通信情報保護學會論文誌
第 10 券 第 4 號, 2000. 12

# 사용자 명령어 분석을 통한 비정상 행위 판정에 관한 연구*

윤 정 혁**, 오 상 현***, 이 원 석***

# A Study on Anomaly Detection based on User's Command Analysis

Jeong-Hyuk Yoon**, Sang-Hyun Oh***, Won-Suk Lee***

요 약

컴퓨터와 통신기술의 발달로 사용자에게 다양한 정보와 편리성이 제공된 반면, 컴퓨터 침입 및 범죄로 인한 피해가 날로 증가하고 있으며 다양한 침입 방법들이 새롭게 사용되고 있다. 따라서 침입자들의 행위를 효과적으로 탐지하기 위해서는 기존의 오용탐지 방법과 더불어 비정상행위 모델의 적용에 대한 필요성이 증가하고 있다. 본 논문에서는 비정상행위 탐지 모델에서 사용자의 정상행위 패턴 생성 시 최근에 관찰된 사용자의 행위에 더 많은 영향을 주도록 하는 새로운 연관 규칙 알고리즘을 제시한다. 또한 생성된 정상행위 패턴을 토대로 사용자별 그리고 사용자간 클러스터링 과정을 수행함으로써 작업의 유사성을 가진 그룹의 명령어 또는 프로그램 이용정도를 파악한다. 이와 더불어 다양한 실험을 통해서 본 논문에서 제안된 비정상행위 판정시스템에서 탐지율을 최대화 할 수 있는 임계치 값들을 제시한다.

ABSTRACT

Due to the advance of computer and communication technology, intrusions or crimes using a computer have been increased rapidly while various information has been provided to users conveniently. As a result, many studies are necessary to detect the activities of intruders effectively. In this paper, a new association algorithm for the anomaly detection model is proposed in the process of generating user's normal patterns. It is that more recently observed behavior gets more affection on the process of data mining. In addition, by clustering generated normal patterns for each user or a group of similar users, it is possible to identify the usual frequency of programs or command usage for each user or a group of users. The performance of the proposed anomaly detection system has been tested on various system parameters in order to identify their practical ranges for maximizing its detection rate.

**keyword** : *intrusion detection, anomaly detection, data-mining, association-rule, clustering*

## Ⅰ. 서 론

컴퓨터와 통신 기술의 발달로 사용자에게 다양한 정보와 편리성이 제공된 반면, 컴퓨터 침입 및 범죄로 인한 피해도 날로 증가하고 있다. 따라서 침입자들의 행위를 보다 효과적으로 탐지하기 위한 비정상행위 판정 기술 연구가 필요하다.

침입자들의 시스템 공격은 초기에는 침투 기법이

* 본 연구는 2000년 한국정보보호센터 위탁 연구과제로 수행하였습니다.
** 유로코넷(hyuk@amadeus.yonsei.ac.kr)
*** 연세대학교 컴퓨터과학과({osh, leewo}@amadeus.yonsei.ac.kr)

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[1-6]. 최근에는 방대한 데이터 분석을 좀 더 지능적이고 자동적으로 수행하기 위해서 데이터 마이닝 기법을 이용하여 사용자의 정상행위를 모델링하고 있다[6-8].

본 논문에서는 연관 규칙 탐사[15]를 이용하여 유닉스(unix) 환경에서 사용자가 실행한 명령어 정보를 시스템 로그로부터 추출하여 사용자 명령어 사용에 대한 정상행위 패턴을 추출한다. 이를 통해서 평소와 다른 형태의 행위를 하는 사용자에 대한 비정상 행위를 판별하게 된다. 그러나 기존의 연관 규칙 방법을 그대로 적용하여 정상행위 패턴을 생성 할 경우에 문제점이 발생될 수 있다. 즉, 기존의 연관 규칙을 이용한 패턴 생성 과정은 아이템이 발생된 트랜잭션의 횟수로 지지도를 구하게 된다. 따라서 최근에 많이 실행한 명령어와 과거에 많이 실행된 명령어의 이용 정도가 정상행위 패턴 생성에 동일한 영향을 미치게 된다. 그러나 일반적으로 사용자가 새로운 명령어를 습득하게 되었거나 새로운 업무로 작업이동을 하는 경우에 이 사용자는 최근에 알게 된 명령어 또는 새로운 업무에 관련된 프로그램을 앞으로 계속 사용할 확률이 높아지게 된다. 따라서 이 사용자에 대해서 생성된 이전 정상행위 패턴으로는 비정상행위를 정확하게 판정하기가 힘들게 된다. 본 논문에서는 최근에 사용자가 사용한 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. 이와 더불어 본 논문에서 제안된 비정상 행위 판정 시스템에서는 생성된 정상행위 패턴을 토대로 사용자별 그리고 사용자간 클러스터링 과정을 수행함으로써 작업의 유사성을 가진 그룹에 대한 정상행위 모델링이 가능하다.

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

## II. 관련 연구

침입이란 권한이 없는 사용자가 발생시키는 문제 또는 합법한 사용자가 권한을 남용하는 것이라고 정의한다[9]. 이와 더불어 자원의 유용성, 기밀성, 그리고 무결성 등에 저해되는 행동 집합을 침입이라고 정의하기도 한다[10]. 본 논문에서 사용자의 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[11].

오용 탐지 모델은 시스템 상에서 잘 알려진 약점을 이용한 공격 탐지 방법으로 알려진 침입 패턴과 일치하는 데이터 또는 이벤트의 발생 순서등을 통해서 탐지하게 된다. 오용 탐지 모델은 전문가 시스템[1], 상태 전이 분석[11,12], 모델 기반 기법[13] 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[11]와 USTAT[12]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.

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. 이에 대한 해결책으로 최근에는 비정상행위 탐지 모델[3,5,9]에 대한 연구가 활발히 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[2,3,14]과 예측 패턴 생성 방법 (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[14], NIDES[2] 및 EMERALD[3]등이 있다.

예측 패턴 생성 모델에서 사용자의 행위는 순서적으로 발생한다는 가설에 근거한 것으로 시간 기반의 규칙을 이용하여 사용자의 각 행위에 시간 요소를 부여하여 발생된 행위들이 순서적으로 올바른지 또는 각 행위들 사이의 시간적인 간격이 올바른지를 조사하여 사용자 행위의 정상 또는 비정상 여부를 결정한다.

비정상행위 탐지 모델에서 주로 사용하는 통계적인 방법의 경우 실시간 판정에서 사용하게 되는 프로파일 데이터를 최소화 할 수 있다는 장점을 가지고 있는 반면, 감사 데이터를 통계적인 수치 값으로 표현함으로써 데이터의 손실이 발생될 수 있다. 예를 들어 오전에 수행하는 업무와 오후에 수행하는 업무가 다른 경우에 통계적인 방법은 이러한 두 가지 작업에 대한 평균값을 유지하게 된다. 따라서 오전 업무와 오후 업무가 확연히 다른 사용자는 오전 업무와 더불어 오후업무의 통계적인 분포에 따라 정상행위가 모델링되므로, 오전 또는 오후 업무 자체를 정확히 모델링하기 어려울 수 있어 본인의 작업에 대해서도 비정상행위도가 상대적으로 높게 탐지 수 있다. 본 논문은 데이터 사이에 존재하는 연관성 및 잠재되어 있는 지식을 용이하게 획득할 수 있는 데이터 마이닝 기법을 이용하여 다량의 데이터에서 사용자의 정상행위 패턴을 찾고, 사용자의 업무 종류에 따라서 사용자 클러스터를 유지하게 된다. 결과적으로, 오전에 수행한 업무와 오후에 수행한 업무가 완전히 다른 사용자에 대해서 각 업무군을 별도로 모델링하여 보다 정확한 모델링이 가능하도록 지원한다.

## Ⅲ. 정상행위 패턴 모델링

본 장에서는 사용자의 정상행위 패턴을 생성하기 위해서 기존의 연관 규칙에 감쇄율을 적용하여 사용자의 정상행위 패턴을 생성하는 알고리즘을 제안한다. 또한 각 사용자의 서로 다른 작업군에 대한 분류 및 유사 작업을 하는 사용자 그룹에 대한 분류를 위한 클러스터링 알고리즘을 소개한다. 여기서는 소단원에 관한 내용을 간단히 살펴보겠습니다.

### 3.1 사용자 정상행위 패턴 생성

대용량의 사건들이 기록되어 있는 데이터베이스에서 자주 발생되는 아이템간의 상호 연관성 탐사 기법으로부터 생성된 패턴을 연관 규칙이라고 한다. 연관 규칙을 이용하여 사용자의 일반적인 행위 패턴을 추출함으로써 사용자의 새로운 행동에 대한 비정상행위도를 파악할 수 있다. 하지만 기존의 연관 규칙을 그대로 이용하게 되었을 때, 사용자의 과거 행동과 최근 행동에 대해서 동일한 비중을 가지고 패턴이 생성된다. 하지만 사용자의 작업이 이동되었을 경우, 사용자의 최근 행동뿐만 아니라 과거의 행동에 대해서도 정상 행위 패턴이 생성됨으로써 사용자의 최근 행동에 대한 비정상 행위도를 올바르게 파악할 수 없다. 이를 해결하기 위해서 사용자의 최근 행동에 보다 높은 비중을 주기 위한 정상 행위 패턴 생성 방법이 필요하다. 사용자의 과거 행위보다 최근 행위에 더 많은 비중을 주기 위해서 감쇄율[14]이 이용될 수 있다.

예를 들어 생성하고자 하는 사용자의 정상 행위 패턴에 대한 영향율이 50%가되는 위치(half-life)를

1일 전으로 설정하면 1, 2, ..., k일 전의 데이터가 현재 시점에 주는 영향율은 각각 50%, 25%, 12.5%, ..., $2^{-k}$*100%와 같다. 여기에서 감쇄율은 다음과 같이 계산된다.

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

이를 이용하여 k번째 전날의 패턴이 정상행위 패턴을 생성하는데 영향을 주는 비율(Pattern Effect Rate)[17]은 아래 식과 같이 계산될 수 있다. 이 식에서 패턴 감쇄율이 너무 높게 설정되면 정상행위 패턴 생성에 참여하는 데이터들 중 최근 데이터에 많은 비중을 주기 때문에 정상행위 패턴이 주로 최근 데이터로부터 생성될 가능성이 높아진다. 반면, 감쇄율이 너무 작게 설정되면 보다 많은 범위의 데이터들로부터 정상행위 패턴이 생성됨으로써 사용자의 최근 행위에 대한 모델링이 힘들어진다.

$$Pattern\ Effect\ Rate(P.E.R) = 2^{-dk}$$

$k$ : 기준 날로부터 k번째 이전날
$d$ : 패턴 감쇄율

〔표 1〕의 변수들을 이용하여 감쇄율을 적용한 사용자의 정상 행위 패턴 탐색 알고리즘은 〔그림 1〕과 같다. 여기에서 트랜잭션은 사용자의 작업단위이며 작업 시작에서부터 작업이 종료되는 시점까지를 나타낸다.

알고리즘 Decay_Association_Algorithm은 시스템에서 발생되는 사용자 트랜잭션 로그를 이용하여 사용자의 정상적인 행위 패턴을 연관 규칙을 적용하여 탐사한다. 이때 생성되는 정상행위 패턴은 각 트랜잭션마다 항상 같이 발생되는 명령어 집합을

〔표 1〕 연관 규칙 알고리즘에서 사용된 변수

| 변수 | 내 용 |
|---|---|
| $D$ | 정상행위 패턴 생성 날짜 |
| $T$ | 트랜잭션이 발생된 날짜(명령어가 실행된 날짜와 시간) |
| $L_k$ | $k$크기의 빈발 아이템 집합 (large item set) |
| $C_k$ | $k$크기의 명령어 후보 아이템 집합(candidate item set) |
| *Sys-Log* | 시스템에서 발생되는 사용자 트랜잭션 로그 데이타 |
| *min-sup* | 최소 지지율 |

```
Procedure Decay_Association_Algorithm()
aging(D)=2^(-(D-T) * ( log2(0.5)/half-life));
min-sup=ΣP.E.R*Min-Support-Rate;
L1={large 1-items};
for (k=2 ; Lk-1 ≠0; k++) begin
    Ck=apriori-gen(Lk-1);

  foreach candidates c∈Ck begin
    foreach 'Sys-Log' transaction t∈D begin
      if (subset(t, c) is true) then
          c.support=c.support+aging(day of t);
      endif
    endfor
    if (c.support>=min-Sup) then
      add c to Lk ;
      foreach l where subset(c, l)(l∈Lk-1) is true
        and c.support-l.support< min-sup)
      begin
        delete pattern l from Lk-1;
      endfor
    endif
  endfor
endfor
End Procedure
```

〔그림 1〕 감쇄율을 이용한 연관 규칙 알고리즘

찾아서 사용자의 정상 행위를 추출하는데 목적이 있다. 각 사용자 트랜잭션은 발생된 날짜가 다를 수 있으며 최근 데이터를 패턴에 더 많이 반영하기 위해서 각 트랜잭션에 감쇄율을 적용시켰다. 알고리즘의 상세한 수행 과정은 다음과 같다.

**[STEP 1]** 시스템 로그에서 추출된 명령어에 대해서 트랜잭션별로 재구성하고 각 트랜잭션이 발생된 날짜에 따라 감쇄율을 적용하여 $P.E.R$을 구한다.

**[STEP 2]** 로그 데이터에서 각 사용자마다 트랜잭션이 발생된 날짜가 다르고 트랜잭션의 수도 다르기 때문에 각 사용자마다의 최소 지지도를 구해야 하며 최소 지지도는 $P.E.R$의 총합에 미리 설정한 최소 지지율을 곱한다.

**[STEP 3]** 최소 지지도를 만족하는 아이템 집합 크기 1인 빈발 아이템 집합($L_1$)을 구성한다.

**[STEP 4]** 최소 지지도를 만족하는 크기 1인 빈발 아이템 집합($L_1$)을 이용하여 Apriori 알고리즘[16]을 적용하여 크기 2인 후보 아이템 집합($C_2$)을 생성한다. $C_2$는 $L_1$과 $L_1$의 조인을 통해서 자신의 아이템 집합보다 크기가 1개 큰 아이템 집합을 생성한다. Apriori 알고리즘의 자세한 설명은[16]을 참고한다.

**[STEP 5]** 생성된 후보 아이템 집합들에 대한 지지도를 구한다. 지지도는 후보 아이템 집합을 포함하고 있는 트랜잭션의 *P.E.R*을 합하여 구할 수 있다.

**[STEP 6]** 계산된 지지도에 대해서 후보 아이템 집합들이 최소 지지도를 만족하지 않는 경우 제거된다. 또한 생성된 후보 아이템 집합이 크기가 작은 후보 아이템 집합을 포함하고 있는 경우에 각 아이템의 지지도가 같다면 크기가 작은 후보 아이템 집합은 제거된다. 이것은 새로 생성된 후보 아이템 집합이 이전 후보 아이템 집합의 의미를 모두 포함하고 있기 때문이다.

**[STEP 7]** 후보 아이템 집합의 크기를 1씩 증가시키면서 더 이상의 정상행위 패턴이 나오지 않을 때까지 [STEP 4]에서 [STEP 6]과정을 반복한다.

**[STEP 8]** 생성된 사용자 정상행위 패턴에 포함되어 있는 명령어의 평균 사용량을 계산한다. 기존의 연관 규칙에서는 한 트랜잭션에서 여러 번 발생되는 아이템 집합들을 별도로 구분하지 않고 모두 한번 발생된 것으로 간주하였다. 하지만 침입 탐지환경에서는 반복되는 행위 자체가 의미를 갖게 된다. 따라서, 기존의 연관 규칙을 보안하고 비정상행위 판정율을 향상시키기 위해서 정상행위 패턴에 포함되는 명령어의 평균 사용량을 활용한다. 명령어(C)에 대한 평균값은 다음과 같이 계산될 수 있다.

$$AVG(C)=\frac{1}{N_C}\cdot\sum_{i=1}^{N_C}T_i(C)$$

$N_C$ : 명령어 $C$가 나타난 트랜잭션의 개수
$T_i(C)$ : $i$ 번째 트랜잭션에서 명령어 $C$의 빈도수

**[STEP 9]** 정상행위 패턴을 구하고자 하는 사용자에 대해서 [STEP 1]에서 [STEP 8]의 과정을 반복한다.

[그림 2]는 감쇄율을 이용한 연관 규칙을 적용하여 트랜잭션단위의 사용자 정상행위 패턴을 만드는 과정이다. Half-life가 1일이고 최소 지지율이 50%이면 감쇄율은 $-\log_2(0.5)/1=1$와 같이 계산된다. [그림 2](a)에서는 P.E.R을 구하고 각 트랜잭션별로 발생한 명령어 ID 순으로 정렬한다. [그림 2](a)에서 각 트랜잭션마다 구해진 *P.E.R*의 합에 최소 지지율을 곱해서 최소 지지도를 구하면 다음과 같다.

| Date | Transaction ID | Program ID(사용 횟수) | Pattern Effect Rate |
|---|---|---|---|
| 6/2 | 1 | 2(2) 3(2) 5(2) | $2^{-5}$ (=0.03125) |
| 6/2 | 2 | 1(1) 2(1) 3(2) 5(3) | $2^{-5}$ (=0.03125) |
| 6/3 | 3 | 2(2) 5(2) | $2^{-4}$ (=0.0625) |
| 6/3 | 4 | 2(2) 3(2) 5(2) | $2^{-4}$ (=0.0625) |
| 6/4 | 5 | 6(3) 9(2) | $2^{-3}$ (=0.125) |
| 6/5 | 6 | 6(3) 7(1) 9(2) | $2^{-2}$ (=0.250) |
| 6/6 | 7 | 5(2) 6(1) 9(1) | $2^{-1}$ (=0.500) |
| 6/7 | 8 | 6(1) 7(1) 9(1) | $2^{0}$ (=1.000) |

(a) 초기 데이터

| Itemset-1 | Support (Percent) |
|---|---|
| {6} | 1.875 (90.9) |
| {7} | 1.250 (60.6) |
| {9} | 0.875 (90.9) |

(b) $L_1$

| Itemset-2 | Support (Percent) |
|---|---|
| {6 7} | 1.250 (60.6) |
| {6 9} | 1.875 (90.9) |
| {7 9} | 1.250 (60.6) |

(c) $L_2$

| Itemset-3 | Support (Percent) |
|---|---|
| {6 7 9} | 1.250 (60.6) |

(d) $L_3$

[ 최종 정상행위 패턴 ]
{6(2) ,7(1), 9(1.5)}

[그림 2] 감쇄율을 이용한 정상행위 패턴 생성 예제

[그림 2](b)에서는 트랜잭션 데이터로부터 크기 1인 아이템 집합에 대한 지지도를 구하여 최소 지지도 1.03125 이상인 빈발 아이템 집합($L_1$)을 생성하였다. [그림 2](c)는 [그림 2](b)의 아이템 집합을 이용하여 후보 아이템 집합들을 생성하고 각각에 대한 지지도를 구하여 최소 지지도를 만족하는 아이템 집합만 남겨 놓았다. [그림 2](d)는 [그림 2](c)의 크기 2인 정상패턴으로부터 크기 3인 정상패턴을 생성하는 과정이다. 여기서 {6 7}, {6 9}, {7 9}는 모두 새로 만들어진 {6 7 9}에 포함관계에 있고 {6 7}, {7 9}는 독립적으로 발생되지 않았다. 그 이유는 {6 7 9}에 대한 지지도와 같기 때문에 항상 같이 발생한다. 따라서 {6 7}, {7 9}는 정상행위 패턴 집합에서 삭제한다. 또한 {6 9}는 {7}과 관련 없이 독립적으로 사용자의 행위 패턴으로 나타날 확률이 0.625/2.0625=30%(2.0625는 1~6일까지의 P.E.R의 총합)이므로 정상행위 패턴의 최소 지지율을 넘지 못하므로 {6 9} 패턴은 정상행위 패턴 집합에서 삭제된다. 결국 정상행위 패턴은 {6 7 9}가 생성되었다. 여기에서 패턴 내의 명령어 6, 7 9에 대한 평균 사용량은 각각 2, 1, 1.5와 같다.

### 3.2 사용자별 클러스터링

본 절에서는 사용자의 정상행위 패턴에 따라서 항상 같이 실행되는 패턴을 하나로 묶는 과정인 사용자별 클러스터링[17]을 설명한다. 사용자별 클러스터링은 사용자가 여러 업무를 수행한 경우 각 업무에 필요한 명령어군을 탐사하는 과정이라 정의할 수 있다.

즉, 연관 규칙을 통해서 만들어진 사용자의 정상행위 패턴이 동시에 발생되는 정상행위 패턴군을 생성하는 과정이다. 예를 들어 어떤 사용자가 오전 업무로 메일만 점검하고 오후 업무로 프로그램만 작성한다면 이 사용자의 정상행위 패턴은 두 종류이며 이 사용자의 업무는 독립적인 두 가지 업무로 분류될 수 있다. 따라서 서로 다른 두개의 클러스터로 분류된다. 반면 어떤 사용자가 프로그램을 작성하는 업무와 메일을 점검하는 업무가 최소 클러스터 유사도보다 큰 값으로 트랜잭션에서 발생하였다면 두 개의 업무는 거의 별개로 일어나는 것이 아니라 함께 발생하는 업무일 것이다. 따라서 한 개의 클러스터에 두 가지 정상패턴이 포함될 수 있다.

사용자별 클러스터링 알고리즘은 다음과 같이 수행된다.

**[STEP 1]** 감쇄율을 적용해서 생성된 정상행위 패턴이 어떤 트랜잭션에서 발생되었는지를 탐색한다.

**[STEP 2]** 정상행위 패턴 사이에 클러스터 유사도를 계산한다. 예를 들어 A 패턴이 발생된 트랜잭션 ID 집합이 $V_a$와 같고, B 패턴이 발생된 트랜잭션 ID 집합이 $V_b$와 같다면 클러스터의 유사도는 다음과 같이 계산된다(|V| : 집합 V의 원소 개수).

$$Min\left(\frac{|V_a \cap V_b|}{|V_a|}, \frac{|V_a \cap V_b|}{|V_b|}\right) \cdot 100 \geq S$$

$S$ : 최소 클러스터의 유사도

이 식에서는 A패턴이 나타난 트랜잭션 ID 집합 $V_a$와 B 패턴이 나타난 트랜잭션 ID 집합 $V_b$에 대해서 $V_a$가 $V_b$에 포함된 확률과 $V_b$가 $V_a$에 포함될 확률중 최소값이 최소 유사도 S를 넘는지를 계산한다. 만일 클러스터 유사도의 최소 값이 미리 정의된 최소 클러스터 유사도 값 이상이면 두 정상행위 패턴은 거의 항상 같은 트랜잭션에서 발생되는 패턴이라고 말할 수 있으므로 하나의 클러스터를 형성한다. 그렇지 않으면 두 개의 클러스터로 분리된다. 한편, 최소 클러스터 유사도 $S$가 너무 높게 설정되면 사용자 정상행위 패턴간에 유사성이 거의 나타나지 않기 때문에 유사한 작업군을 찾지 못할 수 있다. 반면, 클러스터 유사도를 너무 낮게 설정되면 탐색된 유사 작업군 내의 패턴들 간의 신뢰도가 상당히 떨어지게 된다.

예를 들어 [그림 3]에서는 최소 지지율 50%, 최소 클러스터 유사도 70%일 경우 사용자의 정상행위 패턴은 {2 3 5} {1 3}이 생성된다. 여기에서 {2 3 5}에 대한 트랜잭션 ID 집합은 {2, 4}와 같고, {1 3}에 대한 트랜잭션 ID 집합이 {1, 4}와 같으므로 클러스터 유사도를 구하면 50%과 같다. 그러나 계산된 클러스터 유사도가 최소 클러스터 유사도 70%이상을 만족하지 못하므로 두개의 클러스터로 분리된다.

| Transaction ID | ProgramID Set |
|---|---|
| 1 | 1 3 4 |
| 2 | 2 3 5 |
| 3 | 2 5 |
| 4 | 1 2 3 5 |

[그림 3] 사용자별 클러스터링 예제

**[STEP 3]** 더 이상 클러스터가 발생되지 않을 때까지 [STEP 2]과정을 반복한다.

### 3.3 사용자간 클러스터링

본 절에서는 사용자별로 만들어진 클러스터를 이용하여 전체 사용자의 유사한 업무에 대한 작업 군의 생성 과정을 설명한다. 사용자간 클러스터링 과정을 수행함으로써 얻는 효과는 다음과 같이 크게 세 가지로 볼 수 있다.

첫째, 사용자의 그룹을 나누어 사용자의 작업 군을 확인함으로써 각 사용자의 권한을 남용한 내부 오용 여부나 사용자에게 할당된 업무 수행 형태를 파악할 수 있다. 예를 들어, A그룹에 속한 사용자가 자신의 권한을 벗어나는 B그룹의 작업을 계속 수행하였다면 이 사용자의 정상행위 패턴은 B그룹에 포함된다. 결국 자신의 권한을 벗어나는 작업을 수행하였다는 것을 발견할 수 있다. 둘째, 새로운 사용자가 시스템에 추가되면 이 사용자의 정상행위 패턴이 존재하지 않기 때문에 정상행위 패턴 생성 이전에는 이 사용자에 대한 판정이 어려워진다. 그러나 이 사용자의 업무를 알고 있다면 그와 유사한 업무를 하는 사용자의 정상행위 패턴을 이용하여 이 사용자의 행위가 비정상적인지를 판정할 수 있다. 셋째, 사용자가 너무 많아서 개별적인 정상행위 패턴을 저장하기 곤란할 경우에 같은 그룹에 속하는 사용자들의 정상행위 패턴은 유사하므로 이들을 통합하여 정상행위 패턴을 축약하여 전체적인 프로파일의 크기를 최소화할 수 있다.

전체 사용자 그룹에 대한 클러스터링 알고리즘은 다음과 같이 수행된다.

[STEP 1] 사용자별 클러스터에 대해서 정상행위 패턴 집합의 지지율을 구하여 내림 차순으로 정렬한다(지지율이 높은 점이 클러스터의 시작점으로 사용된다).

[STEP 2] 각 사용자에 대해서 자신의 패턴 집합과 유사한 패턴 집합을 찾아서 병합한다. 클러스터 유사도의 계산은 다음과 같다.

$$\frac{\sum_{i=1}^{k}\sum_{j=1}^{n} Min\left(\frac{|A_i \cap B_j|}{|A_i|}, \frac{|A_i \cap B_j|}{|B_j|}\right) \cdot 100}{k \cdot n} \geq S$$

$k$ : $A$ 사용자의 패턴 수
$n$ : $B$ 사용자의 패턴 수
$S$ : 최소 클러스터의 유사도
$A_i$ : $A$ 사용자의 패턴
$B_j$ : $B$ 사용자의 패턴

A 사용자에 대해서 $k$개의 패턴이 존재하고, B 사용자에 대해서 $n$개의 패턴이 존재하면, A가 B에 포함될 확률과 B가 A에 포함될 확률 중에서 최소값을 최소 클러스터 유사도와 비교한다. 사용자별 클러스터와 마찬가지로 최적의 최소 클러스터 유사도 값을 설정했을 때 효과적인 판정을 기대할 수 있다.

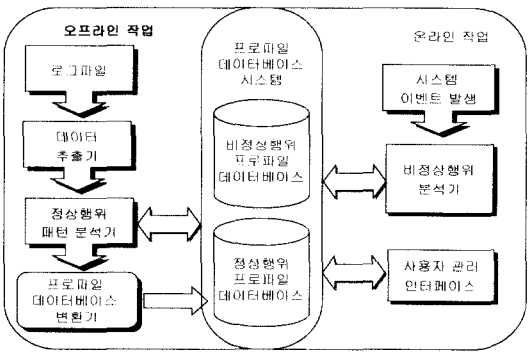
[STEP 3] 병합된 클러스터 사이에서 더 이상 패턴이 병합 될 수 없을 때까지 [STEP 2]과정을 반복한다.

예를 들어, 사용자 A의 정상행위 패턴이 {2 3}, {2 5}이고 사용자 B의 정상행위 패턴이 {2 3 5}이면 A와 B사이에 대한 유사도는 다음과 같이 계산된다.

$A_1$={2 3}, $A_2$={2 5}, $B_1$={2 3 5}.
$A_1$과 $B_1$의 유사도
=*Min*(2/2, 2/3)*100=66.7 %.
$A_2$과 $B_1$의 유사도
=*Min*(2/2, 2/3)*100=66.7 %.
사용자 A와 사용자 B 사이의 유사도
=(66.7+66.7)/2=66.7% .

## Ⅳ. 비정상행위 판정 시스템의 전체 구성

본 논문의 비정상행위 판정시스템은 크게 오프라인 작업과 온라인 작업으로 나뉜다. 오프라인 작업에서는 로그 파일의 정보를 재구성하여 다양한 분석 작업을 지원하기 위한 데이터베이스를 활용한다. 이를 토대로 데이터마이닝 기법을 이용하여 사용자별로 최적화 된 프로파일을 생성한다. 온라인 작업에서는 오프라인 작업에서 생성된 프로파일을 이용하여 사용자에 대해서 발생되는 비정상행위 패턴을 탐지한다. [그림 4]는 전체 시스템 흐름도를 도식화 한 것이다.



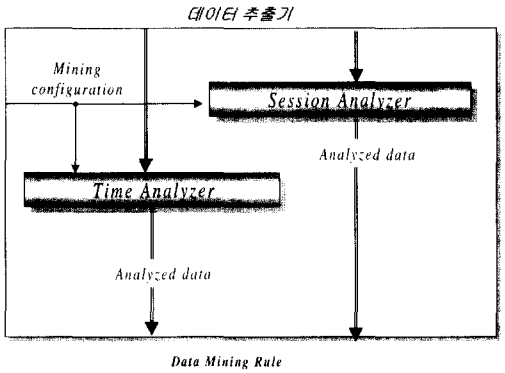
[그림 4] 비정상행위 판정 시스템의 전체구성도

### 4.1 정상행위 패턴을 위한 데이터 수집

시스템을 사용하는 사용자에 대한 정상적인 패턴을 생성하기 위해서는 정상행위에 대한 데이터를 수집해야 한다. 본 논문에서는 UNIX시스템 기반에서 Solaris 2.6용 BSM(Basic Security Module)[18]을 활용하여 로그 데이터를 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 발생한다. 사용자의 정상행위 패턴을 위한 데이터 수집 방법으로 BSM에서 기록하는 이벤트 중에서 exec이벤트와 execve이벤트에 관련된 감사 레코드를 분석하여 어떤 사용자가 어느 위치의 IP 어드레스에서 언제, 어느 디렉토리의 명령어를 실행시켰는지를 찾고, exit 이벤트를 통해서 프로그램 실행의 성공 실패여부에 대한 정보를 추출하여 사용자의 정상행위 패턴을 찾기 위한 입력 데이터로 활용된다.

### 4.2 정상행위 패턴 분석기

정상행위 생성 패턴 분석기는 3.1절에서 소개한 감쇄율을 적용한 연관 규칙 기법을 이용하고, 3.2절과 3.3절에서 언급한 방법으로 사용자간 그리고 사용자




〔그림 5〕 정상행위 패턴 분석기

별 작업 군을 생성한다. 본 논문에서는 사용자의 행위를 다양한 각도로 모델링하기 위하여 연관 규칙의 적용 단위인 사용자 트랜잭션을 세션별 및 시간대별 트랜잭션으로 구분하였다.

세션별 트랜잭션은 한 사용자가 로그인(Log-in)으로부터 로그아웃(Log-out)까지의 작업을 말하며 동시에 로그인된 여러 행위도 하나의 세션으로 간주한다. 또한 사용자가 로그인 한 후에 아무런 작업을 하지 않고 설정된 시간이 지나면 자동 로그아웃한 것으로 간주한다. 시간대별 트랜잭션은 하루를 균등한 크기로 등분한 시간대를 사용자 트랜잭션으로 간주한다. 〔그림 5〕는 정상행위 패턴 분석 과정을 도식화 한 것이다.

### 4.3 비정상행위 분석기

비정상행위 분석기는 온라인 상에서 사용자마다 발생하는 시스템 이벤트를 수집하여 사용자의 비정상행위를 탐지하는 시스템이다. 비정상행위 분석기는 시스템에서 발생되는 이벤트 감시 과정에서 사용자의 로그인에 관련된 이벤트가 발생되면 그 사용자의 프로파일을 미리 가져와서 메모리에 상주시킨다. 그리고 이 사용자가 정해진 수의 명령어를 수행할 때마다 또는 로그아웃했을 경우 비정상행위를 판정하게 된다. 이때 각 세션 및 시간대별 정상행위 패턴과의 비교를 통해서 비정상행위를 판정하게 된다. 사용자가 로그인하여 실행시킨 명령어 리스트는 명령어 ID와 명령어 실행 회수에 대한 누적된 정보를 해당 사용자의 정상행위 패턴 프로파일과 비교하게 된다. 시간별 정상행위 패턴은 미리 설정된 시간대가 시작될 경우 이 시간대에 해당하는 프로파일을 읽어 들여서 판정하게 된다.

사용자의 비정상행위도는 온라인에서의 사용자 행위 집합과 사용자 정상행위 프로파일을 비교함으로써 계산된다. 온라인에서 사용자의 행위 집합을 T라 하고 정상행위 패턴 집합을 $R=\{r_1, r_2, \ldots, r_n\}$이라 하면 비정상행위도는 다음과 같이 계산된다.

비정상행위도 =

$$\left(1-\sum_{i=1}^{n}\frac{support(r_i)}{SS(R)}\cdot\frac{|r_i\cap T|}{|T|}\right)\cdot 100$$

$$SS(R)=\sum_{i=1}^{n}support(r_i)$$

$support(r_i)$ : 패턴 $r_i$에 대한 지지율

위 식에서 온라인에서의 사용자 행위(T)가 정상행위 패턴(R)과 유사하다면 $|r_i\cap T|/|T|$값이 커지게 됨으로써 비정상행위도가 낮게 나타나게 된다. 반면, 사용자가 이상 행위를 하였을 경우에는 정상행위 패턴과 유사하지 않기 때문에 비정상행위도가 높게 나타나게 된다.

한편, 특정 명령어가 한 트랜잭션 내에서 발생되는 빈도는 사용자의 비정상행위의 유무를 파악하는데 상당히 중요한 정보가 된다. 정상행위 패턴내의 명령어 C에 대한 평균 사용량을 $AVG(C)$라 하고 온라인에서 사용자 행위 집합(T)내의 명령어 C의 발생 빈도수를 $CNT(C)$라 하면 명령어 빈도수를 고려한 비정상행위도는 다음과 같이 계산된다.

명령어 빈도수를 고려한 비정상행위도 =

$$\left(1-\sum_{i=1}^{n}\frac{support(r_i)}{SS(R)}\cdot\frac{|r_i\cap T|}{T}\cdot F_i\right)\cdot 100.$$

$$F_i=\frac{1}{|r_i|}\cdot\sum_{a\in r_i}\sum_{b\in T}\left(1-E\cdot\left|\frac{AVG(a)-CNT(b)}{AVG(a)}\right|\right)\cdot I(a,b).$$

$I(a, b)$ : if $a=b$ then 1, otherwise 0

$E$ : 명령어 사용 횟수의 반영 비율

$$\left(0\le E\le\left|\frac{AVG(a)}{AVG(a)-CNT(b)}\right|\right).$$

여기에서 $E$가 작게 설정되면 명령어 사용 빈도수가 비정상행위도에 적게 영향을 주게 된다. 따라서 패턴간의 비교를 위주로 하는 비정상행위도가 계산된다. 반면, $E$가 높게 설정되면 비정상행위도는 패턴간의 비교보다는 명령어 빈도수에 상당히 많은 영향을 받게 된다.

예를 들어, 생성된 정상행위 패턴 {2(2) 3(2) 4(1)}과 {3(2) 4(1) 5(1)}의 지지율이 각각 60%, 90%와

같고 실행된 명령어 사용 회수의 반영 비율(E)이 0.1일 때, 패턴이 {2(1) 3(2) 5(1)}에 대한 비정상행위도는 다음과 같이 계산된다.

(60/150*3/4*1/3*(1-0.1*(2-1)/2+1-0.1*0)
+90/150*3/4*1/3*(1-0.1*0+1-0.1*0))*100
=49.5%

## V. 모의 실험 및 평가

본 장에서는 제안된 판정시스템의 성능에 대한 검증을 위한 모의실험을 수행하며 침입탐지율을 높이기 위해서 최적의 임계치 값의 범위를 파악할 수 있도록 대표적인 모의 실험 결과를 도표로 나타내고 이를 분석한다.

### 5.1 모의 실험 환경

모의 실험 데이터는 UNIX 기반의 Solaris 2.6을 사용하는 사용자에 대해서 두 달 동안에 약 2Gbyte 크기의 로그 데이터를 수집하여 사용자 정상행위 패턴을 생성하였다. 이 로그에 나타나는 사용자의 명령어 개숫는 1031개였다. 로그 데이터에서 사용자는 다음과 같이 크게 3개의 그룹으로 분류된다.

**[그룹1]** 프로그램을 작성하는 그룹: 컴파일러로 gcc를 사용하고, 편집기로 vi등을 사용한다. 그리고 기본적인 유닉스 명령어를 사용한다.

**[그룹2]** 일반 명령어만 사용하는 그룹: 유닉스의 기본적인 명령어만 사용한다. 예를 들면, ls, cp, mv 등의 명령어를 사용한다.

**[그룹3]** 관리자 그룹: 데이터베이스 관리자와 시스템 관리자 등으로 구분하고 관리에 필요한 명령어를 수행한다.

또한, 실험에서 사용될 평가 파라미터(Parameter)는 {표 2}와 같다.

{표 2} 실험 파라미터

| 파라미터 | 실험값 |
|---|---|
| Half-life | 1~1024, 0(∞) |
| 최소 지지율 | 50, 60, 65, 70, 75, 80, 90 |
| 클러스터 유사도 | 50, 60, 70, 80 |
| 시간 간격 | 1일(24시간) |

### 5.2 결과 및 분석

{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. 사용자의 비정상행위를 판정하기 위해서는 정상 행위 패턴을 생성하여 프로파일로 유지하게 된다. 사용자가 많은 시스템의 경우 프로파일의 크기가 커지게 되면 시스템 부하에 큰 영향을 줄 수 있기 때문에 최적화 된 프로파일을 생성하는 것이 중요하다. {표 3}과 {표 4}는 최적화된 프로파일을 위한 half-life와 지지율 값을 찾기 위한 실험 결과이다. {표 3}은 세션 단위에서 지지율과 감쇄율에 따른 한 사용자의 평균 프로파일의 크기이고 {표 4}는 시간 대 단위에서 평균 프로파일의 크기이다. 그림에서와 같이 지지율이 높을수록 프로파일의 크기가 감소하는 것을 알 수 있다.

{그림 6}과 {그림 7}에서는 각 사용자마다 정상행위 프로파일을 생성하고, 생성된 각자의 프로파일을 가지고 자신의 행위에 대한 비정상행위를 찾는 실험을 하였고, {그림 8}과 {그림 9}에서는 {그림 6}과 {그림 7}에서의 자신의 정상행위 프로파일에 대해 다른 그룹에 속한 사용자 행위에 대한 비정상행위도를 평가하는 실험을 하였다.

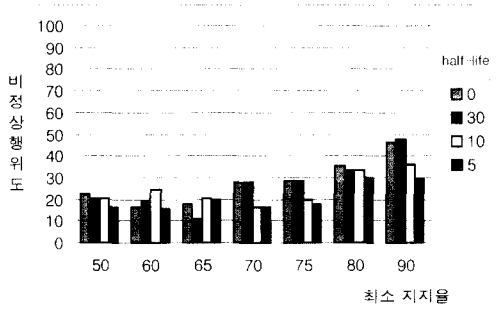
{그림 6}과 {그림 7}에서와 같이 지지율이 80%가 넘는 경우에는 비정상행위도가 다른 지지율에 비해

{표 3} 세션별 프로파일 크기 변화(단위 byte)

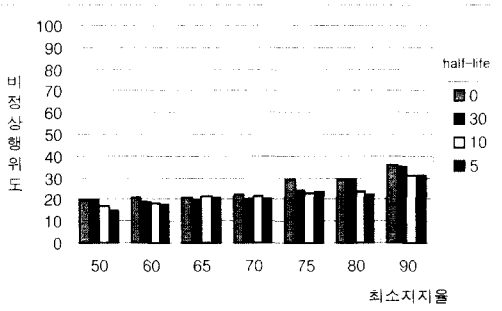
| 지지율 / halflife | 50 | 60 | 65 | 70 | 75 | 80 | 90 |
|---|---|---|---|---|---|---|---|
| 0 | 576 | 560 | 381 | 363 | 285 | 205 | 163 |
| 30 | 580 | 549 | 355 | 357 | 237 | 183 | 163 |
| 15 | 473 | 494 | 346 | 271 | 232 | 187 | 146 |
| 10 | 447 | 394 | 327 | 285 | 226 | 211 | 149 |
| 5 | 369 | 311 | 278 | 261 | 204 | 165 | 153 |

{표 4} 시간별 프로파일 크기

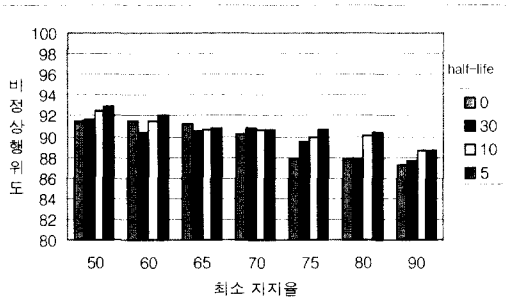
| 지지율 / halflife | 50 | 60 | 65 | 70 | 75 | 80 | 90 |
|---|---|---|---|---|---|---|---|
| 0 | 506 | 444 | 315 | 347 | 275 | 199 | 152 |
| 30 | 559 | 372 | 338 | 340 | 221 | 204 | 163 |
| 15 | 569 | 317 | 288 | 244 | 286 | 204 | 152 |
| 10 | 595 | 300 | 296 | 230 | 230 | 196 | 152 |
| 5 | 492 | 317 | 337 | 210 | 192 | 185 | 153 |



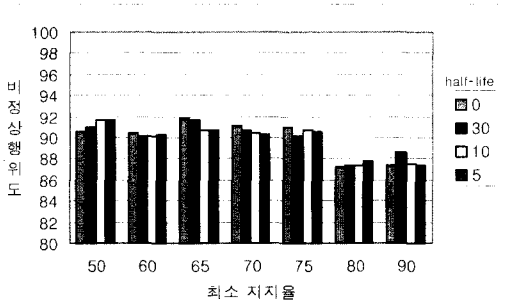

〔그림 6〕 세션별 비정상행위도(자신의 데이타)



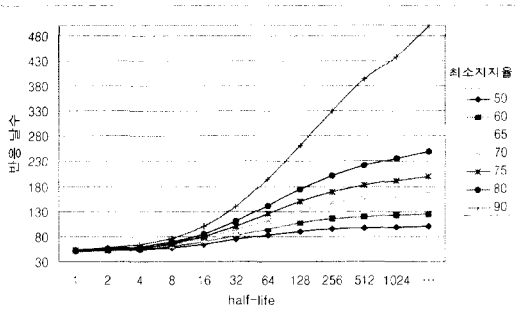

〔그림 7〕 시간별 비정상행위도(자신의 데이터)



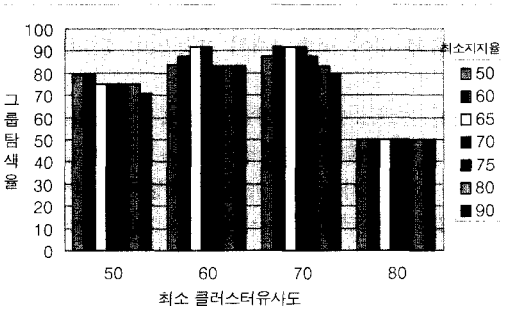

〔그림 8〕 세션별 비정상행위도(다른 그룹 데이타)




〔그림 9〕 시간별 비정상행위도(다른 그룹 데이타)




〔그림 10〕 최소 지지율과 half-life에 따른 반응 날수




〔그림 11〕 사용자간 클러스터링 측정

높게 나타나는 경우가 발생된다. 이것은 지지율이 높아지면서 생성되는 정상행위 패턴이 수가 감소되기 때문이다. 따라서 온라인에서 사용자의 정상적인 행위에도 불구하고 비정상행위도가 높게 나타나게 된다. 한편, 〔그림 8〕과 〔그림 9〕에서는 지지율이 80%이상인 경우에 비정상행위도가 떨어지는 것으로 나타났다. 이와 같이 나타나는 이유는 지지율이 높아지면서 각 사용자의 정상행위 패턴은 일반적으로 누구나 실행하는 명령어로만 이루어질 가능성이 커지게 된다. 따라서 온라인에서 어떤 사용자의 행위가 비정상적인가를 정확하게 파악하기가 힘들게 된다.

결국, 정상행위자인 경우에는 비정상행위도가 낮게 나타나면서 비정상행위자인 경우에는 높게 나타날 경우 최적의 탐지율을 보이게 된다. 본 실험에서는 지지율의 범위는 50%에서 65%로 설정하는 했을 때 오판율을 최소화 할 수 있는 것으로 나타났다

〔그림 10〕은 사용자의 작업이 급변하는 경우에 지지율과 half-life에 따른 반응날수를 나타낸 것이다. 여기에서 반응날수란 새로운 작업에 대해서 정상행위 패턴이 생성되기 위해 필요한 날수를 의미한다. 〔그림 10〕에서는 최소 지지율이 높고 half-life가 높을수록 많은 반응 날수를 필요로 함을 알 수 있다.

즉, 사용자의 작업이 바뀌었음에도 불구하고 계속해서 이전에 수행되었던 작업에 대한 정상행위 패턴을 이용하여 비정상행위 판정을 수행할 가능성을 가지고 있다. 따라서 정상적인 판정이 이루어지지 않는다. 이를 해결하기 위해서, 적용되는 환경에 따라서 어떤 반응 날수를 선택할 지에 따라서 최적의 half-life 및 최소 지지율을 구할 수 있다.

〔그림 11〕에서는 지지율의 변화 및 클러스터 유사도의 변화에 따른 그룹 탐색율을 이용하여 사용자간 클러스터링의 타당성을 검증하는 실험을 수행하였다. 여기에서 그룹 탐색율이란 클러스터링의 결과가 그 사용자의 실제 그룹으로 판정될 확률이다. 이 실험에서는 실제 그룹과의 일치여부를 조사하여 클러스터가 가장 잘 형성되는 클러스터 유사도 값을 찾는다. 이 실험에서는 사용자간 클러스터 유사도의 비율이 60-70% 정도가 가장 적당한 것으로 나타났다. 반면 80%가 넘어갈 경우에는 클러스터가 잘 형성되지 않는 것을 볼 수 있으며 50%이하에서는 구분되어야할 클러스터가 하나로 생성될 확률이 높은 것으로 나타났다.

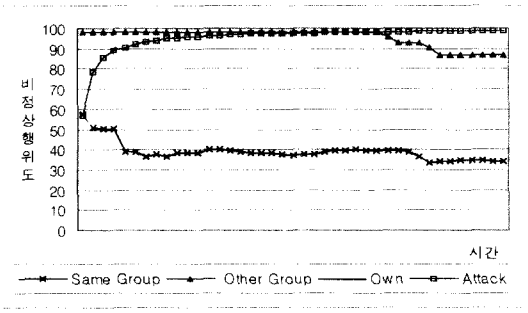
앞의 실험 결과를 토대로 제안된 비정상행위 판정 시스템을 위한 가장 최적화 된 임계치 값의 범위를 〔표 3〕과 같이 제시된다.

실험에서 감쇄율을 사용하였을 경우에 최적화된 프로파일을 생성하게 되었고 동일한 지지율에서 판정율과 오판율을 줄이는 효과를 가져왔다. 지지율이 높은 경우에 프로파일의 크기는 줄일 수 있는 반면, 너무 높은 지지율(80%이상)을 설정하면 판정율이 떨어질 수 있으므로 50-65%의 지지율이 적절하다. 클러스터 유사도의 경우에도 80%이상인 경우는 그룹으로 묶일 확률이 적어지고, 50% 미만인 경우는 다른 그룹의 사용자도 하나의 클러스터로 묶일 수 있으므로 60-70%정도의 값으로 설정하는 것이 적당하다. 또한 온라인에서 사용자의 행위에 대한 비정상행위도는 70%이상으로 나타날 때 이 사용자는 침입자로 간주될 수 있다.

그림 12에서는 〔표 3〕에서 제시된 임계치 값을 이용하여 온라인 판정을 수행한 결과를 보여준다.

〔표 5 비정상행위 판정을 위한 최적 임계치

| 실험에서 사용된 파라미터 | 실험값 |
|---|---|
| 최소 지지율 | 50-65 % |
| 클러스터 유사도 | 60-70 % |
| 판정을 위한 비정상행위도 | 70% 이상 |




〔그림 12〕 시나리오 판정 결과

〔그림 12〕에서 자신(Own)과 같은 그룹(Same Group) 사용자에서는 위에서 설명된 〔그룹 1〕과 〔그룹 2〕의 혼합 데이터 집합을 이용하였다. 다른 그룹(Other Group) 사용자의 데이터는 〔그룹 3〕의 데이터 집합을 이용하였다. 한편 가상 공격 시나리오(Attack)에서는 공격가능한 데이터 집합을 이용하여 판정을 수행하였다. 여기에서 비정상행위 판정에 사용되는 정상행위 프로파일은 Own 데이터를 이용하여 생성하였다.

〔그림 12〕에서 Own 데이터를 이용하여 판정하였을 경우에는 비정상행위도가 점점 낮아지는 것을 볼 수 있다. 이것은 실질적으로 정상행위 패턴 생성에 참여했던 데이터를 이용하여 판정하였기 때문에 비정상행위도가 낮게 나타나게 되는 것이다. 마찬가지로 Same Group 데이터를 이용하였을 경우에도 비정상행위도가 낮아지는 것을 볼 수 있다. Own 데이터에서와 마찬가지로 정상행위 패턴 생성에 포함되는 작업들을 Same Group에서도 유사하게 수행하였기 때문에 나타나는 결과이다. 한편, Other Group 데이터 및 Attack 데이터를 이용하였을 경우에는 비정상행위도가 상당히 높게 나타난다. 이것은 정상행위 패턴 생성에 참여했던 작업과 전혀 다른 작업을 수행하였기 때문에 나타나는 결과이다.

## Ⅵ. 결 론

기존의 많은 연구들이 통계적인 기법을 이용하여 호스트 기반 침입 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 판정

할 수 없다. 이를 해결하기 본 논문에서는 방대한 데이타 분석을 좀 더 지능적이고 자동적으로 수행하기 위한 데이타마이닝 기법 중에서 연관 규칙 탐사 기법을 활용하여 사용자의 정상행위 패턴을 분석하였다. 이때 사용자의 최근 행동이 과거의 행동보다 정상행위 패턴을 생성하는데 많은 영향을 주기 위해서 기존의 연관 규칙 탐사 알고리즘에 감쇄율 개념을 적용하였다. 이를 통해 사용자의 정상행위 패턴에 정확성 및 최적화 된 프로파일을 생성하는 효과를 가져왔다. 이러한 과정으로 생성된 각 사용자 정상행위 패턴의 유사도에 따라서 클러스터를 생성하여 사용자 작업의 종류 및 정상행위 패턴의 종류를 파악할 수 있었고 사용자간 클러스터링을 통하여 침입뿐만 아니라 사용자의 업무 패턴에 대한 분석에 활용될 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] 한국정보보호센터, 호스트기반 침입 탐지시스템 개발에 관한 연구, 1998.12.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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

[12] 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.

[13] 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.

[14] 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.

[15] Rakesh Agrawal, T. Imielnski and A. Swami, "Mining Association Rules between Sets of Items in Large Database,"

In Proc. ACM SIGMOD, pp. 207~216, 1993.
[16] Rakesh Agrawal, Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules," In Proc. Of the 20th VLDB conference, 1994.
[17] 윤정혁, 오상현, 이원석, "사용자 명령어 추적을 통한 정상행위 패턴 탐사", 한국 정보 처리 학회 12회 추계 학술 대회 논문집, 1999. 10.
[18] Sun Microsystems. SunShield Basic Security Module Guide
[19] 윤정혁, 사용자 명령어 분석을 통한 비정상행위 패턴 판정에 관한 연구, 석사 학위 논문, 연세대학교, 1999. 12

## 〈著者紹介〉

**윤 정 혁 (Jeong-hyuk Yoon)**
1996년 2월 : 한양대학교 전산학과 졸업
1996년 1월~1997년 6월: 현재전자 멀티미디어 연구소
2000년 2월 : 연세대학교 컴퓨터과학과 석사
2000년 1월~2000년 10월: 온세통신 인터넷 사업본부
2000년 10월~현재 : 유로코넷
〈관심분야〉 정보보안, 침입탐지 시스템, 데이타마이닝

**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