情報保護學會論文誌
第13券 第5號, 2003. 10

SVM 기반의 효율적인 신분위장기법 탐지*

김 한 성**, 권 영 희**, 차 성 덕**

Efficient Masquerade Detection Based on SVM

Han-Sung Kim**, Younghee Kwon**, Sung-Deok Cha**

요 약

다른 사용자의 계정, 디렉터리, 또는 파일을 침해하는 동안 그 사용자인 척 하거나, 컴퓨터 시스템에서 다른 사용자인 것처럼 자신의 신원을 숨기는 사람을 "신분위장자(masquerader)"라고 한다. "신분위장 공격(masquerade attack)"은 가장 심각한 형태의 컴퓨터 오용(misuse)이다. 왜냐하면, 대부분의 경우 여러 가지 방법으로 타인의 패스워드 등을 확보하여 인증절차를 통과한 후 이므로 이후에 발생하는 컴퓨터의 오용내지 남용에 대하여 시스템은 정상적으로 탐지, 대응할 수 없는 경우가 대부분이기 때문이다. 신분위장자를 자동적으로 발견하기 위한 노력이 "신분위장기법 탐지(masquerade detection)"인데 통상 시스템 감사 데이터로부터 만든 정상 사용자 행동의 프로파일에 대한 심각한 위반을 찾아내는 방법을 사용하여 탐지할 수 있다. 1988년 이후 본격적으로 이에 대한 연구들이 있었지만 현재까지 이러한 접근법에 대한 성공은 제한적이었고, 성능 또한 만족스럽지 않았다. 이 연구에서는 SVM(Support Vector Machine)을 이용, 사용자 프로파일을 만들고, 이를 이용한 효과적인 신분위장기법 탐지 시스템을 제안한다.

ABSTRACT

A masquerader is someone who pretends to be another user while invading the target user's accounts, directories, or files. The masquerade attack is the most serious computer misuse. Because, in most cases, after securing the other's password, the masquerader enters the computer system. The system such as IDS could not detect or response to the masquerader. The masquerade detection is the effort to find the masquerader automatically. This system will detect the activities of a masquerader by determining that user's activities violate a profile developed for that user with his audit data. From 1988, there are many efforts on this topic, but the success of the efforts was limited and the performance was unsatisfactory. In this report we propose efficient masquerade detection system using SVM which create the user profile.

keyword : *intrusion detection, anomaly detection, masquerade detection, SVM(Support Vector Machine), user command*

Ⅰ. 서 론

다른 사용자의 계정, 디렉터리 또는 파일을 침해하는 동안 원래의 그 사용자인 척 하거나, 컴퓨터 시스템에서 다른 사람인 척 자신의 신원을 숨기는 사람을 "신분위장자(masquerader)"라고 한다. 이러한 신분위장자들은 내부자와 외부 침입자로 나눌 수 있다. 대부분의 외부 침입자들은 시스템에 접근 후 즉각적으로 슈퍼 유저(super-user)의 계정에 접근하여 슈퍼 유저의 권한을 획득하려는 특성을 갖기 때문에 실제로, 신분위장자의 대부분은 내부자인 경우가 많다.

Denning은 [표 1]과 같이 공격의 형태를 8가지 유

* 본 연구는 침단정보기술연구센터(AITRC)와 소프트웨어 프로세스 개선센터(SPIC)의 지원을 받아 수행하였습니다.
** 한국과학기술원 전자전산학과 전산학 전공(kimhs@salmosa.kaist.ac.kr, kyhee@ai.kaist.ac.kr, cha@salmosa.kaist.ac.kr)

〔표 1〕 공격의 형태[1]

Type	Methods
Eavesdropping and packet sniffing	Passive interception of network traffic
Snooping and downloading	Browsing private information from world wide web
Tampering or data diddling	Unauthorized changes to data or records
Spoofing	Impersonating other users, e. g. by forging the originating e-mail address, or by gaining password access
Jamming or flooding	Overwhelming a system's resources, e.g. by an e-mail flood or HTTP requests
Injecting malicious code(such as viruses and Trojan horses)	Via floppy disks or e-mail attachment
Exploiting design or implementation flaws	Often buffer overflows, which overwrite other data and can be used to get control over a system
Cracking passwords and keys	Brute force password attacks and dictionary attack

형으로 나누었다. 신분위장 공격(masquerade attack)은 Denning의 분류 중 위장하기(spoofing)에 해당된다. 신분위장자는 신분위장 공격을 통하여 데이터 변경 및 위조(tampering or data diddling), 방해나 넘침(jamming or flooding) 등 다양한 형태의 공격을 할 수 있다. 이러한 경우 계정 사용자의 권한을 침입자 즉, 신분위장자가 이용하기 때문에 탐지에 어려움이 있으며, 탐지를 한다고 하더라도 실제 사용자가 누구인지를 다시 알아내야 하므로, 침입자를 찾아내는 것은 사실상 어려운 문제가 된다.

CSI/FBI의 2002년 보고서[2]에 의하면 내부자에 의한 인터넷 접속 남용(78%), 내부자에 의한 시스템 불법 접근(38%)이 컴퓨터 오용 및 공격의 대표적인 사례로 보고되고 있으며, 이로 인한 피해액도 각각 5천만 불과 4백5십만 불로 주요한 피해원인(각각 3위와 9위)으로 등장함을 알 수 있다. 이와 같이 "내부자에 의한 불법행위"는 보안에 있어서 심각한 취약점 중 하나 인 것을 알 수 있다. 즉 특정 사용자인 것으로 자신의 신원을 숨기고 컴퓨터를 악용하는 "신분위장 공격"은 심각한 형태의 컴퓨터 오용(misuse)인 것이다.

내부자에 의한 불법적인 컴퓨터 범죄의 대표적 사례로 2001년 미국 전 FBI 요원 Robert P. Hanssen의 사건[3]이 꼽힌다. 그는 FBI 내부 시스템인 Automated Case Support System (ACS)에 접속하여 인가되지 않은 비밀 자료를 확보한 후 소련에 오랫동안 거래활동을 한 혐의로 체포되어 실형을 받았다. 이 사건은 내부자에 의한 정보유출의 심각성을 드러낸 사례로 평가되고 있다.

신분위장기법 탐지(masquerade detection)에 관한 연구의 역사는 1988년으로 거슬러 올라가게 되는데 전형적인 접근 방법의 아이디어는 "신분위장자의 행동이 정상 사용자의 프로파일(profile)에서 심각하게 벗어난다."는 데에서 출발한다. 결국 신분위장기법 탐지는 이상 탐지 시스템의 원리와 부합되며 이를 잘 적용하면 효율적인 신분위장기법 탐지 시스템의 제작이 가능하다.

이 연구에서는 사용자 프로파일을 만들어내기 위한 방법으로 SVM(Support Vector Machine)의 사용을 제안한다. SVM의 효용성을 검증하기 위하여 UNIX 환경에서의 사용자 명령어를 신분위장기법 탐지를 위한 감사 데이터(audit data)로 사용하였다.

이 논문은 다음과 같이 구성되었다. 2절에서는 기존의 신분위장기법 탐지 관련 연구와, SVM의 침입탐지 관련 연구를 살펴보고, 3절에서는 SVM이 무엇이고 신분위장기법 탐지에 어떻게 사용될 수 있는지에 관하여 알아본다. 4절에서는 SVM을 이용한 효과적인 신분위장기법 탐지 시스템의 프레임웍에 대하여, 제5절에서는 실험 데이터에 대하여 고찰하고, 실험결과 그리고 마지막으로 결론 및 향후 연구방향을 정리한다.

II. 관련연구

2.1 침입 탐지

침입탐지 시스템은 패턴인식을 이용한 오용 탐지(misuse detection)와 이상 탐지(anomaly detection)의 두 가지 유형으로 나누어진다.[4] 대부분의 상용제품들은 시그너처(signature) 기반의 오용 탐지 시스템으로 탐지속도와 탐지율 측면에서는 빠르고 효율적이라는 장점을 가지지만 각각의 공격에 대한 시그너처를 가지지 않으면 공격을 탐지 할 수 없고, 동일한 형태의 공격을 하더라도 시그너처를 우회할 수 있는 방법이 있다면 또한 탐지를 할 수 없다는 단점을 갖는다.[5,6] 신분위장기법 탐지의 관점에서 볼 때 오용탐지 시스템은 신분위장자가 시스템에서 시그너처가 알려진 공격을 위한 코드를 수행하지 않는 한 신분위장자의 행동을 탐지 할 수 없다는 한계를 갖는다.

반면 이상탐지 시스템은 정상적인 행위를 모델링 함으로서 정상행위의 프로파일에서 벗어나는 비정상행위를 탐지하는 것으로 프로파일을 만들기 어렵고[1] 허위 경보(false alarm)가 높다는 단점을 가지지만 정상행위의 프로파일로 인하여 여기서 벗어나는 행위는 비록 알려지지 않은 공격이라고 할지라도 탐지할 수 있다는 장점을 갖는다. 신분위장자는 이미 여러 가지 방법으로 기존의 다른 보안 기제(방화벽이나 기타의 접근통제 방법)를 회피하였기 때문 이상탐지 시스템은 신분위장기법을 탐지하기 위한 대안으로 가장 효과적이라고 할 수 있다.

이상 탐지를 위하여 사용자 프로파일(user profile)을 만들기 위한 여러 가지 방법들로 통계학적 방법,[7] 데이터베이스 등을 이용한 데이터 마이닝 기법, 또한 인공지능 분야의 여러 기술을 응용하는 기계학습 기법들이 다양하게 사용될 수 있고,[8] 이 중 몇 가지 방법들은 효과적으로 사용될 수 있음을 검증 받기도 하였다.

2.2 신분위장기법 탐지 연구

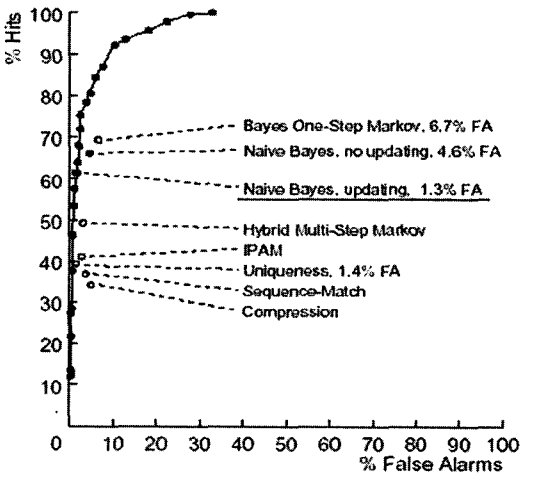
1988년 발표된 IDES 시스템[9]은 신분위장기법 탐지를 가장 먼저 다룬 시스템이다. 이러한 연구들이 활발히 되지 못한 몇 가지 이유는 다음과 같다. 첫째, 실제 침입에 대한 데이터를 얻기가 매우 어렵다는 것이다. 또 실제 침입에 대한 데이터를 확보하게 되더라도 어떤 침입들은 탐지되지 않는다는 것이다. 이것은 침입이 일어나도 탐지 할 수 없는 프로파일을 만들게 되는 원인이 된다. 둘째로는 기존의 연구결과 그 성능이 크게 만족스럽지 못했다는 것이다.[10]

Schonlau[7]는 그의 연구에서 UNIX acct를 기반으로 수집된 사용자 명령어를 감사 데이터[2]로 사용하여 신분위장기법 탐지를 수행하였다. 신분위장기법 탐지를 위하여 [표 2]에서 보는 것처럼 총 6가지 방법을 사용하였다. [그림 1]에서와 같이 6가지 실험방법에 의한 신분위장기법 탐지결과는 39.4%~69.3%의 탐지율과 1.4%~6.7%의 허위 경보 비율을 보였다.

Maxion[10]은 Schonlau의 실험의 결과를 바탕으로 1) 탐지율을 향상 시키고, 2) 각 사용자 간의 데이터와 탐지율간의 상관관계를 밝히기 위하여 Naive Bayes

〔표 2〕 Schonlau의 6가지 분류기법

분류 기법	내 용
Uniqueness	훈련간에 보여 지지 않은 명령어, 적은 사용자가 사용하는 명령어가 신분위장자의 명령어일 가능성이 높다는 생각에서 출발. 신분위장자를 찾는 능력은 가장 저조. 허위 경보를 최소화시키기는 좋은 방법.
Bayes One-step Markov	Uniqueness가 단순히 명령어의 빈도만을 이용한 것이라면 이 방법은 하나의 명령과 그 다음으로의 진행에 기초한 것으로 탐지기는 관측된 변환 확률(transition probability)이 기존의 확률과 비교 일정한지를 확인. 가장 정확한 탐지. 허위경보비율을 줄이는 데는 부적합.
Hybrid Multistep Markov	훈련간에 나타난 데이터일 경우 multistep 또는 high order markov chain에 기초하고, 그렇지 않을 경우에는 independence model 적용.
Compression	원래 사용자의 훈련 데이터에 연결된 데이터가 신분위장자의 데이터를 연결했을 때보다 압축속도가 빠르다는 것을 이용.
IPAM (Incremental Probabilistic action model)	Bayes one-step markov와 유사.
Sequence-match	최근 10개 명령의 유사도를 계산하여 비교.[8]



〔그림 1〕 Schonlau와 Maxion의 신분위장기법 탐지 결과

1) 다음과 같은 두 가지 이유에 의하여 프로파일을 만드는 것이 쉬운 일이 아니다. 첫째, 정상 프로파일을 만들기 위하여 정상적인 행위와 침입의 구분이 명확한 데이터를 확보해야 되는데 이러한 자료를 구하는 것이 어렵고, 둘째 오용탐지시스템에 사용되는 시그너처를 만드는 것보다 프로파일을 만드는 것은 상대적으로 많은 계산 비용을 요구한다. 왜냐하면, 자료를 계속 추적 저장하여야 하고 사용자의 시스템 사용과 행위자체가 계속 변하기 때문에 프로파일을 지속적으로 갱신하여야 하기 때문이다.

2) 실험 데이터에 대한 구체적인 자료와 구성은 4장에서 설명한다.

classifier라는 알고리즘을 도입하였다. 또한 기존의 실험 데이터가 서로 다른 조건으로 임의 구성되어 적절한 분석을 이끌어 내기가 곤란한 점을 해결하기 위하여 데이터의 구성을 두 가지로 구분하였다. Schonlau의 원래 데이터 구성을 "SEA data configuration"으로 정의하였고 새로운 형태의 데이터 구성인 "1vs49 data configuration"을 정의하여 실험에 이용하였다. Maxion은 [그림 1]에서 보는 바와 같이 Schonlau의 연구결과보다 탐지율을 56% 향상시켰지만 아직도 상대적으로 낮은 탐지율을 보여준다.

Maxion은 실험에 사용된 데이터가 탐지율을 높이기 위하여 필요한 추가적인 정보(예를 들면 명령어의 인자(argument), 사용자 세션의 길이, 형태, 사용자의 유형 등)가 부족함을 설명하였고, 실험 데이터의 구성과 분류기의 선택에 따라 탐지율이 크게 좌우됨을 역설하였다.

2.3 SVM을 이용한 침입탐지 시스템

SVM을 침입탐지에 활용한 연구로는 New Mexico Institute of Mining and Technology(NMT)의 연구[11,12,13]와 Los Alamos연구소의 연구[14]가 있다.

[표 3]에서 보는 것과 같이 NMT의 연구는 "DARPA KDD 99 Data3)"를 이용하여 3가지 서로 다른 구성의

〔표 3〕 SVM을 이용한 침입탐지 연구 결과

	감사 데이터	탐지 방법	특징수	탐지율 (%)	비고
NMT[11]	DARPA KDD 99 Data	Neural Net(3,4-layer feed-forward NN)	41	99.25	
		SVM(RBF)	41	99.50	SVM^{Light}
NMT[12]	Command and HTTP	SVM	8	94.00	
Los Alamos Lab.[14]	DARPA KDD 99 Data	Mahalanobis Outlier Detection	41	90.30	
		SVM(RBF)	41		SVM^{Light}, libSVM

3) KDD(Knowledge Discovery and Datamining) 콘테스트에 사용되었던 자료로 1998년 DARPA 침입탐지 평가 프로그램으로부터 만들어진 데이터이다. 총 22개의 공격이 4개의 유형으로 구분되어 label되어 있으며 41개의 정량적이고 정성적인 특징을 갖는 네트워크 데이터로 구성되어 있다. 데이터는 http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html 에서 구할 수 있다.

신경망을 이용한 탐지결과와 SVM을 이용한 탐지성능을 비교하였다. 이 연구결과는 SVM을 침입 탐지에 이용할 경우 신경망을 이용했을 때와 비교될 정도로 매우 높은 탐지율을 보여주는 것과 더불어 신경망보다 훈련시간과 탐지시간이 월등히 빠르다는 결과를 보여준다.

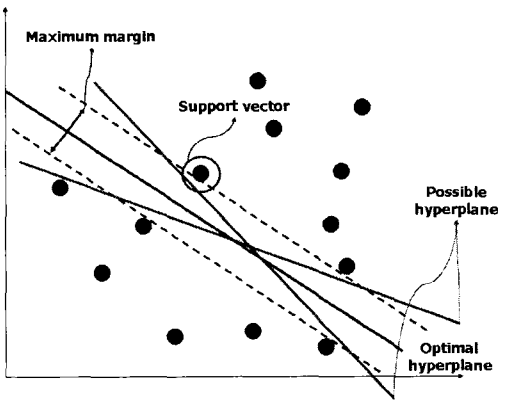
DARPA의 데이터는 총 41개의 특징(feature)으로 구성되어 있다. 이는 각각 네트워크에서의 패킷에 부여되는 각종 정보를 의미한다. 특징의 수가 많아지면 특징의 차원(dimension)이 증가함에 따라 데이터의 처리, 훈련, 그리고 식별에 이르기까지 여러 가지 부담으로 작용할 수도 있다. 그러므로 중요한 특징을 고르는 것이 중요하고 이에 따라 [13]의 연구에서는 41개 특징의 순위를 정하고 고르는 실험과 이를 위한 알고리즘, 그리고 5가지 공격의 유형에 따라 선택적으로 식별할 수 있는 SVM을 이용한 IDS의 구조를 제안하였다.

[14]의 연구는 "DARPA KDD 99 Data"를 이용하여 SVM의 RBF kernel과 Mahalanobis Distance의 두 가지 방법을 비교하였다,

Ⅲ. Support Vector Machine(SVM)

3.1 개요[15]

Support Vector Machine(SVM)은 1995년 Vapnik에 의하여 개발되고 제안된 학습 알고리즘이다. 이것은 원래 이진분류(binary classification) 를 위하여 개발되었으며 현재에는 생물정보학(bioinformatics), 문자인식, 필기인식, 얼굴 및 물체인식 등 다양한 분야에서 성공적으로 적용되고 있다.



〔그림 2〕 SVM을 이용한 이진 분류

이진분류 문제는 수집된 훈련 데이터를 이용해서 두 클래스를 분류하는 대상함수(target function)를 추정해 내는 과정이라고 볼 수 있다. 그렇게 추정된 분류기는 훈련과정에서 이용되지 않은 새로운 데이터 표본에 대해서도 올바른 결과 값을 낼 수 있는 일반화 성능(generalization performance)이 뛰어나야 한다.

SVM은 [그림 2]에서 보는 것과 같이 특징 공간(feature space)에서 데이터를 나눌 수 있는 초평면(possible hyper plane) 중에서 특정한 초평면(optimal hyperplane)을 선택함으로써 과적합 문제(overfitting)를 방지한다. SVM은 초평면으로부터 가장 가까운 훈련 포인트까지의 최소거리를 최대화시키는 초평면 즉, 최대 여백 초평면(maximum margin hyperplane)을 찾게 된다. Support vector(SV)라고 불리는 두 클래스들 사이의 결정 경계(decision boundary)에 가까이 놓여있는 훈련 예만이 non-zero weight를 갖게 된다. SV를 포함하는 초평면 사이의 거리인 여백(margin) 값이 클수록 분류성능은 좋아진다. 이렇게 찾아낸 초평면을 기준으로 테스트를 시행하여 분류 결과를 얻게 된다. 즉 그림과 같이 이진분류의 경우 SVM은 다음과 같은 방정식으로 설명이 되어진다.

$$F(x)=\begin{cases} -1 & Class\,A \\ +1 & Class\,B \end{cases} \quad (1)$$

[그림 2]의 예는 선형으로 분리 가능한(linearly separable)데이터 집합의 경우로 아주 쉽게 분류를 할 수 있지만 대부분의 분류 문제의 경우 비선형적인 분포를 취하고 있으므로 일반화에 심각한 문제를 겪게 된다. 이 문제의 해결을 위하여 슬랙 변수(slack variable)와 페널티 함수(penalty function)의 개념을 도입한 소프트 마진 분류기(soft margin classifier)를 통해서 어느 정도 비선형으로 분리되는 분류문제를 해결할 수 있다. C는 분리되지 않는 데이터(non-separable data)에 대한 페널티로 작용하는 변수로서 모델 복잡성과 트레이드오프 관계에 있다. 즉, C가 커지면 학습된 기계는 최적의 초평면(optimal hyperplane)을 구성하는 답을 제공하는 경향이 있으며, C가 0으로 수렴하는 값일 경우 여백을 극대화 시키려는 조건을 최적화 하려는 효과를 제공하게 되며, 그 결과 분류되지 않는 오류를 최소화하는 조건에는 그다지 큰 중점을 두지 않음으로 인해 여백의 폭이 아주 큰 SVM 분류기를 생성해 내게 된다.

일반적으로는 앞에서 이용한 선형경계(linear boundary)가 입력 벡터를 분류하기에 부적합한 경우가 대부분이다. 이 같은 경우 SVM은 입력 벡터 x를 보다 고차원 특징 공간(high dimensional feature space)내의 벡터로 변형한 후 선형경계를 찾는 문제로 변형하여 SVM을 구성하게 된다. 입력 공간에서 특징 공간으로의 변환은 일반적으로 비선형 사상을 이용하게 되며 이 같은 경우 Cover's Theorem에 의해서 몇 가지 조건이 만족할 때 입력 공간에서 비선형 분리 문제가 특징 공간에서는 선형분리 문제로 변환될 확률이 높음[16]이 알려져 있다. 이러한 고차원 특징 공간으로의 변환에 이용되는 비선형 사상 함수는 Mercer's theorem을 만족하는 함수들의 경우에 일반적으로 가능하다[16]고 알려져 있으며, 차원이 q인 Polynomials, Radial Basis Functions, Two-layer perceptron 등이 그 예이다. SVM^{light}에서는 이것을 지원하기 위하여 linear function, polynomial function, radial based function의 커널(kernel)이 지원된다.[17] 이러한 커널 중 데이터에 가장 적절한 커널과 그에 따른 매개변수(parameter)의 선택은 SVM의 이용한 분류의 성능에 아주 중요한 영향을 미치게 된다.

3.2 침입탐지를 위한 SVM사용의 장단점

SVM은 이진분류를 위하여 설계되었지만, 신분위장기법 탐지를 위한 도구로서 몇 가지의 장점을 제공한다.

첫 번째로 대부분의 분류문제에 있어서 SVM은 다른 기계학습방법과 비교하여 우월한 성능을 보여준다는 것이다.[18,19] 텍스트 분류에 있어서 SVM은 신경망과 Naive Bayes 모델보다 월등한 성능을 보여주었고,[20] 네트워크 침입탐지의 경우에도 신경망보다 우월한 성능을 보여주었다.[11] 그리고 SVM은 Mahalanobis distance-based 방법과 비교했을 때에도 우수한 성능을 보여주었다.[14]

두 번째로 SVM은 고차원 그리고 데이터가 적은 문제인 경우에 있어서도 능력을 잘 발휘한다는 것이다. SVM은 Structural risk minimization의 개념에 기초하여 일반화 성능이 우수하며 또한 비선형 분리 문제와 같은 복잡한 분류문제를 해결하기 위한 다양한 커널을 제공한다. 예를 들면 우리의 실험에 있어서 특징(feature)으로 총 852개의 서로 다른 명령어들을 사용하였다. 만약 신경망을 이용하였다면 이와 같이 큰 입력을 효과적으로 처리 할 수 없거나 처리한다고 하더라도 노드를 만들고 노드간의 가중치를 계산

하기 위하여 복잡한 계산과정을 거치게 되며 그에 따른 계산 비용은 커지게 된다.

세 번째로, 다양한 소프트웨어들(예를 들면 SVM^{light}, libSVM, SVM Torch 등)이 공개적으로 사용 가능하고, SVM이 다른 기계학습 방법들보다 상대적으로 사용하기 용이하다는 것이다. 예를 들면 신경망의 경우 사용자는 노드의 수, 가중치, 그리고 노드가 어떻게 연결될 것인지에 대하여 결정을 하여야 하며 값들을 계산하여야 한다. 그러나 SVM을 사용할 경우에는 커널의 선택과 그에 따른 비용(cost)값과 같은 커널 변수의 선택만 하면 된다. 또한 복잡한 코딩이나 지속적인 훈련패턴의 생성을 요하는 대부분의 기계학습방법과는 다르게 SVM은 새로운 패턴이 나타나면 훈련과정에 모델을 동적으로 갱신할 수 있다.

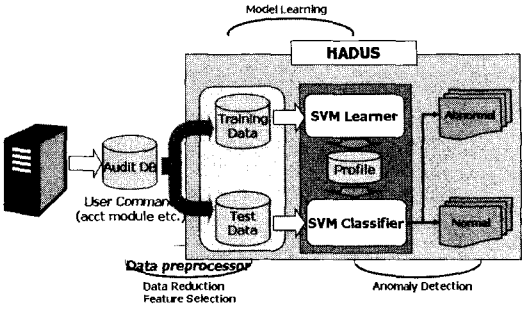
네 번째로, SVM은 상대적으로 데이터의 수에 민감하지 않다는 것이다. 그리고 분류의 복잡도는 특징공간의 차원에 의존적이지 않는다.[19]

그러나 SVM은 몇 가지 일반적인 단점을 갖고 있다. 그 첫째가 SVM은 모델을 만들고 분류하는데 있어서 일반적으로 시간이 많이 걸릴 수 있다는 것이다. 그러나 우리의 실험[4]에 있어서 총 852개의 입력차원을 갖는 24,000개의 데이터를 이용해서 훈련을 시키는데 평균 70.00 초가 소요되었다. 훈련시간은 모델에 따라 0.88초로부터 122.66초에 이르기까지 다양하였다. 모델이 생성된 이후에 신분위장자를 찾기 위한 시험 간에는 50개의 일련의 사용자 명령어를 분류하는데 0.5초미만의 시간이 소요되었다. 그러므로 실제로 사용자가 명령어를 입력하는 환경을 고려한다면 이와 같은 시간들은 SVM에 기초한 신분위장기법 탐지가 실시간으로 이루어 질 수 있음을 보여준다.

두 번째 단점은 Markov모델이나 Bayesian 모델은 명령어의 변이(transition)를 반영하기 쉽지만 SVM은 다른 처리과정을 거치지 않는 한 연속되어 입력되는 명령어들의 변이를 바로 반영하기 어렵다는 것이다.

Ⅳ. 시스템(Host based Anomaly Detection Using SVM :HADUS)구조

이 절에서는 명령어를 감사 데이터로 SVM에 기반 한 이상탐지의 하나로서 신분위장기법 탐지를 하기 위한 시스템의 구조를 [그림 3]과 같이 제안한다.



〔그림 3〕 시스템 구조

이 시스템은 UNIX 환경에서 사용자 명령어를 감사 데이터로 사용하여 사용자의 프로파일을 만드는 모듈(Model Learning)과, 이 프로파일을 기초로 새로 입력되는 사용자의 명령어들이 그 사용자의 프로파일과 비교하여 정상 사용자 또는 신분위장자인지

구분하는 모듈(Anomaly Detection)로 구성된다. 물론 감사 데이터로 사용할 사용자 명령어를 추출하는 부분, 또 이 추출된 사용자 명령어를 사용하여 프로파일을 만들기 위하여 전처리 하는 부분도 시스템을 구성하는 일부분이다.

4.1 명령어 수집기(Audit Data Collector)

사용자의 명령어를 수집하는 부분은 [표 4]에서 보는 것과 같이 여러 가지 방법을 이용하여 구현될 수 있다.

사이트별로 또는 호스트의 기능과 제공하는 서비스에 따라 사용되는 명령어들은 다양한 모습과 특성을 보인다. 이 연구에서는 실험과 결과의 객관적인 비교를 위하여 UNIX에서 제공하는 acct기능을 이용하여 명령어를 수집하여 실험한 [7] 에서 사용된 데이터를 그대로 사용하였다. acct기능을 이용하여 수집된 명령어의 예는 표 5에서 보는 것과 같은 다양한 내용을 포함한다. 그러나 실험에서는 단순히 사용자

〔표 4〕 다양한 사용자 명령어 수집 방법

분류	방법	비고
로그	bash등의 history 파일 등	
OS	UNIX의 acct mechanism	
응용프로그램	ttywatcher, shell 프로그램	
보안프로그램	Solaris BSM module	

4) 실험을 위한 코드는 ANSI C로 구현하였으며, 사용자 프로파일을 만들고 분류의 기능을 하는 SVM으로 SVMlight를 이용하였다. 데이터베이스는 mysql-3.24-54q를 사용하였고, 실험은 PC Server(Intel 1.5 GHz Processor * 2, 2GB RAM) Red Hat LINUX 7.3 하에서 수행 되었다.

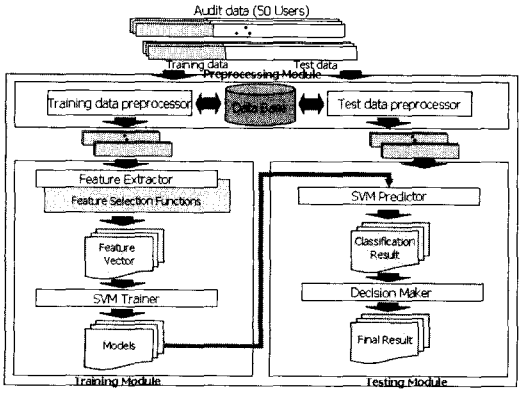
〔표 5〕 UNIX acct 감사기능에 의하여 생성된 감사 데이터의 항목 예[7]

Command Name	User	Terminal	Start Time	End Time	Real (sec)	CPU (sec)	Memory Usage
chmod	matt	pts/93	13:26:29	13:26:29	0.01	0.01	8.00
more	karr	pts/31	13:27:36	13:27:36	3.01	0.01	20.00
cat	vardi	pts/96	13:27:58	13:27:58	0.01	0.01	8.00
whoami	theus	pts/99	13:28:07	13:28:07	0.02	0.01	16.00
sendmail	karr	pts/91	13:28:17	13:28:17	0.02	0.01	124.00

가 입력한 명령어의 이름만 수집되었고 .이렇게 수집된 사용자 명령어들은 감사 데이터베이스(audit data base)라고 하는 데이터베이스에 먼저 저장되게 된다.

4.2 데이터 전처리기(Data Preprocessor)

저장된 데이터를 바탕으로 사용자별로 각각의 사용자 프로파일을 만들기 위하여 가장 효율적인 특징들을 골라내야 된다.[5] [표 5]에서 보는 것과 같이 감사 데이터에서는 아주 다양한 정보들을 제공하게 되는데, 훈련방법이나 도구의 특성에 따라 다양한 특징들을 찾아내기 위하여 감사 데이터를 재처리하는 과정을 거쳐야 한다. 대부분의 사용자 명령어에 대한 감사 데이터들은 사용자들의 입력한 명령어, 사용자, 위치, 시간, 시스템 사용정도 등의 다양한 정보를 제공하지만 단순히 이러한 여러 가지 정보의 나열로만 사용자 프로파일을 만들 수는 없다. 사용자 명령어를 이용한 사용자 프로파일을 만들기 위하여 사용될 수 있는 특징들로는 사용자별 사용 명령어의 순서, 사용한 명령어의 빈도, 사용한 명령어 간의 연간관계 등 다양한 특징들이 있을 수 있다. 이러한 특징들을 기초로 하여 프로파일을 만들기 위하여 감사 데이터 데이터베이스에 저장된 명령어에 관한 정보를 필요에 따라 가공하여야 하고, 이러한 기능을 데이터 전처리기에서 수행하게 된다.



〔그림 4〕 신분위장기법 탐지시스템 내부 구조

5) 이러한 과정을 특징 선택(feature selection)이라고 한다.

데이터 전처리기의 세부적인 구조와 기능은 [그림 4]와 같다. SVM에서 훈련을 시키기 위하여 사용되는 훈련 데이터 전처리기와 테스트 데이터를 만들어내기 위한 테스트 데이터 전처리기의 두 가지 기능을 분리하였다. 또 전처리기에서 처리된 다양한 정보들을 이용하여 SVM에서 사용될 형태로 필요한 특징들을 뽑아내는 특징 추출기(feature extractor)가 있다.

훈련 데이터 전처리기는 감사 데이터에서 SVM 모델링을 위하여 특징들을 산출해 내기 위하여 참조되는 여러 가지 정보들은 데이터베이스에 기록하게 된다. 앞에서 언급한 바와 같이 사용된 명령어를 기본 키(primary key)로 명령어의 사용빈도 등 각 명령어에 대한 다양한 정보를 기록하게 된다. 특징 추출기는 훈련 데이터 전처리기에서 처리된 다양한 정보들을 이용하여 SVM에서 사용될 형태로 필요한 특징들을 뽑아내게 된다.

4.3 SVM 훈련기(SVM Trainer)

전처리기에서 훈련과 테스트에 사용될 데이터에 대한 생산이 끝나고 나면 SVM 훈련기를 이용하여 모델을 만들게 된다. 모델은 데이터의 유형에 따라 다양한 커널과 또 다양한 인수를 적절히 골라서 만들어야 한다.

4.4 SVM 예측기(SVM Predictor)

이렇게 만들어진 모델과 테스트 데이터를 이용하여 분류를 수행하게 되고, 그 결과에 따라 정상(self) 또는 신분위장(침입/공격 또는 nonself) 등을 판단하게 된다.

4.5 최종 탐지기(Decision Maker)

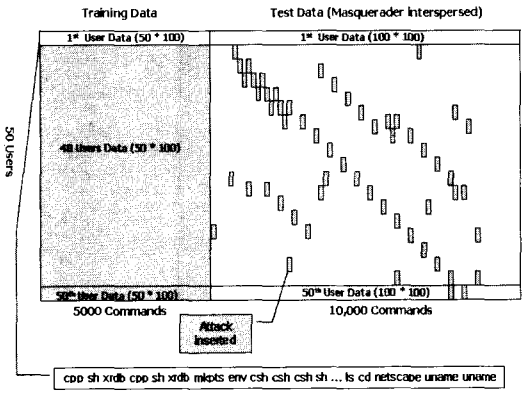
SVM 분류의 결과를 이용해서 마지막으로 최종결정을 하는 기능을 수행하게 된다. 단순히 명령어의 빈도에 의한 분류를 하는 과정을 거치게 되기 때문에 사용자에 따라서 유사한 명령어 사용의 빈도를

보이는 경우가 발생할 수도 있고 명령어의 특성에 따라서 많이 사용되는 명령어의 경우에는 사용자를 분별해내기 어려운 문제를 야기 시킨다. 물론 특수한 명령어를 특별한 사용자만이 사용한다면 쉬운 문제이지만 그렇지 않은 경우가 대부분이므로 SVM 예측기에 의한 분류의 신뢰도가 꼭 정확하다고 할 수 없다. 이러한 제한사항을 극복하기 위하여 최종 탐지기는 1) 비정상 행위가 얼마만큼의 빈도로 나타나고 있는지, 또는 2) 얼마만큼 연속적으로 신분위장 행위가 발생하는지를 기초로 하여 최종적으로 정상행위와 신분위장행위를 구분하는 기능을 한다. 즉 신분위장행위의 빈도 또는 연속적인 비정상행위의 빈도가 특정 임계치를 넘어간다면 최종적으로 신분위장행위로 결정되는 것이고, 넘지 않는다면 정상행위로 간주된다. 이 최종 탐지기를 통해서 한편으로 치우치거나 노이즈가 가미되어 있는 데이터를 직접 사용하여도 탐지의 정확도를 높일 수 있고, 감사 데이터가 원천적으로 갖고 있는 여러 가지 특성도 해결할 수 있었다. 임계치의 선정은 실험을 통하여 가장 정확도를 높일 수 있는 값으로 결정한다.

V. 실험 및 결과

5.1 실험데이터

실험데이터는 50명의 사용자의 명령어들로 구성되어있다. 각 사용자의 명령어들은 15,000개 씩 사용자가 입력한 순서에 따라 정리되어 있다. 각 명령어들은 100개 단위의 블록으로 구분되어 있다. [그림 5]에서 보는 것처럼 최초 5,000개의 명령어들(최초 50개 블록)은 순수하게 그 사용자의 명령어의 집합으로 이루어져 있으며 다음 10,000개의 명령어들(100개 블록)은 그 사용자의 명령어이거나 또는 신분위장자를 묘사하기 위한 50명 이외의 다른 사용자들의 명령어 블록으로 구성되어 있다. 즉 명령어들은 100개 단위(블록)로 구분하여 100개 단위 블록이 순수하게 자신의 명령어 블록이거나 또는 신분위장자를 묘사하기 위하여 100개 전체가 다른 특정 사용자의 명령어 블록으로 삽입되어 있다.



[그림 5] 실험 데이터

모델을 만들고 시험을 하기 위하여 Maxion은 두 가지 방법으로 데이터를 구성하였다. 이 연구에서는 두 가지 방법을 모두 적용하였다. 하나는 "SEA data configuration"이고 또 다른 하나는 "1 vs 49 data configuration"이다.

• SEA data configuration

각 사용자의 처음 5,000개의 명령어를 이용하여 모델을 만들고 각 사용자의 뒷부분의 10,000개의 데이터를 이용하여 시험을 실시한다.

• 1 vs 49 data configuration

각 사용자들은 자신의 처음 5,000개의 명령어를 이용하여 훈련을 실시하고 테스트 데이터로 자신을 제외한 49명의 다른 사용자들의 처음 5,000개의 데이터를 이용한다.

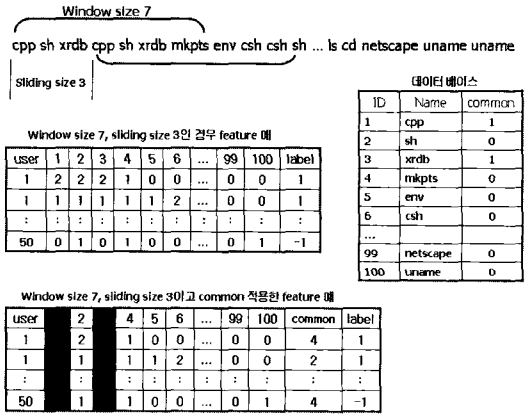
5.2 특징 선택(Feature Selection)

실험에서는 [표 6]에서 보는 것처럼 훈련(사용자 프로파일 생성)을 위한 감사 데이터의 특징으로 특정 길이의 연속되는 명령어(윈도우)에서 나타나는 명령어들의 사용빈도[6]를 사용하였다. 특징 추출의 방법을 [그림 6]을 이용하여 설명하면 다음과 같다. 즉 [그림 6]에서와 같이 사용자 1의 사용 명령어가 cpp sh xrdb cpp sh ……의 순으로 나오고 한번에 보는 명령어의 길이(윈도우 크기)가 7이라고 하자. 서로 다른 전체 명령어의 수는 전처리과정에서 얻어진 데이터베이스를 참고로 총 100개이므로 SVM에서 사용하게 되는 특징의 수는 모두 100개이다. 각각의 명령어에 대하여 윈도우 크기 내에서 등장하는 횟수가 기록되게 된다.

6) 기존의 Schonlau의 실험에서 보았던 6가지의 방법 중 uniqueness와 Maxion의 Naive Bayes classifier에서도 훈련 데이터에 포함되어 있는 명령어들의 빈도를 특징으로 사용하였다.

〔표 6〕 슬라이딩 윈도우내에서의 명령어 빈도

User	Windows	X_1	X_2	⋯	X_{n-1}	X_n	Label
1	1st	3	2	⋯	10	0	1
1	2nd	2	0	⋯	5	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	kth	2	2	⋯	3	0	-1
2	1st	6	0	⋯	9	5	-1
2	2nd	0	2	⋯	0	5	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	kth	3	2	⋯	10	5	-1



〔그림 6〕 특징 추출 과정

단순히 특정 윈도우 크기 내에서 등장하는 명령어의 빈도만 사용했을 경우 허위 경보의 비율이 높았다. 즉, 단순히 특정 윈도우에서의 빈도만을 고려했을 경우 이 특징 자체가 모델을 만드는데 적절하지 않은 특징이라는 것을 반증하는 것이다. 좀 더 일반적인 모델을 잘 만들 수 있는 특징을 선택하기 위하여 한 가지 특징을 추가하게 되었는데 바로 "공통 명령어"(common command)라는 개념이다. 일반적으로 명령어들은 사용자의 수, 사용자별 사용빈도에 의하여 많은 사용자들에 많이 사용되는 명령어 소수의 사용자만 많이 사용자는 명령어, 다수의 사용자가 적게 쓰는 명령어, 소수의 사용자가 적게 쓰는 명령어 등 4가지 유형으로 나누어 볼 수 있다. 이중에 많은 사용자들이 많이 사용하는 명령어들을 하나로 묶어서 "공통"(common)이라는 개념을 부여하는 것이다. 많은 사용자들이 어느 정도 이상의 비율로 사용하는 명령어들이 모델을 만드는데 각각 특징으로 선택하여 사용하게 되면 이것은 모델을 일반화 시키는데 오히려 역작용을 일으킬 수 있기 때문이다. 특정비율

〔표 7〕 공통 명령어를 적용했을 경우 슬라이딩 윈도우내에서의 명령어 빈도

User	Windows	X_1	X_2	⋯	X_{n-c1}	X_{n-c}	C	Label
1	1st	3	2	⋯	7	0	3	1
1	2nd	2	0	⋯	5	0	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	kth	2	2	⋯	3	0	0	-1
2	1st	6	0	⋯	9	5	3	-1
2	2nd	0	2	⋯	0	5	4	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
50	kth	3	2	⋯	10	5	3	-1

이상의 사용자들이 특정비율 이상으로 사용하는 명령어의 개수를 c개라고 하면 여기에 해당되는 명령어의 횟수는 모두 "공통"으로 합하여 지고 전체 특징의 개수는 c-1개만큼 줄어들게 된다. 물론 -1의 의미는 "공통"이라는 새로운 특징의 등장 때문이다. [표 7]은 이것으로 정형화 한 것이다. 이러한 특징을 이용한 경우의 특징 선택의 예는 [그림 6]에서 볼 수 있다.

특정 윈도우 상에서의 서로 다른 명령어의 사용빈도는 현재 가지고 있는 감사 데이터에서 추출할 수 있는 가장 단순한 특징으로 가장 단순한 형태의 특징이 얼마만큼 사용자들을 구분할 수 있는 특징으로 사용될 수 있는지 확인해보는 것도 이 실험의 주요한 목적중의 하나이다. 물론 다양한 특징을 선정하는 것이 분류에 있어서 좋은 결과를 가져올 수 있지만 그 만큼 모델이 복잡해지고, 훈련시간이 길어지는 단점도 가지고 있다.

사용되는 명령어들의 길이(window size)는 50으로 정하고 실험을 하였고 각 사용자들의 명령어들 슬라이딩 사이즈(sliding size)는 10으로 실험하였다. SVM은 특성상 두 가지 이상의 클래스로 구분이 되어야 하므로 여기서는 user i 프로파일을 만들기 위하여 user i의 데이터를 1로 레이블하고 나머지 49명의 데이터는 모두 -1로 레이블 하였다. 나머지 49명의 프로파일을 만들기 위하여 각각 동일한 방법을 사용하였다.

5.3 SVM 커널과 인수 선택

각 사용자들의 데이터에 대한 특징을 추출하고 레이블을 한 다음 SVM을 이용하여 사용자 프로파일을 만들게 되는데, 어떤 커널을 선택하고 또 커널의 인수를 어떻게 선택하는가 하는 것이 모델의 정확성과

신분위장기법 탐지의 성패를 좌우하게 된다. linear function 커널과 작은 차수의 polynomial function 커널의 사용은 감사 데이터의 특징자료를 분리하는데 적절히 않았다. 우리는 실험을 거쳐서 RBF(Radial Based Function) 커널을 선택하였고, 이 커널을 사용하여 실험결과를 완성하였다. 커널의 인수 선택을 위하여 cross-validation을 이용하였으며 데이터의 특성에 따라 또는 데이터의 도메인에 따라 적절한 값을 갖도록 조절해야 할 것이다.

5.4 실험 결과

실험결과에서는 기존의 방법들의 탐지율과 제안된 방법의 결과를 비교하고 실험에서 제안된 최종 탐지기의 역할을 알아본다.

5.4.1 SEA 결과

SEA data configuration을 이용하여 SVM을 적용한 실험결과는 [표 8]에서 보는 것과 같이 기존의 탐지율을 최소 15%이상 향상시키는 결과를 보여준다. Schonlau의 기존 6가지 실험의 결과, 그리고 Maxion의 실험결과와 비교했을 때 SVM이 상대적으로 false alarm 즉 false positive의 비율은 높지만 misses 즉, false negative의 비율은 50%이상 향상시키는 결과를 보여준다.

침입탐지 시스템을 고려할 때 두 가지 중요한 요소가 있는데 허위 경보는 줄이고 탐지율은 높여야 한다는 것이다. 허위 경보는 두 가지로 구분될 수 있는데 false positive와 false negative의 두 가지 요소이다. false positive는 침입, 공격 또는 이상행위가 아닌데 침입, 공격 또는 이상행위로 판단하는 경우를 의미한다. 이는 기존의 실험에서 false alarm으로 분류한 것으로 [표 8]에서 false alarm[7]으로 기록되어 있는 부분이다. 또한 false negative는 침입, 공격 또는 이상행위가 발생하였으나 탐지하지 못하고 정상인 상태로 인식하는 것으로 기존연구에서 misses로 표시되는 부분[8]이다. Maxion의 실험결과에서도 언급한

[표 8] "Cost = False Negative + False Positive"의 순위 함수와 SEA data configuration을 이용한 분류 방법의 순위 비교

Methods	Hits(%)	Misses(%)	FA(%)	Cost
SVM	80.1	19.9	9.7	29.6
Bayes 1-step Markov	69.3	30.7	6.7	37.4
N. Bayes(no Upd)	66.2	33.8	4.6	38.4
N. Bayes(Updating[1])	61.5	38.5	1.3	39.8
Hybrid Markov	49.3	50.7	3.2	53.9
IPAM	41.1	58.9	2.7	60.6
Uniqueness	39.4	60.6	1.4	62.0
Sequence Matching	36.8	63.2	3.7	66.9
Compression	34.2	65.8	5.0	70.8

1) 탐지를 하는 과정에서 침입이 아닌 데이터가 들어오면 지속적으로 새로운 데이터를 추가하여 모델을 만드는 것을 말한다. 이렇게 새로운 정상데이터를 추가함으로써 사용자의 변화하는 행동양상을 지속적으로 모델링 할 수 있다.

것처럼 탐지기의 성능은 결국 false positive 비율을 최소화 시키고 탐지율을 극대화 시키는 것 즉, false negative 비율을 줄이는 것을 의미한다. 이것은 아래와 같이 정의된 비용(Cost: C)을 줄이는 것을 의미한다.

$$Cost = \alpha(False\ \ Negative) + \beta(False\ \ Positive) \quad (2)$$

이 순위 함수(ranking function)를 Maxion은 $Cost = \alpha(Misses) + \beta(False\ alarm)$로 표시하였으나 그 의미는 동일하다. 최상의 α와 β값[9] 은 도메인에 따라 매우 다양하게 결정되어진다. 일반적인 경우 false negative의 값을 줄이는 것이 false positive의 값을 줄이는 것 보다 우선시 되므로 α와 β값은 $\alpha \geq \beta$값인 상태로 결정되는 경우가 일반적이라고 볼 수 있다. False positive를 최대한 고려한다고 하더라도 $\alpha = \beta = 1$이고 [표 8]에서는 이런 경우의 비용 값을 보여준다. 이러한 순위 함수를 사용하여 보면 SVM, Bayes 1-step Markov, N. Bayes 모델의 순서로 탐지의 비용이 증가하며 SVM이 신분위장기법 탐지에 효과적으로 사용되었음을 알 수 있다.

5.4.2 1 vs 49 결과

1 vs 49 data configuration에 SVM을 이용한 신분위장기법 탐지를 한 결과는 [표 9]와 같다. 각 사용

7) SEA data configuration에서는 총 50명의 사용자에 대하여 각각 0회 내지 최대 24회의 신분위장자의 행위가 삽입되어 있고 이것의 총 회수는 231회이다. 그러므로 정상적인 사용자로 묘사되어 있는 블록의 총수는 5,000 - 231 = 4,761(개)이다. false alarm 비율은 정상적인 사용자의 블록 즉 총 4,761개의 블록 중 신분위장자라고 표시된 부분의 블록의 비율이다.

8) 여기에서는 총 231회의 신분위장자의 행위 중 탐지하지 못하는 부분을 의미한다.

9) Maxion의 경우 $\alpha = 1$, $\beta = 6$의 값을 이용하여 cost를 계산하였다. Maxion의 경우에는 false negative alarm보다 false positive alarm의 중요성을 6배로 강조하였고, 이는 false positive alarm의 최소치와 최대치의 비율을 근거로 하였다.

〔표 10〕 Confusion Matrix

Victim	Intruder											Missed Incursions
	1	2	3	4	5	6	7	8	9	10	…	
1	0(0)	0(2)	1(40)	18(47)	6(41)	0(1)	5(40)	0(3)	3(50)	0(9)	…	240(1296)
2	0(0)	0(0)	0(3)	0(3)	0(4)	0(0)	0(4)	0(6)	0(0)	0(1)	…	13(359)
3	5(20)	0(0)	0(0)	0(15)	1(12)	0(1)	3(16)	0(1)	0(3)	0(9)	…	100(549)
4	8(35)	1(3)	0(2)	0(0)	3(44)	0(0)	2(36)	1(14)	0(48)	0(5)	…	199(1110)
5	1(10)	0(3)	1(2)	1(18)	0(0)	0(0)	6(32)	0(3)	1(34)	0(2)	…	244(1028)
6	0(13)	0(0)	0(0)	0(1)	0(0)	0(0)	0(9)	0(0)	0(9)	0(5)	…	34(212)
7	4(36)	0(3)	1(4)	1(44)	2(50)	0(2)	0(0)	0(3)	2(49)	0(4)	…	319(1299)
8	0(20)	0(1)	0(2)	0(38)	0(36)	0(0)	1(9)	0(0)	0(14)	0(0)	…	49(611)
9	0(37)	0(3)	0(2)	2(45)	3(46)	0(1)	1(4)	0(2)	0(0)	0(14)	…	113(1162)
10	2(9)	0(0)	0(2)	0(25)	0(0)	0(3)	0(14)	0(2)	0(26)	0(0)	…	37(365)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Successful Intrusions	103 (1124)	9 (402)	34 (339)	117 (1405)	221 (1617)	33 (74)	121 (1379)	56 (360)	49 (1359)	3 (270)	…	

자별로 49 * 50 = 2,450개의 침입자 블록이 묘사되었고 이들 신분위장자의 행위를 94.74% 탐지 하였다. false positive 즉, 허위 경보의 비율은 0%였다. SVM을 이용한 결과 탐지율은 50%이상 향상시켰으며 이것은 SVM을 사용하여 효과적으로 자신의 모델을 일반화시키고 다른 사용자들과의 차별성을 충분히 나타내는 모델을 만들었다는 것을 보여주고 있다.

[표 10]은 Maxion의 Confusion Matrix와 SVM을 이용하여 만들어진 Confusion Matrix 중 일부를 발췌한 결과[10]이다. 여기서 보이는 것처럼 Maxion의 실험결과 보다 SVM을 이용한 실험에서 missed intrusion(false negative)이 상대적으로 많이 줄었음을 알 수 있다.

〔표 9〕 1vs49 Data configuration을 이용한 실험결과

Methods	Hits(%)	FA(%)
SVM	94.74	0.00
N. Bayes (Base Result)	66.2	4.6

5.4.3 공통명령어와 최종 탐지기를 통한 모델의 일반화

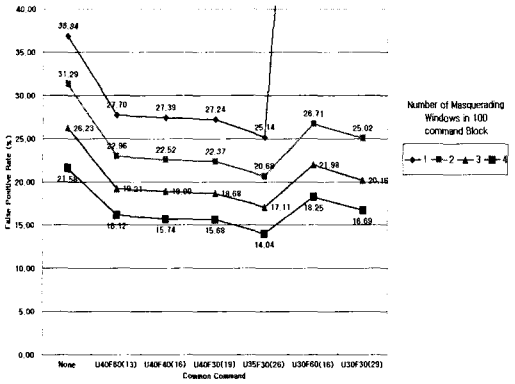
이 실험은 크게 두가지의 의미를 갖는다. 첫 번째는 가장 단순한 특징인 명령어 사용빈도를 선택해서 기존의 실험에 대하여 상대적으로 훨씬 효과적 탐지를 할 수 있었다는 것이다. 물론 좀더 효과적인 특징을 선택함으로써 탐지율을 더욱 향상시키고 허위 경보를 더욱 최소화시킬 수 있을 것이다. 두 번째로 SVM이 신분위장기법 탐지에 효과적으로 사용될 수 있는 분류기(classifier)라는 것을 입증했다는 것이다.

가장 단순한 명령어의 사용빈도가 이렇게 효과적인 특징으로 사용될 수 있었던 과정에는 2가지 중요한 요소가 있었기 때문이다. 첫 번째는 특징에 "공통명령어"라는 개념을 도입한 것이다. 모든 명령어를 각각의 특징으로 사용했을 경우 모델을 일반화 하는데 있어서 많은 문제가 발생하게 된다. 실제로 [그림 7]에서 보여지는 것처럼 공통 명령어를 사용하지 않았을 경우 false positive 비율은 20% 이상이 된다. 그러나 공통 명령어를 특징으로 선택할 경우 그 비율은 줄어들게 되며 공통 명령어를 사용자수의 35%이상 명령어의 사용빈도가 30%이상이 되는 명령어로 정의 할[11] 경우 false positive 비율은 14.04%까지 줄어들게 된다.

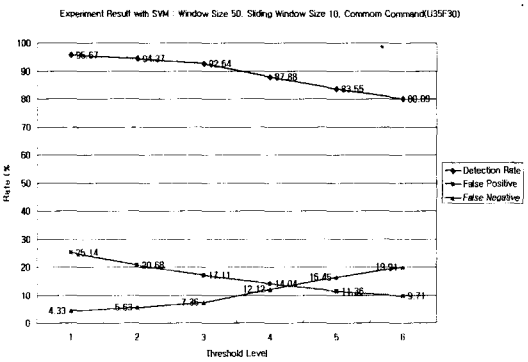
그러나 허위 경보의 비율은 아직 높다. 이것을 좀더 줄이기 위하여 도입한 것이 바로 최종 탐지기이다. [그림 8]에서 보는 것과 같이 최종결정을 위한 임계값을 어떻게 설정하는가에 따라 허위 경보 비율

10) 숫자가 SVM을 이용한 결과이고, 괄호 안의 숫자가 Maxion의 N. Bayes classifier를 이용한 실험결과이다. User 1을 대상으로 한 경우 SVM은 총 240개의 공격을 탐지하지 못했지만 N. Bayes classifier의 경우 1,296개의 공격을 탐지하지 못하였다.

11) 이렇게 정의 할 경우 감사 데이터로 사용된 명령어 총 852개 중에서 26개의 명령어가 공통명령어(common command)로 정의된다.



〔그림 7〕 공통 명령어의 영향



〔그림 8〕 Voting Engine의 효과

은 줄어들게 된다. 물론 허위 경보 비율이 줄어듦에 따라 탐지율도 변하게 된다. 탐지율과 허위 경보 비율은 서로 트레이드오프 관계에 있으므로 각 사이트의 특성에 따라 적당한 임계값을 설정하여 사용하여야 한다.

Ⅵ. 결론 및 향후 연구 방향

신분위장기법 탐지는 이전까지의 침입탐지 분야에서의 구분에서 볼 때 좀 더 새로운 관점을 제공한다. 즉 사용자의 요구 또는 요청 행위자체가 특정한 공격행위인가 아닌가를 구분하는 것이 아니고 현재의 요구나 요청이 정상적인 사용자의 것인지 아닌지가 초점이 되는 것이다. 이러한 문제는 수많은 사용자들을 제공하는 현재의 네트워크 환경, 인터넷 환경에서 더욱 중요한 문제로 부각되게 된다. 이 문제를 직접적으로 해결하게 위하여 시도된 노력은 아직 많지는 않지만 그동안의 이상탐지 분야의 연구가 많은 초석을 제공하게 될 것이다. 이러한 관련 연구를 바탕으로 이 연구에서는 감사 데이터로 가장 쉽게 구할 수 있는 사용자의 명령어를 이용하여 신분위장기법 탐지를 시도 하였고, 요즘 가장 효과적인 분류기로 사용되고 있는 SVM을 분류기로 이용하여 기존의 실험결과에 대하여 효과적인 탐지를 할 수 있도록 하였다. 이를 통하여 SVM이 신분위장기법 탐지를 위한 효과적인 분류기로 사용될 수 있음을 보여주었다. 또한 명령어의 사용빈도라는 가장 단순한 특징으로 이용하였음에도 불구하고 효과적인 분류가 가능하였으며 이를 위하여 공통명령어(common command)와 최종 탐지기의 두 가지가 중요한 역할을 하였다.

기존의 연구에 비하여 향상된 탐지율을 제공함에도 불구하고 허위경보 비율은 상용으로 사용하기에는 높은 수준이다. 이는 몇 가지 방법으로 향상되어질 수 있는데 첫 번째, 업데이트 모듈을 추가하는 것이다. 사용자의 행위는 시간에 따라 변화하므로 최근에 사용된 데이터와 오래된 데이터와의 사이에는 그 중요도가 다르게 되므로 이것을 반영할 수 있어야 하고 이것은 업데이트 모듈을 통하여 시간이 지남에 따라 계속 모델을 업데이트 할 수 있어야한다. 둘째는 좀더 복잡하고 효과적인 특징의 선택이다. 인공지능 분야의 기계학습의 가장 중요한 문제는 바로 가장 효과적인 특징을 찾아내는 것이라고 할 수 있다. 가장 단순한 특징을 선택했음에도 기존 연구에 비하여 상대적으로 효과적인 결과를 얻었지만 좀더 좋은 특징을 선택함으로서 더 좋은 결과를 얻을 수 있고, 특히 명령어 사이의 연관관계를 반영하는 "순서"(sequence)를 반영하면 더 좋은 결과를 얻을 수 있다. 이 두 가지에 대하여 현재 실험 중에 있다. 마지막으로 현재는 신분위장기법 탐지를 위하여 감사 데이터로 UNIX 환경 하에서의 사용자 명령어를 사용하였다. 앞으로 윈도우 환경과 인터넷 환경으로의 확장 또한 검토 되어야 할 것이다.

참 고 문 헌

[1] Dorothy E. Denning, "Internet Besieged : Countering Cyberspace Scofflaw," ACM Press, 1997.

[2] CSI and FBI, "Computer Security Issues and Trends: 2002 CSI/FBI Computer Crime and Security Survey," Computer Security Institute, 2002

[3] William H. Webster et.al, "A Review of FBI Security Programs, Commission for Review of FBI Security Program," U. S Department of Justice, March

2002.

[4] Rebecca Gurley Bace, "Intrusion Detection," Mcamillan Technical Publishing, 2000.

[5] Thomas H. Ptacek, "Insertion, Evasion and Denial of Services : Eluding Network Intrusion Detection," http://secinf.net/info/ids/idspaper/idspaper.html, Oct 16, 2002.

[6] Rain Forest Puppy, "A Look at Whisker's Anti-IDS Tactics," http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html, Nov 1999.

[7] M. Schonlau, W. DuMouchel, W. Ju, Alan F. Karr, M. Theus, and Y. Vardi, "Computer Intrusion : Detecting Masqueraders," Statistical Science, 16(1) 2001.

[8] Terran D. Lane, "Machine Learning Techniques for the Computer Security Domain of Anomaly Detection," Ph. D Thesis, Purdue University, August 2000.

[9] T. F Lunt and R. Jagannathan, "A prototype real-time intrusion-detection expert system," IEEE Computer Society Press, In IEEE Symposium on Security and Privacy, pp.59-66, 18-21 April 1988, Oakland, California, 1988.

[10] Roy A. Maxion and Tahlia N. Townsend, "Masquerade Detection Using Truncated Command Lines," IEEE Computer Society Press, Proceedings of the International Conference on Dependable Systems and Networks (DSN-02), pp.219~228, June 2002.

[11] S. Mukkamala, G. Janowski, and A. H. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," Proceedings of IEEE IJCNN, pp.1702~1707, May 2002.

[12] S. Mukkamala, G. Janowski, and A. H. Sung, "Intrusion Detection Using Support Vector Machines," Proceedings of High Performance Computing Symposium-HPC 2002, pp.178~183, April 2002.

[13] S. Mukkamala, G. Janowski, and A. H. Sung, "Feature Ranking and Selection for Intrusion Detection," Proceedings of International Conference on Information and Knowledge Engineering-IKE 2002, pp.503~509 June 2002.

[14] Mike Fugate and James R. Gattiker, "Anomaly Detection Enhanced Classification in Computer Intrusion Detection," SVM 2002, LNCS 2388, pp. 186~197, May 2002.

[15] Nello Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods," Cambridge University Press, 2000.

[16] Christopher J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, Vol.2, No.2, pp.121~167, 1998.

[17] Thorsten Joachims, "SVMlight Support Vector Machine," Web page, Ver.5.0, http://svmlight.joachims.org, 2002.

[18] S. Dunnung and H. Chen, "Hierarchical classification of web content," In SIGIR, 2000.

[19] Thorsten Joachims, "Estimating the generalization performance of a SVM efficiently," Proceedings of the International Conference on Machine Learning, 2000.

[20] Yiming Yang and Xin Liu, "A re-examination of text categorization methods," Proceedings of SIGIR -99, 22nd ACM International Conference on Research and Development in Information Retrieval, 1999.

〈著者紹介〉

김 한 성 (Han-Sung Kim) 학생회원
1990년 3월 : 육군사관학교 전산학과 졸업(이학사)
1995년 9월 : University of Western Ontario(캐나다) 전산학과 졸업(석사)
2001년 3월~현재 : 한국과학기술원 전산학과 박사과정
<관심분야> 정보보호, 네트워크 보안, 침입탐지

권 영 희 (Younghee Kwon) 학생회원
2000년 2월 : 한국과학기술원 전산학과 졸업(학사)
2002년 8월 : 한국과학기술원 전산학과 졸업(석사)
2002년 9월~현재 : 한국과학기술원 전산학과 박사과정
<관심분야> 패턴인식, 인공지능, 기계학습

차 성 덕 (Sung-Deok Cha) 정회원
1983년 : UC Irvine 전산학과 졸업(학사)
1986년 : UC Irvine 전산학과 졸업(석사)
1991년 : UC Irvine 전산학과 졸업(박사)
1994년~2001년 : 한국과학기술원 조교수
2001년~현재 : 한국과학기술원 부교수
<관심분야> 정형기법 및 명세, 정보보호, 침입탐지