제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

브리지상에서의 패킷 필터링의 구현

김용*, 방용희*, 구하성*
*한서대학교 대학원 전산학과
e-mail : yong@clubjd.co.kr

Implementation of Packet Filtering using Modified Bridge Model

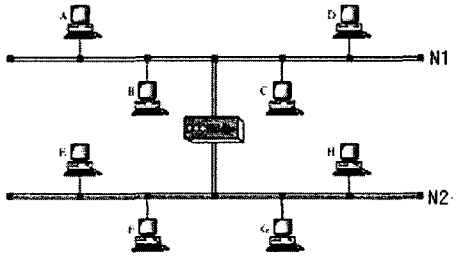
Yong Kim*, Yong-Hee Bang*, Ha-Sung Koo
*Dept. of Computer Science, Han-Seo University

요 약

기존의 브리지들은 여러 네트워크를 하나의 네트워크로 묶어주는 역할 만을 하였지만, 패킷을 검사하고 필터링하는 기능은 포함하지 않았다. 본 논문에서는 브리지상에서 패킷을 검사하고 필터링하는 기능을 포함하는 브리지에 대하여 구현하였다.

1. 서론

두 네트워크에 패킷 필터링과 모니터링을 적용한 투명 브리지를 리눅스상에서 구현하였다.



(그림 1) 브리지 구성

(그림 1)은 두개의 서로 다른 네트워크가 브리지를 통해 연결된 모습을 보여주고 있다. 두 네트워크 간에 패킷은 브리지를 통해 교환된다. N1 상에 포함되어 있는 A 가 N2 상에 속해 있는 G 에게 패킷을 보내기 위해서는 반드시 브리지를 지나야 한다[1]. 이러한 두 네트워크 종단에서 발생하는 패킷을 브리지에서 모니터링 하거나 필터링 할 수 있다면 일정한 보안 규칙을 적용해 패킷에 대한 새로운 권한을 설정할 수 있다.

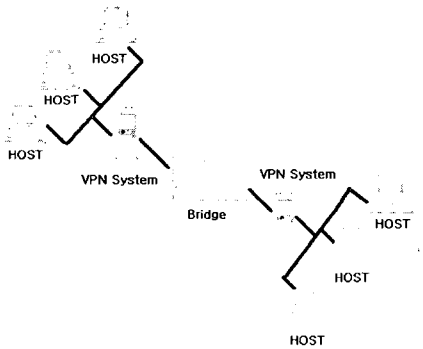
본 논문은 브리지에서 패킷을 필터링하고 모니터링 할 수 있는 리눅스 기반에 네트워크 툴을 구현하는 방안에 대해 제시하고 현재 구현된 테스트 모듈을 통해 패킷이 필터링 되는 모습과 모니터링 되는 모습을 증명할 것이다.

2 장은 패킷 모니터링과 필터링을 위한 브리지 구현과 캡쳐 할 프로토콜 소개, 3 장에서는 물리적인 패킷 캡쳐 방법, 4 장에서는 물리적인 패킷 전송 방법, 5 장에서는 브리지에서 패킷 필터링 구현, 6 장에서는 결론 및 향후 과제에 대해 기술한다.

2. 브리지 구현과 캡쳐 프로토콜

2.1 브리지 구현

브리지를 구현하기 위해 Red Hat Linux 를 일반 PC 에 설치하고 네트워크 카드 2 장을 장착해서 각각에 네트워크에 연결해 간단한 브리지를 구현하였다.

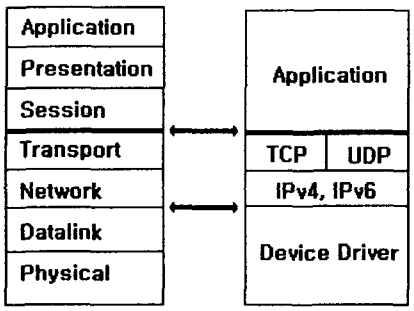


(그림 2) 시스템 구성도

(그림 2)는 전체적인 시스템 구성도를 간단하게 표현한 것이다.

2.2 캡쳐 프로토콜

2.2.1 IP Protocol

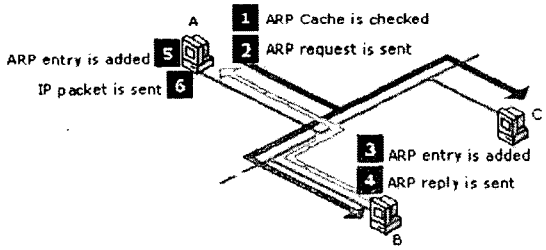


(그림 3) OSI 7 layer 와 Protocol 구성도

위의(그림 3)은 Ethernet 기반에서 TCP/IP 를 이용하여 통신하는 Application 의 모습이다. 이러한 모든 TCP/IP 를 이용하는 Application 은 Network 계층에 IP Protocol 을 이용한다[2].

2.2.2 ARP Protocol

ARP 는 IP 네트워크 상에서 IP 주소를 물리적 네트워크 주소로 대응시키기 위해 사용되는 프로토콜이다. 여기서 물리적 네트워크 주소라 함은 Ethernet 또는 토큰링의 48 bits 네트워크 카드 주소를 의미한다.



(그림 4) ARP Protocol 절차

(그림 4)는 IP 패킷이 전송되기 위한 ARP 프로토콜에 역할을 보여주고 있다. IP 패킷이 전송되기 위해서는 반드시 ARP Protocol 을 통해 패킷이 전송될 목적지에 대한 확인이 필요하다.

3. 물리적인 패킷 캡쳐

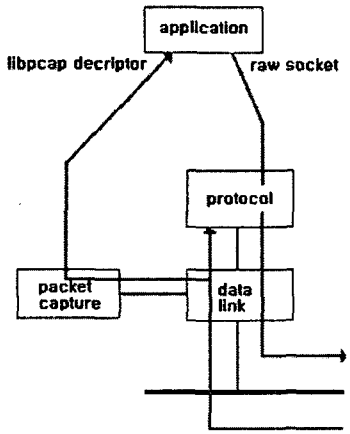
3.1 패킷 캡쳐 라이브러리(libpcap library)

패킷 라이브러리는 Data link 계층에 수신된 물리적인 패킷을 직접 application 에서 수신 할 수 있게 여러 지원 API 를 제공하여 주며, 지원하는 API 에 기능을 정리하면 다음과 같다[3].

1) 패킷을 수신하기 위한 디바이스를 설정 할 수 있다.
2) 받을 패킷에 길이 지정을 통해 패킷에 전체가 아닌 일부만을 수신할 있다.
3) 패킷을 수신하기 위한 디바이스에 네트워크 주소와 서브넷 마스크를 알수 있다.
4) 캡쳐 하고자 하는 프로토콜을 지정 할 수 있다.
5) Data link 계층에서 직접 패킷을 수신한다.
6) 설정된 디바이스를 해제한다.

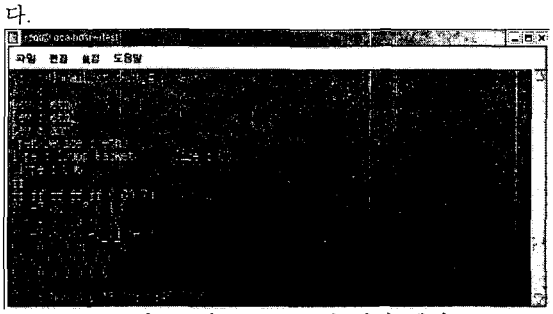
3.2 패킷 캡쳐 테스트 모듈

패킷 캡쳐 테스트 모듈은 패킷 캡쳐 라이브러리를 사용하여 Data link 계층을 지나는 모든 패킷을 캡쳐하고 헤더를 분석하여 패킷의 Source Address, Destination Address, 프로토콜의 종류, 길이 등을 보여 주는 모듈로 제작되었다.



(그림 5) 테스트 캡쳐 모듈에 구조

(그림 5)는 패킷 캡쳐 라이브러리를 사용해 제작된 테스트 모듈에 구조이다. Data link 계층에 수신된 패킷이 application 단에 직접 전달되는 모습을 확인 할 수 있다.



(그림 6) 테스트 모듈에 패킷 캡쳐

(그림 6)는 테스트 모듈에서 캡쳐된 Ethernet 패킷이 Application 을 통해 출력되는 모습이다.

4. 물리적인 패킷 전송

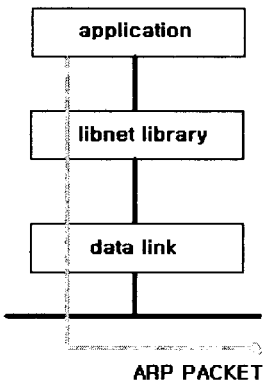
4.1 패킷 전송 라이브러리(libnet library)

패킷 전송 라이브러리는 물리적인 Wire 에 직접 전송할 패킷을 출력 하는 API 를 제공하는 라이브러리 이며, 지원하는 API 에 기능을 정리 하면 아래와 같다[4].

1) 출력되어야 할 디바이스를 연결한다(예:eth0, eth1...)
2) 지정된 패킷을 Wire 에 출력한다.
3) 지정된 디바이스 해제한다.

4.2 패킷 전송 테스트 모듈

패킷 전송 모듈은 임의에 ARP 프로토콜에 패킷을 제작한 후 직접 Wire 에 전송하고 전송된 ARP 패킷에 응답 패킷을 수신 하는 모듈로 제작되었다.



(그림 7) 테스트 전송 모듈에 구조

(그림 7)은 테스트 전송모듈이 패킷 전송 라이브러리를 사용해서 패킷을 직접 Data link 계층에 전송하는 모습을 확인할 수 있다.

(그림 8) 테스트 모듈에 패킷 전송과 응답

(그림 8)은 ARP 패킷을 직접 만들어서 테스트 모듈을 통해 전송하는 모습을 나타내고 있다.

5. 브리지에서 패킷 필터링 구현

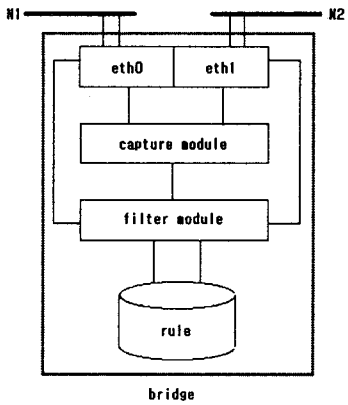
패킷 캡쳐와 전송 모듈을 기반으로 하여 브리지에서 패킷을 필터링 한다. 각각의 네트워크 디바이스에서 읽어드린 패킷을 아래에 명시한 보안 규칙에 의해 필터링 한다.

1) 발신자 IP Address, Port
2) 수신자 IP Address, Port
3) 패킷의 Payload 에서 2 진 데이터

패킷에 관한 필터링은 위에 3 가지 조건들을 조합해 적용하고 보안 규칙이 적용된 패킷은 다음 3 가지로 나누어 진다[5].

1) 패킷의 폐기(Destroy)
2) 패킷의 전송(Forward)
3) 패킷의 수정 후 전송(IP Spoofing)

위와 같은 보안 규칙이 적용된 패킷은 모두 파일이나 데이터베이스와 같은 물리적인 저장 공간에 저장이 된다.



(그림 9)브리지 패킷 필터 구조

위의(그림 9)는 브리지 패킷 필터 구조로써, 두개에 네트워크 N1 과 N2 에서 수신된 전체 패킷 가운데 TCP, UDP, ARP 만을 캡쳐 모듈이 캡쳐해서 필터 모듈로 전송하고 필터 모듈은 수신된 패킷을 보안 규칙에 따라 패킷의 폐기,전송,수정 후 전송을 결정하여 다시 네트워크 N1, N2 에 전송하게 된다. 또한, 캡쳐 모듈은 캡쳐된 모든 TCP,UDP,ARP 패킷을 Application 에 출력하는 역할도 수행하고 필터모듈에서는 보안규칙이 적용된 모든 패킷을 파일이나 데이터베이스와 같은 물리적인 저장 공간에 저장하여 데이터의 열람이 향후 언제나 가능하도록 구현하였다.

6. 결론 및 향후 과제

본 논문에서는 브리지상에서 패킷 필터링과 모니터링이 가능한 보안 툴을 제작 하므로써, 두 네트워크의 모든 패킷을 검사해서 패킷단위에 보안이 가능한 시스템을 구현하였다. 각 모듈이 아직은 테스트 버전으로 제작된 상태이고 기본적인 기능만을 구현했으므로 좀 더 많은 시간을 두고 보안하여야 할 것이며, Ethernet 기반에서만이 아닌 좀더 다양한 네트워크에 지원을 연구할 것이다. 또한, 패킷에 대한 보안 규칙을 좀 더 상세히 두고 패킷에 대한 트래픽양을 체크할 수 있는 기능을 추가 할 것이며 Application 이 아닌 리눅스 커널에 직접 현재 모듈들을 적용해 커널단에 보다 안전하고 강력한 기능을 제공하는 보안 기법을 연구 제작 할 것이다.

참고문헌

[1] Andrew S. Tanenbaum. "Computer Networks", 3rd Ed. Prentice Hall, 1996
[2] Karanjit S. Siyan and Joern Wettern. "Inside TCP/IP", New Rider Publishing, 1997
[3] pcap library : http://www.tcpdump.org
[4] libnet library: http://www.packetfactory.org
[5] IP Spoofing : http://www.takedown.com
[6] W.Ricahrd Stevens. "Unix Network Programming", 2rd Ed. Prentice Hall, 1998