情報保護學會論文誌
第12券 第5號, 2002. 10

패킷간 연관 관계를 이용한 네트워크 비정상행위 탐지

오 상 현*, 이 원 석*

Network Anomaly Detection based on Association among Packets

Sang-Hyun Oh*, Won-Suk Lee*

요 약

최근에 컴퓨터 침입으로 인한 피해가 날로 증가하고 있으며 다양한 침입 기법들이 새롭게 개발되고 있다. 따라서 침입자들의 행위를 효과적으로 탐지하기 위해서 기존의 오용탐지 방법과 더불어 비정상행위 모델의 적용에 대한 많은 연구가 진행되었다. 본 논문에서는 네트워크를 통해서 수신되는 패킷에 대한 정상행위 패턴을 생성하기 위해서 패킷 내 뿐만 아니라 패킷간의 연관성을 탐사하는 새로운 연관 규칙 알고리즘을 제안한다. 이와 더불어 다양한 실험을 통해서 본 논문에서 제안된 비정상행위 판정시스템에서 탐지율을 최대화 할 수 있는 임계치 값들을 제시한다. 결과적으로 효과적인 비정상행위 판정이 가능하다.

ABSTRACT

Recently, intrusions into a computer have been increased rapidly and also various intrusion methods have been developed. As a result, many researches have been performed to detect the activities of intruders effectively. In this paper, a new association mining algorithm for anomaly network intrusion detection is proposed. For this purpose, the proposed algorithm is composed of two different phases: intra-packet association and inter-packet association. The performance of the proposed anomaly detection system is evaluated based on several experiment according to various system parameters in order to identify their practical ranges for maximizing its detection rate. As a result, an anomaly can be detected effectively.

Keyword : *비정상 행위 판정, 침입 탐지, 데이터마이닝, 연관 규칙*

Ⅰ. 서 론

침입이란 권한이 없는 사용자가 발생시키는 문제 또는 합법적인 사용자가 권한을 남용하는 것이라고 정의한다[1]. 이와 더불어 자원의 유용성, 기밀성, 그리고 무결성 등에 저해되는 행동 집합을 침입이라고 정의하기도 한다[2]. 본 논문에서 비정상행위는 침입을 포함한 사용자 자신의 업무 권한을 벗어난 행동 일체까지를 포함한다. 일반적으로 침입 탐지 모델은 오용 탐지 모델(Misuse Detection Model)과 비정상행위 탐지 모델(Anomaly Detection Model)로 분류된다. 오용 탐지 모델은 시스템 상에서 잘 알려진 약점을 이용한 공격 탐지 방법으로 알려진 침입 패턴과 일치하는 데이터 또는 이벤트의 발생 순서등을 통해서 탐지하게 된다. 하지만 오용 탐지 모델에서는 기존에 알려진 패턴에 대한 처리만이 가능하다는 단점을 가지고 있다. 즉, 알려지지 않은 침입 패턴 방법으로의 시스템 접근은 막을 수가 없다. 따라서 침입자들이 새로운 침입 방식을 개발하여 침입을 시도하게 되면 대응이 상당히 어렵게 된다. 이에 대한 해결책으로 최근에는 비정상행위 탐지 모델[2,3,4]에 대한 연구가 활발히 진행되고 있다.

* 연세대학교 컴퓨터과학과({osh, leewo}@amadeus.yonsei.ac.kr)

비정상행위 탐지 모델은 사용자의 시스템 이용 또는 행동 패턴의 변화를 통해서 침입을 탐지하는 방법으로 정상 행위 모델을 벗어나는 경우를 침입으로 간주하게 된다.

침입자들의 시스템 공격은 초기에는 침투 기법이 단순하였지만 정보 통신의 발전과 더불어 시스템 침투 기법도 고도화되고 전문적으로 변화해가고 있다. 따라서 이에 대응하는 침투 방지 기법들도 그 복잡성을 더해 가고 있으므로 과거와 같이 개별적이며 근대적인 수 작업 관리 방식으로는 충분한 보안 유지를 기대할 수 없다. 이러한 문제를 해결하기 위해서 자동화된 판정 시스템 개발이 필요하게 되었고 방대한 양의 감사 자료(audit data)를 필터링 등의 방법으로 자료의 저장 및 분석에 따른 오버헤드를 최소화시킬 필요가 있게 되었다. 특히 비정상행위 탐지 모델의 핵심이라 할 수 있는 비정상 행위 판정 기술과 관련하여 보안 관련 감사 자료의 수집, 저장, 분석 및 해석 기술에 대한 연구가 추진 중이다[3~7]. 최근에는 방대한 데이터 분석을 좀 더 지능적이고 자동적으로 수행하기 위해서 데이터 마이닝 기법을 이용하여 사용자의 정상행위를 모델링하고 있다[4,8,9]. 이러한 연구들 중에서 JAM[8,9] 시스템이 가장 주목할 만한 성능을 보여주고 있다. 하지만 JAM에서는 네트워크 작업단위를 네트워크 연결이 발생됐을 때부터 연결이 끊어질 때까지로 하고 있다. 따라서 연결중에 발생되는 다양한 작업에 대해서는 모델링할 수 없는 단점을 가지고 있다.

본 논문에서는 네트워크 패킷 레벨에서 비정상 행위 탐지를 수행하는 알고리즘을 소개한다. 이를 위해서 수집된 네트워크 로그를 연관 규칙 생성 알고리즘을 이용하여 네트워크 정상행위 패턴을 생성한다. 이때, 기존의 연관 규칙을 그대로 이용하였을 때 네트워크 행위에 대해서 정확하게 모델링할 수 없는 단점을 가지고 있다. 즉, 네트워크 패킷에 포함되어 있는 속성 항목들간의 연관관계가 패킷의 종류에 따라 확연히 구분되기 때문에 다양한 네트워크 행위에 대해서 정상행위 패턴이 충분히 파악되지 못하는 단점을 갖는다. 하나의 네트워크 행위는 서로 연관된 여러 개의 패킷들로 발생되므로 패킷간의 연관관계를 파악하여 반영함으로써 이러한 문제점을 해결할 수 있다. 결과적으로 네트워크 행위를 정확하게 모델링하기 위해서는 패킷 내 뿐만 아니라 패킷간의 연관성을 탐사하는 알고리즘이 필요하다. 한편, 제안된 알고리즘에 의해서 생성된 네트워크 행위 프로파일에 대해서 새로운 행동에 대한 비정상행위를 탐지하는 새로운 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 침입 탐지 모델의 기본적인 형태를 분류하고 다양한 침입 탐지 시스템을 소개한다. 3장에서는 데이터 마이닝 기법을 이용하여 네트워크의 정상행위 패턴 생성을 위한 방법을 제안하며 4장에서는 3장에서 제시한 알고리즘을 활용한 비정상행위 탐지 방법을 소개한다. 5장에서는 비정상행위 판정 시스템의 성능향상을 위한 모의 실험 결과를 비교하고, 6장에서 최종적인 결론 맺는다.

II. 관련 연구

오용 탐지 모델은 전문가 시스템[5], 상태 전이 분석[10,11], 모델 기반 기법[12] 등에 이용된다. 전문가 시스템은 지식 기반의 침입 탐지 방법으로써 공격 패턴을 규칙(if-then-rule)형태로 표현하고 감사 추적 이벤트를 사실로 나타내며 일치하는 공격 패턴이 존재하면 규칙에 따라서 수행한다. 상태 전이 분석 모델에서는 침입자의 공격 패턴 상태 전이를 통해서 표현된다. 상태 전이 다이어그램은 상태 전이 분석 그래프를 표현하는 방법으로써 침입의 요구 및 이에 대한 결과를 표현하고 침입을 수행한 경로를 알 수 있다. 상태 전이 다이어그램은 침입이전의 상태를 시작 상태로 표현하고 여러 가지 중간 상태를 거쳐서 침입이 성공되었다면 최종 상태에 도달하게 된다. 대표적인 시스템으로는 STAT[10]와 USTAT[11]를 들 수 있다.

모델 기반 기법은 사용자 행동이 시나리오 형태로 표현되고 이 행동은 지식 기반의 침입 시나리오와의 일치 여부를 찾아서 침입을 탐지한다. 기본적으로 모델 기반 기법은 예측자, 계획자 그리고 해석기 모듈로 이루어져 있다. 예측자는 다음 단계에서 나오게 될 시나리오 모델을 예상하는 역할을 하고, 계획자는 이 가설을 감사 레코드에서 나타낼 수 있는 형식으로 변형하며, 해석기는 감사 데이터에서 이 모델이 존재하는지의 여부를 조사한다.

비정상행위 탐지 모델은 사용자의 시스템 이용 또는 행동 패턴의 변화를 통해서 침입을 탐지하는 방법으로 정상 행위 모델을 벗어나는 경우를 침입으로 간주하게 된다. 대표적인 분석 방법으로는 통계적인 방법[3,6,13], 예측 가능한 패턴 생성(Predictive Pattern Generation)[4]등이 있다. 통계적인 방법은 비정상

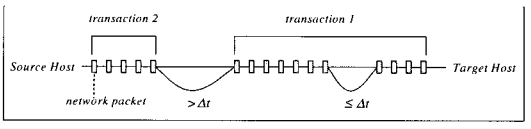
행위 탐지 기법 중에서 가장 많이 사용되는 방법으로 과거의 경험에 대한 자료를 통계적인 값으로 유지하고 있으며 이를 바탕으로 사용자의 비정상행위를 판단하게 된다. 이 방법으로 개발된 대표적인 시스템으로는 SRI에서 개발한 EMERALD[3], NIDES[6] 및 IDES[13], 등이 있다. 예측 가능한 패턴 생성 모델에서 사용자의 행위는 순서적으로 발생한다는 가설에 근거한 것으로 시간 기반의 규칙을 이용하여 사용자의 각 행위에 시간 요소를 부여해서 발생된 행위들이 순서적으로 올바른지 또는 각 행위들 사이의 시간적인 간격이 올바른지를 조사하여 사용자 행위의 정상 또는 비정상 여부를 결정한다.

최근에는 분산 환경에서 보다 효과적인 침입 탐지를 수행할 수 있도록 에이전트 기법을 이용하고 있으며 여기에는 JAM[8,9]과 AAFID[7]등이 있다. JAM (Java Agent for Meta-Learning)은 데이타마이닝 응용 프로그램을 평가하는 데에 있어 일반적 접근 방법인 메타 학습(Meta-Learning)을 채용하고 있으며 분산환경에서의 이식성과 확장성을 제공하는 에이전트 기반 데이타마이닝 시스템이다. 즉, 분산된 여러 사이트에서 데이타마이닝 결과를 상위 사이트에서 조합(메타 학습)하여 사용자의 부정행위를 탐지한다. AAFID 시스템은 기존의 IDS에 대하여, 계층적이고 분산된 에이전트의 구조를 가짐으로써 하나의 에이전트가 서비스를 중지해도 다른 에이전트들이 수행을 계속할 수 있도록 하며 각 에이전트들이 독립적으로 수행되므로 전체의 시스템을 다시 시작해야 하는 번거로움을 해결한다. 또, 각 계층에 있는 에이전트들은 수집한 정보를 간단하게 정리하여 상위 계층으로 전달하므로 침입자가 잘못된 데이타를 발생하려는 시도를 할 때 쉽게 감지될 수 있다.

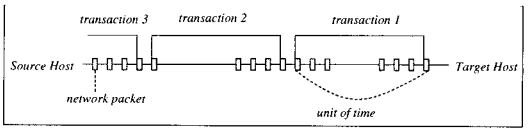
Ⅲ. 네트워크 정상행위 패턴 모델링

대용량의 사건들이 기록되어 있는 데이터베이스에서 자주 발생되는 데이터 항목간의 상호 연관성 탐색 기법을 연관 패턴 탐사[14,15]라고 한다. 연관 패턴을 이용하여 네트워크의 정상적인 행위 패턴을 추출함으로써 새로운 행동에 대한 비정상행위도를 파악할 수 있다. 본 논문에서는 패킷단위의 로그 분석 방법을 이용하여 정상행위 패턴을 추출한다. 기존의 연관 패턴 탐사 알고리즘을 이용할 때 트랜잭션의 단위는 하나의 패킷이 된다. 하지만, 트랜잭션 단위를 단순히 하나의 패킷으로 설정하게 되면, 네트워크 패킷에 포함되어 있는 속성 항목들간의 연관관계가 패킷의 종류에 따라 확연히 구분되기 때문에 다양한 네트워크 행위에 대해서 충분한 정상행위 패턴 생성이 불가능하다. 하나의 네트워크 행위는 서로 연관된 여러 개의 패킷들로 발생되므로 패킷간의 연관관계를 파악하여 반영함으로써 이러한 문제점을 해결할 수 있다. 이것은 네트워크 행위는 서로 연관된 여러 개의 패킷들 단위로 발생된다는데 전제를 두고 있다. 패킷간 연관 관계를 고려한 트랜잭션은 특정 호스트로부터의 연속적인 네트워크 행위로 정의할 수 있다. [그림 1]은 소스 호스트와 대상 호스트간의 트랜잭션 구분 단위를 나타낸다.

[그림 1(a)]에서는 패킷간의 시간차가 주어진 임계치(threshold) Δt를 넘었을 때 현재까지의 패킷들을 하나의 트랜잭션으로 생성한다. 여기에서 Δt의 값은 트랜잭션 생성에 상당히 많은 영향을 주게 된다. 만일 Δt를 크게 하면 서로 관련 없는 패킷들을 하나의 트랜잭션으로 생성하기 때문에 부정확한 정상행위 패턴을 생성하게 된다. 한편 Δt를 작게 하면 연관 관계가 큰 패킷들을 강제로 분리되기 때문에 역시 정확한 정상행위 패턴생성이 어렵게 된다. [그림 1(b)]에서는 고정길이의 단위 시간으로 소스 호스트로부터 대상 호스트까지 전송된 패킷들의 트랜잭션을 구분하였다. 이 그림에서 단위 시간을 크게 하면 서로 관련이 없는 패킷들을 하나의 트랜잭션으로 생성될 수 있으며 단위 시간을 작게 하면 가변길이에서와 마찬가지로 연관관계가 큰 패킷들을 다른 트랜잭션으로 분리하기 때문에 정확한 정상행위 패턴 생성에 어려움을 갖게 된다. 따라서 Δt 값과 단위 시간을 설정하는 하는 것은 상당히 중요하며, 이를 위해서 실험을 통해 최적의 Δt값과 단위 시간을 설정하도록 하였다. 또한 가변길이 트랜



(a) 가변길이 트랜잭션



(b) 고정길이 트랜잭션

[그림 1] 트랜잭션 구분 단위

잭션과 고정길이 트랜잭션으로부터 생성된 정상행위 패턴의 정확도를 비교하도록 하였다.

정상행위 패턴을 생성하기 위한 네트워크 로그 데이터를 D라 하고 D의 원소인 트랜잭션 T에 포함된 패킷의 개수가 m일 때 트랜잭션 T는 $T=\{p_1, p_2, \ldots, p_i, \ldots, p_m\}$와 같이 표현된다. 이때 패킷 p_i에 포함된 속성 항목의 개수가 n일 때 $p_i = \{a^i_1, a^i_2, \ldots, a^i_n\}$와 같이 표현된다.

〔정의 1〕

트랜잭션에 T에 대한 패킷 p의 포함관계 함수 $IN(T, p)$

$q \in T$ 인 임의의 패킷 q 대해서 $p \subseteq q$ 이면, 패킷 p가 트랜잭션 T에 포함되었다고 할 수 있으며 IN(T, p)=1이다. 그렇지 않으면 IN(T, p)=0과 같다.

〔정의 2〕

트랜잭션 T에 대한 패킷집합 $P=\{p_1, p_2, \ldots, p_k\}$의 포함관계 함수 $SUB(T, P)$

패킷집합 P와 트랜잭션 T가 1:1 대응관계에 있고, 임의의 패킷 $p_i \in P$에 대해서 IN(T, p_i)=1이면, 패킷집합 P는 트랜잭션 T에 포함되었다고 할 수 있으며 SUB(T, P)=1과 같다. 그렇지 않으면 SUB(T, P)=0이다. 이때, 패킷집합 P의 임의의 원소 p_i, p_j와 트랜잭션 T의 원소 p_s, p_t에 대해서 $p_i \subseteq p_s$, $p_j \subseteq p_t$일 때, $p_s \neq p_t$ 임을 말한다.

정의 1에서는 하나의 패킷이 어떤 트랜잭션에 포함되는지를 판별한다. 이때 하나의 패킷을 이루는 전체 항목 집합에 대한 포함관계 뿐만 아니라 패킷의 부분 항목 집합이 트랜잭션에 포함되는지를 판별할 수 있다. 정의 2에서는 패킷집합이 트랜잭션에 포함되는지를 판별하며 정의 1에서와 마찬가지로 패킷내 부분적인 항목 집합의 포함관계도 고려할 수 있다.

기존의 연관 규칙 생성 알고리즘 〔14, 15〕에서와 마찬가지로 본 논문에서는 네트워크 로그 데이터로부터 유용한 규칙을 생성하기 위해서 최소 지지율(*support*)을 이용할 수 있다. 패킷 내 연관성을 규칙에 적용시키기 위해서 패킷 내에 존재하는 속성 항목 집합의 지지율이 최소지지율이 넘는지를 판별해야 한다. 여기에서 최소 지지율이 넘는 항목 집합을 빈발 항목 집합이라 정의한다. 한편 패킷간 연관성을 규칙에 적용시키기 위해서 패킷 집합의 지지율이 최소지지율을 넘는지를 판별해야 하며 최소지지율 이상인 패킷 집합을 빈발 패킷 집합이라 정의한다.

패킷 p와 패킷 집합 P의 지지율은 식 (1)과 식 (2)와 같이 계산된다.

$$sup(p) = \frac{1}{|D|} \sum_{T \in D} IN(T, p) \quad (1)$$

$$sup(P) = \frac{1}{|D|} \sum_{T \in D} SUB(T, P) \quad (2)$$

$|D|$: *네트워크 로그 데이터에서 전체 트랜잭션의 개수*

패킷 내 및 패킷간의 연관 패턴을 탐사 알고리즘은 다음과 같이 크게 네 가지 과정으로 구성된다.

〔단계 1〕 기존의 Apriori〔15〕 알고리즘을 이용하여 패킷내의 연관성을 탐사한다.
〔단계 2〕 탐사된 연관 패턴의 ID를 로그에 매핑시켜서 새로운 로그를 생성한다.
〔단계 3〕 새로운 로그에 대해서 패킷간 연관성 탐사 알고리즘을 이용하여 네트워크 로그에 대한 정상행위 패턴을 생성한다.
〔단계 4〕 생성된 패턴간의 포함관계가 존재하지 않는 최대 연관 집합을 구한다.

첫 번째 단계에서는 Apriori 알고리즘을 이용하여 패킷내에 존재하는 속성 항목들간에 연관성을 탐사한다. 먼저 네트워크 로그로부터 유일한 속성 항목 집합을 탐색한다. 탐색된 항목 집합은 오름차순으로 정렬하고 각 항목에 대해서 빈발 1-항목집합을 구한다. 빈발 1-항목집합으로부터 항목의 길이가 2인 후보 2-항목집합을 생성하게 된다. 이때 네트워크 로그에는 패킷의 정보를 표현하는 필드들로 이루어진다. 따라서 후보 2-항목 집합을 생성하는 과정에서 같은 필드에 포함되는 항목간의 조합을 하지 말아야 한다. 이와 같이 생성된 후보 2-항목 집합에 대해서 최소지지율 이상이면 빈발 2-항목집합으로 생성하게 된다. 이러한 과정은 새로운 빈발 항목집합이 탐사되지 않을 때가지 반복된다. 예를 들어, 그림 2의 로그 데이터에서 최소 지지율을 2로 설정했을 때 빈발 1-항목집합을 구하면 다음과 같다.

그림 2의 로그 데이터로부터 후보 1-항목 집합들을 구하면 *{telnet}, {http}, {ftp}, {smtp}, {A}, {B},*

*{C}, {E}, {F}, {G}, {H}, {F0}, {F1}*과 같다. 항목 집합 *{telnet}*에 대해서 *(telnet)* $\in_p T_1, T_4$이므로 최소지지율 이상이 되어서 빈발 1-항목 집합이 될 수 있다. 이와 같은 방법으로 모든 아이템 집합에 대해서 빈발 1-항목 집합을 구하면 다음과 같다.

빈발 1-항목 집합 => {telnet}, {ftp}, {smtp}, {A}, {C}, {E}, {F}, {F0}, {F1}

Transaction ID	service	src host	dst host	flag
T1	telnet http telnet	A A A	E F E	F0 F0 F0
T2	ftp smtp	B C	G H	F0 F0
T3	smtp ftp	C C	E F	F0 F1
T4	telnet telnet	A A	F E	F1 F0

〔그림 2〕 로그 데이터

빈발 1-항목 집합을 이용하여 길이가 2이상인 빈발 항목 집합을 생성하는 과정은 〔그림 3〕과 같다.

Large 2-itemset	support
{telnet, A}	2
{telnet, E}	2
{telnet, F0}	2
{smtp, C}	2
{smtp, F0}	2
{A, E}	2
{A, F}	2
{A, F0}	2
{C, F0}	2
{E, F0}	3
{F, F1}	2

Candidate 3-itemset	support
{telnet, A, E}	2
{telnet, A, F0}	2
{telnet, E, F0}	2
{smtp, C, F0}	2
{A, E, F0}	2

Candidate 4-itemset	support
{telnet, A, E, F0}	2

〔그림 3〕 길이 2 이상인 빈발 항목 집합 생성과정

두 번째 단계에서는 첫 번째 단계에서 생성된 빈발 항목집합을 로그에 적용하여 패킷간 연관성을 찾기 위한 새로운 로그를 생성하게 된다. 이때 생성된 빈발 항목집합에 각각 유일한 식별자를 부여하게 된다. 이것은 세 번째 단계에서 빈발 패킷 집합을 구하기 위한 항목으로 이용된다. 식별자는 길이가 가장 긴 빈발 항목 집합으로부터 가장 작은 빈발 항목 집합으로 부여한다. 이 과정에서 빈발 n-항목 집합에 포함되는 빈발 (n-1)-항목 집합이 있을 때 두 항목 집합의 지지율과 패킷의 개수가 같으면 빈발

빈발 항목집합	식별자	지지율	패킷개수
{telnet, A, E, F0}	1	2	3
{smtp, C, F0}	2	2	2
{telnet, A}	3	2	4
{A, F}	4	2	2
{A, F0}	5	2	4
{E, F0}	6	3	4
{F, F1}	7	2	2
{ftp}	8	2	2
{F}	9	3	3
{F0}	10	3	7

(a) 식별자가 부여된 빈발 항목 집합

Transaction ID	식별자 리스트
T1	{1, 3, 5, 6, 10} {4, 5, 9, 10} {1, 3, 5, 6, 10}
T2	{8, 10} {2, 10}
T3	{2, 6, 10} {7, 8}
T4	{3, 4, 7, 9} {1, 3, 5, 6, 9, 10}

(b) 변환된 로그(LOG)

〔그림 4〕 빈발 항목 집합의 식별자와 식별자가 적용된 네트워크 로그

(n-1)-항목집합을 삭제한다. 예를 들어서 〔그림 3〕에서 *{telnet, A, E, F0}*는 *{telnet, A, E}, {telnet, A, F0}, {telnet, E, F0}*를 포함하고 지지율과 패킷의 개수가 같다. 따라서 이들 빈발 3-항목집합은 삭제된다. 〔그림 4(a)〕는 식별자가 부여된 빈발 항목 집합을 나타내고 〔그림 4(b)〕는 식별자가 적용된 새로운 로그를 나타낸다.

세 번째 단계에서는 두 번째 단계에서 생성된 로그를 이용하여 빈발 패킷 집합을 구하게 된다. 앞에서도 설명했지만, 빈발 패킷 집합은 서로 연관된 패킷 집합을 의미한다. 빈발 패킷 집합을 구하기 위해서, 길이가 1인 빈발 패킷 집합을 생성해야 하며 이것은 이미 두 번째 단계에서 구한 것이다. 즉, 식별자가 부여된 빈발 항목 집합이 빈발 1-패턴집합이 된다.

또한, 빈발 1-패킷 집합을 이용하여 후보 2-패킷 집합을 구할 수 있다. 〔그림 4(a)〕의 데이터를 이용하여 후보 2-패킷 집합을 구하면 다음과 같다.

후보 2-패킷 집합={{1,1},{1,2},{1,3},..., {10,10}}

```
generate candidate n packet-sets /*SQL like code*/
begin
  insert into candidate n packet-sets
    select p.id1, ... p.idn-1, q.idn-1
    from large n-1 packet-sets p, q
    where p.id1 = q.id1 and ... and p.idn-1 ≤ q.idn-1
end

generate large n packet-sets
begin
    for each c ∈ candidate n packet-sets
    begin
          sup = 0;
          while get t ∈ LOG do begin
          if c ⊂p t then sup++;
    end
    if (sup ≥ min_support) then
          add c to large n packet-sets
    end
end
```

〔그림 5〕 빈발 n-패킷집합 생성 알고리즘

후보 2-패킷 집합에서 패킷 집합 *{1, 1}* $\subset_p T_1$이므로 최소지지율보다 작다 따라서 후보 2-패킷 집합에서 삭제된다. 한편 *{1, 3}* $\subset_p T_1$, T_4이므로 빈발 2-패킷집합이 된다. 이러한 과정으로 길이가 n인 후보 패킷 집합과 빈발 패킷 집합을 구하는 알고리즘은 〔그림 5〕와 같다. 알고리즘에 의해서 생성된 전체 빈발 패킷집합은 〔그림 6〕과 같다.

네 번째 단계에서는 생성된 빈발 패킷집합대해서 서로간에 포함관계가 없는 최대 빈발 패킷집합을 생성하게 된다. 최대 빈발 패킷 집합을 생성하는 목적은 마이닝 과정에서 생성된 중복된 패턴들을 제거하기 위함이다. 최대 빈발 패킷 집합(MLP:Maximal Large Packet-set)을 구하기 위해서, 먼저 길이가 가장 긴 빈발 패킷 집합을 최대 빈발 패킷 집합에 입력한다. 이때 MLP에는 세 번째 단계에서 구한 ID 리스트로 입력되는 것이 아니라 원래 패킷 정보로 환원하여 입력된다.

예를 들어, 〔그림 6〕에서 길이가 가장 긴 패턴집합의 크기는 2이지만 이들 중에서 크기가 가장 작은 ID로 이루어진 패턴 집합이 가장 긴 패턴집합이 된다. 따라서, (1, 3)이 가장 먼저 MLP에 입력된다. 나머지 빈발 패킷 집합에 대해서 다음 두 가지 조건을 만족하면 삭제되고 그렇지 않으면 최대 빈발 패킷 집합에 추가한다.

c_1. For X $\subset p$ *P where P* $\in$ *MLP.*
c_2. sup(X) equals to sup(P)

빈발 패킷 집합	지지율	빈발 패킷 집합	지지율	빈발 패킷 집합	지지율
{1}	2	{1, 3}	2	{4, 5}	2
{2}	2	{1, 4}	2	{4, 6}	2
{3}	2	{1, 9}	2	{4, 10}	2
{4}	2	{3, 3}	2	{5, 9}	2
{5}	2	{3, 4}	2	{6, 7}	2
{6}	3	{3, 5}	2	{6, 9}	2
{7}	2	{3, 6}	2	{6, 10}	2
{8}	2	{3, 9}	2	{7, 10}	2
{9}	3	{3, 10}	2	{9, 10}	2
{10}	3			{10, 10}	2

(a) 빈발 패킷 집합

빈발 패킷 집합	빈발 패킷 집합	지지율
{1, 3}	{(telnet, A, E, F0), (telnet, A)}	2
{1, 4}	{(telnet, A, E, F0), (A, F)}	2
{2, 8}	{(smtp, C, F0), (ftp)}	2
{6, 7}	{(E, F0), (F, F1)}	2
{10, 10}	{(E0), (E0)}	2
{6}	{(E, F0)}	3
{9}	{(F)}	3
{10}	{(E0)}	3

(b) 최대 빈발 패킷 집합

〔그림 6〕 빈발 패킷 집합과 최대 빈발 패킷집합

〔그림 6〕의 빈발 패킷집합을 이용하여 MLP를 구하면 〔그림 6(b)〕와 같다.

Ⅳ. 비정상행위 탐지 모델링

온라인 상에서 발생되는 네트워크 행위에 대한 비정상행위를 탐지하기 위해서는 온라인상에서 발생되는 데이터를 트랜잭션으로 가공하여 정상행위 프로파일과 비교해야 한다. 이때 네트워크 패킷에 대한 감시 과정에서 트랜잭션의 끝을 나타내는 조건이 발생됐을 때 비정상행위를 판정하게 된다. 연관 규칙을 이용한 정상행위 프로파일은 다음과 같이 MLP와 속성-지지도 집합(ESS:Element Support-Set)으로 구성된다.

$MLP = \{(R_1, support(R_1)), ..., (R_n, support(R_n))\}$
$R_i = \{(a^i_1, support(a^i_1)), (a^i_2, support(a^i_2)), ..., (a^i_j, support(a^i_1))\}$

$$ESS = \{(e_1, support(e_1)), (e_2, support(e_2)), \ldots, (e_m, support(e_m))\}$$

MLP는 연관 규칙 탐사 알고리즘에서 생성된 규칙의 집합을 의미한다. MLP는 여러 개의 규칙과 이에 대한 지지율을 포함한다. MLP의 구성요소인 규칙은 여러 개의 패턴 내 연관 규칙과 이에 대한 지지율을 포함하고 있다. ESS는 연관 규칙 생성단계 중 첫 번째 단계에서 크기가 1인 항목 집합의 원소와 지지율을 포함한다. 여기에서 ESS에 포함되는 원소는 빈발 항목뿐만 아니라 저빈도의 항목도 함께 포함된다.

MLP와 ESS를 이용하여 온라인 트랜잭션 $T = \{p_1, p_2, \ldots, p_l\}$에 대한 비정상 행위를 구하는 과정은 다음과 같다. 먼저 정상행위 프로파일을 이용한 최대 정상행위도(MNA)는 다음과 같이 계산된다.

$$MNA = \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} support(R_i) + \beta \cdot \frac{1}{n \cdot l} \sum_{i=1}^{n} \sum_{j=1}^{l} support(a_j^i) \cdot |a_j^i|$$
$$where\ R_i \in MLP,\ a_j^i \in R_i$$
$$and\ \alpha + \beta = 1.$$

이 식에서는 규칙에 대한 지지율 평균과 규칙의 구성요소인 항목의 평균을 구하여 이 두 수치에 대한 가중치 평균을 구하였다. 이와 같이 계산된 MNA는 온라인 트랜잭션에 대한 판정률을 정규화하기 위한 값이다. 한편, 온라인 트랜잭션 T가 주어졌을 때, 정상행위 패턴과의 매치된 정도(RNA)는 다음과 같이 계산된다.

$$RNA = \frac{1}{N_c} \sum_{i=1, R_i \subseteq T}^{n} support(R_i)$$

$$where\ R_i \subseteq T\ and\ R_i \subseteq MLP$$

$$N_c = \sum_{i=1}^{n} I(R_i, T)$$

$$I(R_i, T):\ if\ R_i \subseteq T\ then\ 1,\ otherwise\ 0$$

이 수식은 트랜잭션 T가 포함하는 정상행위 패턴들의 평균을 나타낸다. 또한 트랜잭션 T의 원소가 연관 규칙의 각 원소와 매치된 정도(ANA)를 다음과 같이 계산될 수 있다.

$$ANA = \frac{1}{n \cdot l \cdot t} \sum_{i=1}^{n} \sum_{j=1}^{l} \sum_{k=1}^{t} |p_k \cap a_j^i| \cdot asup$$

$$asup = \begin{cases} support(a_j^i)\ if\ a_j^i \subseteq p_k \\ support(e)\ if\ \exists e \in ESS\ and\ e \in p_k \\ 0\ otherwise \end{cases}$$

이 수식에서 트랜잭션 T에 포함되는 원소 p_k와 하나의 규칙에 포함되는 원소 a_j^i 사이의 매치율은 $|P_k \cap a_j^i| \cdot asup$와 같이 계산된다. 여기에서, 만일 a_j^i가 p_k에 포함되면 $asup = support(a_j^i)$이다. 만일 a_j^i가 p_k에 포함되지 않고 ESS의 어떤 원소 e가 p_k에 포함되면 $asup = support(e)$의 값을 취한다. 위의 두 가지 경우가 아닌 경우의 $asup = 0$이 된다.

위의 식에서 온라인에서의 네트워크 행위 트랜잭션(T)가 정상행위 패턴(R)과 유사하다면 값이 커지게 됨으로써 비정상행위도가 낮게 나타나게 된다. 반면, 네트워크로부터 이상 행위가 발생됐을 때 정상행위 패턴과 유사하지 않기 때문에 비정상행위도가 높게 나타나게 된다.

최종적으로 온라인 트랜잭션 T에 대한 비정상행위도는 다음과 같이 계산될 수 있다. 이 수식에서도 MNA에서처럼 가중치를 RNA과 ANA사이에서 가중치 합을 하였고, 이 수치를 정규화하기 위해서 MNA으로 나누었다.

$$Abnormality = 1 - \frac{\alpha \cdot RNA + \beta \cdot ANA}{MNA}$$

$$where\ \alpha + \beta = 1.$$

V. 실험 및 성능 평가

본 장에서는 판정시스템의 성능에 대한 검증을 위하여 수행한 모의 실험 환경 및 모의 실험 시나리오와 동작에 관하여 기술한다. 또한 임계치를 최적화하기 위한 특성을 파악할 수 있는 대표적인 모의 실험 결과를 도표로 나타내고 이를 분석한다. 모의 실험을 위해서 UNIX 기반의 Solaris 2.6에서 tcpdump를 이용하여 두 달 동안의 정상행위 로그 데이터를 수집하였다. 전체 로그의 크기는 약 4Gbyte정도이며 지역 탐지 영역은 크게 두 지역으로 나누었다. 실험에서 사용되는 파라미터는 〔표 1〕과 같다.

〔표 1〕에서 정리한 바와 같이 판정 시스템에서 판정율 향상을 위해서 적용해야 할 변수들은 다양하다. 따라서 각각의 변수를 적용하여 네트워크의 정상행위 패턴을 생성하고 최대 판정율을 가진 변수

〔표 1〕 실험에 사용하게 될 파라미터

파라미터	적용값
최소 지지율	10, 20, 30, 40, 50, 60, 70, 80
클러스터링 범위	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024
날짜	1, 3, 7, 14, 21, 30, 37, 44, 51
트랜잭션 구분 기준	가변 트랜잭션, 고정 트랜잭션
분석 방법	클러스터링, 연관 규칙, 순차패턴
분석 범위	지역 탐지 서버, 전역 탐지 서버

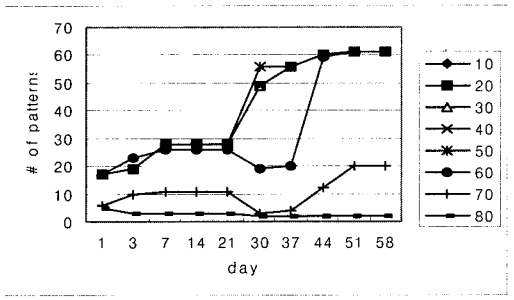
값을 찾고자 한다. 이를 위해서 본 논문은 크게 세 가지 측면으로 실험을 하였다. 이 실험에서는 프로파일 크기에 적정 임계치 범위를 파악하도록 한다. 네트워크 행위에 대한 비정상행위를 판정하기 위해선 정상행위 패턴을 생성하여 프로파일로 유지하게 된다. 이때 많은 양의 네트워크 행위가 발생되는 시스템의 경우는 용량이 큰 프로파일이 생성될 수 있기 때문에 실시간 판정 성능에 저하를 가져 올 수 있다. 따라서 때문에 최적화 된 프로파일을 생성하는 것이 중요하다.

〔그림 7〕에서 〔그림 9〕까지는 연관 마이닝을 통해서 생성된 프로파일의 최적화를 위한 실험을 하였다. 〔그림 7〕은 생성된 연관 규칙의 개수를 나타낸다. 여기에서 가변길이 트랜잭션인 경우에는 지지율이 50%이하이고 분석 날짜가 44일 되는 위치에서 연관 규칙의 개수가 수렴함을 볼 수 있다. 반면 고정길이 트랜잭션에서는 수렴되는 정도가 최소지지율에 따라 큰 차이를 나타내고 있다. 이것은 의미 단위로 나누어진 가변길이 트랜잭션이 고정길이 트랜잭션보다 정확한 모델링이 가능하다는 것을 보여주고 있다.

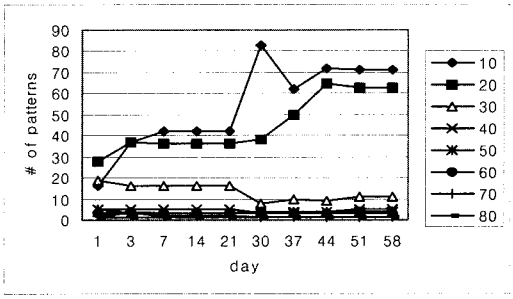
〔그림 8〕은 연관 규칙의 평균 크기를 나타내고 있다. 연관 규칙의 평균 크기는 프로파일에 존재하는 패턴이 비정상행위 탐지를 위해서 많은 정보를 포함하고 있는지의 여부를 판별한다. 가변길이 트랜잭션에서는 지지율이 50%이하인 경우에 수렴하는 것을 볼 수 있지만 고정길이 트랜잭션에서는 최소지지율에 따라서 크게 변하는 것을 볼 수 있다.

마찬가지로 〔그림 9〕에서는 연관 규칙 마이닝에 의해서 생성된 프로파일의 크기를 나타내며 위 두 실험에서와 마찬가지로 최소지지율이 50%, 분석 날짜가 44일 되는 지점에서 수렴함을 볼 수 있다.

결국, 본 논문에서 수행된 실험 환경에서는 연관 패턴 생성시 분석 날짜가 44일 최소지지율이 50%

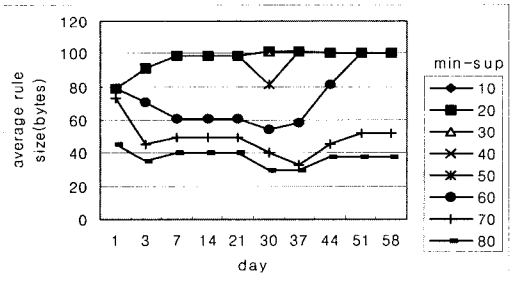


(a) 가변길이 트랜잭션

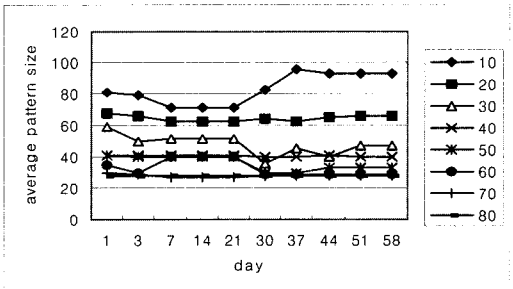


(b) 고정길이 트랜잭션

〔그림 7〕 생성된 패턴의 개수

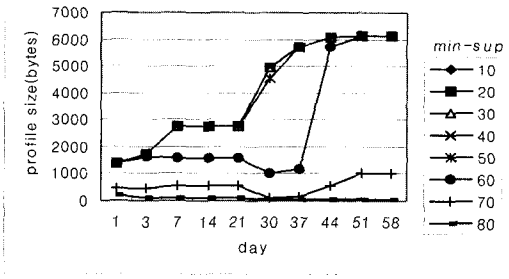


(a) 가변길이 트랜잭션

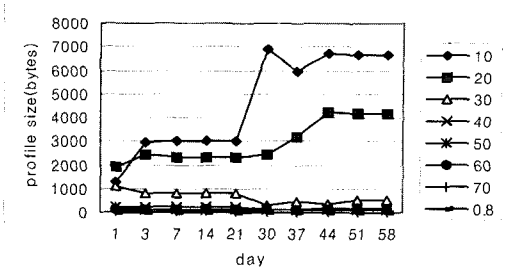


(b) 고정길이 트랜잭션

〔그림 8〕 생성된 패턴의 평균 크기

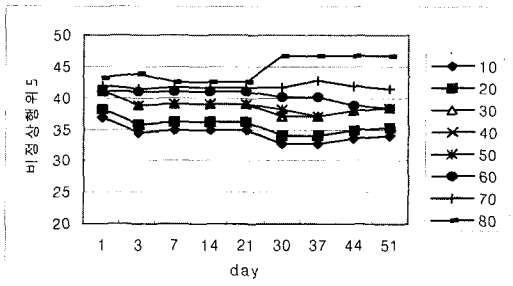


(a) 가변길이 트랜잭션

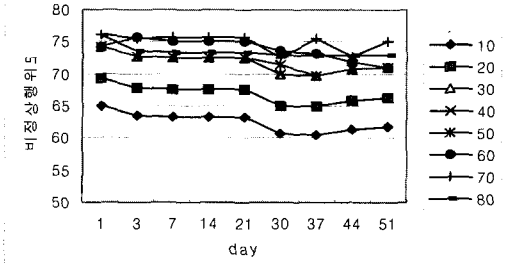


(b) 고정길이 트랜잭션

〔그림 9〕 프로파일 크기



〔그림 10〕 정상행위 데이터를 이용한 비정상 행위도



〔그림 11〕 다른 지역 데이터를 이용한 비정상 행위도

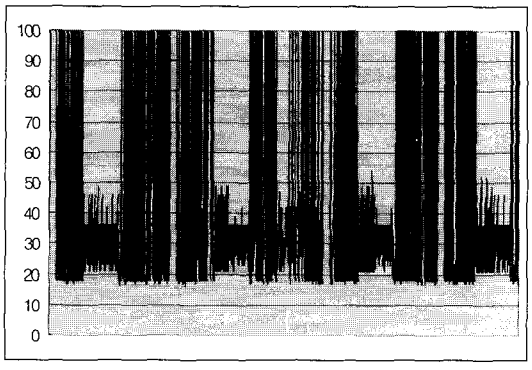
정도 되었을 때 최적의 프로파일을 생성할 수 있었다.

〔그림 10〕에서는 판정율 향상을 위한 적정 임계치 범위를 파악하기 위해서, 주어진 최소지지율과 분석 날짜에 따라 정상행위에 대한 프로파일을 생성하고, 생성된 각 프로파일을 이용하여 정상행위 행위에 대한 비정상행위를 찾는 실험을 하였다. 이 그림에서는 70%이상인 경우 그 이하의 지지율보다 비정상 행위도가 높아짐을 볼 수 있다. 이것은 지지율이 높아지면서 정상 행위 프로파일의 크기가 작아지기 때문에 정확한 판정을 기대할 수 없기 때문이다. 〔그림 11〕에서는 〔그림 10〕에서 생성된 정상행위 프로파일을 이용하여 다른 지역 데이타에 대한 비정상행위도에 대한 실험을 하였다. 이 그림에서 정상행위 데이터와는 달리 비정상행위도가 상당히 높게 나타남을 알 수 있다. 한편, 이 그림에서는 최소지지율이 작아지게 되면 판정율이 떨이지게 된다. 즉, 비정상적인 행위임에도 불구하고 정상행위 데이터의 비정상행위도에 가까워지게 됨으로써 정확한 판정을 기대할 수 없게 된다. 따라서 최소지지율을 30%이상으로 설정하였을 때 최적의 판정율을 기대할 수 있다.

본 논문에서는 앞의 실험 결과를 토대로 비정상행위 판정 모델을 위한 최적화 된 임계치 값은 〔표 2〕와 같이 나타났다.

〔표 2〕 실험에 사용한 최적 설정 범위

실험에 사용하는 변수	설정 범위
분석 날짜	44일
최소 지지율	30 - 60%
클러스터링 범위	8 - 16



〔그림 12〕 온라인 판정

〔표 2〕를 이용하여 온라인에서 판정을 수행한 실험이 〔그림 12〕이다.

실험에서 사용된 공격 방법의 종류는 UDP bomb, ICMP flooding, Syn flooding, Port scan등 4가지이며 이들을 여러 차례 반복해서 로그를 생성하

였다. 실험에서 사용된 파라미터로는 최소지지율이 50%, 클러스터링 범위를 10으로 설정하여 정상행위 패턴을 생성하였다. 이 실험에서 UDP bomb, ICMP flooding, Port scan 과 같은 공격 방법에 대해서는 상당한 효과를 보았으나 Syn flooding과 같은 공격방법은 탐지하지 못하였다. 이러한 원인으로 Syn flooding에서는 어떤 호스트의 특정 포트에 자주 사용하는 서비스에 대한 거부 공격방법이다. 따라서 정상행위 패턴에는 이러한 공격 패턴을 포함하는 규칙을 가지게 되기 때문이다.

Ⅵ. 결 론

본 논문에서는 방대한 네트워크 로그 데이터 분석을 좀 더 지능적이고 자동적으로 수행하기 위해서 지식 탐사 분야에서 활용되고 있는 데이터 마이닝 기법을 활용하였다. 본 논문에서는 네트워크 패킷 레벨에서 비정상 행위 탐지를 수행하는 연관 패턴 탐사 알고리즘을 소개하였다. 이때, 기존의 연관 패턴 탐사 알고리즘을 이용하였을 경우에는 네트워크 패킷에 포함되어 있는 속성 항목들간의 연관관계가 패킷의 종류에 따라 확연히 구분되기 때문에 다양한 네트워크 행위에 대해서 정상행위 패턴 생성이 불가능하다. 이를 해결하기 위해서, 본 논문에서는 패킷 내의 연관관계뿐만 아니라 패킷간의 연관관계를 탐사하였다. 이것은 네트워크 행위가 서로 연관된 여러 개의 패킷들 단위로 발생된다는데 전제를 두고 있다. 결과적으로 효과적인 정상행위패턴을 생성할 수 있었다. 한편, 제안된 알고리즘에 의해서 생성된 네트워크 행위 프로파일에 대해서 새로운 행동에 대한 비정상행위를 탐지하는 방법을 소개하였다. 이와 더불어 다양한 모의 실험을 통해 판정 시스템의 탐지율을 높이고 오판율을 줄이기 위한 최적의 임계치 값에 대한 결과를 보였다.

참 고 문 헌

[1] B. Mukherjee, T. L. Heberlein, and K. N. Kevitt, "Network intrusion Detection," IEEE Network, 8(3):26-41, May/June 1994.

[2] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The Architecture of a Network Level Intrusion Detection System," Technical Report, Computer Science Department, University of New Mexico, August 1990.

[3] P. A. Porras and Peter G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC, October 1997.

[4] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," In Proc. of the 7th USENIX Security Symposium, San Antonio, Texas, January *26-29, 1998*.

[5] S. Kumar, Classification and Detection of Computer Intrusions. Ph.D. Dissertation, August 1995.

[6] H. S. Javitz and Alfonso Valdes, "The NIDES Statistical Component Description and Justification," Annual report, SRI International, 333 Ravenwood Avenue, Menlo Park, CA 94025, March 1994.

[7] J. S. Balesubramaniyan, J. O. Garcia-Fernandes, David Isacoff, Engene Spafford, Diego Zamboni, "An Architecture for Intrusion Detection using Autonomous Agents," Technical Report 98-05, COAST Laboratory, Purdue University, West Lafayette, IN 47907-1398, May 1998.

[8] W. Lee, S. J. Stolfo and P. K. Chan, "Learning Patterns from Unix Process Execution Traces for Intrusion Detection," Proc. AAAI-97 Work. on AI Methods in Fraud and Risk Management, 1997.

[9] S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. Fan, P. K. Chan, "JAM:Java agents for Meta-Learning over Distributed Databases," .Proc. KDD-97 and AAAI97 Work. on AI Methods in Fraud and Risk Management), 1997.

[10] K. Illgun, R. Kemmerer, P. A. Porras, "State Transition Analysis : A rule-based intrusion detection approach," IEEE Transaction on Software Engineering pp. 181~199, March. 1995

[11] K. Illgun, "USTAT: A Real-Time Intrusion

Detection System for UNIX," in Proc. Of the 1993 Symposium Security and Privacy, pp. 16~28, May 24-26, 1993.
[12] T. D. Garvey, T. F. Lunt, "Model based intrusion detection," In Proc. Of the 14th National Computer Security Conference, pp. 372~385, October 1991.
[13] H. S. Javitz, A. Valdes, "The SRI IDES Statistical Anomaly Detector," In Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, May 1991.
[14] R. Agrawal, T. Imielnski, A. Swami, "Mining Association Rules between Sets of Items in Large Database," In Proc. ACM SIGMOD, pp. 207~216, 1993.
[15] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," In Proc. Of the 20th VLDB conference, 1994.

〈著者紹介〉

오 상 현 (Sang-hyun Oh)
1996년 2월 : 제주대학교 정보공학과 졸업
1998년 2월 : 연세대학교 컴퓨터과학과 석사
1998년 3월~현재 : 연세대학교 컴퓨터과학과 박사과정
〈관심분야〉 침입탐지 시스템, 데이터마이닝, 에이전트 시스템

이 원 석 (Won-suk Lee) 정회원
1985년 : 미국 보스턴 대학교 컴퓨터과학과 졸업(학사)
1987년 : 미국 퍼듀 대학교 컴퓨터공학과 졸업(석사)
1990년 : 미국 퍼듀 대학교 컴퓨터공학과 졸업(박사)
1990년~1992년: 삼성전자 선임 연구원
1993년~1999년: 연세대학교 컴퓨터과학과 조교수
1999년~현재: 연세대학교 컴퓨터과학과 부교수
〈관심분야〉 분산 데이터베이스, 멀티미디어 데이터베이스, 객체지향 시스템, 데이터마이닝