# 라우터기반 바이러스 감염파일에 대한 블록킹 시스템 설계

정종근* · 이윤배**

## The design of the blocking system against file infected with virus on router

Jong-Geun Jeong* · Yun-Bae Lee**

요 약

인터넷 기술은 사용자의 수만큼이나 빠르게 발전하고 있다. 그러나, 이기술이 사용자들에게 항상 좋은 면만을 보여주지는 않는다. 해커나 어린이조차도 컴퓨터 바이러스를 만들어 네트웍상에 배포한다. 그래서 바이러스의 위협은 점점 심각해져가고 있다. 본 논문에서는 이전에 개발되었던 바이러스 블록킹 시스템을 분석한후 라우터 상에서 바이러스에 감염된 파일을 차단하는 시스템을 개선하여 설계하였다. 다른 시스템과 비교하여, 라우터와 블록킹 시스템간의 전송시간을 단축하여 통신의 복잡성에서도 더 효율적이다.

ABSTRACT

The technology of the internet has made advanced progress the number of users increase rapidly. but, the technology doesn't only show users good and beautiful sides. the immoral hackers containing even children create computer viruses, and then spread them over network. so it makes the threat of viruses more serious. In this paper, we designed an improved blocking system against the infected file with viruses on router after analysing blocking systems against virus previously developed. comparising with other systems, the system designed is more efficient in terms of communication in complexity since it omits the transmission time between router and blocking system.

키워드

바이러스, 네트웍, 라우터, 방화벽

## Ⅰ. 서 론

일반적으로 인터넷에 대한 접속은 네트웍 서버를 우회하기 때문에 인터넷 태생의 바이러스들은 서버 중심의 바이러스 방역 소프트웨어로는 차단되지 않는다. 이 때문에 바이러스들은 아무런 저지를 받지 않고 내부 네트웍으로 침투할 수가 있다. 인터넷 게이트웨이와 그 네트웍 서버는 비 호환적인 프로토콜과 상호 다른 기계들을 사용하기 때문에 인터넷 게이트웨이로의

* 조선대학교 전자계산학과
** 조선대학교 컴퓨터공학과

접수일자: 2001. 12. 10

접속은 네트웍 서버를 우회한다[1, 2, 3] . 대개 네트웍 서버는 IPX프로토콜을 사용하는 NetWare하에서 작동하며, 전체 인터넷 게이트들 중 80%는 전송제어 프로토콜/인터넷 프로토콜 (TCP/IP)을 사용하는 UNIX 하에서 작동한다[10]. 바이러스 방역과 관련하여 종종 언급되곤 하는 메커니즘 중 하나는 바로 네트웍 접속 제어 파이어월이다. 파이어월은 승인을 받지 않은 외부인이 네트웍에 접속하는 것을 방지할 수 있으며, 문제를 야기시킬 여지가 있는 정보가 외부에서 내부로 들어오는 것을 막을 수 있다[5]. 본 논문은 이러한 상황에서 바이러스 방역시스템을 내부 네트웍의 문이라 할 수 있는 라우터 단위에서 차단하여 내부 네트웍을 보호하고 여러 바이러스 방역 시스템을 분석하여 문제점을 파악하고 이를 보안하기 위하여 설계하였다.

## Ⅱ. 기존의 바이러스 블럭킹 시스템의 문제점

### 2.1 네트웍 서버 상에서 바이러스 차단 시스템의 문제점

그림1에서와 같이 네트웍 서버상에서 바이러스 차단 시스템은 내부 네트웍에 여러 네트웍 서버가 존재한다면 이 모든 서버에 대한 바이러스 방역 시스템이 존재해야만 그 효과를 볼 수 있다. 내부 네트웍에 서버1, 서버2가 존재하고 서버1에 바이러스 방역 시스템이 존재하더라도 서버2는 바이러스에 대하여 무방비 상태이기 때문에 클라이언트들이 언제라도 서버2의 바이러스 감염 파일을 엑세스 할 수 있고 감염된 파일에 의해 내부 네트웍 클라이언트들이 언제라도 서버1에 바이러스 감염 파일을 복사시킬 수 있으므로 이 내부 네트웍은 바이러스로부터 안전 할 수 없다.

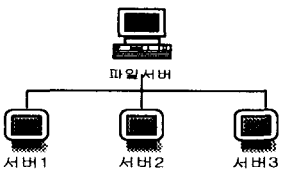


그림1. 네트웍상의 블럭킹 시스템
Fig1. Blocking system on network

### 2.2 방화벽 상에서 바이러스 차단 시스템의 문제점

보통 방화벽 상에서 바이러스 차단 시스템은 프락시 서버에 시스템을 설치하거나 아니면 독립형 차단 시스템을 구축한다. 이 시스템의 문제점은 패킷이 스크리닝 라우터를 한번 거치고 내부 프락시 서버에 의해 패킷이 한번 더 복사되어 진다. 만약 프락시 서버와 독립형인 바이러스 방역 시스템을 설치하게 되면 이 시스템에 의해서 패킷이 다시 한번 복사되어지는 부담을 가지게 되며 이는 네트웍 패킷 전송 속도의 저하를 가져온다. 또한 물리적 바이러스 방역 서버와 소프트웨어를 설치하기 위한 경제적 부담을 감수하여야 하며, 유지 보수로 인한 인적, 물적 자원의 낭비를 가져온다.

## Ⅲ. 네트웍 상에서 바이러스 감염 파일 블럭킹 시스템 설계

소규모로 내부 네트웍을 구성하는 곳에서는 라우터, 방화벽, 프락시, 바이러스 차단 서버를 두어 내부 네트웍을 운영한다는 것은 경비와 유지 보수에 있어 상당한 부담감을 준다. 따라서 게이트웨이에 방화벽과 바이러스 차단 시스템을 통합한다는 개념으로 시스템을 설계하였다. 모든 패킷은 라우터를 경유하며 라우터의 기본적인 기능인 패킷 경로 설정을 이용하여 내부로 들어온다. 따라서, 패킷 헤더를 분석하여 해당 내부 네트웍으로 들어오는 패킷 중에서 FTP, HTTP, SMTP 프로토콜을 사용하는 패킷을 필터링 하고 해당 패킷이라면 마지막 패킷을 제외한 모든 패킷을 목적지 주소로 보낸다. 마지막 패킷을 받은 후 프로토콜에 단편화되어 들어왔던 모든 패킷을 프로토콜별, 출발주소, 목적주소별로 구분하여 원래 파일 구조를 형성하여 바이러스를 검사한다. 네트웍 상에서 바이러스를 차단하는 시스템의 목적은 내부 네트웍으로 들어오는 바이러스 감염파일이 들어올 수 없도록 하는데 있다. 또한 네트웍 상에서 바이러스를 차단하기 위하여 패킷 처리하는 시간을 최소화하여야 하며, 네트웍 부하가 발생하지 않는 방향으로 설계되어야 한다.

본 논문에서 제안된 시스템은 스크리닝 라우터를 거쳐 외부로부터 내부 네트웍으로 향하는 모든 패킷에 대하여 패킷 캡쳐 모듈, 프로토콜 분석 및 파일 생성

모듈, 압축해제, 바이러스 체크 및 치료 모듈, 예외 상황 모듈을 통과하여 패킷을 프로토콜별로 분리하여 패킷 단편화되기 전 파일형태로 복구하여 바이러스를 검사, 치료하는 시스템으로 내부 네트웍을 바이러스로부터 안전하게 하며, 기존 시스템의 문제점을 해결하도록 설계하였다.

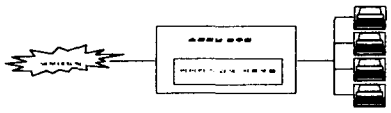

그림2. 제안된 시스템 구성도
Fig2. Suggested system structure

### 3.1 제안된 시스템 흐름

#### 3.1.1 패킷 캡쳐 모듈

네트워크를 통과해 지나가는 패킷을 모두 잡을 수 있어야 하고, 잡은 패킷들 중 미리 입력한 필터의 정보를 참조하여 어떤 것이 우리의 관심의 대상이 되는 것인지를 판단을 해야한다. 이러한 필터링을 통해서 필요한 패킷만이 프로토콜 분석 모듈로 올려진다. 패킷의 IP헤더와 TCP패킷으로부터 본 논문에서 제안한 시스템에 사용되어질 출발지의 주소, 목적지 주소, 포트, 순서 번호등의 정보를 알아낼 수 있다[7]. 패킷을 캡쳐하는 모듈은 패킷에 들어있는 프로토콜 별로 헤더를 분석하여 특정한 호스트 즉 내부 네트웍의 패킷일 경우에 받아들이고 그렇지 않은 패킷인 경우는 라우터 경로 설정 알고리즘에 의해 다른 곳으로 보내진다.

#### 3.1.2 프로토콜 분석 및 파일 생성 모듈

내부 네트웍으로 들어온 모든 패킷들을 관심 있는 프로토콜에 해당한 것인지를 판별한다. 관심 있는 프로토콜은 파일을 전송할 수 있는 프로토콜로써 FTP, SMTP, HTTP 세 가지 프로토콜로 설정하였다. FTP 프로토콜인 경우는 제어 패킷을 빼면 데이터 전송 패킷들로 이루어져 있으므로 제어 패킷은 바로 목적 어드레스로 전송한다. E-Mail을 전송하는 SMTP 프로토콜은 파일을 첨부할 경우 메일에 바이러스 감염 파일을 동봉할 수가 있고 그렇지 않은 경우는 대부분 프로토콜 헤어이던지 아니면 메일 Text 데이터 내용이다. 파일을 동봉하지 않은 경우는 목적 어드레스로 패킷을 전송한다. 인터넷 웹서비스에 관련된 HTTP는 웹 페이지에서 파일을 바로 다운로드 할 수 있는 기능을 가지고 있다. 텍스트 파일이나 브라우저에 브라우징을 할 수 있는 데이터 형식은 목적 어드레스로 패킷을 바로 전송한다.

위 세 프로토콜을 사용해서 파일이 전송되고 있다면 해당 프로토콜에서 출발, 목적 주소 별로 마지막 패킷인가를 검사하여 마지막 패킷을 제외한 모든 패킷을 목적 어드레스로 전송한다.

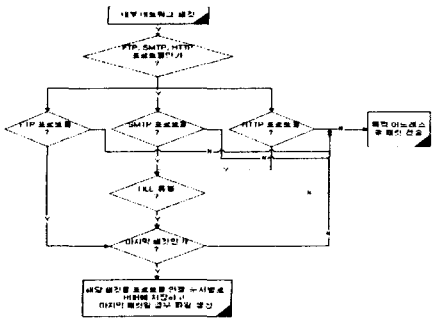

그림3. 프로토콜 분석 및 파일 생성 모듈
Fig3. protocol analysis and file generation module

#### 3.1.3 압축해제 및 바이러스 체크 및 치료 모듈

패킷으로부터 생성된 파일이 압축이 되었다면 이를 압축해제 한다. 압축이 되어있지 않은 파일은 바이러스 체크 루틴으로 점프한다. 압축 해제가 불가능할 경우는 예외상황 1로 제어권을 넘긴다. 압축해제 가능하다면 압축을 해제하고 압축 해제된 모든 파일에 대해서 바이러스 존재여부를 검사한다. 바이러스가 존재하고 이 바이러스가 치료 가능하다면 바이러스를 치료하고 마지막 패킷을 목적 어드레스로 전송한다.바이러스 치료가 불가능하다면 예외상황 2로 제어권을 넘긴다. 압축해제 파일에 바이러스가 존재하지 않는다면 바로 마지막 패킷을 목적지 어드레스로 전송한다. 마지막 패킷을 포함한 파일은 이제 본 시스템에 의미가 없어지므로 삭제한다.

#### 3.1.4 예외 상황 모듈

생성된 파일이 압축되어 있고 압축해제가 불가능한 경우 이를 무조건 삭제한다는 방식을 채택하였다. 만

약 압축해제 불가능한 파일을 목적 어드레스로 전송하고 그 파일이 목적 어드레스에서 압축이 정상적으로 해제되고 실행되었을 때, 그 파일에 바이러스가 존재하였을 때 본 시스템이 네트워 상에서 바이러스 치료해야 하는 역할 제대로 이행하지 못하기 때문에 압축해제 불가 파일은 무조건 삭제하는 것으로 설계하였다.

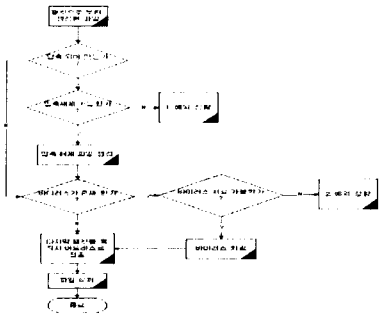

그림4. 압축해제, 바이러스 체크 및 치료 모듈
Fig4. uncompression, virus check, treatment module

목적 어드레스로 전송되지 않은 이유를 목적 어드레스 시스템이 공유하여야 하기 때문에 목적 어드레스 시스템으로 텍스트 형식의 파일을 전송하여 이 사실을 알리고 현재 프로세스를 종료시킨다. 바이러스 치료 불가 파일에 대해서는 예외 상황 2로 제어권을 넘긴다. 본 시스템은 현 네트워 관리자에게 바이러스 발견 메일 전송을 하고 패킷 출발 어드레스 서버 관라자에게 해당 파일이 바이러스가 검출되었음을 알리는 경고 메일을 전송한다. 목적지 어드레스 시스템에 바이러스 검출 메시지 파일을 전송한다. 검출된 바이러스를 치료하지 못했다면 이는 새로운 바이러스일 가능성이 많으므로 바이러스 연구단체에 바이러스 감염 파일을 첨부하여 메일 전송을 한다. 파일 삭제하고 현 프로세스를 종료시킨다.

## Ⅳ. 성능평가

### 4.1 실험 환경

본 실험은 펜티엄 700에 소켓을 4개 생성한 상태에서 리눅스 환경에서 실시하였다. 패킷이 해당 시스템에 들어오면 시스템이 패킷을 복사하고 헤더를 분석하여 이를 네트웍 디바이스에 쓰는데 걸리는 처리 속도와 패킷이 전송될 때 거치는 시스템간의 패킷 이동속도 이 두 가지를 고려하였다. (그림8)은 본 논문에서 제시한 시스템을 평가하기 위하여 패킷 구조를 단순화시켜 기존 바이러스 차단 시스템과 본 논문에서 제시한 시스템에서 처리할 가정된 패킷 구조이다.

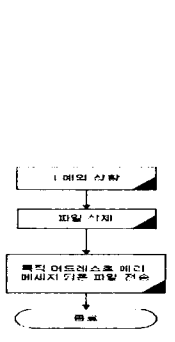


그림5. 예외 상황 모듈 1
Fig5. Exception module1

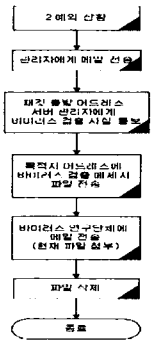


그림6. 예외상황 모듈2
Fig6.Exception module2

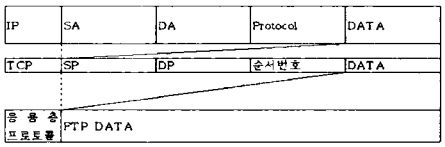


그림7. 시스템 비교를 위한 TCP/IP 간소화 패킷 구조
Fig7. TCP/IP simple packet structure for system compare

### 4.2 성능 비교 및 고찰

4.2.1 기존 바이러스 블럭킹 시스템

그림8은 바이러스 감염 파일이 패킷으로 단편화되어 내부 네트웍의 관문이라 할 수 있는 라우터에 들어와 허가된 패킷인지 아닌지를 검사하는 과정을 표시한 것이다. 보여지는 바와 같이 처음 패킷이 라우터에 17 : 40 : 18 : 895에 도착하였으며 마지막 패킷이 17 : 40 : 19 : 26에 들어 왔고 허가된 패킷임을 확인하고 라우팅하고 있음을 보여주고 있다.

바이러스에 감염된 파일의 단편화된 처음 패킷이 17

: 40 : 18 : 905에 도착하였으며 마지막 패킷이 17 : 40 : 19 : 45에 들어 왔고 마지막 패킷을 포함한 모든 패킷을 조합해서 데이터를 검사한 후 바이러스임을 확인한 시간이 17 : 40 : 19 : 45임을 보이고 있다.

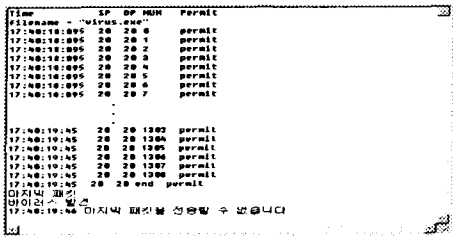

그림8. 기존 바이러스 차단 시스템에서 패킷처리
Fig8. packet process in Existence virus blocking system

4.2.2 본 논문에서 제안된 시스템

그림9는 제안된 시스템에서 내부 네트웍으로 허가된 패킷임을 확인되어진 패킷 중에서 상위 프로토콜을 검사하여 패킷 순서 번호대로 버퍼에 저장한 뒤에 마지막 패킷이 도착하자 바이러스 검사를 하여 바이러스에 감염된 파일임을 확인하고 마지막 패킷을 목적지 주소로 보내지 않음을 보이고 있다. 처음과 마지막 패킷의 들어온 시간은 17 : 40 : 18 : 895와 17 : 40 : 19 : 26이며 바이러스 검사를 마친 시간은 17 : 40 : 19 : 27임을 보이고 있다.

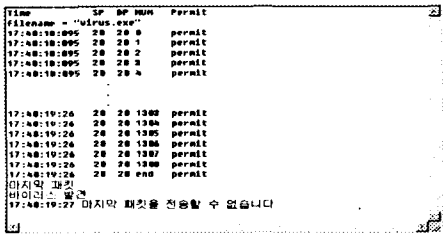

그림9. 제안된 시스템에서의 패킷처리
Fig9. packet process in suggested system

### 4.3 실험 결과

실험결과에 의하면 110KB 파일을 기존 바이러스 차단 시스템에서 라우터와 바이러스 차단 시스템을 거치는 과정 동안 바이러스를 검색하기 위해 151 milli second의 시간을 소요하였으며 제안된 시스템은 바이러스를 검색하기 위해서 132 milli second의 시간을 소요하였다.

이러한 결과를 얻게된 이유는 첫째, 기존 바이러스 차단 시스템은 네트웍 전송 시간을 낭비하기 때문이다. 라우터에서 바이러스 차단 시스템으로 패킷을 전송하는 동안 10 milli second의 시간이 필요하다. 라우터에서 바이러스 차단 시스템으로 패킷을 전송하는 동안 10 milli second의 시간이 필요하다. 둘째, 라우터에서 패킷을 복사하고 네트웍 디바이스에 쓰는 시간, 바이러스 차단 시스템에서 패킷을 다시 한번 복사하고 라우팅 헤더를 제외하고 그 상위 프로토콜을 분석하여 실제 데이터를 추출하고 이를 다시 네트웍 디바이스에 쓰는 시간이 소요되기 때문이다.

## V. 결론 및 향후 연구 방향

많은 바이러스 방역 백신과 바이러스 방역 시스템이 제안 설계되어 왔다. 특히, cisco사에서 개발한 라우터의 대부분은 방화벽이나 VPN을 지원하는 통합된 형태를 보여주고 있지만, 이것만으로는 완전하게 바이러스를 탐지해 내지 못한다는 단점이 있다. 본 논문에서는 이러한 시스템들을 분석하여 문제점을 파악하고 보다 효율적인 네트웍 바이러스 방역 시스템을 설계하였다.

본 논문에서 제안하여 설계된 시스템은 바이러스 검색, 치료 모듈이 안정적이라면 외부 바이러스가 내부 네트웍으로 진입하는 것을 완전하게 차단할 수 있도록 설계하였다. 외부 네트웍과 내부 네트웍의 분리점인 스크리닝 라우터에만 시스템을 설치함으로써 네트웍 서버 상에서 바이러스 차단 시스템의 내부 네트웍에 대한 바이러스 위협을 제거하고 방화벽 상에서 바이러스 차단 시스템의 과중한 네트웍 부하를 줄이고 경제적, 유지보수의 부담을 최소화하도록 설계하였다. 마지막 패킷만이 바이러스 체크 이후 전송됨으로 클라이언트로부터 본 시스템의 투명성을 확보하도록 설계하였다. 하지만, 스크리닝 라우터의 필터링 기능에 패킷 의미 데이터를 추출하여 저장하는 기능과 바이러스를 검사하는 루틴이 추가되어 속도 저하의 문제점이 제기되고 있다. 따라서, 이러한 문제점 때문에 본 시스템은 소규모 네트웍에 적합한 바이러스 차단 라우터이다.

## 참 고 문 헌


[1] Avolio, Fred. & Sebes, J. Application Gateways and Filtering Gateway: A comparison of Firewall Designs. Data Security Letter #59 Trusted Information Systems, 1995.

[2] Bellovin, Steven M. Security Problems in the TCP/IP Protocol Suite. Computer Communications Review, 19(2):32-48, April 1989.

[3] Chapman,Brent D. Network Security Through IP Packet Filtering. In USENIX Security Symposium,1999

[4] Bellovin, Steven M. Firewall-Friendly FTP. RFC 1579, February 1994.

[5] Cheswick, William R. and Bellovin, Steven M. Firewalls and Internet Security. Addison-Wesley, Reading, MA, 1994

[6] Ranum, Marcus J. Internet Firewall Frequently Asked Questions 1995.

[7] Ranum, Narcus J. Thinking About Firewalls. Proceeding of first International Conference on Systems and Network Security and Management, November 1992.

[8] 성안당, "TCP/IP 완전 정복", 2000.

[9] 교학사, "TCP/IP 및 윈도우 네트워킹 프로토콜", 1999.

[10] 한빛미디어, "TCP/IP 네트워크 관리", 1999.

[11] 한국정보보호센터, "실시간 LAN기반 패킷 모니터링 개발",『정보보호센터 연구개발결과보고서』, 1998.


## 저 자 소 개


정종근(Jong-Geun Jeong)

1995년 조선대학교 전자계산학과 졸업(이학사)

1997년 조선대학교 대학원 전 자계산학과졸업(이학석사)

1999년 3월 - 현재 조선대학교 대학원 전자계산학과 대학원 박사과정

1999년 3월 - 2001년 현재 서린정보시스템 팀장

1999년 3월 - 2001년 현재 동강대학 전자정보과 겸임교수

※관심분야 : 인공지능, 전문가 시스템, 멀티미디어, 정보보안, 네트워크, 전자상거래, 바이러스

이윤배(Yun-Bae Lee)

1980년 광운대학교 전자계산학과 졸업(이학사)

1983년 광운대학교 대학원 전자계산학과 졸업(이학석사)

1993년 숭실대학교 대학원 전자계산학과 졸업(공학박사),

1988년 4월 - 현재 조선대학교 컴퓨터공학부 교수,

1999년 7월 - 2000년 현재 광주광역시 시정정책자문회의 위원

1996년 7월 - 2000년 현재 광주광역시 및 전라남도 지역 정보화 추진위원

2000년 - 2000년 현재 광주교육신문사 회장

1997년 9월 -2000년 2월 조선대 정보과학대학장,

1996년 12월 - 1997년 2월 호주 타스마니아대학 초청교수

※관심분야 : 인공지능, 전문가시스템, 멀티미디어, 데이터베이스, 정보보안, 바이러스