# 침입 사례를 포함하는 훈련 데이터를 이용한 침입 탐지

이재흥° 박용수 이영기 조유근
서울대학교 전기·컴퓨터공학부
{jhlee°, yspark, yklee, cho}@ssrnet.snu.ac.kr

Intrusion detection using training data with intrusion instances

Jaehung Lee° Youngsu Park Youngki Lee Youkun Cho
Dept. of Computer Science and Engineering, Seoul National University

요 약

침입 탐지 시스템에 이상 탐지 기법(anormal detection)을 적용할 때 정상적인 시스템호출 순서에 대한 훈련이 필요하다. 이 때 발생하는 가장 큰 문제점중 하나는 침입 없는 훈련 데이터의 확보이다. 훈련 데이터에 침입이 있으면 이 침입을 정상으로 간주해서 이후에 같은 침입이 일어나도 이를 탐지해 내지 못하기 때문이다. 하지만, 침입 없는 훈련 데이터를 얻는 것은 매우 어렵다. 본 논문에서는 훈련 데이터에 침입이 포함되어 있더라도 효과적으로 침입을 탐지할 수 있는 시스템 호출 기반 침입 탐지 기법을 제안한다. 제안 기법은 훈련 데이터에 침입이 존재할 경우 침입 부분에서 빈도가 매우 적은 데이터들이 연속적으로 나타나는 성질을 이용한다. 이를 위해 훈련 데이터를 일정 개수씩 블록으로 묶은 뒤 평균 빈도를 계산해서 그 값이 임계치보다 작은 경우 이를 침입 데이터로 간주하여 훈련 데이터에서 제외하는 방법을 사용하였다. 실험 결과 블록 크기를 적절하게 잡았을 경우 기존의 Eskin 기법보다 향상된 결과를 얻을 수 있었다.

## 1. 서 론

인터넷이 널리 사용되면서 늘어나는 시스템 침입에 대응하기 위해 침입 탐지 기술이 최근 활발히 연구되고 있다. 침입 탐지는 침입을 판단하는 방법에 따라 오용 탐지와 이상 탐지로 나뉘어 지는데, 현재 널리 사용되는 오용 탐지는 새로운 공격을 탐지하지 못하는 단점을 가진다. 이를 보완하기 위해 최근 이상 탐지가 많이 연구되고 있다.

기존의 이상 탐지 방법들은 침입을 판단하기 위한 모델을 만들기 위해 정상 상태일 때 수집한 데이터를 필요로 한다. 그러나 이것은 다음과 같은 문제점들을 가진다 [3]. 첫째, 깨끗한 데이터를 얻기가 쉽지 않다. 둘째, 만약 훈련 집합에 침입 데이터가 숨겨져 있을 경우, 심각한 결과를 야기할 수 있다. 셋째, 깨끗하다고 보장된 데이터로 훈련을 해야 하기 때문에, 온라인상에서 적응력 있는 시스템을 만드는 것이 어려워진다.

본 논문에서는 훈련 데이터에 침입이 포함되어 있더라도 효과적으로 침입을 탐지할 수 있는 시스템 호출 기반 침입 탐지 기법을 제안한다. 제안 기법은 훈련 데이터에 침입이 존재할 경우 침입 부분에서 빈도가 아주 적은 데이터들이 연속적으로 나타나는 성질을 이용한다. 이를 위해 훈련 데이터를 일정 개수씩 블록으로 묶은 뒤 평균 빈도를 계산해서 그 값이 임계치보다 작은 경우 이를 침입 데이터로 간주하여 훈련 데이터에서 제외하는 방법을 사용하였다. 실험 결과 블록 크기를 적절하게 잡았을 경우 기존의 stide나 t-stide, Eskin방법보다 향상된 결과를 얻을 수 있었다.

## 2. 관련연구

### 2.1 stide

stide[1]기법은 두 단계로 구성된다. 첫 번째 단계에서는 정상일 때 불리는 시스템 호출들에 대한 추적을 통해 정상 행동을 표현하는 데이터베이스를 만들고, 두 번째 단계에서는 실제 프로세스의 행동이 앞에서 만든 데이터베이스에 표현된 정상 행동과 얼마나 다른지를 통해 침입을 탐지한다.

예1. 슬라이딩 윈도우의 크기가 3이고, 시스템 호출들이 다음과 같은 순서로 불렸다고 하자.

open, read, mmap, mmap, open, read, mmap

stide의 첫 번째 단계가 끝난 후, 다음과 같은 정상 데이터베이스가 생성된다.

open, read, mmap
read, mmap, mmap
mmap, mmap, open
mmap, open, read

stide에서는 트리 구조를 채택하여 탐색 시간과 데이터베이스의 크기를 줄였다. 위의 데이터베이스를 트리 형태로 표현한 것이 그림 1이다. 우리는 이 트리를 "시스템 호출 수열 트리(system call sequence tree)"라 부른다.

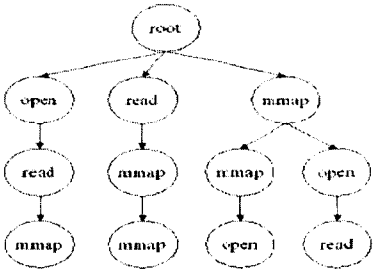


그림 1 시스템 호출 수열 트리의 예

stide는 일정 개수의 연속된 수열(locality frame; 지역 프레임)에서 몇 개의 미스매치가 존재하는 가를 통해 침입을 판단한다. 즉, 지역 프레임에 존재하는 미스매치의

수가 일정 개수(locality frame count: 지역 프레임 카운트) 이상이면, 침입이 일어난 것으로 판단한다. 예를 들어 지역 프레임의 크기가 20이고, 지역 프레임 카운트의 크기가 5라면, 최근의 시스템 호출 수열 20개 중에서 5개 이상이 미스매치이면 이를 침입으로 판단한다.

### 2.2 t-stide

stide에서는 정상 데이터베이스에 없는 시스템 호출 수열만을 미스매치로 간주하였지만, t-stide[2]에서는 정상 데이터베이스에 없는 시스템 호출 수열뿐만 아니라 드문 수열(rare sequence)도 미스매치로 간주한다. 드문 수열이란 정상 데이터베이스에 있으면서 훈련 과정 중에 나타난 빈도가 일정 비율보다 낮은 수열로서, 일반적으로 전체 훈련 집합의 0.001%보다 낮은 빈도로 나타난 수열들이 드문 수열로 간주된다[2,3].

예2. 훈련 과정이 끝나고 나서, 그림 2과 같은 확장 시스템 호출 수열 트리의 일부가 만들어졌다고 하자. 여기서 시스템 호출 수열 `A B B C'와 `B B C C', 그리고 `C C A A'의 빈도가 0.001%보다 낮으므로 이 부분은 제거된다. 그 결과 확장 시스템 호출 수열 트리는 그림 3과 같이 된다.

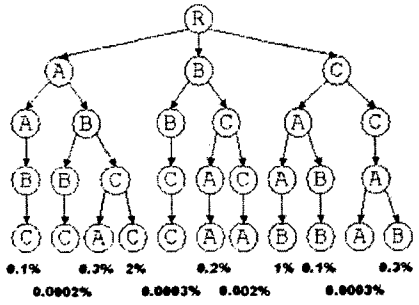


그림 2 확장 시스템호출 수열트리의 예

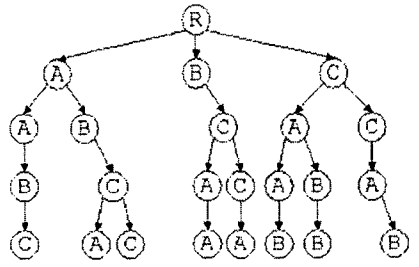

그림 3 침입 탐지에 사용되는 수열 트리의 예

## 3. 제안 기법

### 3.1 lft-stide

본 논문에서 제안하는 lft-stide는 침입과 관련된 다음과 같은 세 가지의 가정을 기반으로 한다.

- 침입 시 만들어지는 데이터들은 정상 상태일 때 만들어지는 데이터와 많이 다르다.
- 훈련 데이터의 대부분은 정상 상태일 때 만들어진 것이고, 침입 데이터는 없거나 아주 적다.
- 침입 데이터는 이곳 저곳에 분산되어 있지 않고, 대부분 서로 모여 있다.

lft-stide의 정상 데이터베이스는 다음 두 단계를 통해 만들어진다. 첫 번째 단계에서는 t-stide에서와 같이 훈련 데이터에 어떠한 수열들이 얼마나 자주 나타나는 지를 살펴봄으로서, 그림 2와 같은 확장 시스템 호출 수열 트리를 만든다. 두 번째 단계에서는 처음부터 훈련 데이터를 다시 추적하면서 수열들을 일정 개수씩 블록으로 묶은 뒤에, 앞에서 만든 확장 시스템 호출 수열 트리를 참고하여 각 블록의 평균 빈도를 계산한다. 그리고 나서 평균 빈도가 일정 비율보다 높은 블록만을 사용하여 stide 방식의 훈련을 수행한다. 그러므로 블록 크기가 1일 경우에는 t-stide가 된다.

예3. 첫 번째 단계의 훈련이 끝난 뒤, 그림 2와 같은 확장 시스템 호출 수열 트리의 일부가 만들어졌다고 하자. 그리고 훈련 데이터의 앞부분이 다음과 같이 시작한다고 하자.

A B C C A B B C C A A B C C A A B C C A A B C A A B C C

기준이 되는 임계값이 0.001%이고, 블록 크기를 4로 한다면, 그림 4와 같은 과정이 이루어진다.

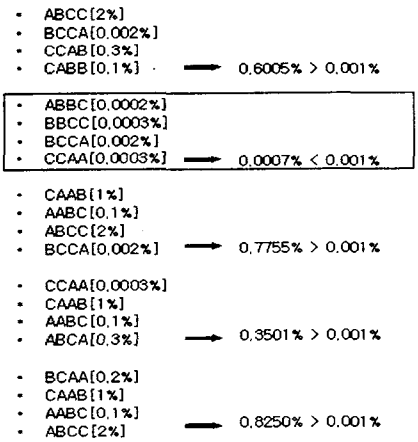


그림 4 블록 빈도 계산에 따른 분류

위의 그림 4에서는 `A B B C', `B B C C', `B C C A', 그리고 `C C A A', 이렇게 4 개의 시스템 호출 수열이 훈련에 사용되지 않았다. 하지만 t-stide에서와는 다르게 lft-stide에서는 한 곳에서 훈련에 사용되지 않는다고 해서 침입 탐지에 사용되는 시스템 호출 수열 트리에 나타나지 않는 것은 아니다. `B C C A'나 `C C A A' 같은 경우는 다른 블록에서 지워지지 않고 남았기 때문에 침입 탐지에 사용되는 시스템 호출 트리에 남아있게 된다. 한 시스템 호출 수열이 침입 탐지에 사용되는 시스템 호

출 수열 트리에서 사라지기 위해서는 그 수열이 있는 모든 블록이 훈련 데이터에서 사라져야만 한다.

그림 5는 그림 4에 나타난 과정이 이루어진 뒤에 만들어진 시스템 호출 수열 트리를 보여준다.

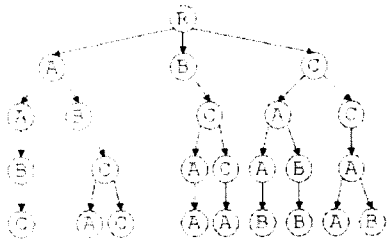

그림 5 침입탐지에 사용되는 수열 트리

## 4. 실험 및 분석

우선 침입이 포함된 훈련 데이터를 사용할 때의 성능을 측정한다. 이를 위해 정상 데이터와 침입 데이터를 섞어서 훈련하고 나서, 침입이 일어났을 때 이를 탐지해 낼 수 있는지(detection rate)와 정상 데이터를 침입으로 착각하지 않는지(false positive rate)를 조사하였다. 슬라이딩 윈도우의 크기는 6으로 하였고, 크기 20인 지역 프레임과 지역 프레임 카운트를 사용하여 침입을 판단하였다. 즉, 20번 연속으로 미스매치가 일어났을 때에만 침입이 일어난 것으로 간주한 것이다.

| 탐지방법(블럭크기) | 침입 탐지율 | 거짓 탐지율 |
|---|---|---|
| Eskin | 1.0 | 0.000028 |
| stide | 0 | 0 |
| lft-stide(1) | 1.0 | 0.000209 |
| lft-stide(10) | 1.0 | 0.000168 |
| lft-stide(30) | 1.0 | 0 |
| lft-stide(50) | 1.0 | 0 |
| lft-stide(100) | 1.0 | 0 |
| lft-stide(300) | 1.0 | 0 |
| lft-stide(500) | 0 | 0 |

표 1 침입데이터가 포함된 경우(xlock)

| 탐지방법(블럭크기) | 침입 탐지율 | 거짓 탐지율 |
|---|---|---|
| Eskin | 1.0 | 0.000050 |
| stide | 0 | 0 |
| lft-stide(1) | 1.0 | 0.000149 |
| lft-stide(10) | 1.0 | 0.000016 |
| lft-stide(30) | 1.0 | 0.000005 |
| lft-stide(50) | 1.0 | 0.000004 |
| lft-stide(100) | 1.0 | 0 |
| lft-stide(300) | 0 | 0 |
| lft-stide(500) | 0 | 0 |

표 2 침입데이터가 포함된 경우(named)

표 1와 표 2은 훈련 데이터에 침입이 들어있을 때의 침입 탐지 방법에 따른 침입 탐지율과 거짓 탐지율을 보여주고 있다. stide는 훈련 데이터에 침입이 들어있으면 같은 침입이 들어왔을 때 이를 전혀 탐지해내지 못한다. 본 논문에서 제안한 lft-stide는 블록 크기가 작으면 기존의 Eskin의 방법보다 더 높은 거짓 탐지율을 가지지만, 블록 크기가 커지면서 거짓 탐지율도 작아지게 된다. 하지만 너무 크면 stide와 같이 침입을 전혀 탐지해내지 못하게 된다. 이 때 적절한 블록 크기를 정하는 것은 lft-stide에서 중요한 요소이다. 제안된 기법으로 실험한 결과 블록 크기가 10이상이고 100이하 일 때 높은 침입 탐지율을 보이는 반면 거짓 탐지율은 0.000168로 tstide(블럭크기가 1인 lft-stide)보다 낮은 수치를 보였다. 또한 블록 크기가 30이상 일 때는 거짓 탐지율에서 Eskin의 방법보다 더 좋은 성능을 보여준다.

## 5. 결론

본 논문에서는 훈련 데이터에 침입이 포함되어 있더라도 효과적으로 침입을 탐지할 수 있는 시스템 호출 기반 침입 탐지 기법인 lft-stide를 제안했다. lft-stide는 훈련 데이터에 침입이 존재할 경우 침입 부분에서 빈도가 아주 적은 데이터들이 연속적으로 나타나는 성질을 이용한다. 이를 위해 훈련 데이터를 일정 개수씩 블록으로 묶은 뒤 평균 빈도를 계산해서 그 값이 임계치보다 작은 경우 이를 침입 데이터로 간주하여 훈련 데이터에서 제외하는 방법을 사용하였다. lft-stide의 성능을 평가하기 위해, xlock과 named 프로그램에서 추출된 침입이 포함된 훈련 데이터와 침입이 포함되지 않은 훈련 데이터를 사용하여 실험을 수행했다. 실험 결과, lft-stide는 블록 크기가 100일 때 침입이 포함된 훈련 데이터에 대해 stide, t-stide 뿐만 아니라 본 논문과 같은 목적을 가지고 만들어진 Eskin의 이상 탐지 방법보다도 나은 성능을 보였다.

## 6. 참고문헌

[1] S.A. Hofmeyr, S. Forrest "Intrusion detection using sequences of system calls," Journal of Computer Security, vol.6, no.3, pp.151-180, 1998.

[2] C.Warrender, S.Forrest, and B.A. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in IEEE Symposium on Security and Privacy}, pp.133-145, 1999.

[3] E. Eskin, "Anomaly detection over noisy data using learned probability distributions," in Proc. 17th International Conf. on Machine Learning, pp.255-262, Morgan Kaufmann, San Francisco, CA, 2000.

[4] CERT/CC, "CERT/CC statistics 1988-2002." http://www.cert.org/stats/cert_stats.html, 2002.

[5] W.Lee, S.Stolfo, "A data mining framework for building intrusion detection models," in IEEE Symposium on Security and Privacy, pp.120-132, 1999.

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