데이터 마이닝 기반 침입탐지 패턴 알고리즘의 설계 및 구현

이 상 훈† · 소 진††

요 약

본 논문에서 우리는 방대한 패킷 데이터로부터 침입탐지를 위한 규칙들을 자동으로 생성하는 방법으로 기존 연관규칙의 연역적 알고리즘을 분석하고, 그 결과를 기반으로 침입탐지 시스템에 적용되기 위한 침입 패턴 규칙들을 빠르게 생성할 수 있도록 연역적 알고리즘을 제안하였다. 본 논문에서 제안하고 있는 연역적 알고리즘은 대량의 데이터를 항목별로 분류하고 제거하는 클러스터링 개념에 적합하도록 설계하였다. 이 알고리즘은 적용될 침입탐지 시스템 패턴생성 및 분석 모듈방식에 직접적으로 연계되어 있으며, 이것은 침입탐지 시스템에 관한 패턴관리를 위한 규칙 데이터베이스를 구축함으로서 응용범위의 확장은 물론 기존 침입탐지 시스템의 탐지속도를 높일 수 있다. 제안된 연역적 알고리즘의 패턴 생성 기법은 침입탐지 시스템에서 생성되는 데이터의 지원율에 따라 적절히 변경될 수 있는 알고리즘을 사용하였으며, 이 기법에 의한 규칙 생성율의 향상에 따른 규칙생성 속도개선 가능성에 대해 알고리즘 시뮬레이션을 통하여 분석하였다.

Design and Implementation of the Intrusion Detection Pattern Algorithm Based on Data Mining

Sang-Hoon Lee† · Jin Soh††

ABSTRACT

In this paper, we analyze the associated rule based deductive algorithm which creates the rules automatically for the intrusion detection from the vast packet data. Based on the result, we also suggest the deductive algorithm which creates the rules of intrusion pattern fast in order to apply the intrusion detection systems. The deductive algorithm proposed is designed suitable to the concept of clustering which classifies and deletes the large data. This algorithm has direct relation with the method of pattern generation and analyzing module of the intrusion detection system. This can also extend the application range and increase the detection speed of exiting intrusion detection system as the rule database is constructed for the pattern management of the intrusion detection system. The proposed pattern generation technique of the deductive algorithm is used to the algorithm which can be changed by the supporting rate of the data created from the intrusion detection system. Finally, we analyze the possibility of the speed improvement of the rule generation with the algorithm simulation.

키워드 : 데이터마이닝(Dataminung), 연관법칙(Association rules), 침입탐지패턴 알고리즘(Intrusion Detection Pattern Algorithm)

1. 서 론

현대사회는 정보화 사회로 특징지어진다. 과거와는 달리 현대사회는 구조적으로 대단히 복잡한 양상을 띠고 있으므로 처리해야 할 지식이나 정보의 양이 폭발적으로 증가하고 있다. 현대 사회에서 인터넷 및 인트라넷 등의 네트워크 기반구조는 기존의 도로망 및 철도 등의 산업 기반구조 못지않게 중요한 산업 기반이 되어 가고 있다. 네트워크시스템의 정보를 안전하게 유지하기 위한 보안은 여러 가지 영역이 존재하는데 침입 차단, 침입탐지, 암호화, 인증, 접근제어 등의 수많은 보안 기술들이 기업 및 개인의 정보를 보호하기 위해서 사용되고 있다[1].

네트워크 보안 및 관리체계는 정보 사회에 있어서 매우 중요한 역할을 담당하고 있으며, 개인 및 기업의 정보들을 정보 범죄들로부터 안정적이면서 효율적으로 보호하고, 네트워크 상에 존재하는 각종 자원들을 감시하기 위해 반드시 필요하다. 네트워크 상에서 수집한 수많은 종류의 패킷은 이를 실현하는 네트워크 정보의 최소 단위로서, 이를 분석함으로써 외부로부터의 침입을 사전에 탐지 할 수 있는 체계를 구성할 수 있다.

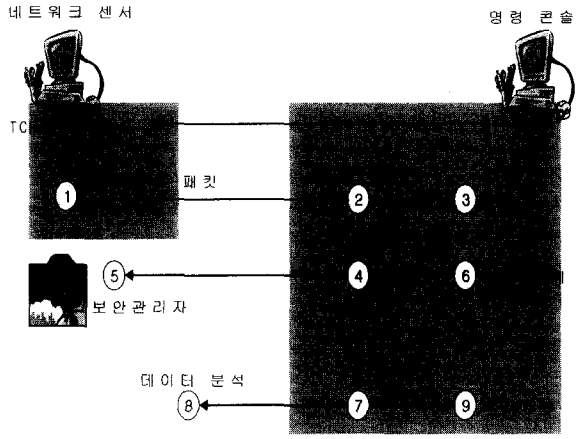
패킷정보는 네트워크 트래픽을 통해 돌아다니는 작은 흔적이라 할 수 있으며, 불법 접속자들은 일반 사용자들과 동일한 경로로 수많은 패킷들로 구성되어 송·수신되는 정보들을 사용하여 개인정보 및 시스템 자원을 위협하기 때문에 네트워크 시스템의 보안관리 중요성은 더욱더 강조되고 있는 것이다. 이를 위해서는 불법 접속자들이 침입을 목적으로 하는 패킷이 어떤 패킷이고, 또 어떤 유형의 패킷이

† 종신회원 : 국방대학교 전산정보학과 교수
†† 준 회 원 : 공군 정비창 전산장교
논문접수 : 2003년 7월 22일, 심사완료 : 2003년 9월 15일

침입에 사용되는지를 알아야만 한다[11].

네트워크 기반 침입탐지 시스템(Network Based Intrusion Detection System : 이하 NIDS)은 침입내용, 침입시간, 침입유형 등 다양한 침입기록이 저장된 로그정보를 이용한 호스트 기반 침입탐지 시스템보다 빠르게 탐지할 수 있으며 실시간 처리와 내·외부 침입 흔적에도 탐지가 가능한 역추적 시스템으로 크게 부각되고 있다.

본 논문에서는 NIDS가 침입탐지 능력을 향상시키기 위한 방법으로 분산된 대량 데이터 집합에 대한 연관성을 이용함으로써 새로운 침입유형을 탐지하고 예측하는 데이터 마이닝 기법을 제안한다. 데이터 마이닝 기법 중에서는 대량의 데이터들의 상관관계를 처리하여 불규칙성 속에서 새로운 특징을 추출하는 연관 규칙(Association Rule) 기법을 이용하였는데, 이 기법은 데이터간의 상호관련을 이용해 알려지지 않은 규칙을 발견하는 방법으로 수학적, 통계학적 그리고 기계-학습론적인 방법을 사용한다. 본 논문에서는 연관규칙을 이용하여 네트워크 데이터에서 공통적으로 관측된 규칙 집합들을 생성함으로서 보이지 않은 공격에 대한 탐지를 예측할 수 있도록 패킷 내에 데이터들간의 상관관계를 분석하였다.



(그림 1) 일반적인 NIDS 구조

본 논문은 데이터 마이닝 기법 중 연관규칙을 이용한 기존의 연역적 알고리즘을 보완하는 것으로, (그림 1)의 이반적인 네트워크 기반 침입탐지 시스템 전체 구조 중 ②단계인 탐지 엔진부분의 침입탐지 규칙생성 부분에 대한 내용을 연구범위로 한다. 연구 방법은 기존 수기식 규칙생성을 연관규칙을 이용하여 자동으로 생성하는 방법을 제안하였다.

본 논문은 다음과 같이 구성되어 있다. 제 2장에서는 데이터 마이닝의 개념과 침입 데이터에 대한 유형에 대해 설명하고, 제 3장에서는 데이터 마이닝 기법 중 연관규칙을 알아보고, 연관 규칙의 연역적 알고리즘에 대해 분석하였다. 그리고, 기존 시스템의 패턴 생성부분을 보완하기 위해 제안된 연역적 알고리즘을 제안하고, 제 4장에서는 제안된 알고리즘에 따라 시뮬레이션을 통해 평가하고 결과를 분석하였다. 제 5장에서는 연구에 대한 결론과 향후 연구방향에 대하여 정리하였다.

2. 관련 연구

2.1 데이터 마이닝

데이터 마이닝이란 방대한 양의 데이터 속에서 쉽게 드러나지 않는 유용한 정보를 찾아내는 개념적인 정보추출 방법론이다. 다시 말하면, 대용량 데이터에 존재하는 데이터간의 관계, 패턴, 법칙 등을 찾아내고 모형화해서 조직의 경쟁력 확보를 위한 의사결정을 돕는 유용한 정보로 변환하는 일련의 과정이다[2]. 1990년대 초반부터 지식발견(KDD : Knowledge Discovery in Database), 정보발견(Information Discovery), 정보수확(Information Havesting) 등의 이름으로 소개되어 왔는데, 일반적으로 대량의 데이터로부터 새롭고 의미 있는 정보를 추출하여 의사결정에 활용하는 작업이라 정의된다. 1995년 캐나다 몬트리올에서 개최된 지식발견과 데이터 마이닝에 관한 국제학술대회에서 데이터 마이닝은 지식발견 프로세스 중에서 데이터로부터 정보를 추출하기 위해 기법을 적용하는 단계라고 제안하였다. 또한 데이터베이스로부터의 지식발견이라고도 정의되는데 차이점은 데이터 마이닝라는 용어는 일반적으로 통계학자, 데이터베이스 연구가, 기업체 등에서 많이 사용한 반면, 지식발견은 인공지능이나 전문가 시스템관련 연구에 주로 등장한다는 점이다. 지식발견 프로세스는 데이터로부터 유용한 정보를 발견하는 프로세스의 전 과정이라 정의하며 데이터 마이닝은 지식발견 프로세스 중에서 데이터로부터 정보를 추출하기 위해서 기법을 적용하는 특정한 과정이라 정의할 수 있다.

데이터 마이닝의 기본개념은 새로운 것이 아니라 인공지능 분야의 기계학습(Machine Learning) 이론에 그 뿌리를 두고 있다. 기계학습은 규칙을 찾아내기 위한 자동화된 유도과정(Inductive Process)이라 할 수 있는데, 트레이닝 셋(Training Set)이라 불리는 적은 양의 실험용 데이터를 사용하여 알고리즘을 만들어 낸다. 이처럼 데이터 마이닝은 현실 세계의 대규모 데이터베이스를 트레이닝 셋으로 간주해서 이로부터 유용한 지식을 캐내는 작업을 수행하는 것이다.

2.2 침입탐지 시스템(Intrusion Detection System, 이하 IDS)

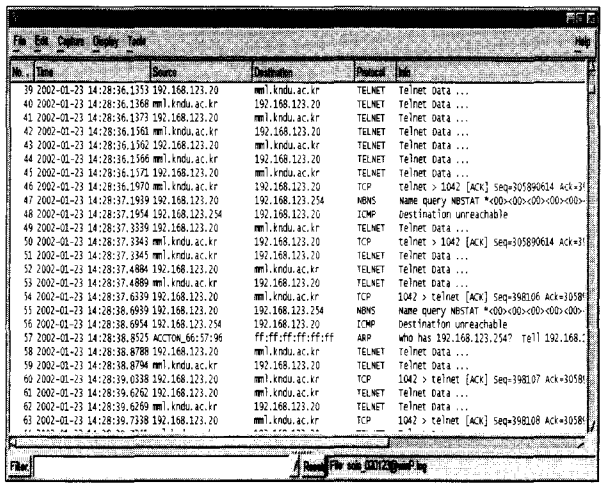
침입탐지 시스템은 네트워크 또는 각 호스트 환경 하에서 불법 행동 및 트래픽 침입패턴을 탐지·대응·보고하는 시스템이라 정의할 수 있다. 탐지 데이터는 로그파일, 프로세스 정보, 네트워크 패킷 등을 이용하며, 탐지방법은 비정상행위(anomaly)나 오용(misuse)에 대해 탐지하고, 대응방식은 운영체제의 로그파일을 분석하여 탐지하는 수동방식

과 해킹 세션에 대해 패킷을 없애는 능동적인 방식이 있다.

데이터 마이닝 기법을 침임탐지에 적용하는 주요 목적은 네트워크 데이터를 수집한 IDS가 패턴 중심의 네트워크 트래픽에 대한 규칙들(Rules)을 어떻게 실행시키고, 현재 처리되는 데이터는 얼마나 많으며, 어떻게 보고하고(Report) 표시(display)하는가, 시스템이 원하는 데이터는 무슨 종류의 데이터이며, 그 데이터를 획득하려면 어떤 질의 구조를 가지고 있어야 하는 것인가 등 다양한 문제가 발생된다[6]. 이를 위하여 네트워크를 PROMISC 모드로 개방하여 필터링을 통해 Ethernet, TCP, IP, UDP 헤더(header)등에서 자료를 수집하고, 수집된 감사자료는 탐지 모듈을 거쳐 보고 모듈과 인터페이스를 통해 관리자에게 침입보고 및 대응을 알려준다.

2.3 침입탐지를 위한 마이닝 데이터

성공적인 마이닝 기술을 지원하여 침입탐지 시스템을 개발하는데 있어서 가장 중요한 것은 현재 네트워크의 명확한 데이터들 중에서 특징을 가지고 있는 패턴규칙 집합의 초기화작업이다. 차후 이 규칙은 추가, 삭제되고 수정되어지면서 정확성을 유지하고 학습되는데 이러한 변화과정에서 추출되는 규칙집합들은 초기 규칙집합과는 분명히 차이가 있다는 점이다. 또한, 데이터 레코드는 많은 속성을 가진다. TCPDUMP 레벨에서 추출된 데이터 형태로는 소스 및 목적지 IP주소, Port 번호, Date/Time, 전송프로토콜(TCP, UDP, ICMP, etc), 트래픽 기간(Traffice Duration) 등이다. (그림 2)는 추출 프로그램을 이용하여 각종 헤더에서 추출된 데이터를 보여주고 있으며, 추출 프로그램으로는 Tcpdump, Tcptrace, libpcap 등이 있다[9, 10].



(그림 2) 패킷 필터에 의한 각 프로토콜 내용

다음은 패킷에서 얻을 수 있는 데이터 형태들이다.

- Packet Type : IP, UDP, ICMP, TCP 등
- 원천지 IP addr. : 패킷을 보낸 시스템 주소
- 목적지 IP addr. : 패킷의 도착지 주소
- Source TCP/UDP Port : 연결을 시도하는 호스트의 서비스 포트 번호
- 목적지 TCP/UDP Port : telnet, ftp, nfs 등과 같은 서비스 포트 번호

특징을 추출하는 방법은 몇 가지 다른 레벨을 가지고 다음과 같이 정의할 수 있다.

① 패킷(Packet) : service type, address, any spectial flags 등

② 연결상태(Connection) : 패킷들이 새롭게 추가되면서 패턴의 특징 부여하며, 예를 들면, 접속시간(duration), 전송된 데이터 양 등

③ 작업내용(Contents) : 텍스트 분석과 같이 상대적으로 마이닝적용에 희박한 레벨로서, 유용한 탐지 정보로는 웹브라우저나 E-mail의 IP header 정보가 있다.

(그림 3)은 데이터 필터링 TCPDUMP를 통해 얻은 패킷 레벨 정보를 보여준다.

```
[jso@mml_db log]$ tcptrace -pw 0325@1142-snort.log
Running file '0325@1142-snort.log'
Checking for file format 'tcpdump' (tcpdump -- Public domain
        program from LBL)
Using 'pcap' version of tcpdump
Tcpdump format, physical type is 1 (Ethernet)
File format is 'tcpdump' (tcpdump -- Public domain program
        from LBL)
Trace file size : 672 bytes
Packet 1
        Packet Length : 92
        Collected : Sat Mar 25 14 : 47 : 25.482322 2000
        ETH Srce : 00 : 10 : a4 : ec : 54 : e0
        ETH Dest : 00 : 80 : c8 : 8f : 9e : bc
              Type : 0x800 (IP)
        IP VERS : 4
        IP Srce : 192.168.0.210
        IP Dest : 192.168.0.230
              Type : 0x11 (UDP)
              HLEN : 20
                TTL : 128
                LEN : 78
                  ID : 4170
              CKSUM : 0xa74c
            OFFSET : 0x0000
        UDP SPRT : 137
        DPRT : 137
        UCKSUM : 0xbb79
        DLEN : 58 (only 50 bytes in dump file)
6 packets seen, 0 TCP packets traced, 6 UDP packets traced
elapsed wallclock time : 0 : 00 : 00.045390, 132 pkts/sec analyzed
trace file elapsed time : 0 : 00 : 07.508062
        first packet : Sat Mar 25 14 : 47 : 25.482322 2000
        last packet : Sat Mar 25 14 : 47 : 32.990385 2000
no traced TCP packets
```

(그림 3) TCP header 정보 추출

3. 연관규칙의 연역적 알고리즘 제안

3.1 기존 기법

데이터 마이닝을 이용한 패턴인식 기술은 주로 행동패턴간의 상관관계나 추가적인 패턴을 표현하는 기법을 사용한다.

- 분류(Classification)

데이터 레코드의 카테고리 생성하고 특별한 한 레코드에 속하는 카테고리를 예측한다. 개인 공격에 대한 탐지에 사용되며 미리 정의된 기술에 따라 동일한 패턴에 대한 카테고리를 생산한다. 이에 속하는 기술로는 신경회로망의 Kohonen Network과 SOM, K-mean 알고리즘, VQ, LVQ등이 있다[7].

- 연관 규칙(Association Rule)

시스템 특징들 가운데의 상관성을 나타낸다. 즉, $X \rightarrow Y[c, s]$, c : *confidence*, s : *support* 형식으로 표현하며, 셀 명령으로 *Mail* → *am*, *hostA* [0.3,0.1] 레코드들 안에 관계를 기술한다. 연관 규칙 기법의 특징은 정상적인 패턴을 정의하고 많은 레코드들이 불규칙성을 탐지하여 변칙적 행동을 발견하는 마이닝 기술로서 침입탐지에 널리 이용한다[4].

- 빈번한 에피소드(Frequent Episodes)

연속적인 정보 행동패턴에 기반을 둔 기술로서 서로 다른 레코들에서 동시에 발생하는 레코드들을 인식함으로서 시간에 따른 데이터 흐름에서 관계성을 기술한다 형식은 $X, Y \rightarrow Z\ [c, s, w]$으로 나타내며 X, Y, Z는 서로 다른 레코드들이며, w는 초(second) 단위를 나타내며 이러한 레코드들이 w초동안에 발생한다는 것을 의미한다. 예를 들면,

(*vi*, *C*, *am*) → (*gcc*, *C*, *am*) [0.6, 0.2, 5]은 vi editor로 C 언어를 사용하면 gcc 컴파일러를 사용하는 패턴이 5초안에 행동하는 c와 s의 기대치를 나타낸다[4].

- Clustering

유사한 특징을 나타내는 레코드의 그룹으로 어떤 특정한 카테고리안으로 데이터를 수집함으로서 유사성을 생성하거나 외부자 및 규칙 카테고리를 형성하는데 사용된다[2].

- Meta-rules

시간에 따른 규칙집합 속에 변동사항을 기술하여 패턴분석 자료를 제공하고 서로 다른 두개의 집합을 4가지 분류로 고려하여(expired, changed, unchanged, new) 어떠한 제한사항에 따라 기존 규칙에서 메타 규칙을 생성 및 저장하는 이론이다[3].

3.2 연관 규칙

연관 규칙이란 방대한 데이터베이스에서 다른 항목간의 관계성을 말한다. 이 규칙은 데이터 안에 존재하는 항목간의 종속 관계를 찾아내는 것으로 그 개념을 자세히 살펴보면 다음과 같다[8].

연관 규칙은 "$X \rightarrow Y\ s, c$"라고 표현하며, 여기서 X와 Y는 항목들의 집합이다. 이 규칙은 "~이면, ~이다"라는 'IF-THEN' 형식을 취한다. 즉, "항목 X이면 항목 Y이다"라는 뜻으로 주어진 항목집합 X와 Y에 얼마만큼 분포되어 있는지를 판단하는 지원율(s : *support*)과 이 규칙이 어느 정도 신뢰할 수 있는지를 판단하는 신뢰율(c : *confidence*)를 계산하여 표현한 규칙이다. 예를 들어 "*vi* → *pm* (0.33,0.98)"라는 규칙이 생성되었다고 하자. 이 규칙의 의미는 "편집기 *vi*를 사용한다면, 사용자의 98%는 오후(*pm*)에 편집 작업을 한다."라는 규칙이 생성되었다고 하자. 이 규칙의 의미는 *vi*와 *pm*라는 항목이 전체 항목집합에서 차지하는 지원율이 33%되고 신뢰도가 98%라는 규칙을 나타낸다.

연관규칙에서 중요한 값은 신뢰율이다. 따라서, 사용자가 정의한 최소한의 지원율과 신뢰율을 만족시키는 모든 규칙을 찾아내는 것이 연관 규칙 기법이다. 알고리즘을 유도하기 위해 연관규칙을 수학적으로 정의하면 다음과 같다[5].

$A = \{i_1, i_2, \cdots, i_m\}$은 항목(Item)이라고 하는 문자들의 집합이다. 데이터베이스 D를 트랜잭션 T의 집합이라 하자. 각 트랜잭션 T는 $T \subseteq A$를 만족하는 항목들의 집합이다. 각 트랜잭션 T는 TID라고 하는 유일한 식별자가 있다. 만약, $X \subseteq T$가 성립하면, A의 부분집합으로 구성되어 있는 X를 트랜잭션 T가 포함하고 있다는 것을 의미한다. 데이터베이스 D에서 트랜잭션 수(레코드 수)에 대한 항목집합 X의 비율을

$$support(X) = \frac{\text{항목집합 } X\text{의 갯수}}{|T|}$$

라 정의하자(단, |T|는 전체 트랜잭션 수).

$X \subset A$, $Y \subset A$, $X \cap Y = \phi$인 상황에서 다음과 같이 표현한다. 여기서 X와 Y는 항목집합이다.

$$X \rightarrow Y\ \ s, c$$

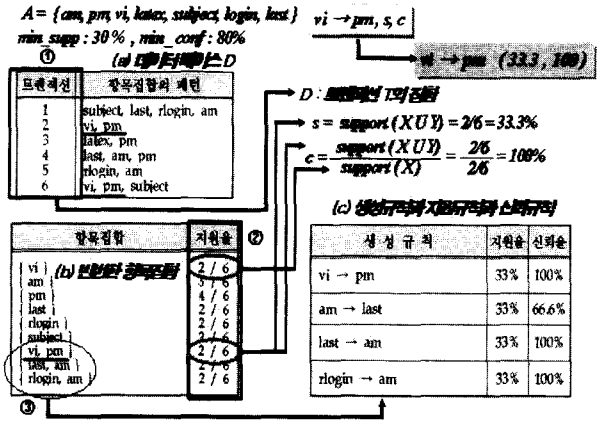
$$s = support(X \cup Y),\ \ c = \frac{support(X \cup Y)}{support(X)}$$

그리고, s는 지원율이고, c는 신뢰율을 의미한다.

연관규칙은 지원율과 신뢰율을 기반으로 두 항목간의 상관관계를 비율로서 표현함으로서 항목간의 연관성를 지원하는 것이다. 또 하나의 특징이 있다면, 제한 사항이 존재하는데, 방대한 데이터베이스 속에서는 데이터간의 연관성을 정량적으로 지정해야 한다는 것, 즉, 최소한의 경계 값으로 제한해야 한다는 것이다. 지원율과 신뢰율이 정량적으로 계산되어지기 때문에 최소한의 경계 값을 정하지 않으면 무수히 많은 규칙이 생성된다. 이러한 경계 값을 "min_supp"과 "min_conf"라고 정의한다.

트랜잭션 집합 D에서 연관 규칙을 찾아내는 문제는 사용

자가 정의한 최소 지원율과 최소 신뢰율보다 큰 값을 가진 규칙들을 생성하는 것이다. 이 값은 일반적으로 사용자가 임의로 설정하는 값으로 보통 최소 지원율은 10%, 최소 신뢰율은 70~80%정도를 가지고 계산된다.



(그림 4) 연관 규칙 활용 예제

(그림 4)는 연관규칙의 활용예제로서 항목이라는 문자들의 집합인 A = {am, pm, vi, latex, subject, rlogin, last}와 트랜잭션의 집합인 데이터베이스 D가 있다. 각 트랜잭션은 독립된 값으로 유일하며, $T \subseteq A$을 만족하는 항목들의 집합임을 확인하자. 최소 지원율은 30%이고, 최소 신뢰율은 60%으로 설정한다. 데이터 베이스가 구성되면, 규칙을 생성해야 한다.

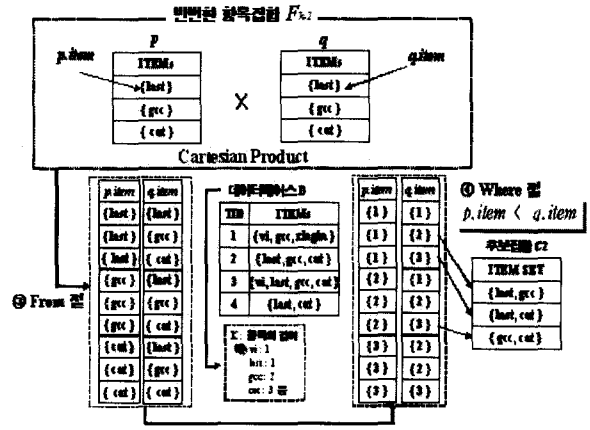
3.3 연역적 알고리즘

연관 규칙에서 규칙을 발견하기 위한 기본적인 알고리즘은 (그림 5)와 같이 세 단계로 구성되어진다[5, 7].

① 사용자가 정의하는 지원율보다 큰 지원율을 가진 항목들의 집합을 모두 찾는다. 여기서 생성된 항목집합은 빈번한 항목집합이라고 한다.
② 우리가 원하는 규칙을 생성하기 위해서는 이러한 빈번한 항목집합을 사용한다. 예를 들면, ABCD와 AB를 항목집합이라 하면,
$conf = \frac{support(ABCD)}{support(AB)}$, AB → CD라는 규칙이 생성되는 것이다. 여기서, 조건 $conf \geq \min conf$ 만족하면, 완전하게 규칙을 생성되는 것이다. 이렇게 생성된 규칙은 ABCD가 빈번하기 때문에 최소한의 지원율을 가지는 것이다.
③ 이러한 규칙들 중에서 최소 지원율보다 낮은 규칙들을 제거한다.

(그림 5) 연관 규칙의 기본적인 알고리즘

알고리즘을 설명하면, 알고리즘의 첫 번째 단계는 초기 빈번한 항목집합을 결정하기 위해서 항목들의 개수를 계산한다. 각 선택되어지는 항목의 종류는 실제 값을 토대로 항목들의 개수가 카운트된다. 두 번째 단계는 두 개의 서브단계를 가진다. 첫단계는 k번째 반복에서 찾아낸 빈번한 항목집합 F_{k-1}은 후보 집합 C_k를 생성하는데 사용된다. 즉, F_{k-1}에서 C_k를 유도한다. 후보집합 생성 알고리즘은 (그림 6)과 같다. 두번째 단계는 전체 데이터베이스를 스캔하고 C_k에서 후보들의 지원율을 계산한다. 다음으로 최소한의 지원율보다 낮은 규칙들은 제거한다. 다음은 ① 빈번한 항목집합 F_k와 ② 후보집합 C_k를 생성하는 알고리즘이다.



(그림 6) 후보집합 생성 알고리즘

〈표 1〉 알고리즘 변수 표기와 의미(Notation)

k-itemset	k개의 항목을 가진 항목집합
F_k	빈번한 k-항목집합들의 집합 (최소 지원율을 가진 것들)
C_k	후보 k-항목집합들의 집합 (향후 빈번한 항목집합이 될 집합)

침입탐지 시스템 특성상 패킷을 기반으로 하기 때문에 규칙생성을 위한 의미 없는 데이터 제거할 필요가 있으며, 의미 있는 항목으로의 분류작업이 필요하였다. 또한 규칙을 생성하기 위한 방법으로 데이터 마이닝의 기법중 연관규칙을 이용하여 적용한 결과, 기존 알고리즘 역시 분류 작업 없이 항목집합 생성시 의미 없는 규칙이 생성하는 것을 확인할 수 있었다. 또한, 패턴 비교시 전체 스캔작업 보다는 조인연산을 이용하여 빈번한 항목집합을 생성하는 부분들을 추가하여 알고리즘을 구성하였다((그림 7) 참조).

```
procedure Apriori Algorithm ( )
  begin
   F1 := {frequent 1-itemsets} ;
   k := 2 ;
   While ( Fk-1 ≠ 0 ) do {
   Ck := New Candidates of size k generated form Fk-1
     forall transaction t ∈ D do {
      Add all of each item in t to Ck ,
            removing any duplicates.
      Increment the count of all candidates in Ck
            that are contained in t.
     }
```

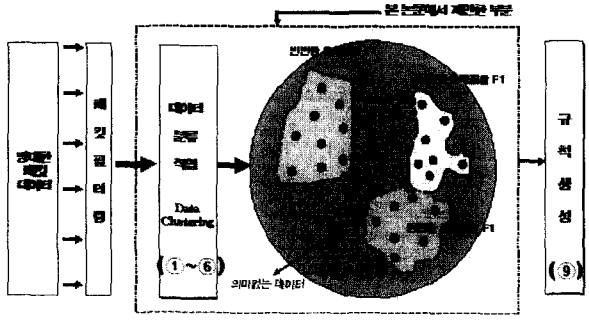
```
    F_k = All Candidates in C_k with minimun support.
    k := k + 1 ;
  }
  Answer := k F_k
end
```

(그림 7) 연역적 알고리즘

3.4 연역적 알고리즘 제안

전 절의 연관규칙의 연역적 알고리즘은 초기 빈번한 항목집합(F_1) 생성시 방대한 데이터를 한꺼번에 입력으로 사용하여 의미 없는 규칙들이 대량으로 생성되었다. 이렇게 무작위로 패턴을 생성하는 것보다는 필수적인 패턴이 무엇이며, 가장 빈번하게 사용되는 데이터와 값은 무엇인지를 구별하여 초기 빈번한 항목집합을 생성하는 방법이 필요하다. 또한, 후보집합 생성시 항목들을 일일이 검색하여 찾는 방법보다는 후보집합 생성속도 개선을 위해 각 항목들을 조인 연산하여 생성하는 방법을 제안한다. 중요한 개념은 초기 빈번한 항목집합을 구하기 전에 각 데이터를 클러스터링 하여 항목의 값에 따라 분류작업을 하는 것이다. 분류된 각 클래스는 서브-빈번한 항목집합으로 생성된다. 각 서브-빈번한 항목집합은 생성된 클래스만큼 반복적으로 빈번한 항목집합을 생성하게 된다. 이렇게 생성된 빈번한 항목집합은 기존 알고리즘과 동일하게 수행되어 후보항목집합 C_k을 생성하면서 최종적인 빈번한 항목집합의 규칙들을 생성하게 된다. 이 알고리즘은 패킷을 필터링한 방대한 데이터를 한꺼번에 입력으로 사용하는 것보다는 의미 없는 데이터를 제거하고, 특징적인 데이터를 분할하여 클러스터링한다는 내용을 가진다. 이 내용은 각 클래스로 분할된 데이터를 가지고 규칙을 생성하는 방법이 패턴을 생성하는데 인식율이 우수하다는 이론적인 배경을 고려하여 제안한 방법이다.

(그림 8)을 설명하면 방대한 네트워크상의 패킷 데이터를 필터링시킨 데이터들이 일단 로그파일에 누적되고, 누적된 데이터는 여러 개의 입력패턴인 빈번한 항목집합으로 특징패턴(Feature Pattern)에 따라 분류되어진다. 분류된 빈번한 항목집합의 항목들은 일정한 규칙들을 생성한다. (그림 9)는 제안된 알고리즘이다. 이 부분을 단계적으로 설명한다.



(그림 8) 제안된 연역적 알고리즘 모델

① 빈번하게 사용되지 않는 데이터들을 스캔하여 의미 없는 데이터를 제거하고 빈번하게 사용된 데이터로 전환하는 전처리 작업을 통해 의미 없는 데이터 및 변하지 않는 데이터 값은 전체 데이터베이스에서 제거한다.

② 일단 방대한 데이터를 한꺼번에 처리하는 것이 아니고, 데이터 항목에 따라 실제 값을 검색하고 클러스터링을 이용하여 각각의 클래스를 생성하기 위해 클래스 생성 방법으로, 우선 레코드중 임의의 항목을 선택한다.

③ 데이터베이스 분류작업은 해당 클래스 결정기준이 각 레코드와 임의로 선택된 항목들의 값과 가장 가까운 값을 해당 클래스 값으로 한다.

④ 두 항목간의 경계선은 두 중심 값과 동일한 값을 가진 지역에 위치한 항목들의 집합이 된다.

⑤ 각 클래스에 속한 레코드들의 중심 값을 재측정하여 클래스를 형성한다.

⑥ ③④⑤과정을 항목값의 변화가 거의 없을 때까지 반복한다((그림 3)~(그림 6) 참조).

⑦ 모든 클래스 개수만큼 반복적으로 분할된 F_1을 계산한다.

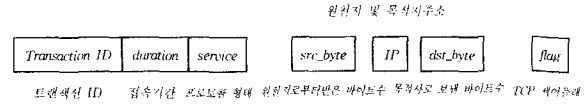
⑧ 각 클래스에서 항목들의 지원율을 생성하기 위해 전처리 작업을 한다.

```
① Delete any item in D that consist of
                uninterested an item & string vaule ;
② Select Item of Each Record Randomly ;
  for ( i = 1 ; Threshold == 0 ; i++ ) do {
③   GetValue (Shortest_Center_value) ;
④   Generating_Class_get_vlaue (Each Class Center_valuc) ;
⑤   Cluster.Class[i] := Equal_of_Item_center_value ;
⑥   }
⑦   While (Cluster.Class[i] ≠ 0) {
⑧    F1 := Pre-Processing ({Cluster.Class[i]}) ;
```

(그림 9) 제안된 연역적 알고리즘

본 알고리즘의 특징은 첫째, 기존 알고리즘의 문제점이 되었던 데이터들(즉, 의미 없는 데이터/변화가 없는 데이터)을 제거하는 전처리 부분이 있다. 둘째, 실제 값에 따라 항목집합을 데이터만으로 분류하는 작업이 수행된다. 그러므로, 분류된 각 클래스들은 부분적으로 빈번한 항목집합으로 생성되어 규칙을 생성하게 된다.

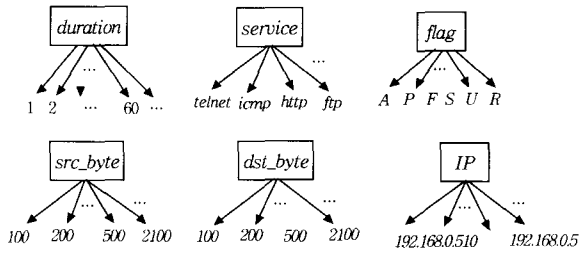
본 논문에 사용한 실험데이터는 침입탐지를 위한 패킷 데이터를 전처리하여 (그림 10)과 같은 클래스로 분류하게 된다.



(그림 10) 클래스 분류 형태

클래스의 트랜잭션 ID는 각 트랜잭션의 고유값으로 클래

스에서 제외되며 접속시간(*duration*), 프로토콜 형태(*service*), 원천지로부터 받은 바이트 수(*src_byte*), 목적지로 보낸 바이트 수(*dst_byte*), TCP 제어플래그(*flag*), 원천지 및 목적지 주소(*IP*)로 구성되었다. 접속시간 클래스는 동일한 패턴이 존재하기 때문에 규칙으로 생성이 가능하다. 값은 일반적으로 초 단위(sec)로. $duration = \{1, 2, \cdots, 10, 60\}$, 프로토콜 형태는 { *telnet, icmp, ftp, http* ···}등의 값을 가진다. TCP 제어 플래그 값은 $\{A, P, F, S, U, R\}$등의 값으로 분류되어 지는데 A는 ACKnowledgement 플래그로서 양방향 통신시 상호 인정 신호 값, P는 Push 플래그로서 telnet, rlogin시 사용되는 플래그, F는 FINish 플래그로서 데이터 전송 종료, S는 SYNchronize Sequence Numbers 플래그로서 3-way 핸드 쉐이킹시 표시하는 플래그, U는 Urgent 플래그로서 Out of band 데이터 처리시 생성되는 플래그, R은 Reset 플래그로서 연결해제를 나타낸다. IP는 일반적인 인터넷 주소를 나타낸다. 이것을 그림으로 표현하면 (그림 11)과 같다.



(그림 11) 시뮬레이션을 위한 클래스 분류

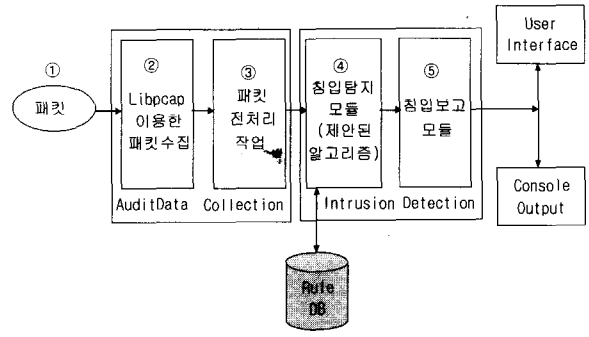
기존 연관규칙을 이용한 방법은 클래스로 나누는 작업을 하지 않고 모든 항목집합을 한꺼번에 입력하여 후보집합과 빈번한 항목집합을 생성하는데 반해 제안된 알고리즘에서는 패킷 처리된 데이터들을 스캔 작업을 통해 클러스터링하는 전처리 작업을 한다는 것이다. 그러므로, 빈번한 항목집합 및 후보집합을 생성하는 시간이 필요 없다. 따라서, 클래스로 분할하여 계산하고 조인방법을 이용하여 간단하게 후보집합을 생성하는 방법을 제안하는 것이다. 이러한 특징이 기존 연관규칙과 다른 상이한 점이다. 결과적으로, 항목들이 분류되어지는 과정에서 이 알고리즘의 특징을 확인할 수 있다. 즉, 의미 없는 데이터는 제거되고, 분류되어진 항목들은 알고리즘을 통해 재학습되기 때문에 빈도수가 많은 정확한 패턴이 생성될 수 있다는 것이다.

4. 구현 프레임워크

4.1 프레임워크 실행

제안된 데이터 마이닝 기법 중 연관규칙을 이용한 침입탐지 시스템의 타당성을 검증하기 위해 제안된 알고리즘에 따라 프로그램을 작성하였다. 실제 환경은 리눅스를 도메인 서버로 하였고, 윈도우 98 운용 하에 TCP/IP 네트워크 상에서 가상의 PC에서 사용자가 서버에 접근하여 일련의 작업을 시행한 내용을 실시간 로그로 생성하였다. 생성한 로그를 입력받아 제안된 알고리즘을 실행시켜 패턴을 생성하는 시뮬레이션을 만들어 실험하였으며, 실험 결과는 여러 가지 성능평가 척도로 이용되고, 실제 구동함으로써 그 효용성을 입증할 수 있었다.

클래스를 (그림 11)과 같은 방법을 통해 여섯 가지로 분류하였다. 각 클래스는 정의와 같이 실제 값을 분류하여 레코드 형태로 입력 파일을 형성한다. 그밖에 최소 지원율과 신뢰율은 0.1, 0.8로 정의하고 시뮬레이션을 수행하였다. 시뮬레이션 구현 프레임워크 흐름도는 (그림 12)와 같으며, 위와 같은 절차로 구현한 내용을 입력과 출력, 그리고 프로세스에 따른 결과를 출력하고 정리하였다.



(그림 12) 시뮬레이션 구현 프레임워크 흐름도

①② 패킷 및 libpcap을 이용한 수집 결과

네트워크 데이터를 수집하는 부분으로서 자신의 네트워크에 지나가는 모든 패킷정보를 수집한다.

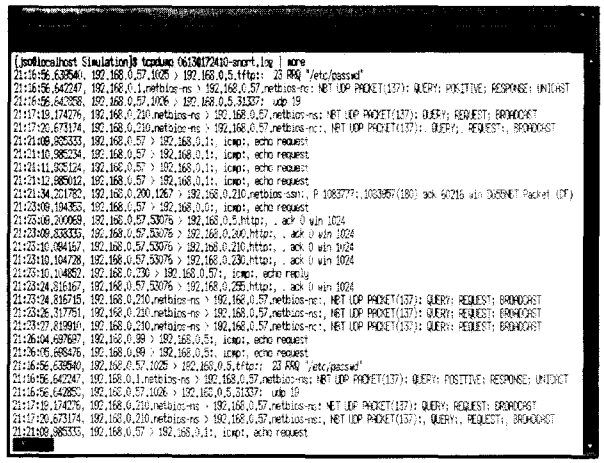
③ 패킷 전처리 작업

연관규칙에 입력되는 데이터를 생성하는 부분으로 하나의 패킷에 따라 트랜잭션 ID, 시간, 원천지 및 목적지에서 출력된 데이터 용량, 프로토콜 서비스, 플래그, 기타 데이터별로 패킷을 항목집합으로 분리한다. (그림 13)은 Tcpdump 프로그램을 통해 패킷을 전처리하여 각 분리된 항목들을 텍스트 테이블에 저장시켜 놓은 결과를 보여준다.

1) 입력 : libpcap을 이용하여 패킷을 필터링 한 데이터는 연구실의 리눅스 서버 장비에서 패킷 캡쳐 프로그램을 실험을 통해 수집한 데이터
2) 프로세스 : 패킷에 따라 각 특징별로 트랜잭션 ID, 시간, 원천지 및 목적지에서 출력된 데이터 용량, 프로토콜 서비스, 플래그 등의 식별자에 따라 분류하여 해당된 실 데이터 값을 각 항목집합으로 분류하여 테이블 형식으로 저장한다.
3) 출력 : 각 항목별로 분류된 특징들과 데이터를 저장한 테이블 형식이다.

(그림 13)의 왼쪽부터 첫 번째 항목은 시간 항목, 두 번

째 항목은 IP 주소, 세 번째 항목은 서비스(프로토콜 형태), 네 번째 항목은 패킷에 대한 내용을 나타낸다.

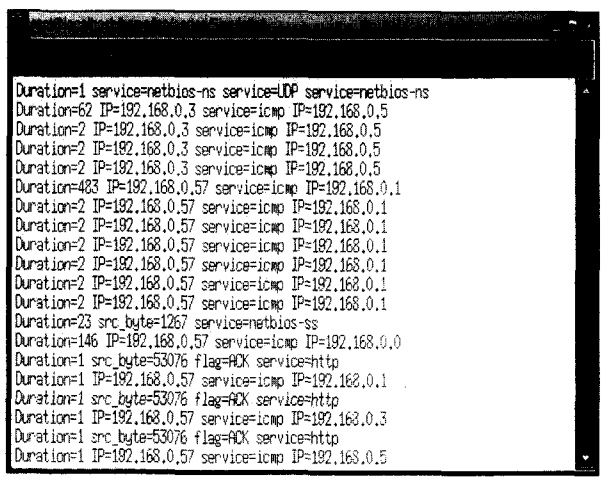


(그림 13) 패킷 전처리 작업을 한 결과

④ 침입탐지 모듈(제안된 연관규칙 적용)

다음은 침입탐지 모듈에 입력되는 패턴들과 제안된 연관규칙 알고리즘에 의해 실행하여 실질적인 패턴을 생성하는 부분이다.

1) 입력 : 패킷 전처리에 출력된 데이블 형식의 데이터 베이스로서 시뮬레이션을 위한 자료를 생성하기 위해 실 패킷을 전처리한 데이터를 기반으로 분류한 클래스에 따라 분류하는 자동 프로그램을 작성하여 실행한 결과데이터를 입력하였다.
2) 프로세스 : 입력된 데이터베이스 파일을 가지고, 각 항목집합 속에 어느 정도 빈번하게 어떤 항목이 분포되어 있으며(지원율), 항목간의 연관성이 어느 정도 신뢰를 가질 수 있는 관계(신뢰율)인지를 확률로서 계산한다. 생성된 패턴 중 가장 빈번한 규칙들을 규칙 패턴으로 확정하고 텍스트 데이터베이스에 저장된다.



```
Duration=1 service=netbios-ns service=UDP service=netbios-ns
Duration=62 IP=192.168.0.3 service=icmp IP=192.168.0.5
Duration=2 IP=192.168.0.3 service=icmp IP=192.168.0.5
Duration=2 IP=192.168.0.3 service=icmp IP=192.168.0.5
Duration=2 IP=192.168.0.3 service=icmp IP=192.168.0.5
Duration=483 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=2 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=23 src_byte=1267 service=netbios-ss
Duration=146 IP=192.168.0.57 service=icmp IP=192.168.0.0
Duration=1 src_byte=53076 flag=ACK service=http
Duration=1 IP=192.168.0.57 service=icmp IP=192.168.0.1
Duration=1 src_byte=53076 flag=ACK service=http
Duration=1 IP=192.168.0.57 service=icmp IP=192.168.0.3
Duration=1 src_byte=53076 flag=ACK service=http
Duration=1 IP=192.168.0.57 service=icmp IP=192.168.0.5
```

(그림 14) 입력 패턴 파일

3) 출력 : 연관규칙으로 표현한 연관된 항목들, 해당 지원율과 신뢰율을 계산한 확률 값을 출력한 텍스트 파일 형식이다.

항목은 (그림 14)의 왼쪽부터 접속기간, 서비스(프로토콜 형태), 원천지에서 받은 데이터량, 목적지로 보낸 데이터량, 원천지 IP 주소, 목적지 IP 주소, TCP 플래그로 구성된다.

4.2 알고리즘 분석

분석 방법은 분석 항목에 대한 정의하고, 제시한 기준에 따라 시뮬레이션을 통해 그 값을 구하여 분석하였다. 기존 연역적 알고리즘과 본 논문에서 제안된 연역적 알고리즘의 성능을 분석하기 위하여 우선 분석항목의 객관성을 보장하는데 중점을 두었으며, 그 분석 항목은 다음과 같다.

① **규칙 생성수** : 주어진 최소 지원율에 따라 입력패턴에 대한 규칙생성 수. 최소 지원율은 10, 15, 20, 30, 35, 50%에 따라 규칙생성 수를 측정하였다. 최소 신뢰율은 80%으로 규칙을 생성하였다.
② **실행시간** : 패킷을 전처리한 데이터를 연관규칙 알고리즘에 적용한 후 시뮬레이션에 소모된 총 실행시간(단위, 초(sec))

입력 패턴에 대하여 규칙을 생성하는 성능을 비교하기 위하여 규칙 생성수와 실행시간에 대한 시뮬레이션을 하였다. 기존 알고리즘과 제안된 연관규칙 알고리즘에 대한 시뮬레이션 한 결과는 <표 2>와 같다. 최소 지원율 10%에서의 생성된 항목 수는 기존 알고리즘이 139개, 제안된 알고리즘이 30개이며, 입력 트랜잭션수는 66으로 동일하고, 생성된 규칙의 수는 각각 2401개와 25개이며, 이 결과가 <표 2>의 최소 지원율 10% 항목에 나타나 있다.

〈표 2〉 제안된 연관규칙의 알고리즘의 성능 비교

(a) 기존 알고리즘

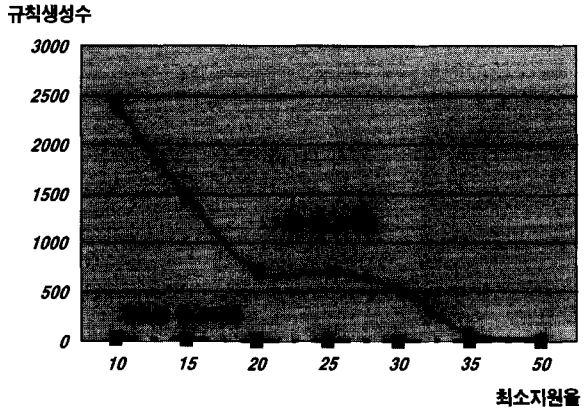
평가 항목 \ 최소지원율	10%	15%	20%	25%	30%	35%	50%
규칙 생성수	2401	1474	716	716	524	72	20
실 행 시 간	0.7	0.6	0.5	0.5	0.5	0.4	0.4

(b) 제안된 알고리즘

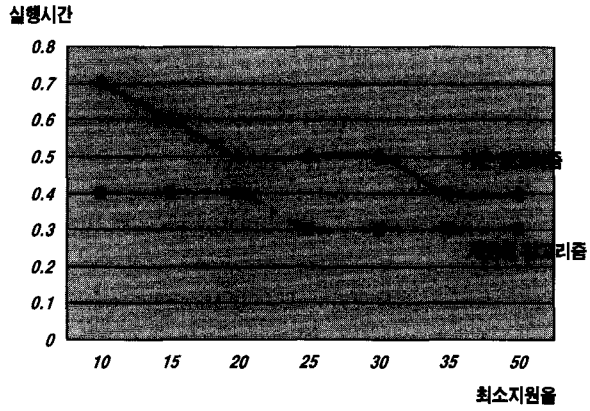
평가 항목 \ 최소지원율	10%	15%	20%	25%	30%	35%	50%
규칙 생성수	25	9	4	2	2	2	2
실 행 시 간	0.4	0.4	0.4	0.3	0.3	0.3	0.3

(그림 15)는 기존 및 제안된 알고리즘의 성능 비교를 위한 시뮬레이션 한 결과를 보여주고 있다. 앞에서 제시한 문제점에 대해 데이터를 클러스터링한 후 패턴을 생성해야

한다는 개념으로 시뮬레이션을 통해 적용한 실험한 결과, 최소 지원율이 높을수록 기존 알고리즘과 제안된 알고리즘이 규칙을 생성하는 비율이 점점 줄어가는 것을 볼 수 있다. 최소 지원율이 높다는 것은 전체 데이터베이스에서 그만큼 지지하는 항목패턴이 많이 분포되어 있으며, 빈번하게 사용되었다는 뜻이다. 최소 지원율이 증가할수록 가장 빈번하게 사용된 패턴이 규칙으로 생성되었다는 것을 의미한다.



(a) 규칙 생성수



(b) 실행 시간

(그림 15) 규칙 생성수와 실행시간 비교

(그림 15)(a)의 결과에서 보는 바와 같이, 최소 지원율이 증가함에 따라 전체적으로 규칙 생성수는 제안된 알고리즘보다 기존 알고리즘이 높게 나타났다. 기존 알고리즘의 규칙생성은 항목자체의 의미와 관계없이 항목사이의 단순하게 계산만으로 규칙을 생성하고 있다. 그러나, 제안된 알고리즘은 전처리 작업을 통해 클러스터링 한 후 빈번한 항목들을 대상으로 규칙을 생성하기 때문에 빠르게 수렴하고 있는 것이다.

(그림 15)(b)와 같이, 실행시간도 기존방식보다 빠르게 수행되고 있는 것을 볼 수 있다. 필요 없는 데이터들이 제거되고 클러스터링한 패턴들만을 연관규칙에 적용함으로서 패킷을 전처리하지 않고 입력패턴을 사용하는 기존방식보다는 소량의 입력데이터가 처리되므로 실행시간도 기존방식보다 빠르게 측정되었다.

결과적으로, 제안된 알고리즘은 기존 알고리즘을 수행하기 전에 전체 데이터베이스를 클러스터링 전처리 작업을 거쳐 입력패턴을 형성하기 때문에 규칙을 생성하는 시간이 기존 알고리즘보다 빠르게 나타났다. 즉, 이 알고리즘은 빈번하게 사용하는 패턴 규칙들을 빠르게 생성할 때 유용한 방식이다.

5. 결 론

침입탐지 시스템은 제한적인 네트워크 환경 속에서 지속적으로 처리되는 대량의 패킷에 대하여 새로운 침입에 대한 유형을 찾아내어 대응해야 하므로 빠르고 신속한 탐지는 물론 방대한 데이터 속에서의 정확한 인지능력이 요구된다. 본 논문에서는 일반적인 침입탐지 시스템과 데이터 마이닝 기법을 살펴보고, 개선해야 할 기능적인 부분인 패턴생성부분을 연관규칙의 연역적 알고리즘을 통해 제안하였다. 그리고, 제안된 알고리즘의 시뮬레이션로 그 성능을 비교해 보였다. 즉, 데이터 마이닝 기법을 네트워크 기반 침입탐지 시스템에 적용하기 위해서 많은 종류의 데이터 항목들을 내포하고 있는 패킷을 클래스로 분류하고, 데이터 항목간의 규칙을 생성하는 알고리즘을 제안하였다. 이 알고리즘은 침입탐지 시스템 전체구성에서 규칙 생성 모듈부분에 연관규칙 알고리즘을 적용하여 빈도수가 높은 패턴을 빠르게 생성하는 방법이다. 제안된 알고리즘을 시뮬레이션한 결과, 기존 알고리즘보다 빈번하게 사용된 데이터 항목의 패턴규칙을 생성하였고, 실행시간도 기존 알고리즘보다 빠르게 나타났다.

향후 연구과제로는 대량의 실제 네트워크 패킷 데이터들 속에서 패턴 식별자와 값을 분석하는 연구와 이러한 규칙들을 관리하는 데이터 베이스을 구축하여 기존 규칙과 새롭게 생성된 규칙들을 기반으로 침입을 탐지할 수 있는 엔진을 구축하는 연구가 수행되어야 할 것이다.

참 고 문 헌

[1] 이경하 외, "네트워크 패킷 정보를 기반으로한 보안 관리", 한국정보과학회논문지, Vol.25, No.12, pp.1405-1412, Dec., 1998.

[2] 김화수, "지능정보시스템 개론", 국방대학교, May, 2001.

[3] Tamas Abraham, "IDDM : Intrusion Detection using Data Mining Techniques," In Information Technology Division Electronics & Surveillance Research Laboratory, 2001.

[4] W. Lee and S. J. Stolfo. "Data mining approaches for intrusion detection," In *In Proceedings of the 1998 USENIX Security Symposium*, 1998.

[6] Ramakrishnan Srikant, Rakesh Agrawal, "Mining Generalized Association Rules," Proceedings of the 21st VLDB Conference, IBM Almaden Research Center, 1995.

[6] Kristin R. Nauta and Frank Lieble, "Offline Network Intrution Detection : Mining tcpdump Data to Identify," In http://www.sas.com/service/library/onlinedoc/itsv/intrusion.pdf, 1999.

[7] Eric Bloedorn 외 "Data Mining for Network Intrusion Detection : How to Get Started," The MITRE Corporation, In http://www.afcea.org/pastevents/db2001/Bloedorn_files/frame.htm, 2001.

[8] Wenke Lee, Salvatore J. Stolfo, Kui W. Mok, "A Data Mining Framework for Building Intrusion Detection Models," IEEE Symposium on Security and Privacy, In http://citeseer.nj.nec.com/154973.html, 1999.

[9] http://www.snort.org/.

[10] http://www.tcpdump.org/.

[11] Paul E. Proctor, "The Practical Intrusion Detection Handbook," Prentice Hall PTR, www.phptr.com, Feb., 2000.

이 상 훈

e-mail : hoony@kndu.ac.kr

1978년 성균관대학교 전자공학과(공학사)

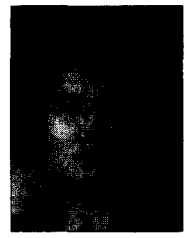
1989년 연세대학교 전자계산학(공학석사)

1997년 일본 교토대학 정보공학(공학박사)

1998년 충남산업대학교 멀티미디어과 교수

2000년~현재 국방대학교 전산정보학과 교수/전산실장

관심분야 : 협조작업처리(CSCW), 멀티미디어 데이터베이스, 객체지향 데이터베이스, 멀티미디어시스템, 정보보호



소 진

e-mail : jsoh01@hanmail.net

1994년 한남대학교 컴퓨터공학(공학사)

1997년 한남대학교 컴퓨터공학(공학석사)

2003년 국방대학교 전산정보(공학석사)

2003년~현재 공군 정비창 전산장교

관심분야 : 리눅스/자바 프로그래밍, 침입탐지시스템, 데이터베이스