네트워크 패킷 감사를 이용한 침입탐지시스템의 구현

김영진*

*(주)데이콤

Implementation of Intrusion Detection System Using Network Packet Capture

Young-Jin Kim*

*Dacom Corporation

E-mail : kyj2000@chollian.net

요 약

컴퓨터 확산 및 네트워크 이용의 급격한 증가에 따른 부작용으로 컴퓨터 보안 문제가 중요하게 대두되고 있다. 공격자들의 공격은 운영체제, 프로토콜, 응용프로그램에서 취약점을 이용하고 있으며 그 기술이 고도화, 전문화 되어가고 있다. 그러므로 정보통신망의 기반구조를 구성하는 구성요소들에 대한 구조, 관리에서의 문제점을 해결하기 위한 기반구조 보호기술이 필요하다. 본 논문에서는 효과적으로 침입자를 차단하여 중요 시스템에서 분리시키기 위한 침입탐지시스템을 개발하고, IDS 모델을 설계 및 구현한다.

ABSTRACT

Computer security is considered important due to the side effect generated from the expansion of computer network and rapid increase of use of computers. A attack of intruders using a vulnerability of operating system, protocol and application programs. And so, The attack methods is to be high technology and professional. Thus It must be necessity that we necessary a solution to structure, management for framework of information technology. This paper develope intrusion detecting system for separating intruders form critical system and design IDS model and implementation of it.

키워드

보안, 침입탐지시스템, 불법침입, 패킷

Ⅰ. 서 론

20세기말 컴퓨터와 정보·통신 기술의 비약적인 발전에 힘입어, 컴퓨터는 이제 행정, 경제, 사회, 문화 등 각 분야에서 없어서는 안 될 필수적인 수단으로 자리 잡게 되었다. 또한 컴퓨터의 성능이 급속도로 발전하여 컴퓨터 네트워크가 확산되고 개방화 되어감에 따라 수많은 정보가 디지털(Digital)화 되고, 전자화된 정보 자체가 부가가치를 지닌 재화로 인식되어 의사표시와 법률행위 등이 전자적으로 이루어지면서 부가가치의 창출속도 또한 날로 급진되고있다. 반면 이의 역기능으로 컴퓨터 시스템에 대한 불법적인 침입 및 정보의 유출이 크나큰 문제로 부상되었다. 이에 대한 대비책으로 침입탐지시스템이 제안되었다. 침입탐지시스템은 감시하고자 하는 대상에 따라 크게 네트워크 기반 IDS(NIDS)와 호스트 기반 IDS(HIDS)로 나뉘고, 컴퓨터 시스템의 비정상적인 사용을 탐지하거나 알려진 해킹 패턴이나 OS 취약성 등에 대비하고 이를 보안하는 역할을 수행한다.

본 논문의 구성은 2장에서 침입탐지시스템에 대해서 분류하고 그 문제점 및 특징을 살펴보고 Ⅲ장에서는 Libpcap를 이용한 침입탐지시스템을 설계하고 개선된 보안 모델의 개념과 구성요소의 기능에 대하여 설명한다. Ⅳ, Ⅴ장에서는 제안된 침입탐지시스템을 구현 및 분석하고, Ⅴ마지막으로 본 논문의 결론 및 향후 연구 방향에 대하여 기술한다.

II. 본 론

침입탐지 시스템은 외부의 침입 시도로부터 내부의 특정 호스트나 네트워크를 보호하기 위한 보안기술로 접근정책을 통한 내외부간 트래픽을 제어하는 기술이다. 그 기본 목표는 보호 대상이 되는 네트워크에 불법 사용자들의 접근해 컴퓨터 자원들을 사용하는 것을 방지하고, 자신의 정보들이 불법적으로 외부에 유출되는 것을 방지하는 것이다. 침입탐지시스템은 1987년 Denning에 의해서 최초로 Model이 제안되었다. 이 보안기술은 정보접근, 조작, 시스템무력화 등의 특정 시스템에 불법적으로 접속하여 시스템을 사용, 오용, 남용하는 침입자들을 감지하고 문제점에 대응하는 기술을 말한다.[1,2,3]

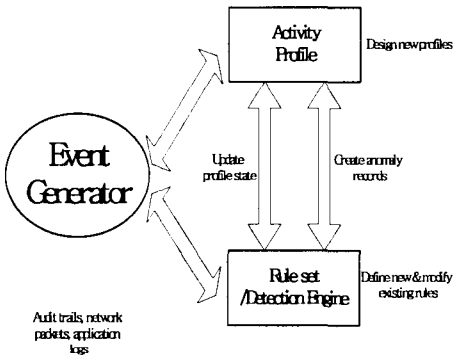

그림 1. 침입탐지시스템의 모델

III. Libpcap의 구조 및 패킷 필터링 (Packet Filtering)

네트워크를 통하는 모든 데이터들은 패킷의 형태를 띠고, 패킷이 어디로 향하고 있는지 어디서 왔는지, 어떤 종류의 패킷인지 그리고 그 밖의 관리 상 필요한 세부 사항이 적혀 있다. 이 부분을 패킷의 헤더(header)라 부르고 나머지 부분에는 전송하고자 하는 실제 자료가 들어있으며 몸체(body)라 부른다.[4,5,6] Libpcap를 이용한 패킷 추출 과정은 그림 2와 같으며, 파일을 다룰 때 사용되는 File Descriptor와 유사한 개념의 Packet Capture Descriptor를 사용한다. Packet Capture Descriptor는 다음과 같은 구조를 가진다.

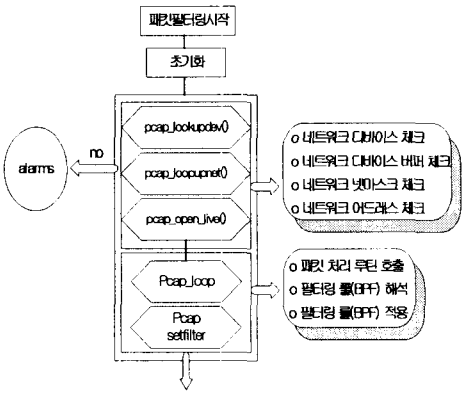

그림 2. Libpcap를 이용한 패킷 추출 과정

IV. 침입 탐지 시스템의 구현

본 논문에서 구현한 침입탐지시스템은 그림 3과 같이 Manager, Database, Agent, Sensor의 4부분으로 구성되어 있다.

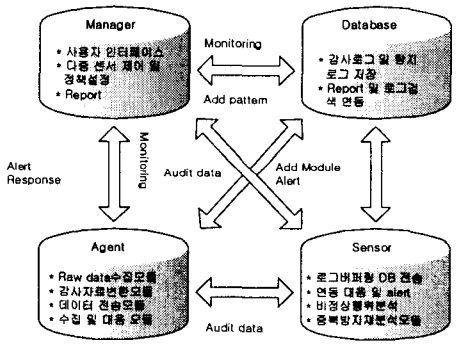

그림 3. 제안한 침입탐지시스템의 블록 구성도

구현한 침입탐지시스템은 다음과 같은 순서로 동작한다.

o. 필터링 파일로부터 필터링 룰을 불러들인다.

특정 서버만을 보호하기 위한 침입탐지 시스템이라면 모든 패킷을 감사할 필요가 없으므로 필터링을 통해 해당 프로토콜을 가진 패킷만 신속히 처리한다.

```
void packet_cap::Open()
{
fopen("FILTER_RULE", "r");
fgets(FILTER_RULE, STDBUF, FP);
}
```

o. 룰 파일을 읽어들여 메모리에 로드시킨다.

룰 구조체에 룰을 파싱하여 공격 코드와 대응 메시지 등을 집어넣는다.

```
void Detect::Rule_Reader(void)
{
fopen("./attack_rule", "r");
for(rule_count = 0; rule_count < RULE_NUM;
rule_count++)
{
AttackRule[rule_count] = ReadRule();
ParsingRule();
}
}
```

ㅇ. PCAPLIB를 통해 네트워크로부터 패킷을 수집한다. 라이브러리는 아래와 같은 순서로 사용하게 된다.

```
pcap_lookupdev()->pcap_lookupnet()->pcap_
open_live()->pcap_loop()
```

ㅇ. 캡쳐된 패킷을 분석 체크한다.

IP Header, TCP Header, UDP Header, ICMP Header등 각 프로토콜별로 정상적인 필드 값을 가지고 있는지 체크를 한다. 비정상적인 패킷일 경우 경고를 한다.

```
//정상적인 패킷 체크
if(PACKET->pkth->caplen
< ETHER_HEADER_LENGTH)
{
cout << "Capture Data Length < Ethernet
Header
Length";
return;
}
//필드 체크
if(!PACKET->TCP_HEADER->ack)
PACKET->ack = 0x01;
//프로토콜별 디코드 함수를 호출한다.
switch(PACKET->IP_HEADER->protocol)
{
IPPROTOTCP:
DecodeIP(PACKET *P);
break;
IPPROTOUDP:
DecodeUDP(PACKET *P);
break;
IPPROTOICMP:
DecodeICMP(PACKET *P);
break;
}
```

ㅇ. 침입 탐지를 시작한다.

미리 읽어들인 룰 파일과 패턴 매칭을 실시한다. 매칭 될 경우 침입으로 판정하여 로그를 남기고 이에 대응하며 매칭 되지 않을 경우 다음 패킷을 읽어들인다.

```
for(i = 0; I < RULE_NUM; I++)
{
//검사하는 패킷을 매칭 코드와 검사해 일치할 경우 체
크섬을 증가시킨다.
//모두 매칭 되었을 경우 체크섬 값은 매칭 코드의 사
이즈의 길이와 같게된다.
for(j = 0; j < rule_size; j++)
if(*(buf + j) == Attack_Rule[i].Code[j]
checksum++;
```

```
//침입으로 판정되었을 경우
if(checksum == rule_size)
{
cout << "Found Attack!!\n";
//공격지의 주소와 공격 목표점의 주소를 보여
준다.
cout << SrcAddress << "-->" <<
DstAddress;
//어떤 공격이 들어왔는지를 보여준다.
cout << Attack_Rule[i].msg;
//공격임이 판명 되었으므로 로그를 남긴다.
LogToFile();
}
}
```

V. 침입탐지시스템의 구현 결과

침입탐지시스템은 C++환경에서 구현을 하였으며, g++ 컴파일러를 이용해 링크하였다. 컴파일시에는 반드시 libpcap library가 설치되어 있어야하며 실행할 때 반드시 root의 권한으로 실행해야 한다. 대몬 모드로 동작하게 하려면 "-d" 옵션을 주면 된다.

```
#make
g++ -c main.cpp
g++ -c packet_cap.cpp
g++ -c analyzer.cpp
g++ -c detect.cpp
g++ main.o packet_cap.o analyzer.o
detect.o log.o -o hl-ids -lpcap

#./hl-ids
Initializing Interface Device....
Device : eth0
Network : 211.213.45.0
Mask : 255.255.255.0
Now Detecting....
```

본 논문에서 구현된 침입탐지시스템이 가상으로 구현된 네트워크에 외부의 공격을 적용한 실험결과이다.

다음의 그림은 Source IP 203.237.111.177에 대한 외부에서 공격이 들어왔을 때의 화면이다. 공격이 탐지되면 탐지시간과 탐지 내용, 출발지 주소, 도착지 주소 등의 정보가 남게된다. ps는 서버에서 어떤 프로세스들이 동작하는지 알 수 있게 해주는 명령어이다.

ps명령어를 실행하여 프로세스들을 확인하고 감시 변조 가능하기 때문에 외부에서 접근 시 탐지할 필요가 있다. 그림은 유닉스 명령어 ps 커맨드를 웹상에서 요청했을 시 탐지된 것이다. 그림 4은 유닉스 명령어 ps 커맨드를 웹상에서 요청했을시 탐지한 것이다.

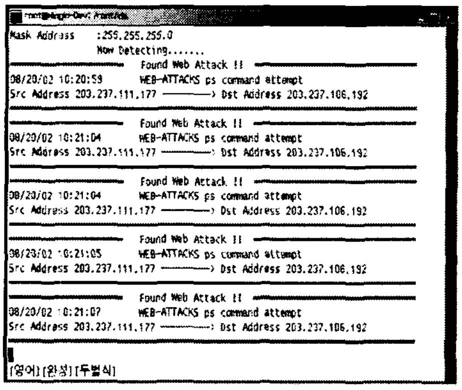

그림 4. 유닉스 명령어 ps 커맨드를 웹상에서 요청했을시 탐지

nc는 네트워크 유틸리티이지만 스크립트나 포트 연결 등을 통해서 해킹에도 이용된다. 이런 nc 명령어 또한 침입에 이용되기 때문에 탐지할 필요가 있다. 그림 5은 유닉스 명령어 nc를 웹으로 요청했을 시 탐지된 것이다.

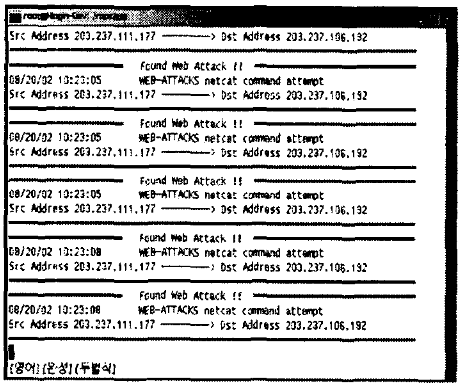

그림 5. 유닉스 명령어 nc를 웹으로 요청했을시 탐지

일반적으로 bufferoverflow나 formatstring공격의 마지막 단계는 이 쉘을 띄우는 것이다. 공격 프로그램이 /bin/sh에 해당하는 코드를 보냄으로서 공격을 수행한다. 그림 6은 리눅스 루트 쉘을 얻기 위한 공격을 했을 시 탐지된 것이다.

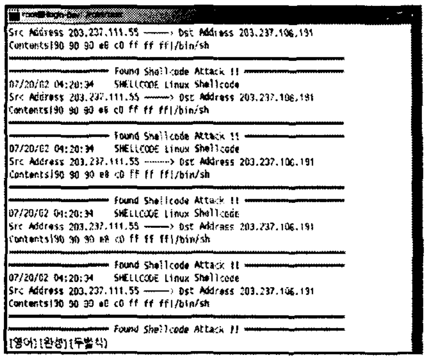

그림 6. 리눅스 루트 쉘을 얻기 위한 공격을 했을 시 탐지

IGMP는 LAN상에서 호스트와 라우터사이에서 그룹에 대한 정보를 교환할 때 사용되는 프로토콜로서 IGMP는 그룹에게 전달되는 메시지 중 목적지 IP 및 정보를 조작해서 한 호스트에게 집중시켜서 서비스 거부 상태에 이르게 된다. 그림 7은 Denial Of Service 공격중의 하나인 IGMP 누크를 탐지한 것이다. pop3나 메일 관련 공격은 25번 포트를 통해서 이루어지며 buffer overflow는 쉘을 얻기 위한 공격이다. 메일데몬의 취약점을 이용해서 공격하게 되는데 25번 포트를 사용하기 때문에 공격 또한 25번 포트로 들어올 것이고 /bin/sh코드를 보내기 때문에 25번 포트와 /bin/sh 코드를 감지해서 침입을 알 수 있다. 그림8은 리눅스 POP3 서버 버퍼 오버플로우 exploit 탐지한 것이다.

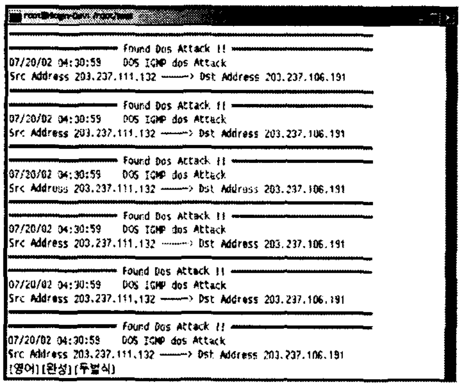

그림 7. Denial Of Service 공격중의 하나인 'IGMP 누크' 탐지

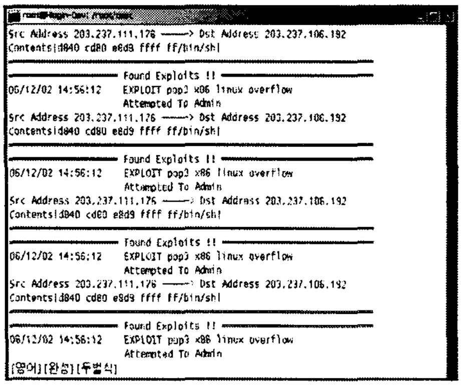

그림 8. 리눅스 POP3 서버 버퍼 오퍼플로우 exploit 탐지

VI. 결 론

본 논문에서는 패킷캡처를 바탕으로 한 IDS 모델을 제안하고, 이를 설계 및 구현하여 그 타당성을 보였다. 제안한 IDS 모델은 특권프로세스로 하여 이것들의 수행을 모니터링하여 침입여부를 판단한다. 제안한 IDS 모델은 특권 프로세스 행위를 정상 행위와 다르게 하는 경우 이를 침입으로 탐지할 수 있는 비정상 행위 침입탐지시스템이다. 본 논문에서는 침입탐지시스템의 타당성을 입증하기 위한 프로토타입을 단일 시스템에서 구현하여 탐지시간, 탐지정확도의 관점에서 제안한 모델의 성능을 평가하였다. 가상환경 하에서의 침입에 대한 평가를 해본 결과 기존의 HIDS나 NIDS의 문제점을 보완하기에 충분하였다. 본 논문에서 구현된 침입탐지시스템은 통신망 운영 및 이용기관의 인프라에서 사전 취약점 점검을 통해 전산망 침해사고를 예방 및 피해를 최소화하고, 효과적으로 침해사고에 대응할 수 있을 것으로 생각된다.

향후 연구과제는 룰 파일을 실시간으로 업데이트할 수 있도록 DB 서버와의 연동 및 새로운 룰이 침입탐지시스템이 설치된 서버로 자동으로 갱신 될 수 있도록 해야 할 것이며, 윈도우의 GUI환경을 이용해서 관리자가 쉽게 상황을 보고 대처 할 수 있는 매니지먼트 프로그램의 개발이 필요할 것이다.

참고문헌

[1] James Cannady, Jay Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies," 1998.2.

[2] Mansour Esmaili, Rei Safavi-Naini, "Case-Based Reasoning for Intrusion Detection, "Computer Security Applications Conference PP.214-222. 1996.

[3] 이종성, 채수환, "분산 침입 탐지 에이전트를 기반으로 한 지능형 침입탐지시스템 설계," 한국정보처리학회 논문지 제6권 제5호, 1999년 5월

[4] Herve Debar, Marc Dacier and Andres Wespi, "Towards a Taxonomy of Intrusion-Detection Systems", Research Report of IBM Research Division, Zurich Research Laboratory, Jen, 1998

[5] Taimur Aslam, Invan Krsul and Eugene Spafford, "Use of A Taxonomy of Security Faults". In 19th National Information System Security Conference Proceedings, Baltimore, MD, Oct. 1996

[6] Denning, Dorithy, "An Intrusion-Detection Model", IEEE Transaction on Software Enginneering, Vol. SE-13, No.2, Feb.1987

[7] Mansour Esmaili, Rei Safavi-Naini, "Case-Based Reasoning of Intrusion Detection, "Computer Security Applications Conference PP.214-222, 1996

[8] Teresa F. Lunt. "Detecting intruders in computer systems," 1993 Conference on Auditing and Computer Technology, 1993.

[9] Karl Levitt, Calvin Ko, and George Fink. "Automated detection of vulnerabilities in privileged programs by execution monitoring," 1994 Computer Security Application Conference, 1994.