데이터소스기반의 침입탐지시스템 설계

조아앵*, 박익수, 이경효, 오병균
*목포대학교 정보보호학과

e-mail: {agcho, ispark, mediakh, obk}@intra.mokpo.ac.kr

Design of Data Source Based-IDS

A-Aeng Cho*, Ik-Su Park, Kyoung-Hyo Lee,
Byeong-Kyun Oh
*Dept of Informaion Security, MokPo National University

요 약

현재까지 IDS는 관리자의 개입 없이는 효과적인 운용이 불가능하고, IDS를 사용하더라도 여전히 침입 발생 가능성이 있고, 다양한 우회 가능성이 존재한다. 본 논문에서는 기존에 제안된 침입탐지 시스템을 분석하고, C-Box에 규정된 정책을 이용한 데이타소스 기반의 침입탐지 시스템을 설계하여 이를 실험하였다. 본 연구는 데이터 소스 기반에서 침입 탐지 방법 기준의 비정상적인 형태에 의한 탐지와 오류에 의한 탐지기법을 적용하였으며, IDS에 침입 탐지 정책을 설계하였고, 규정에 의한 정책중심의 침입탐지 기법을 정상적인 동작과 비정상적인 동작을 구분하는 경계를 정의한다. 또한, 침입탐지 정책을 이용한 호스트기반 IDS를 설계하고 구현함으로서 정보시스템의 취약성을 보완할 수 있었다. 침입탐지 실험을 위한 시스템 호출 기술은 커널에 프로세스들의 특성을 자세하게 정의하고, 이를 실행할 수 있도록 기반을 구축함으로서 가능하게 하였다.

1. 서론

정보 시스템(Information System)에 대한 고의적 불법적인 행위를 침입(Intrusion)이라 하며, 컴퓨터 또는 네트워크 자원의 무결성(integrity), 비밀성(confidentiality), 가용성(availability)을 저해하는 일련의 행위를 말한다. 침입탐지시스템(IDS: Intrusion Detection System)은 컴퓨터와 네트워크에서 발생하는 이벤트들을 모니터링(Monitoring)하고, 침입 발생 여부를 탐지하고, 대응(Response)하는 자동화된 보안시스템이다[1-5].

현재까지 IDS에 적용되는 기준이나 기법에는 크게 데이터 소스 기준(E-Box), 침입 탐지 방법 기준(A-Box), 대응 행동 기준(R-Box) 등으로 분류될 수 있다[2, 3, 4].

데이터 소스 기준은, 침입탐지를 위한 데이터 수집 방식에 따라 네트워크 기반, 호스트 기반, 하이브리드 기반이 있다. 네트워크 기반 IDS는 네트워크 패킷과 Promiscuous mode를 데이터 소스로 한다. 호스트 기반에서 탐지 불가능한 침입 탐지가 가능하고, 전체 네트워크에 대한 침입 탐지와 기존 네트워크 환경의 변경이 필요없는 장점이 있는 반면, 탐지된 침입의 실제 공격 성공 여부를 알지 못하고, 고부하(High-volume)·스위치(Switch) 네트워크에는 적용이 어렵고, 다양한 우회 가능성이 존재하는 단점이 있다. 호스트 기반 IDS는 OS 감사자료(audit trail), 시스템·응용 프로그램 로그를 데이터 소스로 한다. 네트워크 기반에서 탐지 불가능한 침입 탐지가 가능하고, 고부하·스위치 네트워크에도 적용 가능하고, 우회 가능성이 거의 없는 장점이 있는 반면, 모든 개별 호스트에 대한 설치 및 관리가 어렵고, IDS가 설치된 플랫폼에 성능 저하, 플랫폼 자체에 침입 가능성이 있는 단점이 있다. 하이브리드 IDS는 네트워크 기반 IDS와 호스트 기반의 IDS의 장점을 모두 수용한다[6, 7, 8].

침입 탐지 방법 기준은 오용 탐지(misuse)와 비정상행위 탐지(anomaly)가 있다. 오용 탐지 IDS는 침입 시그너쳐(signature)를 이용하여 CGI, Buffer Overflow와 같은 특정 침입을 나타내며, 상대적으로 낮은 오판율과 침입에 사용된 특정한 도구·기술에 대한 분석이 가능한 반면, 새로운 침입유형에 대한 탐지가 불가능하고, 다양한 우회 가능성이 존재한다. 비정상행위 탐지 IDS는 정상행위에서 벗어남을 탐지함으로, 새로운 침입 유형에 대한 탐지가 가능한 반면, 방대한 사용자·네트워크 활동과 많은 시간이 요구되므로 정상 행위를 예측하기 어렵고, 높은 오판율 등의 단점이 있다[9].

대응 행동 기준 분류는 수동적 대응행동(passive response)와 능동적 대응행동(active response)이 있다. 수동적 대응행동은 관리자에게 침입 정보만 제공하고 실제 대응행동은 제공된 정보(발생시간, 침입유형, 침입자 주소 등)를 기초로 관리자가 수행한다. 능동적 대응행동은 실제 대응행동을 IDS가 자동적으로 수행한다[10, 11]. 다음 표 1은 IDS에 적용되는 기준이나 기법에 따라 분류한 것이다.

[표 1] IDS 적용되는 기준이나 기법의 분류

적용 기준	적용 기법
데이터 소스 기준	Host-based IDS, Network-based IDS
탐지 모델 기준	Misuse, Anomaly, Hybrid Detection
감사데이터 분석시점	Real-time Analysis, Batch-time Analysis
침입분석 기법	Signature, Statistical, Integrity
탐지시간 기준	Real time, Non-real time
대응방법 기준	Active method, Passive method
감사자료 대상	System-log, Network-packet

기존에 제안된 IDS로부터 보안전략, 보안정책의 부재 등으로 조직 내의 근본적인 문제점 해결이 불가능하고, 침입탐지 결과 대응 및 분석을 관리자의 개입 없이는 효과적인 운용이 불가능하고, 높은 오판율로 관리자에게 불필요한 시간과 자원 소비가 발생하고, IDS를 사용하더라도 침입 발생 가능성이 여전히 존재하여 다양한 우회 가능성이 존재함을 알 수 있었다. 또한, IDS를 다른 보안 메커니즘의 기능으로 대체가 불가능함을 알 수 있었다[5, 6].

본 연구는 IDS에 적용되는 기준이나 기법의 분류에서 데이터 소스 기반에서의 침입탐지 정책을 이용한 호스트 기반 IDS를 설계하고 구현함으로서 정보시스템의 취약성을 보안하는 기법을 개발한다. 본 연구의 호스트 기반 IDS에 적용된 탐지기법은 비정상적인 형태에 의한 탐지, 오류에 의한 탐지를 기반으로 한다.

제안된 데이터 소스 기반 IDS는 침입 탐지 정책을 설계하여 정보시스템의 취약성을 보완하도록 하였으며, 규정(rule)에 의한 정책중심의 침입탐지 기법을 정상적인 동작과 비정상적인 동작을 구분하는 경계를 정의하였다. 이 규정에는 프로그램 실행에서 침입이라고 판단되는 불법적인 시도들을 구체적으로 열거한다.

논문의 구성은 2장에서 데이터 소스 기반 IDS와 규정의 적용범위(scope)를 제시하고, 3장에서 데이터 소스 기반 C-Box IDS의 정책 기술 내용을 기술한다.

제안된 데이터 소스 기반 IDS는 Signature 기반이나, Profile 기반의 고전적인 공격을 극복하는데 많은 장점이 있다. C-Box에 포함된 보안의 범위는 시스템 자원의 사용 의도에 따라 명확하게 정의하며, 보안규정은 프로그램의 동작을 탐지하고 분석하는 기반이 된다.

2. 데이터 소스 기반 IDS와 규정의 적용범위

IDS는 시스템의 비정상적인 사용과 오남용을 탐지하여 이를 차단하는 보안시스템으로 외부 및 내부 사용자의 모든 행위를 분석하고 내부 자료의 외부 유출 등을 탐지하여 시스템의 자원을 관리 및 통제한다. 따라서 IDS는 침입탐지와 분석을 통하여 침입자에 대한 피해를 최소화하고, 해커의 공격에 적절하게 대응하기 위하여 침입에 대한 정책을 규정(Rule)하여 적용한다.

호스트 기반 IDS에 적용된 탐지기법은 정상적인 동작(normal behavior)의 Profile를 특성화하여, 정상적인 형태와 분명하게 차이가 나는 패턴을 침입으로 탐지하는 오용탐지(Misuse)와 알려진 침입의 특징(collection of signatures)을 규정한 다음, 그러한 패턴과 일치하는 활동을 침입으로 탐지하는 이상탐지(Anomaly)를 기반으로 한다. 또한 모든 침입 가능성을 파악하기 어렵고, 모든 동작은 사용된 패턴에 따라 변경되기 때문에 비정상적인 동작을 정상적인 동작으로 탐지되거나 정상적인 동작을 침입으로 판단하는 탐지오류가 발생할 수 있기 때문에 이들 두 가지 기법은 서로 보완적이어야 한다.

정보시스템에 대한 침입은 의도적이지 않은 것처럼 시스템의 자원에 접근하기 때문에 C-Box에 침입탐지 정책을 설정하고, 시스템 자원에 대한 프로세스의 접근을 제어하는 세부적인 규정을 이용하여 시

행한다. 이러한 규정에 의해 시행되지 않은 접근은 침입으로 판단한다.

1) 시스템의 자원과 각 자원에 대한 접근형태들을 영역별로 구분한다.

2) C-Box에서 시행 규정에 대한 세부사항의 특성은 다음과 같다.

- 파일 시스템의 Object에 대한 접근 허가 권한
- 파일 시스템에 대한 접근 방법
- Uid와 Gid에 대한 허가권의 변화
- send, receive, block, ignore, handle 할 수 있는 Signals
- Scheduling priority를 수정할 수 있는 프로세스 특징
- 다른 시스템 자원에 대한 제어 등이다.

3) 정책과 규정에 대한 세부사항을 잘 표현하고 여러 형태의 침입에 대하여 효과적으로 제어하기 위해서는 시스템 호출 리스트가 제공되어야 한다.

4) 시스템의 자원은 시스템 호출을 통하여 접근되기 때문에 시스템 호출이 허용되지 않으면 자원에 접근할 수 없다.

본 논문에서 Linux 커널에서 시스템 호출을 구현하기 위하여 다음과 같은 시스템 호출을 규정하였다.

afs_syscall	getgid	lsatat64	sched_yield
alarm	getgroups	mpx	setitimer
break	getitimer	msymc	sgetmask
brk	getpgid	nanosleep	stat
capget	getpgrp	newselect	stat64
chdir	getpid	oldfstat	statfs
fchdir	getppid	oldlstat	stty
fdatasysnc	getpriority	olduname	sysfs

프로그램에 대한 정책은 상속의 영역을 표시하였으며, 상속 규정은 프로그램들이 규정을 공유함으로 중복성을 제거하고, 매번 컴파일을 하지 않기 때문에 효과적으로 구현할 수 있다. 프로그램에 대한 정책과 규정을 구현하기 위해서는 여러 기법과 도구들이 이용되는데 본 연구에서는 Intended Semantics, Audit Trail, Configuration, Existing Templates 등을 중심으로 규정을 설정하고, 이들 규정을 Linux OS의 커널 기반 참조 모니터에 적용하였다.

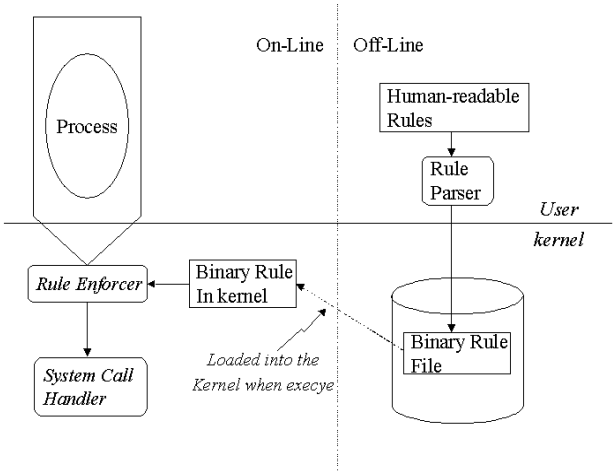
프로그램의 동작을 제어하는 최근의 기법들을 살표보면 언어기반은 Program Correctness Based Mechanism, Program Typed-Based Mechanism이 있고, 패턴 기반은 System Call이 있고, 커널 기반은 Linux Intrusion Detection System, linux Security Module, Sub-Domains, Sandboxing Systems, Domin-and-Type-Enforcement Based System이 있다.

3. 데이터 소스 기반 C-Box IDS 정책 기술

이 장에서는 정책이 어떻게 정의되고 수행되는지를 나타내는 C-Box 시스템의 구조를 기술하고, 정책의 세부사항을 설명한다.

3.1 C-Box 시스템의 구조

C-Box는 정책의 세부사항과 구문분석, 정책의 적재와 시행으로 설명할 수 있다. 그림 1은 C-Box 시스템 구조이다.



[그림 1] Check-Box System Architecture

[정책의 세부사항과 구문분석]

프로그램의 실행에 대한 C-Box 정책은 편집기에서 사용된 것처럼 사람이 읽을 수 있는 형태로 자세히 기술한 다음, 구문분석 프로그램에 의해 Binary File로 분석한다. 이 부분은 프로그램이 실행되기 전 Off-Line으로 행해진다.

[정책의 적재와 시행]

C-Box 정책은 시스템 호출을 통해서만 접근할 수 있도록 시스템 자원의 접근을 제어하는 방법을 의미하며, 규정 시행을 위한 장소는 시스템 호출에 대한 커널의 시작점이다.

3.2 자원의 여러 형태에 대한 규칙

이 절에서는 파일 시스템의 Object, Uid/Gid list, Signal, Socket, Device file 등과 같은 시스템 자원의 여러 부류에 대한 규정을 기술한다. 각 부류의 자원은 각각의 자원과 자원에 관련된 의미를 규정한 구문의 기술은 C-Box 시스템에서 모든 가능한 규정을 서술한 것으로 그 내용은 Rules of File System Objects, Rules of Identities, Rules of Signals, Socket Rules, Device Rules 이다.

3.3 프로세스의 상태

규정으로 구체화한 프로세스의 상태는 가능한 여러 형태의 공격에 대하여 프로세스를 보호한다. 시스템 호출에서 상태의 구체화는 프로세스 상태의 갱신과 추적이 필요한 경우를 중심으로 소수의 프로세스 상태를 설정하였으며, 상태의 설정기준은 꼭 필요할 때만 상태를 추가한다. 본 연구에서 설정한 상태는 Identity state(initial root state, user state, re-root state), system call count, signal handler 이다.

3.4 커널에 의한 영향력

본 연구에서 시스템에 대한 가장 중요한 설계의 기준은 커널에대한 영향력을 최소화하는 것이다. 기능의 배치는 커널에 대한 영향력을 감소시키기 위하여 실행한다. Linux에서 침입회피 구현에 대한 참조는 시스템 호출 통로점과 프로세스생성과 종료에 대한 커널코드에서 제2의 시점에서 가로채기한다. 커널자원에 대한 전체적인 영향은 10 또는 20라인으로 제안한다. 프로세스 시행의 나머지는 메모리에 할당하고 설치 규정은 독립적 모듈이다.

4. 결론

컴퓨터와 네트워크 자원의 무결성, 비밀성, 가용성을 저해하는 일련의 행위를 침입이라 하며, IDS는 컴퓨터와 네트워크에서 발생하는 이벤트들을 모니터링하고, 침입 발생여부를 탐지하고, 대응하는 자동화된 보안 시스템이다.

기존에 제안된 IDS의 분석결과 보안전략, 보안정책의 부재 등으로 조직 내의 근본적인 문제점 해결이 불가능하고, 침입탐지 결과 대응 및 분석을 관리자의 개입 없이는 효과적인 운용이 불가능하며, 높은 오판율로 관리자에게 불필요한 시간과 자원 소비가 발생하고, IDS를 사용하더라도 침입 발생 가능성이 여전히 존재하며, 다양한 우회 가능성이 존재함을 알 수 있었다.

IDS에 적용되는 기준에서 데이터 소스 기반에서 침입탐지 정책을 이용한 호스트 기반 IDS를 설계하고 구현함으로서 정보시스템의 취약성을 보안하는 기법을 개발하였다. 호스트 기반 IDS에 적용된 탐지기법은 비정상적인 형태에 의한 탐지, 오류에 의한 탐지를 기반으로 하여, 규정(rule)에 의한 정책중심의 침입탐지 기법을 정상적인 동작과 비정상적인 동작을 구분하는 경계를 정의하였다. 이 규정에는 프로그램 실행에서 침입이라고 판단되는 불법적인 시도들을 구체적으로 열거하였다.

제안된 데이터 소스 기반 C-Box IDS 정책은 Signature 기반이나, Profile 기반의 고전적인 공격을 극복하는데 많은 장점이 있다. C-Box에 포함된 보안의 범위는 시스템 자원의 사용 의도에 따라 명확하게 정의하며, 보안규정은 프로그램의 동작을 탐지하고 분석하는 기반이 된다. C-Box기법은 시스템 호출을 시행할 때 프로그램의 실행에 대한 점검을 규정시행 모듈을 이용하여 실시함으로 효율적이며, 점검을 규정하는 구문과 의미는 침입을 탐지하기에 효율적이다. 실험결과의 성능은 설계의 요소와 밀접하기 때문에 성능보다는 안전성을 기준으로 하였다.

본 연구에서는 서명기반 시스템과 통계적 윤곽기반 시스템의 공격을 결합하여 분석함으로서 더욱 효과적인 안전성을 높였다.

참고문헌

[1] Debar, H., Dacier, M., and Wespi, A. Toward a taxonomy of intrusion detection systems. Computer Networks 31. 1999.

[2] Wrlingsson, U. and Schneider, F.B. IRM enforcement of Java stack inspection. In IEEE Symposium on Security and Privacy. 2000.

[3] Jackson, K. A. Intrusion Detection System(IDS) product review. IBM internal confidential document, IBM Research Division, 1999.

[4] Jain, K. Sekar. User-level infrastructure for system call interposition; A platform for intrusion detection and confinement. in proceedings of the Network and Distributed Systems Security Symposium. 2000.

[5] Ko, C., raser, T., Badger, L., & Kilpatrick, D. Detecting and countering system intrusions using software wrappers. In proceedings of the 9th USENIX Security Symposium. 2001.

[6] Paxson, V. Bro: A system for detecting network intruders in real-time. In the 7th USENIX Security Symposium. 1998.

[7] Sekar, R. & Uppuluri, P. Synthesizing fast intrusion detection systems from high-level specification. In the 8th USENIX Security Symposium, pp. 63-78. 1999.

[8] Wagner, D. A. and Dean, D. Intrusion Detection via analysis. In proceedings of the IEEE Symposium on Security and Privacy. 2001.

[9] Xie, H. and Biondi, P. The Linux Intrusion Detection Project. 2001.

[10] Bernaschi, M., Gabrielli., & Mancini, L. REMUS: A security-enhanced Operating System. ACM Trans. Inf. Syst. Sec. 5, 1. 2002.

[11] Atkinson, R. Security architecture for the Internet protocol Internet RFC-1825. 1995.