐지 모듈 구성

3.3.3 시스템 처리 흐름(시나리오)

본 연구에서 설계된 방화벽 시스템의 바이러스 탐지 모듈 동작 시나리오는 다음과 같다.



(그림 4) 바이러스 탐지 방화벽 시스템 처리 흐름

① 내부 사용자가 외부 인터넷으로부터의 파일 전송을 위해 방화벽 호스트에 접속한다.
② *inetd*는 클라이언트의 연결 요청에 따라 *services*와 *inetd.conf* 파일을 참조하여 프록시 데몬을 호출한다.
③ 간단한 인증 절차를 거친 후 원래 접속하고자 하는 외부 호스트로 접속한다.
④ 사용자는 필요한 파일을 전송 받는다.
⑤ 파일 전송 시 *FTP-GW*는 내부 사용자의 호스트로 파일을 전송함과 동시에 방화벽 호스트에 동일의 파일을 임의의 이름으로 저장한다.
⑥ 파일 전송이 끝나기 직전 해당 파일의 바이러스 감염 여부 조사한다.
⑦ 바이러스 감염시 경고문을 사용자에게 전송한다.
⑧ 파일 전송을 마친다. 임시 저장 파일은 삭제한다.
⑨ 사용자 요구 시 외부 호스트 접속 종료한다.
⑩ 사용자 요구 시 방화벽 호스트 접속 종료한다.

4. 바이러스 탐지 방화벽 시스템 구현 및 평가

4.1 구현 환경

본 연구에서 구현된 시스템의 하드웨어 기본 환경은 Pentium Ⅲ-333Mhz CPU를 탑재한 IBM PC를 기반으로 구성되었으며, 소프트웨어 기본 환경은 RedHat Linux 6.2 운영체제이며 방화벽 구축을 위한 패키지로는 TIS의 Firewall Toolkit을 사용하였다. 또한 바이러스 탐지를 위한 라이브러리는 (주)하우리의 Virobot 바이러스 검색 엔진과 API 함수를 사용하였다.

구축된 네트워크망은 FDDI 백본망에 라우터로 연결된 서브넷 상에 바이러스 탐지용 방화벽 시스템 서버를 설치하고, Bastion Host 역할을 하는 서버는 외부망과 내부망 사이에

서 라우팅을 금지하여 모든 TCP/IP 트래픽을 막고 외부망과 내부망을 연결하는 2개의 네트워크 인터페이스를 통해 응용 계층에서 TCP/IP 트래픽을 중계하는 Dual Homed 게이트웨이 방식을 사용하였다.

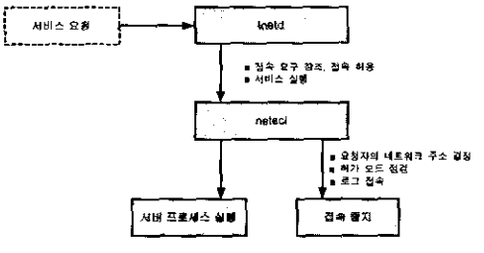
4.2 구현 내용

4.2.1 접근 제어 규칙

네트워크상의 바이러스 탐지 및 차단 시스템은 접근 허용 여부를 위해 TIS Firewall Toolkit 기반의 *netacl* 프로그램을 사용한다. *netacl*은 inetd 데몬에 의해 기동되며, 원격 사용자의 시스템으로부터의 서비스 요구를 허용하거나 거부하는 기능을 수행한다[16]. 본 연구에서는 이를 이용하여 바이러스 감염 파일을 유포한 사용자나 시스템의 접근을 효율적으로 제어 및 거부할 수 있게 한다. *inetd.conf* 파일에서 *netacl*을 설정하면 *netacl*은 오직 하나의 인수만을 취하는데 이 인수로는 시작하고자 하는 서비스의 이름이 입력된다. 또한 이외의 인수들은 *netacl*이 기동하는 서비스가 사용된다. *inetd.conf* 파일의 FTP 서비스와 관련된 사항은 다음과 같다.

```
ftp stream tcp nowait root /usr/local/etc/netacl /usr/sbin/in.ftpd
```

위의 경우는 FTP 서비스 접속 요청이 inetd 데몬에 의해 받아들여지게 되면 *netacl* 프로그램이 */usr/sbin/in.ftpd*를 인수로 하여 동작을 시작하게 한다.



(그림 5) netacl과 inetd의 상관 관계

ftpd 데몬이 시작되기 전 *netacl*은 해당 서비스 요구가 *netperm-table* 내 접속 규칙에 부합되는지를 검사하여 ftpd 데몬의 실행 여부를 판단하게 된다. 접속 요청 서비스이 수용과 거부는 syslog 데몬에 의해 다음과 같이 기록되며 이는 방화벽 시스템의 분석에 사용된다.

```
Jan 3 00 : 10 : 43 firewall netacl[339] : deny host = security.chosun.ac.kr/
203.237.110.75 service = in ftpd
Jan 3 00 : 13 : 28 firewall netacl[354] : deny host = security.chosun.ac.kr/
203.237.110.75 service = in.ftpd excute = /usr/sbin/in.ftpd
```

4.2.2 FTP 프록시 설정

Linux 시스템에서 네트워크 서비스는 inetd 데몬에 의해 기동된다. inetd은 시스템 부팅 시에 구동되는데 구동 시 */etc/inetd.conf* 파일에서 서비스의 목록을 얻는다. 그러나 일반적인 데몬처럼 항상 실행되어 있는 것이 아니라, inetd 데몬에서 일괄적으로 관찰하여 요청이 오는 서비스가 필요할 때 실행시켜 시스템 리소스를 절약할 수 있다. 따라서 다음과 같이 */etc/inetd.conf* 파일을 수정하여 FTP 프록시 동작을 구현할 수 있다.

FTP에 대한 접속요청이 있을 시에는 */usr/local/etc/ftp-gw*를 실행하고, Telnet에 대한 접속요청이 있을 경우에는 */usr/local/etc/tn-gw*를 실행하여 연결을 수립하게 된다.

```
# fwtk setting
ftp stream tcp nowait root /usr/local/etc/ftp-gw ftp-gw
telnet stream tcp nowait root /usr/local/etc/tn-gw tn-gw
```

이러한 환경 구성에서 FTP 포트로의 접속 시도가 발생되면 ftp-gw가 동작하게 되며, ftp-gw는 요청 호스트가 프록시 접속이 허용된 호스트인지 검사하게 된다. ftp-gw는 *netperm-table*에 설정되어 있는 접근 규칙에 따라 접속 허용 여부를 판별하게 되는데 ftp-gw를 위한 접근 규칙은 <표 1>과 같다.

〈표 1〉 ftp-gw의 접근 규칙

옵 션	설 명
userid 사용자	숫자로 표시된 UID나 /etc/passwd 내에 기록된 사용자 이름
directory pathname	서비스 프로그램을 호출하기 위해 ftp-gw가 chroot(2) 명령어를 실행하는 디렉토리
prompt 문자열	명령어 모드에서의 ftp-gw를 위한 프롬프트
denial-msg 파일	프록시 사용이 거부되었을 때 원격 사용자에게 표시할 메시지 파일명
timeout 초	프록시의 연결을 끊을 대기 시간
welcome-msg 파일	프록시 사용이 허용되었을 때 원격 사용자에게 표시할 메시지 파일명
help-msg 파일	'help' 명령어에 대하여 원격 사용자에게 표시할 도움말 파일명
denydest-msg 파일	사용자 인증이 거부되었을 때 원격 사용자에게 표시할 메시지 파일명

ftp-gw와 관련된 *netperm-table*에는 다음과 같이 접근 규칙이 설정되어 있다.

```
ftp-gw : denial-msg        /usr/local/etc/ftp-deny.txt
ftp-gw : welcome-msg       /usr/local/etc/ftp-welcome.txt
ftp-gw : help-msg /usr     /local/etc/ftp-help.txt
ftp-gw : denydest-msg      /usr/local/etc/ftp-baddest.txt
ftp-gw : timeout           3600
```

FTP 프록시에 대한 접근 허용 및 거부 규칙은 다음과 같이 추가 옵션에 의해 변경될 수 있다.

```
# telnet proxy
tn-gw : permit-hosts 203.237.*.*
tn-gw : deny-hosts unknown

# ftp-proxy
ftp-gw : permit-hosts 203.237.*.*
ftp-gw : deny-hosts unknown
```

이 규칙이 적용되면 도메인 이름을 DNS에서 발견할 수 없을 경우 접속이 거부되며, 203.237.*.* 네트워크로부터의 접근만을 허용하게 된다.

FTP 프록시를 통한 접근이 이루어지게 되고 허가된 호스트로 판단되면 접근 규칙에 따라 사용자 인증이 요구될 수 있다. 사용자 인증이 사용된 경우의 netperm-table의 내용은 다음과 같이 설정할 수 있다.

```
ftp-gw : permit-hosts 203.237.*.* -authall -log { retr stor }
```

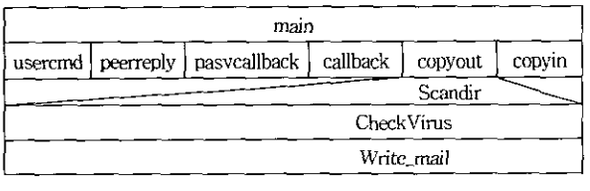
4.2.3 바이러스 탐지 및 차단 모듈 적용

FTP의 *get* 명령을 이용하여 리모트 서버에서 로컬 클라이언트로 파일을 전송하면 *copyout*( ) 함수가 실행됨으로써 방화벽의 바이러스 탐지 모듈이 동작을 시작하게 된다. *copyout*( ) 함수에 의해 파일을 다운로드하면서 *ScanDir*( ), *CheckVirus*( ) 함수로 바이러스 감염유무를 진단하게 되고 만약 바이러스에 감염된 파일이 발견되었을 때에는 사용자의 화면에 치료를 요구하는 메시지가 나타나고, 동시에 *write_ mail*( ) 함수에 의해 감염된 파일의 이름과 바이러스 이름이 관리자(*root@localhost*)로 메일이 발송된다. 네트워크 상의 바이러스 탐지 시스템에 사용된 주요 함수는 <표 2>와 같다.

〈표 2〉 바이러스 진단 관련 함수

함 수 명	함 수 인 자	리 턴 값
copyout	void	void
ScanDir	const char *MyFile	void
CheckVirus	unsigned char *pFilePath	int
write_mail	FILE *stream, char *VName	int

전송되는 모든 파일의 사본은 */tmp* 아래에 같은 이름으로 생성되고 바이러스 검사 엔진으로 바이러스 감염여부를 진단하여 바이러스 발견 시 조치하도록 하고 있다. 네트워크상의 바이러스 탐지 및 차단 기능을 수행하는 함수의 상관도는 (그림 6)과 같다.



(그림 6) 바이러스 진단 함수의 상관도

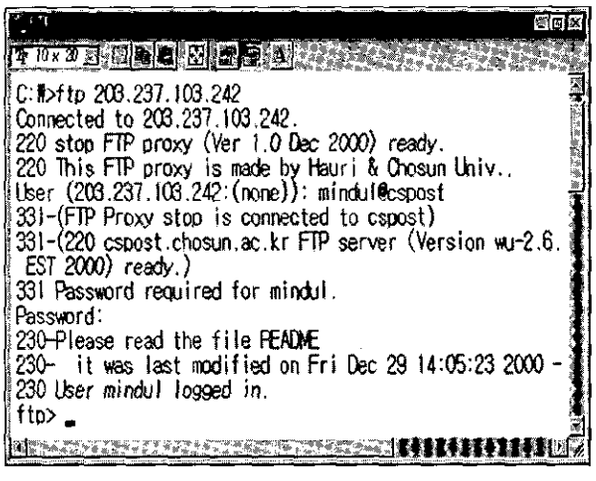
또한 본 시스템에서 바이러스를 탐지하는 모듈은 현재 컴퓨터 바이러스의 대부분이 16bit IBM-PC환경을 기반이므로 탐지 모듈내의 기본 타입을 16bit기준으로 설정하였다.

4.3 실험 및 고찰

본 논문의 바이러스 탐지 및 차단 방화벽 시스템 구현 결과를 살펴보고자 한다. FDDI 백본망에 라우터로 연결된 서브넷 상에 바이러스 탐지용 방화벽 시스템 서버를 설치하고, 외부망과 내부망을 연결하는 2개의 네트워크 인터페이스를 통해 응용 계층에서 TCP/IP 트래픽을 중계하도록 환경을 구성하였다.

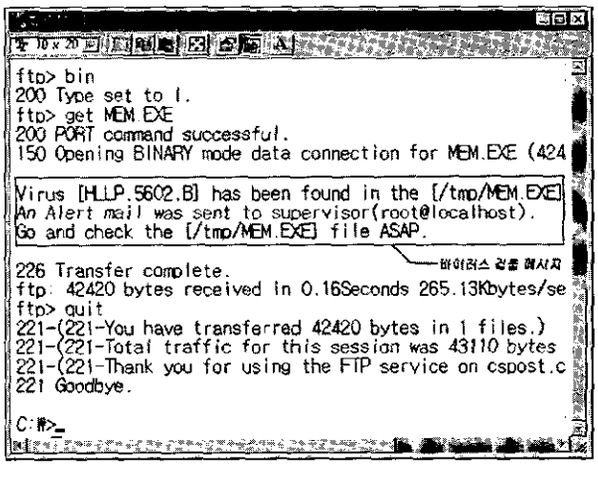
4.3.1 바이러스 진단 실험

보안 정책의 구현을 위한 보안 규칙을 입력하고 username@hostname 형식으로 로그인한다. 보안 규칙은 FTP, Telnet, rlogin 등의 서비스 접근 제어를 정의하고 있다. (그림 7)은 FTP 프록시를 통해 외부망의 사용자가 내부망으로 접속한 예이다.



(그림 7) 시스템 접근 인증

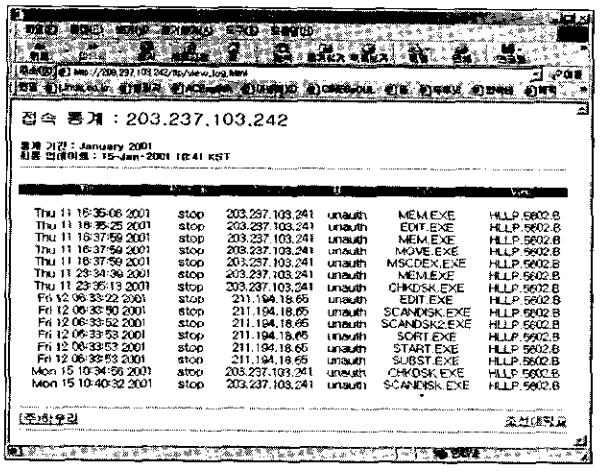
(그림 8)은 리모트 서버에서 로컬 클라이언트로 Binary Mode 상태의 파일 다운로드 과정을 보여주고 있다. 이 과정에서 42420 byte 크기의 mem.exe 파일이 바이러스 HLLP.5602.B에 감염되었음을 탐지하였다.



(그림 8) 파일 전송과 네트워크상의 바이러스 진단

전송된 파일(MEM.EXE)의 사본이 프록시 서버에 저장

된 후 프록시 서버의 임시 디렉토리 /tmp에서 바이러스 진단 및 치료하는 모듈이 동작을 하여 해당 파일의 바이러스를 치료할 수 있다.



(그림 9) 탐지된 바이러스 목록의 웹상의 리포트

4.3.2 성능 평가

구현된 시스템에 대한 성능 평가는 네트워크를 통해 전송되는 파일에 대한 바이러스 탐지율과 바이러스 탐지 모듈 동작 시 전체 방화벽 시스템의 네트워크 응답 시간에 대한 지연률을 측정하였다.

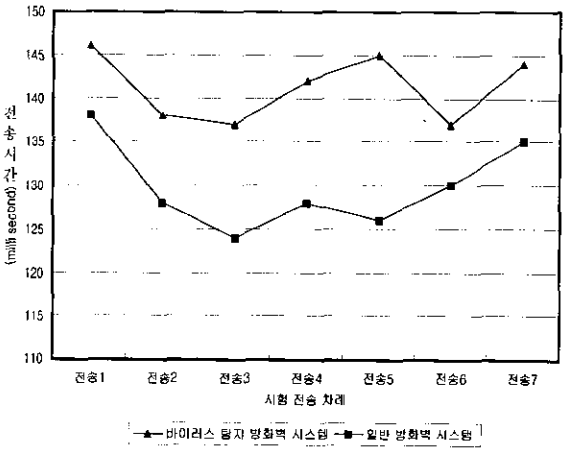
〈표 3〉 바이러스 검사 결과

No.	Time	Local	File	Virus
1	MON 15 13 : 34 : 56 2001	203.237.110.75	mem.exe	HLPP.5062.B
2	MON 15 13 : 35 : 02 2001	203.237.110.75	edit.exe	HLPP.5062.B
3	MON 15 13 : 35 : 03 2001	203.237.110.75	edit2.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
9	MON 15 13 : 35 : 15 2001	203.237.110.75	fc.exe	HLPP.5062.B
10	MON 15 13 : 35 : 16 2001	203.237.110.75	mem.exe	HLPP.5062.B
11	MON 15 13 : 35 : 16 2001	203.237.110.75	label.exe	HLPP.5062.B
12	MON 15 13 : 35 : 17 2001	203.237.110.75	move.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
25	MON 15 13 : 35 : 24 2001	203.237.110.75	mscdex.exe	HLPP.5062.B
26	MON 15 13 : 35 : 25 2001	203.237.110.75	fdisk.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
47	MON 15 13 : 35 : 30 2001	203.237.110.75	find.exe	HLPP.5062.B
48	MON 15 13 : 35 : 31 2001	203.237.110.75	chkdsk.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
55	MON 15 13 : 35 : 37 2001	203.237.110.75	attrib.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
65	MON 15 13 : 35 : 40 2001	203.237.110.75	move.exe	HLPP.5062.B
66	MON 15 13 : 35 : 40 2001	203.237.110.75	sort.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
77	MON 15 13 : 35 : 43 2001	203.237.110.75	subst.exe	HLPP.5062.B
76	MON 15 13 : 35 : 44 2001	203.237.110.75	start.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮
97	MON 15 13 : 35 : 53 2001	203.237.110.75	xcopy32.exe	HLPP.5062.B
98	MON 15 13 : 35 : 54 2001	203.237.110.75	debug.exe	HLPP.5062.B
⋮	⋮	⋮	⋮	⋮

위 <표 3>은 100개의 파일을 FTP 프로토콜을 이용하여 전송할 때 바이러스에 감염된 파일들을 탐지한 결과를 나타낸 것이다. 총 100개의 파일 중 22개의 파일이 바이러스에 감염되어 있는데, 바이러스 샘플은 모두 [HLLP.5062. B]라 명명된 파일 바이러스에 감염이 되어 있다. 이는 현재 한 종류의 바이러스 샘플만을 사용하였기 때문에 탐지율은 100%를 보이고 있다.

본 연구에서 설계 및 구현된 바이러스 탐지를 위한 방화벽 시스템은 (주)하우리의 바이로봇 엔진을 탑재하여 바이러스 진단에 대한 바이러스 정보가 독립된 바이너리 파일 형태로 존재하기 때문에 새로운 바이러스에 대한 대처는 전체 시스템의 구조 변경 없이 해당 파일만 업그레이드하도록 구성되어 있다.

다음으로는 임의의 110KB 크기를 갖는 파일을 FTP를 통해 7차례 전송했을 때 일반적인 방화벽 시스템과 바이러스 탐지 모듈이 작동할 때의 전송 시간 지연율을 측정하였다((그림 10) 참조).



(그림 10) 파일 전송 시간 측정표

위 전송 시간 측정표를 살펴보면, 일반 방화벽 시스템을 통과하는데 걸리는 시간은 평균 130ms가 걸렸으며, 바이러스 탐지 기능을 추가적으로 수행하는데는 평균 141ms가 걸려서, 평균 8%정도의 시간적 지연이 발생하였다. 이는 인증 절차 및 방화벽 자체 기능을 모두 배제한 바이러스 탐지와 관련된 전송 시간을 체크한 것이다. 따라서 실험에서 나타난 속도 저하는 바이러스 탐지 모듈을 거치는 과정에서 발생한 현상이므로 전체 모듈이 동작하는 경우에 있어서는 극히 미약한 저하율이라고 간주할 수 있다. 일반적으로 방화벽의 보안 기능이나 기타 모듈이 추가되면 성능이 다소 떨어지는 경향이 있지만, 본 논문에서 제안된 바이러스 탐지 기능을 추가한 방화벽 시스템은 전체 방화벽 시스템의 성능을 크게 떨어뜨리지 않음을 볼 수 있다.

5. 결론 및 향후 연구방향

본 논문에서 설계 및 구현된 바이러스 탐지 및 차단 방

화벽 시스템은 방화벽 고유의 기능을 유지하면서 지금까지 네트워크 상에서 유입되는 파일의 바이러스 진단 및 치료의 과정을 효과적으로 수행할 수 있음이 평가되었다. 이를 이용함으로써 날로 증대되고 있는 바이러스의 침입에 대해 시스템의 관리자 및 사용자는 수동적이고 정기적인 바이러스 검사 방법에서 탈피하여 내부 및 외부 네트워크 상에서 데이터가 유입되는 시기에 실시간으로 바이러스 검사를 행할 수 있음으로 보다 안전하고 신속한 바이러스 예방 효과를 얻을 수 있다. 하지만 네트워크 상에서의 바이러스 유입은 바이러스가 침입할 수 있는 여러 가지 경로 중 한 부분이므로 바이러스에 대한 절대적인 방지책이 될 수는 없다. 향후에는 네트워크 범위와 로컬 시스템이 보다 효율적인 연동 관계를 유지하여 바이러스를 탐지 할 수 있는 시스템 설계를 위한 연구가 필요하다.

참 고 문 헌

[1] RFC 959 "File Transfer Protocol(FTP)".

[2] W. Richard Stevens "Advanced Programming in the UNIX Environment," Addison Wesley 1992.

[3] William R. Cheswick and Steven M. Bellovin, "Firewall and Internet Security," pp.51-83, Addison-Wesley, 2nd edition 1994.

[4] D. B. Chapman and E. D. Zwicky, "Building Internet Firewalls," O'Reilly & Associates, Inc. 1995.

[5] Computer Viruses In Unix Networks Peter V. Radatti CyberSoft, Incorporated, February 1996 by Peter V. Radatti.

[6] FreeBSD Handbook, The FreeBSD Documentation Project, http : //www.freebsd.org/handbook/, July. 1998.

[7] SSH Secure Shell protocol, SSH Communications Security Ltd., http : //www.ssh.fi/, 1998.

[8] Keith Haviland, Dina Gray, Ben Salama "UNIX System Programming, 2/E," 1999.

[9] Kurt Wall "Linux Programming by Example," QUE 2000.

[10] "UNIX System V/386 Release 4 프로그래머 지침서", UNIX PRESS 캘라 1992.

[12] "방화벽 프로토타입 개발 보고서", 한국전산원, 1996. 12.

[13] "정보 보호 총서", pp.361-395, 한국정보보호센터, 1996년 12월.

[14] 권석철, 주영흠, 김관구, "컴퓨터 바이러스 완전소탕", 크라운 출판사, 1997.

[15] 조선대학교 "컴퓨터 바이러스 진단 치료 프로그램 및 감염예방 시스템 구현", 한국정보보호센터, 1997.

[16] 송의, 김남욱, 이병만, 송관호, "TIS-FWTK를 이용한 방화벽 구현", 정보과학회지, 제15권 제4호, pp.30-36, 1997.

최 준 호

e-mail : spica@hitel.net

1997년 호남대학교 컴퓨터공학과 졸업 (공학사)

2000년 조선대학교 전자계산학과(이학석사)

2001년~현재 조선대학교 전자계산학과 박사과정

관심분야 : 정보보안, 침입탐지시스템, 컴퓨터 바이러스, 영상처리

김 판 구

e-mail : pkkim@mina.chosun.ac.kr

1988년 조선대학교 컴퓨터공학과(공학사)

1990년 서울대학교 컴퓨터공학과(공학석사)

1994년 서울대학교 컴퓨터공학과(공학박사)

1995년~현재 조선대학교 컴퓨터공학부 교수

관심분야 : 시스템 보안, 운영체제, 정보검색, 영상처리