순서기반 비정상행위 탐지 센서의 임계치 결정 방법

김 용 민[†] · 김 민 수[††] · 김 홍 근[†††] · 노 봉 남[††††]

요 약

본 논문은 SOM과 HMM을 이용하여 시스템 호출 수준에서 순서기반의 비정상행위 탐지 센서를 구현하였다. 그리고, 시스템 호출에서 중요한 정보가 무엇이고 임계값은 어떻게 설정해야하는 지를 분석하였다. 본 논문에서 사용한 SOM의 새로운 필터링 규칙과 축약 규칙은 HMM의 입력 크기를 줄일 수 있었다. 이러한 축약은 HMM기반 비정상행위 탐지의 실시간 처리능력을 보장해 준다. 또한, 비정상행위 수라는 개념을 도입하여 HMM의 탐지결과에 대한 민감성을 둔화시켜서, 사용자가 탐지결과를 쉽게 이해하고 false positive를 줄이는 효과가 있었다. 그리고, 능동적으로 threshold 값을 조정하여 시스템 상황에 따라 탐지센서가 적응할 수 있도록 하였다.

The Decision Method of A Threshold in Sequence-based Anomaly Detection Sensor

Yong-Min Kim[†] · Min-Soo Kim[††] · Hong-Gun Kim[†††] · Bong-Nam Noh[††††]

ABSTRACT

In this paper, we implement sequence-based anomaly detection sensor using SOM and HMM, and analyze what is important information in system call and how a threshold is decided. The new filtering and reduction rules of SOM reduces the input size of HMM. This gives real time processing to HMM-based anomaly detection sensor. Also, we introduced an anomaly count into the sensor. Due to lessened sensibility, a user easily understand easily the detection information and false positive was decreased. And the active coordination of the threshold value makes the detection sensor adapt according to the system condition.

키워드 : 침입탐지시스템(IDS), 비정상행위 탐지(anomaly detection), HMM, SOM

1. 서 론

Denning이 침입탐지 시스템에 대한 기본적인 모델을 제시한 이래로 지금까지 많은 침입탐지 시스템(Intrusion Detection System)이 발표되었다. 기존의 침입 탐지 시스템은 각기 다른 특징을 가지고 있다. 침입 탐지 시스템은 알려진 공격을 탐지하는 오용 탐지와 비정상적인 행위를 탐지하는 방법에 따라 크게 분류된다. 그리고, 어떠한 방법으로 침입 탐지를 수행하는가, 탐지 범위는 어디까지 하겠는가에 따라 여러 종류로 나뉘어진다[18]. 그리고, 침입탐지 시스템의 평가에 대한 기술적 실제적 논점은 실제 침입이 아닌데 침입으로 판정하는 경우(false positives)와 실제 침입인데 탐지하지 못하는 경우(false negatives)를 어떻게 처리하는가에 있다[3]. 좋은 침입 탐지 시스템은 낮은 false positives와 false negatives를 갖는 것이다. 일반적으로 침입탐지 시스템에 대한 성능평가는 false-negatives, false-positivie, 실시간성, 자원사용량, 견고성 등으로 이루어진다. 특히 비정상행위 탐지 방법에서는 false-positives를 줄이고 학습 시간 및 탐지 시간을 줄이는 것이 관건이다.

본 논문에서는 시스템 호출 정보를 기반으로 하였으며 발생하는 시스템 호출의 순서를 HMM(Hidden Markov Model)[10] 알고리즘을 이용하여 정상행위를 학습한다. HMM은 학습을 통해서 사용자 행위에 대한 일련의 시퀀스를 모델 행렬 속에 내포시킴으로써 사용자 행위의 유사도를 검사할 수 있도록 한다. 시스템 호출을 이용할 때 HMM을 이용한 비정상행위 탐지 방법의 효능은 Warrender가 분석하였다[14]. 그러나, 시스템 호출 정보의 경우 단시간에 대량의 로그를 생성하게 되므로 많은 계산을 필요로 하는 HMM 모델에는 적합하지 못하다. 그래서, HMM 모델의 입력에 사용되는 데이터는 필터링과 축약과정을 거쳐서 나온 적은 량을 사용하도록 하였다. 본 논문에서 사용한 방법은 시스템 호출에 필터링 규칙을 적용하였고, [17]와 같은 방법으로 HMM 모델에 적용하기 위한 적은 크기의 데이터를 생성하기 위해 SOM(Self-Organizing Map)[5]을 이용한 축약을 수행하였다.

이러한 SOM과 HMM을 동시에 적용하는 방법에는 많은 장·단점이 있을 수 있다. SOM은 입력데이터를 분류하는 특성이 있는 반면에 HMM은 이벤트의 발생 순서를 확률 네트

※ 한국정보보호진흥원 위탁과제관련 논문임.
† 정 회 원 : 전남대학교 대학원 전산통계학과 박사과정
†† 종신회원 : 전남대학교 정보보호협동과정 객원교수
††† 정 회 원 : 한국정보보호진흥원 기술개발팀장
†††† 종신회원 : 전남대학교 컴퓨터정보학부 교수

논문접수 : 2001년 8월 1일, 심사완료 : 2001년 9월 26일

워크로 표현하여 이벤트 순서를 잘 나타낸다. 장점은 SOM의 분류 특성을 이용하여 데이터 양을 줄이고 HMM을 이용하여 순서를 표현함에 따라 침입행위의 시퀀스를 정의할 수 있게 된다. 또한, SOM 축약으로 인하여 HMM의 계산량을 줄이게 된다. 단점은 SOM과 HMM의 이중 잣대를 적용함으로써 이벤트의 특성이 중화될 수 있다. 또한, 계산의 량이 많아지며 절차가 복잡해진다. 실제적으로 [17]에서는 실시간 침입탐지가 이루어지지 않았으며, false positive가 높게 나타났다.

따라서, 본 논문에서는 HMM을 이용한 호스트 비정상행위 탐지센서를 구현하는데 있어서 성능향상을 위해 임계 변수의 설정에 따라 false-positive를 줄이고, 실시간성을 반영하며, 사용자별 학습 처리로 능동적인 기능을 강화하는 방법을 연구하고자 한다. 그리고, SOM에 의한 축약과 HMM에 의한 탐지를 적용한 모델의 발생 가능한 문제점을 해결하고, 여기에서 사용되는 임계값의 정의와 데이터의 의미를 찾을 것이다.

2. 비정상행위 탐지 시스템

침입이란 컴퓨터가 사용하는 자원의 무결성, 비밀성, 가용성을 저해하는 일련의 행위들의 집합을 말한다[2]. 또 다른 정의에서는 컴퓨터 시스템의 보안 정책을 파괴하는 행위를 말한다. 이러한 침입 행위는 해마다 증가하고 있으며 방법도 다양화되고 있다.

침입 탐지 시스템이란 시스템의 비정상적인 사용, 오용, 남용 등을 알려주는 시스템이다[2,3,8]. 이러한 시스템은 단일 컴퓨터나 또는 네트워크로 연결된 여러 컴퓨터를 감독한다. 침입탐지 시스템은 감사 기록, 시스템 테이블, 네트워크 부하(traffic) 기록의 자원으로부터 사용자 행위에 대한 정보를 분석함으로써 수행된다.

침입탐지 시스템의 목표는 크게 두 가지 방법으로 설명된다. 하나는 침입자에 의한 불법적인 사용을 명시하는 것이고 다른 하나는 합법적인 사용자에 의한 오용이나 남용을 알아내는 것이다[6].

2.1 오용 침입

오용(misuse) 침입이란 시스템이나 응용 프로그램의 약점을 통하여 시스템에 침입할 수 있는 잘 정의된 공격 형태를 말한다[9, 13]. 예를 들면, 버퍼 오버플로우 공격이나 대규모 패킷을 생성하는 서비스 거부 공격이 오용 침입의 대표적인 경우라고 말할 수 있다.

오용 탐지는 침입자의 행위에 대한 시나리오를 결정하여 저장한다. 시스템에 접속하여 사용하는 행위에서 이러한 시나리오와 같은 행위를 침입이라고 판정한다. 즉 알려진 침입 패턴을 이용한다. 이러한 패턴은 명령어에 대한 시나리오나 시스템 호출에 대한 시나리오를 선택하여 적용된다. 따라서, 오용 탐지는 규칙 기반 침입 확인 방법이 가장 많이 사용되고 상태 변이 방법도 종종 사용된다. 오용 침입 탐지 방법으로는 조건부 확률(conditional probability), 전문가 시스템, 상태 전이 분석, 키스트로크(keystroke) 관찰, 모델에 근거한 침입 탐지, 패턴 매칭 등이 있다[18].

2.2 비정상적인 침입

비정상적인 침입이란 컴퓨터 자원의 비정상적인 행위나 사용에 관한 침입을 말한다[1, 9, 13]. 예를 들면, 컴퓨터 주인이 자리를 잠시 비운 사이에 동료가 들어와 중요한 내용을 복사해 가는 경우를 들 수 있다. 즉, 일반적인 행위 패턴으로부터 벗어남을 탐지하는 방법이다. 알려진 침입 방법뿐만 아니라 알려지지 않은 침입 방법도 탐지할 수 있다는 장점이 있다. 그러나, false positive가 높아지는 단점이 있다. 비정상행위 탐지 방법으로 통계적인 방법, 특징 추출, 비정상적인 행위 측정방법의 결합, 예측 가능한 패턴 생성, 신경망(neural networks), 데이터마이닝 등이 사용된다[12]. 위 방법은 어떠한 특성이나 매개체를 사용하느냐에 따라 분류하였을 뿐 본질적으로 이상 침입을 탐지하는 목적은 같아서 거의 유사하다. 이중에서 가장 많이 사용했던 방법은 통계적인 방법으로 과거의 경험적인 자료에서 침입을 탐지하기 때문에 자료의 양이 많을수록 정확하게 침입을 탐지할 수 있다. 그 외의 방법은 다른 특성이나 매개체를 사용하여 이상 침입을 탐지할 수 있다는 가능성을 보이는 의의가 있을 뿐, 현실적으로 사용하기에는 많은 문제점을 가지고 있다. 또한 각 방법은 독립적으로 사용하기보다는 각각의 단점을 보완하기 위해 서로 결합하여 사용하는 것이 현실적으로 많은 이점을 제공할 수 있다.

실제로 시스템에서 비정상행위라는 것은 합법적인 사용과 오용 사이의 모호한 행위이다. 비정상행위 탐지 시스템의 경우 일반적으로는 합법적인 사용을 기준으로 하나 시스템의 오용을 기준으로 하고 있는 경우도 있다[18].

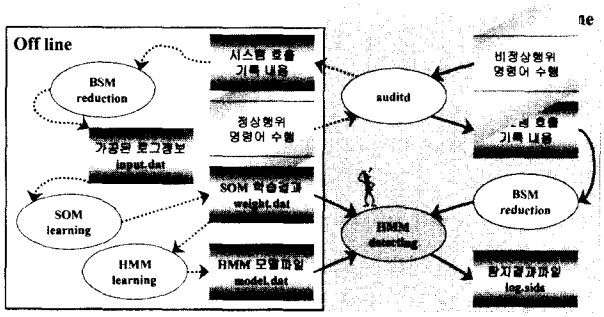
3. 순서기반 비정상행위 탐지방법

본 논문에서는 비정상행위에 대한 순서기반 탐지를 수행한다. 현재 구현된 시스템은 Solaris 2.4 ~ 2.7에서 사용할 수 있도록 되어 있다. 탐지하기 위한 기본 정보로는 커널에서 제공해주는 시스템 호출 로그 정보(Solaris에서 BSM)를 사용한다. 만약 명령어 수준에서 탐지한다면, 새로운 명령어에 대한 정보가 없는 상황에서 사용자가 새로운 파일을 생성하는 경우에 비정상이라고 판단하는 상황이 발생한다. 이러한 문제 때문에, 호스트 기반에서 탐지는 주로 시스템 호출을 사용하고 있다.

본 논문에서는 사용자 행위에 대한 일련의 시퀀스를 기반으로 하기 때문에, 순서를 잘 표현할 수 있는 HMM을 사용하기로 하였다. 그런데, 기본 로그 데이터가 시스템 호출이므로 일정 시간에 많은 자료가 발생하게 되어 있다. 더군다나 HMM 방법에서는 최적으로 매핑하는 시퀀스를 찾기 때문에 많은 실수 연산이 이루어진다. 방대한 시스템 호출 정보를 모두 수

용하는 것은 실시간 탐지를 어렵게 만든다. 따라서, 본 논문에서는 HMM 입력 데이터를 축약과정을 거쳐 단순화시켰다.

(그림 1)은 본 논문에서 사용한 비정상행위 탐지센서의 전체 정보 흐름을 볼 수 있다. HMM 학습 및 탐지를 수행하기 전에 BSM 축약과 SOM 학습 및 축약을 거치게 된다. BSM 축약(reduction) 모듈에서는 Solaris BSM 로그를 필터링 규칙을 통해 일정한 형태의 데이터로 만들고, SOM 축약모듈에서는 BSM 축약 결과를 고정 크기의 데이터로 변환한다.



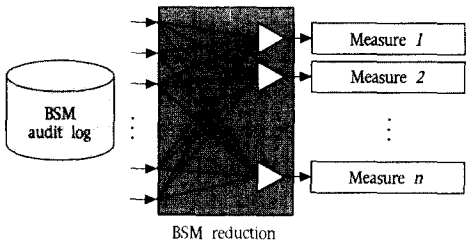
(그림 1) HMM을 이용한 비정상행위 탐지센서 구조도

3.1 BSM 축약

Solaris BSM을 이용한 전처리 기법은 BSM으로부터 생성된 많은 양의 감사기록에서 불필요한 자료를 제거하고 여과된 자료를 축약하여 다음 모듈이 침입탐지에 사용할 수 있도록 한다. 이때 생성된 순서적인 정보는 다음 판정모듈에서 사용되는 것으로 중요정보가 제거되거나 잘못된 축약은 탐지율의 저하를 초래하게 되므로 탐지에 중요한 영향을 미칠 수 있다[7].

3.1.1 정보의 축약

데이터 축약 모듈은 호스트 기반 침입탐지를 위해 BSM audit 데이터로부터 중요정보를 추출하고 추출된 정보들의 크기를 빈도정보나 사용범위 정보에 의해 줄인다. 이러한 정보들은 다변량이므로 HMM을 이용한 정상행위 모델링에 이용될 수 있도록 고정크기를 갖는 일차원 정보로 변환한다. 입력으로는 BSM의 audit 데이터가 사용되며 이러한 입력에 대해 고정크기의 대표값이 출력된다. (그림 2)에서는 BSM audit 로그를 입력으로 주었을 때, BSM 축약(reduction) 과정에 의해 일정한 형태의 measure값이 나오는 것을 보여주고 있다.



(그림 2) BSM reduction

이러한 감사기록이 가지는 문제점은 감사범위에 따라 다르지만 감사기록의 양이 방대하다는 점이다. 모든 이벤트에 대해 감사를 할 경우 하루에도 수 백 Mbyte이상의 감사기록을 남기게 된다. 이러한 문제는 시스템관리자의 적절한 감사범위 지정으로 어느 정도는 해결될 수 있지만 침입을 위해 처리되어야 할 정보량을 고려해보면 너무 방대한 양이다. 따라서 이러한 감사기록에서 필요한 데이터를 추출하고 축약할 필요가 있다.

감사기록에서 필요한 정보를 추출하기 위해 기존의 시스템에서는 시스템 호출, 파일 액세스, CPU 사용시간, I/O 시간, 명령어 등과 관련된 measure를 추출하고 있다. 이러한 정보들 또한 어떠한 방법으로 사용되는가에 따라 정보의 크기가 축소되어져야 한다. 정상행위의 패턴을 살펴보면 이러한 measure들이 가지는 값은 대개의 경우 sparse matrix 형태를 보이게 되는데 measure들이 사용되는 값의 범위나 빈도 정보를 이용하여 정보의 크기를 줄일 수 있다[17].

BSM 감사기록에서 추출될 수 있는 정보는 많지만 추출 가능한 모든 정보를 그대로 사용하지 않고 필요한 정보만을 추출해야 한다. 여기서 필요한 정보량은 실시간 탐지에 있어서 탐지시간과 관련되는 중요한 변수로 볼 수 있다. 많은 양의 정보를 추출하여 사용한다면 많은 처리시간이 요구되며 적은 양의 정보만을 추출한다면 탐지율의 저하를 초래할 수 있다. 탐지에 필요한 효율적 measure들을 선별적으로 추출하고 이를 효과적으로 탐지에 사용하는 것이 일반적인 방법이다. BSM감사기록에서 추출된 정보의 축약에는 시스템 호출 정보를 선별하는 축약과 데이터를 작은 크기의 정보로 변환하는 축약이 있다.

3.1.2 필터링에 의한 축약

감사 기록에 대한 빈도수와 중요도에 따른 축약 방법이다.

〈표 1〉 선별된 시스템호출 목록(Solaris 시스템)

event no.	event name	event no.	event name	event no.	event name	event no.	event name	event no.	event name
1	exit(2)	22	readlink(2)	48	utimes(2)	111	core dumped	239	sysinfo(2)
2	fork(2)	23	execve(2)	51	setrlimit(2)	183	socket(2)	6147	cron-invoke
4	creat(2)	25	vfork(2)	76	open(2)_w	184	sendto(2)	6151	inetd
5	link(2)	32	connect(2)	77	open(2)_wc	185	pipe(2)	6152	login-local
6	unlink(2)	33	accept(2)	78	open(2)_wt	191	recvfrom(2)	6154	login-telnet
8	chdir(2)	34	bind(2)	79	open(2)_wct	200	old setuid(2)	6155	login-rlogin
10	chmod(2)	35	setsockopt(2)	80	open(2)_rw	203	old nice(2)	6158	rsh access
11	chown(2)	42	rename(2)	81	open(2)_rwc	205	old setgid(2)	6159	su
15	kill(2)	46	shutdown(2)	82	open(2)_rwt	214	setegid(2)	6162	rexecd
21	symlink(2)	47	mkdir(2)	83	open(2)_rwct	215	seteuid(2)	6165	ftp access

Solaris의 경우 0~65535의 시스템호출 영역중 정의된 시스템 호출은 500여개인데, 실제로 자주 사용되는 시스템 호출은 100여개 정도이다. 여기에서 선별한 시스템 호출은 100여개 시스템호출에서 실제 공격 패턴에서 중요하게 여기는 시스템 호출을 선택하고 어떠한 프로세스에서나 공통적으로 보이는 시스템 호출은 제거하는 방법을 사용하였다. <표 1>에서는 이렇게 선별된 50개의 시스템 호출을 보여준다.

3.1.3 작은 크기로 변환하는 축약

탐지에 사용되는 measure들은 침입탐지 시스템의 목적에 따라 다르다. 침입탐지 시스템에서 사용되어져야 할 표준적인 measure들은 정해져 있지 않다. 통계적 기법을 사용하는 시스템에서는 다변량을 유지하는 것이 중요할 것이며, HMM이나 신경망을 사용할 경우는 탐지에 필요한 최소의 measure들만을 사용하는 것이 더 효율적일 수 있다[15]. 아래는 기존의 시스템에서 많이 사용되며 실제 침입 패턴에서 중요한 measure들로서 BSM에서 추출 가능한 measure를 보여주고 있다.

- 시스템 호출 관련 : systemcall number, return value, return status
- 프로세스 관련 : process ID, exit value, exit status
- 파일 액세스 관련 : file name, path, file system ID, file permission

이러한 measure에 대한 테이블을 설정하고 사용되는 값을 테이블의 위치로 매핑시키면 간단하게 measure의 크기를 줄일 수 있다. <표 2>에서는 시스템에서 중요한 파일과 경로에 대한 테이블을 보여주고 있다. 이러한 매핑 테이블의 색인만을 계산에 사용하므로 데이터의 크기를 줄일 수 있다.

본 논문에서는 각 measure에 대한 엔트로피(entropy)를 검사한다. 엔트로피는 데이터를 분류(classification)할 때, 다양한 분포로 나뉘어지면 큰 값을 갖지고 일관된 분포로 단순화되면 작은 값을 가진다. 따라서, 좋은 measure는 엔트로피가 크게 나타나는 것이다[16].

〈표 2〉 중요 파일과 경로에 대한 테이블

중 요 경 로		중 요 파 일	
/.dt	/.hotjava	.Xauthority	.Xdefaults
/.ssh2	/devices	.bash_history	.bash_logout
/etc/cron.d	/etc/default	.bash_profile	.bashrc
/etc/inet	/etc/init.d	.cshrc	.desksetdefaults
/etc/mail	/etc/rc0.d	.dtprofile	.forward
/etc/rc1.d	/etc/rc2.d	.group.lock	.htaccess
/etc/rc3.d	/etc/rcS.d	.htpasswd	.kshrc
/etc/security	/etc/ssh2	.lgmsg	.login
/opt	/proc	.logout	.mailrc
/sbin	/tmp	.mailcap	.mnttab.lock
/usr/4lib	/usr/bin	.mysql_history	.profile
/usr/etc	/usr/kernel	.pwd.lock	.rhosts
/usr/lib	/usr/local	.sh_history	Makefile
/usr/openwin	/usr/proc	core	dfstab
/usr/sbin	/usr/share	group	hosts
/usr/snadm	/usr/ucb	hosts.allow	hosts.equiv
/usr/ucblib	/usr/var	imapd.conf	index.html
/var/adm	/var/audit	inetd.conf	inittab
/var/crash	/var/cron	license_log	mbox
/var/dt	/var/imap	mnttab	nfssec.conf
/var/log	/var/lp/logs	passwd	power.conf
/var/mail	/var/preserve	printers.conf	resolv.conf
/var/sadm	/var/snmp	rmmount.conf	shadow
/var/spool	/var/statmon	syslog.conf	vfstab
/var/tmp	/vol/rmt	vold.conf	wtmp

3.2 SOM을 이용한 다변량 축약

실시간으로 침입을 탐지하기 위해서 중요한 정보들만을 추출하게 되는데 추출된 다변량 데이터는 통계적인 방법에 의해 사용되어질 수 있지만 HMM과 같은 고정길이의 시퀀스를 필요로 하는 방법에는 적용할 수 없다는 한계가 있다. 즉, 신경망이나 HMM과 같은 여러 인공지능적인 방법에 의해 정상행위를 모델링하기 위해서는 추출된 다차원 정보를 저차원 정보로 변환할 필요가 있는데 이는 다차원 정보는 통계적 기법에서는 이용이 가능하지만 고정크기의 저차원 정보를 요하는 다른 방법에는 사용될 수 없기 때문이다[17].

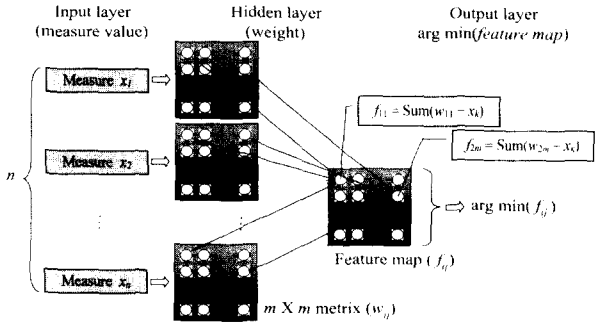
학습과정은 3단계로 나누어지는데 우선 감사기록에서 사용되는 measure들의 값의 범위를 저장하고 이러한 정보를 이용하여 정보의 크기를 정규화시킨 후, SOM을 입력으로 사용하여 SOM의 가중치를 학습시킨다. 이러한 과정이 끝나면 SOM에 의한 출력을 이용하여 HMM의 정상행위 모델링 작업이 진행된다.

이러한 SOM의 가중치 학습과 HMM을 이용한 정상행위 모델링 작업이 끝나면 실제 침입 판정을 위해 사용될 수 있는데 실제 침입탐지 시에는 저장된 measure들의 값을 이용하여 자료들의 크기를 축약하고 SOM을 통하여 고정된 크기의 대표값들을 생성한다. 이러한 대표값들은 순서적인 고정길이 시퀀스로 분할되어져 HMM의 침입판정 루틴에 사용된다.

다차원 정보를 일차원 정보로 변환하기 위해서는 통계적인 방법이 사용되지만 여기서는 입력패턴에 따라 자기 조직화하여 이차원상의 대표값으로 출력해주는 SOM을 이용한다. 비교사 학습(unsupervised learning) 신경망인 SOM은 다차원 입력벡터를 Euclidean distance와 같은 유사도 측정을 통해 자기조직화하고 입력 값에 가장 가까운 대표값으로 출력해준다.

3.2.1 SOM[5]

SOM은 Kohonen이 사용한 feature map으로 가장 많이 알려진 인공지능 신경망 알고리즘 중 하나이다. 주로 사용하는 영역은 입력 데이터를 클러스터링하거나 맵의 공간적 순서화(spatial ordering)이다. (그림 3)에서는 SOM을 일반 신경망처럼 3계층으로 구성할 수 있다. 각 measure 값을 입력하는 입력 계층(input layer), 일정한 형태의 출력값을 주는 출력 계층(output layer), 그리고 가중값(weight value)에 따라 값이 조정되는 숨겨진 계층(hidden layer)으로 나눈다. 일정한 형태의 measure 값이 입력되면 숨겨진 계층에서 각 measure와 가중값을 연산하여 feature map을 생성한다. 이러한 feature map 중에서 가장 작은 값을 출력하는 형태로 SOM 축약이 이루어진다.



(그림 3) SOM layer

SOM에서 사용하는 데이터는 입력 데이터와 가중값이다. SOM의 크기는 N × M × L로 표현하는데, N과 M은 가중값에 대한 이차원 배열인데, N과 M을 합쳐서 일차원으로 사용하기도 한다. L은 입력 크기로 척도에 대한 번호이며 입력 시퀀스의 번호이기도 하다. 초기에 가중값 w는 N × M × L로 정의되고 입력값은 일차원 배열 x[L]로 정의된다. η는 반경으로 신경망에서 사용하는 조정값이다. 여기서는 입력값을 0~49 사이의 값으로 설정하고 7개 척도를 측정하므로 x[7]로 선언되었다. 각 척도의 의미는 아래와 같다[19].

x[0] = system call number,
x[1] = file system number,
x[2] = return value,
x[3] = system status,
x[4] = argument length,
x[5] = file permission,
x[6] = filename and path,
x[7] = directory path

3.2.2 SOM 학습 알고리즘

SOM을 이용하여 데이터를 축약하기 위해서는 BSM으로부터 추출된 정보의 크기를 정규화하여 SOM의 입력으로 사용한다. SOM의 이러한 입력값을 잘 분류해내기 위해서는 학습과정이 필요하다. SOM의 학습과정은 매개변수들과 가중치를 초기화하고 조건이 만족할 때까지 입력된 데이터와 유사한 것으로 대표값이 결정되도록 가중치를 갱신하는 반복과정으로 볼 수 있다.

일반적인 SOM의 학습 알고리즘은 다음과 같다.

① weight vector(W)의 초기화($n = 0$)
weight vector W는 아래와 같이 2차원 배열로 구성된다.
$W_j = \{w_{1j}, w_{2j}, \cdots, w_{nj}\}$
② input vector (X)를 가져온다.
$X = \{x(1), x(2), \cdots, x(n)\}$
③ 유사도를 비교한다.
유사도는 feature map에 저장한다.

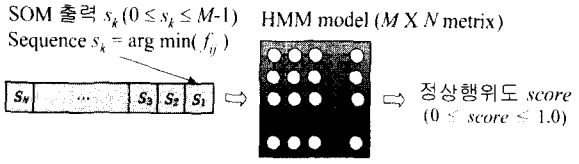
$$Fmap_{ij} = \sum_{t=0}^{L-1} [x(t) - w_{ij}(t)],$$

L : input size

④ feature map이 가장 작을 때의 행렬 색인 i와 j를 구한다.
i and j = arg min ($Fmap_{ij}$)
⑤ weight vector의 i, j 번째 값에 대한 증감치를 구한다.
$\Delta w_{ij}(t) = \eta_j(t) \cdot (x(t) - w_{ij}(t))$
여기에서 η는 학습률을 의미한다.
⑥ weight vector의 값을 갱신한다.
$w_{ij}'(t) = w_{ij}(t) + \Delta w_{ij}(t)$
⑦ 입력 데이터 수만큼 ②에서 ⑥까지 반복 수행한다.

3.3 HMM 학습 모듈

정상행위를 모델링하는 과정으로 사용자의 정상행위 정보를 HMM의 Baum-Welch 재추정식을 사용해서 모델링한다. HMM의 입력으로는 전처리에 의해 필터링되고 축약된 사용자의 순서적인 정상행위 데이터이며 출력은 해당 식별자의 HMM 매개변수를 조정하는 것이다. (그림 4)는 이러한 정상행위 모델링을 위한 HMM을 이용한 학습과정을 보여준다.



(그림 4) HMM 과정

3.3.1 HMM 정의

HMM은 음성인식을 위해 주로 사용되었으나 침입탐지에도 적용할 수 있다. HMM은 실세적인 생성모델을 알 수 없고 단지 생성된 시퀀스에 의해서만 확률적으로 관측할 수 있는 확률적인 절차로서, 사용자의 행위시퀀스를 모델링하기에 유용한 도구이다. 하나의 시퀀스에 대해서 여러 가지의 모델을 가정할 수 있다. 중요한 것은 시퀀스를 효과적으로 설명해주는 모델을 결정하는 것과 시퀀스를 잘 표현하도록 모델을 학습하는 것이다[19].

HMM은 고정된 값인 관찰 시퀀스의 길이, 상태수, 심볼수와 학습에 의해 조정되는 전이확률, 관측확률, 초기상태분포로 구성이 된다. 전이확률은 한 상태에서 다음상태로 전이할 확률을 나타내며, 관측확률은 한 상태에서 특정 심볼이 관측될 확률을 나타낸다. 초기 상태 분포는 처음에 해당 상태에서 시작할 확률을 나타낸다. HMM은 아래와 같이 표현되며, 모델 λ는 학습에 의해 조정되는 변수들로 간략히 (A, B, π)로 나타낸다[4, 11, 17, 19].

- T : 관찰 시퀀스의 길이 (전체 clock time의 수)
- N : 모델의 상태 수
- M : 관찰 심볼의 수
- $Q = \{q_1, q_2, \cdots, q_N\}$: 상태
- $O = \{o_1, o_2, \cdots, o_M\}$: 가능한 관찰 심볼의 이산집합
- $A = \{a_{ij}\}$,
  $a_{ij} = \Pr(q_j \text{ at } t+1 \mid q_i \text{ at } t)$
  : 상태전이 확률분포

- $B = \{b_j(k)\}$
  $b_j(k) = \Pr(o_k \ at \ t \mid q_j \ at \ t)$
  : 관찰 심볼 확률분포
- $\pi = \{\pi_i\}, \ \pi_i = \Pr(q_i \ at \ t = 1)$
  : 초기 상태 분포

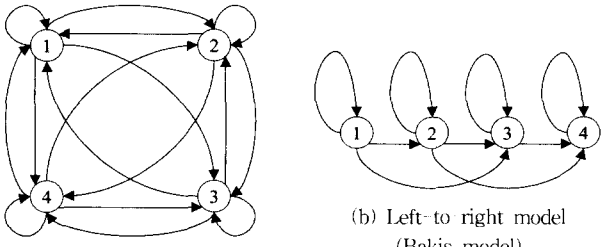
HMM은 여러 가지 형태로 구성될 수 있다. Ergodic 모델은 한 상태에서 다른 상태로 나갈 확률과 다른 상태에서 해당 상태로 들어올 확률이 동일한 형태이다. Left-right 형태는 자기 상태에서 자기로 되돌아오거나 이후 상태로의 전이만을 허용하는 형태이다. 일반적으로 left-right 형태의 HMM이 시간이나 순서적 정보를 잘 표현한다고 알려져 있다. (그림 5)에서는 HMM 모델의 형태를 그래프로 보여주고 있다. 이것을 행렬식으로 보면 다음과 같다.

- 표준 ergodic 모델

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

- left-right 모델 (Bakis 모델)

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$



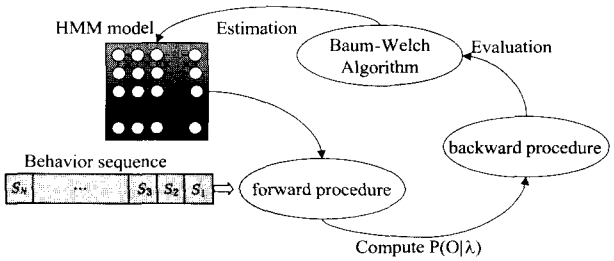
(a) Ergodic model　　(b) Left-to right model (Bakis model)

(그림 5) HMM model 형태

본 논문에서는 두가지 모델 모두 비정상행위 탐지에 적용하여 보았다. 그러나, left-right 모델의 경우 사용자별 정상행위 시퀀스에 대한 학습 결과가 비슷하게 나타나서, 실제 탐지할 때에 오탐율이 높게 나타났다. 따라서, 많은 계산이 필요하더라도 ergodic 모델을 사용하기로 하였다. 여기의 시간을 보상하기 위해 SOM에서 필터링과 축약과정을 거치게 된다.

3.3.2 HMM model

HMM 모델은 (그림 6)에서처럼 4가지 절차를 통해서 학습 및 판정이 이루어진다. 행위 시퀀스(behavior sequence)가 들어오면 forward procedure와 backward procedure를 통해서 $\Pr(O \mid \lambda)$ 값을 구해나간다. 이렇게 계산된 값을 기반으로 Baum_Welch 알고리즘에서 재평가에 필요한 값을 계산한다. 이러한 평가 값은 HMM 모델을 수정하는데 사용된다. 이와 같은 학습과정은 입력 데이터 수만큼 수행되며 그 결과로 생성된 HMM 모델은 실시간 판정에 이용된다.



(그림 6) 정상행위 모델링 모듈 동작 시나리오

HMM 모델에서는 일반적으로 forward procedure에서 생성되는 값으로서 침입 여부를 판단하게 된다. 다른 판정 방법으로는 Viterbi 알고리즘이 있다. 일반적으로 HMM의 판정은 $\Pr(O \mid \lambda)$을 기반으로 이루어진다.

위에서 설명한 HMM을 실제로 적용할 때 다음 세 가지 주요 문제가 제기된다.

(1) 모델학습

$\Pr(O \mid \lambda)$를 최대로 하기 위해 모델 파라메터 $\lambda = (A, B, \pi)$를 어떻게 조정할 것인가의 문제이다. 이는 원 모델을 더 잘 표현하도록 HMM을 학습시키는 문제이다.

Forward-backward procedure에서는 forward 변수인 α와 backward 변수인 β를 사용해서 입력시퀀스가 해당 모델로부터 나왔을 확률 $\Pr(O \mid \lambda)$를 계산한다. Forward 변수 α는 시간 t에 부분관찰 시퀀스 $o_1, o_2, \cdots, o_t$를 보고 상태 q_i에 있을 확률이다.

$$\alpha_t(i) = \Pr(o_1, o_2, \cdots, o_t, i_t = q_t \mid \lambda)$$

이 정의에 따르면 $\alpha_T(i)$는 입력시퀀스 O의 모든 심볼을 순서에 맞게 가지고 있으면서 최종상태가 i인 확률을 나타낸다. $\alpha_T(i)$를 모든 상태 i에 대해 고려하면 $\Pr(O \mid \lambda)$를 구할 수 있다.

Backward 변수 β는 주어진 시간 t에 부분관찰 시퀀스 $o_{t+1}, o_{t+2}, \cdots, o_T$를 보고 상태 q_i에 있을 확률을 의미하며 forward 변수 α와 함께 입력시퀀스가 해당 모델로부터 나왔을 확률 $\Pr(O \mid \lambda)$를 계산한다. Backward procedure에 사용되는 변수 β의 계산은 foward 변수의 계산과 유사한 과정에 의해서 구할 수 있다.

(2) 시퀀스/모델 평가

두 번째 문제는 주어진 관찰 시퀀스 $O = (o_1, o_2, \cdots, o_T)$가 모델 $\lambda = (A, B, \pi)$에 주어지면 효과적으로 $\Pr(O \mid \lambda)$를 구하기 위해 어떠한 방법을 사용할 것인가 이다. 이는 HMM으로 모델을 구축해 놓은 후 관찰된 시퀀스가 해당 모델에 의해 생성 확률을 계산하는 것으로 생각할 수 있다. 역으로, 해당 모델로부터 생성된 시퀀스가 있다면 HMM으로 얼마나 효과적으로 모델링했는지를 평가하는데도 사용할 수 있다.

정상행위 모델링은 전처리 단계에서 생성된 정상행위 시퀀스를 기반으로 HMM의 파라미터를 결정하는 과정이다. HMM

의 파라미터 결정은 주어진 시퀀스 O가 해당 모델 λ로부터 나왔을 확률인 $\Pr(O|\lambda)$값이 최대가 되도록 $\lambda=(A, B, \pi)$를 조정한다. 이를 계산하는 해석적인 방법은 알려져 있지 않고 반복적으로 λ를 결정하는 방법으로 Baum-Welch의 재추정식이 있다.

Baum-Welch 재추정식에서는 두 개의 변수가 추가로 사용된다. $\xi_t(i,j)$는, 시간 t에 상태 q_i에 있다면, 시간 $t+1$에 상태 q_j에 있을 확률로 정의되며 변수 α와 β를 사용하여 구할 수 있다. $\gamma_t(i)$는 시간 t에 상태 q_i에 있을 확률이며 역시 변수 α와 β를 사용하여 구할 수 있다.

두 값을 시간 t에 대해 각각 합을 취하면 하나의 시퀀스에서 각각 상태 i에서 j로 변할 기대값과 상태 i에 있을 기대값을 구할 수 있다. 위의 값이 구해지면 다음 수식에 의해서 새 모델 $\overline{\lambda}=(\overline{a}, \overline{b}, \overline{\pi})$를 구할 수 있다.

$$\overline{\pi_i} = \text{시간 } (t=1)\text{에 } S_i \text{에 있을 빈도} = \gamma_1(i)$$

$$\overline{a_{ij}} = \frac{i\text{에서 } j\text{로 전이할 기대횟수}}{i\text{에서 전이한 기대횟수}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}$$

$$\overline{b_j}(k) = \frac{j\text{에서 } k\text{를 볼 기대횟수}}{j\text{에 있을 기대횟수}} = \frac{\sum_{t=1}^{T-1}\gamma_t(j)\,1_{o_t=k}}{\sum_{t=1}^{T-1}\gamma_t(j)}$$

시퀀스 O를 관찰한 결과로 $\overline{\lambda}$를 구한 후 $\Pr(O|\lambda)$와 $\Pr(O|\overline{\lambda})$를 비교한다. 두 값의 차이가 $\varepsilon(0.00001)$보다 작다면 유도 함수의 임계점에 도달했으므로 재추정 과정을 종료한다.

(3) 최적상태추적

세 번째 문제는 주어진 관찰시퀀스에 대해 최적의 상태시퀀스 $Q=(q_1, q_2, \cdots, q_T)$를 어떻게 선택할 것인가의 문제이다. 최적의 의미는 여러 가지로 해석할 수 있겠지만 여기서는 $\Pr(O|\lambda)$가 최대가 되도록 하는 상태 시퀀스로 간주한다.

(4) Scaling

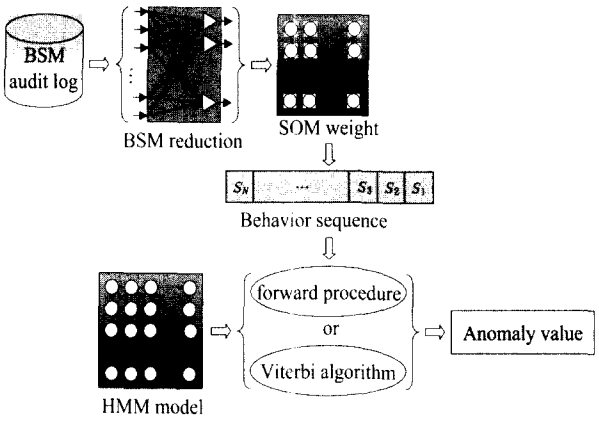
HMM 학습과정에서는 독립확률의 연속으로 표현하기 때문에 실수 곱셈이 많다. 따라서, 값이 계속해서 커지거나 작아지는 경우가 많이 발생된다. 이러한 값을 조정하기 위해서 scaling 과정을 거친다. Scaling은 주어진 상황에서 발생할 수 있는 경우 확률의 합은 1이라는 이론에 입각하여 확률값을 다시 계산하게 된다. 즉, 아래 수식처럼 N개의 배열이 각각 확률값을 가지고 있고 그 값의 합이 1이 되어야 한다면, N개의 배열의 합을 구하고 그 합으로 각 배열 요소를 나눈다. 아래는 $\alpha_t(i)$에서 $\hat{\alpha}_t(i)$를 구하는 수식이다. 이렇게 구해진 $\hat{\alpha}_t(i)$는 다시 $\alpha_t(i)$로 반영된다.

$$\hat{\alpha}_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{N}\alpha_t(i)}$$

이러한 scaling은 그 자체로 부가적인 연산을 수행하기 때문에 자주 하는 것은 바람직하지 못하다. 그러나, 변화된 α 값을 HMM 학습에 반영해야 하므로 하나의 입력 시퀀스에 대한 학습이 수행될 때마다 forward 변수 α, backward 변수 β, 초기값 π를 scaling한다.

3.4 판정 모듈

구축되어 있는 정상행위 모델을 근거로 사용자의 정상행위 여부를 판별하는 과정이다. 여기서의 입력은 전처리에 의해 걸러지고 축약된 사용자 행위 데이터이며 출력은 현재 사용자의 행위가 정상행위인지 비정상행위인지를 나타낸다. (그림 7)은 이러한 평가모듈의 동작 시나리오를 보여준다.



(그림 7) 비정상행위 판정모듈 동작 시나리오

비정상행위 판정에서는 이미 구축되어 있는 정상행위별 HMM에 사용자행위시퀀스를 입력으로 넣어 각 정상행위에서 현재 행위가 생성되었을 확률을 구한다. 확률을 구하는 방법으로는 forward backward procedure나 Viterbi 알고리즘을 사용할 수 있다. 각 모델별로 구해진 확률은 판정모듈에 전달되어 비정상행위인지 판정한다.

Forward procedure를 거쳐서 구해지는 α값으로 $\Pr(O|\lambda)$를 구할 수 있다. 여기에서 정상행위값 score는 아래와 같이 계산된다.

$$score = \max_{i=1}^{N}[\alpha_T(i)]$$

즉, $\alpha_T(i)$ 값에서 가장 높은 확률을 선택하는 것이다.

Viterbi 알고리즘은 HMM 모델에서 주어진 관찰 시퀀스에 대한 최적의 경로를 찾는 방법이다. 따라서, 주어진 관찰 시퀀스에 대한 최대 가능한 확률값을 구할 수 있다. forward procedure와 유사하나 경로를 추적하기 위해 ψ변수를 사용하는 것만 차이가 난다.

탐지 결과는 forward procedure와 약간의 차이가 나타났다. 그것은 HMM 모델에 대한 학습이 forward backward procedure에 의해 수행되었기 때문에, forward procedure의 결과가 더 정확하였다.

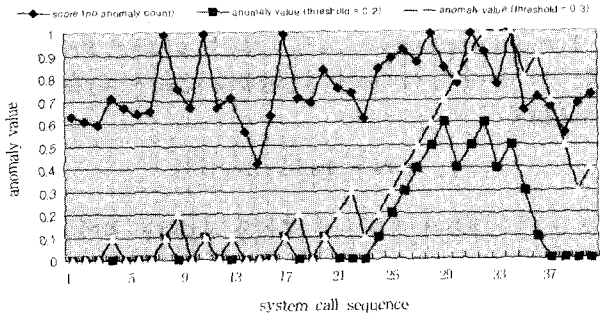
4. 임계값 비교 분석

4.1 기본 상수

먼저 SOM에서 사용되는 상수를 들 수 있다. 본 논문에서는 [17]에서 발표했던 것과 마찬가지로 $5 \times 5 \times 3$(시스템 호출 정보), $6 \times 6 \times 6$(파일 접근 정보), $5 \times 5 \times 8$(시스템 호출과 파일 접근 정보)의 크기를 갖는3가지 SOM을 사용하였다. 그리고 η 값은 0.2를 사용하였다. SOM에서 사용되는 상수는 HMM 구조의 크기를 결정하는 요소이다. $5 \times 5 \times 3$의 경우 $5 \times 5 = 25$이므로 입력 상태의 크기가 25가 된다. 그리고, 사용되는 A 집합과 B 집합의 크기가 이 값에 비례한다.

HMM에서 사용되는 상수는 HMM 구조의 크기를 나타내는 HMM 상태 수와 비정상행위도를 판정하는 시퀀스의 길이가 있다. 이 값은 [17]에서 이미 실험을 거쳐 가장 적절한 값을 설정하였다. HMM 상태 수는 10이고 시퀀스 길이는 30으로 하였는데, 이 값은 적용하는 환경에 따라 달라질 수 있다.

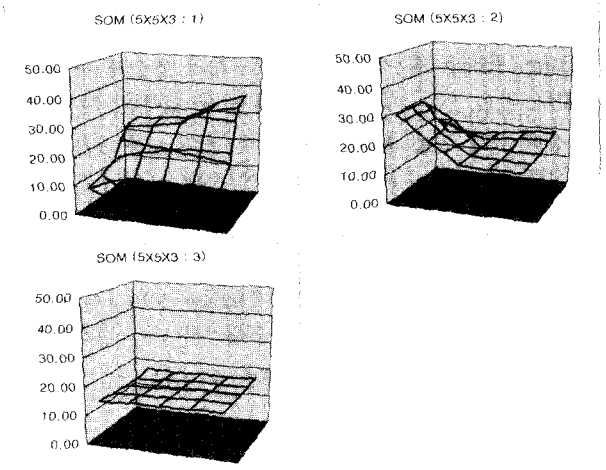
본 논문에서 HMM 판정모듈에서 나타나는 결과를 분석해 본 결과 예상치 않는 값에 따라 결과값이 진동하는 경우가 많음을 알 수 있었다. 이에 탐지센서의 민감도(sensibility)라는 개념을 적용하고 결과값이 요동치지 않도록 비정상행위 수를 정의하고 10으로 설정하였다. 물론 이 값은 실험에 의한 것이다. (그림 8)은 비정상행위 수를 적용하지 않은 순수 score 값과 임계값(threshold) 0.2에서의 비정상행위도, 임계값 0.3에서의 비정상행위도를 보여주고 있다. 여기서 score 값은 시스템 호출 시퀀스가 입력될 때마다 급격하게 변함을 알 수 있다. 반면에 비정상행위 수를 적용한 방법에서는 점진적인 진행이 됨을 알 수 있다. 임계값이 높으면 비정상행위 판정값이 쉽게 높아지고 낮으면 쉽게 높아지지 않는다. 이렇게 민감도를 줄여서 false-positive를 줄일 수 있었다. 이러한 임계값의 결정은 탐지센서가 설치되는 환경과 HMM 학습이 정확히 이루어졌는가에 따라 달라질 수 있다.



(그림 8) 비정상행위도 측정 결과(예)

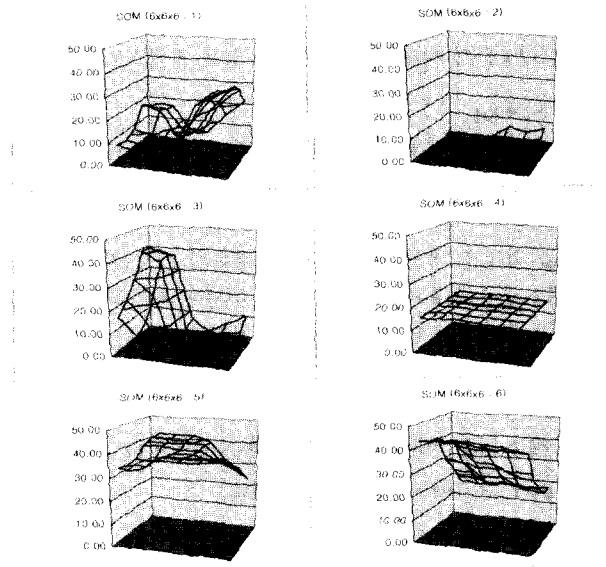
4.2 SOM 가중값(weight values)

SOM weight는 시스템 호출, 파일 시스템, 시스템 호출과 파일 시스템에 대하여 각각 존재한다. 먼저 시스템 호출 관련 SOM weight 값은 (그림 9)와 같다. (그림 9)에서 layer는 여러 가지 measure를 나타낸다. 시스템 호출 관련 SOM에서 layer 1은 시스템 호출 번호, layer 2는 시스템 호출 반환값, layer 3은 시스템 호출 수행 상태에 대한 값을 의미한다. 전반적으로 시스템 호출 번호에 가장 큰 영향을 받음을 알 수 있다.



(그림 9) SOM weight 값 (5x5x3)

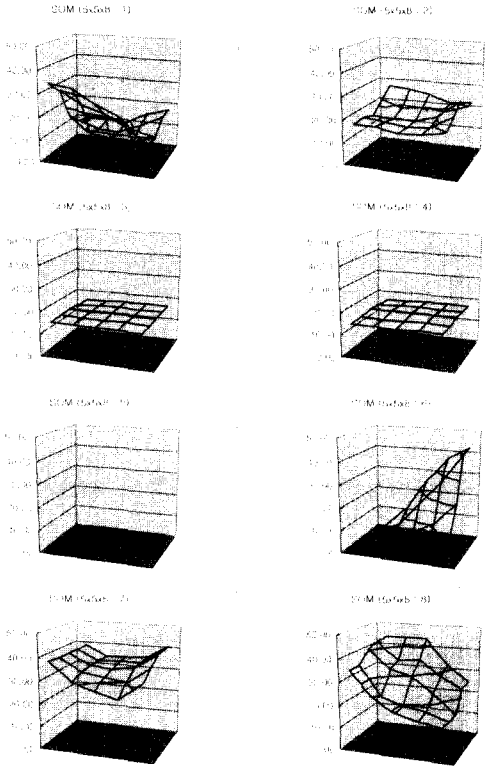
파일 시스템 관련 SOM weight 값은 (그림 10)에서 보여 주고 있다. Layer 1은 시스템 호출 번호, layer 2는 중요 파일 이름, layer 3는 중요 디렉토리 이름, layer 4는 매개변수 길이, layer 5는 파일 시스템 번호, layer 6는 파일 허가권(permission)을 나타낸다. (그림 10)에서 알 수 있듯이 파일 시스템 관련에서 중요 파일 이름이나 매개변수 길이는 변화가 적게 나타나고 있다. 이것은 실제로 그 디렉토리나 파일에 접근하는 경우가 적어서 학습 데이터에 충분히 반영되지 않기 때문이다. 따라서, 파일 시스템 관련 SOM의 값을 결정하는 중요 요소는 시스템 호출 번호, 중요 디렉토리, 그리고 파일의 허가권(permission)임을 알 수 있다.



(그림 10) SOM weight 값 (6x6x6)

시스템 호출과 파일 시스템을 종합한 SOM weight 값은 (그림 11)에서 5 × 5의 3차원 표면 그래프로 보여주고 있다. 여기서 Layer 1은 시스템 호출 번호, Layer 2는 시스템 호출 반환값, Layer 3은 시스템 호출 수행 상태, Layer 4는 매개변수 길이, Layer 5는 중요 파일 이름, Layer 6는 중요 디렉토리 이름, Layer 7은 파일 허가권, Layer 8은 파일 시스템 번호를 나타낸다. 여기에서 알 수 있는 것은 시스템 호출과 파일 시스템을 종합한 판정에서는 시스템 호출 번호, 중요 디렉토리 이름, 그리고 파일 시스템 번호가 중요한 요소가 됨을 알 수 있다.

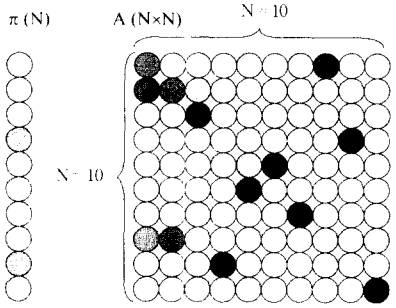
따라서, 학습 데이터에 대한 엔트로피가 높은 것을 종합하면, 시스템 호출 번호, 중요 디렉토리 이름, 파일시스템 번호, 파일 허가권, 시스템 호출 반환값을 들 수 있다.



(그림 11) SOM weight 값 (5x5x8)

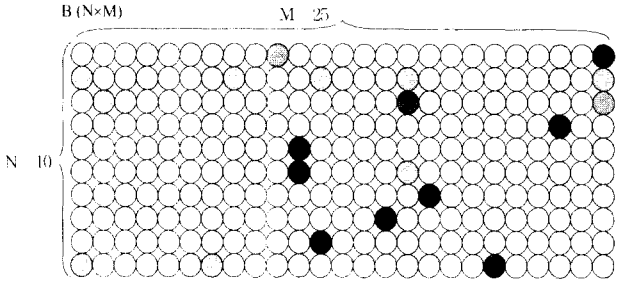
4.3 HMM model 데이터

(그림 12)는 본 논문에서 root에 대한 HMM 학습을 수행한 결과 생성된 모델값을 나타낸다. (그림 12)에서 색깔이 진할수록 확률이 높다. 초기값 π의 경우 4번째와 9번째에서 상태가 주로 시작됨을 알 수 있다. 초기 상태가 9에서 시작했다면 다음 상태로 진행될 확률은 a_{94}가 가장 높으므로 상태 4로 보통 전이된다. 상태전이가 확률이 높은쪽으로 이동된다면 정상행위에 가까워지는 것이고, 확률이 낮은쪽으로 이동된다면 비정상행위에 가까워진다. 실제 실행과정에서는 오랜 관찰에서 얻은 임계값(threshold)을 설정하여 그보다 작은 확률이 나타나면 비정상행위 수를 증가시킨다. 이러한 비정상행위 수가 일정 횟수 이상 증가하게 되면 warning 메시지를 보내게 되고, 그보다 훨씬 많은 횟수로 증가하면 alert 메시지를 보내게 된다. 이러한 비정상행위 수는 HMM을 이용한 비정상행위 탐지 모델의 민감도를 결정하는 요소가 된다.



(그림 12) HMM model 예 (root에 대한 π와 A)

(그림 13)에서는 본 논문에서 root에 대한 HMM 학습을 수행한 결과 생성된 모델값의 B를 보여주고 있다. (그림 13)에서도 역시 색깔이 진할수록 그 방향으로 진행할 확률이 높다는 것을 의미한다. 여기의 B 행렬은 상태수 10과 객체수 25에 대한 내용이며, 어떤 상태에서 객체가 나타날 확률을 의미한다. 예를 들어 상태 1에서 가장 많이 나타날 객체는 객체25번째이다. 초기 π값에 인속되는 A 행렬과 B 행렬의 조합으로 입력 시퀀스에 대한 확률값을 계산하게 된다. 이러한 확률값이 높게 나타나는 것은 정상행위 모델과 비슷한 방향으로 진행한다는 것을 의미하게 된다.



(그림 13) HMM model 예(root에 대한 B행렬 값)

4.4 진화적인 특성

임계치는 시스템 상황이나 침입판정 상태에 따라 조정될 수 있는 값이다. HMM 방식에서 조정할 수 있는 임계치로는 threshold와 비정상행위 수가 해당된다. (그림 8)에서도 threshold와 비정상행위 수에 따라 anomaly 값이 다르게 나타남을 볼 수 있었다. 따라서, 이 값은 위험도에 따라 사용자별로 다르게 설정할 수 있다. 또한, 시스템 상황에 따라 적응하여 값을 변동시킬 수 있다. 특히, 학습 데이터 양, 학습정도, measure에 따라 엔트로피를 부여할 수 있으며 사용자에 따라 개별 학습으로 그 특성을 강화할 수 있다.

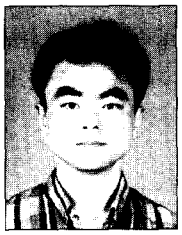
5. 결 론

본 논문에서는 SOM과 HMM을 이용하여 시스템 호출 수준에서 순서기반 비정상행위 탐지 알고리즘에 대하여 연구하였다. 순서기반의 비정상행위 탐지 센서의 경우 중요 파일 접근 및 시스템 호출 정보로 사용자들의 행위 시퀀스의 비정상행위를 판별한다. 실험 결과 시스템 호출 정보 중에서 학습에 대한 엔트로피가 높은 measure가 무엇인지 알 수 있었다.

본 논문에서는 SOM에 새로운 필터링 규칙과 축약 규칙을 적용하여 HMM의 입력 크기를 줄일 수 있었다. 이러한 축약은 HMM기반 비정상행위 탐지의 실시간 처리능력을 보장해 준다. 또한, 비정상행위 수라는 개념을 도입하여 HMM의 탐지결과에 대한 민감성을 둔화시켜서, 사용자가 탐지결과를 쉽게 이해하고 false-positive를 줄이는 효과가 있었다. 그리고, 능동적으로 threshold 값을 조정하여 시스템 상황에 따라 탐지센서가 적응할 수 있도록 하였다. 이러한 결과를 바탕으로 엔트로피가 높은 임계 변수와 measure에 대하여 탐지센서를 재구성하여 그에 대한 학습 효율을 높이고 실시간 탐지 능력을 높일 수 있다.

참 고 문 헌

[1] R. Büschkes, M. Borning, and D. Kesdogan, "Transaction-based Anomaly Detection," Proc. of the Workshop on Intrusion Detection and Network monitoring, USENIX, Apr., 1999.
[2] D. E. Denning, "An Intrusion-Detection Model," IEEE Trans. on Software Engineering, No.2, Feb., 1987.
[3] M. Esmaili, R. Safavi-Naini, and J. Pieprzyk, "Intrusion Detection : a Survey," ICCC '95, pp.409-414.
[4] A. K. Ghosh, A. S. Schwartzbard and M. Shatz, "Learning program behavior profiles for intrusion detection," Proc. USENIX Workshop on IDS and Network Monitoring, April 1999.
[5] T. Kohonen, "The Self Organizing Map," Proc. IEEE, Vol. 78, No.9, Sep., 1990, pp.1464-1480.
[6] S. Kumar, Classification and Detection of Computer Intrusions, Purdue University, Aug., 1995.
[7] T. Lane and C.E. Brodly, "An application of machine learning to anomaly detection," 20th NISSC, 1997.
[8] T. F. Lunt, "A Survery of Intrusion Detection Techniques," Computer & Security, Vol.12, No.4, Jun., 1993.
[9] A. Mounji and B. L. Charlier, "Continuous Assessment of a Unix Configuration : Integrating Intrusion Detection and Configuration Analysis," Symposium on Network and Distributed System Security, IEEE, Feb., 1997.
[10] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," IEEE ASSP Magazine, 1986.
[11] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. of the IEEE, Vol.77, No.2, 1989.
[12] J. Ryan, M. Lin, "Intrusion detection with neural networks," AAAI workshop, 1997.
[13] M. Sebring, E. Shellhouse, M. Hanna, and R. Whitehurst, "Expert Systems in Intrusion Detection : A Case Study," Proc. 11th National Computer Security Conference, Oct., 1988.
[14] C. Warrender, S. Forrest and B. Pearlmutter, "Detecting Intrusions Using System Calls : Alternative Data Models," The 1999 IEEE Symposium on Security and Privacy, 1999.
[15] Nong Ye, "A Markov Chain Model of Temporal Behavior for Anomaly Detection," Proc. of the 2000 IEEE Workshop on Information Assurance and Security, Jun., 2000.
[16] W. Lee and D. Xiang, "Information-Theoretic Measures for Anomaly Detection," The 2001 IEEE Symposium on Security and Privacy, 2001.
[17] J. Choy, S. Cho, "Hidden Markov Model for Sequence Recognition in Intrusion Dectection System," Proc. IEEE ICARDT '99, 1999.
[18] 김민수, 은유진, 노봉남, "UNIX 환경에서 퍼지 Petri net을 이용한 호스트 기반 침입 탐지 시스템 설계", 한국정보처리학회 논문지, 제6권 제7호, 1999.
[19] 최종호, 조성배, "비정상행위 탐지기반의 침입탐지시스템을 위한 은닉 마르코프 모델을 사용한 행위 모델링과 탐지", '99 정보보호 우수논문집, 1999.



김 용 민

e-mail : chonnam.ac.kr
1989년 전남대학교 전산통계학과 졸업(학사)
1991년 전남대학교 대학원 전산통계학과 (이학석사)
1996년~현재 전남대학교 대학원 전산통계학과 박사과정
관심분야 : 네트워크보안, 워크플로우, 통신망관리 등

김 민 수

e-mail : phoenix@athena.chonnam.ac.kr
1993년 전남대학교 전산통계학과 졸업(학사)
1995년 전남대학교 대학원 전산통계학과 (이학석사)
2000년 전남대학교 대학원 전산통계학과 (이학박사)
2000년~2001년 한국정보보호진흥원 선임연구원
2001년~현재 전남대학교 정보보호협동과정 객원교수
관심분야 : 시스템 보안, 네트워크 보안, 정보보안, 신경망 등

김 홍 근

e-mail : hgkim@kisa.or.kr
1985년 서울대학교 컴퓨터공학과 졸업
1987년 서울대학교 대학원 컴퓨터공학과 (공학석사)
1994년 서울대학교 대학원 컴퓨터공학과 (공학박사)
1994년~1996년 한국전산원 선임연구원
1996년~현재 한국정보보호진흥원 기술개발단장
관심분야 : 컴퓨터 보안, 병렬 알고리즘

노 봉 남

e-mail : bongnam@chonnam.ac.kr
1978년 전남대학교 수학교육과 졸업
1982년 KAIST 대학원 전산학과(이학석사)
1994년 전북대학교 대학원 전산통계학과 (이학박사)
1983년~현재 전남대학교 컴퓨터정보학부 교수
관심분야 : 객체지향시스템, 통신망관리, 정보보안, 시스템 및 네트워크 보안 등