침입탐지를 위한 X^2 거리기반 다변량 분석기법을 이용한 프로그램 행위 프로파일링

김 정 일[†] · 김 용 민[††] · 서 재 현[†††] · 노 봉 남[††††]

요 약

프로그램 행위기반 침입탐지 기법은 데몬 프로그램이나 루트 권한으로 실행되는 프로그램이 발생시키는 시스템 호출들을 분석하고 프로그램 행위 프로파일을 구축하여 잠재적인 공격을 효과적으로 탐지한다. 그러나 각 프로그램마다 매우 큰 프로파일이 구축되어야 하는 문제점이 있다. 본 논문은 프로파일의 크기를 줄이기 위해, 프로그램 행위 프로파일링 및 이상행위 탐지에 X^2 거리기반 다변량 분석 기법을 응용하였다. 실험 결과, 프로파일을 비교적 작게 유지하면서 탐지율에서는 의미있는 결과를 보였다.

Profiling Program Behavior with X^2 distance-based Multivariate Analysis for Intrusion Detection

Chong-Il Kim[†] · Yong-Min Kim[††] · Jae-Hyeon Seo[†††] · Bong-Nam Noh[††††]

ABSTRACT

Intrusion detection techniques based on program behavior can detect potential intrusions against systems by analyzing system calls made by demon programs or root-privileged programs and building program profiles. But there is a drawback : large profiles must be built for each program. In this paper, we apply X^2 distance-based multivariate analysis to profiling program behavior and detecting abnormal behavior in order to reduce profiles. Experiment results show that profiles are relatively small and the detection rate is significant.

키워드 : 침입탐지(Intrusion Detection), X^2 거리기반 다변량 분석 기법(X^2 Distance-based Multivariate Analysis), 프로그램 행위(Program Behavior)

1. 서 론

안전한 네트워크 아키텍쳐 설계, 안전한 프로그램 설계, 주의깊게 구성된 네트워크 서비스들, 방화벽, 사용자 감시 등 공격을 방지하는 수많은 대책들이 활용됨에도 불구하고 침입은 여전히 성공적으로 이루어지고 있다. 예를 들면, 내부자 공격이나 악의적인 이동 코드는 대부분의 보안 방어책을 뚫고 수행된다. 특히 대부분의 공격은 소프트웨어의 잘못된 설정이나 버그로 인해 가능해진다. 따라서 사용자, 네트워크 또는 컴퓨터 시스템의 행위를 모니터링하여 컴퓨터 시스템을 대상으로 수행되는 공격을 탐지하는 침입탐지 시스템이 요구되어진다[1, 2].

침입탐지 기법은 오용탐지(misuse detection) 기법과 이상행위 탐지(anomaly detection) 기법으로 구분할 수 있다. 오용탐지 기법은 공격의 특정 패턴을 명시하여, 이 패턴에 해당하는 공격을 탐지하는 방법으로 이미 알려진 공격을 탐지하는 데는 효과적이지만 새로운 공격을 탐지할 수는 없다. 반면에 이상행위 탐지기법은 정상적인 행위를 프로파일링하여, 이 프로파일로부터 벗어나는 이상행위를 측정함으로써 공격을 탐지한다. 따라서 이상행위 탐지기법은 새로운 공격을 탐지할 수 있는 방법이다. 그러나 이 기법은 오탐율이 높고, 프로파일 구축시 이용되는 데이터에 공격이 포함되는 경우 대처할 수 없고, 발생한 공격의 정확한 종류를 식별할 수 없다는 문제점이 있다. 그러나 오용탐지 기법은 새로운 공격을 탐지하기가 불가능하고 또한 변형된 공격을 탐지하기가 어렵고 공격 유형을 분석하여 인코딩(encoding)하는 작업에 시간이나 비용이 많이 소요되기 때문에, 최근에는 이상행위 탐지 기법을 중심으로 연구가 진행되고 있다[2, 3].

최근에 이상행위 탐지 기법은 사용자 행위에서 벗어나 데몬 프로그램이나 루트 권한으로 실행되는 프로그램들의 행위를 분석하고 프로파일을 구축하여 이상행위를 탐지하는

※ 본 연구는 대학 IT 연구센터 육성/지원사업의 연구결과로 수행되었음.
† 준 회 원 : 전남대학교 대학원 전산학과
†† 정 회 원 : 전남대학교 리눅스시스템 보안연구센터 Post-doc.
††† 정 회 원 : 목포대학교 정보공학부 교수
†††† 종신회원 : 전남대학교 컴퓨터정보학부 교수
논문접수 : 2003년 2월 19일, 심사완료 : 2003년 6월 11일

프로그램 행위 기반 탐지기법이 연구되고 있다[4-7]. 데몬 프로그램이나 루트 권한으로 실행되는 프로그램 행위는 폭넓은 사용자 행위에 비해 매우 제한적이며, 시간이 지남에 따라 훨씬 그 변화가 없기 때문이다. 또한 UNIX 운영체제 환경에서 루트 권한으로 수행되는 프로그램들은 주로 일반 사용자들이 접근할 수 없는 시스템 자원들에 대해 일반 사용자들에게 서비스를 제공하는데 빈번히 공격의 대상이 되고 있기 때문이다. 즉, 일반 프로그램들은 그것을 실행시킨 사용자의 권한 내에서 시스템 자원을 접근하는 데 반해, 이들 프로그램은 특정 서비스와 관련된 자원뿐만 아니라 시스템 전체 자원을 접근할 수 있다. 따라서 이러한 프로그램에 프로그램 오류가 있거나 구성이 적절하지 않은 경우, 이들 문제점들을 통해 일반 사용자는 루트 권한을 획득할 수 있기 때문이다.

프로그램 행위 기반 침입탐지 기법의 전제는 대부분의 공격은 프로그램의 오용 때문에 가능하다는 것이고 프로그램이 오용될 때는 그것의 정상적인 사용과는 그 행위가 다르다는 데 있다. 따라서 프로그램의 행위가 적합하게 표현될 수 있다면 침입 탐지를 위해 행위적인 특성들이 이용될 수 있다. 특히 N-gram 기법[7-9]은 프로그램에 의해 발생하는 시스템 호출들을 고정된 길이의 시퀀스로 분할하고 이 시퀀스들을 정상 행위로 간주하여 데이터베이스를 구축한다. 만약 새로운 시퀀스가 데이터베이스에 존재하지 않으면 이상 행위로서 간주한다. N-gram 기법은 알고리즘이 단순하고, 높은 탐지율(detection rate)을 가지지만 프로파일 데이터베이스가 매우 커지기 때문에 저장 공간의 요구량이 많고 실행 시간에도 오버헤드가 발생한다[4].

본 논문은 프로파일의 크기를 작게 유지하면서 탐지율을 높이기 위해 X^2 거리기반 다변량 분석 기법을 응용하였다. 이 기법에서는 프로그램이 정상적으로 동작할 때 발생시키는 각 시스템 호출들의 발생 빈도를 측정하여 평균적인 발생 빈도 분포를 그 프로그램의 행위 프로파일로 구축하고, X^2 거리기반 다변량 분석 기법을 이용하여 프로파일에서 많이 벗어나는 시스템 호출들을 판별하여 프로그램의 세션별로 이상행위를 탐지한다. 성능 평가를 위해 sendmail 프로그램이 정상적으로 동작할 때 발생하는 시스템 호출들과 공격을 받았을 때 발생하는 시스템 호출들에 대해 실험 결과를 제시하고 N-gram 기법과 비교하여 분석하였다. 실험 결과, 프로파일의 크기는 크게 감소시키면서 N-gram 기법보다 다소 높은 탐지율을 보였다.

본 논문의 구성은 다음과 같다. 2장에서는 프로그램 행위 기반 침입탐지 시스템들을 살펴보고, 3장에서는 X^2 거리기반 다변량 분석 기법을 응용한 프로그램 행위 프로파일링 기법과 비정상행위 탐지 방법을 제시한다. 4장에서는 제시한 방법과 N-gram 방법의 실험 결과를 제시하고 분석한다. 5장에서는 결론 및 향후 연구방향을 제시한다.

2. 관련 연구

침입탐지를 위해 프로그램 행위를 분석하여 프로파일을 구축하는 기법들은 사용자 행위 기반 침입탐지 기법의 대안으로 연구되어 왔다. 프로그램 행위 프로파일은 정상적인 운용환경에서 프로그램이 발생시키는 시스템 호출들을 수집 및 분석하여 구축된다. 만약 프로그램의 정상행위가 간결하고 정확하게 표현된다면, 정상행위와는 다른 행위, 즉 프로그램이 공격을 위해 오용되었을 때 나타나는 행위들을 탐지하는데 이용될 수 있다.

명세기반 접근방법(specification-based approach)[10, 11]에서는 ASL(auditing specification language)를 이용하여 프로그램의 정상행위를 정의한다. 즉, 수행이 허용되는 프로그램의 오퍼레이션(operation)을 규칙으로 명시하고, 이 규칙을 벗어나는 행위가 발생할 때 침입으로 판정한다. 그러나 프로그램 행위는 매우 다양하고 변하기 쉽기 때문에 프로그램의 수행 정책 또는 모델을 상세히 명시하기가 어렵다.

정적 분석기법(static analysis)[12]에서는 프로그램 행위를 모델링하기 위해 모든 가능한 시스템 호출들을 도출하여 정적인 모델을 구성한다. 즉, 프로그램이 발생시키는 시스템 호출들은 프로그램의 소스 코드(source code)와 일치한다고 가정하여, 프로그램의 소스코드로부터 정적으로 프로그램 행위를 구성하는 시스템 호출들을 도출하여 이상행위를 판정하는데 이용한다. 그러나 이 기법은 프로그램의 정상적인 수행에 따르는 모든 가능한 시스템 호출들을 도출하는 게 현실적으로 힘들고, 특히 실행시간의 오버헤드가 매우 높다.

프로그램의 정상 행위를 자동적으로 생성하고 정의하기 위한 대표적인 것은 뉴 멕시코 대학의 Forrest 연구팀에서 개발한 N-gram 기법이다[7-9]. 이 기법은 면역학(immunology)의 개념들을 응용하여 침입탐지에 적용하였는데, 프로그램이 발생시킨 일정 길이의 일련의 시스템 호출들, 즉 N-gram 또는 스트링(string)으로 프로파일 데이터베이스를 구축한다. 프로파일 데이터베이스가 구축된 후, 프로그램이 발생시킨 시스템 호출들 중에서, 특정 길이의 일련의 시스템 호출들이 프로파일에 존재하지 않는다면 비정상행위로 간주하여 개수를 센다. 세션 내의 총 스트링 개수에 대해 비정상행위로 간주된 스트링의 개수의 비율(percentage)이 매우 크다면, 그 세션을 비정상으로 판정한다. Forrest 연구팀에서는 이 기법을 UNIX 프로그램중에서 루트권한으로 실행되는 주요 데몬 프로그램들, 즉 sendmail, ftpd, inted 등에 적용하여 높은 탐지율을 보였다. 그러나 이 기법은 프로그램마다 매우 큰 프로파일이 필요하다는 문제점이 있다[4, 6].

N-gram 기법을 기반으로 탐지율을 더 높이기 위한 방법들로는 Teiresias 알고리즘을 이용하여 가변 길이의 스트링

을 발굴하고 프로파일 데이터베이스를 구축하여 이상행위를 탐지하는 방법[13]과 RIPPER를 이용하여 스트링의 각 위치마다 시스템 호출들의 발생확률을 측정하여 이상행위를 탐지하는 방법[14]등이 있다. 또한 로그인 세션 중에서 비정상적인 세션을 효율적으로 판정하기 위해 침입탐지 기법에 다변량 분석 기법을 도입한 방법[15]이 있다.

본 논문에서는 N-gram 기법의 탐지율을 떨어뜨리지 않으면서, N-gram 기법의 문제점인 프로파일의 크기를 줄이기 위해 시스템 호출들의 발생 확률에 유의하여 X^2 거리기반 다변량 분석 기법을 응용하였다.

3. 다변량 분석을 이용한 프로그램 행위 프로파일링

프로그램의 특정 기능들은 최종적으로 시스템 호출들을 이용하여 구현된다. 따라서 프로그램의 행위는 일련의 시스템 호출들로 구성된다고 볼 수 있으며, 각 시스템 호출들은 그 종류로서 특징지어진다. 만약 정상 행위에서 벗어난 이상행위를 프로그램이 수행한다면, 정상행위를 구성하는 시스템 호출들의 발생 빈도가 보이는 분포와는 다르게 시스템 호출들이 발생될 것이다. 이 장에서는 프로그램의 행위를 시스템 호출들의 발생 빈도에 주목하여, 그 기대되는 발생 빈도를 간결하게 벡터 형태로 표현하고, X^2 거리기반 다변량 분석 기법을 응용하여 기대되는 행위에서 벗어나는 이상행위를 탐지하는 기법을 기술한다.

3.1 X^2 거리기반 다변량 분석 기법

효과적인 공정 관리(process control)를 위해 다변량 분석 기법을 활용한 공정 관리 기법이 개발되어 이용되고 있으며, 대표적으로 Hotellings's T^2, MCUSUM(multivariate cumulative sum), MEWMA(multivariate exponentially weighted moving average), X^2 거리기반 다변량 분석 기법(X^2 distance-based multivariate analysis) 등이 있다[15-17]. 이론적으로 다변량 공정 관리 기법(multivariate process control techniques)은 컴퓨터 시스템에서 프로그램의 이상행위(anomaly)를 탐지하는 것과 비슷하기 때문에 적용가능성이 있다. 특히 X^2 거리기반 다변량 분석 기법은 Hotelling's T^2을 비롯한 고전적인 다변량 분석 기법의 계산량을 획기적으로 감소시킨 방법으로서, 정보시스템에서 프로그램에 의해 발생하는 이벤트의 종류가 매우 종류가 많고 발생량이 많다는 점을 고려한다면 다른 방법에 비해 적합하다. X^2 거리는 다음과 같이 정의된다[15, 16].

$$X^2 = \sum_{i=1}^{n} \frac{(X_i - E_i)}{E_i} \tag{1}$$

여기서 X_i는 관측된 i번째 변수이고, E_i는 i번째 변수의 기대치이고, n은 변수의 개수이다. 발생한 이벤트 값이 기대치에 가깝다면 X^2 거리는 작다.

식 (1)에서 기대치 E_i를 학습 데이터에서 계산된 $\overline{X_i}$로 추정한다면, X^2 거리는 다음과 같이 정의된다.

$$X^2 = \sum_{i=1}^{n} \frac{(X_i - \overline{X_i})}{\overline{X_i}} \tag{2}$$

만약 변수의 개수가 충분히 크다면, 즉 30보다 크다면, 중심극한정리(central limit theorem)에 의해, X^2의 값은 정규 분포를 따른다. 그리고 발생한 이벤트가 정상적인지 그렇지 않은지를 결정짓는 관리한계(control limit)는 3-시그마 관리 한계(3-sigma control limits)에 따라 다음과 같이 정의된다[16].

$$\left[\ \overline{X}^2 - 3S_{X^2},\ \overline{X}^2 + 3S_{X^2}\right] \tag{3}$$

이때 X^2의 표본으로부터 X^2의 평균 ($\overline{X}^2$)과 표준편차 (S_{X^2})를 추정한다. 즉, 발생한 이벤트가 위 구간을 벗어나면 이상행위로서 간주된다.

3.2 프로그램 행위 프로파일링

프로그램 행위는 일련의 시스템 호출들로 구성된다. 시스템 호출 정보에는 시스템 호출 종류, 프로세스 번호, 매개변수 등이 포함되는데 프로그램 행위 프로파일을 간결하게 하기 위해 시스템 호출 종류만을 고려한다. 그리고 프로그램이 정상적으로 운용될 때 발생한 시스템 호출들을 종류에 따라 발생 빈도를 측정하기 위해 프로그램이 정상적으로 동작할 때 발생하는 시스템 호출들을 학습 데이터로 구성하고, MEWMA(multivariate exponentially weighted moving average technique)[17]에 따라 학습 데이터내의 각 시스템 호출을 다음과 같이 벡터($X_1, X_2, \cdots, X_n$)로 표현한다. 여기서 n은 서로 다른 종류의 시스템 호출의 개수이다.

① 만약 t번째 발생한 시스템 호출이 i번째 시스템 호출 종류이면,

$$X_i(t) = \lambda * 1 + (1-\lambda) * X_i(t-1) \tag{4}$$

② i번째 시스템 호출 종류가 아니라면,

$$X_i(t) = \lambda * 0 + (1-\lambda) * X_i(t-1)$$

여기서, $X_i(t)$는 t번째 발생한 시스템 호출에 대한 벡터 표현이고, λ는 평활상수(smoothing constant)이고, i = 1 … n 이다. 초기값을 위해 t = 0인 경우, $X_i(0) = 0$이다. 시스템 호출 종류는 플랫폼에 따라 달라지는데 Solaris 2.5의 경우 284개이다. 어떤 특정 프로그램은 플랫폼에서 제공하는 모든 시스템 호출들을 사용하지는 않기 때문에 실제로 고려하는 시스템 호출 종류는 더 줄어든다. λ에 따라 과거에 발

생한 시스템 호출과의 관계를 반영하는데, t번째 발생한 시스템 호출은 가중치 λ, t-1번째 발생한 시스템 호출은 $\lambda(1-\lambda)$, t-k에 발생한 시스템 호출은 $\lambda(1-\lambda)^k$의 가중치가 적용된다. MEWMA에서는 일반적으로 0.3을 이용한다. 결과적으로 $X(t)$는 λ에 의해 결정되는 범위 내에 발생한 시스템 호출들의 발생 빈도를 나타낸다.

장기간에 걸쳐 운용되는 프로그램의 행위를, 위와 같이 벡터 형태로 표현된 시스템 호출들을 이용하여 간략히 표현하기 위해, 발생한 시스템 호출들에 해당하는 벡터들에 대한 발생 빈도 평균 벡터 $\overline{X} = (\overline{X}_1, \overline{X}_2, \cdots, \overline{X}_n)$를 이용한다. 한편, 프로그램의 행위를 구성하는 시스템 호출들은 한꺼번에 모두 발생하는 것이 아니라 장기간에 걸쳐 순차적으로(sequentially) 발생하기 때문에 다음과 같은 공식으로 계속하여 평균벡터를 갱신한다.

$$\overline{X}_{k,i} = \frac{(k-1)\,\overline{X}_{k-1,i} + X_{k,i}}{k} \tag{5}$$

여기서 n은 시스템 호출의 종류이고, i = 1 … n이고 k는 시스템 호출의 발생 지점이다. 즉, 프로그램이 정상적으로 동작할 때 발생하는 시스템 호출들의 평균적인 발생빈도로서 그 프로그램의 프로파일을 구성한다.

3.3 비정상행위 판정

이상행위를 구성하는 시스템 호출들은 프로그램이 정상적으로 동작할 때 발생하는 시스템 호출과는 다를 것이다. 따라서 이러한 시스템 호출들은 정상행위의 시스템 호출들에 비해 정상행위 프로파일, 즉 정상적인 시스템 호출들의 평균적인 발생빈도로부터 매우 거리가 멀 것이다. 프로그램의 기대되는 행위로부터 벗어나는 시스템 호출들을 판정하기 위해 X^2 거리기반 다변량 분석 기법을 적용한다. 식 (2)에서 각 변수를 시스템 호출들의 종류, 기대치를 발생빈도 평균벡터, 관측치를 발생한 시스템 호출로 대체하여 다음과 같이 X^2 거리를 계산한다.

$$X^2 = \sum_{i=1}^{n} \frac{X_i - \overline{X_i}}{\overline{X_i}} \tag{6}$$

여기서, n은 시스템 호출의 종류, $\overline{X_i}$는 i번째 시스템 호출의 평균 발생 빈도, X_i는 i 번째 시스템 호출의 발생 빈도이다.

계산된 X^2이 작다면 발생한 시스템 호출의 벡터는 평균 벡터에 가깝다. 다시 말해 발생한 시스템 호출은 프로그램 행위 프로파일에 가깝다고 할 수 있다. 따라서 정상행위시 발생하는 시스템 호출들의 X^2 거리는 매우 작을 것이고, 이상행위시 발생하는 시스템 호출들은 상대적으로 X^2 거리가 클 것이다. 이때 정상행위의 시스템 호출들의 X^2 거리와 비정상행위의 시스템 호출들의 X^2 거리를 구분짓는 임계치가 필요하다. 본 논문에서는 이상행위를 구성하는 시스템 호출들은 X^2 거리가 매우 크다고 가정하기 때문에 식 (3)에서 다음과 같이 상한 관리한계(upper control limit)만을 고려한다.

$$\overline{X}^2 + 3S_{X^2} \tag{7}$$

즉, 발생한 시스템 호출을 프로파일과 비교하여, X^2 거리가 $\overline{X}^2 + 3S_{X^2}$ 이상이면, 그 시스템 호출은 비정상 시스템 호출로 간주한다.

한편 일반적으로 공격 행위는 단지 하나의 시스템 호출로서 이루어지지 않는다. 즉, 악의적인 행위는 관련된 일련의 시스템 호출들로 구성된다. 따라서 응용 프로그램의 시작에서 종료까지 또는 데몬 프로그램의 경우, 하나의 서비스 요청의 처리 과정을 하나의 세션으로 규정할 때, 하나의 시스템 호출이 비정상적이라고 하여 전체 세션이 비정상적이라고 할 수 없다.

따라서 SSR(session signal ratio)을 다음과 같이 정의하여 세션이 정상적인지 비정상적인지를 판정한다.

$$SSR(S) = \frac{\text{세션 } S\text{의 비정상 시스템호출 개수}}{\text{세션 } S\text{의 총 시스템호출 개수}} \tag{8}$$

만약 세션 S에 공격이 포함되었다면, 그 공격을 구성하는 일련의 시스템 호출들이 비정상적으로 보일 것이고, 따라서 SSR은 높은 수치를 보일 것이다. 따라서 SSR이 최종 이상행위 임계값(anomaly score threshold) α보다 크면 그 세션은 비정상적이고, 그렇지 않다면 그 세션은 정상적이다.

4. 실험 결과 및 분석

X^2 거리기반 다변량 분석기법을 이용한 프로그램 행위 프로파일링의 성능을 평가하기 위해 뉴 멕시코 대학의 sendmail 데몬 프로그램의 시스템 호출 자료를 대상으로 실험하였고, N-gram 기법과 비교 분석하였다.

4.1 실험 데이터

뉴 멕시코 대학(University of New Mexico)의 Forrest 연구팀에서는 UNIX 기반 운영체제(특히, Solaris 2.5)에서 루트 권한으로 실행되고 빈번히 침입 공격의 통로로 이용되는 주요 프로그램들, 즉 sendmail 데몬, ftpd, lpd, named, inetd 등에 대해서, 이들 프로그램들이 운영되는 도중 발생하는 시스템 호출들을 추적하여 데이터베이스로 구축하였다[18].

특히 sendmail 데몬 프로그램의 시스템 호출 자료는, 정상적으로 동작하는 환경에서 일어날 수 있는 모든 경우를 고려하여 시스템 호출들을 수집하고, 또한 sendmail 데몬 프로그램에 대한 다양한 공격이 이루어질 때 발생하는 시

스템 호출들을 수집하여 구축되었다. 따라서, 본 논문에서는 sendmail 데몬 프로그램의 시스템 호출들을 대상으로 실험하였다.

〈표 1〉 Sendmail 데몬 프로그램의 시스템 호출 자료

정상/공격 유형	Trace	시스템 호출 횟수	세션 개수
Normal	UNM-sendmail-daemon	1,571,583	52
	Cert-sendmail-daemon	1,556,560	52
sunsendmailcp intrusion	sm-10763	373	1
	sm-10801	373	1
	sm-10814	373	1
decode intrusion	sm-280	1,534	1
	sm-314	1,533	1
forwarding loops	fwd-loops-1	634	1
	fwd-loops-2	489	1
	fwd-loops-3	557	1
	fwd-loops-4	635	1
	fwd-loops-5	254	1
local syslogd intrusion	syslog-local-1	1,516	1
	syslog-local-2	1,574	1
remote syslogd intrusion	syslog-remote-1	1,861	1
	syslog-remote-2	1,553	1
sunOS 4.1.3 vulnerability	sm5x	1,537	1
	sm565a	275	1

<표 1>은 sendmail 데몬 프로그램의 시스템 호출 자료를 정리한 것이다. 시스템 호출 자료는 크게 정상적으로 동작할 때의 자료와 비정상적으로 동작할 때의 자료로 구성된다.

정상적인 동작시 수행된 시스템 호출 자료에서 UNM-sendmail-daemon은 Forrest 연구팀에서 수집한 것이고, cert-sendmail-daemon은 CERT에서 수집한 것이다. sendmail 데몬 프로그램은 장기간 운영되면서 각 메일 관련 서비스를 요청받아 이를 처리하는데, 본 논문에서는 이러한 하나의 처리과정을 세션으로 규정하였다. UNM-sendmail-daemon의 경우 이러한 세션이 52개로 구성되어 있다. 비정상적인 동작시 수행된 시스템 호출 자료는 sendmail 데몬 프로그램이 공격을 받을 때 발생하는 것들로서, sunsendmailcp, decode intrusion, forwarding loops, SunOS 4.1.3 취약점 등 침입 공격이 발생할 때 수집한 것이다. 각 공격은 sendmail 데몬 프로그램에 대해 여러 번 수행되었는데, 이를 각각 trace로 구분하였다. 이 공격들은 전형적으로 sendmail 데몬 프로그램에게 메일 관련 서비스를 요청하여 침입공격을 수행하며, 보통 한번의 세션에 걸쳐 이루어진다.

4.2 실험 결과 및 분석

Forrest 연구팀에서 수집한 Sendmail 데몬 프로그램에서 사용하는 시스템 호출의 종류는 184개이다. 따라서 각 trace에 포함된 시스템 호출은 식 (4)에 따라 벡터 $X = (X_1, X_2, \cdots, X_{184})$로 표현된다. 이때 평활상수 λ는 0.3을 사용하였다.

먼저 정상적인 시스템 호출들로 구성된 UNM-sendmail-daemon으로부터 각 시스템 호출을 벡터 $X = (X_1, X_2, \cdots, X_{184})$로 표현하여 평균 벡터를 구하고, 이 평균 벡터를 이용하여 각 trace의 X^2 거리를 구하였다. <표 2>는 각 trace의 X^2 거리를 보여준다. 여기서 정상적인 행위의 trace는 "*"로 표시되어 있다.

〈표 2〉 각 trace별 X^2 거리

Trace	평균 ($\overline{X^2}$)	표준 편차 (S_{X^2})	최소값 (X^2 - min)	최대값 (X^2 - max)
UNM-sendmail-daemon*	9.47	4.14	0.19	234468.06
cert-sendmail-daemon*	6.54	3.38	0.19	36735.52
cert-sm565a-1	12188.70	135.05	0.52	204526.08
cert-sm5x-1	1.15E+96	1.51E+48	0.18	9.00E+98
fwd-loops-1	5.57E+96	3.31E+48	0.20	1.34E+99
fwd-loops-2	5074.74	92.72	0.19	234468.06
fwd-loops-3	6.34E+96	3.53E+48	0.20	1.34E+99
fwd-loops-4	5.56E+96	3.31E+48	0.20	1.34E+99
fwd-loops-5	14092.26	144.95	0.23	234468.10
sm-10763	6857.39	105.05	0.20	234468.06
sm-10801	6857.39	105.05	0.20	234468.06
sm-10814	6857.39	105.05	0.20	234468.06
sm-280	2452.42	65.85	0.19	234468.06
sm-314	2406.66	65.17	0.19	234468.06
syslog-local-1	2.16E+97	6.46E+48	0.19	4.32E+99
syslog-local-2	2.19E+97	6.50E+48	0.19	4.32E+99
syslog-remote-1	1.95E+97	6.13E+48	0.18	4.32E+99
syslog-remote-2	1.77E+97	5.89E+48	0.18	4.32E+99

X^2가 작을수록 프로파일에 가깝고 클수록 프로파일과는 동떨어진 행위이다. <표 2>에서 정상적인 행위의 trace, 즉 UNM-sendmail-daemon과 cert-sendmail-daemon에 대해서 X2의 평균은 매우 작고, 비정상적인 행위의 trace들에 대해서는 X^2의 평균은 매우 크다. 즉, 정상 행위의 trace에 포함된 시스템 호출들은 대체적으로 프로파일의 일정 범위 내에서 나타나고, 비정상 행위의 trace에 포함된 시스템 호출들은 대체적으로 프로파일의 범위를 멀리 벗어난다고 볼 수 있다.

UNM-sendmail-daemon에서, 평균($\overline{X}^2$)과 표준편차(S_{X^2})는 각각 9.47, 4.14이다. 이 값들을 이용하여 식 (7)에 의해 이상행위 임계치를 21.88로 설정하였다. 이 임계치를 이용하여 각 trace의 SSR을 구하면 <표 3>과 같다. 정상행위의 trace로서 UNM-sendmail-daemon은 프로파일 구축에 이용되었기 때문에 제외하고, 또 다른 정상행위의 trace인 cert-sendmail-daemon에 대해 SSR을 구하였고, 이때 SSR 값은 그 trace에 포함된 52개의 세션들의 SSR을 평균한 값이다.

〈표 3〉 각 trace별 SSR

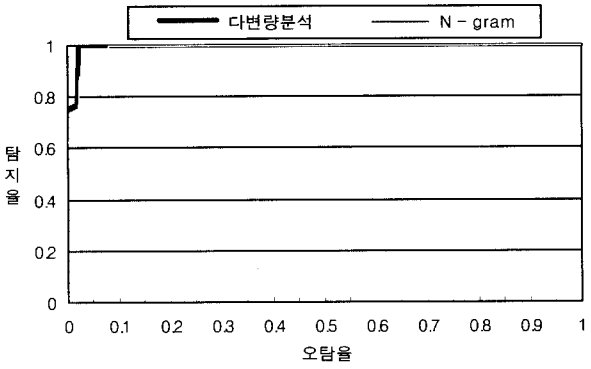
Trace	N-gram	다변량분석	증감율
cert-sendmail-daemon	0.01	0.01	+0.0
cert-sm565a-1	0.93	0.89	-0.04
cert-sm5x-1	0.99	0.76	-0.23
fwd-loops-1	0.80	0.86	+0.05
fwd-loops-2	0.56	0.81	+0.25
fwd-loops-3	0.77	0.84	+0.07
fwd-loops-4	0.80	0.86	+0.06
fwd-loops-5	0.95	0.92	-0.04
sm-10763	0.96	0.90	-0.06
sm-10801	0.96	0.90	-0.06
sm-10814	0.96	0.90	-0.06
sm-280	0.09	0.48	+0.38
sm-314	0.17	0.48	+0.31
syslog-local-1	0.97	0.86	-0.11
syslog-local-2	0.97	0.86	-0.12
syslog-remote-1	0.98	0.83	-0.15
syslog-remote-2	0.97	0.79	-0.18

<표 3>에서 N-gram 기법의 경우, 시퀀스(sequence) 길이는 80이다. sendmail 데몬 프로그램의 시스템 호출 자료에서 시퀀스 길이는 6이상인 경우, 탐지율이 유의할만하고, 시퀀스 길이가 커질수록 탐지율은 높아진다[7-9]. 여기에서는 또다른 정상적인 행위의 trace인 cert-sendmail-daemon에 대해 얻어지는 두 기법의 SSR 값을 동일하게 하기 위해 시퀀스 길이를 80으로 정하였다.

<표 3>에서 보는 바와 같이 대체적으로 다변량 분석기법의 경우, 비정상적인 trace의 SSR 값들이 정상적인 trace의 SSR 값보다 훨씬 크다. 특히, sm-280, sm-314의 경우, N-gram 기법에서는 SSR의 값이 정상적인 trace의 SSR과 그 차이가 매우 작지만, 다변량 분석 기법에서는 SSR 값들이 크게 증가하여 그 격차가 매우 커진다.

반면에 cert-sm5x, syslog-remote-1, syslog-remote-2의 경우, 상대적으로 다변량 분석 기법에서 생성되는 SSR 값들이 N-gram 기법을 적용한 경우보다 낮게 나타난다. 그러나 정상적인 trace와 비교하여 그 격차는 충분히 크다.

<표 3>의 각 trace별 SSR을 이용하여 최종 이상행위 임계값(anomaly score threshold) α를 변화시키면서 탐지율과 오탐율의 상관 관계를 살펴보기 위해 ROC 곡선을 구하면 (그림 1)과 같다.



(그림 1) ROC 곡선

ROC 곡선은 왼쪽 상단에서 1에 가까울수록 우수한 성능을 보인다. (그림 1)에서 다변량 분석 기법의 ROC 곡선로부터 다변량 분석 기법은 오탐율(false alarm rate)이 0인 경우 75%의 탐지율을 보이고, 오탐율이 0.02에서부터 탐지율이 100%이다. 반면에 N-gram 기법의 경우, 오탐율이 0인 경우 62%의 탐지율을 보이고, 오탐율이 0.08에 이르렀을 때 비로소 100%의 탐지율을 보인다. 결과적으로 다변량 분석 기법의 경우, N-gram 기법에 비해 더 적은 오탐율 상에서 온전한 탐지율(100%)에 수렴함을 확인할 수 있다. 즉, 다변량 분석 기법은 최종 이상행위 임계값에 대해 N-gram 기법보다 덜 취약하다고 할 수 있다.

한편 두 기법의 프로파일의 크기를 비교해보면, 위의 실험 결과에서 N-gram 기법의 경우, 프로파일을 구성하는 스트링들의 개수는 3916이다. 이에 덧붙여 스트링의 크기를 고려하여야 하기 때문에, 결과적으로 프로파일의 크기는 3916 ×80이다. 반면에 다변량 분석 기법을 응용한 경우, 프로파일은 프로그램 행위에 대한 평균 벡터이기 때문에, 그 크기는 184에 불과하다. 또한 프로그램의 정상 행위가 추가될 때마다 N-gram 기법의 프로파일은 계속 증가하지만, 다변량 분석 기법의 경우 비교적 적은 고정 크기의 프로파일을 계속 유지한다.

이러한 프로파일의 크기는 시간 복잡도에 영향을 미친다. 즉, N-gram 기법의 경우 어떤 스트링이 의심스러운지 판정하기 위해서는 시간 복잡도는 $O(N \times k)$이다. N은 프로파일 데이터베이스에 포함되는 스트링들의 개수이고, k는 스트링의 길이다. 반면, 다변량 분석 기법의 경우 시간복잡도는 $O(n)$이다. n은 벡터의 크기이다. 즉, 두 기법 모두 시간복잡도는 프로파일의 크기에 비례하는데, 다변량 분석 기법을 이용하여 프로파일의 크기를 작게 유지하면서 결과적으로 시간복잡도를 줄일 수 있다. 이는 실시간 침입탐지 측면에서 다변량 분석 기법이 더 유리하다고 할 수 있다.

5. 결론 및 향후 연구 방향

프로그램 행위 기반 침입탐지 기법은 컴퓨터 시스템을 대상으로 이루어지는 잠재적인 공격을 탐지하는데 효과적인 방법이다. 루트권한으로 수행되면서 네트워크 서비스를 제공하는 프로그램들에 대해 빈번히 공격이 행해지고, 다양한 보안 공격 방지책에도 불구하고 프로그램 상의 코드 오류나 잘못된 설정으로 인해 루트 권한이 권한이 도용되는 등 여전히 침입공격이 이루어지고 있기 때문이다.

N-gram 기법을 비롯하여 대표적인 프로그램 행위 기반 침입탐지 기법은 높은 탐지율을 보이는 반면 생성되는 프로파일의 크기는 매우 크다. 일반적으로 시간복잡도는 프로파일의 크기에 비례하기 때문에, 실시간 탐지에 어려움이 있다.

본 논문에서는 프로그램 행위 프로파일링에 X^2 거리기반 다변량 분석 기법을 적용하여 프로파일의 크기를 작게 유지하도록 시도하였다. 즉, 여러 시스템 호출 정보 중에서 시스템 호출 종류만을 이용하여, 프로그램이 정상적으로 운용될 때 발생되는 시스템 호출들을 간결한 형태의 행위 벡터로 표현하고, X^2 거리기반 다변량 분석 기법을 이용하여 세션별로 이상 행위를 탐지하는 척도를 제시하였다. 성능 평가를 위해, UNIX 환경에서 정상적으로 운용된 sendmail 데몬 프로그램의 시스템 호출들과 sendmail 데몬 프로그램에 각종 공격이 이루어질 때 발생하는 시스템 호출 정보들을 대상으로 실험한 결과, N-gram 기법보다 프로파일의 크기는 훨씬 줄이면서 탐지율도 다소 높게 나타났다.

향후 연구과제로서 UNIX 환경에서 빈번히 공격의 대상이 되고 있는 다른 프로그램들에 대해서도 실험을 확대할 것이고, 프로그램이 전형적으로 수행하는 다양한 행위들을 고려하여 각각의 전형적인 행위들에 대한 시스템 호출들간의 특징을 도출하여 탐지율과 탐지의 적시성(timely alarm)을 향상시키는 방안에 대해 연구하는 것이다.

참 고 문 헌

[1] S. Axelsson, "Intrusion detection systems : A survey and taxonomy," Technical report. Department of Computer Engineering, chalmers University of Technology, Goteborg, Sweden, 2000.

[2] S. Noel, D. Wijesekera and C. Youman, "Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt," Applications of Data Mining in Computer Security, Kluwer Academic Publishers, 2002.

[3] S. Kumar and E. H. Spafford, "A Software Architecture to Support Misuse Intrusion Detection," Proceedings of the 18th National Information Security Conference, pp.194-204, 1995.

[4] A. K. Ghosh, A. Schwarzbard and M. Shatz, "Learning program behavior profiles for intrusion detection," Proceedings of the 1st UNENIX Workshop on Intrusion Detection and Network Monitoring, April, 1999.

[5] C. Krugel, T. Toth and E. Kirda, "Service Specific Anomaly Detection for Network Intrusion Detection," Symposium on Applied Computing (SAC), ACM Digital Library, March 2002.

[6] A. K. Ghosh, J. Wanken and F. charron, "Detecting anomalous and unknown intrusions against programs," Proceedings of the 1998 Annual computer Security Applications conference(ACSAC '98), 1998.

[7] S. Forrest, S. Hofmeyr, A. Somayaji and T. Longstaff, "A sense of self for unix processes, In IEEE Symposium on Security and privacy," pp.120-128, 1996.

[8] S. A. Hofmeyr, A. Somayaji and S. Forrest, "Intrusion Detection using Sequences of System Calls," Journal of Computer Security, Vol.6, pp.151-180, 1998.

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

[10] C. Ko, G. Fink and K. Levitt, "Execution monitoring of security-critical programs in distributed systems : A specificatin-based approach," Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp.134-144, 1997.

[11] C. Ko, G. Fink, K. Levitt, "Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring," Proceedings of the 1994 Computer Security Applications Conference, 1994.

[12] D. Wagner and R. Dean, "Intrusion detection via static analysis," In IEEE Symposium on Security and Privacy, IEEE Computer Society, 2002.

[13] 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.

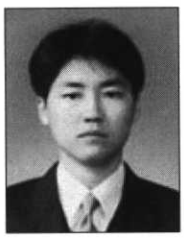
[14] W. Lee and S. Stolfo, "Learning Patterns from Unix Process Execution Traces for Intrusion Detection," AAAI Workshop : AI Approaches to Fraud Detection and RISK Management, pp.50-56, July, 1997.

[15] N. Ye, Q. Chen, S. Vilbert, "Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection," IEEE Transactions of computers, Vol.51, No.7, pp.810-820, July, 2002.

[16] D. Montgomery, "Introduction to Statistical Quality Control," John wiley & Sons, 2000.

[17] C. A. Lowry, W. H. Woodall, C. W. Champ and S. E. Rigdon, "A Multivariate Exponentially Weighted Moving Average Chart," Technometrics, 34, pp.46-53, 1992.

[18] S. Forrest, Computer immune systems data sets, http://www.cs.unm.edu/~immsec/data-sets.htm, 1997.



김 정 일

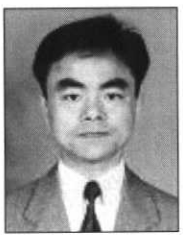
e-mail : kci@chonnam.ac.kr

1997년 전남대학교 전산학과(학사)

1999년 전남대학교 대학원 전산학과(이학석사)

1999년~현재 전남대학교 대학원 전산학과 박사과정

관심분야 : 서열 학습(sequence learning), 시스템 및 네트워크 보안, 인공지능 등



김 용 민

e-mail : ymkim@chonnam.ac.kr

1989년 전남대학교 전산통계학과(학사)

1991년 전남대학교 전산통계학과(이학석사)

2002년 전남대학교 전산통계학과(이학박사)

2003년~현재 전남대학교 리눅스시스템 보안연구센터 Post-doc.

관심분야 : 시스템 및 네트워크 보안, 정보보호, 네트워크 관리, 퍼지 이론 등

서 재 현

e-mail : jhseo@mokpo.ac.kr

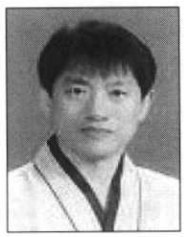
1985년 전남대학교 계산통계학과(학사)

1988년 중앙대학교 전자계산학과(이학석사)

1996년 전남대학교 전산통계학과(이학박사)

1996년~현재 목포대학교 정보공학부 정보보호전공 부교수

관심분야 : 네트워크 프로토콜 및 서비스, 네트워크 보안, 통신망관리, 정보보호 등



노 봉 남

e-mail : bongnam@chonnam.ac.kr

1978년 전남대학교 수학교육과(학사)

1982년 KAIST 대학원 전산학과(공학석사)

1994년 전북대학교 전산통계학과(이학박사)

1983년~현재 전남대학교 컴퓨터정보학부 교수

관심분야 : 통신망관리, 정보보호, 시스템 및 네트워크 보안 등