제19회 한국정보처리학회 춘계학술발표대회 논문집 제10권 제1호 (2003. 5)

PrefixSpan 알고리즘을 이용한 침입 탐지 방법

박재철*, 이승용**, 김민수**, 노봉남**
*전남대학교 정보통신협동과정
**전남대학교 정보보호협동과정
e-mail:chori@lsrc.chonnam.ac.kr

An Intrusion Detection Method using the PrefixSpan Algorithm

Jae-Chul Park*, Seung-Yong Lee**,
Min-Soo Kim**, Bong-Nam Noh**
*Dept of Interdisciplinary Program of Information Communication, Chon-Nam University
**Dept of Interdisciplinary Program of Information Security, Chon-Nam University

요 약

알려진 공격 방법에 대해서는 다양한 방법으로 공격을 탐지하여 적절한 대응을 할 수 있는 반면, 알려지지 않은 방법에 의한 공격은 침입탐지 시스템에서 공격 자체를 인식하지 못하므로 적절한 대응을 할 수 없게 된다. 따라서 비정상행위에 대한 탐지를 위해 데이터마이닝 기술을 이용하여 새로운 유형의 공격을 추출하고자 하였다. 특히 대용량의 데이터에 공통적으로 나타나는 순차적인 패턴을 찾는 순차분석 기법 중 PrefixSpan 알고리즘을 적용하여 비정상 행위 공격을 탐지할 수 있는 방법을 제시하였다.

1. 서론

현재 많은 침입탐지 시스템이 상용화되어 판매되고 있는데, 대부분 오용행위를 탐지하기 위한 제품으로 주로 알려진 공격에 대한 데이터를 기반으로 침입을 탐지하고 하고 있는 실정이다. 새로운 유형의 공격이 발생할 경우 침입탐지 시스템 제공자 측에서 그 공격에 대한 탐지 규칙을 분석, 시스템을 수정하여 새로운 공격형태를 탐지할 수 있도록 시스템 또는 데이터를 다시 배포해야 하는 기술적 어려움을 가지고 있다.

비정상행위를 탐지하기 위한 데이터마이닝 기술로는 군집화(clustering), 분류(classification), 순차분석(sequential analysis), 연관성(association)을 이용한 정상행위 특성 추출 방법 등이 있는데, 본 논문에서는 시스템의 실행 명령어의 순차 패턴을 이용하였다. 제안된 PrefixSpan 기법은 사용자별로 공통적인 명령어 입력 패턴을 알아내고 그것들을 모아 패턴 DB를 구축한 후 특정 사용자가 입력한 명령어 패턴을 가지고 저장된 패턴 DB를 참조하여 비정상행위 여부를 판단하는 방법을 제시한다[1, 2].

본 논문의 구성은 다음과 같다. 2절에서 데이터마이닝을 이용한 정상행위 프로파일링, 즉 정상행위 패턴 DB 구축 방법에 대해 설명하고, 3절에서는 정상행위 패턴DB를 기반으로 비정상행위를 탐지할 수 있는 방법을 제시한다. 4절에서 침입이 있었던 사용자 명령어 로그 데이터를 제안한 탐지 방법에 적용하여 탐지 결과를 평가한다.

2. 정상행위 프로파일링

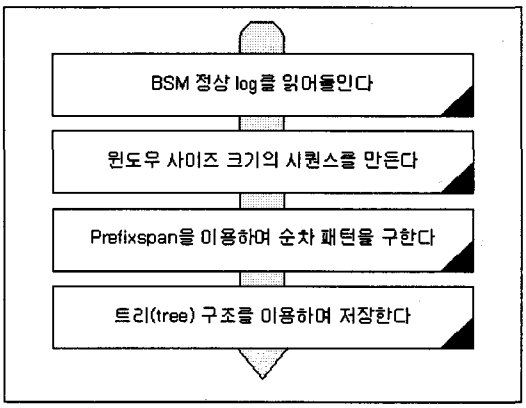
2.1 순차패턴

순차 패턴(sequential pattern)은 연관규칙에 시간 정보가 포함된 형태로, 대표적인 알고리즘에는 GSP 알고리즘[3], FreeSpan 알고리즘, PrefixSpan 알고리즘, SPIRIT 알고리즘[8] 등이 있다.

본 논문에서 적용한 PrefixSpan 알고리즘은 기존의 GSP 알고리즘이 후보패턴을 만들고, 그 후보 패턴이 데이터베이스에서 몇 번 발견되는지 카운터 하면서 시간이 걸리는 단점을 없애기 위해, 후보 패턴을 만들지 않고 빈번한 패턴을 찾는 방법이다. PrefixSpan 알고리즘에서는 PrefixSpan tree를 만들어 가면서 빈번한 패턴을 찾는다[4].

2.2 패턴 DB생성 순서

정상 행위 프로파일링 절차는 그림 1에서와 같이 시스템의 실행명령어 BSM 로그를 읽어 들인 후 각 사용자 별로 특정 윈도우 크기로 나눈다. 윈도우 크기로 나누어진 파일들을 이용하여 PrefixSpan 알고리즘을 이용하여 순차 패턴을 구한다. 이 순차 패턴은 대용량의 데이터이기 때문에 이 정보를 저장하고, 읽고, 검색하는데 많은 시간이 필요하게 된다. 따라서 이 패턴을 나중에 비정상행위 패턴 탐지에 사용되는 패턴 적재, 패턴 검색하는데 걸리는 시간을 짧게 하기 위하여 트리 형태의 구조를 이용하여 저장한다[5, 6].

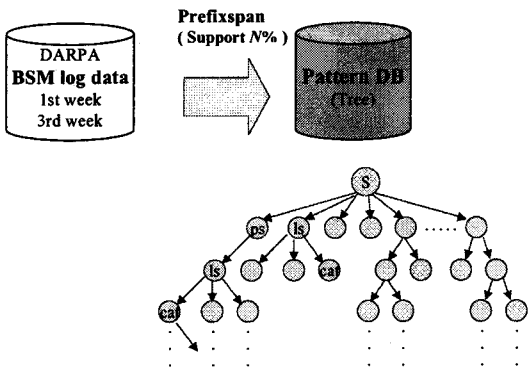


<그림 1> 정상 행위 프로파일링 절차

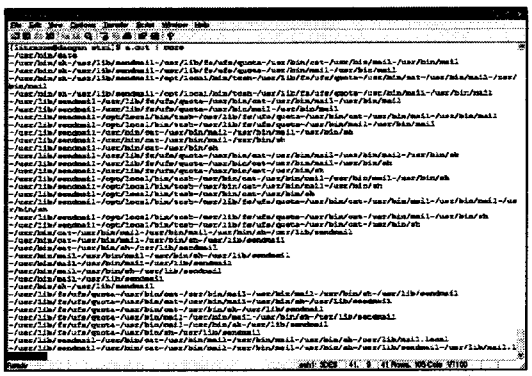
2.3 BSM로그를 이용한 패턴 DB생성

다음은 실제 DARPA에서 가져온 BSM 로그를 이용하여 패턴 DB를 생성하는 과정이다. DARPA의 BSM 로그 데이터에는 프로세스의 사용자 id, 프로세스 id, 시스템 명령어, 시스템 호출 등 프로세스가 실행되는 동안의 모든 정보들이 로그로 기록되어 있다. 공격이 없는 정상행위 로그 데이터를 이용하여 사용자들의 시스템 실행 패턴을 구할 수 있다. 이 패턴을 구하는 방법으로는 많은 양의 데이터로부터 빠른 시간에 패턴을 얻을 수 있는 PrefixSpan 알고리즘을 이용하여 정상적인 명령 실행 패턴을 생성한다. 생성되는 패턴의 개수는 PrefixSpan 알고리즘에서 사용되는 지지도에 좌우된다. 지지도가 높을수록 채택되는 명령어 집합의 개수가 작아지므로 패턴의 가지 수는 적어지고 생성되는 하나의 패턴의 길이도 작아지게 된다. 반대로 지지도가 낮을수록 생성되는 패턴의 가지 수는 많아진다. 이 실행명령 패턴들의 수는 매우 많으므로 차후의 패턴 탐색을 빠르게 하기 위하여 트리형태로 유지된다.

그림 2는 DARPA의 BSM 로그 데이터에서 공격이 없는 정상행위의 1, 3주 로그 데이터를 기반으로 지지도 N%로 PrefixSpan알고리즘을 적용하여 정상행위 패턴을 생성하여 Pattern DB를 생성하는 과정이다. 그리고 그림 3은 프로파일링을 수행하는 과정을 보인 것이다.



<그림 2> PrefixSpan 알고리즘을 이용한 패턴생성



<그림 3> Profiling 실행 화면

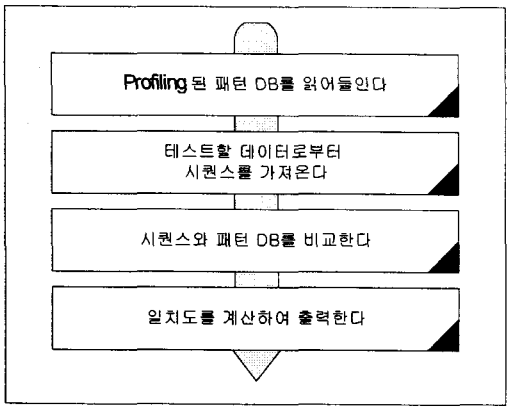
3. 비정상행위 패턴 탐지

비정상행위 패턴 탐지는 일반적인 시스템 사용을 벗어난 행위를 탐색해야 하므로 사용자의 명령 실행에 대하여 패턴 트리를 검색하여 정상행위 정도를 파악하는 것이다. 버퍼 오버플로우 공격, 레이스 컨디션 공격 등은 일반적인 정상행위를 벗어난 시스템 사용이므로 정상행위 패턴에 존재하지 않게 된다. 이와 같은 경우 정상행위 정도는 매우 낮게 나타난다.

3.1 비정상행위 패턴 탐지 과정

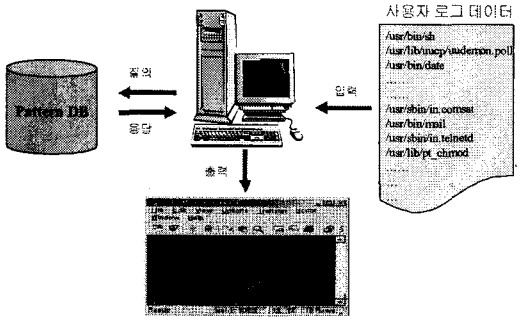
비정상행위를 탐지하는 방법은 침입행위가 없는 일반적인 시스템의 실행 명령어를 기반으로 정상행위를 프로파일링하여 그 정상행위와의 차이로써 판단하게 된다. 사용자뿐만 아니라 시스템에 대한 정상적인 수행 상태를 정상행위로 프로파일링 한다면, 새로운 공격이 이루어졌을 때 그것을 탐지할 수 있는 가능성이 높아지게 된다.

비정상 행위 탐지 절차는 그림 4에서 프로파일링된 패턴 DB를 읽어들인 후 테스트할 데이터로의 시퀀스와 패턴 DB를 비교하고 그 일치도를 계산하여 출력하게 된다.



<그림 4> 비정상행위 패턴탐지 수행 절차

3.2 비정상행위 패턴 탐지 알고리즘



<그림 0> 사용자 입력에 대한 정상행위 판단

사용자의 실행명령에 대하여 일정크기의 개수만큼 큐에 저장되면 정상행위 정도를 구하기 위하여 다음과 같은 방법을 사용한 실행 명령어들이 완전히 정상행위 패턴에 존재하지 않을 수 있으므로 순차적으로 일치하는 일부 패턴까지만 일치하는 것으로 간주하고 이 일치하는 패턴에 대해서는 일치도(consistency)를 구한다. 이 일치도는 실행명령 패턴이 저장된 패턴과 완전 일치하는 경우 1이고, 불일치하는 경우 0으로 정했다. 부분적으로 일치하는 패턴의 일치도는 순차적으로 부분 일치하는 패턴의 길이를 부분 일치하는 패턴을 포함하는 패턴의 최대 길이로 나눈 값으로 정했다. 따라서 부분 일치하는 패턴의 일치도는 0보다 크고 1보다 작은 값을 갖게 된다. 이 일치도는 부분적으로 일치하는 패턴의 길이가 클수록 이 패턴을 포함하는 패턴의 최대 길이와 같아지므로 1에 가까운 값을 가지게 되며 부분 일치 패턴의 길이가 작을수록 이 패턴을 포함하는 패턴의 최대 길이가 커지므로 0에 가까운 값을 갖게 된다. 각 실행 명령에 대하여 일치도를 구하여 각 명령의 일치도의 평균을 구하여 정상행위 정도를 구할 수 있다. 수식 1은 N개의 실행 명령 중에서 i번째 실행 명령부터 j번째 실행 명령까지 실행 명령들이 정상행위 패턴과 순차적으로 얼마나 일치하는가를 구하는 식이다. 수식 2는 i번째 실행 명령의 일치도를 구하는 값으로서 기존에 구한 일치도 값이 새로운 패턴(sliding한 후 패턴)을 가지고 구한 일치도 값에 비하여 작을 경우 i번째 일치도 값을 새로운 패턴을 가지고 구한 일치도 값으로 변경시키는 식이다. 수식 3은 N개의 명령에 대하여 각 명령의 일치도 값의 평균을 구한 값으로 N개의 전체 명령에 대하여 정상행위 패턴과 일치 정도를 나타내는 값이다.

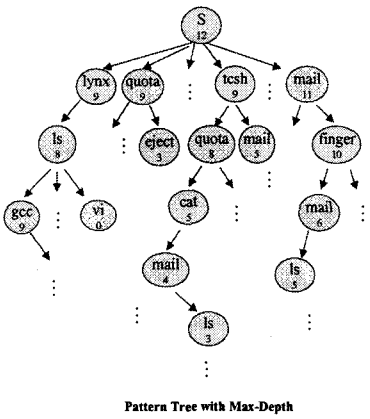
$$C_{ij} = \frac{\partial y\, Consistent\ Pattern\ Len}{Max\ Len\ of\ \partial y\, Consistent\ Pattern} = \frac{j-i}{j-i+Max\ Depth\ of\ j-Node},\ i<j \qquad [수식1]$$

$$C_i = \begin{cases} C_i & , C_i > C_{ij} \\ C_{ij} & , C_j \le C_{ij} \end{cases} \qquad [수식2]$$

$$C = \frac{\sum_{k=1}^{N} C_k}{N} \qquad [수식3]$$

3.3 탐지방법

command	consistency
lynx	0.2
ls	0.2
tcsh	0.5
quota	0.5
cat	0.5
mail	0.5
gcc	0.3
tcsh	0.2
quota	0.2
eject	0.0
mail	0.7
finger	0.7
mail	0.7
sh	0.7
sendmail	0.7
tcsh	0.7
mail	0.7



<그림 6> 정상행위 패턴과 비교

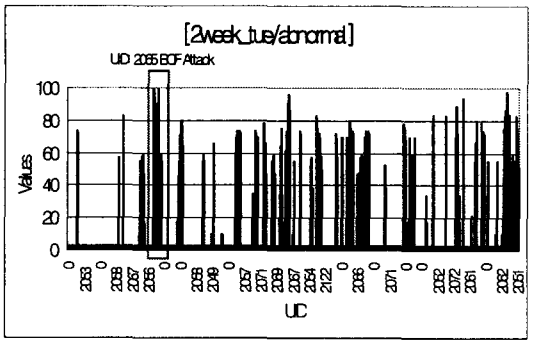
순차 패턴을 이용한 본 실험에서 사용한 데이터는 1999년 DARPA 침입패턴 평가 데이터 집합(DARPA Intrusion Detection Evaluation Dataset)이다. 정상행위 프로파일링을 위하여 공격행위가 전혀 포함되어 있지 않은 1번째 주와 3번째 주의 로그 데이터를 이용하였으며, 실험을 위하여 일반적인 정상행위에 공격이 일부 포함되어 있는 2번째, 4번째, 5번째 주의 로그를 사용하였다.

(1) 정상행위 로그 데이터에 대한 실험
DARPA BSM 데이터 중 정상 데이터인 1주와 3주 데이터로 학습(지지도:70, 윈도우 크기:10)하여 정상 패턴 DB를 생성하였다.

(2) 공격행위가 포함된 로그 데이터에 대한 실험 및 결과 분석

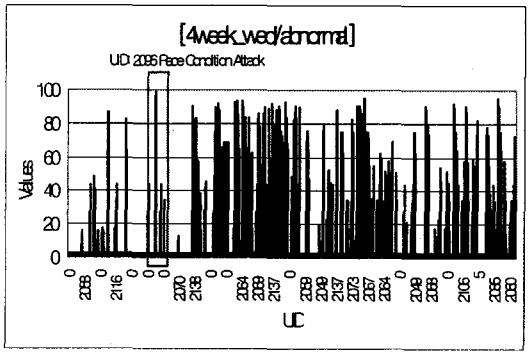
DARPA BSM 데이터 중 1번째 주와 3번째 주 로그 데이터를 이용해 학습(지지도:70, 윈도우 크기:10)하여 생성한 정상패턴 DB를 바탕으로 하여 공격을 탐지한 결과이다. 공격이 포함되어 있는 2번째 주와 4번째 주, 그리고 5번째 주의 로그 데이터 중에서 대표적으로 2번째 주 화요일 데이터를 가지고 실험하였다. 공격을 포함한 데이터에 대한 탐지수행이므로 비정상값(=100)으로 판단되는 부분에 실제로 공격이 포함되어 있음을 확인할 수 있다.

아래 그림 7은 UID가 2136인 사용자가 1999년 3월 11일 목요일 13시 24분경에 버퍼 오버플로우 공격을 시도한 것을 탐지한 것을 보여주고 있다.



<그림 7> BOF 공격 탐지 결과

그림 8은 UID가 2096인 사용자가 1999년 4월 1일 목요일 00시 39분경에 레이스 컨디션 공격을 시도한 것을 탐지한 것을 보여주고 있다.



<그림 8> Race Condition 공격 탐지 결과

4. 결론

데이터마이닝은 데이터베이스에서 발견된 지식으로서, 데이터베이스 연구를 위한 가능성 있는 새로운 영역으로 인식되어지고 있다.

본 논문에서는 비정상행위를 탐지하기 위하여 정상행위 명령 로그를 바탕으로 데이터마이닝 기법 중의 하나인 PrefixSpan 알고리즘을 적용하여 정상행위 패턴을 구하고, 사용자의 실행 명령 순서에 정상행위 패턴과 순차적으로 일치하는 정도를 검사함으로써 정상행위를 체크하였다.

PrefixSpan 알고리즘은 후보패턴을 만들지 않으면서 빈번한 패턴을 찾는 방법으로 데이터베이스의 양을 줄일 수 있었다.

본 논문에서 제시한 탐지 알고리즘을 적용하였을 경우 새로운 유형의 공격에 대비할 수 있게 된다. 그러나 결과 분석에서 볼 수 있듯이 상당히 많은 과탐지(false positive)가 존재하며 향후 윈도우 크기 조절, 사용자 입력에 대한 입력 패턴의 길이 조정을 하여 이러한 부분을 줄이는 연구가 필요하다.

참고문헌

[1] H. Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann, 2001.

[2] Jiawei Han etc, "Data mining", Morgan Kaufmann, 2000.

[3] 이정원 외 13명, "데이터마이닝 알고리즘 분석", 이화여대, 2000.

[4] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, etc, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proc. 2001 Int. Conf. Data Engineering(ICDE '01), 2001.

[5] Ramakrishnan Srikant, Rakesh Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements", Proc. 5th Int. Conf. Extending Database Technology, EDBT, 1996.

[6] Ramakrishnan etc 3, "Mining sequential Patterns : Generalizations and Performance Improvements", IBM, 1996.

[7] Minos garofalakis etc 2, "Mining Sequential Patterns with Regular Expression Constraints, IEEE, 2000.

[8] Minos N. Garofalakis, etc 3, "SPIRIT: Sequential Pattern Mining with Regular Expression Constraints", VLDB, 1999.