데이터베이스 보안을 위한 사용자 정상행위 패턴탐사

박 정 호, 오 상 현, 이 원 석
연세대학교 컴퓨터과학과
e-mail : parang97@amadeus.yonsei.ac.kr

Discovering User's Normal Patters for Database Security

Park, Jeong Ho, Oh, Sang Hyun, Lee, Won Seok
Dept. of Computer Science, Yonsei University

요 약

최근의 네트워크를 통한 침입과 유형은 갈수록 다양화되고 있으며, 지능적으로 변하고 있다. 그러나 외부의 침입자뿐만 아니라 내부의 권한 오용으로 인한 침입의 탐지도 중요시되고 있으며, 그에 따른 운영체제와 네트워크 분야의 보안에 관한 연구 또한 활발히 진행되어 어느 정도의 성과를 얻고 있다. 그러나 데이터베이스의 보안은 데이터베이스 관리시스템에 거의 의존하고 있는 실정이다. 본 논문에서는 사용자의 정상행위를 효과적으로 모델링하기 위해서 데이터마이닝 기법인 연관규칙과 순차패턴을 이용하여 사용자의 정상행위 패턴을 추출하였다. 결과적으로 외부침입자 및 내부의 권한 오용자에 대한 비정상행위를 효과적으로 판정할 수 있다.

1. 서론

최근에 인터넷이 발전함에 따라 네트워크를 통하여 침입하는 사례가 상당히 늘어나고 있으며 이러한 침입에 효과적으로 대응하기 위한 보안관련 연구가 활발히 진행중이다. 특히 유닉스와 같은 운영체제, 그리고 네트워크의 약점을 이용한 침입탐지의 연구는 어느 정도의 성과를 거두고 있다. 그러나 실제 고객정보와 같은 중요한 데이터가 저장되어 있는 데이터베이스의 보안은 데이터베이스 시스템 내부의 기본적인 보안기능에만 의존하고 있는 실정이다. 본 논문에서는 데이터베이스의 기본적인 보안기능 외에 데이터마이닝을 적용하여 불법침입자와 권한 요용을 통한 침입탐지를 자동화하는 보안시스템에 관련된 연구를 수행한다.

본 논문의 구성은 다음과 같다. 2 장에서는 보안시스템에 관련된 연구를 분류하여 소개하며 3 장에서는 사용자의 정상행위 패턴을 생성하는 과정과 비정상행위를 판정하는 방법론을 제시한다. 4 장에서는 전체 시스템의 구성, 5 장에서는 모의실험 결과를 보여주며 6 장에서 결론을 맺는다.

2. 관련연구

시스템 보안분야에 있어서 데이터마이닝 기법의 적용은 상당히 많은 진전을 보았다. 운영체제와 네트워크 보안에 있어서 사용자 행위의 패턴추출에 관련된 연구는 꾸준히 수행되고 있으나 데이터베이스 보안에 있어서는 최근에 DEMIDS 시스템에서 소개되었다[2].

시스템의 침입이란 권한이 없는 사용자가 문제를 일으키는 것 또는 합법적인 사용자가 권한을 남용하는 것이라고 정의한다[1]. 외부 침입자란 권한이 없는 사용자가 문제를 일으키는 것을 말하며 내부 권한오용자란 합법적인 사용자이지만 자신의 권한을 남용하는 사용자를 말한다. 이러한 침입을 탐지하는 방식은 오용탐지 모델과 비정상행위 탐지 모델로 구분될 수 있다. 오용탐지 모델에서는 사전에 침입자의 공격패턴을 모델화하여 일치하는 침입자의 패턴을 탐지하는 방식이다. 컴퓨터 바이러스 백신 프로그램이 알려지지 않은 바이러스를 찾지 못하는 것처럼 오용탐지 모델의 단점은 알려진 공격 패턴만을 탐지할 수 있게 된다는 것이다. 비정상행위 탐지 모델에서는 사용자의 행동과 정상행위 패턴의 변화를 통해 기존의 정상행

위를 벗어나는 침입 행위를 탐지하는 방식이다. 이러한 정상행위 패턴을 생성하기 위해 데이터마이닝 기법을 적용하는 사례가 늘어나고 있으며 기존의 정상적인 값에서 벗어나는 행위를 탐지하는 이탈탐지(Deviation Detection) 또한 보안 시스템에 적용할 수 있는 분야 중 하나이다.

최근 발표된 DEMIDS (DEtection Misuse in Database System)에서는 이러한 데이터마이닝 기법을 적용한 데이터베이스 보안 시스템이다. DEMIDS 에서는 보안 관리자가 지정한 영역(테이블, 속성 등)에 관계된 사용자의 반복적인 행위의 패턴을 추출하며, 객체들간의 관계를 수치로 표현한 스키마 차이도(Schema Distance)와 속성들간의 관련성을 수치로 표현한 속성간 연관성(Access Affinity) 그리고 이를 기반으로 계산되는 차이도 측정치(Distance Measure)를 적용하여 비정상행위를 판정한다.

3. 정상행위 패턴 추출과 비정상행위의 판정

사용자의 정상행위 패턴을 추출하기 위해 일정기간 동안의 로그데이터를 수집하며 그 기간동안의 행위는 모두 정상적인 행위로 간주한다. 수집된 사용자의 SQL 명령은 SQL 명령 추출기를 통하여 패턴 탐색에 사용될 DML 과 DDL 문장을 저장하게 되는데, 이를 통해 저장된 사용자 명령은 일반적인 UNIX 명령과는 다른 면을 보이게 된다. 결론적으로는 같은 명령이지만 사용자에 따라 SQL 명령문에 테이블과 함께 소유자를 입력하는지, 그리고 컬럼명에 테이블명을 함께 입력하는지에 따라 문자열이 상당히 달라질 수 있다. 이는 SQL 구문의 융통성에 따라 다르게 나타나게 된다. 따라서 정상행위 패턴을 추출하기 위해서는 어느 정도의 통일성이 필요하게 되므로 문자열 변환과정을 거치게 된다. 문자열 변환과정에서는 생략된 문장의 보충, 비표준 문법의 표준화 등의 작업이 수행된다. 그러나 이와 같이 변환된 SQL 명령을 곧바로 정상행위 패턴 추출에 사용할 수 없다. 이것은 한 명령 내에 부속절이 한번 이상 존재할 수 있기 때문이다. 따라서 사용자 명령 자체를 그대로 적용하기 보다는 부속절을 분리 함으로서 그 부속절 자체 또한 사용자가 사용했던 명령으로 보는 것이 타당하다. 본 논문에서는 온라인 시스템상에서 빠른 문자열 매치를 통한 비정상행위 판정과 부속절의 적절한 처리, 그리고 정상행위 패턴의 빠른 추출을 위해 패턴트리를 적용한다.

패턴트리의 구조와 적용

패턴트리는 빠른 문자열 매치를 위해 가장 널리 사용되고 있는 Suffix Tree[3]를 적용하였다. Suffix 트리의 생성과정은 그림 1 과 같이 문자열과 매치되는 노드를 탐색하여 더 이상 일치하는 부분이 없을 경우에 말단 노드(Leaf Node) 생성한다. 그러나 SQL 명령은 몇 가지의 구문으로 분리될 수 있기 때문에 패턴트리의 생성을 위해 사용자가 실행한 각각의 SQL 명령을 구문별로 분리하여 각각 Suffix 트리를 생성하였다. 따라서 객체별, 컬럼별, 기타 구문별 정상행위를 효율적으로 추출될 수 있다. 패턴트리의 각각의 노드는 그림 1 과 같이 고유번호 외에 전체세션에서 나타난 횟수와 최근의 접근 시간기록을 유지하도록 되어있다.

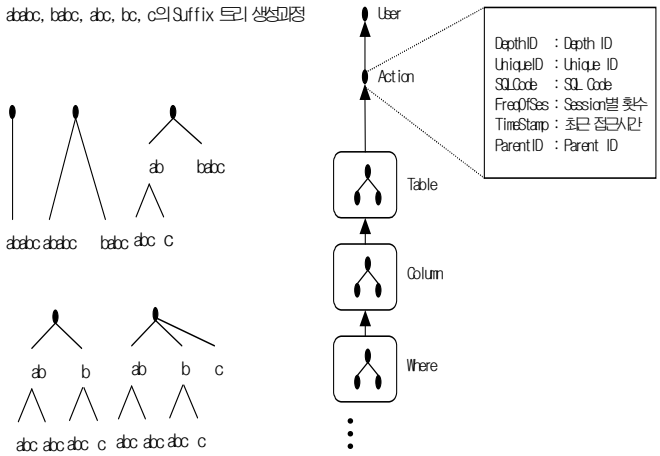


그림 1 Suffix 트리의 생성과정과 패턴트리의 구조

정상행위 패턴 단계에서 적용되는 감쇄율은 최근의 사용자 행위에 가중치를 부여하는 방식이므로 일정기간 이전의 사용자 행위는 정상행위 패턴이 생성될 수 없게 되며 전체 세션에서 임계치 이하의 SQL 명령 또한 정상행위 패턴 생성에 영향을 줄 수 없게 된다. 이와 같이 불필요한 노드를 사전에 제거함으로써 많은 시간을 필요로 하는 정상행위 패턴 생성단계가 효율적으로 수행될 수 있다. 여기에서 구성된 트리의 노드를 연결한 경로를 다시 복원하면 원래의 SQL 명령이 나타나게 된다. 부속절은 분리되어 별개의 명령으로 패턴트리를 구성하게 되며 구분자를 두어 독립된 문장인지 부속절인지를 표현하게 된다. 패턴트리와 각각의 경로를 통해 사용자의 정상행위 패턴을 추출할 수 있으며 수행속도를 위해 각각의 경로는 숫자로 매핑된다.

정상행위 패턴 추출

사용자의 정상행위 패턴은 주기적으로 갱신되어져야 한다. 즉 사용자의 작업이 바뀌었다면 최근의 정상행위 패턴이 생성되어져야 하며 동시에 사용되지 않는 이전의 패턴의 정상행위도는 감소되어져야 한다. 따라서 최근의 사용자 행위에 대해 가중치를 부여하기 위해 감쇄율과 패턴생성 영향률을 적용한다[4]. 감쇄율과 패턴생성 영향률의 자세한 사항은 [4]를 참조한다.

사용자 정상행위 패턴을 생성하기 위해서 본 논문에서는 세가지의 데이터마이닝 기법을 적용한다. 첫번째는 지지도를 이용한 정상행위 패턴 생성 방식이다. 먼저 감쇄율을 적용하여 사용자의 명령이 최소지지도를 만족하는 아이템 크기 1 인 후보키를 생성한다. Apriori 알고리즘[5]을 적용하여 후보키 생성작업을 반복하여 패턴을 생성한다. 이때 생성된 후보키가 최소지지율을 만족하지 않는다면 이 후보키는 제거된다.

그리고 생성된 후보키가 크기가 하나 작은 후보키의 지지도와 같다면 크기가 하나 작은 후보키를 제거한다. 두 번째는 신뢰도를 적용하여 정상행위 패턴을 생성하는 방식이다[6]. 최소 지지도를 만족하는 사용자의 명령을 추출하여 각 신뢰도 계산 후에 관리자가 정의한 임계치를 만족하면 패턴이 생성된다. 세 번째는 순서를 고려한 순차패턴[7]을 적용하는 방식이다. 데이터베이스를 접근하는 방식은 크게 프로그램 접근 방식과 대화식 접근방식으로 나뉠 수 있는데 프로그램에서 접근할 경우 트랜잭션의 명령은 순서적으로 반복되어 나타나는 경우가 대부분이므로 순차패턴을 통해 정상행위 패턴을 추출해야 한다. 순차패턴을 적용하여 정상행위 패턴을 추출하는 과정은 그림 2 와 같다.

Timestamp	SessionID	ProgramID	PER
2000/02/01	1	#0002#0001#0003#0004#0005	2^{-4}
2000/02/01	2	#0003#0004#0005#0002#0002#0001	2^{-4}
2000/02/02	3	#0002#0003#0004	2^{-3}
2000/02/03	4	#0006#0007#0008#0001	2^{-2}
2000/02/03	5	#0004#0005#0003#0002#0001	2^{-2}
2000/02/04	6	#0009#0002#0002#0001#0008#0007	2^{-1}
2000/02/05	7	#0009#0001#0003	2^{-0}

(a) 초기 데이터

ItemSet (1)	Support Rate
#0001	2.125
#0003	2
#0009	1.5

(b) 1 정상행위 패턴

ItemSet (2)	Support Rate
#0009#0001	1.5

(c) 2 정상행위 패턴

최종 정상행위 패턴
#0001, #0003, #0009#0001

(d) 2 정상행위 패턴

그림 2. 감쇄율을 고려한 순차패턴추출 과정

비정상행위의 판정

온라인 시스템에서의 비정상행위 판정은 위와 같은 정상행위 패턴생성 방식에 따라 다르게 적용된다. 첫 번째로 지지도를 사용하여 정상행위 패턴을 생성하였을 경우 각 세션당 수행된 명령의 평균횟수를 고려하여 비정상행위도를 계산하게 된다. 식 1 에서 보는 바와 같이 사용자가 수행한 세션의 명령이 포함된 모든 정상행위 패턴의 비교계산을 통해 임계치 이하의 값이 발생될 경우 비정상행위로 판정하게 된다. 두 번째로 신뢰도를 이용하여 정상행위 패턴을 생성하였을 경우 그림 3 과 같이 사용자가 입력한 명령은 각각의 단계마다 신뢰도의 평균값을 취하게 된다. 최종 단계까지 임계치 이상의 신뢰도를 만족하였을 경우에 각 단계별 신뢰도의 평균값을 정상행위도로 가지게 된다. 세 번째로 순차패턴을 적용하여 정상행위 패턴을 생성하였을 경우 온라인상의 명령이 포함되는 정상행위 패턴의 지지도를 모두 고려하여 정상행위도를 계산한다. 예를 들어 {#0001, #0002, #0003, #0007}의 지지도가 80%이고 {#0005, #0006, #0007}의 지지도가 70%였다면 {#0001, #0002, #0005, #0006, #0007}의 정상행위도는 (80)/(80+70)*(3/5) + (70)/(80+150)*(3/5) = 0.6 로서 60%의 정상행위도를 가지게 된다. 임계치를 만족한다면 결국 정상행위로 판정하게 된다.

Rule	Confidence
#0001->#0002	80
#0001->#0003	70
#0001->#0003#0004	60
#0001->#0003#0002	70
#0003->#0001	70
#0001#0003->#0002	60
#0001#0002->#0001	80
#0001#0003#0002->#0004	80
#0001#0003#0002->#0003	90

#0001#0003#0002#0004의 정상행위도

#0001#0003 : (70 + 60 + 50) / 3
#0001#0003#0002 : (60)
#0001#0003#0002#0004 : (80)

(60 + 60 + 80) / 3 = 68.6

그림 3. 신뢰도를 적용한 비정상행위도 계산

각 방식별 비정상행위도 계산 방식은 다음과 같다.

$$\sum_{i=1}^{n}\left(\frac{S(R_i)}{\sum S(R_i)}\cdot\frac{|R_i\cap T|}{T}\cdot\frac{1}{|R_i|}\cdot\sum_{j=1}^{l}\sum_{k=1}^{m}\left(1-C\cdot\frac{\left\||R_{ij}|-|V_k|\right\|}{|R_{ij}|}\right)\cdot I(j,k)\right)\cdot 100 \qquad \text{식 1}$$

$$\sum_{i=1}^{n}\left(\frac{S(R_i)}{\sum S(R_i)}\cdot\frac{|R_i\cap T|}{T}\right)\cdot 100 \qquad \text{식 2}$$

$$\frac{1}{m}\left(\sum_{i}^{m}\left(\frac{1}{n_i}\sum_{j}^{n_i}C(R_{ij})\right)\right) \qquad \text{식 3}$$

$S(R_i)$: 사용자의 i번째 정상행위 패턴의 지지도
$|R_i|$: i번째 정상행위 패턴의 identified 명령어 총수
R_{ij} : i번째 정상행위 패턴 중에서 j번째 명령어
$|R_{ij}|$: i번째 정상행위 패턴 중에서 j번째 명령어의 평균 사용량
V_k : 판별하고자 하는 패턴(온라인)의 k번째 명령어
$|V_k|$: 판별하고자 하는 패턴(온라인)의 k번째 명령어의 사용 횟수
T : 온라인에서 실행한 명령어의 총 수
C : 실행된 명령어 사용 횟수의 반영 비율
n : 정상행위 패턴의 수
m : 온라인에서 실행된 identified 명령어의 수
$|R_i \cap T|$: 정상행위 패턴과 온라인에서 실행된 명령어와 매칭된 수
$I(j,k)$: if $V_k=R_{ij}|\ (j,k) = 1$, otherwise 0
$C(R_{ij})$: 사용자의 i단계의 j번째 정상행위 패턴의 신뢰도

4. 전체시스템의 구성

전체 시스템의 구성은 그림 4 와 같다. 데이터베이스 로그파일을 추출하기 위해 오라클의 유틸리티 중의 하나인 SQL Trace[8]를 사용하였다. SQL 명령 추출기는 SQL Trace 를 통해 파일 형태로 저장된 각각의 세션을 탐색하여 필요한 정보를 추출한다. 이를 통해 추출된 사용자 ID, 세션 ID, 시간기록, SQL 명령 등은 부속절의 분리를 위해 SQL 파서[9]를 거치게 되며 3 장에서 설명한 문자열 변환 과정을 수행하게 된다. 코드 변환기는 정상행위 패턴을 효과적으로 추출하기 위해 SQL 명령을 지정된 길이의 코드로 변환하는 작

업을 수행한다. 본 논문에서는 이를 위해 각각의 코드를 6 자리의 문자열로 변환하였다. 코드변환을 위해서는 데이터사전을 참조하게 되는데 데이터사전에는 데이터베이스에서 지원하는 함수와 연산자 등의 기본적인 정보와 각각의 명령을 코드로 변환하기 위해 생성한 테이블이 존재한다. 즉 SELECT 명령의 코드번호는 A00010, SELECT TABLE 명령은 C00011, SELECT WHERE 명령은 C00013 등으로 변환된다. 그리고 패턴이 생성될 때마다 갱신되는 사용자 객체명, 칼럼명, 상수 등의 정보도 데이터사전에 저장된다.

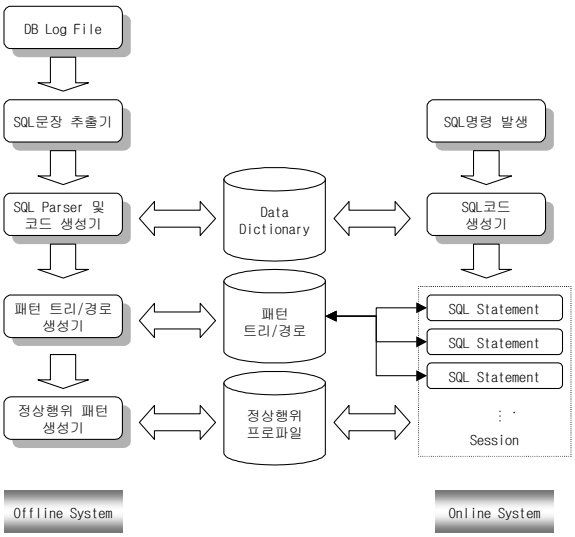


그림 4. 전체 시스템 구성도

온라인 시스템에서는 사용자가 입력한 SQL 명령의 코드변환작업 수행 후 정상행위 프로파일 데이터베이스를 이용하여 비정상행위도를 판정한다. 비정상행위도 판정과정은 두 단계로 나뉘어져 수행된다. 첫 번째는 사용자의 명령을 개별적으로 비교하여 비정상행위를 판정하는 단계이며 두 번째는 사용자의 세션이 종료되었을 경우 세션별 정상행위도를 판정하는 단계이다. 사용자의 명령을 개별적으로 비교하는 단계에서는 SQL 명령의 유사도를 고려하게 된다. 대화식 작업을 수행할 경우 기존의 명령과 약간의 차이가 있다고 해서 전혀 다른 명령으로 처리할 수는 없다. 대화식 환경에서 작업을 할 경우 조금씩 다른 명령을 수행할 확률이 많기 때문에 기존에 생성된 정상행위와의 유사도를 계산하는 단계를 거친다. 정상행위패턴과 온라인 상에서 사용자가 실행한 명령패턴과의 유사도는 패턴트리와 트리의 각 노드를 연결한 경로를 참조하여 수행된다. 유사도는 다음과 같은 단계로 수행된다. 첫 번째 단계는 패턴트리를 탐색하여 최종적으로 매치되는 노드를 찾는다. 두 번째 단계는 최종 노드로부터의 경로를 추출해낸다. 추출된 경로는 이전의 정상행위와 일치되지 않는 명령의 일부분이 된다. 마지막으로 각각의 경로마다 식 4 와 같이 유사도를 계산하여 최대값을 취한다

예를 들어 기존의 정상행위 패턴에 SELECT COLUMN1, COLUMN2 FROM MYTABLE 이 있을 경우 SELECT COLUMN1 FROM MYTABLE WHERE COLUMN1 = 2 와 같은 문장의 유사도는 100 – ((0/1)*33) + (1/2)*33 + (3/3) * 33) = 50.5 가 되어 전혀 다른 문장으로 인식하게 된다.

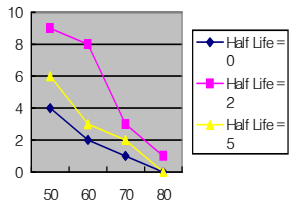
$$\frac{1}{m}\sum_{i=1}^{m}\frac{P_i \cap T_i}{P_i \cup T_i} \qquad \text{식 4}$$

m : SQL명령의 구문 개수
P_i : 온라인 명령의 i번째 구문의 코드수
T_i : 정상행위 패턴의 i번째 구문의 코드수

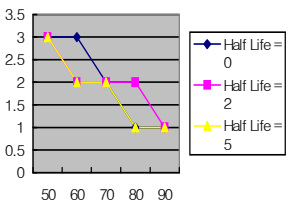
위와 같이 각각의 SQL 명령 비교가 수행된 후 세션이 종료되면 3 장에서 설명한 바와 같이 세션단위로 비정상행위를 판정하게 된다.

5. 실험 및 성능평가

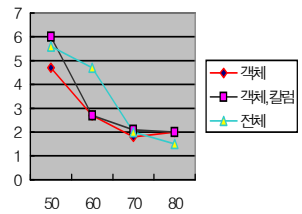
연구실험을 위해 유닉스 운영체제 기반의 오라클 데이터베이스를 사용하여 1 개월 동안 로그 데이터를 수집하였다. 프로그램환경과 대화식 환경으로 구분하여 정상행위 패턴을 생성하였으며 객체(테이블, 뷰)만을 고려한 정상행위 패턴과 객체와 컬럼을 고려한 정상행위 패턴, 그리고 SQL 명령 전체를 고려한 정상행위 패턴을 생성하여 비교하였다. 프로그램 접근방식의 경우 세션마다 각각 다른 행위가 나타나기 때문에 하루를 기준으로 세션을 설정하였다. 연구실험을 위해 프로그램 접근 방식의 경우 공지사항과 게시판 등의 기능이 구현된 교내 행정 시스템의 로그데이터를 사용하였으며 대화식 접근 방식을 위해서는 본 시스템의 구현시 생성된 데이터를 사용하였다. 실험을 위해 Half-Life 는 각각 2 일과 5 일로 정하였으며 0 의 값은 감쇄율을 적용하지 않았음을 뜻한다.



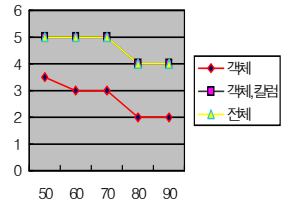
(a) Half-Life 에 따른 정상행위 패턴 수(프로그램)



(b) Half-Life 에 따른 정상행위 패턴 수(대화식)



(c) 정상행위 패턴 평균 크기(프로그램)



(d) 정상행위 패턴 평균 크기(대화식)

그림 5. 정상행위 패턴 생성결과

그림 5 의 (a)와 (c)는 데이터베이스를 접근하는 프로그램 환경에서 정상행위 패턴을 추출한 결과이다. 실험결과 객체만을 고려한 정상행위 패턴의 수와 객체, 컬럼을 고려한 정상행위 패턴의 수는 같았으나 각 패턴의 크기는 다르게 나타났다. SQL 명령 전체를 고려하였을 경우 정상행위 패턴의 수는 늘어났으며 패턴의 크기는 증가하였다. (b)와 (d)는 대화식 환경에서 데이터베이스를 접근하는 방식의 정상행위 패턴을 추출한 결과이다. 프로그램 환경과는 달리 객체만을 고려하였을 경우 정상행위 패턴의 수가 많았으며 객체와 컬럼, 전체를 고려하는 순으로 패턴의 수가 적어졌다.

결과적으로 대화식 접근에서는 정상행위 패턴의 수가 상대적으로 적었으나 객체와 컬럼만을 고려하였을 경우 사용자의 작업을 잘 나타내었으며 프로그램 접근 방식의 경우 각 세션마다의 행위가 정상행위 패턴으로 생성되었다.

6. 결론 및 향후 연구과제

본 논문에서는 데이터베이스 사용자의 정상행위 패턴을 추출 함으로서 불법침입자뿐 아니라 내부의 권한오용 사용자의 비정상행위를 판정하는 방안을 제시하였다. 효율적인 정상행위 패턴생성을 위해 패턴 트리 알고리즘을 제안하였으며 프로그램 환경과 대화식 환경의 차이점을 감안하여 지지도와 신뢰도를 고려한 정상행위 패턴 추출방식과 함께 순서를 고려한 순차패턴 추출방식을 적용하였다.

향후 연구과제로는 최적의 정상행위 패턴 추출을 위한 다양한 환경에서의 실험과 이를 기반으로 한 적절한 임계치 값의 정의가 요구된다. 또한 장기간 연결이 이루어지는 환경에서의 정상행위 패턴 추출방안이 요구된다.

참고문헌

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

[2] C. Chung, M. Gertz, K. Levitt. “DEMIDS: A Misuse Detection System for Database Systems,” IFIP WG11.5 1999.

[3] You-Wu Huang, Philip S. Yu, “Adaptive Query Processing for Time-Series Data,” KDD-99, ACM August 1999.

[4] 윤정혁, 오상현, 이원석, “사용자 명령어 추적을 통한 정상행위 패턴 탐사”, 한국 정보 처리 학회 12 회 추계 학술 대회 논문집

[5] Ming-Syan Chen, Jiawei Han, Philip S. Yu, “DataMining: An Overview from Database Perspective,”

[6] Rakesh Agrawal, Ramakrishnan Srikant, “Fast Algorithms for Mining Association Rules,” In Proc. Of the 20th VLDB conference, 1994.

[7] Rakesh Agrawal, Ramakrishnan Srikant, “Mining Sequential Patterns,” Research Report

[8] Oracle Corporation, Oracle7 Server Tuning Appendix A, SQL Trace Utility

[9] JavaCC(Java Parser Generator) Online Documentation, http://www.metamata.com/JavaCC/docs/index.html