범용 패킷 포획 도구

°*이 준원, **심 영철
*홍익대학교 대학원 정보공학과
**홍익대학교 컴퓨터 공학과

General Purpose Packet Capturing Tool

°*Joon-Won Lee, **Young-Chul Shim
*Dept. of Infomation Engineering, Hong-Ik Univ.
**Dept. of Computer Engineering, Hong-Ik Univ.

요 약

컴퓨터 통신의 발달로 인하여 정부기관, 학교, 연구소, 기업체등 사회의 모든 분야에 걸쳐 인터넷 환경이 널리 보급되고 있다. 이를 통하여 컴퓨터 간의 단순한 정보와 자원의 공유에 국한되었던 범위를 넘어 전자 결제 전자 상거래, 상용 서비스 등 많은 편리함이 제공되고 있다. 그러나 최근 들어 이러한 인터넷을 이용한 불법 침입자들의 정보 유출이나 파괴 혹은 금융 사고와 같은 범죄가 더욱 많아지고 지능화 되고 있다 이러한 문제점을 보완하기 위해 불법적인 침입자들이 네트워크를 통해 시스템에 접근하여 중요한 정보를 유출 혹은 파괴하는 행위를 감시할 수 있는 시스템이 필요하게 되었다. 본 논문에서는 이러한 목적을 위해 사용하는 네트워크 모니터링 혹은 침입 탐지를 위한 도구를 제안하고 기술할 것이다 이 도구는 현존하는 도구들이 특정 패킷만을 모니터링할 수 있는 문제점을 강력한 명세언어를 사용하여서 응용프로그램에서 사용하는 패킷까지도 포획할 수 있는 기능을 제공한다.

1. 개 요

컴퓨터를 이용하는 사람들이 많아지면서 여러 형태의 다양한 서비스들이 나타나고 있다. 특히 컴퓨터 통신망을 이용한 서비스들이 폭발적으로 늘어나고 있다 예를 들면 은행에 직접 나가지 않고 집에서 컴퓨터와 통신망을 이용하여 계좌 이체같은 은행 업무를 볼 수 있는 홈뱅킹(Home Banking), 매장에 직접 나가지 않고 물건을 구매하는 홈쇼핑(Home Shopping), 업무처리와 관련된 결재 시스템, 그리고 초기적인 전자 상거래(Electronic Commerce)등이 있다. 과거에는 단순히 정보를 공유하고 자원을 공유하는데 통신망을 이용했으나 컴퓨터 네트워크와 관련된 기술들이 널리 확산, 보급됨에 따라서 컴퓨터 통신망의 활용 범위가 확대되어 가고 있는 것이다.

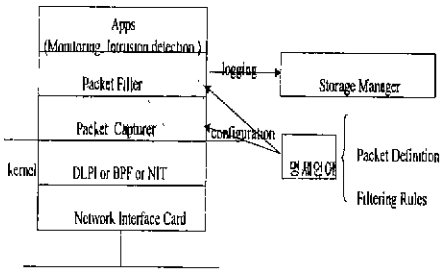
이러한 새로운 형태의 서비스들이 나타나면서 여러 긍정적인 효과와 함께 부정적인 효과가 많이 나타나고 있다. 컴퓨터 네트워크를 이용하여 불법적으로 정보를 유출시켜서 개인, 회사, 또는 국가에 막대한 손해를 입히거나 다른 사람의 계좌를 임의로 조작시키는 것과 같은 금융사고등이 그러한 예이다 이러한 문제들을 해결하기 위해 컴퓨터 네트워크 보안에 관한 많은 연구가 진행 되어 왔으며 두 가지 방식으로 접근할 수 있다. 첫 번째 방법으로 컴퓨터 네트워크를 통해 중요한 시스템의 접근을 막는 방법이다. 이러한 도구의 예로 tcp_wrappers등이 있다. 두 번째 방법으로 컴퓨터 네트워크를 통한 침입을 발견하는 방법을 들이 있다 이러한 도구의 예로 tcpdump와 특별한 패킷을 관찰할 수 있는 ARPwatch, TCPwatch, Netman, clog, Netwatch, Argus등을 들수 있다 또한 위에서 언급한 두가지 방법을 모두 지원하는 많은 도구가 존재한다. 이러한 도구의 예로 SessionWall을 들 수 있다.

위에서 언급한 도구는 물리 계층(Physical Layer) 바로 위에 있는 데이터 링크 계층(Data Link Layer) 인터페이스를 이용하여 데이터 링크 계층에서 패킷(Packet)들을 포획하여 헤더(Header)와 내용을 살펴봄으로써 컴퓨터 네트워크의 사용시의 문제점을 해결하는 데 많은 도움을 준다. 그러나 이런 도구들은 SMTP(Simple Mail Transfer Protocol), FTP(File Transfer Protocol), HTTP(Hyper Text Transfer Protocol) 등의 이미 존재하는 한정된 프로토콜에 대해서만 동작한다 즉 새로운 형태의 패킷에 만들어지면 패킷을 포획할 수 없는 문제점을 가지고 있다. 프로토콜의 종류가 다양하게 변하기 때문에 이러한 도구만으론 앞서 언급했던 문제들을 해결할 수 없다. 따라서 좀더 유연한 구조를 가진 패킷 포획 도구가 필요하게 되었다. 이러한 유연한 구조를 가진 범용 패킷 포획 도구는 과거에 존재하던 여러 도구의 특징을 포함하고 있으나 과거의 도구가 가지고 있던 한정된 패킷만을 포획하여 분석할 수 있던 문제를 사용자가 패킷을 정의하게 하는 방법으로 문제를 해결 할 수 있는 새로운 형태의 도구이다.

본 논문의 구성은 범용 패킷 포획 도구에 관하여 설명 할 것이며 구성은 다음과 같다. 2장에서 범용 패킷 포획 도구에 관한 설명을 할 것이며, 3장에서는 명세 언어에 관하여 설명을 할 것이다. 마지막으로 4장 결론으로 끝을 맺을 것이다

2. 범용 패킷 포획 도구의 구조

범용 패킷 포획 도구의 전체적인 구조는 아래 <그림 2-1>과 같다.

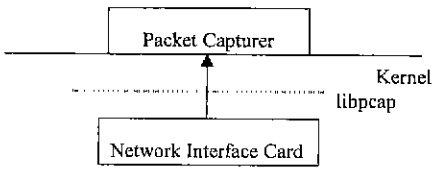


<그림 2-1> 범용 패킷 포획 도구의 구조

위 <그림 2-1>과 같이 범용 패킷 포획 도구는 여러 모듈로 구성되나 논문에서는 그중 가장 중요한 4개 모듈을 기술한다.

2.1. Packet Capturer

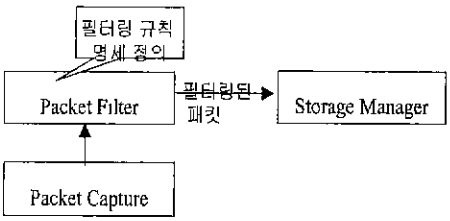
운영체제에서는 시스템을 지나가는 패킷을 포획하기 위한 각각의 운영체제마다 promiscuosu 모드하의 상이한 데이터 링크 인터페이스를 가지고 있다 BSD계열 Unix 에서는 BPF(Berkely Packet Filter), System V계열의 Unix에서는 DLPI(Data Link Provider Interface). Windows계열에서는 NDIS (Network Device Interface Specification)를 제공한다. 본 논문에서 제시한 도구는 여러 운영 체제의 데이터 링크 인터페이스 위에서 범용적인 패킷 포획 기능을 제공하는 libpcap 라이브러리를 이용한다 Packet Capturer는 현재의 시스템을 지나가는 모든 패킷들을 포획하여 Packet Filter에게 보낸다. 그 구조는 아래의 <그림 2-2> 과 같다.



<그림 2-2> Packet Capture의 구조

2.2. Packet Filter

Packet Filter는 Packet Capturer에서 보내온 패킷들을 받아서

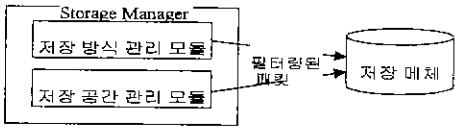


<그림 2-3> Packet Filter의 구조

명세언어에서 정의한 여러 명세에 따라 패킷을 분류하여 Storage Manager에 넘긴다. 그 구조는 <그림 2-3>와 같다.

2.3. Storage Manager

Storage Manager는 <그림 2-4>과 같은 구조로 이루어진다
위 <그림 2-4>과 같이 Storage Manager는 2개의 모듈로 구성



<그림 2-4> Storage Manager의 구조

되어 진다. 각각의 역할은 아래와 같다

○ 저장 방식 관리 모듈 : Packet Filter에서 필터링하여 보내온 패킷을 받아서 사용자가 선택한 방법에 따라 로그를 남긴다.

○ 저장 공간 관리 모듈 : 모든 저장 메체는 한정된 저장 공간을 가지고 있으므로 저장 공간 관리 모듈은 주기적으로 사용자가 로그를 남기기로 지정한 저장 장치의 저장 공간을 검사하여 필요하지 않은 로그를 삭제한다

2.4. 명세 언어

명세 언어는 다른 언어와 같은 형태를 가지고 있으나 범용 패킷에 관련된 여러 가지 일을 명세 언어로 표현할 수 있는 풍부한 표현력을 가지고 있다. 이러한 명세 언어의 역할은 Packet Capturer가 포획할 패킷을 정의하고 Packet Filter가 필터링할 패킷의 규칙을 정의한다

명세 언어에 관한 자세한 설명은 3장에서 한다.

3. 명세 언어

범용 패킷 포획 도구는 현존하는 프로토콜뿐 아니라 응용 프로그램에서 사용하는 여러 종류의 패킷을 포획하여 패킷필터링과 모니터링을 동시에 수행할 수 있어야 한다 이러한 요구 사항을 충분히 만족하도록 명세 언어를 설계하였다

범용 패킷 포획 도구의 명세 언어는 많은 사람이 여러 방면에서 많이 사용하고 있는 언어인 Perl을 기반으로 한다 Perl을 기반으로 하므로 범용 패킷 포획 도구의 명세 언어는 일반적인 언어가 가지고 있는 특징인 흐름 제어, 함수, 변수등을 지원하며 특히 네트워크상에서 일반적으로 사용하는 IP Address와 Host Name을 Perl의 변수를 이용하여 언어 구문의 확장 없이 지원할 수 있다. 그러나 범용 패킷 포획 도구의 필터링과 모니터링에 관하여 명세 언어상에서 표현하기 위해 Perl 구문을 확장하였다

확장된 Perl 구문은 Packet Filter가 Packet Capturer에서 보낸 패킷의 내용을 접근할 수 있도록 하는 부분과 필터링 하는 부분의 두 가지로 나뉜다.

Packet Filter가 Packet Capturer에서 보낸 패킷의 내용을 접근할 수 있도록 thing::thing이라는 형태의 구문을 사용한다. 예를 들어 만약 IP 프로토콜의 Source 필드와 Destination 필드를 접근하기를 원하면 IP::src, IP::dst라는 구문을 사용하여 접근할 수 있으며 TCP 프로토콜의 header 전체를 접근하기를 원하면 TCP::hdr이라는 구문을 사용하여 접근할 수 있다.

범용 패킷 포획 도구의 명세 언어는 Packet Filter가 Packet Capturer에서 보낸 패킷의 내용을 가지고 패킷을 필터링 기능을 수행하기 위해 permit, deny, analyze라는 예약어를 추가하였다 이 예약어의 기능은 아래와 같다.

allow source, destination, services

예약어 allow는 source, destination, services와 일치하는 패킷을

통과를 허락한다.

deny source destination, services

예약어 deny는 source, destination, services와 일치하는 패킷을 통과를 막는다.

analyze source, destination, services

예약어 analyze는 source, destination, services와 일치하는 패킷만을 분석한다.

이러한 예약어를 사용한 구문은 아래 같은 형태이다 아래의 구문에서 action은 위에서 언급한 예약어 중 하나를 사용하여 Packet Filter의 행동을 정의한다. track은 action이 수행된 후 어떠한 행동을 할 것인가를 정한다. 예를 들면 로그를 남기거나 관리자에게 email을 보내는 등의 행동을 들수 있다.

```
if( action )
   track
```

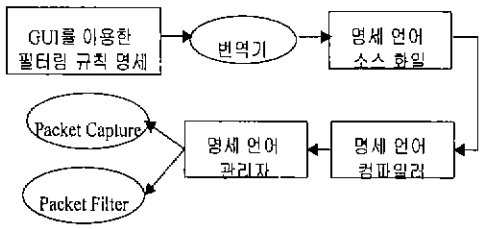
위 언급한 구문과 예약어를 이용하여 간단한 HTTP detector를 작성하면 아래와 같다

```
open(LOGFILE,">/var/test.log") || die "Cannot open file";

 if( analyze all, $myWebServer, TCP::sport  == 80 && HTTP::GET)
        write;

close(LOGFILE);

format STDOUT =

@###.###.### ###  @### ###.### ### @## @##

IP::src, IP::dsc, TCP::sport, TCP::dport
```

몇몇 경우에는 범용 패킷 포획 도구에서 제공하는 패킷 필드로는 응용 프로그램에서 이용하는 패킷 포획 능력의 부족할 수 있다. 이러한 경우에는 사용자가 명세 언어의 module 개념을 이용하여 새로운 패킷의 필드를 접근할 수 있도록 정의할 수 있다 이와 같은 방법으로 응용 프로그램에서 사용하는 프로토콜을 분석할 수 있다. 예를 들어 smtp 프로토콜의 sender와 recipient와 같은 필드를 접근할 수 있도록 명세언어를 통해 module로 작성 하였다면 spam mail에 관한 분석을 할 수 있다. 그에 관한 코드는 아래와 같다

```
 if(analyze all, all, TCP::dport == 25)
 {

  if( $STMP::sender == $known_spammers)
   if ( $STMP::recipients > 10000 ) {
   ... possible spam
 }
```

범용 패킷 포획 도구의 명세 언어의 명세는 기본적으로 GUI를 이용하도록 고려 하였으며 또한 일반적인 text입력이 가능하다. 아래 <그림 3>는 명세 언어의 개발 과정을 나타낸다. 범용 패킷 포획 도구 사용자는 GUI 방식으로 사용자가 포획할 패킷과 필터링 규칙을 입력한다. 그 후 사용자가 입력한 GUI를 기반으로한 명세를 번역기를 통과하면 명세 언어 소스 파일이 생성한다. 이 파일이 명세 언어 컴파일러를 통과하여 Packet Filter와 Packet Capturer에 사용할 수 있는 방식으로 번역된다 이렇게 번역된 코드를 명세 언어 관리자를 통하여 Packet Capture와 Packet Filter로 적재한다.



<그림 3> 명세 언어의 개발 과정

4. 결론

본 논문에서 제안된 범용 패킷 포획 도구는 불법적인 침입자들이 네트워크를 통해 시스템에 접근하여 중요한 정보를 유출 혹은 파괴하는 행위를 감시할 수 있는 도구를 제안 하였다. 또한 이 도구를 활용하여 간단한 패킷 필터 방화벽, 침입 감지 도구, 네트워크 분석등에 이용할 수 있다.

발전 방향으로는 TCP/IP만을 고려하였으나 앞으로 개선 방향으로는 TCP/IP 이외의 다른 프로토콜으로의 확장, 사용자 인터페이스을 개선, 1Gbit Ethernet와 같은 고속망에서의 패킷 포획과 포획된 패킷에 저장에 관하여 연구 할 것이다

참고문헌

[1]S. Parameswar, "Universal Packet analyser - A Network Packet Filterig Tool," Project Report, Department of Computer Science, Texas A&M University, College Station, TX, 77843, Fall 1995.

[2]Steve McCanne and Van Jacobson, "The BSD Packet Filter," In Winter USENIX, pages 259-269, USENIX Association, January 1993.

[3]Ouri Wolfson, Soumitra Sengupta, and Yechaiam Yemini, "Managing Communication Networks by Monitoring Databases", IEEE Transactions On Software Enginnering, vol. 17, no 9 Septemper 1991

[4]Jeffrey C. Mogul, Richard F. Rashid, and Micheal J. Accetta. "The Packet Filter' An Efficient Mechanism of User-Level Network Code," In Proceedings of the 11th Symposium on Operating Systems Principles, pages 39-51, ACM, November 1987.

[5]Christopher Wee, "LAFS' A Logging and Auditing File System", Department of Computer Science. University of California Davis.

[6]Ehab Al-Shaer. High-Performance Event Filtering. Survey and Evalution. Technical Report ODU-CS-96, Computer Science Department, Old Dominion University, March 1996

[7]Ehab Al-Shaer, Hussien Abdel-Wahab, and Kurt Maly Design and Implementation of High-Performance Event Filtering Agent for Monitoring Distributed Multimedia Application.Submitted to High Performace Networking(HPN'97). White Plains, New York, April 1997