韓國 컴퓨터情報學會 論文誌
第10卷 第5號, 2005. 11.

2005-10-5-1-4

텍스트 마이닝 기법을 이용한 컴퓨터 네트워크의 침입 탐지

오 승 준*, 원 민 관**

Using Text Mining Techniques for Intrusion Detection Problem in Computer Network

Seung-Joon Oh *, Min-Kwon Won **

요 약

최근 들어 데이터 마이닝 기법을 컴퓨터 네트워크의 침입 탐지에 적용하려는 많은 연구가 진행되고 있다. 본 논문에서는 침입 탐지 분야에서 프로그램 행위가 정상적인지 비정상적인지를 분류하기 위한 방법을 연구한다. 이를 위해, 텍스트 마이닝 기법중의 하나인 k 최근접 이웃 (kNN) 분류기를 이용한 새로운 방법을 제안한다. 본 논문에서는 텍스트 분류 기법을 적용하기 위해 각각의 시스템 호출을 단어로 간주하고, 시스템 호출의 집합들을 문서로 간주한다. 이러한 문서들은 kNN 분류기를 이용하여 분류된다. 간단한 예제를 통하여 제안하는 절차를 소개한다.

Abstract

Recently there has been much interest in applying data mining to computer network intrusion detection. A new approach, based on the k-Nearest Neighbour(kNN) classifier, is used to classify program behaviour as normal or intrusive. Each system call is treated as a word and the collection of system calls over each program execution as a document. These documents are then classified using kNN classifier, a popular method in text mining. A simple example illustrates the proposed procedure.

▸ Keyword : 침입 탐지(Intrusion Detection), 텍스트 마이닝(Text Mining), 분류(Classification)

• 제1저자 : 오승준
• 접수일 : 2005.08.29, 심사완료일 : 2005.09.12
* 경기공업대학 산업경영시스템과 교수, ** 경기공업대학 e-비즈니스과 교수

Ⅰ. 서 론

보안 정책에 허점이 있거나, 컴퓨터 구성을 잘못하거나, 완벽하지 못한 소프트웨어를 사용하는 한 컴퓨터 보안의 취약함은 항상 존재한다. 침입 탐지 시스템은 이러한 취약함들을 이용하는 공격들을 탐지하는데 중요한 역할을 한다.

이상적인 침입 탐지 시스템은 공격 탐지 비율이 100%이고 false positive 비율(정상적인 행동을 잘못 분류하는 비율)이 0% 인 시스템이다. 그러나 현재의 침입 탐지 시스템은 높은 false alarm 비율과 낮은 공격 탐지 비율로 문제가 되고 있다.

침입 탐지 시스템에 대해서는 두 개의 접근법이 있다. 오용 탐지 (misuse detection)와 비정상행위 탐지 (anomaly detection)이다.

서명 검증을 통한 오용 탐지는 사용자의 행위와 시스템을 침입하려는 공격자들의 이미 알려진 서명을 비교한다. 오용 탐지는 이미 알려진 탐지 기법들을 찾아 내는데는 유용하지만 새로운 공격들을 발견할 수는 없다.

오용탐지와 달리 비정상행위 탐지는 사용자, 시스템 또는 네트워크에 대한 통계적 패턴들로부터 벗어나는 행위들을 찾아낸다. 기계 학습 기법들이 정상적인 사용 패턴들을 찾아내고 새로운 행위들이 정상인지 비정상인지를 분류하는데 사용되어 왔다. 알려지지 않은 공격들을 찾아내는 능력에도 불구하고 비정상 행위 탐지 시스템은 높은 false alarm 비율로 문제가 된다. 정상적인 사용자의 프로파일과 시스템이나 네트워크의 행위가 광범위하게 변화할 때 비정상 행위 탐지는 서명 검증과 합쳐져서 공격들을 보다 효율적으로 찾아내는데 사용 될 수 있다.

본 논문에서는 침입 탐지 기법에 있어 새로운 기법을 제안한다. 여기서는 텍스트 마이닝 기법으로 유명한 k-Nearest Neighbour(kNN) 분류기를 이용하여 새로운 프로그램의 행동들을 정상적인지 비정상적인지 구분한다. 시스템 호출은 '단어'로, 각각의 프로세스들은 '문서'로 취급된다. 문서 분류 영역에서 성공적으로 사용되어온 kNN 방법은 침입 탐지에 쉽게 적용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 연구를 다루며, 3장에서는 제안하는 방법에 대해 설명한다. 4장에서는 3장에서 제안한 방법을 이용한 간단한 예제를 보여주며, 마지막으로 5장에서 결론을 기술한다.

Ⅱ. 기존 연구

UC Davis의 Ko et al. [1] 이 처음으로 몇몇 특권을 가진 프로그램들을 이용하여 고의적인 행동들을 찾아내는 것을 제안했다. 프로그램 실행 중에 특정한 행동의 위반들은 '오용'으로 간주된다. 이 모델의 주요 단점은 고의적인 행위와 모든 프로그램들에 대한 보안 사양서를 기술하는데 어려움이 있다는 것이다. 그럼에도 불구하고 이 연구는 침입 탐지를 모델링 하는데 문을 열었다.

뉴 멕시코 대학의 Forest 그룹은 실행중인 프로그램에 의해 발생하는 시스템 호출의 짧은 시퀀스들을 침입 탐지를 위한 판별기로 이용하는 아이디어를 도입하였다[2]. 정상적인 행동들은 유닉스 프로세스에서 실행중인 일정 길이내의 짧은 시퀀스들에 의해 정의된다. 정상적인 행동들의 데이터베이스가 각각의 관심 있는 프로세스에 대해서 만들어진다. 이 작업은 인공 면역 시스템[3], 규칙 학습[4], Hidden Markov Model[5]와 같은 다양한 분류기법들로 확장된다.

각각의 프로그램 프로파일들을 만드는데 집중한 대부분의 연구들과 달리 Asaka et al. [6]은 판별 분석에 기반을 둔 방법을 도입하였다. 모든 시스템 호출들을 실행하지 않고, 단지 11개의 시스템 호출을 분석하고 Mahalanobis 거리를 계산하여 침입 탐지 결정이 이루어진다. 42개의 샘플들 중 4개만이 오분류 되었다. 하지만 샘플 데이터의 작은 크기 때문에 이 기법의 실현 가능성은 낮다.

kNN 방법을 이용하여 시퀀스들을 분류한 연구로 Deshpande et al.[7]과 Juan et al. [8]이 있고, 침입탐지 시스템에 이용된 연구로는 최인수 외[9] 김강 외[10] 등이 있다.

III. 제안하는 방법

3.1 kNN을 이용한 제안하는 알고리듬

각각의 시스템 호출을 문서의 '단어'로 간주하고, 프로세스에 의해 발생된 시스템 호출들의 집합을 '문서'로 간주한다. 이러한 방법들은 기존의 많은 문서 처리 방법들을 침입 탐지 시스템에 이용할 수 있도록 해 준다. 문서 처리 방법들 중에서는 kNN 분류 방법을 이용하며, 각각의 프로세스들은 시퀀스들로 표현된다.

kNN을 이용한 분류 기법은 알고리즘의 단순함과 비교적 낮은 오분류율로 인해 텍스트 마이닝 분야에서 널리 사용되고 있는 기법중의 하나이다.

텍스트 분류기법에서 사용되는 용어와 본 논문에서 사용되는 용어들을 비교하면 〈표 1〉과 같다.

표 1. 분류기법과 침입 탐지 용어들 사이의 유사점
Table 1. Analogy between text categorization and intrusion detection

용어	분류기법	침입 탐지
N	문서의 총 개수	프로세스의 총 개수
M	단어의 총 개수	시스템 호출의 총 개수
Dj	j번째 트레이닝문서	j번째 트레이닝 프로세스
X	테스트 문서	테스트 프로세스

제안하는 알고리듬은 (그림 1)과 같다.

```
build the training normal data set D;
for each process Dj in training data
  calculate sim(X, Dj);
  if sim(X, Dj) equals 1.0 then
    X is normal; exit;
find k biggest scores of sim(X, Dj);
calculate sim_avg for k-nearest neighbours;
if sim_avg is greater than thershold then
    X is normal;
else
    X is abnormal;
```

그림 1. 제안하는 알고리듬
Fig. 1. The proposed algorithm

처음에는 정상적인 프로그램 행위들에 기반을 두고 침입 탐지 시스템을 구현하였다. 모든 가능한 정상적인 프로그램 행위들이 포함되도록 하기 위하여, 침입 탐지 시스템에서는 큰 트레이닝 데이터 셋이 선호된다. 반면에 큰 트레이닝 데이텃 셋은 프로그램 행위들을 모델링 하는데 있어 큰 오버헤드를 의미한다.

트레이닝 데이터 셋이 만들어지면, kNN 문서 분류 기법을 침입 탐지에 쉽게 응용할 수 있다. 테스트 데이터의 각각의 프로세스들을 시퀀스들로 변환한다. 새로운 프로세스와 트레이닝 데이터 셋에 있는 각각의 프로세스들 간의 유사도를 계산한다. 만약 유사도가 1이라면, 이것은 새로운 프로세스와 트레이닝 셋에 있는 프로세스가 완벽하게 일치하는 것을 의미하며, 새로운 프로세스는 정상적인 프로세스로 분류될 것이다. 그렇지 않으면, 유사도 점수가 정렬되어, k 개의 최근접 이웃들이 새로운 프로세스가 정상적인지 아닌지를 결정하는데 사용된다. k 개의 최근접 이웃들의 유사도 평균을 계산한다. 유사도 평균이 threshold 이상인 경우에만 새로운 프로세스가 정상으로 간주된다.

3.2 유사도 계산 방법

본 논문에서는 오승준 [11]에서와 같이 두 시퀀스들간의 유사도를 계산한다.

데이터베이스 D는 시퀀스들의 집합이고, 시퀀스 S는 n 개의 항목들의 모임이며 〈 x1 x2 … xi … xj … xn 〉으로 표시하고, 여기서 xi 는 범주형 값을 가지는 항목이다. S의 크기는 S에 있는 항목들의 개수이며, |S| 로 나타낸다. 시퀀스 S에서 순서를 가지는 2개의 항목들로 구성된 xi xj (i<j)를 시퀀스 요소 ek 라고 하며, ek 들의 모임을 E = (e1, e2 , … , ek, …)라 한다. E의 크기는 E에 있는 요소들의 개수이며, |E| 로 나타낸다.

[예 1] 시퀀스 S = 〈 A B C E 〉에서 |S| = 4이고, 시퀀스 요소들의 모임은 E = (AB, AC, AE, BC, BE, CE)이며, |E| = 6이다.

트레이닝 데이터 셋에서 나타나는 시스템 호출들의 예는 (그림 2)와 같은데, 본 논문에서는 이들을 간단히 항목들로 표현하여 유사도를 계산한다.

access audit audition chdir chmod
close execve exit fchdir fcntl
fork getmsg kill link login logout
mkdir mmap nice open pipe
readlink rename rmdir setaudit
seteuid setgroups setuid stat

그림 2. 시스템 호출의 예
Fig. 2. Example of system calls

시퀀스내의 항목들뿐만 아니라 항목들 간의 순서도 고려를 해서 식(1)과 같이 유사도 계산 방법을 제안한다.

[정의 1] 두 시퀀스 S1 = 〈 a1 a2 … an 〉과 S2 = 〈 b1 b2 … bm 〉의 시퀀스 요소들의 모임을 각각 E1 = (ea1, ea2, … , eai , …), E2 = (eb1, eb2, … , ebj , …)라고 하면, S1, S2 의 유사도 sim(S1, S2)는 다음과 같이 정의한다.

$$sim(S_1, S_2) = \frac{|E_1 \cap E_2|}{\frac{|E_1| + |E_2|}{2}} \quad \cdots\cdots\cdots\cdots (1)$$

여기서, $|E_1 \cap E_2|$ 는 E1과 E2의 공통 요소들의 개수이며, E1과 E2사이에 공통 항목들이 많을수록 유사도는 높고, 이 값을 $(|E_1| + |E_2|)/2$로 나누는 것은 유사도를 0과 1사이의 값을 갖도록 하기 위해서이다.

[예 2] 두 시퀀스 S1 = 〈 A B D A 〉, S2 = 〈 A C D A C 〉에서 시퀀스 요소들의 모임은 각각 E1 = (AB, AD, AA, BD, BA, DA) 과 E2 = (AC, AD, AA, AC, CD, CA, CC, DA, DC, AC) 이며, |E1| = 6, |E2| = 10, E1∩E2 = (AD, AA, DA), |E1∩E2| =3이다. 따라서, 두 시퀀스의 유사도 sim(S1, S2)는 3/8이다.

Ⅳ. Case Study

이번 장에서는 본 논문에서 제안하는 알고리듬의 절차를 보여주기 위한 예제를 소개한다.

K라는 회사에서는 솔라리스 시스템을 운영하고 있는데, 이 시스템에서 발생되는 5일 동안의 시스템 호출을 수집하였다. 이중 4일치의 데이터는 트레이닝 데이터 셋으로 사용하였고, 1일치는 테스트 데이터 셋으로 사용하였다.

본 논문에서 제안하는 절차를 보여주기 위해 전체 트레이닝 데이터 셋과 테스트 셋에서 일부를 뽑았는데, 이는 (그림 2)와 같이 구성되어 있다.

D1 = mmap setuid nice audition
D2 = setuid audition chdir access mkdir
D3 = audit chdir access mmap
D4 = setuid audition access chmod
D5 = mkdir setuid audit chdir access
D6 = audition chmod nice setuid

X1 = audit chdir access mmap
X2 = setuid chdir audition access
X3 = chdir access mkdir nice

그림 2. 트레이닝 데이터 셋과 테스트 프로세스
Fig. 2. Training data set and test processes

트레이닝 데이터 셋은 D1부터 D6 등 6개의 프로세스들로 구성되어 있고, 테스트 데이터 셋은 X1, X2, X3 등 3개의 프로세스들로 구성되어 있다. 트레이닝 데이터 셋과 테스트 데이터 셋의 시스템 호출을 제안하는 알고리듬에 적용하기 위해 변환을 하면 (그림 3)과 같다.

D1 = 〈 R P S C 〉
D2 = 〈 P C D A Q 〉
D3 = 〈 B D A R〉
D4 = 〈 P C A E〉
D5 = 〈 Q P B D A 〉
D6 = 〈 C E S P〉

X1 = 〈 B D A R 〉
X2 = 〈 P D C A 〉
X3 = 〈 D A Q S 〉

그림 3. 트레이닝 데이터 셋과 테스트 프로세스 (변환 후)
Fig. 3. Training data set and test processes (after transformation)

세 개의 테스트 프로세스에 대하여, 이들이 정상적인지 비정상적인 분류를 하면 다음과 같다. 여기서는 k=3을 사용하였고, threshold로 0.5를 사용하였다.

먼저, X1의 경우에는 (그림 1)에서 sim(X1, D3) = 1이 되므로, 정상적인 행위로 판정이 된다.

X2의 경우에는 sim(X2, D1) = 1/6, sim(X2, D2) = 5/8, sim(X2, D3) = 1/6, sim(X2, D4) = 1/2, sim(X2, D5) = 3/8, sim(X2, D6) = 0이 된다. 그러므로 최근접 이웃들은 D2, D4, D5가 된다. 이들 최근접 이웃들과의 유사도 평균은 1/2이 되고, 이는 threshold 이상이 되므로 정상적인 행위로 판정이 된다.

X3의 경우에는 sim(X2, D1) = 0, sim(X2, D2) = 3/8, sim(X2, D3) = 1/6, sim(X2, D4) = 0, sim(X2, D5) = 1/8, sim(X2, D6) = 0 이 된다. 그러므로 최근접 이웃들이 D2, D3, D5가 된다. 이들 최근접 이웃들과의 유사도 평균은 2/9이 되고, 이는 threshold 보다 작으므로 비정상적인 행위로 판정이 된다.

V. 결론

본 논문에서는 침입 탐지 시스템에 k-nearest neighbour 분류 방법에 기반을 둔 새로운 알고리듬을 제안하였다. 여기서는 텍스트 마이닝 기법으로 유명한 k-Nearest Neighbour(kNN) 분류기를 이용하여 새로운 프로그램의 행동들을 정상적인지 비정상적인지 구분하는 알고리듬을 제안하였다. 시스템 호출은 '단어'로, 각각의 프로세스들은 '문서'로 취급되며, kNN 방법을 이용하여 침입 탐지에 적용하였다. 또한, 간단한 예제를 통하여 본 논문에서 제안한 알고리듬의 절차를 소개하였다.

추후 연구과제로는 kNN 방법의 신뢰성을 평가하는 것이 필요하다. 특히, 분류를 위해 가장 적절한 시스템 호출들만을 선택하는 방법과 kNN 방법과 다른 기계 학습 방법들과의 비교가 필요하다.

참고문헌

[1] C. Ko, G. Fink and K. Levitt, "Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring", Proceedings of 10th Annual Computer Security Applications Conference, FL, pp 134-144, 1994.

[2] S. Forrest, S. A. Hofmeyr, A. Somayaji and T. A. Logstaff, "A Sense of Self for Unix Process", Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, pp 120-128, 1996.

[3] S. Forrest, S. A. Hofmeyr and A. Somayaji, "Computer Immunology", Communications of the ACM, Vol. 40, pp 88-96, 1997.

[4] W. Lee, S. J. Stolfo and P. K. Chan, "Learning Patterns from Unix Process Execution Traces for Intrusion Detection", Proceedings of AAAI97 Workshop on AI Methods in Fraud and Risk Management, pp 50-56, 1997.

[5] C. Warrender, S. Forrest and B. Pearlmutter, "Detecting Intrusions Using System Calls: Alternative Data Models", Proceedings of 1999 IEEE Symposium on Security and Privacy, pp 133-145, 1999.

[6] M. Asaka, T. Onabuta, T. Inoue, S. Okazawa and S. Goto, "A New Intrusion Detection Method Based on Computer Audit Data", IEEE TRANS. Information & System, Vol. E84-D, pp 570-577, 2001.

[7] M. Deshpande and G. Karypis, "Evalutation of Techniques for Classifying Biological Sequences", PAKDD 2002, Taiwan, 2002.
[8] A. Juan and E. Vidal, "On the Use of Normalized Edit Distance and an Efficient k-NN Search Technique (k-AESA) for Fast and Accurate String Classification", Int'l Conf. on Pattern Recognition, Spain, 2000.
[9] 최인수, 차홍준, "대규모 네트워크를 위한 침입탐지결정 모듈 설계," 컴퓨터정보학회 논문지, 제7권, 제2호, 2002.
[10] 김강, 전종식, "보안정책 기반 침입탐지 시스템 모델 설계", 컴퓨터정보학회 논문지, 제8권, 제4호, 2003.
[11] 오승준, "범주형 시퀀스 데이터의 K-Nearest Neighbour 알고리즘", 컴퓨터정보학회 논문지, 제10권, 제2호, 2005.

저 자 소 개

오 승 준

2004년 8월 한양대학교 산업공학과 공학박사
2005~현재 경기공업대학 산업경영시스템과 교수
〈관심분야〉 데이터마이닝, 인공지능

원 민 관

2003년 8월 청주대학교 무역학과. 경영학박사
2005~현재 경기공업대학 e-비즈니스과 교수
〈관심분야〉 전자무역, e-비즈니스