네트워크 트래픽 필터링 시스템의 설계 및 구현

김민수○ 김성조
중 앙 대 학 교 컴 퓨 터 공 학 과
{mskim, sjkim}@konan.cse.cau.ac.kr

Design and Implementation of Network Traffic Filtering System

Min-Soo Kim○ Sung-Jo Kim
Dept. of Computer Science & Engineering, Chung-Ang University

요 약

인터넷의 속도가 점차 빨라지고 대역폭이 증가함에도 불구하고 네트워크 대역폭은 여전히 부족한 자원이다. 이는 네트워크 속도가 빨라짐에도 불구하고 그만큼 이를 모두 활용하는 어플리케이션이 계속 개발되어 사용되기 때문이다. 기업 차원에서는 주요한 자산인 네트워크 대역폭을 효율적으로 활용하는 것이 매우 중요하며, 개인 차원에서는 미성년들이 유해한 정보에 노출되는 것을 방지하는 것이 필요하다. 본 논문에서는 이러한 필요성에 따라 네트워크 트래픽을 분석하고, 이를 바탕으로 관리 정책에 따라 네트워크 트래픽을 필터링하는 시스템을 설계하고 이를 구현하였다.

1. 서론

인터넷 속도가 점차 고속화되어 가고 있으나 새로 등장하는 소프트웨어 역시 점차 많은 대역폭을 사용함으로써 인터넷 속도가 증가되어도 여전히 더 큰 대역폭을 요구하는 상황이 계속되고 있다. 또한 개인 사용자들이 집이나 PC방 등을 통해서 고속의 네트워크를 쓸 수 있게 됨에 따라 네트워크 대역폭을 사용하는 사용자 비율이 꾸준히 증가하고있다. 개인 사용자와 달리 기업체에서는 이러한 인터넷 속도의 증가가 기업에게 반드시 이익을 가져왔다고 볼 수 없는데 이는 기업체의 개개인이 인터넷을 통해서 얻는 정보가 업무와 관련된 정보인 지를 알 수 없으며 이를 관리할 수 있는 방법도 없기 때문이다. 업무와 관련없는 불필요한 인터넷 서핑을 업무 시간에 함으로써 기업은 인력의 생산성 저하와 네트워크 대역폭 낭비라는 불이익을 얻는다. 마찬가지로 인터넷을 쉽게 사용할 수 있는 여건이 갖추어지면서 인터넷을 주로 사용하는 청소년이 인터넷을 통하여 유해한 정보에 무방비 상태로 노출되는 것을 볼 때 이를 차단할 수 있는 방안이 반드시 필요하다.

파이어월(Firewall)은 내부 네트워크와 외부 네트워크 사이에 존재하면서 보안을 해칠 수 있는 요소를 차단하는 목적을 가지고 운용된다. 이러한 파이어월은 보안을 쟁점으로 했을 때 그 기능을 충분히 발휘할 수 있지만 네트워크 대역폭 낭비, 유해 정보의 유입등을 완전히 차단할 수 있는 해결책은 제시하지 못한다. 이러한 것들은 보안과 관련된 요소들이 아니기 때문에 파이어월로 차단하는 것에 한계가 있다. 따라서 보다 차별화된 시스템을 통해서 유해 정보의 유입이나 불필요한 트래픽을 차단하는 것이 필요하다. 본 논문에서는 이를 위해 클라이언트의 네트워크 트래픽을 분석하고 관리하는 네트워크 트래픽 필터링 시스템을 설계하고 구현한다.

본 논문의 구성은 다음과 같다. 2절에서는 기존의 패킷 필터링 시스템과 관련된 연구를 소개하고, 3절에서는 본 연구에서 설계하고 구현한 인터넷 트래픽 필터링 시스템의 네트워크 구성과 필터링 알고리즘, 트래픽 분석 툴을 설명한다. 마지막으로 4절에서는 결론 및 향후 연구 과제에 대하여 기술한다.

2. 관련 연구

2.1 패킷 필터링 시스템

패킷 필터링은 어떤 데이터가 네트워크로 들어오고 나가는 것을 통제하는 보안 메커니즘으로 파이어월에서 쓰이는 기법이다. 네트워크를 통과하는 모든 트래픽은 패킷의 형태로 전송된다. 패킷 필터링은 이러한 패킷의 헤더를 읽고 전체 패킷을 통과시킬 지 차단할 지를 결정하는 것이다. 패킷을 통제하는 기준은 기본적으로 패킷의 출발지 주소, 패킷의 목적지 주소, 데이터를 전송하는데 사용한 포트와 어플리케이션 프로토콜을 바탕으로 한다. 예를 들어, 출발지가 내부 클라이언트 IP이고 목적지가 외부 네트워크 IP, 사용된 프로토콜이 Telnet 프로토

콜이면 이 패킷을 차단 또는 허용하는 식이다. 패킷 필터링은 프록시와는 달리 클라이언트가 패킷 필터링이 이루어 진다는 사실을 모르는 투명성(Transparency)을 제공한다. 대부분의 상용 라우터는 패킷 필터링을 제공하고 있으며 리눅스에서는 커널 내부에 패킷 필터링이 구현되어 있다. 리눅스 버전2.4에서는 'iptables'와 같은 툴을 이용해서 필터링 규칙을 추가, 삭제할 수 있도록 지원한다.

이러한 필터링 시스템의 문제점은 패킷 필터링 규칙을 설정하기가 어렵고 속도를 위해서 전송(transfer) 계층에서 패킷을 필터링하기 때문에 보다 세밀한 패킷 필터링이 어렵다는 점이다. 보다 세밀한 패킷 필터링을 위해서는 어플리케이션 계층에서의 필터링이 필요하다.

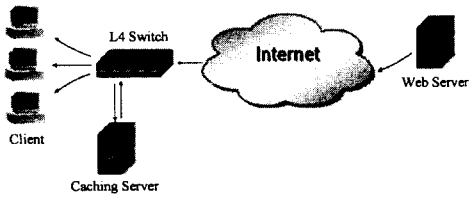
2.2 어플리케이션 계층에서의 필터링 시스템

HTTP 프로토콜에 대하여 트랜스포트 계층에서의 필터링은 출발지 주소, 목적지 주소, 포트 번호를 기준으로 필터링이 이루어진다. 특정 사이트를 차단할 때는 그 사이트의 IP주소를 이용하여 차단하게 된다. 하지만 하나의 IP에 대하여 여러 사이트가 존재하거나 사이트 중에서도 일부분을 차단하고자 할 때는 IP만으로 차단하는 것이 불가능하다. 어플리케이션 계층에서의 필터링은 IP 대신 URL을 이용하여 차단한다. 따라서 하나의 IP를 여러 사이트가 쓰는 경우나 사이트의 일부분만을 차단하는 것이 가능하다. 또한 어플리케이션 계층의 패킷에는 실제로 의미를 파악할 수 있는 문장이 그대로 전달되는 경우도 있으므로 이 내용을 읽어서 필터링하는 것도 가능하다.

3. 시스템 설계

3.1 네트워크 구성

본 논문에서 설계된 네트워크 트래픽 필터링 시스템은 HTTP, FTP, NNTP프로토콜에 대하여 어플리케이션 계층에서 필터링을 한다. 이들 프로토콜로 전달된 클라이언트의 요청은 캐시서버를 거친 다음 실제 필터링 시스템에는 URL과 클라이언트 IP를 포함한 정보들이 전달된다.



[그림 1] 필터링 시스템의 네트워크 구성

네트워크 트래픽 필터링 시스템의 전반적인 네트워크 구성은 [그림 1]과 같다. 클라이언트의 요청은 Layer 4 스위치를 통해서 캐시 서버에 전달된다. 캐시 서버는 클라이언트의 요청이 캐시에 존재하면 이를 곧 클라이언트에게 전달하고 캐시에 내용이 없으면 인터넷을 통해서 클라이언트가 요청한 정보를 가져와서 캐시에 저장한 다음 클라이언트에게 넘겨준다. 이 과정에서 캐시 서버는 클라이언트의 요청을 받은 시점과 동시에 필터링 시스템에 그 요청을 전달하여 필터링이 이루어지도록 한다. 필터링 시스템은 캐시 서버에서 플러그인 형태로 동작하면서 캐시 서버가 클라이언트의 요청을 전달하였을 때 이를 처리한다.

여기서 클라이언트의 요청이 캐시 서버로 전달되도록 하는 역할을 하는 것은 Layer 4 스위치이다. Layer 4 스위치는 Layer 3 스위치가 출발지와 목적지의 IP주소만으로 패킷을 전달하는 것과는 달리 특정 포트번호로 패킷을 전달하는 기능을 제공한다. 따라서 HTTP와 같이 80번 포트를 사용하는 패킷이 클라이언트로부터 전달되면 이를 외부 네트워크로 보내기 전에 캐시 서버로 보내서 캐시를 먼저 조사한다. 캐시 서버는 동시에 필터링 시스템에게 클라이언트의 요청을 보냄으로써 필터링 시스템이 이를 차단할 것인지 허용할 것인지 판단할 수 있도록 한다. 클라이언트의 요청은 어떠한 웹 브라우저를 사용하는가에 관계없이 필터링 시스템을 거치게 된다. 또한 캐시 서버가 클라이언트에게 투명성을 갖는 것과 마찬가지로 클라이언트는 필터링 시스템의 존재 여부를 알 수 없다. 본 논문에서 제시한 네트워크 구성외에도 네트워크 트래픽을 스누핑하여 필터링을 수행하는 경우, 프락시 서버에서 필터링하는 경우, 파이어월에서 필터링하는 경우 등 다양한 네트워크 구성을 통해서 시스템을 구축할 수 있다. 어떠한 네트워크 구성이든 클라이언트에게 투명성과 빠른 속도를 제공하는 것이 필수적이다.

3.2 필터링 알고리즘

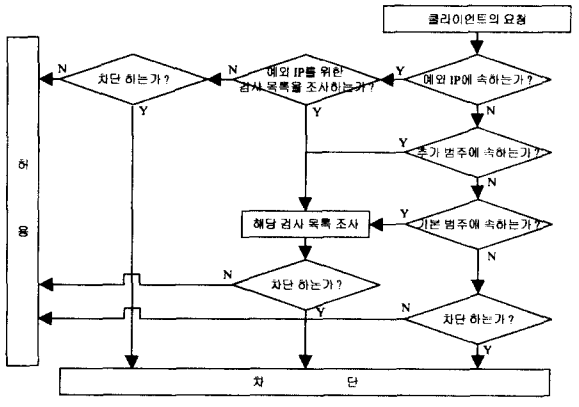
클라이언트의 요청은 그 내용에 따라 여러 가지 범주로 분류할 수 있다. 본 시스템에서는 이를 음란물, 해킹 관련 정보, 채팅, 게임, 인터넷방송, 스포츠, 구인정보, 뉴스의 8가지로 분류하며 이를 기본 범주라고 한다. 각 범주에 속한 사이트, FTP 서버, 뉴스그룹들에 대하여 차단하거나 허용할 것들을 모아놓은 목록이 필요하다. 이를 검사 목록이라고 하며 검사 목록에 있는 것은 모두 하나 이상의 범주에 포함된다. 필터링 시스템의 기본 처리방식은 클라이언트의 요청이 검사 목록에 존재하는 목적지 IP와 경로명에 속하면 이를 필터링 정책에 따라 차단 또는 허용하는 것이다. 필터링 정책은 시간대 별로 나뉘어져서 목록에 해당하는 것을 차단할 지 허용할 지를 결정한다. 검사 목록에 대하여 일방적으로 차단하는 대신 허용하도록 하는 것은 블랙 리스트와 함께 화이트 리스트를 구성할 수 있도록 해준다. 즉, 특정 정보만 클라이언트에게 허용하고 나머지는 모두 차단하는 형태로 네트워크를 운영할 수 있다.

범주에는 기본적으로 구성된 8가지 범주와 필요에 의해서 추가할 수 있는 추가 범주가 있다. 추가 범주는 클라이언트의 요청을 보다 세부적으로 나눌 필요가 있을 때 이용할 수 있는 범주이다. 추가 범주에 속한 검사 목록에 포함되는 클라이언트의 요청도 역시 검사 목록에 대한 처리 방식에 따라 차단 또는 허용된다.

이러한 범주와 검사 목록을 이용하여 클라이언트의 요청을 필터링하는 것과 함께 특정 클라이언트에 대한

별도의 필터링이 필요하다. 이는 한 시점에 네트워크를 사용하는 모든 클라이언트를 동일한 필터링 정책에 의해 관리하는 대신 클라이언트마다 서로 다른 필터링 정책으로 관리함으로써 보다 유연하게 네트워크를 관리하도록 한다. 별도의 필터링 정책에 적용되는 클라이언트는 예외 IP라는 그룹으로 묶는다. 예외 IP에 속한 클라이언트에 대해서는 모든 요청을 허용 또는 차단하거나 예외 IP만을 위한 검사 목록을 적용하여 허용 또는 차단을 한다. 이러한 과정은 [그림 2]와 같다.

클라이언트의 요청은 <클라이언트 IP, 프로토콜, 포트번호, 목적지 IP, 경로명, 경로명길이>의 형태로 분류되어 캐시 서버에서 필터링 시스템으로 전달된다. 필터링 시스템은 첫 번째 단계에서는 클라이언트가 예외적으로 처리되는 예외 IP에 포함되는 가를 검사한다. 예외 IP에 포함되면 목적지 IP와 경로명에 관계없이 무조건 허용 또는 차단하거나 예외 IP만을 위한 검사 목록을 이용해서 차단 또는 허용한다. 이러한 예외 IP를 위한 검사 목록이 적용되는 것은 시간을 기준으로 한다. 일정 시간대에는 예외 IP처리를 위한 검사 목록을 통해서 요청을 처리하고 나머지 시간대에는 공통으로 적용되는 검사 목록을 통해서 요청이 처리된다. 두 번째 단계에서는 추가

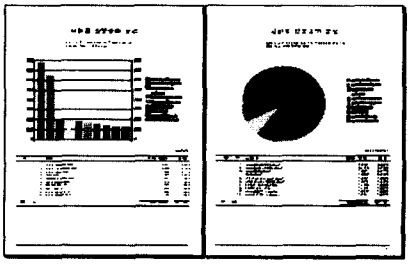


[그림 2] 필터링 알고리즘

범주에 대하여 조사한다. 클라이언트의 요청이 추가 범주에 속하는 검사 목록에 있으면 이를 필터링 정책에 따라 허용 또는 차단한다. 세 번째 단계에서는 기본 범주에 대하여 조사한다. 클라이언트의 요청이 기본 범주에 속하는 검사 목록에 있으면 이를 필터링 정책에 따라 허용 또는 차단한다. 마지막 단계에서는 기본 범주와 추가 범주에 속하지 않은 클라이언트의 요청을 필터링 정책에 따라 허용 또는 차단한다. 이렇게 필터링의 단계를 순차적으로 구성함으로써 각각의 필터마다 우선 순위를 가지도록 한다. 즉 예외 IP가 가장 높은 우선 순위를 가지고 검사되며 그 다음에 추가 범주, 기본 범주가 우선 순위를 갖는다. 마지막으로 예외 IP에도 속하지 않고 어떠한 범주에도 속하지 않는 요청이 처리된다.

3.3 네트워크 트래픽 분석 툴

필터링 시스템은 클라이언트의 모든 요청을 로그에 저장한다. 트래픽 분석 툴은 로그에서 중복되는 정보를 통합하여 로그의 크기를 1/10로 축소한다. 로그를 바탕으로 분석 툴은 클라이언트의 요청에 대하여 차단된 요청과 허용된 요청의 비율, 프로토콜별 비율, 범주별 비율, 시간대 별 요청의 변화, 빈도가 높은 요청을 분석한다. 또한 각 클라이언트별 요청 회수와 빈도가 높은 요청, 범주별 요청을 분석한다. 분석한 내용은 [그림 3]에서와 같이 표와 그래프를 통해서 출력한다. 이러한 분석은 네트워크를 실제적이고 효율적으로 관리할 수 있도록 하는 기반이 된다. 관리자는 이를 바탕으로 하여 네트워크 사용 특성에 맞는 관리 정책을 세울 수 있다.



[그림 3] 로그 분석 툴의 보고서

4. 결론 및 향후 연구 과제

본 논문에서는 클라이언트의 네트워크 사용 현황을 분석하고 이를 토대로 하여 네트워크를 보다 효율적으로 사용할 수 있는 시스템을 설계하고 구현하였다. 이 시스템을 통하여 클라이언트의 네트워크 사용 현황을 쉽게 파악할 수 있었으며 이에 맞는 필터링 정책을 통해 네트워크를 관리할 수 있었다. 필터링 시스템의 핵심은 필터링 속도와 투명성이라고 할 수 있다. 향후 연구 과제는 독립 형태로 수행되는 필터링 시스템을 통해서 보다 빠른 속도로 다양한 프로토콜에 대하여 필터링할 수 있는 시스템을 구현하는 것이다.

참 고 문 헌

[1] Subhash Suri, George Varghese, "Packet Filtering in High Speed Networks", ACM-SIAM symposium, 1999

[2] Steven McCanne, Van Jacobson, "The BSD Packet Filtering: A New Architecture for User-level Packet Capture", December 1992

[3] V. Srinivasan, G. Varghese, S. Suri and M. Waldvogel, "Fast Scalable Algorithms for Level Four Switching", SIGCOMM '98, 1998

[4] D. Brent Chapman, Elizabeth D. Zwicky, "Building Internet Firewalls", O'Reilly, 1995

[5] Chad Stewart, "Linux IP Firewalling Chains", http://netfilter.kernelnotes.org/ipchains/