연관 마이닝 기법을 이용한 침입 탐지 생성 알고리즘 연구

양동수°†, 전태건†, 정동호‡, 김창수†
부경대학교 전자계산학과†, 전산정보학과‡

A Study on the Generation Algorithm of Intrusion Detection using Association Mining Technique

Dong-Soo Yang°†, Tae-Gun Jeon†, Dong-Ho Jung‡, Chang-Soo Kim†
†Dept. of Computer Science, PuKyong Nat'l University
‡Dept. of Computer Science and Information, PuKyong Nat'l University

요 약

본 논문에서는 상태 전이 분석과 연관 마이닝 기법을 이용하여 새로운 침입 탐지 알고리즘인 침입 시나리오 자동 생성 알고리즘(Automated Generation Algorithm of the Penetration Scenarios : AGAPS)을 개발하고자 한다. 침입을 탐지하기 위하여, 먼저 상태 전이 기법을 이용하여 네트워크를 통해 전달된 명령어들에 대한 상태 테이블을 생성한다. 그리고 연관 마이닝 기법을 이용하여 명령어들의 연관 규칙을 발견한 후, 이러한 명령어들이 불법 침입과 관련된 명령어들인지를 판별함으로서 불법 침입 여부를 판단한다.

1. 서론

본 논문에서는 컴퓨터와 정보통신 기술의 발달로 인해 다양한 네트워크 서비스와 시스템을 제공함으로써 인터넷이 보편화되고, 사용자들은 많은 자료를 쉽게 공유할 수 있다. 반면 시스템에 피해를 줄 수 있는 각종 정보와 도구들 역시 쉽게 구할 수 있으며, 또한 이러한 정보와 도구들로 많은 피해가 발생하고 있다.

그러므로 본 연구에서는 데이터 마이닝 기법 중 연관 기법을 침입 탐지 알고리즘에 적용하여 수집된 자료로부터 침입판정 정보를 생성할 수 있는 알고리즘을 개발하고자한다. 침입판정 정보를 위해 상태 전이 기법과 침입 명령의 속성들을 분석하기 위해 연관 기법을 적용한다. 연관 기법은 데이터 집합을 조사하여 데이터 내의 속성들간의 연관성을 발견하는 기법으로써 새로운 정보를 추론하고자할 때 유용하다.

본 논문은 다음과 같이 구성된다. 먼저 2장에서 관련 연구를 기술하고, 3장에서는 침입 탐지 알고리즘과 불법적인 침입에 대한 규칙 생성에 관해서 설명한다. 그리고, 4장에서는 결론에 대해 기술한다.

본 논문은 2000년 대학기초연구지원사업(2000 045-01)에 의해 연구되었음.

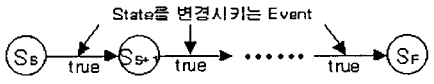
2. 관련 연구

침입 탐지(Intrusion Detection)에 사용되는 기법으로 전문가 시스템, 규칙 기반, 상태 전이 등이 있다.

전문가 시스템은 전문가의 지식을 encode한 규칙의 집합으로 구성되며, 이 규칙은 침입 탐지 시스템에 의해 도출되는 자료를 결론 내리기 위하여 사용된다[1].

규칙 기반(Rule-Based) 탐지 방법은 감사 데이터의 사용 패턴을 표현하고, 저장하기 위해 규칙의 집합들을 사용한다. 이러한 규칙기반 방법을 사용하여 구현된 침입 탐지 시스템으로 TIM(Time-based Inductive Machine)[2] 과 W&S(Wisdom and Sense)[3]등이 있다.

상태 전이 기법은 공격 패턴을 특정 시스템의 상태 전이 순서로 표현한다. <그림 1>은 침입 탐지 분석을 위한 상태 전이도를 나타낸 것이다. 하나의 상태에서 이벤트가 주어지고 일정한 조건을 만족하면 다음 단계로 전이되는 과정을 보여주고 있다.



<그림 1> 상태 전이도

상태 전이 기법을 이용한 침입 탐지 시스템에는 STAT (State Transition Analysis Tool)[4], NetSTAT[8] 등이 있다.

연관 규칙은 특정 트랜잭션에 있는 항목간의 규칙을 표현하는 기법이다[5, 6, 7]. 즉, 어떤 사건(x)이 발생될 때 다음 사건(y)이 발생하는 관련성을 의미한다. 또한 연관 규칙을 찾기 위한 척도인 지지도와 신뢰도를 이용하여 각 아이템간의 연관성을 찾는다. 지지도란 전체 트랜잭션에 대한 x와 y를 만족하는 비율을 의미한다. 그리고 신뢰도란 x를 만족하는 트랜잭션에 대한 y를 만족하는 트랜잭션의 비율을 나타낸다.

주어진 데이터베이스에서 연관 규칙을 찾을 때 다음과 같이 두 단계로 나타낼 수 있다.

① 첫 번째 단계 : 빈발 항목집합들(large itemsets)을 발견한다. 즉, 미리 결정된 최소 지지도인 S_{min} 이상의 지지도를 가지는 항목집합을 찾는다.

② 두 번째 단계 : 빈발 항목집합을 사용하여 최소 신뢰도 C_{min}를 데이터베이스로부터 연관 규칙을 생성한다.

본 논문에서는 연관 규칙 중에서 Apriori 연관 알고리즘을 이용하여 침입에 대한 규칙을 자동적으로 생성한다.

3. 침입 탐지 알고리즘

불법 침입에 대한 새로운 알고리즘을 기술하기 위해 다음 두 가지 단계로 접근하고자 한다. 첫 번째는 일반적으로 많이 알려진 해킹 프로그램에 대해 실제 예제를 중심으로 상태 전이 분석과 연관 기법을 적용한 불법 침입 유형을 분석하는 단계를 설명한다. 두 번째는 다양한 침입 유형이 데이터베이스에 구축되어 있을 때, 이를 적용하기 위한 일반화된 침입 탐지 추론 생성 알고리즘을 기술한다.

3.1 단일 프로그램 기반의 침입 탐지 기법

3.1.1 명령 단위 상태 전이 분석

일반적으로 불법 침입은 시스템의 특정 권한을 얻기 위해서 여러 개의 명령 단위가 필요한 경우가 많다. 그러므로 명령 단위의 상태 전이를 분석하여 침입을 탐지하거나 새로운 유형의 침입을 탐지하는 방법을 제시하고자 한다. 따라서 본 절에서는 불법 침입시 여러 개의 명령어를 필요로 하는 Chup 프로그램의 해킹 방법을 제시하고 이를 탐지하는 방법을 나타내고자 한다.

다음은 Chup 프로그램을 수행하여 루트 권한을 얻기 위한 UNIX 명령을 나타낸 것이다.

```
[S1] id
     uid=100(ssmcl) gid=100 groups=100
[S2] chup
     usage : chup <pid> [uid [euid]]
[S3] csh
[S4] ps
     PID  TTY  TIME  COMMAND
     16302 pts/6  14:20       csh() /* PID-Process ID
                  :
[S5] chup 16302  100  100
     Set to 0  0
[S6] id
     uid=0(root) gid=100 groups=100
```

<그림 2> Chup 프로그램을 이용한 해킹

위의 명령들을 [표 1]과 같이 표현할 수 있다.

[표 1] Chup 프로그램에 의한 명령 상태의 전이도

State / Intrusion	S_1	S_2	S_3	S_4	S_5	S_6
P_1	id	chup	csh	ps	chup <pid> <uid><euid>	id

[표 2]는 앞에서 설명한 Chup 프로그램에 의한 단일 사용자의 각 침입별 명령 유형들의 예들을 나타낸 것이다.

[표 2] Chup 프로그램에 의한 침입별 명령 유형들

State / Intrusion	S_1	S_2	S_3	S_4	S_5	S_6
P[1,1]	id	chup	csh	ps	chup <pid> <uid><euid>	id
P[1,2]	chup	ps	csh	ps		
P[1,3]	id	csh	id	chup 16302 100 100	id	
P[1,4]	ps	csh	chup	chup 16302 100 100	id	

3.1.2 연관 기법에 의한 침입 유형분석

연관 기법의 특징을 이용하여 사용자가 시스템에 접근하기 위해 시도한 명령들 중에서 침입 탐지시 핵심이 되는 중요 명령들을 찾아낸다.

연관 규칙에 적용되는 기호는 다음과 같다.

- I_DB : 명령어 유형별 초기 데이터베이스
- C_Table : I_DB에 대한 명령 대응표
- M_DB : I_DB에서 단계별 분석을 위해 C_Table을 적용한 수정된 데이터베이스
- C_k : k단계에서의 후보 항목 조합
- L_k : k단계에서의 빈도 항목 집합
- T_Min : 침입 판정 추론을 위한 단계별 최소 임계값
- N_Set : 침입 판정의 필수적인 명령 집합들 즉, [표 4]에 대한 N_Set은 {S3, S4, S5} 임.

[표 2]에서 제시된 명령 유형은 크게 4개의 명령 대응표[표 3]로 분류하고, Chup 명령에서 <pid>, <uid>, <euid>에 대한 값은 Chup 한가지 명령 단위로 간주한다.

[표 3] 초기 I_DB에 대한 명령 대응표 (C_Table)

commands	id	chup	csh	ps
corresponding value	A	B	C	D

[표 4]는 초기 I_DB의 각 명령어를 C_Table의 대응값으로 대치하여 M_DB를 생성하였다.

[표 4] 수정된 데이터베이스 (M_DB)

Intrusion \ State	S_1	S_2	S_3	S_4	S_5	S_6
I[1,1]	A	B	C	D	B	A
I[1,2]	B	D	C	D		
I[1,3]	A	C	A	B	A	
I[1,4]	D	C	B	B	A	

<그림 3>는 [표 4]에서 제시된 M_DB를 기준으로 각 단계별 C_k와 L_k를 나타낸 것이고, 빈도 항목 집합을 구하기 위해 T_Min(최소 임계값)은 1로 가정하였다.

항 목	A	B	C	D
빈도수	6	6	4	4

(a) C_1 후보 항목 집합

항 목	A	B	C	D
빈도수	6	6	4	4

(b) L_1 빈도 항목 집합

항 목	{A,B}	{A,C}	{A,D}	{B,C}	{B,D}	{C,D}
빈도수	2	2	1	2	2	2

(c) C_2 후보 항목 집합

항 목	{A,B}	{A,C}	{B,C}	{B,D}	{C,D}
빈도수	2	2	2	2	2

(d) L_2 빈도 항목 집합

항 목	{A,B,C}	{A,B,D}	{A,C,D}	{B,C,D}
빈도수	1	1	1	2

(e) C_3 후보 항목 집합

항 목	{B,C,D}
빈도수	2

(f) L_3 빈도 항목 집합

<그림 3> 각 단계별 C_k와 L_k의 수행 과정

3.1.3 침입 추론 정보 분석

전체적인 불법침입을 추론하는 단계는 다음과 같다.

- 단계 1 : M_DB를 이용하여 최종 후보 항목 집합을 구하면 {B, C, D}가 된다.
- 단계 2 : 최종 후보 항목 집합에 대한 I_DB의 대응값에 대한 명령 유형을 구한다. 즉, B={chup}, C={csh}, D={ps}을 구한다.
- 단계 3 : 단계 2에서 구해진 명령 집합에 대해 [표 3]의 상태 전이도 대응되는 명령의 집합은 S_2={chup}, S_3={csh}, S_4={ps}, S_5={chup <pid> <uid> <euid>}가 된다.
- 단계 4 : 단계 3에서 구한 명령집합과 N_Set 값을 비교함으로써 최종적으로 침입을 판정한다.

3.2 침입 시나리오 자동 생성 알고리즘(Automated Generation Algorithm of the Penetration Scenarios : AGAPS)

본 절에서는 3.1절의 실제 예제 내용에 대한 일반적인 모델로 확장한 알고리즘과 이들의 수행과정에 대해 기술한다.

이미 알려진 다양한 유형의 불법 침입에 대해 수행된 명령어 집합과 각각의 단일 침입(I_i)에 대한 가장 필수적인 명령어의 집합을 [표 5]과 같이 나타내고자 한다.

[표 5] 다양한 유형의 불법 침입 상태표

Intrusion \ State	S_1	···	S_j	S_4
I1	IT(1,1)	···	IT(1,j)	$\{S_{1k}\}\in S$
:	:		:	:
Ii	IT(i,1)	···	IT(i,j)	$\{S_{ik}\}\in S$

불법 침입추론 알고리즘에 사용된 표기는 다음과 같다.

- Scenario : 생성된 침입 시나리오
- CMP : 시나리오와 기존의 침입유형의 비교결과
- Report : 침입 예상 시나리오와 대응하는 침입유형의 집합
- MLV : Minimum Limitation Value(최소 임계치)
- Warning : 침입이라고 간주되는 침입유형

3.2.1 메인 모듈(침입 탐지 모듈)

침입 시나리오는 사용자 Log 테이블의 자료를 Associate() 함수를 호출하여 연관 마이닝 기법에 의해 시나리오를 생성하고, 생성된 시나리오를 기존의 침입유형(IT)와 비교하여 기대치 이상 일치하는 침입유형을 찾아낸다.

```
1 : Algorithm Intrusion-Detection
2 :     Scenario = Associate(LT);
3 :     forall Intrusion i ∈ IT do begin
4 :       CMP = Compare( IT(i), Scenario );
5 :       if( CMP > MIN(E) ) then
            Report = ∪CMP
6 :     end
7 :     Warning = Intrusion_Decision( Report )
8 : end
```

<그림 4> 불법 침입 탐지 알고리즘

3.2.2 연관 상태 생성 모듈

<그림 5>은 연관 규칙을 이용하여 침입 시나리오를 생성하기 위해 연관성이 있는 상태들을 찾아내는 과정이다.

```
1 : Algorithm Associate
2 : L1 = { s | s ∈ LT(i,j) }
3 : for ( k=2; L_{k-1} ≠ ∅; k++ ) do
4 :   C_k = Candidate_Item_Generation( L_{k-1} );
5 :   forall trial t ∈ULT do
6 :     C_t = subset( C_k, t);
7 :     forall candidates c∈C_t do
8 :       c.count++;
9 :   end
10 :  L_k = { c∈C_k | c.count ≥ MIN(sup)}
11 : end
12 : Scenario = ∪_k L_k
```

<그림 5> 연관 규칙 알고리즘

초기 단계에서 하나의 상태를 후보로 하여 각 단계별 상태를 하나씩 추가하여 관련 있는 상태집합을 추출하며, 각 단계에서는 Candidate_Item_Generation()함수를 호출하여 후보 집합을 구하고, 각 후보에 대해 빈도수를 측정하여 연관성이 적은 데이터를 제거하는 과정을 거치게 된다.

3.2.3 후보 항목 집합 생성 모듈

<그림 6>은 이전단계의 빈발 항목 집합으로부터 후보 집합을 생성하는 알고리즘이다. 이전단계의 항목집합 L_{k-1}을 join하고 이 항목의 (k-1)항목의 subset이 이전단계의 항목집합에 포함되지 않는 집단을 삭제함으로써 k-항목의 후보집합을 생성하게 된다.

```
1 : Algorithm Candidate_Item_Generation
2 :   insert into C_k          // Join step, when L is sorted
3 :   select p.item_1, p.item_2, · · · , p.item_{k-1}, q.item_{k-1}
4 :   from L_{k-1} p, L_{k-1} q
5 :   where p.item_1 = q.item_1, · · · ,
            p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1} ;
6     forall item c ∈ C_k do // Prone step: now prune rules
                                with subsets missing in
                                L_{k-1}
7 :     forall (k-1) subsets s of c do
8 :       if ( s ∉ L_{k-1} ) then
9 :         delete c from C_k ;
10:     end
11: end
```

<그림 6> 후보 항목 생성 알고리즘

3.2.4 침입 판정 모듈

<그림 7>는 최종적인 침입 판정을 수행하는 알고리즘이다. 이미 알려진 침입의 형태를 기반으로 추출된 침입유형(Report)과 사용자의 공격유형을 비교하여 침입을 위한 필수적인 명령어들이 포함되었는지를 찾아내고, Warning 값을 생성하게 된다.

```
1 : Algorithm Intrusion_Decision
2 : forall intrusion i∈Report do
3 :   forall trial t∈LT do
4 :     forall state s∈{s∈IT(i,j) | s>minweight} do
5 :         k++;
6 :     if ( LT(t)∈IT(i, s)) then count++;
7 :      if ( count == k ) then
8 :         Warning = ∪Report(i)
9 :        break
10:     end if
11:     if {(count >= MLV) || count <= k)} then
12:     insert LT(t) into IT(i,s)
13:     end if
14: end
```

<그림 7> 침입 판정 알고리즘

4. 결론

본 논문에서는 침입 시나리오 자동 생성 알고리즘(AGAPS)을 제시하였고, 이를 구현하기 위해 상태 전이 기법과 연관 규칙(Apriori Algorithm)을 적용하였다. 상태 전이 기법은 다양한 입력 명령어들에 대한 상태 전이를 나타낸다. 그리고 연관 규칙은 처리 단계에서 Intrusion Table로부터 항목집합을 찾고, Log Table로부터 시나리오를 추출한다. 마지막으로 본 논문에서 제시한 알고리즘은 시나리오 정보와 Intrusion Table의 비교함으로서 침입 여부를 판정하게 된다.

그러나 본 연구에서는 보다 정확한 침입 판정 알고리즘을 제시하기 위해서 수집된 명령어에 대한 임계치 설정 여부와 침입 탐지에 대한 핵심 명령어를 어떻게 선별할 것인가에 대한 연구가 요구된다.

[참고문헌]

[1] T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalai, H.S. Javitz, A. Valdes and P.G. Neumann, "A Real-Time Intrusion Detection Expert System", SRI CSL Technical Report, SRI-CSL-90_05, June 1990.

[2] K. Chen, S.C. Lu and H.S. Teng, "Adaptive Real-Time Anomaly Detection Using Inductively Generated Sequential patterns", presented at the Intrusion Detection Workshop, SRI Internation, Menlo Park, CA, May 1990.

[3] H.S. Vaccaro and G.E. Liepins, "Detection of Anomalous Computer Session Activity", Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp.280-289, May 1989.

[4] P. Porras, "STAT-A State Transition Analysis Tool for Intrusion Detection", Master's thesis, Computer Science Department, University of California, Santa Barbara, June 1992.

[5] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules in large databases", In Proceedings of ACM SIGMOD Conference on Management of Data, Washington D.C., pp.207-216, May 1993.

[6] W.Lee and S.J. Stolfo. Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium, San Antonio, TX, January 1998

[7] H.Toivonen, "Sampling Large Databases for association rules", In proc, of the 22nd VLDB conference, 1996.

[8] G. Vigna and R. Kemmerer, "NetSTAT: A network-based intrusion detection approach", Proceedings of the 14th Annual Computer Security Applications Conference, Scottsdale, Arizona, December 1998.