제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

침입탐지 감사자료 분석을 위한 연관규칙 생성 기술

소 진, 이상훈
국방대학교 전산정보학과
e-mail:jso@kndu.ac.kr, hoony@kndu.ac.kr

Generating Technology of the Association Rule for Analysis of Audit Data on Intrusion Detection

Jin Soh, Sang-Hoon Lee
Dept. of Computer Science, Korea National Defense University

요 약

최근 대규모 네트워크 데이터에 대한 패턴을 분석하기 위한 연구에 대하여 관심을 가지고 침입탐지 시스템을 개선하기위해 노력하고 있다. 특히, 이러한 광범위한 네트워크 데이터 중에서 침입을 목적으로 하는 데이터에 대한 탐지 능력을 개선하기 위해 먼저, 광범위한 침입항목들에 대한 탐지 적용기술을 학습하고, 그 다음에 데이터 마이닝 기법을 이용하여 침입패턴 인식능력 및 새로운 패턴을 빠르게 인지하는 적용기술을 제안하고자 한다. 침입 패턴인식을 위해 각 네트워크에 돌아다니는 관련된 패킷 정보와 호스트 세션에 기록되어진 자료를 필터링하고, 각종 로그 화일을 추출하는 프로그램들을 활용하여 침입과 일반적인 행동들을 분류하여 규칙들을 생성하였으며, 생성된 새로운 규칙과 학습된 자료를 바탕으로 침입탐지 모델을 제안하였다. 마이닝 기법으로는 학습된 항목들에 대한 연관 규칙을 찾기 위한 연역적 알고리즘을 이용하여 규칙을 생성한 사례를 보고한다. 또한, 추출 분석된 자료는 리눅스 기반의 환경 하에서 다양하게 모아진 네트워크 로그파일들을 분석하여 제안한 방법에 따라 적용한 산출물이다.

1. 서 론

최근 컴퓨터 및 네트워크의 환경변화에 따라 기존에 운영되고 있는 침입탐지 시스템은 새로운 침입유형에 대한 탐지방법을 개선할 필요성이 요구되고 있다[1]. 침입탐지능력을 향상시키기 위한 방법으로 대량의 분산된 데이터 집합을 재사용하는 데이터 마이닝 기법중 연관규칙을 적용한 침입탐지모델을 제안한다. 데이터 마이닝 기법 중 대량의 데이터들의 상관관계를 처리하여 불규칙성속에서 새로운 특징을 추출하는 연관 규칙 기법 및 여러 가지 기법들을 소개한다. 특히, 네트워크 데이터에서 공통적으로 관측된 규칙 집합들을 생성함으로서 보이지 않는 공격에 대한 탐지를 예측할 수 있게 한다[2]. 이렇게 예측을 위한 훈련된 학습규칙들과 새로운 특징패턴들을 기반으로 침입탐지시스템(이하 "IDS"로 칭함)을 지원함으로서 대량의 트래픽 데이터 속에서 탐지속도를 증가시키고 새로운 침입유형에 대한 대응능력을 향상시킬 수 있는 방법론을 제시한다.

본 연구의 구성은 다음과 같다. 2 장에서는 데이터 마이닝 기술과 마이닝에 필요한 침입탐지 데이터의 중요 구성요소들을 살펴본다. 3 장에서는 여러가지 데이터마이닝 기법들을 간략하게 설명하고 그러한 기법들중 연관규칙과 알고리즘에 대해 살펴보고, 4 장에서는 연관규칙을 이용하여 규칙을 생성하는 방법과 사례를 제시한다. 5 장에서는 결론과 미래 연구 방향에 대해 제시한다.

2. 데이터 마인닝 기술과 침입탐지 감사자료

2.1 데이터 마이닝 기술

데이터 마이닝이란 광대한 데이터로부터 기존에 알려지지 않았거나(unknown), 실행 가능한 정보(actionable information)를 추출하여 데이터 속에 내

<표 2-1> 모델링 대상과 데이터마이닝 기술[3]

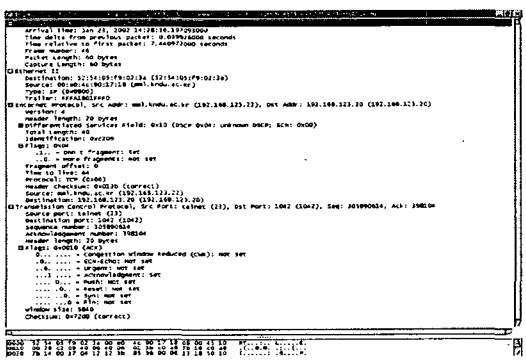
모델링 대상	데이터 마이닝 기술	
	Supervised	Unsupervised
Prediction(예측)	- Regression - Logistic regression - Neural Networks - Decision Trees	실행 불가능함(Not Feasible)
Classification (분류)	- Decision Trees - Neural Networks - Discriminant Analysis	- Clustering (K-means, etc) - Neural Networks - Kohonen Networks - Self Organizing Maps
Exploration	- Decision Trees	- Principal Components - Clustering (K-means, etc)
Affinity (유사성, 밀접한관계)		- Associations Rules - Sequences - Factor Analysis

재된 의미있는 상관관계, 패턴, 경향 등을 찾아내는 일련의 프로세스다. 여기에는 통계(Statistics) 및 수학적(mathematical)기법 뿐만 아니라 신경망(Neural Networks)등을 비롯한 여러 가지 패턴인식 기법이 사용된다. <표-1>에는 데이터 마이닝에 적용되는 대상과 알고리즘에 대해 정리하였다.

2.2 침입탐지 감사자료

일반적으로 침입탐지를 위한 데이터는 패킷 필터링을 통하여 전처리(Pre-processing)된 자료로서 원시 자료(raw data)보는 정제된, 인지 가능한 정보로서의 데이터를 학습시키게 된다. 따라서, 수많은 패킷 데이터에서 추출되는 데이터는 반복적이고, 특징적인 요소를 인지하여 선택하게 된다. 다음은 실질적인 데이터 마이닝 기법에 적용되는 패킷 데이터를 살펴보면 다음과 같다[4]. <그림2-2>는 패킷에서 추출한 TCP 헤더정보를 보여준다.

1) 일반 서비스 종류 (telnet,ftp,smtp,http,ping etc)
2) 프로토콜 : TCP, UDP, IP, ICMP
3) Timestamp, duration, 호스트 IP 및 포트 번호
4) 원천지로부터 받은 byte 수 (src_bytes)
5) 목적지로 보낸 byte 수(dst_bytes)
6) flag : ACK PSH FIN SYN URG RST
7) 주어진 시간에 목적지 IP 횟수
8) 원천지에서 목적지로 접근한 시간차이
9) 목적지 포트 번호
10) 원천지 IP, 목적지 IP에 의한 목적지포트 횟수



<그림 2-2> TCPDUMP에 추출한 TCP 헤더 정보

데이터 마이닝을 이용한 침입탐지 시스템의 주요기능은 TCP/IP 트래픽 데이터 분석 및 식별된 위협들을 시스템관리자에게 통보하거나, 불법 행위에 대한 데이터를 탐지하는 일반 IDS와 동일하며, 일반적으로 모든 탐지기능 능력은 오용 탐지와 변칙 탐지로 나누어 한다. 주요 공격 탐지 능력은 다음과 같다.

1) host-based 공격 : teardrop, ping of death, bonk, oob, Winnuke 등

2) Network-based 공격 : 포트 및 동일 호스트 공격 스캔 , Ping 스캔, Syn Flood , ICMP Flood 등

- IP Spoofing : half open connections
- rlogin : 원격 로인 공격시 시스템 자원을 획득
- Network Scanning : 취약한 포트를 결정, 제한된 포트보다 높게 설정하여 침입 결정
- Network Hopping : 침입자의 추적 흔적을 제거함. 시스템을 접근을 획득하기 위한 원초적인 취약점을 찾는 것은 어렵기 때문에 로인한 후 행동하는 행위에 따라 탐지 기능 수행.

3. 연관규칙과 개선된 방법

3.1 연관 규칙(Association Rules)[3]

속성 집합(A), A의 실제 값의 집합(Γ), Γ 의 부분집합을 항목집합이라 한다. 항목집합의 개수는 항목집합의 길이(k)를 나타낸다. n개의 속성을 가진 데이터베이스를 D라 하면, D에서 항목집합 X를 포함하는 트랜잭션의 비율을 $support(X)$라 정의하고, 연관 규칙은 " $X \rightarrow Y, c, s$ "라고 표현한다. 여기서 X와 Y는 항목집합이며, $X \cap Y = \phi$, $s = support(X \cup Y)$은 지원규칙(support rule)이고, $c = \frac{support(X \cup Y)}{support(X)}$ 은 신뢰규칙(confidence)이라 한다. 보통 감사 자료는 대량의 값을 가진 다중적 특징 속성들(columns)을 다루고 있기 때문에 항목집합의 자료구조는 "열 벡터(row vector)" 구조를 갖는다. 길이가 k인 항목집합 c_k은 두 개의 길이가 k-1인 빈번한 항목 집합 l^1_{k-1}과 l^2_{k-1}을 조인함으로써 생성된다. c_k의 열 벡터는 간단하게 항목 집합 l^1_{k-1}과 l^2_{k-1}의 열 벡터의 AND 연산이다. c_k의 지원율은 c_k의 열벡터에서 쉽게 첫단계에서 계산되어 얻어진다. 데이터베이스 스캔작업은 길이가 k인 빈번한 항목집합의 리스트를 생성하기 위해 오직 한번만 필요하다.

3.1.1 연역적 알고리즘(Apriori Algorithm)

생성된 행동 패턴은 각 항목 레코드 데이터베이스에 저장하여 규칙을 생성하기 위한 항목집합들의 패턴의 수를 찾는다. 이러한 과정에서 지원(Support)규칙의 최소 값과 신뢰(Confidence)규칙을 계산한다.

```
procedure Apriori Algorithm()
  begin
  L1 := {frequent 1-itemsets};
  for ( k := 2; Lk-1 0; k++ ) do {
       Ck= apriori-gen(Lk-1) ; // new candidates
  for all transactions t in the dataset do {
       for all candidates c Ck contained in t do
            c:count++  }
       Lk = { c Ck | c:count >= min-support }
  }
  Answer := k Lk
end
```

<그림 3-1> 연역적 알고리즘(Apriori Algorithm)

즉 빈번하게 발생하는 항목집합들에 대한 행동패턴의 수를 백분율로 나타내는 지원 및 신뢰규칙을 생

성하여 저장한다. 물론, 모든 규칙은 관련된 항목집합끼리 연관시켜야한다. 자세한 알고리즘은 <그림 3-1>에 기술하였다.

3.2 빈번한 에피소드 방법(Frequent episodes)

주어진 데이터베이스 D는 각 트랜잭션은 타임스템프(timestamp)와 관련이 있으며, 시간간격[t_1, t_2] 라 하면, 시작 타임스템프 t_1과 끝 부분 t_2사이의 일련의 트랜잭션을 나타낸다. 시간간격의 폭(width)은 $t_2 - t_1$라 정의한다. 만약 A를 포함하고 A를 포함하는 부분시간간격들이 없다면, D에서 주어진 하나의 항목집합 A에서 시간간격은 A의 최소한의 사건들(minimal occurrences)로 나타낼 수 있다. $support(X)$은 항목집합 X를 포함하는 최소 사건수와 D에서 레코드수의 비율을 나타내다. 빈번한 에피소드 규칙은 " $X, Y \rightarrow Z, c, s, window$" 라고 표현하고, 여기서 X, Y 그리고 Z는 D의 항목집합이다. $s = support(X \cup Y \cup Z)$은 지원규칙이고, $c = \frac{support(X \cup Y \cup Z)}{support(X \cup Y)}$ 은 신뢰규칙이다. 각 사건의 폭은 " $window$ "보다 작아야 하며, 일련의 에피소드 규칙은 추가적인 제한조건을 가지고 있다. 즉, X,Y 그리고 Z는 시간의 순서에 따라 Y 다음에 Z, X 다음에 Y 순으로 트랜잭션이 발생해야 한다. 여기서 중요한 것은 불필요한 속성 값에 대한 연관 규칙에 대해서 새로운 개념을 소개하는 것이다. 즉, 타임 스템프를 이용한 시간간격이라는 특징을 찾아내어 속성의 값에 대한 상관관계를 찾는 방법을 제시하는 것이다.

3.3 개선된 빈번한 에피소드 방법

알고리즘은 연관규칙을 생성하는데 있어 많은 데이터간의 상관관계를 일정한 규칙을 가지고 생성할 때 의미 없이 빈번하다는 이유로 규칙으로 생성되는 문제점을 보완하기 위해 개선하였다.

1) 핵심속성(axis attribute) : 레코드의 속성 중에서 특별한 속성을 설정하여 별로 특별하지 않는 속성과의 상관관계를 고려하는 방식으로 후보항목을 생성하는 동안에 하나의 항목집합은 핵심속성 값을 포함해야한다. 일반 연관 규칙은 " $src\_byte = 200 \rightarrow flag = SF$ "이라 생성할 것이다. 이러한 규칙은 유용하지 못하며, 잘못 판단될 가능성이 높다. 원천지에서 받은 바이트 량과 정상적인 통신상태(flag=SF)은 직관적으로 연관성이 없다. 그러므로, 가장 중요한 연결 상태에서의 정보는 무슨 서비스를 받느냐가 가장 중요한 핵심속성이 되는 것이다. 그러므로, 일반적인 연관 규칙으로 규칙을 생성한다면, 불필요한 속성값을 포함한 일련의 에피소드 규칙을 생성할 수 있다.

예를 들면, " $src\_byte = 200, src\_byte = 200 \rightarrow src\_byte = 200, src\_byte = 200$ " 이와 같이 규칙이 생성되면 일련의 총 규칙은 대량화되고, 필요 없는 규칙만이 증가될 뿐이다. 따라서, 빈번한 에피소드 알고리즘을 사용하는 대신에 핵심속성을 이용한 빈번한 연관 규칙을 제안하고, 이러한 연관성에서 일련의 빈번한 패턴을 생성한다. 다음과 같이 생성될 수 있다.

" $(service = smtp, src\_bytes = 200, dst\_bytes = 300, flag = SF), (service = telnet, flag = SF) \rightarrow (service = http, src\_bytes = 200)$ " 이러한 규칙을 생성할 수 있다. 여기서, 속성들 간의 연관성과 레코드들간의 일련의 패턴들을 조합함으로서 하나의 규칙으로 생성되는 것이다. 이러한 규칙 공식은 감사자료에 대한 유용하고 풍부한 정보를 제공한다.

2) 변수속성(variable attribute) : 일반적인 연관 규칙은 제안적인 규칙을 생산한다. 패턴의 일순위의 성질에 대해 고려한다. 비록 일반적인 일순위 규칙은 완전히 다른 알고리즘을 요구할지도 모르지만, 기본적인 연관 규칙을 확장시킬 수 있다는 이론이다. 일단 웹 싸이트에 방문하여 검색하는 행위에 대해 연구해보면 다음과 같다. 사용자들이 특징적으로 행위하는 명령이나 디렉토리를 대상으로 로그파일을 전처리작업을 한다. 그런 다음, 접근했던 호스트나 명령어, 주로 사용자하는 디렉토리 영역을 가지고 규칙을 생성할수 있다. 예를 들면, $(host = X, request\_dir = a), (host = X, request\_dir = b) \rightarrow (host = X, request\_dir = c)$.여기서, $request_dir$은 동일한 $host$와 연관이 있으며, 실제 $host$ 값은 규칙에서 주어지지 않으며, 변수(Variable)로서 지정하여 사용한다. 실제 값은 유동적이고 어떤 특별한 $host$ 값이 빈번하지 않기 때문에 변수로서 지정한다. 즉, 변수속성의 성격을 이용하여 빈번한 에피소드 알고리즘을 수정 보완하였다. 변수로 사용되지 않은 속성 값에 대한 항목집합이 일련의 레코드들 안에서 변수속성 등을 추가적인 재구성할 때 변수속성의 값은 동일해야 한다.

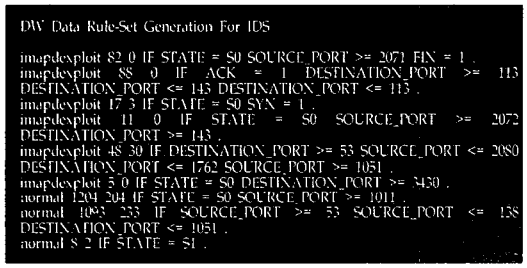
3) 레벨을 이용한 방법 : 때때로 빈도가 낮은 패턴을 찾는 것이 중요할 때가 있다. 예를 들면, 하루에 사용하는 서비스, 예를 들면, gopher는 그 사용빈도가 매우 낮다. 아직까지는 gopher의 패턴을 잡을 필요가 없어 프로파일에 첨가할필요는 없지만, 이처럼 데이터 마이닝에서 매우 낮은 지원규칙 값은 패턴으로 분류하여 사용하게 되면, 불필요하게 높은 빈도를 가진 서비스, 예를 들면, stmp 등과 연관하여 많은 양의 패턴을 저장하게 되므로 패턴을 관리하는데 어렵게 된다. 따라서, 빈번한 에피소드을 찾기위해 중요한 빈도에 따라 패턴을 분류하는 "레벨에 적합한 마이닝 절차"를 제안한다. 예를 들면, $(service = smtp, src\_bytes = 200), (service = smtp, src\_bytes = 200) \rightarrow (service = smtp, src\_bytes = 200)$ 이미 출력된 "old" 핵심값을 특별하게 제한함으로써 빈도수가 낮은 핵심 값과 관련된 에피소드를 찾기 위해 지원규칙 임계값을 반복적으로 낮게 책정하게 되면, 후보

항목집합을 생성할 때 적어도 한번은 "new" 빈도가 낮은 핵심 값을 포함하게 된다. 예를 들면, smtp 가 과거 핵심값을 가진 두 번째 반복단계에서, 얻을 수 있는 에피소드 규칙은 다음과 같다.

$(service=smtp, src\_bytes=200), (service=http, src\_bytes=200) \rightarrow (service=smtp, src\_bytes=200)$ 반복단계를 통해서 얻어진 에피소드는 새로운 핵심 값이든 새로운 것과 과거의 것을 조합한 핵심 값이든 양쪽 모두 공존하게 되어, 매우 낮은 지원규칙 값으로 설정된 상태에서 모든 처리절차가 종료시에는 모든 핵심 값이 가장 빈도가 낮은 것으로 될 수 있다는 것이다.

4. 연관 규칙 생성

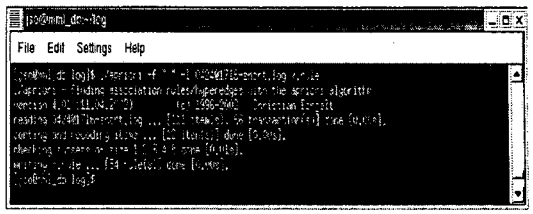
모든 규칙은 패킷형태의 조합으로 나타낸다. 예들 들면, ICMP, TCP, UDP 등의 조합으로 표현될 수 있다. 이는 패킷 크기, 소스와 목적지 포트 개념을 포함한다. 잘 알려진 포트일 경우 계층적 관리에 따라 저장된다. 일반적으로 각 일련의 연속적인 패턴(sequential pattern)의 규칙집합을 구별하는 방법은 그 속성 중에서 가장 식별하기 쉽고 정확하고 정밀한 속성이나 값에 따라 규칙을 설정할 수 있다. <그림4-1>은 침입탐지를 위한 규칙집합을 나타낸다.



<그림 4-1> 침입탐지를 위한 규칙 집합 생성

<그림4-2>과 <그림4-3>에서는 연관 규칙을 이용하여 데이터 로그파일(logfile)을 입력을 받아 연역적 알고리즘에 따라 규칙을 생성(Generating Rules)한 결과를 나타낸다.

<그림 4-2> 사용자 행동패턴 규칙 생성(예: x.rule).



<그림 4-3> 패턴 규칙 프로그램 실행 화면

5. 결론 및 향후 연구

침입탐지 패턴을 분석하기 위해 감사자료를 기반으로 연관 규칙을 적용하여 패킷 분석과 침입 유형에 따른 호스트 기반 탐지 기법들을 대해 연구하였다. 리눅스 시스템 환경 하에서 TCPDUMP를 이용하여 IDS시스템 로그파일을 수집하였으며, 침입에 대한 탐지를 어떻게 하면 실시간으로 빠르게 인식할 수 있는지에 대한 연구로 데이터 마이닝 기법의 연관 규칙을 적용하였다. 또한 새로운 침입유형을 탐지하기 위한 방법과 알고리즘을 제시하였다.

새로운 공격패턴과 예측 불가능한 네트워크 트래픽 패턴에 대한 연구는 자동화 생성을 목적으로 대량의 네트워크 패킷 데이터들 속에서 패턴 식별자와 값을 분석한다. 향후 이러한 데이터들의 속성을 광범위하게 분석하여 학습에 따른 패턴 분류 알고리즘을 연구하고, 이를 통해 새로운 규칙들을 생성하여 침입탐지의 대한 적중율을 높이고, 좀더 빠른 탐지속도 개선을 통하여 시스템 효율성 증대에 대한 연구가 요구된다.

참고문헌

[1] http://www.certcc.or.kr/statistics/hack/2002/hack-200206.html (한국정보보호센터)

[2] Sandeep Kumar, "An Application of Pattern Matching in Intrusion Detection",Technical Report CSD-TR-94-013 Coast TR 94-07, June 17,1994

[3] W. Lee and S. J. Stolfo: "Data mining approaches for intrusion detection." In *In Proceedings of the 1998 USENIX Security Symposium*, 1998.

[4] Wenke Lee, Salvatore J. Stolfo, Kui W. Mok, "A Data Mining Framework for Building Intrusion Detection Models ," IEEE Symposium on Security and Privacy, In http://citeseer.nj.nec.com/154973.html, 1999

[5] http://www.snort.org/

[6] http://www.tcpdump.org/

[7] Moitra, A. Real-time Audit Log Viewer and Analyzer. In Proceedings of the 4th Workshop on Computer Security Incident Handling (Forum of Incident Response and Security Teams - FIRST), August 1992.

[8] Karuna Pande Joshi, "Analysis of Data Mining Algorithms." In http://userpages.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm#apriori, 1997