情報保護學會論文誌
第16卷 第1號, 2006. 2

다중서열정렬의 유사도 매칭을 이용한 순서기반 침입탐지*

김 용 민
여수대학교 정보기술학부

Sequence based Intrusion Detection using Similarity Matching of the Multiple Sequence Alignments*

Yong-Min Kim
Div. of Information Technology, Yeosu National University

요 약

대부분의 침입탐지 방법은 알려진 침입 정보를 축적하고 임의의 행위 데이터에 대해 침입 여부를 결정하는 오용행위 탐지의 방법에 기반하고 있다. 그러나 생성된 공격행위 패턴은 새로운 공격 및 변형된 공격행위에 대응하는 방법에 어려움이 있다. 현실적으로 비정상행위 탐지기법의 높은 오탐을 고려하면, 대용량 순서기반 침입패턴은 알려진 공격에 대한 탐지와 함께 침입패턴의 유사도를 측정하는 방법의 보완을 통해 변형된 공격 및 새로운 공격에 대한 탐지의 가능성을 높이는 대책이 요구된다. 본 논문에서는 순서기반 침입패턴의 유사성 매칭을 위해 다중서열정렬 기법을 적용하는 방법을 제안한다. 그 기법은 침입패턴 서열의 통계적 분석을 가능하게 하고 구현이 용이하며, 서열 크기의 변경에 따라 공격에 대한 탐지 경보 및 오탐의 수를 줄이는 결과를 보였다.

ABSTRACT

The most methods for intrusion detection are based on the misuse detection which accumulates known intrusion information and makes a decision of an attack against any behavior data. However it is very difficult to detect a new or modified attack with only the collected patterns of attack behaviors. Therefore, if considering that the method of anomaly behavior detection actually has a high false detection rate, a new approach is required for very huge intrusion patterns based on sequence. The approach can improve a possibility for intrusion detection of known attacks as well as modified and unknown attacks in addition to the similarity measurement of intrusion patterns. This paper proposes a method which applies the multiple sequence alignments technique to the similarity matching of the sequence based intrusion patterns. It enables the statistical analysis of sequence patterns and can be implemented easily. Also, the method reduces the number of detection alerts and false detection for attacks according to the changes of a sequence size.

Keywords : *Multiple Sequence Alignments, Similarity Matching, Sequence based Intrusion Detection*

Ⅰ. 서 론

대부분의 침입탐지 방법은 알려진 침입 정보를 축적하고, 임의의 행위 집합에 대해 침입 여부를 결정하는 오용행위 탐지의 방법에 기반하고 있으며, 공

접수일 : 2005년 11월 9일 ; 채택일 : 2005년 12월 24일
* 본 연구는 여수대학교 2004년도 학술연구과제지원비에 의하여 연구되었습니다.
† 주저자, 교신저자 : bluearain@yosu.ac.kr

격행위를 전문가가 직접 분석하여 해당하는 공격행위 패턴을 생성하기 때문에 새로운 공격 및 변형된 공격에 대응하는 방법에 어려움이 있다. 현실적으로 비정상행위 탐지기법의 높은 오탐율을 고려할 때 대용량의 공격행위 데이터로부터 유사한 또는 변형된 공격에 대한 침입을 탐지하는 보완적인 방법이 필요하다. 즉, 이미 축적된 다양한 침입패턴으로 이에 대응하는 공격 뿐 만 아니라 침입패턴의 유사도를 비교하는 방법의 개발을 통해 변형된 공격 및 새로운 공격에 대한 탐지의 가능성을 높이는 방법에 대한 대책이 요구된다[1,2].

일반적으로 패턴은 임의 척도들이 교차되어지는 순서들의 집합으로써 정의하며 정규식으로 표현한다. 정규식 패턴을 모델링 하고 패턴들 간에 동일성 및 유사성을 분석하기 위한 연구들은 다양한 영역에서 연구되고 있다[3]. 특히 침입패턴에서 순서열의 중요성은 생명정보학에서 서열(sequence)의 중요성과 그 의미가 유사하다.

본 논문에서는 최근에 다양한 영역의 연구에 적용되어지는 서열정렬(sequence alignment)의 유사성 매칭 기법을 분석하여 순서기반의 침입탐지 방법을 제안하고자 한다. 수 억 개 이상의 DNA 서열에 대한 패턴의 유사성 매칭 기법은 알려진 침입패턴에 대한 유사성 분석과 변형된 공격의 탐지 및 새로운 침입패턴의 재구축으로 향후 알려지지 않은 공격에 대한 침입탐지의 근거를 발견할 메타 패턴의 구축 및 유사성 매칭 방법으로 고려되기 때문이다.

II. 관련연구

2.1 시스템호출 순서기반의 침입탐지

시스템 호출의 순서정보를 프로파일하기 위해 다양한 방법들이 제안되어 왔다[4]. 대표적으로 *Forrest*[5]는 슬라이딩 윈도우 기법을 침입탐지에 사용하였으며, 거리기반을 이용한 유사도 비교를 수행하였다. 그러나 비교연산 수행으로 인한 컴퓨팅 파워의 제약으로 윈도우의 크기의 제한과 단순 거리비교를 통하여 유사도를 판정함으로 탐지된 공격에 대한 분석적 방법을 제공하지 못한다. 분석적 방법을 제공하기 위하여 유한상태 기계의 기법을 이용한 침입탐지의 방법[6]은 시스템 호출의 순서를 매크로로 치환하고, 이러한 매크로 또는 서열을 인식하는 방법을 제안하였다. 유한상태기계의 생성은 자동으로 서열을 생성하고 공격의 분석을 가능하게 하는 방법으로서 의미가 있다. 그러나 정상행위의 프로파일링을 통한 비정상행위의 탐지의 방법으로 오토마타의 생성에 자동화가 미흡하며, 최적의 서열을 만들지 못한다. 또한 면역시스템에 의한 방법[7]은 자기/비자기(self/non-self) 인식의 부정선택 알고리즘을 이용하여 항체의 자동생성에 의한 자기 인식에 좋은 결과를 보였지만 블록단위의 변경에 대한 탐지는 낮은 성능을 보였으며 많은 시스템 호출로 이루어진 경우의 비교에는 제한적이다.

시스템 호출을 이용한 제안된 방법의 다수의 연구들은 정상행위만을 프로파일하여 침입탐지를 시도함으로써 시스템 호출을 이용한 대부분의 공격이 정상행위와 90%이상 유사함을 고려할 때 생성한 침입패턴이 중복되어 오탐의 가능성이 항상 제시되었다[8,9]. 따라서 시스템 호출의 순서를 이용하여 구성한 침입패턴을 이용한 침입탐지 시에 해당 공격의 탐지 및 분석적 방법을 제공하고 변형 및 알려지지 않은 공격의 탐지를 위한 진화의 방법을 도입하는 것이 요구된다.

2.2 다중서열정렬

다중서열정렬은 데이터베이스에서 유사한 서열을 찾는 모델링의 방법으로 일치서열(Consensus Sequence), 패턴(Pattern), PSSM(Position Specific Score Matric), 일반화 프로파일(Generalized Profile), HMM(Hidden Markov Model) 등이 있다[10]. 각 방법의 비교 내용을 표 1에 보였다.

일치서열은 다중서열정렬에서 모델을 만드는 가장 간단한 방법으로 매우 빠르고 쉽게 구현이 가능하다. 그러나 이 모델은 비교 서열의 각 열에 있는 변수들에 대한 정보를 무시하며, 측정(scoring)이 없고 이진 결과만을 알 수 있다. 패턴은 단일 표현으로 선택적인 서열의 집합을 나타낼 수 있으며 빠르고 구현이 쉽다. 그러나 삽입/삭제가 쉽지 않고, 짧은 패턴에서는 과탐지가 발생하고 긴 패턴에서는 정교하게 만들기가 매우 어렵다. 훈련 집합의 서열로만 인식하는 예측에는 한계가 있고 통계적 측정 방법이 아니며 이진 응답만을 나타낸다.

PSSM은 다중서열의 특정 위치에서 각 구성의 잔여 빈도수에 기반하는 방법이다. 짧거나 일정한 영역에서 좋은 성능을 보이며, 상대적으로 빠르고 구현이 쉬우며 일치된 점수(match score)는 통계적 이론을 바탕으로 해석될 수 있다.

표 1. 시스템호출을 이용한 순서기반 침입탐지 방법의 비교

구분		내용
슬라이딩 윈도우		-수행연산이 빠르고 구현 용이 -긴 크기의 패턴을 비교하기 위한 과다한 자원의 사용으로 윈도우 크기 제한 -통계적 분석 방법 제공하지 못함
유한상태 기계 / HMM		-통계적 이론에 기반한 분석 -삽입과 삭제의 지연 연산과 자원의 사용량 과다함 -자동화한 패턴의 생성 미흡
다중 서열 정렬	일치 서열	-수행연산이 빠르고 구현 용이 -서열의 각 요소정보 표현 없음 -측정점수 없음(분석할 수 없음)
	패턴	-수행연산이 빠르고 구현 용이 -학습에 의해 쉽게 만들 수 있음 -작은 크기의 패턴은 오탐 높음 -긴 크기의 패턴은 정확하지 못함 -측정점수 없음(분석할 수 없음)
	PSSM	-수행연산이 빠르고 구현 용이 -통계적 이론에 기반한 분석 -삽입과 삭제의 제한 -긴 크기의 서열에 제한적임
	일반화 프로파일	-삽입과 삭제의 연산 가능 -유사도 측정 민감 -프로파일의 자동화 가능 -삽입과 삭제의 지연 연산과 자원의 사용량 과다함

일반화 프로파일은 PSSM을 일반화한 것이다. 이 모델은 삭제와 삽입이 발생하는 특정 위치에서도 적용 가능하고 유사성을 탐지하는데 매우 민감하며 좋은 측정 시스템으로 프로파일을 자동으로 구축할 수 있으나, 시스템 자원의 사용량이 많고 삽입과 삭제에 따른 지연 시간을 갖게 된다.

HMM은 확률이론인 유한 수 상태들의 전이들로 연결되어 있는 Markov Chains 이론의 확장이다. HMM은 일종의 오토마타로서 모델을 구성하고 있는 상태들 간의 전이가 특정한 확률 값을 통하여 이루어진다. 그러나 일반화 프로파일의 방법과 같이 많은 컴퓨팅 자원의 요구와 처리에 지연을 갖는 단점이 있다.

따라서 본 논문에서는 서열정렬의 유사성을 비교하는데 있어 구현이 용이하며, 매칭된 서열의 통계적 분석을 가능하게 하는 서열내의 특정 위치기반의 점수행렬(PSSM)[10]의 방법을 침입패턴의 유사성 매칭에 적용하고자 한다. 또한, 비교되는 침입패턴의 길이에 대한 실험으로 각 공격에 대한 과탐 및 오탐의 수를 줄이는 서열의 크기를 비교하여 보인다.

Ⅲ. 침입패턴의 유사성 측정

3.1 다중서열정렬 모델링 접근

침입탐지에 사용하는 이름(nominal) 또는 카테고리(category), 순서(ordinal), 수량(quantitative)의 다양한 척도들 중에서 순서에 기반한 시스템 호출 순서는 생명정보학의 다중서열과 유사한 특성을 가지고 있다. 이러한 순서열의 척도를 대상으로 침입패턴의 형식을 정의하고, 특정 위치에서 빈도수에 기반한 침입패턴을 생성할 수 있다.

다중서열에서 가장 큰 점수를 갖는 배열이 생물학적 의미를 가진다고 할 때, 다중의 서열을 최적으로 배열하는 점수를 계산하는 방법이 필요하다. 점수 행렬은 두 서열을 비교할 때 각 서열들이 일치 혹은 치환될 확률을 계산해 주는 행렬이다. 점수행렬의 선택은 분석 결과에 중대한 영향을 갖기 때문에 서열이 임의로 발생할 확률과 의미 있는 사건이 발생할 확률을 비교할 때, 의미 있는 사건의 순서열이 발생할 확률이 크다면 양의 로그값을 가지고, 임의 순서열이 우연히 일어날 확률이 크다면 음의 로그값이 된다. 점수가 양의 방향으로 커질수록 순서열이 중요해질 가능성이 더욱 커지게 되어 해당하는 비교 입력데이터는 유사성이 많다는 것을 의미한다. 그림 1은 PSSM의 방법을 적용한 침입패턴의 다중서열정렬 모델링의 수행 단계를 보인 것이다.

각 단계에서 사용하는 기호는 다음과 같이 정의한다.

- i : 시스템 호출 번호
- j : 생성된 순서열에서 열의 위치
- $N_{i,j}$: 순서열의 위치 j에서 나타난 시스템 호출 i의 개수
- Ns : 순서열에 나타난 시스템 호출의 전체 개수

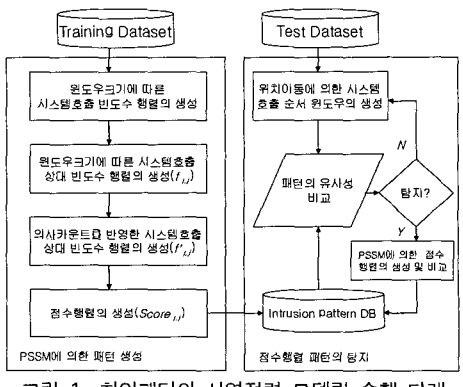


그림 1. 침입패턴의 서열정렬 모델링 수행 단계

- $f_{i,j}$: 생성된 순서열의 위치 j에서 나타나는 시스템 호출 j의 상대 빈도수
- Pc : 의사카운트 값
- Ws : 윈도우 크기
- $f'_{i,j}$: 의사카운트를 반영한 패턴의 위치 j에서 나타나는 시스템 호출 j의 상대 빈도수
- q_i : 임의 순서열에서 윈도우 크기에 대한 시스템 호출 의 기대되는 상대 빈도수
- $Score_{i,j}$: 임의 순서열에서 q_i 에 대한 $f'_{i,j}$ 의 상대 빈도수의 로그 값

3.2 침입패턴 서열의 생성

다중서열정렬의 PSSM을 이용하여 침입패턴 또는 서열을 생성하기 위해 이미 학습된 *eject*의 버퍼 오버플로우 공격의 예를 이용하여 보인다. 그림 2는 *eject* 공격의 시스템 호출 순서의 예를 보인 것이다. 그림 1에서 패턴을 생성하는 각 단계의 절차는 다음과 같다.

〔단계 1〕 윈도우 크기에 따른 빈도수 행렬 생성

그림 2의 시스템 호출 순서열에 대해 PSSM 방법을 적용하기 위하여 먼저, 연속적으로 반복되는 시스템 호출 번호를 제외한 나머지 시스템 호출 번호를 대상으로 일정한 윈도우의 크기로 나누고, 각 위치에서 빈도수를 기반으로 하는 행렬을 구성한다.

표 2는 *eject* 공격의 시스템 호출 순서에 따른 빈도수를 나타낸 것이며, 예를 보이기 위하여 윈도우의 크기(Ws)는 10으로 하였으며, 시스템 호출의 빈도수가 모두 0인 것은 생략하여 나타내었다. 공격에 대한 탐지 및 오탐의 결과에 따라 윈도우의 크기를 결정할 수 있다.

〔단계 2〕 윈도우 크기에 따른 시스템호출 상대빈도수 행렬 생성

단계 1에서 생성한 빈도수 행렬은 순서열에서 나타난 시스템 호출의 전체 개수에 대한 임의의 시스

23 72 210 72 210 210 213 210 210 112 72 210 112 72
210 210 213 210 112 112 213 158 158 27 23 72 210 72
210 210 213 210 210 112 72 210 210 213 210 112 72 210
210 213 210 112 72 210 210 213 210 210 112 72 210 210
213 210 112 72 210 210 213 210 112 72 210 112 72 210
210 213 210 112 112 213 80 158 112 16 16 16 16 16 16
16 80 158 112 16 16 16 16 16 16 16 23

그림 2. *eject* 버퍼오버플로우 공격의 시스템 호출 순서

표 2. *eject* 공격의 시스템 호출 순서에 따른 빈도수

시스템호출	1	2	3	4	5	6	7	8	9	10
16	0	0	0	0	0	1	1	1	1	2
23	2	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1
72	13	13	12	12	11	11	11	11	11	10
80	0	0	1	1	1	1	2	2	2	2
112	11	11	11	11	12	12	12	12	12	12
158	1	1	1	2	2	2	2	3	3	3
210	22	22	22	21	21	20	20	19	19	19
213	10	11	11	11	11	11	10	10	10	10

템 호출이 윈도우내의 각 위치에서 나타나는 상대 빈도수를 갖는 행렬로서 구성할 수 있다. 각 시스템 호출 번호에 대해 각 열에서의 상대 빈도수($f_{i,j}$)를 식(1)과 같이 계산한다. 표 3은 *eject* 공격의 시스템호출 상대 빈도수를 나타낸 것이다.

$$f_{i,j} = \frac{N_{i,j}}{N_s} \tag{1}$$

$$\text{열 } 1: f_{16,1} = \frac{0}{9} = 0,\ f_{23,1} = \frac{2}{9} = 0.22, \ldots$$

$$\text{열 } 2: f_{16,2} = \frac{0}{9} = 0,\ f_{23,2} = \frac{1}{9} = 0.11, \ldots$$

$$\text{열 } 10: f_{16,10} = \frac{2}{9} = 0.22,\ f_{23,10} = \frac{1}{9} = 0.11, \ldots$$

〔단계 3〕 의사카운트를 반영한 시스템호출 상대빈도수 행렬의 생성

단계 2에서 생성한 상대 빈도수를 갖는 행렬에서 각 항이 0의 값을 가질 수 있다. 이것은 선택 가능한 순서열의 시스템 호출을 삭제하여 유사성 비교시의 비교 점수를 낮게 하는 원인이 된다. 따라서 0의 값을 갖는 각 항에 작은 값을 추가하여 해당하는 시스템 호출이 의미를 갖도록 하는 것이 필요하다. 따라서 치환행렬에 의사카운트(pseudo-count) 값을 더하여 변환하는 것과 윈도우 크기에 대한 각 항의 비율을 반영한 상대빈도수의 계산이 필요하다. 의사카운트(Pc)를 반영한 상대 빈도수($f'_{i,j}$) 행렬의 각 항의 계산은 식(2)와 같다. 의사카운트는 0인 값들

표 3. *eject* 공격의 시스템 호출의 상대 빈도수($f_{i,j}$)

시스템 호출	1	2	3	4	5	6	7	8	9	10
16	0.00	0.00	0.00	0.00	0.00	0.11	0.11	0.11	0.11	0.22
23	0.22	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
...					...					
213	1.11	1.22	1.22	1.22	1.22	1.22	1.11	1.11	1.11	1.11

에 과다한 반영을 하지 않는 1의 값을 사용한다.

$$f'_{i,j} = \frac{N_{i,j} + Pc}{Ns + Ws} \quad (2)$$

〔단계 4〕 점수행렬의 생성

단계 3에서 생성한 의사카운트를 반영한 상대 빈도수는 임의 사건의 시스템호출 순서와 비교를 위한 점수행렬을 식(3)과 같이 생성한다. q_i는 윈도우 크기에 대한 각 위치 i의 기대 빈도의 값이며, 윈도우의 크기가 10일 경우, 0.1(1/10)의 값을 갖는다. 각 행렬의 점수는 로그를 취함으로써 음의 점수를 보일 경우는 우연히 발생할 가능성을 보인 것이며, 양의 점수를 나타내는 경우는 의도된 또는 의미있는 사건의 시스템호출 순서열이 될 수 있는 가능성을 보이기 위한 것이다.

$$Score_{i,j} = \log(\frac{f'_{i,j}}{q_i}) \quad (3)$$

3.3 순서기반 침입패턴의 유사성 비교

생성된 각 공격의 점수행렬은 시험 데이터의 순서열이 윈도우 크기가 되도록 하여 비교한다. 각 시스템 호출의 위치를 하나씩 이동하면서 점수의 합이 가장 큰 순서열을 선택하면, 선택한 순서열의 각 요소의 통계적 해석을 가능하게 한다. 탐지된 시스템 호출의 순서열은 선택한 침입패턴의 점수행렬과 비교하여 데이터베이스에 반영할 수 있다.

순서기반 침입패턴의 유사성 매칭을 위해 시스템 호출 순서의 시험 데이터를 *eject* 공격에 대한 점수행렬에 적용하여 각 위치를 한 단계씩 이동하는 방법으로 점수의 합이 가장 큰 값을 갖는 시스템 호출

표 4. *eject* 공격의 확장된 점수행렬

시스템 호출	1	2	3	4	5	6	7	8	9	10
16	-0.27	-0.27	-0.27	-0.27	-0.27	0.02	0.02	0.02	0.02	0.19
23	0.19	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
24	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27
. . .	. . .									
122	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27	-0.27
210	1.08	1.08	1.08	1.06	1.06	1.04	1.04	1.02	1.02	1.02
213	0.76	0.80	0.80	0.80	0.80	0.80	0.73	0.73	0.73	0.76

```
Input Sequence : 24, 110, 210, 210, 210, 213, 26, 92, 210,
210, 213, 210, 213, 27, 23, 210, 89, 89, 92, 122, 16, 16
연속된 값 삭제한 Input Sequence로 변환
24, 110, 210, 213, 26, 92, 210, 213, 210, 213, 27, 23, 210,
89, 92, 122, 16
----------------------------------------------
Position 0: Score = 4.358 = (-0.279) +(-0.279) +(1.083)
+(0.800) +(-0.279) +(-0.279) +(1.043) +(0.763) +(1.022)+(0.763)
Transformed Input Sequence: (24, 110, 210, 213, 26, 92, 210,
213, 210, 213), 27, 23, 210, 89, 92, 122, 16
Position 1: Score = 4.659
Transformed Input Sequence: 24, (110, 210, 213, 26, 92, 210,
213, 210, 213, 27), 23, 210, 89, 92, 122, 16
Position 2: Score = 5.039
Transformed Input Sequence: 24, 110, (210, 213, 26, 92, 210,
213, 210, 213, 27, 23), 210, 89, 92, 122, 16
            . . .
Position 7: Score = 3.050
Transformed Input Sequence: 24, 110, 210, 213, 26, 92, 210,
(213, 210, 213, 27, 23, 210, 89, 92, 122, 16)
```

그림 3. *eject* 공격 입력데이터 및 변환 입력 순서열

순서열을 점수행렬로 선택하여 침입탐지에 이용한다. 공격의 탐지 및 정상에 대한 오탐을 줄이기 위하여 서열 윈도우의 크기를 조정할 수 있다. 표 4는 점수행렬의 정확한 순서열을 선택하기 위하여 시스템 호출의 개수 및 정확도를 확장하여 보인 것이다.

그림 3은 비교되기 위한 입력데이터의 시스템 호출 순서열의 효과적인 탐지를 위하여 연속적으로 중복된 시스템 호출을 삭제하며, 표 5의 eject 점수행렬에 적용하여 시스템 호출 순서열의 점수 행렬을 계산하는 과정을 보인다.

이 결과에서 Position2가 가장 높은 점수인 5.039로 계산되었으며, 시스템 호출 순서의 일치가 가장 많이 이루어짐을 알 수 있다. 제안한 바와 같이 전체 순서열이 일치되는 패턴들만 고려하지 않고, 부분적으로 일치되는 순서열을 반영하여 점수를 계산하는 방식을 사용하게 되면, 유사한 패턴이나 변형된 공격의 패턴에 대해서도 보다 효율적으로 탐지 할 수 있는 가능성을 제시할 수 있다.

Ⅳ. 실험 및 결과

4.1 실험데이터 및 공격 프로파일

본 논문에서는 제안된 방법의 신뢰성 있는 실험을 위한 순서기반의 시스템 호출 데이터는 표 5에서 보인 것처럼 UNM(University of New Mexico)과

표 5. 실험 데이터 리스트

데이터 유형			공격파일(*.int)	시스템 호출수
UNM	sunsendmailcp intrusion	공격	sm-10763	373
			sm-10801	373
			sm-10814	373
	decode intrusion (sendmail)	공격	sm-280	1534
			sm-314	1533
		정상	bounce-1	293
			bounce-2	739
			bounce	818
			sendmail.log	31821
CERT	decode intrusion	공격	syslog-local-1	1516
			syslog-local-2	1574
			syslog-remote-1	1861
			syslog-remote-2	1553
		정상	sendmail.daemon	1556560
			sendmail	19526

CERT에서 제공한 *sendmail* 데이터의 공격과 정상 데이터를 이용하였다[11].

먼저 각 공격에 대한 시스템 호출의 프로파일을 만들기 위하여 정상데이터를 제외한 모든 공격에 대해 시스템 호출 순서의 침입패턴 서열을 만든다. 표 6은 *sendmail* 공격 중 *sm-10763*에 대해 윈도우 크기 변화에 따른 시스템 호출 순서의 변화를 나타낸 것이다. 표 7은 윈도우 크기 10의 경우에 *sm-10763* 공격의 시스템 호출 순서를 보인 것이다.

4.2 공격 및 정상에 대한 탐지 및 오탐의 결과

표 8은 UNM *sendmail*의 공격에 대한 탐지 및 정상에 대한 오탐의 개수를 나타낸 것이다. *sunsendmailcp*공격은 *sm-10763*으로 탐지 프로파일을 생성한 결과이며, *sm-10801*과 *sm-10763*도 같은 프로파일을 생성한다. 정상에 대해서는 윈도우 크기 15에서 오탐의 수가 0이 됨을 보였다. decode(*sendmail*) 공격은 *sm-280*으로 탐지 프로파일을 생성한 결과이며, *sm-314*도 같은 결과를 생성한다. 윈도우 크기 20에서 정상에 대한 오탐의 개수가 0인 결과를 보였으며, 윈도우 크기의 증가에 따라 decode 공격의 개수가 감소함을 보였다. *sendmail.log*는 윈도우 크기 10에서 많은 오탐을 보였지만 윈도우 크기의 증가에 따라 제거 되었음을 알 수 있다. 정상행위의 탐지에 대해 윈도우 크기가 증가함에 따라 오탐이 감소한 것은 정상행위의 짧은 시퀀스에 공격과 유사한 시스템 호출 순서가 많음을 보임으로써 작은 윈도우의 크기로 탐지 시에 오탐의 가능성을 보인 것이다.

표 9와 그림 4는 CERT에서 제공한 *sendmail* 공격에 대한 탐지 및 정상에 대한 오탐의 수를 나타낸 것이다. 공격 *syslog-local1*, *syslog-local2*, *syslog-remote1*, *syslog-remote2*와 정상 *sendmail.daemon*,

표 6. 윈도우 크기에 따른 침입패턴 서열의 예

윈도우 크기	sm-10763 공격 시스템 호출 순서
5	19;112;4;2;5;
10	112;19;112;4;2;5;112;50;27;4;
15	5;112;19;128;112;19;128;4;3;112;19;128;112;3;19;
20	2;50;27;2;3;5;112;19;128;112;19;128;4;3;112;19;128;112;3;19

표 7. *sm-10763* 공격 점수행렬의 예(윈도우크기 10)

시스템 호출	1	2	3	4	5	6	7	8	9	10
1	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68
2	0.49	0.49	0.46	0.46	0.46	0.46	0.46	0.46	0.46	0.46
3	0.36	0.39	0.39	0.39	0.43	0.43	0.43	0.43	0.43	0.43
4	0.71	0.69	0.69	0.69	0.68	0.68	0.68	0.68	0.68	0.69
5	0.66	0.61	0.68	0.68	0.68	0.69	0.69	0.69	0.69	0.69
18	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
19	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
26	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68
27	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
49	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68
50	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
111	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68
112	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.69	0.69
113	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68	-0.68

표 8. UNM *sendmail*의 공격과 정상의 탐지 결과

(단위 : 개)

데이터유형			윈도우크기		
			10	15	20
sunsendmailcp	공격	sm-10763	1	1	1
		sm-10801	1	1	1
		sm-10814	1	1	1
	정상	bounce-1	0	0	0
		bounce-2	1	0	0
		bounce	0	0	0
		sendmail.log	48	0	0
decode (sendmail)	공격	sm-280	3	2	1
		sm-314	3	2	1
	정상	bounce-1	3	2	0
		bounce-2	1	0	0
		bounce	3	2	0
		sendmail.log	49	0	0

표 9. CERT *sendmail*의 공격과 정상에 대한 탐지 결과

(단위 : 개)

데이터유형			윈도우크기				
			10	13	15	20	50
sendmail	공격	syslog-local1	75	69	63	52	20
		syslog-local2	75	68	63	52	20
		syslog-remote1	83	76	69	56	20
		syslog-remote2	59	55	51	44	20
	정상	sendmail.daemon	2	1	0	0	0
		sendmail	49	0	0	0	0

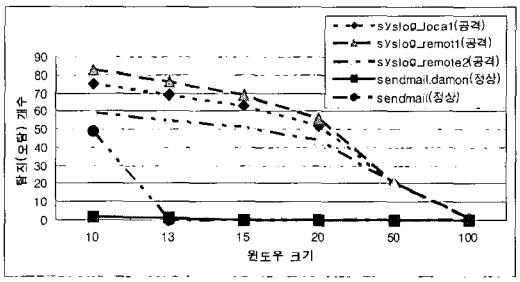


그림 4. CERT *sendmail*의 윈도우 크기에 따른 탐지 및 오탐의 결과

*sendmail*으로 침입패턴 서열을 생성한 결과이다. 탐지 결과 정상은 윈도우 크기 15에서 오탐의 개수는 0이 되었으며, 윈도우의 크기가 단계적으로 증가함에 따라 시스템 호출 순서의 반복된 알람의 개수가 축약됨을 알 수 있다. 실험의 결과 *syslogd* 공격의 탐지는 윈도우의 크기를 증가함에 따른 탐지 결과의 변화에 대한 분석을 요구한다. 대표적인 시스템 호출은 *read(2), close(5), getpid(19), gettimeofday(112)*으로 나타났다..

실험에 따른 결과로 윈도우 크기에 따라 공격의 탐지 및 정상에 대한 오탐의 결과를 보임으로써 각 공격 및 정상에 따라 최적의 윈도우의 크기를 찾을 수 있으며, 각 윈도우 크기에 따른 최적의 침입패턴 서열을 생성할 수 있다. *Forrest*의 연구 결과는 순서기반의 정상행위 시스템 호출에 슬라이딩 윈도우의 방법을 사용하였으며, 오류를 포함한 윈도우의 크기를 10으로 하였다. 제안한 실험의 결과는 동일한 윈도우 크기에서 각각 공격 및 이상행위를 탐지하였다. 그러나 정상에 대한 오탐으로 실험의 결과에서는 제한적이지만 윈도우의 크기는 15-20이 적정함을 보이고 있다. 따라서 윈도우의 크기를 증가시키면서 정확도를 높일 수 있으며, 각 행위에 따른 최적의 윈도우의 크기를 결정할 수 있음을 보였다.

표 10. 정상행위 프로파일에 의한 이상행위 탐지의 비교

(단위 : 개)

구분	슬라이딩 윈도우	특정위치 기반의 점수행렬	
	window size =10	window size =10	window size =15
sunsendmailcp	92	0-48	0
decode	7-22	1-49	0
syslogd	248-529	2-49	0-1

표 10은 *Forrest*의 윈도우 크기 10인 경우의 이상행위 탐지의 결과와 본 논문에서 제안한 방법의 윈도우 크기에 따른 오탐의 수를 보인 것이다.

비교 연산의 수행 측면에서 *Forrest*의 방법은 윈도우 크기가 크면 클수록 보다 정확한 이상행위를 탐지 하겠지만, 윈도우의 크기에 따라 유사도를 구하기 위한 비교 연산의 과부하가 있게 된다. 그러나 본 논문에서 제안한 특정 위치기반의 점수행렬을 이용한 방법은 윈도우의 크기에 따른 곱셈과 합의 연산만이 필요하다. 따라서 윈도우의 크기가 비교 연산의 수행에 과부하를 주지 않는다. 또한, 유사도 비교의 매칭된 결과에 대해 *Forrest*의 방법은 거리기반의 유사도에 의한 탐지의 결과만을 보이지만 제안한 방법은 침입패턴 서열을 구성하는 시스템 호출 순서의 통계적 분석이 가능하다는 장점이 있다.

V. 결론 및 향후 연구

본 논문은 시스템 호출 순서에 기반한 침입탐지를 위한 방법으로 기존의 대용량 오용행위 침입패턴을 대상으로 다중서열정렬 기법의 유사도 매칭의 방법을 제안하였다. 구현이 용이하며 침입패턴의 생성 및 탐지 결과의 분석에 통계적 접근 방법을 제공하고 비교 연산 시간이 빠른 특정위치 기반의 점수행렬 방법을 적용하였다.

UNM 및 CERT의 *sendmail* 데이터를 이용하여 실험을 하였으며, 그에 따른 결과는 윈도우 크기에 따라 공격의 탐지 및 정상에 대한 오탐의 결과를 보임으로써 각 공격별 최적의 윈도우의 크기를 찾을 수 있으며, 이에 따른 침입패턴 서열을 구축하여 공격 탐지에 적용할 수 있는 패턴을 만들 수 있음을 보였다. *Forrest* 방법과의 비교에서는 윈도우 크기에 따른 비교연산의 시간을 빠르게 할 수 있으며, 탐지된 결과를 분석할 수 있음을 보였다. 단, 실험의 결과에서는 CERT *syslogd* 데이터의 공격에 대한 탐

지의 실험 및 분석의 보완이 필요하다.

향후 연구에서는 현재 적용된 방법이 탐지 척도로 자주 사용되는 이름 또는 카테고리, 순서, 수량의 유사성 비교에 적용 가능하므로 다양하게 생성된 침입패턴에 적용하는 것이다. 또한 기존의 이상행위 프로파일링 방법과 다양한 척도에서 비교될 수 있는 유사성 측정 수행의 보완, 다중서열정렬의 확장, 침입패턴 데이터베이스에서 중복패턴의 제거, 침입패턴의 진화계층의 관계 분석 등의 연구가 필요하다.

참고 문헌

[1] S. Kummer and E. H. Spafford, "An Application of Pattern Matching in Intrusion Detection," *Purdue University, Technical Report* CSD-TR-94-013, 1994.

[2] A. Floratos, et al., "DELPHI: A patern-based method for detecting sequence similarity," *IBM J. RES. & DEV.*, Vol. 45, No. 3/4, May/July, 2001.

[3] E.C. Rouchka, "Pattern Matching Techniques and Their Applications to Computational Molecular Biology-A Review," *WUCS-99-09*, March, 1999.

[4] C. Warrender, S. Forrest and B. Pearlmutter. "Detecting intrusions using system calls: Alternative data models." *IEEE Symposium on security and Privacy*, pp. 133-145, 1999.

[5] S. Hofmeyr, S. Forrest and A. Somayaji, "Intrusion Detection Using Sequences of System Call," *IEEE Journal of Computer Security*, Vol. 6, pp. 151-180, 1998.

[6] A. Kosoresow, "Intrusion Detection using Sequence of System Call Traces," *IEEE Software*, Vol. 14, No. 5, pp. 35-42, 1997.

[7] D. Dasgupta and S. Forrest, "An Immune Agent Architecture for Intrusion Detection," *Proceed- ing of the GECCO 2000 Workshop Prog.*, pp. 42-44, 2000.

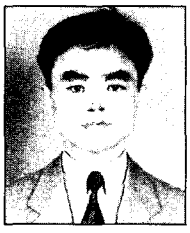
[8] A. Wespi, M. Dacier and H. Debara, "Intrusion detection using variable-length audit trail patterns," *Recent Advances in Intrusion Detection* (RAID 2000), pp.110-129, 2000.

[9] W. Lee and S. Stolfo, "Learning Patterns from Unix Process Execution Traces for Intrusion Detection," *AAAI Workshop*, pp. 50-56, 1997.

[10] M. Pagni, "Introduction to Patterns, Profiles and Hidden Markov Models," Swiss Institute of Bioinformatics, http://www.ch.embnet.org,Aug. 2002.

[11] S. Forrest, "Computer immune systems data sets," *University of New Mexico*, http://www. cs.unm.edu, 1997.

〈著者紹介〉



김 용 민 (Yong-Min Kim) 종신회원

1989년 : 전남대학교 전산통계학과 학사
1991년 : 전남대학교 전산통계학과 석사
2002년 : 전남대학교 전산통계학과 박사
2004년 3월~현재 : 여수대학교 정보기술학부
〈관심분야〉 시스템 및 네트워크 보안, 전자상거래 보안 등