유전자 알고리즘을 이용한 침임탐지 오토마타의 생성

안영준[0], 위규범
아주대학교 정보통신전문대학원
{darkx[0], kbwee}@ajou.ac.kr

Generation of intrusion detection automata using genetic algorithms

Youngjun Ahn[0], Kyubum Wee
Graduate School of Information and Communication, Ajou University

요 약

비정상 행위와 정상행위를 구별하여 침입을 탐지하는 기법 중 오토마타를 이용해 정상 행위를 프로파일링 하는 기법이 연구되어왔다. 최근엔 다중 서열 정합(multiple sequence alignment)방법을 이용하여 오토마타 생성을 자동화 하는 방법이 소개 되었다. 그러나 이 방법은 시스템 콜의 서열을 정열하기 위해 추가적인 상태가 들어가게 때문에 오토마타가 너무 커지는 단점이 있다. 본 논문에서는 유전자 알고리즘을 이용하여 정상 서열을 인식하는 오토마타를 생성하는 방법을 제안한다.

1. 서론

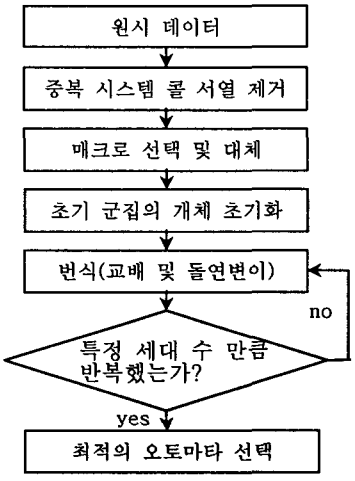
침입탐지 기법 중 하나인 오용 탐지(misuse detection)는 알려진 취약성을 통한 공격에 대한 정보를 가지고 실제적인 공격이 시도될 때 이를 탐지하는 방식이고, 비정상 행위 탐지(anomaly detection)는 정상적인 시스템 사용에 대한 프로파일 상태를 유지하며 이에 어긋나는 행위를 탐지하는 방식이다. 따라서 비정상 행위 탐지를 위해서는 정상 행위에 대한 프로파일링이 선행되어야 한다.

비정상 행위 탐지에 관한 연구 중에 시스템 콜 서열을 이용한 연구가 많이 있어 왔다. 그 중에서 Kosoresow가 제안한 매크로를 이용해 오토마타를 만드는 방법은 일일이 수동으로 생성해야 하기 때문에 어려움이 많았다[3]. 최근에 자동으로 오토마타를 생성하는 연구가 있었는데, 이 방법은 사용자의 개입이 필요하지 않다는 점에 의의가 있지만, 오토마타에 다중서열정합을 위해 많은 상태가 생성되기 때문에 너무 큰 오토마타를 생성하게 된다[1, 2].

본 논문에서는 위의 단점을 보완하기 위해 유전자 알고리즘을 사용하여 효율적인 오토마타를 생성하고자 한다.

2. 시스템 구조 및 알고리즘

원시 데이터의 형태는 프로세스 번호와 시스템 콜이다. 일단 같은 프로세스 번호 별로 시스템 콜을 모아 서열(sequence)을 만들고 이렇게 만들어진 서열 중에서 같은 것을 제거한다. 그 후 서열내의 반복되는 시스템 콜을 매크로로 대체한다. 이렇게 만들어진 서열의 집합이 최종으로 만들어질 오토마타가 승인해야 하는 프로파일이 된다. 유전자 알고리즘을 적용하기 위해 먼저 오토마타의 정보를 교배가 가능하도록 표현해야만 한다. 여기서는 상태와 심볼의 정보를 이차원 배열로 표현했다. 초기 군집을 형성한 후 균일 교배 방식으로 충분히 교배를 시킨 다음 최종으로 만들어진 가장 적합도가 높은 오토마타를 선택한다.



3. 매크로 대치

주어진 시스템 콜을 프로세스 번호로 서열을 만든 후 할 일은 서열에 중복적으로 나타나는 시스템 콜을 매크로로 대치하는 것이다. 이를 위해 [5]에서 제안된 suffix trie를 사용하였다.

Suffix trie에서 루트노드(root node)로부터 각 단말노드(leaf node)까지의 경로는 주어진 문자열의 각 suffix를 나타낸다. 따라서 루트노드로부터 각 중간노드까지의 경로는 각 부분문자열(substring)을 나타낸다.

한 노드는 그 노드가 나타내는 부분문자열이 끝나는 위치를 저장하고 있다. 이를 통해 각 부분문자열이 몇 번 반복됐는지를 알 수 있다. 또한 각 노드의 깊이를 통해 매크로로 대치할 문자열의 길이를 알 수 있다. 이 두 정보를 이용하면 주어진 횟수 이상 및 주어진 길이 이상 나타나는 부분문자열들을 추출하는 것이 가능하다.

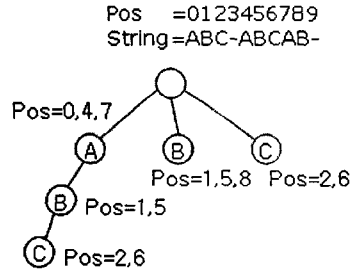


그림1. ABC-ABCAB- 의 Suffix Trie

그림1 은 ABC-ABCAB- 에서 발생 빈도가 2회 이상인 부분문자열들만의 suffix trie를 나타낸다. 여기서 '-'는 구분자로서 매크로가 ABC와 ABCAB를 걸쳐서 선택되지 않게 하기 위해 넣은 것이다.

4. 유전자 알고리즘을 이용한 오토마타 생성기법

유전자 알고리즘(Genetic Algorithm)은 적자 생존과 유전의 메카니즘을 바탕으로 하는 탐색 알고리즘이다. 주어진 환경에 잘 적응하는 유전자만을 선택(selection)하고 교배(crossover)하고 때에 따라서는 돌연변이(mutation)도 하며 다음 세대에 우수한 유전 형질이 전달되게 된다. 따라서 진화(evolution)가 거듭될수록 주어진 환경에 더 적합한 유전자들만이 남아있게 된다. 이를 응용하면 정상 서열을 인식하는 오토마타를 만들 수 있다. 본 연구에 사용된 유전자 알고리즘은 Belz의 알고리즘을 참고하였다[4].

4.1 개체(individual)

유전자 알고리즘을 적용하기 위해 먼저 한 개체를 하나의 오토마타로 보자. 그림2의 오토마타는 표1과 같이 행이 상태고 열이 심볼에 해당하는 2차원의 배열로 표현이 가능하다.

표1. 그림2의 배열식 표현

	a	b	c
S0	S3	S2	S2
S1	-	-	-
S2	-	-	S1
S3	S2	-	-

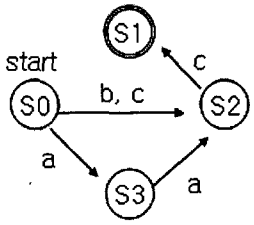


그림2. 오토마튼

표 1의 이차원 배열에서 각 필드가 유전자가 되는 것이고 염색체에 해당하는 것은 2차원 배열의 행을 일열로 나열한 것이라 할 수 있다.

4.2 초기 군집(initial population)

초기 군집에서 개체의 개수는 사용자가 임의로 충분하게 선택한다. 군집의 개체는 격자모양으로 배치하여 주위에 8개의 이웃을 가지게 한다. 이는 나중에 부모를 선택하기 위한 방법이다. 초기화에서 오토마타의 크기는 승인하기 위한 시퀀스의 최대 길이에서[-2~+2]로 설정한다. 한 개체를 초기화 할 때 각 유전자를 절반의 확률로 부인하고 나머지 절반에 대해서 남아있는 오토마타의 상태 중에서 랜덤하게 다음 상태를 고르게 한다(부인한다는 말은 주어진 심볼을 읽었을 때 다음 상태로 갈 수 없다는 의미이다. 예를 들어 표1의 S2상태에서 'a'를 읽었을 때 다음 상태로 진행할 수 없다). 상태에는 표1처럼 순서대로 번호가 있어 자기보다 높은 번호로만 점프하게 하여 싸이클을 만들지 않게 한다.

4.3 적합도

유전자 알고리즘을 오토마타에 적용하기 위해선 다음 세 가지를 주요 적합성 지표로 삼는다.

일관성(consistency) : 주어진 오토마타가 서열을 잘 승인하는지의 정도.

보편성(generalization) : 원하는 서열 이외의 것에 대한 인식 여부.

크기(size) : 오토마타의 상태(state)수.

여기서 제일 중요한 부분은 일관성으로 이 점수가 높을수록 정상 시스템 콜 서열을 잘 인식하게 된다. 그 외

보편성은 낮게 크기는 작은 것이 선택되도록 적합성을 계산한다. 본 알고리즘에는 적합도의 반영비율을 순서대로 0.7, 0.2, 0.1 로 하였다.

4.4 교배(crossover) 와 돌연변이(mutation)

먼저 주어진 군집에서 부모1을 임의로 선택한 다음 부모1 주위의 8개의 이웃 중에 가장 적합도가 높은 것을 선택하여 부모2라 하고 이 둘을 교배하여 자손을 생성한다. 군집의 경계에 있는 개체일 경우 2차원 배열을 도넛 모양(torus)으로 양 끝을 옆에 붙인 상태에서 8개의 후보를 고른다.

교배방식은 균일 교배(Uniform crosso ver)를 사용하였다. 표1에서 나와있는 오토마타를 순서대로 'S3 S2 S2 - - - - - S1 S2 - -' 나열한 것을 염색체라 정의하였다. 두 부모로부터 뽑은 위 염색체를 앞에서부터 순서대로 하나의 형질을 비교하여 교배를 하는데, 이 때 오토마타의 적합도에 따라서 적합도가 큰 것의 유전 형질이 선택될 확률이 높도록 하였다.

자식의 크기는 임의로 선택한 부모1의 크기에서 [-2~+2]사이로 하여 성장과 쇠퇴를 할 수 있도록 한다. 자식의 크기가 두 부모보다 클 경우 나머지 부분은 초기화에서 했던 대로 절반의 확률로 부인을 하고 나머지 절반의 확률로 유효한 상태를 가리키게 한다. 두 부모의 크기가 다를 경우 균일교차를 할 수 없는 부분은 그대로 자식에게 전달된다.

교배를 한 후 자식의 적합도를 구해 적합도가 부모1 보다 크면 부모1을 대치 하는 방식으로 진행한다.

돌연변이는 교배 후 생성된 자식에 대해 약 10%의 비율로 발생한다. 돌연변이가 발생하면 각 유전형질에 대해서 다시 10%의 비율로 상태가 변화한다.

5. 실험 결과

실험을 위해 Forrest 등의 연구에서 사용한 데이터를 가지고 오토마타를 만들고 테스트를 하였다[6]. 표2는 주어진 4개의 프로그램 별 정상서열로 오토마타를 만든 후의 오토마타 탐지율을 나타낸다. 각 경우 초기 군집의 개수는 900으로 하였다.

표2의 TN(true negative)부분에서 보듯이 정상 서열을 100% 인식하지 못하는 경우도 있다. 하지만 이 경우에도 높은 비율로 침입을 탐지 할 수 있었다.

표2. 프로그램 별 탐지율

	synthetic sendmail	synthetic lpr	login	ps
TP	18/18	1001/1001	10/13	26/26
TN	105/147	9/9	16/16	12/24
FP	42/147	0/9	0/16	12/24
FN	0/18	0/1001	3/13	0/26
DR	100%	100%	76%	100%
FPR	28%	0%	0%	50%
OA	75%	100%	90%	76%

TP(True Positive) : 침입을 침입으로 판단
TN(True Negative) : 정상을 정상으로 판단
FP(False Positive) : 정상을 침입으로 판단
FN(False Negative) : 침입을 정상으로 판단
DR(Detection Rate) = TP / (TP+FN)
FPR(False Positive Rate) = FP / (TN+FP)
OA(Overall Accuracy) = (TP+TN)/(TP+TN+FP+FN)

6. 향후 과제

TN(True negative)이 100%에 미치지 못하는 sendmail과 ps의 경우 그 서열들을 살펴보면 다른 두 개의 프로그램과 달리 서열의 길이가 균일하지 않았다. 프로파일링할 데이터의 성격이 오토마타를 진화시키는데 중요한 요소가 됨을 볼 수 있는데 향후 이 부분에 대한 연구를 계속 진행할 것이다.

7. 참고 문헌

[1] 임영환, 위규범, "침입탐지를 위한 유한 상태 기계의 생성기법", 정보처리학회논문지, 제10권 제2호, pp. 119-124, 2003

[2] K. Wee and B. Moon, "Automatic Generation of Finite State Automata for Detecting Intrusions using System Call Sequences", Proceeding of MMM-ACNS, LNCS 2776, pp. 206-216, St. Petersburg, Russia, Sept. 2003.

[3] P. Kosoresow, "Intrusion Detection via System Call Traces", IEEE Software, Vol.14 No.5, pp. 35~42 1997.

[4] A. Belz and B. Eskikaya, "A Genetic Algorithm for Finite State Automata Induction with an Application to Phonotactics", Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing, pp. 9~17, August 1998

[5] J. Vilo, "Discovering Frequent Patterns from Strings,", Department of Computer Science, University of Helsinki, Technical Report C-1998-9, May, 1998.

[6] http://www.cs.unm.edu/~immsec/systemcalls.htm