소규모 네트워크의 통합보안관제를 위한 SnSA 설계 및 구현*

이 동 휘** · 신 영 준** · 김 귀 남***

요 약

지난 2003년 1월 25일 오후 국내 최상위 DNS(도메인 네임 시스템)서버가 다운되면서 유·무선 인터넷 서비스가 일제히 마비되는 사상 초유의 사태가 벌어지고 전국 곳곳에서 크고 작은 혼란과 피해를 빚었다. 원인은 여러 가지의 형태로 분석 되었지만 가장 큰 이유는 보안 불감증에서 기인했다고 할수있다. 특히 소규모 네트워크에서 보안관리자는 거의 없고 정보보호를 위한 물리적인 장치가 전무한 상태였다. 대부분의 보안시스템이 경제성을 이유로 대규모 네트워크에 중점적으로 개발되고 있으며 실시간으로 네트워크의 보안상태를 점검해 주는 보안관제센터들도 IDC나 대규모 네트워크를 중점으로 서비스 되는 실정이다.

본 논문에서는 제기되어온 문제점을 해결하기 위하여 소규모 네트워크 전용 SnSA를 설계하여 경제적인 보안관제 서비스를 연구하였다. 첫 번째로 소규모 네트워크에서 보다 많은 정보를 수집하기 위해서 네트워크 침입탐지기능 N_SnSA와 호스트 침입탐지기능 H_SnSA를 구현하였고 두 번째로 경기대학교 통합보안관제센터와 같은 공익성을 가진 센터에서 저렴한 비용으로 연동시켜 기존의 취약성을 해결하는 방법을 연구하였다.

SnSA Design and Embodiment for ESM of Small Scale Network*

Dong Hwi Lee** · Young Jun Shin** · Kuinam J. Kim***

ABSTRACT

At the end of last January, 2003, a domestic top-level domain name server (DNS) shut down the server and it caused the wired and wireless internet services to be completely paralyzed in the aftermath of a virus attack incurring a various range of losses nationwide. The main reason of this event is the lack of our awareness of cyber security. In particular, in the small-scale network, there are few security administrators and no operating devices to protect information as well. Under this circumstance, using ESM center to service real-time security supervision and correspondence for network, it can be one option. However, due to the economic efficiency, most of security systems have been being developed focusing on the large-scale network first. Therefore, ESM centers which inspect security state of network concentrate on IDC or large-scale network services.

This dissertation studies economical ESM service by designing exclusive SnSA for small-scale network for widespread use. Firstly, network invasion feeler function N_SnSA and host invasion feeler function H_SnSA are embodied to collect more informations in the small-scale network. Secondarily, the existing vulnerability is studied to find the solutions linked with a low cost to a public center such as Kyonggi Univ ESM center.

* 본 연구는 2002년도 산학협동재단 학술연구비 지원을 받아 이루어졌음.

** 경기대학교 통합보안관제센터 씨티알씨

*** 경기대학교 정보보호기술공학과

1. 서 론

초고속 통신망 시설의 발달과 네트워크분야의 진보 그리고 PC 보급률의 증가에 따라 인터넷은 누구나 쉽게 접할 수 있는 생활의 일부분이 되었다. 기업의 컴퓨팅 환경도 인터넷으로 급격히 분산 재편되며 보안은 이제 선택이 아닌 필수 사항으로 자리 잡았다. 국내 네트워크 환경의 급격한 성장에 따라 발생하는 문제가 바로 네트워크 보안이다. 호스트가 네트워크에 연결되어 있고 그 네트워크가 인터넷에 연결되어 있다면 호스트는 언제든지 외부의 침입을 받을 수 있다. 그래서 이러한 발전적인 기술의 이면에는 보안이라는 또 하나의 극복해야만 하는 과제를 남기게 되었다.

해킹기술의 발달과 더불어 새로운 해킹기술을 이용한 네트워크 시스템 환경에 대한 침입시도가 늘어나고 있으며, 이에 대한 방지책으로 시스템 환경을 보호하고 정보를 보호하기 위한 다양한 방법이 제안되고 있다. 하지만 대규모 네트워크 환경의 주도로 소규모 네트워크에서는 고가의 침입차단 시스템과 침입탐지 시스템을 구비하여 대처하기에는 어려운 현실이다.

시스템관리자 혹은 보안관리자의 의식결여 문제까지 겹쳐 사이버 상에서 다양하고 방대한 분량의 프로토콜들의 문제점과 함께 그대로 침입자에게 노출 되었고, 침입자에 대해 거의 무방비적인 상태가 대부분이다. 이런 취약한 약점들은 곧 정보자원을 외부로부터 침해를 발생시킬 수 있는 어떤 모든 것이 요소가 되어, 정보자원에 피해를 입는 결과를 가져오게 된다.

지난 예로 2003년 1월 25일 국내 최상위 DNS (도메인 네임 시스템)서버가 다운되면서 유. 무선 인터넷 서비스가 일제히 마비되는 사상 초유의 사태가 벌어지면서 전국 곳곳에서 크고 작은 혼란과 피해를 빚었다. 이는 세계 최첨단을 자랑하던 인터넷 강국이 암흑시대로 되돌아가는 데는 반나절이면 충분했으며 전국의 주요 유무선 인터넷 서비스가 수 시간 동안 마비되면서 전국이 일대혼란에 빠졌다.

위의 예에서 보듯이 네트워크 상에서 소규모 네트워크 혹은 호스트 단위 작은 규모에서의 보안의식 결여가 큰 혼란으로 이루어지는 것을 경험 하였다. 소규모 네트워크에서 각종 정보보호 장비의 설치는 경제적인 문제 때문에 어려움을 겪는다.

이러한 문제를 해결하기 위하여 통합보안관제 시스템(ESM : Enterprise Security Management)이 대두되었지만 대형 네트워크 또는 대형 IDC (Internet Data Center)에만 상용화되어 있다.

본 논문에서는 현실에 맞는 그 중요성을 인식하고 보호하기 위하여 통합보안관제 시스템과 소규모 네트워크 연동을 위한 경제적이며 효율적인 SnSA(Small Network System Agent)를 구현하여 시스템이 내·외부의 침입을 당하는 것으로부터 보호해야 한다.

2. SnSA 개요

논문에서 구현하려는 SnSA는 통합보안관제 시스템이 실질적으로 소규모 네트워크에 적용되는 실용적이 모델 가운데 하나다. SnSA 저변에 깔린 개념은 소규모 네트워크는 보안에 취약점과 해킹에 항상 노출 되어 있으나, 소규모 네트워크에서 고비용의 보안 시스템을 적용하기 어려운 현실을 볼 수 있다. 하지만 SnSA의 구현으로 통합보안관제 시스템과 연동하며 보안의 취약점을 해결 하고자 한다.

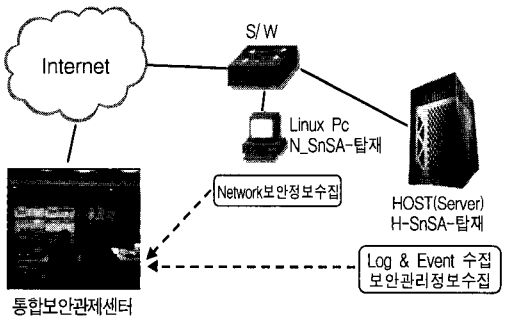
SnSA는 소규모네트워크에서 Linux 기반의 보안 에이전트로서 통합보안관제 시스템과 통신을 목적으로 Network Based IDS와 Host Based IDS 및 다양한 기능으로 취약점을 제거하며 보안관리에 간편하게 사용되어야 한다. SnSA는 소규모 네트워크에서 통합보안관제 시스템으로 서

버, 네트워크, 기타 장비들로부터 발생하는 보안 관련 로그를 수집하여 중앙에서 분석, 침입시도 및 보안사고 징후를 탐지 실시간 대응을 할 수 있는 환경을 제공하는 것이다. 또한 보안사고 예방, 보안사고 탐지, 보안사고 대응, 보안사고 통계 등 보안관리에 대한 모든 필요한 기능을 포함, 소규모 환경에 맞춰 사용가능한 기능이 제공되는 시스템이다.

3. SnSA 설계

네트워크로 오는 모든 트래픽을 의심스럽게 봐야 한다는 개념에서 SnSA시스템은 작용한다. SnSA는 단독서버 또는 50대 미만의 호스트를 가진 네트워크에서 운용되는 것으로 기본으로 한다.

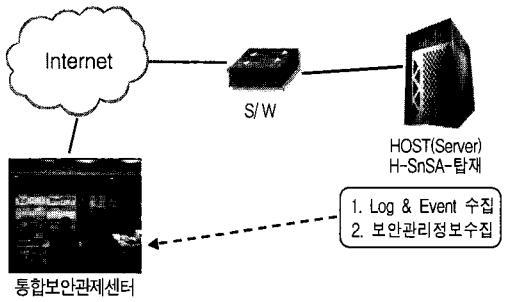
단독서버일 경우 H-SnSA(Host Based IDS 기능 SnSA)으로 통합보안관제 시스템으로 로그 및 이벤트를 전송하여 실시간 대응 환경을 제공한다.



(그림 3-1) H-SnSA 구성도

소규모 네트워크 그룹인 경우 N-SnSA(Network Based IDS 기능 SnSA)를 허브와 미러링하여 네트워크의 모든 트래픽을 감시하게 되며 각 호스트에는 H-SnSA의 설치로 로그 및 이벤트를 전송하여 실시간 대응 환경을 제공한다.

H-SnSA는 호스트를 감시하고 침입자가 시스템을 사용하는 것을 최대한 억제하고 통제하는 역할을 한다. 호스트의 위치에 따라, 공격이 외부에서 오든지 내부에서 오든지 감시할 수 있다. H-SnSA의 기능을 수행하기 위해서는 판단의 근거가 되는 감사 자료나 로그 파일이 존재하여야하며, 시스템이 원활하게 그 기능을 수행하고 있어야만 한다. root 권한을 획득한 해커가 침투기록인 로그파일을 삭제, 변조하거나 침입탐지 시스템 혹은 OS의 실행관련 파일의 변조를 통해 시스템의 이상 작동을 유발한다면 침입탐지 시스템으로서의 기능을 수행할 수 없기 때문이다 [20].



(그림 3-2) N_SnSA와 H-SnSA 혼합구성도

Linux 계열 OS의 경우 Kernel 패치가 빠르게 이루어지며, 이에 따라 본 논문에서 구현하는 SnSA에서는 Kernel 보호차원에서 특정 파일이나 중요 디렉토리 등에 대해 허용되지 않는 사용자의 접근을 통제하는 기능을 수행한다[12].

- OS관련 시스템 파일(Kernel file 포함)의 경우는 root 권한을 가진 관리자도 임의로 수정할 수 없도록 보호한다.
- 침입탐지 시스템과 OS 관련 프로세스의 경우 임의로 종료시킬 수 없으며, 파일 시스템에서 보이지 않도록 구현한다.
- 관리자 시스템에서 Web Browsing을 통해 조작할 수 있도록 구현한다.
- 로그파일과 라우터 정보 등 침입탐지 여부를 판단할 수 있는 근거가 되는 파일 시스

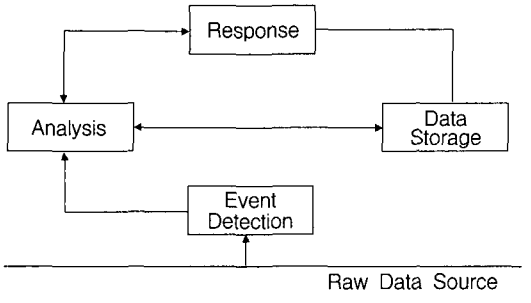
템에 대한 삭제, 변조를 임의로 할 수 없도록 보호한다.

4. SnSA 구현

본 논문에서는 소규모 네트워크에서 실질적이며, 경제적으로 정보보호 방안을 수립할 수 있도록 침입탐지 기능과 호스트 보호를 통해 통합보안관제 시스템과 실시간 통신을 통해 네트워크의 보안을 강화하는 데 중점을 두었다.

4.1. 침입탐지 기능

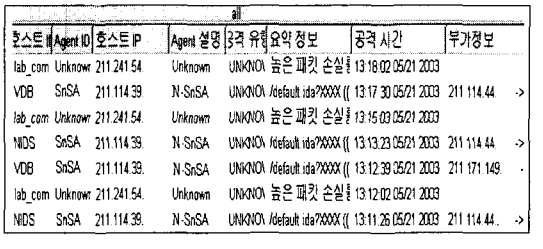
TCP/IP상의 각종 Application 프로토콜별 트래픽을 24시간 실시간으로 모니터링 할 수 있는 기본 기능은 아래 그림과 같이, 원시 데이터(raw data source), 사건 탐지(event detection), 분석(analysis), 대응(response), 데이터 저장소(data storage)의 5 가지 요소로 구성된다[6].



(그림 4-1) 침입탐지 기본모델

'원시데이터'는 여러 곳의 시스템 자원으로부터 얻어진 감사 자료들과, 시스템 자원의 사용내용(CPU 사용도, 메모리 사용도, 시스템 자원의 고갈도 등), 네트워크 관련 정보 등을 말하며, '사건탐지'는 실제 사건을 탐지하는 방안으로 여기서의 사건이란 특정 데이터, 환경, 행동 등의 발생 상황을 말한다[3]. 이 사건들은 간단한 사건들과 복잡한 사건의 두 부분으로 나눌 수 있다. 사건들을 분석하여 실제 침입이 발생할 확률을 결정하는 것이 '분석' 기능이다. 분석시 이용하는 정보들은 '사건탐지' 결과와 이전의 분석으로부터 얻어진 결과들, 각 사용자들의 행동양식에 대한 정보, 각 개체 및 시스템의 수행 양식 정보, 기타 위험하다고 알려진 사이트 및 개인에 대한 정보 등이다. '대응'은 침입이 발생했다고 판단된 경우 이를 관리자에게 알려주는 방안에 대한 것으로, 결과는 이벤트 전송프로토콜에 의해 통합보안관제 시스템에 나타나며, 기타 이메일, 페이저, 메신저 서비스 등이 이용될 수 있다. '데이터 저장소'에는 탐지된 사건 결과, 분석에 필요한 모든 데이터들, 알려진 침입에 대한 프로파일들, 세부적인 원시 데이터들(추후 추적 등을 위하여 사용될 수 있다)이 저장된다. 데이터 저장소는 저장된 데이터를 보호할 수 있는 정책 하에 관리되어야 한다.

SnSA는 시스템의 부팅과 동시에 기본적으로 침입탐지 기능과 파일 시스템의 보호가 이루어진다. SnSA가 활성화를 통하여 패턴매칭을 통한 IDS의 침입탐지기능을 수행하며 이벤트를 저장하는 기능을 통해 관리하고 있는 네트워크 환경에 대한 상황을 파악할 수 있다. 침입탐지 시스템을 통해 현재 네트워크를 사용 중인 사용자의 프로파일과 날짜, 출발지 IP 주소들을 Database로 구축되어 있는 패턴과 비교하여 침입여부를 감시하게 된다[23].



(그림 4-2) 통합보안관제센터 Event Manager 침입탐지

N-SnSA를 실행시키기 위해서는 우선 설치된

컴퓨터와 root 권한이 필요하다. 이것이 준비되었다면 다음과 같은 명령어로 N-SnSA를 실행시킬 수 있다.

```
N-SnSA# N-SnSA -A full - S HOME_NET = xxx.xxx.xx.xx/16 -S EXTERNAL_NET = any -h xxx.xxx.xx.xx/16 -c/etc/snsa/snsa.conf -l /var/log/snsa -i eth0 -D.
```

(그림 4-3) SnSA 설정

N-SnSA 실행의 기본적인 형식은 snsa(options) expression 이다.

-A full은 알람모드를 full로 설정(경고 메시지와 함께 패킷 헤더를 기록한다.)한다. -S 옵션은 변수를 지정할 수 있는데 HOME_NET과 EXTERNAL_NET의 변수를 지정해 준 것은 snsa의 rule 파일에 $HOME_NET과 $EXTERNAL_NET에 대입되게 된다. 주소를 쓰는 형식은 IP-adress/CIDRblock이다. 32bit의 IP 어드레스 중 CIDR 비트만큼이 네트워크 주소에 해당하고 그 나머지 부분이 호스트 주소에 해당되게 된다. 예를 들어 192.168.0.1/24 이렇게 썼다면 192.168.0.*의 주소를 의미하게 된다. CIDR 블록이 8이라면 A class의 network를, 16이라면 B class 크기의 network를, 24라면 C class의 network를, 32라면 호스트하나만을 가리키게 된다. -D 데몬 모드로 시작해서 -h로 체크할 네트워크의 범위를 결정했다. -c로 /etc/snsa/snsa.conf의 설정내용을 읽어 들였고, /var/log/snsa 디렉토리에 eth0에서 들어오는 공격을 탐지해 full alert모드로 로그를 남긴다는 뜻이다.

N_SnSA의 룰은 하나의 룰이 반드시 한 줄에 끝나야 한다. 아래는 룰의 한 예다.

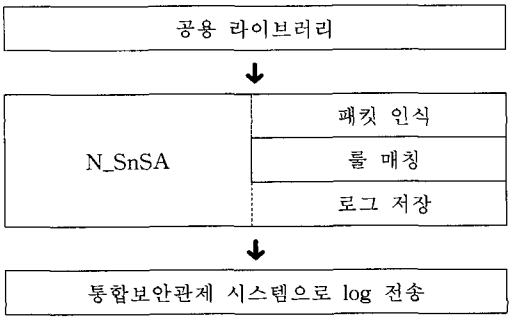
```
alert tcp any any → 192.168.X.X/24 111 (content : "|00 01 86 a5|" ; msg : "mountd access" ; )
        ←——— rule header ———→
        ←——— rule option ———→
```

(그림 4-4) Rule 설정

룰은 위와 같이 헤더부분과 옵션 부분으로 나누어서 생각할 수 있다.

룰 헤더에는 어떤 패킷인가, 어디서 온 것인가에 관한 정보가 있고, 그에 따른 event도 이곳에 정의한다. 룰 헤더 첫머리에 오는 것이 rule action인데, 조건에 맞는 패킷을 발견했을 때 무엇을 해야 할 것인가를 정해준다. N_SnSA가 할 수 있는 action은 기본적으로 5가지가 있다[32].

- alert : 경고 후 패킷을 기록한다.
- log : 패킷을 기록한다.
- pass : 패킷을 무시한다.
- activate : 경고 후 다른 dynamic 룰을 진행시킨다.
- dynamic : activate룰에 의해 active될 때까지 기다리다가, active되면 임무를 수행한다. 새로운 action을 output plugin과 연계하여 만들 수도 있다.



(그림 4-5) N_SnSA 흐름도

4.2 H-SnSA의 기능

소규모 네트워크에서 호스트가 해커에 의해 root의 권한을 도용당하거나 허가받지 않은 사용자에 의한 접근통제 기능이 상실하였을 경우 시스템의 기능을 상실할 위험이 있으며, 권한의 남용을 통해 파일 시스템의 조작, 변조의 위험이 있다[7]. 또한 침입 대응에 필요한 로그 파일 등 관련 로그 파일이 삭제, 조작, 변조 되었을 경우

침입에 대해 대응을 할 근거 자료를 갖지 못하게 된다. 이를 방지하기 위해선 시스템 운영에 중요한 파일 시스템과 디렉토리를 보호할 필요가 있다. 이러한 파일의 변경을 방지하기 위해 암호화 메커니즘이 사용되지만, 시스템 파일 자체가 자주 업데이트 되고 변경되므로 암호화 기능으로는 충분하지가 않다. 뿐만 아니라 암호화 매커니즘은 파일을 지우는 행위로부터 시스템을 방어하지 못한다. 또한 공격자가 프로세스 중에서 하나 혹은 그이상의 프로세스들을 멈추게 한다면 시스템은 그 자체로도 탐지기능을 충분히 할 수가 없다. 그러므로 이러한 행위로부터 침입탐지시스템을 보호할 필요가 있다. 슈퍼유저의 권한은 시스템의 작동을 멈추게 할 수도 있고, 침입탐지 프로세스를 추가할 권한도 가지고 있다. H-SnSA에서 보호하는 파일 시스템과 디렉토리는 다음과 같다[12].

〈표 4-1〉 중요/home 디렉토리의 서브디렉토리

디렉토리	중 요 사 항
/home/ftp	proftpd, ncftpd와 같은 FTP 서비스를 하는 프로그램에서 사용
/home/httpd	Apache와 같은 웹 서비스가 사용하는 디렉토리로 /home/httpd/html 디렉토리 아래에는 html 문서를 포함하여, /home/httpd/cgi-bin/ 아래에는 CGI 프로그램을 설치하여 웹 서비스를 할 수 있는 관련 파일이 존재
/home/samba	Samba와 같은 파일 공유서비스를 하는 소프트웨어가 사용
/home/(user account)	Linux User의 홈 디렉토리

〈표 4-2〉 /usr 디렉토리의 서브디렉토리

디렉토리	중 요 사 항
/usr/bin	Linux에서 일반적으로 사용되는 유틸리티 프로그램이 포함되어 있으며, Linux의 핵심 부분이 아닌 많은 Linux Application에 대한 실행파일을 수록
/usr/X11R6	XFree86(X Window system) 소프트웨어를 수록
/usr/etc	Linux Application에 대한 설정 파일이 위치한다.
/usr/doc	Linux Application에 대한 Help file이 위치한다.
/usr/include	C와 C++에 대한 library를 수록하고 있다.
/usr/local	Local file을 수록하고 있다.
/usr/man	man command를 사용해 읽을 수 있는 매뉴얼 페이지를 수록하고 있다.
/usr/sbin	e-mail과 networking에 관한 command 같은 관리적인 command를 수록하고 있다.
/usr/src	Linux kernel이나 기타 source code를 수록하고 있다.

하드 디스크에 있는 중요한 파일을 보호할 수 있다. 이때 파일 시스템의 타입은 문제가 되지 않는다. 루트를 포함한 누구든지 지정된 파일을 변경할 수 없게 된다. 이 기능은 중요한 프로세스에 대해 kill 되어지는 것을 방지할 수 있다. 파일보호기능은 인증되지 않은 프로그램으로부터의 RAW IO operation을 보호할 수 있다.

하드디스크를 포함하여 MBR까지 보호할 수 있다. 첫째로, 보호하려고 하는 파일을 정해야 한다. 대부분의 경우에서 /usr, /sbin, /etc, /var/log/와 같은 System binary files이나 system 설정 파일을 보호하려고 할 것이다.

둘째로, 그 파일을 보호하는 방법을 결정해야 한다. 여기서는 3가지 타입의 보호방법을 제공한다.

- Read Only Files. 읽기 전용의 파일로 아무도 그 파일을 변경할 수 없게 된다.

/etc/passwd 같은 파일이 알맞은 용도가 될 것이다.

사용법 :

snsadm -A -r filename_to_protect

예제 :

1. /sbin/ 디렉토리를 읽기 전용으로 보호

/sbinsnsadm -A -r /sbin/

2. /etc/passwd 파일을 읽기전용으로 보호
   # /sbin/snsadm -A -r /etc/passwd

〈표 4-3〉 Linux 시스템의 중요 디렉토리

디렉토리	중 요 사 항
/	파일 시스템의 기초를 이루며 모든 파일과 디렉토리는 위치에 상관없이 루트 디렉토리에 수록
/bin	Linux의 실행프로그램을 수록한다. cat, 체, is, more, tar 등과 같은 리눅스 명령어를 포함
/dev	디바이스 파일을 수록한다. Linux는 디바이스를 특수한 형태의 파일로 다룬다.
/home	모든 사용자의 홈 디렉토리이다. 예를 들어 유저 abc의 홈 디렉토리는 /home/abc의 형태로 된다.
/etc	시스템의 구성파일과 초기화 스크립트(init Script)를 수록
/lost +found	잃어버린 파일에 대한 디렉토리로서 모든 디스크 파티션에는 /lost_found 디렉토리가 존재한다.
/lib	C Language에 대한 Library 파일과 다른 프로그램 언어를 포함
/mnt	플로피 디스크와 디스크 파티션처럼 임시로 장치를 마운트하는데 사용하는 디렉토리
/proc	Linux 시스템의 다양한 정보를 수록하는 특수한 디렉토리이다. 실제로 존재하는 디렉토리는 아니며, 시스템이 관리를 목적으로 메모리 상에 만들어 놓은 가상의 디렉토리이다. 디렉토리 안의 파일은 현재의 시스템 설정을 보여준다.
/root	루트 사용자에 대한 홈디렉토리
/sbin	시스템 관리 작업에서 보통 사용되는 명령을 나타내는 실행파일을 수록한다. mount, halt, umount, shutdown과 같은 명령이 존재
/usr	X-Window 시스템과 매뉴얼 페이지와 같은 중요 프로그램의 하위 디렉토리를 수록
/tmp	유저들에게 사용가능한 임시 디렉토리
/var	시스템 정의 파일과 임시정보를 저장하는 디렉토리를 포함

- Append Only Files. /var/log/message 나 /var/log/secure와 같은 시스템 로그 파일에 첨가전용 파일타입이 사용된다. 이 파일들은 오직 append mode에서만 열릴 수 있으며, 위와 마찬가지로 변경할 수 없다.

사용법:

snsadm -A -a filename_to_protect

예제:

1. 시스템 로그 파일들을 보호
   # /sbin/snsadm -A -a /var/log/message
   # /sbin/snsadm -A -a /var/log/secure
2. apache httpd 로그파일들을 보호
   # /sbin/snsadm -A -a /etc/httpd/logs/
   # /sbin/snsadm -A -a /var/log/httpd/

아래는 일반적인 보호의 예이다.

```
snsaadm -Z
snsadm -A -r /boot
snsadm -A -r /vmlinuz
snsadm -A -r /lib
snsadm -A -r /root
snsadm -A -r /etc
snsadm -A -r /sbin
snsadm -A -r /usr/sbin
snsadm -A -r /bin
snsadm -A -r /usr/bin
snsadm -A -r /usr/lib
snsadm -A -a /var/log
```

만약 /etc/snsa.conf 파일을 읽기 전용으로 보호한다면 snsa가 적용되지 않은 커널로 부팅 할지라도 어떤 파일의 형식도 변경할 수가 없게 된다.

'snsa.conf' 파일의 위치를 고치기 위해서는 'snsadm.c' 파일의 아래 라인을 고친다.

```
# DEFINE snsa_CONF "/etc/snsa.conf"
```

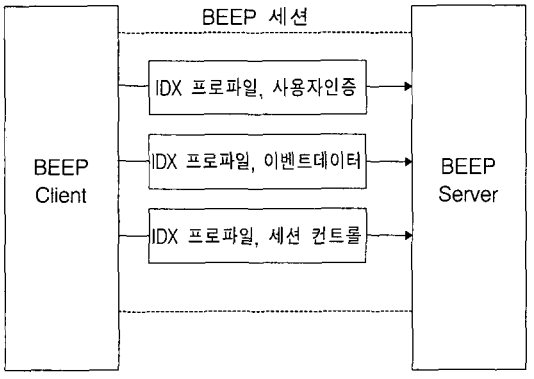
그리고 컴파일을 다시 하면 된다.

4.3 보안관제 시스템과의 통신

소규모 네트워크에서 통합보안관제 시스템과 이벤트 및 로그 통신을 위하여 BEEP 프로토콜의 기본구성과 보안 이벤트 통합을 위한 Linux 기반의 프로토콜 프로파일들을 기술한다. 표준 이벤트 통합을 위한 이벤트 전송시 이벤트를 보내는 측은 분산 설치된 SnSA가 되며, 받는 측은 통합보안관제 시스템의 이벤트 게이트웨이(Event Gateway)가 된다.

4.3.1 BEEP 프로토콜

BEEP 프로토콜은 TCP/IP상에서 안전한 이벤트의 전송을 위해 application계층 기반으로 고안되었으며, 통신 Peer의 상호 인증과 전송시 보안을 제공하는 기능을 포함하고 있다[6]. BEEP Peer간에 하나의 세션을 유지하며, 하나의 세션은 여러 개의 채널로 구성된다. 각 채널은 BEEP Peer간의 전송 중 인증, 데이터 전송, 보안 설정 등을 담당하는 고유한 역할의 기능들과 연결된다. 또 각 채널은 전송 메시지의 고유한 구문(syntax)과 의미(semantics)를 정의한 프로토콜 파일을 갖는다. 전송메시지 형태는 text의 MINE(Multipurpose Internet Mail Extensions) 컨텐트를 채택하였다[8].



(그림 4-6) BEEP 세션과 멀티채널

(그림 4-6)에서는 멀티채널로 구성된 하나의 BEEP 세션을 보인다. 각 채널은 프로파일과 고유의 역할을 가진다. 특별히 세션 자체의 컨트롤은 0번 채널이 수행한다.

4.3.2 이벤트 전송

IDX 프로파일은 보안 툴과 통합 보안 시스템 간의 이벤트 전송방법을 제공한다. 이벤트 전송을 위해 프로파일은 syntax와 semantics를 사용하여 IDX-Greeting, Option, IDMEF-Message의 세 가지 기본요소를 갖는다. IDXP-Greeting 요소는 BEEP Peer간의 상호 확인을 위해 사용된다. 이는 BEEP 채널상에서 BEEP Client와 BEEP Server의 판별이 가능해야 하며, Peer의 URI(Unified Resource Identifier) 정보를 포함해야 한다. 또는 Peer의 IP 정보를 가질 수도 있다. 또 하나 이상의 Option 하위 요소도 포함 할 수 있다[13]. 다음은 성공적인 채널 설정을 위한 IDXP-Greeting의 예이다. 먼저 BEEP Client가 Greeting을 전송하면, BEEP Server는 해당 메시지에 응답(acknowledge)하고, 이어 Greeting을 전송하며 이에 BEEP Client가 응답한다.

```
MSG 0 10 . 1592 187
Contnent: text
<start number = '1'>
  <profile
uri = 'http://ctrclab.com/beep/transient/idwg/idxp'>
  <![CDATA[
    <IDXP-Greeting
            uri = 'http://ctrclab.com/alice'
      role = 'client' />
    ]]>
   </profile>
  </start>
  END
```

(그림 4-7) Event 전송

4.3.3 syslog 전송

syslog 함수는 syslogd을 Program을 통해 control할 수 있는 C Library 함수로 syslog, openlog, closelog, setlogmask 등의 함수가 있다. 함수의 구성은 다음과 같다.

```
# include <syslog.h>

 void openlog(const char * ident, int logopt, int
facility) ;

 void syslog(int priority, const char * message,
…/* arguments */) ;

 void closelog(void) ;

 int setlogmask(int maskpri) ;
```

(그림 4-8) Syslog 함수구성

syslog 함수는 어떠한 실행되는 프로그램(즉, Daemon)에 대해 /etc/syslog.conf 파일에 정의한 파일이나 시스템 콘솔이나 현재 접속중인 User들이나 네트워크의 또 다른 호스트의 syslogd에게 특별히 정의한 메시지를 보내주는 기능을 한다. 즉, system에서 자연적으로 발생하는 메시지 이외에 이 함수를 이용해 메시지를 발생시킬 수 있다는 것이다. 기록되는 메시지들은 메시지 Head와 메시지 Body를 포함한다. 메시지 Head는 Facility, Level, Timestamp, Tag 문자열, 그리고 Process ID로 구성된다. 메시지 Body는 메시지와 함께 printf 함수와 흡사한 방법을 통해 생성되어진다. printf 함수와 다른 점이 있다면, printf 함수는 프로그래밍 시 errno.h 헤더파일을 이용해 errno변수에 의한 error 메시지 출력은 하는 기능이 있는 반면에 syslog 함수에는 그러한 기능이 없다[8].

syslog 함수는 syslog.h 헤더파일을 참조하는데, syslog.h 함수에는 *Facility*와 Level에 대해 다음과 같이 정의가 되어있다.

```
 if((sd = socket(PF_INET,SOCK_DGRAM, 0)) <
0) {
 printf("Client : Can't open stram socket.\n") ;
 exit(0) ;
    /* 서버의 소켓주소 구조체 server_addr을 '0'으
       로 초기화 */
 bzero((char *) & server_addr, sizeof(server_addr)) ;
  /* server_addr을 셋팅 */
    serv_addr.sin_family = AF_INET ;
    serv_addr.sin_addr.s_addr = inet_addr(argv[1]) ;
    serv_addr.sin_port = htons(atoi(argv[2])) ;
  /* 로그 파일에서 읽어 들여 버퍼에 넣음 */
  while(fgets(buf, sizeof(buf), fp) != NULL)
 {
  /* 메시지 전송 */
 if(sendto(sd, buf, strlen(buf), 0, (struct sockaddr*)
& serv_addr, sizeof(serv_addr)) < 0 )
 {
 printf("메시지 보내기에 실패했습니다. /n") ; #
include <fcntl.h>
 }
```

(그림 4-9) 메세지 전송

4.4 기타 기능

SnSA구현으로 네트워크 보안을 향상시킬 수 있다. network security with capability(capability로 네트워킹 보안) 각종 능력을 가지고 네트워킹 보안을 향상 시킬 수 있다. anti-snifferring과 같은 것은 1024보다 아래의 포트를 bind할 수 없다. routing 규칙이나 침입차단 시스템도 변경할 수 없게 된다.

Scanner detector in kernel(커널 안의 스캐너 탐지) SnSA는 시스템을 scan하려는 사람들을 탐지해낼 수 있도록 scanner detector를 제공한다. scanner는 nmap이나 satan과 같은 툴을 이용하여 half-open scans 혹은 nornam scans 등을 결정할 수 있다. 이때는 raw sockets을 사용불가로 하면 유용하다. 이런 경우에는 어떤 소켓도 사용할 수가 없게 된다. user space detector보다 한 차원 높은 보안 기능이 될 것이다.

침입자 반응 시스템 SnSA는 정의된 규칙에 따라 위반사항을 결정할 때, 다음과 같은 방법에 의해서 행동에 반응할 수 있다.

Logging the message(메세지 기록하기) 누군가 규칙을 위반했을 때 snsa_security_log는 klogd에 메시지를 기록한다.

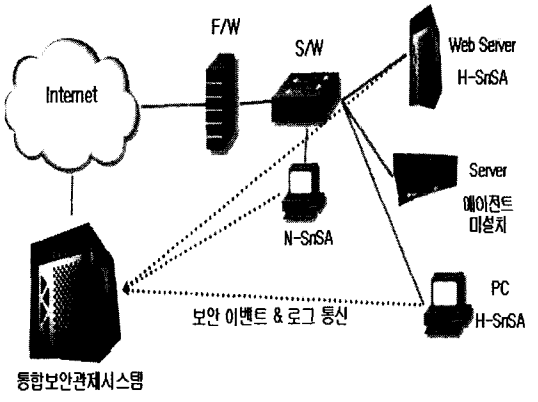
또한 anti_logging_flood의 능력도 가지고 있다. 커널 컴파일 할 때 이 부분을 설정해 줄 수 있다. Logging the message via mail server (mail 서버를 거쳐서 메시지 기록하기) SnSA는 메일로 메시지를 보내는 새로운 특징을 가지고 있다. 메일서버의 ip와 주소 등을 커널 컴파일 할 때 정해줄 수 있다.

Shutdown the console 사용자가 규칙을 위반했을 때 유저콘솔을 shutdonw시킨다.

5. 성능 평가

5.1 테스트 환경

본 논문에서는 소규모 네트워크 실시간 모니터링 및 호스트간의 통신을 통하여 불법적입 침해사고에 대하여 대응할 수 있는데 중점을 두었다.



(그림 5-1) SnSA 적용 소규모 네트워크 평가환경

테스트 환경을 세팅하는 과정에서 고려해야 할 것은, 침입을 경고 받는 방법, 해커가 네트워크의 제품 서버를 침입하는 것을 막는 방법이다. 여기에 대한 해결책으로 침입차단 시스템 뒤의 자체의 서브넷(subnet) 위에 N-SnSA를 설치하여 네트워크의 모든 침입을 탐지 하고 H-SnSA를 설치하여 호스트의 파일 보호 및 로그파일을 탐지 통합보안관제 시스템으로 전송하여 소규모 네트워크 정보보호에 대응한다.

〈표 5-1〉 Test System Configuration

	N-SnSA	H-SnSA 적용 Web Server	H-SnSA 적용 PC	통합보안 관제 시스템
CPU	X86 (PIII 800)	X86 (PIII 1000)	X86 (PIII 800)	4 ultra SPARC
RAM	128M	256M	128M	4G
OS	Redhat 7.1	Redhat 7.1	WOW 7.0	Sun OS 5.8
DB	MySQL	MySQL		Oracle

5.2 공격 시나리오

테스트에서 소규모 네트워크에 침입을 가정한 공격을 가하여 SnSA에서 공격을 탐지하여 통합보안관제 센터로 보낼 수 있는지를 실험하였다.

소규모 네트워크에 대하여 다음과 같은 4가지 공격을 사용하였다.

(1) Case 1 : 로그파일 삭제/로그파일

변조 setuid program에 buffer-overflow 공격을 수행하고, privileged mode에서 'kill' 명령어를 사용하여 로그파일을 삭제하거나 변조한다.

(2) Case 2 : 루트권한 불법 획득

리눅스 시스템에서는 기본적으로 NFS 서버 가 mountd를 구동하기 때문에 매우 위험하며, NFS 클라이언트가 NFS 서버의 파일에 접근하기 위해서는 먼저 파일 시스템을 마운트 한다는 요

청을 하게 되는데, 이 NFS 마운트 요청을 처리하는 소프트웨어(mountd 프로그램)에 버퍼 오버플로우 취약점으로 루트권한을 획득 한다.

(3) Case 3 : SYN Port Scan

해킹의 전단계로 대상 컴퓨터가 TCP/IP의 포트를 Open하여 서비스를 하는지를 알아내기 위한 방법으로 포트번호 0번부터 65,535번 포트까지 순차적 또는 무순위로 SYN을 보내어 서버가 응답하는 SYN/ACK를 확인한다.

(4) Case 4 : IP Spoofing

TCP/IP의 프로토콜의 구조적 결함을 이용하여 신뢰성 있는 호스트로 위장하여 공격에 취약한 시스템에 트로이 목마 프로그램이나 백도어(Backdoor)가 설치 될 수 있다.

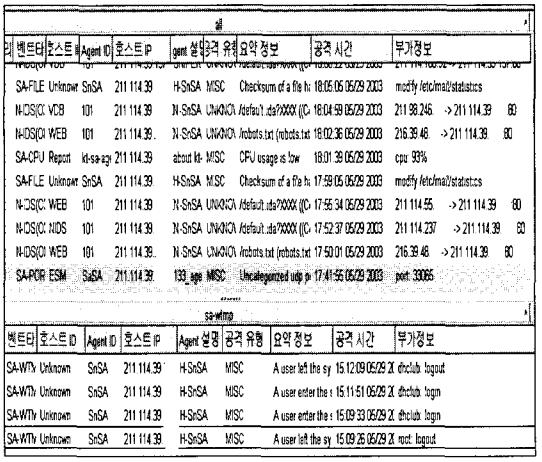
5.3 테스트 성능결과

소규모 네트워크에 직접적으로 가해진 침입에 따른 통합보안관제 시스템과 연동 테스트 결과 구현된 테스트환경에서의 실험결과는 <표 5-2>과 같다. 가상 공격으로 선정한 Case 1~4의 시나리오의 모든 조건에서 소규모 네트워크에 직접적으로 가해지는 침해사고에 대해 설치된 SnSA에서 탐지하여 통합관제 시스템으로 연동하는 결과를 나타냈다.

〈표 5-2〉 테스트 결과

	Case 1	Case 2	Case 3	Case 4
탐 지	Detect	Detect	Detect	Detect

기타 Target Host로 가해진 침입에 따른 탐지 결과 가상의 외부의 공격자(Intruder)로부터 Target Host에 가해진 일상적인 공격에 대하여 통합보안관제 시스템과 연동이 된 현황은 다음 그림과 같다.



(그림 5-2) 통합보안관제 시스템 연동

(그림 5-2)에서와 같이 SnSA 에이전트가 호스트의 Syslog를 통합관제 시스템에 연동하여 요약 정보, 공격시간, 부가정보에 대하여 나타내고 있다.

6. 결 론

본 논문에서 통합보안관제 시스템은 소규모 네트워크에 365일 24시간 외주를 통하여 네트워크 내외부의 실시간 불법침입에 탐지하여 그에 따른 대응 방안을 모색하는데 그 목적이 있다. 소규모 네트워크에서 통합보안관제 시스템 연동을 위한 에이전트의 개발은 보안성을 향상시키는 필요성을 제시하였고, 통합보안관제를 위한 필수 요소인 침입탐지 기능과 호스트 보안기능을 기반으로 통합보안관제 시스템과 연동 가능한 SnSA를 실제 구현하고 소규모 네트워크에 적용하여 효율적이고 합리적인 소규모 네트워크와 통합보안관제 시스템 간의 모델을 수립하였다.

소규모 네트워크에서는 경제적인 문제로 높은 비용이 들어가는 보안장비의 설치는 중소기업의 여건상 어려운 현실이다. 그렇지만 정보보호는 꼭

필요한 요소라는 사실에서 출발한다. 본 논문에서 구현한 SnSA의 특징은 네트워크 또는 호스트에 대한 불법 침입 탐지기능, 통합보안관제 시스템의 연동기능, 호스트 내부자원의 보호기능으로 소규모 네트워크 자체의 보안 기능 강화의 역할을 수행할 수가 있었다.

SnSA의 테스트 결과 소규모 네트워크의 불법 침입에 대하여 탐지 즉시 통합보안관제 시스템으로 보내져서 실시간 대응이 가능하였지만 기능부분에서 상용되는 침입탐지 시스템과 비교하여 성능부분에 문제가 있었다. 하지만 소규모 네트워크에서 가장 문제가 되고 있는 보안에 대한 경제적인 문제를 고려할 경우 SnSA를 통하여 실시간 관제 서비스를 가능케 한다면 소규모 네트워크는 불법적인 침입사고를 실시간으로 대응할 수 있고 통합보안관제 시스템과 연동을 통해 보안사고나 침해사고에 대한 통계 분석 등으로 소규모 네트워크의 관리자와 경영자의 부담은 줄이고 보안효율은 높이 증대할 수 있다.

현재 통합보안관제를 운용하는 회사들의 공통점은 기업의 이익을 목적으로 대형 IDC나 대형 네트워크에 초점으로 개발되고 운영된다. 지난 2003년 1월 서버 보안패치의 문제로 전국이 마비된 경우가 있었고, 해킹의 경유지로 이용되는 빈도가 높으며, 그에 따른 오인으로 국가적 차원의 명예 또한 손상되고 있다. 그렇기에 소규모 네트워크에 대한 보호기능을 강화하여 무분별한 외부의 침입에 대해 내부 네트워크 환경을 보호하기 위한 노력을 강화하고 새로운 침입대응 능력을 강구해야만 한다.

그 방법으로 먼저 소규모 네트워크 환경에 적합한 침입차단 시스템이나, 침입탐지 시스템을 개발하여 무분별한 외부의 침입에 대해 내부 네트워크자원을 보호하며, 정부나 기업에서는 경기대학교 통합보안관제 시스템과 같은 중, 소규모 네트워크를 저비용으로 보안관리해주는 시스템의 대중화가 필요하다.

본 논문에서의 구현된 SnSA의 경우 Linux기반의 에이전트로서 통합보안관제 시스템과의 연동기능을 이용하기 위하여 침입탐지 기능을 강화하였다. 보완할 점으로는 통합보안관제 시스템과 소규모 네트워크간의 연동을 위하여 어떤 OS에서나 어떤 시스템에도 작동하며 본래의 시스템 성능을 저해하지 않는 에이전트에 대한 연구를 할 필요성이 있다.

참 고 문 헌

[1] Rebecca Bace, ICSA, Inc., "An Introduction to Intrusion Detection Assessment for System and Network Security Management", 2000.

[2] E. Zwicky, S. Cooper and D. Chapman, Buiding Internet Firewalls, O'reilly, 2000.

[3] E, Amosoro, Fundamentals of Computer Security Architecture, Design, Deployment & Operations, Osbourne/McGraw-Hill, 2001.

[4] Bruce Schneier. 역자 : 채윤기, Secrets & Lies. 서울 : 나노미디어. 2002.

[5] Ruixi Yuan, Virtual Private Network, Technologies and Solutions.

[6] 이병영, "시스템 보안을 위한 침입탐지 시스템의 설계 및 구현", 2001.

[7] William Stallings. 역자 : 최용락, 소우영, 이재광, 이임영, 컴퓨터 통신보안, 그린, 2001.

[8] 임성수, "통합보안 시스템 구축을 위한 보안툴의 이벤트 통합에 관한 연구", 2001.

[9] "정보보호 컨설팅 사업 현황과 전망", 정보보호21c, 2001. 6.

[10] http://icat.nist.gov/icat.cfm(취약성으로 인한 피해 사례).

[11] http://www-rnks.informatik.tu-cottbus.de/~sobirey/ids.html.

[12] http://www.linuxsecurity.com/feature_stories/feature_story-12.html.
[13] Overview of IETF, http://www.ietf.org.
[14] 사이버패트롤(http://www.cyberpatrol.co.kr) 사이버패트롤 소개.
[15] 이글루시큐리티(http://www.e-patrol.co.kr) 사업소개.
[16] 코코넛(http://www.coconut.co.kr)사업소개.
[17] 해커스랩(http://www.hackerslab.com)사업소개, ESM 기술.
[18] 시큐어소프트(http://www.securesoft.co.kr).
[19] 한국ESM(http://www.kesm.co.kr) ESM 기술.
[20] http://www.linuxnewbie.org/nhf/intel/security/snort.html.
[21] http://www.linuxsecurity.com/feature_stories/feature_story-12.htm.
[22] http://stat.nca.or.kr/main3.html 한국전산원 2002 정보화통계 DB.
[23] http://www.linuxnewbie.org/nhf/intel/security/.
[24] 국내 보안관제 서비스 기술 동향, 주간기술동향, 2002. 5.

이 동 휘

2000년 경기대학교 전자계산학과(이학사)
2003년 경기대학교 정보보호기술공학과(공학석사)
현재 경기대학교 통합보안관제센터 씨티알씨 팀장

신 영 준

2002년 삼척대학교 컴퓨터공학과(공학사)
2003년 경기대학교 정보보호기술공학과 석사과정
현재 경기대학교 통합보안관제센터 씨티알씨 연구원

김 귀 남

미국 캔자스대학 수학과(응용수학사)
미국 콜로라도주립대학 통계학과(통계학석사)
미국 콜로라도주립대학 기계·산업공학과(기계·산업공학박사)
현재 경기대학교 정보보호기술공학과 주임교수