사용자 행위 클러스터링을 활용한 비정상 행위 탐지

오 상 현† · 이 원 석††

요 약

컴퓨터를 통한 침입을 효과적으로 탐지하기 위해서 많은 연구들이 오용탐지 기법을 개발하였다. 최근에는 오용 탐지 기법을 개선하기 위해서 비정상행위 탐지 기법에 관련된 연구들이 진행중이다. 이 논문에서는 비정상행위 탐지에서 사용자의 정상행위 패턴을 생성하기 위해 지지율에 기반한 새로운 클러스터링 알고리즘을 제시한다. 제시된 알고리즘에서는 사용자의 과거행위보다 최근행위에 보다 많은 비중을 두는 방법을 적용하였다. 한편, 사용자의 행위를 다양한 각도에서 분석될 수 있도록 사용자의 행위를 여러 판정요소로 분류하고 각 판정요소에 대해서 제시된 알고리즘을 이용하여 사용자의 정상행위 패턴을 생성한다. 결과적으로 사용자의 비정상행위가 효과적으로 탐지될 수 있다.

Anomaly Detection based on Clustering User's Behaviors

Sang-Hyun Oh† · Won-Suk Lee††

ABSTRACT

For detecting various computer intrusions effectively, many researches have developed the misuse based intrusion detection systems. Recently, works related to anomaly detection, which have improved the drawback of misuse detection technique, have been under focus. In this paper, a new clustering algorithm based on *support constraint* for generating user's normal activity patterns in the anomaly detection can proposed. It can grant a user's activity observed recently to more weight than that observed in the past. In order that a user's anomaly can be analyzed in various angles, a user's activity is classified by many measures, and for each of them user's normal patterns can be generated by using the proposed algorithm. As a result, using generated normal patterns, user's anomaly can be detected easily and effectively.

1. 서 론

컴퓨터와 통신 기술의 발달로 사용자에게 다양한 정보와 편리성이 제공된 반면, 컴퓨터 침입 및 범죄로 인한 피해가 날로 증가하고 있다. 따라서 침입자들의 행위를 보다 효과적으로 탐지하기 위한 비정상행위 판정 기술 연구가 필요하다. 여기에서 침입이란 권한이 없는 사용자가 발생시키는 문제 또는 합법한 사용자가 권한을 남용하는 것이라고 정의한다[8]. 이와 더불어 자원의 유용성, 기밀성, 그리고 무결성 등에 저해되는 행동 집합을 침입이라고 정의하기도 한다[9]. 본 논문에서 사용자의 비정상행위는 침입을 포함한 사용자 자신의 업무 권한을 벗어난 행동 일체까지를 포함한다.

침입자들의 시스템 공격은 초기에는 침입 기법이 단순하였지만 정보 통신의 발전과 더불어 시스템 침입 기법도 고도화되고 전문적으로 변화해가고 있다. 따라서 이에 대응하는 침입 탐지 기법들도 그 복잡성을 더해 가고 있으므로 과거와 같이 각 침입 방식에 대한 개별적인 대처 방안으로는 충분한 보안 유지를 기대할 수 없다. 이러한 문제를 해결하기 위해서 자동화된 판정 시스템 개발이 필요하게 되었고 방대한 양의 감사 자료를 필터링 등의 방법으로 자료의 저장 및 분석에

※ 본 논문은 2000년도 한국정보보호센터의 '네트워크 기반 비정상 행위 탐지 시스템 설계 및 프로토 타입 구현' 위탁연구과제로 수행되었음.
† 준 회 원 : 연세대학교 대학원 컴퓨터과학과
†† 정 회 원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 2000년 5월 12일, 심사완료 : 2000년 8월 9일

따른 오버헤드를 최소화시킬 기술이 필요하게 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 추진중이다[1-5]. 최근에는 방대한 데이타 분석을 지능적이고 자동적으로 수행하기 위해서 데이타마이닝 기법을 이용하여 사용자의 정상행위를 모델링하고 있다[5-7].

기존의 몇몇 연구들[2, 3, 13]은 통계적인 기법을 이용하여 호스트 기반 침입 탐지를 수행하였다. 비정상행위 탐지 모델에서 주로 사용하는 통계적인 방법의 경우 실시간 판정에서 사용하게 되는 프로파일 데이타를 최소화 할 수 있다는 장점을 가지고 있는 반면, 감사 데이타를 통계적인 수치 값으로 표현함으로써 데이타의 손실이 발생될 수 있다. 따라서 사용자의 비정상행위가 부정확하게 판정될 가능성을 가지게 된다. 또한, 적은 빈도로 발생되는 사용자 작업에 대해서 효과적으로 관리하지 못하는 단점을 가지고 있다. 즉, 발생빈도가 적은 사용자 행위가 주기적으로 발생되었다면 상당히 의미 있는 정보라 할 수 있지만 기존의 통계적 방법에서는 이러한 정보들을 희소 카테고리(Rare Category)로 관리하였다. 따라서 서로 관련 없는 사용자 행위들을 하나의 단위로 묶어서 관리하기 때문에 효과적인 판정이 이루어지지 않는다.

본 논문에서는 비정상행위 탐지에서 방대한 데이타 분석을 좀 더 지능적이고 자동적으로 수행하기 위해 지지율(support)에 기반한 클러스터링 기법을 활용하였다. 이를 위해서 사용자의 시스템 자원 이용에 관련된 행위를 클러스터링 기법을 이용하여 사용자 정상행위 패턴을 생성하며 이를 바탕으로 사용자의 새로운 행동에 대한 비정상행위를 판별한다. 이때 사용자의 비정상행위를 보다 다양한 각도에서 판별할 수 있도록 사용자의 행위가 여러 가지 판정요소(Measure)들로 분류될 수 있다. 이러한 판정 요소들에 대해서 클러스터링을 수행하여 사용자의 정상행위 패턴이 생성될 수 있다. 기존의 데이터마이닝을 이용한 침입 탐지 기법[5-7]들은 사용자 행위의 순차 패턴을 탐색하여 침입을 탐지하였다. 여기에서는 사용자의 정상행위 데이터뿐만 아니라 침입 데이터를 이용하여 사용자의 정상행위에 대한 허용범위를 학습한다. 하지만 파일 크기 등과 같이 양을 위주로 하는 데이터를 정확하게 모델링할 수 없다는 단점을 가지고 있다.

한편, 기존의 클러스터링 기법[15-18]을 그대로 적용하여 정상행위 패턴을 생성할 경우에 문제점이 발생될 수 있다. 즉, 기존의 클러스터링에서는 사용자 트랜잭션의 분포가 많이 밀집되어 있을 때 클러스터로 생성되며 클러스터 생성 시 최근 사용자의 행위와 과거 행위에 동일한 영향을 미치게 된다. 그러나 일반적으로 사용자가 새로운 업무로 작업이동을 하는 경우에 이 사용자는 새로운 작업에 대한 행위를 계속할 확률이 높아지게 된다. 따라서 이 사용자에 대해서 생성된 이전 정상행위 패턴으로는 비정상행위를 정확하게 판정하기가 힘들어 진다. 본 논문에서는 최근에 행한 사용자 패턴이 앞으로 발생할 확률이 높고, 과거에 행한 패턴은 발생할 확률이 적다는 사실에 중점을 두어서 사용자의 정상행위 패턴을 생성한다. 결과적으로 사용자의 가장 최근 행동을 반영한 사용자 정상행위 패턴이 생성됨으로써 보다 효과적으로 비정상 행위를 판별할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 침입 탐지 모델의 기본적인 형태를 분류하고 다양한 침입 탐지 시스템을 소개한다. 3장에서는 클러스터링 기법을 이용하여 사용자의 정상행위 패턴 생성 방법을 제안하며 4장에서는 3장에서 제시한 정상행위 패턴을 이용하여 사용자의 비정상 행위를 탐지하는 방법을 소개한다. 5장에서는 비정상행위 판정 시스템의 성능향상을 위한 모의 실험 결과를 비교 분석하며 6장에서 최종적인 결론을 맺는다.

2. 관련 연구

일반적으로 침입 탐지 모델은 오용 탐지 모델(Misuse Detection Model)과 비정상행위 탐지 모델(Anomaly Detection Model)로 분류된다[10]. 오용 탐지 모델은 시스템 상에서 잘 알려진 약점을 이용한 공격 탐지 방법으로 알려진 침입 패턴과 일치하는 데이타 또는 이벤트의 발생 순서등을 통해서 탐지하게 된다. 오용 탐지 모델은 전문가 시스템[1], 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 데이타에서 이 모델이 존재하는지의 여부를 조사한다.

오용 탐지 모델들의 기본적인 단점은 기존에 알려진 패턴에 대한 처리만이 가능하다는데 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. 이를 보완하기 위해서 최근에는 비정상행위 탐지 모델[3, 8]에 대한 연구가 상당히 많이 진행되고 있다.

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[2, 3, 13], 예측 가능한 패턴 생성(Predictive Pattern Generation)[1]등이 있다. 통계적인 방법은 비정상행위 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 IDES[13], NIDES[2], EMERALD[3]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[7]과 AAFID[14]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 데이타 발생하려는 시도를 할 때 쉽게 감지될 수 있다.

3. 사용자 정상행위 패턴 생성

대용량의 사건들이 기록되어 있는 데이타베이스에서 사용자의 유사한 작업군을 탐색하는 기법을 클러스터링이라고 한다. 기존의 클러스터링 기법[15-18]에서는 사용자 트랜잭션 정보를 이용하지 않고 클러스터를 생성하였다. 하지만 사용자의 작업 단위가 트랜잭션 단위이기 때문에, 침입탐지 환경에서는 트랜잭션 정보가 사용자의 정상행위 패턴 생성에 있어서 상당히 중요하다. 연관 규칙[19]과 순차 패턴[20] 탐사에서는 트랜잭션 정보를 지지율을 이용한 데이터마이닝을 수행하였다. 본 논문에서도 연관 규칙과 순차패턴에서와 유사하게 지지율을 이용하여 사용자의 정상행위 패턴을 생성하도록 하였다. 결과적으로 침입 탐지 환경에서 데이터를 정확하고 효과적으로 모델링할 수 있다.

본 논문에서는 사용자의 행위를 다양한 판정 요소들로 분류하여 각 판정 요소마다 클러스터링을 수행함으로써 사용자 행위에 대한 세부적인 비정상 행위도를 평가할 수 있다. 이를 위해서 <표 1>과 같은 판정 요

〈표 1〉 판정 요소

종 류	판정 요소
시스템 리소스	CPU 사용량, IO 사용량, 메모리 이용량, 실행 시간
시스템 호출	사용자 행위에 대해서 각 시스템 호출의 발생량
파일 관련	파일 접근, 파일 읽기, 파일 쓰기, 파일 생성, 파일 삭제, 파일 변환

소들을 이용할 수 있다.

일반적으로 각 사용자에 대한 정상행위 패턴을 생성하기 위해서는 일정 기간 내에 존재하는 사용자 트랜잭션을 분석해야 한다. 각 트랜잭션에는 사용자의 다양한 행위가 포함될 수 있으며 클러스터링을 수행하기 위해 사용자의 행위를 판정 요소에 의해서 수치 값으로 표현된다. 즉, i번째 트랜잭션에서 j번째 사용자 행위를 $a_{i,j}$라 하면, 판정 요소 m_k에 대한 행위 값은 $m_k(a_{i,j})$와 같이 표현될 수 있다. 또한, 전체 트랜잭션에서 판정 요소 m_k에 대한 데이타의 집합을 M_k와 같이 표현될 수 있으며 이를 이용하여 실질적인 클러스터링이 수행된다. 집합 M_k에 대한 클러스터링 수행 시 클러스터 분류를 위해서 사용자의 특정 행위 $a_{i,j}$에 유사한 행위 집합을 다음과 같이 표현될 수 있다.

$$N_\lambda(a_{i,j}) = \{x \mid m_k(a_{i,j}) - \lambda \le x \le m_k(a_{i,j}) + \lambda, x \in M_k\}$$

λ : 클러스터링 범위

여기에서 M_k내에서 인접한 두 행위를 $a_{s,t}$와 $a_{s,t+1}$이라 했을 때, 두 행위가 동일한 클러스터에 포함되기 위한 조건은 다음과 같다.

① $N_\lambda(a_{s,t}) \cap N_\lambda(a_{s,t+1}) \neq \varnothing$.

② $sup(N_\lambda(a_{s,t})), sup(N_\lambda(a_{s,t+1})) \ge min\text{-}support$

$sup(N_\lambda)$: N_λ의 지지율.

조건 ①은 클러스터링 범위 λ에 대해서 두 집합 $N_\lambda(a_{s,t})$와 $N_\lambda(a_{s,t+1})$에서 가장 인접한 두 행위 값간의 차가 λ이하임을 나타낸다. 즉, 인접한 두 행위 값이 클러스터링 범위내에 존재하게 되므로 같은 클러스터로 묶일 수 있다. 그렇지 않으면 서로 다른 클러스터로 분리된다. 조건 ②에서는 두 집합 $N_\lambda(a_{s,t})$와 $N_\lambda(a_{s,t+1})$가 클러스터링 수행을 위해서 주어진 최소 지지율(min-support)을 만족하는지의 여부를 판별하게 된다. 여기에서 N_λ의 지지율을 계산하기 위해서 N^i_λ를 i번째 트랜잭션 T_i에 포함되는 행위 집합이라 정의할 수 있다. 따라서 $sup(N_\lambda)$은 다음과 같이 계산된다.

$$sup(N_\lambda) = \sum_{i=1}^{n} I(T_i, N^i_\lambda)$$

$I(T_i, N^i_\lambda)$: if $T_i \supseteq N^i_\lambda$, then 1, otherwise 0

한편, 사용자의 작업이 변할 경우, 사용자의 과거 행동보다는 최근 행동에 비중을 둔 정상 행위 패턴이 생성이 필요하다. 이를 위해서 감쇄율(decay rate)[2]이 적용될 수 있다. 예를 들어 생성하고자 하는 사용자의 정상 행위 패턴에 대한 영향율이 50%가되는 위치(half-life)[2]를 1일 전으로 설정하면 1, 2, ..., k일 전 데이타의 데이타에 대한 영향율은 각각 50%, 25%, 12.5%, ..., $2^{-k}*100\%$와 같다. 여기에서 감쇄율 d는 다음과 같이 계산된다.

$$d = -log_2(0.5) / half\text{-}life$$

따라서 클러스터링 과정에서 사용자의 유사한 행동 패턴 그룹의 지지율을 날짜에 대해서 감쇄시키게 되면 과거보다는 최근 사용자 행동 패턴에 더 많은 비중을 두고 클러스터를 생성하게 된다. 감쇄율을 적용한 유사집합 N_λ의 지지율은 정의 1과 같다.

[정의 1] 감쇄율을 적용한 N_λ의 지지율

$$sup_d(N_\lambda) = \sum_{i=1}^{n} I(T_i, N^i_\lambda) \cdot 2^{-d\cdot(n-i)}$$

d : decay rate

결과적으로 감쇄율을 적용한 클러스터링 알고리즘은 (그림 1)과 같다. 알고리즘 FNA는 각 사용자에 대해서 판정요소(MeasureID), 최소 지지율(MinSupport),

```
Find_Normal_Activity(D(UserID,Measure ID), MinSupport,
λ, decay-rate)
Nλ(a,decay-rate) : 행위 값 a의 클러스터링 반경 λ내에 존재하
는 데이타 집합

Sort by UserID as major key and MeasureID as minor key;
for all data a in D(UserID, Measure ID) {
  if (a is unclassified) {
    retrieve Nλ(a, decay-rate);
    if (sup(Nλ(a, decay-rate)) < MinSupport) set noise to a
       and return;
    else {
     set new cluster-id to all data in Nλ(a, decay-rate);
     push all data from Nλ(a, decay-rate) onto stack;
     while(not stack.empty()) {
       current = stack.pop();
       retrieve Nλ(current, decay-rate);
       if (sup(Nλ(current, decay-rate)) >= MinSupport) {
         select all data unclassified/marked as noise in
         Nλ(current, decay-rate);
         stack.top = -1
         set current cluster-id to these data;
         push the unclassified data onto stack; }
       }
    }
}
```

(그림 1) 클러스터링 알고리즘 Find Normal Activity(FNA)

클러스터링 범위(λ)와 감쇄율이 주어졌을 때 사용자의 유사한 작업군을 탐색하게 된다. 탐색된 작업군이 주어진 최소 지지율 이상의 사용자 트랜잭션을 포함하고 있다면 클러스터로 생성되고 그렇지 않을 경우에는 잡음(noise)으로 처리되어 클러스터에 포함시키지 않는다. 알고리즘 FNA의 상세한 수행과정은 다음과 같다.

[1단계] 클러스터링을 위해서 주어진 데이타를 오름차순으로 정렬한다.

[2단계] 클러스터링은 정렬된 데이타의 맨 처음 데이타부터 시작하며, 데이타의 유효 반경 안에 존재하는 데이타들의 집합 N_λ을 구한다.

[3단계] N_λ내에 존재하는 모든 데이타들에 대해서 날짜에 대해 감쇄된 지지율을 구하고 이를 최소 지지율과 비교한다. 이때 최소 지지율보다 작으면 이 데이타의 상태는 Noise로 설정된다. 만일 최소 지지율 이상이면 이 데이타에 새로운 클러스터 ID를 부여하고 N_λ에 포함되어 있는 데이타들을 스택(stack)에 넣고 스택이 빌 때까지 다음 단계를 수행한다. 만일 스택이 비어있으면 2단계를 다시 수행한다.

[4단계] 스택의 맨 상단으로부터 데이타를 읽어와서 이 데이타에 대한 N_λ을 구한다. 만일 N_λ의 지지율이 최소 지지율보다 크면 N_λ내에 존재하는 unclassified /noise 데이타에 현재의 클러스터 ID를 부여한다.

[5단계] 스택을 초기화 한다. N_λ내에서 이전에 잡음(noise)이나 클러스터 ID가 부여되지 않은 새로운 데이타들을 스택에 넣는다. 이 과정을 모든 데이타가 접근이 될 때까지 반복한다.

최소 지지율과 감쇄율이 지지율에 영향을 주는 반면 클러스터링의 범위는 실질적으로 유사한 작업 그룹을 묶는 정도에 영향을 준다. 만일 클러스터링 범위가 너무 좁게 설정되면 클러스터가 만들어지지 않을 수 있으며 너무 크게 설정되면 생성된 클러스터의 정확도가 떨어지게 된다. 따라서 최적의 클러스터링 범위를 설정하는 것은 생성될 클러스터의 개수와 클러스터의 정확도에 상당히 많은 영향을 미치게 된다. 이를 위해 실험을 통해서 최적의 클러스터링 범위를 찾도록 하였다.

4. 비정상 행위 판정 방법

비정상행위 판정은 시스템에서 발생되는 이벤트 감시 과정에서 사용자의 로그인에 관련된 이벤트가 발생되면 그 사용자의 프로파일을 미리 가져와서 메모리에 상주시킨다. 그리고 관리자가 설정한 비교 기준인 명령어 또는 시스템 호출의 발생 횟수가 되었을 때 그 사용자의 정상행위 패턴과의 비교를 통해서 비정상행위를 판정하게 된다. 여기에서 특정 사용자에 대한 비정상 행위 판정은 그 사용자의 정상행위 프로파일을 이용하여 온라인에서 발생되는 사용자의 행위정도에 대한 차이가 높을 경우 비정상 행위도가 높아지게 된다. 사용자 정상행위 프로파일에는 클러스터링 결과로 생성된 각 판정 요소의 클러스터 리스트를 유지한다. 사용자 프로파일에 포함되는 클러스터는 클러스터 평균값(C), 클러스터 지지율(S), 최대 정상 행위값(max), 최소 정상 행위값(min)으로 구성된다. 클러스터 평균값은 클러스터 내에 존재하는 데이타들의 평균값이고, 클러스터 지지율은 클러스터 내에 존재하는 감쇄된 트랜잭션의 수이다. 또한 최대 정상행위 값과 최소 정상행위 값은 각각 사용자 정상행위 중에서 클러스터 평균값과의 차가 최대, 최소인 값이다.

클러스터링을 통하여 생성된 클러스터들은 각각 서로 다른 지지율을 가질 수 있다. 따라서 모든 클러스터들을 동일한 척도로 비정상 행위를 판정할 때 판정율에 문제점을 가져오게 된다. 예를 들어, 특정 판정 요소에 대해서 클러스터 A의 지지율이 0.9인 반면 클러스터 B의 지지율이 0.5인 경우에 분명히 비정상 행위도에 대한 척도가 달라야 한다. 즉, 생성된 클러스터를 이용하여 온라인에서 비정상 행위를 판정할 때 사용자의 행위에 대해서 클러스터 A와 클러스터 B의 평균값과의 차가 같다면 클러스터 A에서 더 높은 비정상 행위도가 나타나야 한다. 이를 위해서 정의 2에서와 같이 클러스터 평균값으로부터의 상대거리를 정의할 수 있다.

[정의 2] 클러스터 평균값으로부터의 상대 거리

클러스터의 평균값을 C라 하고 지지율을 s라 하면 클러스터에서 사용자 행위 x까지 상대거리(D)는 다음과 같다.

$$D = |C - x| \cdot \frac{s_m}{s}$$

s_m : *모든 클러스터 지지율들의 평균*

상대거리는 절대거리 |C - x|에 상대 지지율 s_m / s를 곱하였다. 이 수식의 의미는 만일 s가 평균 지지율 s_m보다 크면 x가 클러스터에 더 많은 영향을 받게 된다. 즉 클러스터의 지지율이 높을수록 온라인 데이타에 대한 클러스터의 영향력이 크도록 할 수 있으며 클러스터의 지지율이 작을수록 클러스터의 영향력이 작도록 할 수 있다.

정의 2를 이용하여 클러스터가 두 개 이상 생성됐을 때 온라인 데이타의 비정상 행위도를 판정할 경우 어떤 클러스터와 비정상 행위를 판정할 것인지를 결정해야 한다. 이를 위해서 정의 3에서와 같이 두 클러스터 사이의 상대적 평형점을 구하여 온라인 데이타와 상대적으로 가까운 위치에 존재하는 클러스터와 비정상 행위도를 계산하게 된다.

[정의 3] 클러스터 사이의 상대적 평형점

두 클러스터 사이의 상대적 평형점은 클러스터에 대한 상대거리를 비교함으로써 얻을 수 있다. 즉, 클러스터 A와 B에 대해서, 임의의 점 ε를 선택하고 두 클러스터와 ε에 대한 상대 거리가 같을 경우 ε는 두 클러스터의 상대적 평형점이 된다. 예를 들어, 두 클러스터 A와 B의 평균값과 지지율이 각각 C_A, C_B, s_A, s_B라면 상대적 평형점(ε)은 다음과 같이 계산될 수 있다.

$$|C_A - \varepsilon| \cdot \frac{s_m}{s_A} = |C_B - \varepsilon| \cdot \frac{s_m}{s_B} \quad \Rightarrow$$

$$\varepsilon = \frac{s_A \cdot C_B + s_B \cdot C_A}{s_A + s_B} \quad (C_A \le \varepsilon \le C_B)$$

만일 사용자의 새로운 행위값 x가 ε보다 크면 클러스터 B로부터 비정상 행위도를 구할 수 있으며 ε보다 작으면 클러스터 A로부터 비정상 행위도를 구할 수 있다. 또한 x가 ε와 같을 때 절대 거리의 차가 큰 클러스터에 대한 비정상 행위도를 구할 수 있다. 두 클러스터 사이에 존재하는 상대적 평형점의 의미는 지지율이 높은 클러스터일수록 영향을 미치는 범위가 넓고 온라인에서의 사용자 행위는 지지율이 높은 클러스터에 더 영향을 받아야 함을 나타낸다. 따라서 영향력이 큰 클러스터로부터 비정상 행위도를 구할 수 있다.

사용자의 비정상 행위 정도는 온라인에서의 사용자 행위가 클러스터의 평균값과의 차($|C_k - m_k(a)|$)로 나타낼 수 있다. 예를 들어 판정 요소 k에 대한 비정상 행위도는 다음과 같이 구할 수 있다.

$$Match(C_k, a) = \frac{\| C_k - m_k(a) | - \min_k |}{\max_k - \min_k} \cdot 100$$

a : 온라인 데이타
C_k : $m_k(a)$에 상대 거리가 가장 작은 클러스터
max_k : 판정 요소 m_k에 대한 최대 정상행위 값
min_k : 판정 요소 m_k에 대한 최소 정상행위 값

이 수식에서는 서로 다른 판정요소의 특징들을 정규화 해주기 위해 $max_k \sim min_k$로 나누어준다. 만일 사용자가 정상행위를 하였다면 비정상 행위도는 $min_k \sim max_k$ 범위내에 존재하게 된다. 따라서 특징이 다른 각 판정요소들을 최대 정상행위와 최소 정상행위와의 차이를 이용하여 정규화 시킬 수 있다. 결과적으로 온라인 데이타 a에 대해서 모든 판정 요소에 대한 비정상 행위 판정은 다음과 같이 계산될 수 있다.

$$A = \frac{1}{\gamma} \cdot \sum_{k=1}^{\gamma} Match(C_k, a_k)$$

γ : 판정 요소의 개수

일반적으로 비정상 행위를 판정할 때 연속적인 사용자의 행동 패턴이 파악돼야 하며 하루를 기준으로 했을 때 사용자의 행동에 대해서 비정상 행위도가 여러 차례 계산돼야 한다. 이때 한번의 사용자 행위에 대한 비정상 행위도를 사용자 감시 과정에서 현재 사용자의 행위로 파악 될 경우에 잘못된 판정을 할 가능성이 있다. 예를 들어 사용자가 새로운 작업을 수행하기 위해서 새로운 파일을 생성하는 것은 전체적으로 보면 정상적인 행위이지만 이 시점에서는 비정상적인 행위로 파악될 수 있다. 이에 대한 해결책으로 지금까지 계산된 모든 비정상 행위에 대한 평균을 사용자의 현재 행동으로 파악될 수 있다. 그러나 비정상 행위도에 대한 단순 평균은 최근의 사용자 행위를 감출 수 있다. 예를 들어, 최근에 비정상적인 행동을 했다 하더라도 지금까지 정상적인 행동을 해왔기 때문에 전체적으로는 정상행위에 가깝게 나타날 수 있다. 따라서 비정상 행위를 판정할 때 식 5와 같이 온라인 감쇄율을 적용하여 사용자의 비정상 행위도가 가장 최근의 사용자 행위에 영향을 받도록 하였다.

$$Abnormality = \frac{1}{N_\delta} \cdot \sum_{i=1}^{n} 2^{-\delta \cdot i} \cdot A_i$$

δ : 온라인 감쇄율
n : 판정 횟수
A_i : i 번째 비정상 행위도

5. 실험 및 성능 평가

본 논문에서는 4장에서 제시한 수식을 이용하여 다양한 실험으로 통해 비정상행위 판정율을 향상시키도록 하였다. 모의 실험 데이타는 UNIX 기반의 Solaris 2.6을 사용하는 사용자에 대해서 두 달 동안의 데이타를 수집하여 사용자 정상행위 패턴을 생성하였다. 이를 위해서 UNIX시스템 기반에서 Solaris 2.6용 BSM (Basic Security Module)[21]을 활용하여 로그 데이타를 수집하였다. BSM은 C2 레벨의 보안(security)을 제공하는 툴로써, 228개의 커널 신호(signal)를 인식하고 감사 로그파일에 기록한다. 보통 UNIX의 ls 명령에서 BSM의 경우는 50개에서 100개까지 이벤트 레코드(시스템 호출)가 발생된다.

한편, 본 논문에서 클러스터링을 위한 사용자 트랜잭션은 세션별 및 시간대별 트랜잭션으로 구분될 수 있다. 세션별 트랜잭션은 한 사용자가 로그인(Log-in)으로부터 로그아웃(Log-out)까지의 작업을 말하며 여러 개 로그인을 동시에 해서 작업을 하더라도 하나의 세션으로 간주된다. 또한 사용자가 로그인 한 후에 아무런 작업을 하지 않고 설정된 시간이 지나면 자동 로그 아웃한 것으로 간주한다. 시간대별 트랜잭션은 하루를 등분한 시간대를 사용자 트랜잭션으로 간주한다. 실험에서 시스템 사용자는 크게 프로그램을 작성하는 그룹, 일반 명령어만을 사용하는 그룹, 그리고 관리자 그룹으로 나뉘어진다. 실험에서 사용된 평가 파라미터(Parameter)는 <표 2>와 같다.

〈표 2〉 실험에 사용된 평가 파라미터

파라미터	실 험 값
온라인 Half-life	1, 2, 4, 8, 16, 32, 64
최소 지지율	50, 60, 65, 70, 75, 80, 90
클러스터링 범위	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024
시간 간격	1일(24시간)

<표 2>에서 정리한 바와 같이 판정 시스템에서 판정율 향상을 위해서 적용해야 할 평가 파라미터들은 다양하다. 따라서 각 파라미터의 변화에 따른 정상행위 패턴을 생성하고 최대 판정율을 가진 파라미터 값의 범위를 찾고자 한다.

<표 3>에 <표 6>까지는 클러스터링 범위 및 지지율에 따라서 생성된 클러스터의 개수를 표현하였다. 이 실험에서는 클러스터링의 범위를 좁게 설정하면 클러스터의 개수가 많아지고 클러스터링 범위가 늘어날수록 생성되는 클러스터는 점점 줄어들게 된다. 이것은 좁은 클러스터링 범위에서는 클러스터가 많이 생성되었지만 이들이 큰 클러스터링 범위에서 병합되었기 때문이다. 또한 최소 지지율이 작으면 데이터의 극히 일부분에서도 클러스터가 생성될 수 있기 때문에 클러스터의 개수가 많아지지만 최소 지지율이 증가할수록 이들이 제거되기 때문에 클러스터의 개수가 줄어들게 된다.

〈표 3〉 시스템 리소스 관련 판정 요소에 대한 클러스터의 평균 개수

support \ range	1	2	4	8	16	32	64	128	256	512	1024
50	60.5	55	45.5	36.5	25.5	30	32	24	11	9.5	8.5
60	37.5	35	36	32	21.5	17	24	19.5	12	7.5	7.5
70	30	25	24	25	20	10	18	16.5	14.5	8.5	6.5
80	24	17.5	18	20.5	16.5	11	10	11.5	11.5	8	6.5
90	21	15.5	11.5	12	12	9.5	7	7.5	6.5	8	6

〈표 4〉 시스템 호출 관련 판정 요소에 대한 클러스터의 평균 개수

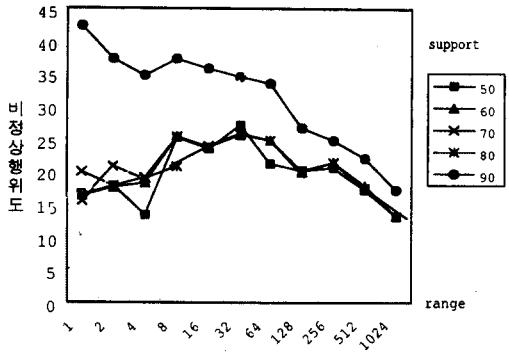
support \ range	1	2	4	8	16	32	64	128	256	512	1024
50	63	56.5	46.5	38.5	26.5	31	34	25	12.5	10.5	9.5
60	38.5	37	37	33	23	18	25	20.5	13	8.5	8.5
70	31.5	26.5	25.5	26	21.5	11.5	19	17.5	15.5	9.5	7.5
80	25	18.5	19.5	21.5	17.5	12.5	11	12.5	12.5	9	7.5
90	22	16.5	13	13	13	10.5	8	8.5	7.5	9	7

〈표 5〉 파일 접근 관련 판정 요소에 대한 클러스터의 평균 개수

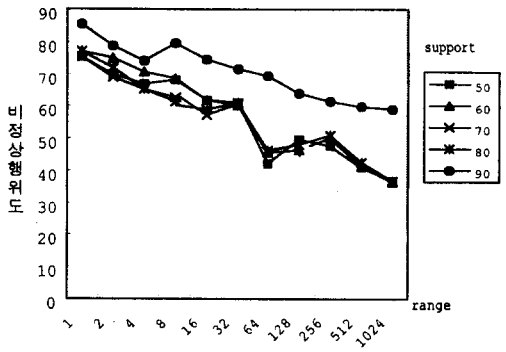
support \ range	1	2	4	8	16	32	64	128	256	512	1024
50	13	11	10	7.5	7.5	6	6.5	6.5	6	6	6
60	12	10	8.5	8.5	7	6.5	6	6	6.5	6	6
70	11	10	8	7.5	6.5	6	6	6	6	6	6
80	8.5	7	9	7	6.5	6	6	6	6	6	6
90	7	7	6.5	6	6	6	6	6	6	6	6

〈표 6〉 기타 판정 요소에 대한 클러스터의 평균 개수

support \ range	1	2	4	8	16	32	64	128	256	512	1024
50	14.5	12	10.5	10	11	9	8.5	8	7	7	8
60	12	11.5	9.5	8.5	8.5	9.5	8	7.5	7	7	7
70	11.5	10	9	8	8	8.5	8.5	7.5	7	7	7
80	10.5	9	10	8	7.5	7.5	7.5	7.5	7	7	7
90	7.5	8	8.5	7	7	7	7	7	7	7	7



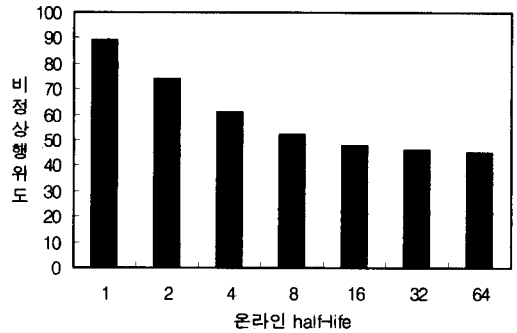
(a) 정상 데이타



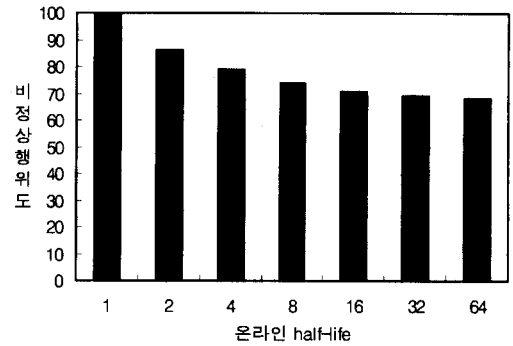
(b) 비정상 데이타

(그림 2) 클러스터링 범위/지지율에 따른 비정상 행위도

<표 3>, <표 4>, <표 5>, <표 6>에서 생성된 클러스터들을 이용하여 비정상 행위도를 판정한 것이 (그림 2)이다. (그림 2(a))에서는 정상행위 프로파일을 생성한 데이타를 이용하여 비정상행위를 판정하였고, (그림 2(b))에서는 다른 사용자의 프로파일을 이용하여 비정상행위를 판정하였다. (그림 2(a))에서 지지율이 높을 때에는 정상행위 데이타임에도 불구하고 비정상행위도가 상당히 높아짐을 알 수 있다. 이것은 지지율이 높아지게 되면 생성되는 클러스터의 개수가 적어지게 됨으로써 사용자의 다양한 행위에 대해서 비정상 행위도가 높아지게 된다. (그림 2(b))에서는 침입자에 대해서 비정상 행위도가 높게 나타나야 함에도 불구하고 클러스터링 범위가 커지게 되면 비정상 행위도가 상당히 떨어짐을 알 수 있다. 이것은 클러스터링 범위가 커짐에 따라 침입자의 행위도 이에 근접할 수 있기 때문이다. 따라서 정상 사용자인 경우에는 비정상 행위도가 작게 나타나고 침입자인 경우에는 비정상 행위도는 높게 나타나는 지점을 선택해야 한다. 이 실험에서는 지지율을 90%이하로, 클러스터링 범위를 32이하로 설정됐을 때 효과적인 판정을 기대할 수 있었다.



(a) 자신의 프로파일을 이용한 최대 비정상 행위도



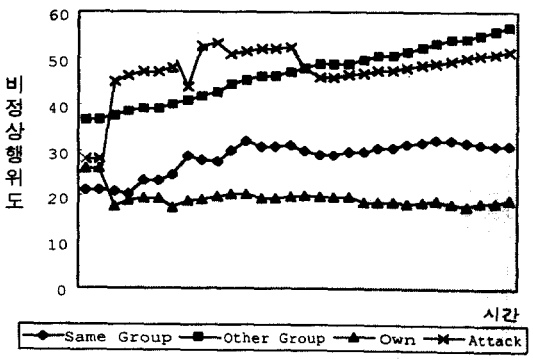
(b) 침입자에 대한 최대 비정상 행위도

(그림 3) 온라인 half-life에 따른 비정상 행위도

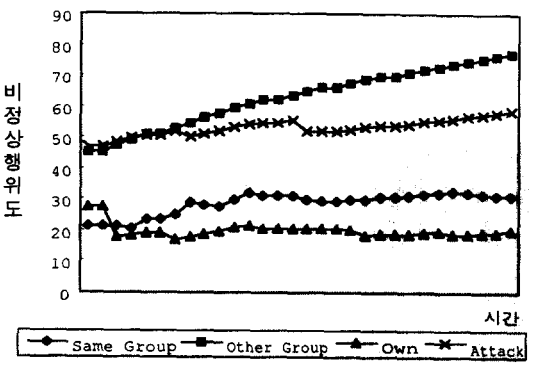
(그림 3)에서는 온라인 감쇄율을 적용하여 사용자의 비정상행위를 판정하는 실험을 하였다. 온라인 감쇄율의 의미는 사용자의 가장 최근의 비정상 행위도를 반영하기 위함이다. 여기에서 어떤 사용자에 대한 순간적인 비정상 행위를 판정하기 위해서는 시스템의 환경에 따라 최적의 온라인 감쇄율을 적용하는 것이 바람직하다. 만일 온라인 half-life를 작게 설정하게 되면 이 사용자의 비정상 행위도를 즉각적으로 파악할 수 있는 반면 침입자가 아니라 하더라도 비정상행위도가 높아지는 문제점이 발생된다. 이 실험에서는 50% 이상을 비정상 행위로 파악될 때, 온라인 half-life는 16으로 설정해 주는 것이 바람직하게 나타났다.

(그림 4)는 half-life가 30일, 최소 지지율이 60%,

온라인 half-life가 20일 때 자신의 프로파일을 사용한 경우(Own), 다른 그룹의 프로파일을 사용한 경우 (Other Group), 같은 그룹의 프로파일을 사용한 경우 (Same Group), 그리고 사용자가 계정을 도용 당해서 비정상행위를 하는 경우(Attack)에 대한 변화 결과를 보여 주고 있다. 자신(Own)인 경우는 점점 비정상행위도가 낮아지며 같은 그룹의 프로파일을 이용한 판정에서도 사용자 자신의 프로파일 정도는 아니지만 점점 비정상행위도 낮아진다. 그러나 공격을 당한 사용자의 경우 다른 그룹의 사용자와 비슷한 비정상 행위도가 나타난다. 이것은 이 사용자의 경우에 이전과는 다른 작업 수행 또는 권한 남용으로 자신과 작업이 다른 그룹의 비정상 행위와 파악되기 때문이다.



(a) 세션별 비정상 행위도



(b) 시간대별 비정상 행위도

(그림 4) 정상행위 및 비정상행위 변화 과정

6. 결 론

기존의 많은 연구들이 통계적인 기법을 이용하여 호스트 기반 침입을 탐지를 수행하였다. 하지만 통계적인 기법은 사용자 행위에 대한 평균치를 이용하여 사용자의 비정상 행위가 상당히 부정확하게 판정될 가능성을 가지게 된다. 또한 적은 빈도수지만 주기적으로 발생되는 사용자 행위에 대해서 희소 카테고리로 관리됨으로써 효과적으로 비정상 행위를 판정할 수 없다. 이를 해결하기 본 논문에서는 방대한 데이타 분석을 좀 더 지능적이고 자동적으로 수행하기 위한 데이타마이닝 기법 중에서 클러스터링 기법을 활용하여 사용자의 정상행위 패턴을 분석하였다. 이를 위해서 사용자의 비정상행위를 다양한 각도에서 분석될 수 있도록 사용자의 행위가 여러 가지 판정요소들로 분류될 수 있으며 각 판정 요소에 대해서 클러스터링을 수행하여 사용자의 정상행위 패턴이 생성될 수 있다. 이때 사용자의 최근 행동이 과거의 행동보다 정상행위 패턴을 생성하는데 많은 영향을 주기 위해서 기존의 클러스터링에 감쇄율 개념을 적용하였다. 이를 통해서 비정상 행위 판정 환경이 바뀌게 될 때 사용자에 대해서 효과적으로 비정상 행위를 탐지할 수 있다. 이와 더불어 다양한 모의 실험을 통해 판정 시스템의 탐지율을 높이고 오판율을 줄이기 위한 최적의 임계치 값에 대한 결과를 보였다.

참 고 문 헌

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

[2] Harold 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.

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

[4] Jai Sundar Balesubramaniyan, Jose Omar 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] K. Illgun, R. Kemmerer, Phillip 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 and Teresa 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] Jai Sundar Balesubramaniyan, Jose Omar 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.

[15] Martin Ester, Hans-Peter Kriegel, Sander, Michael Wimmer, Xiaowei Xu, "Incremental Clustering for Mining in a Data Warehousing Environment," Proceedings of the 24th VLDB Conference, New York, USA, 1998.

[16] Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, "ROCK : A Clustering Algorithm for Categorical Attributes," the 15th International Conference on IEEE Data Engineering, Sydney, Australia, 1999.

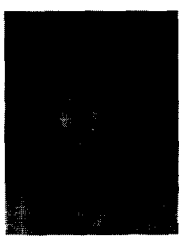
[17] Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, "CURE : An Efficient Clustering Algorithm for Large Databases," ACM SIGMOD International Conference on Management of Data, Seattle, Washington, 1998.

[18] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "Birch : An Efficient data clustering method for very large databases," Proceedings for the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996.

[19] R. Agrawal, R. Srikant : "Fast Algorithms for Mining Association Rules," Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept. 1994.

[20] R. Agrawal, R. Srikant : "Mining Sequential Patterns," Proc. of the Int'l Conference on Data Engineering (ICDE), Taipei, Taiwan, March 1995.

[21] Sun Microsystems. SunShield Basic Security Module Guide.



오 상 현

email : osh@amadeus.yonsei.ac.kr

1996년 제주대학교 정보공학과 졸업(학사)

1998년 연세대학교 컴퓨터과학과 졸업(석사)

1998년~현재 연세대학교 컴퓨터과학과 박사과정

관심분야 : 침입탐지시스템, 데이터마이닝, 분산 에이전트 시스템

이 원 석

email : leewo@amadeus.yonsei.ac.kr

1985년 미국 보스턴 대학교 컴퓨터공학과 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 : 침입탐지, 멀티미디어 데이타베이스, 비디오 데이타 모델링, 분산 데이타베이스