정책을 이용한 Host 기반의 침입탐지 시스템 설계 및 구현

박익수° 이경효 이군승 명근홍 오병균

{ispark°, kyoung}@mis2.mokpo.ac.kr, comelgs@kornet.net, hhanaro@hitel.net, obk@mokpo.ac.kr

Desing and Implementation of Host-Based IDS for Policy-Driven

IkSu Park°, kyoungHyo Lee, GunSo Lee, GnHong Meo, ByeongKyun Oh

Dept. of Information Security , Mokpo National University

요 약

정보시스템에 대한 침입탐지는 네트워크 기반의 침입탐지시스템에 의존하였으나, 네트워크 규모의 확대와 암호사용의 증가로 인하여 호스트 기반의 침입탐지시스템을 중심으로 연구되고 있다. 본 논문에서는 CB(Check-Box)에 규정된 정책을 이용한 호스트 기반의 침입탐지 시스템을 설계하여 이를 실험하였다. 침입탐지 실험을 위한 시스템호출 기술은 커널에 프로세스들의 특성을 자세하게 정의하고, 이를 실행할 수 있도록 기반을 구축함으로서 가능하게 하였다. 이러한 기법의 특성은 실행 가능한 프로세스가 시스템에 자원에 정당하게 접근할 수 있는 정책을 자세하게 규정해야 하며, 규정을 기술하기 위한 언어는 보안영역을 효과적으로 표현하고 번역될 수 있어야 한다.

본 연구는 Linux의 커널에서 침입탐지기법에 대한 모형을 제시하고, 공격에 대한 탐지와 탐지결과를 검증할 수 있는 정책을 설정하였다. 제안된 시스템은 커널의 변화에 대한 영향력을 최소화하도록 함으로서 새로운 커널을 쉽게 설치할 수 있기 때문에 정책에 의한 호스트기반의 침입탐지시스템은 운영, 탐지, 분석을 통하여 침입을 예방할 수 있는 방안을 마련할 수 있다.

1. 서 론

오늘날 컴퓨터와 통신기술의 급속한 발전은 다양한 형태의 정보 서비스를 신속하게 제공하는 정보화 사회를 이룩하였다. 정보화 사회에서 정보의 수집, 분석 및 활용능력은 생존과 경쟁력을 좌우하는 중요한 자산이 되고 있을뿐만 아니라, 정보는 모든 사회활동의 주요한 원천이다. 그러나 정보를 취급하는 과정에서 정보에 대한 무단 유출, 파괴, 변조 등과 같은 공격이 자행되고 있으며, 인가 받지 않은 불법적인 사용자에 의한 정보시스템의 침입, 파괴, 방해, 불건전한 정보의 유통 등 피해가 급증하고 있다.

이러한 불법적인 행위에 대한 정보시스템의 취약성을 보호하고 관리하기 위하여 침입차단이나 침입탐지 등을 이용한 정보보호 기술의 연구가 활발히 이루어 지고 있다.

침입차단 시스템은 방화벽을 이용하여 정보시스템에 대한 외부로부터의 침입을 막을 수 있으나, 내부에서 일어나는 해킹사고나 불법행위에 대해서는 보안에 한계가 있다. 이처럼 침입차단에 의한 단점을 보완하기 위하여 침입탐지시스템의 기법들이 연구되고 있다[1][2].

침입탐지시스템(IDS: Intrusion Detection System)은 시스템의 비정상적인 사용과 오남용을 탐지하여 이를 차단하는 정보보안시스템으로서 외부 및 내부 사용자의 모든 행위를 분석하고 내부 자료의 외부유출 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[3].

「본 논문은 2003 학년도 목포대학교 학술연구비 지원에 의하여 연구되었음」

다음 [표 1]은 IDS에 적용되는 기준이나 기법에 따라 분류한 것이다.

[표 1] IDS의 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

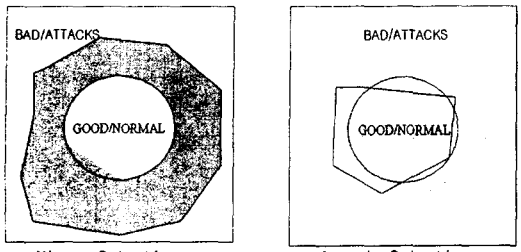
[표 1]에 의한 데이터 소스(Data Source) 기준은 침입탐지를 위한 데이터를 어디에서 수집하느냐에 따라 Host-based IDS와 Network-based IDS로 구분하는데, Network이 일반화되지 않은 초기에는 Host중심이었고, Network가 일반화되면서는 Network 중심이였는데, 최근에는 다시 Host중심의 IDS를 구현하는 추세이다[4].

본 연구에서는 침입탐지 정책을 이용한 Host-based IDS를 설계하고 구현함으로서 정보시스템의 취약성을 보완하는 기법을 개발한다. 본 연구의 Host based IDS에 적용된 탐지기법은 다음의 두 가지 기법을 기반으로 한다.

<기법-1> 비정상적인 형태에 의한 탐지(Anomaly Detection)
먼저 정상적인 동작(normal behavior)의 통계적 형태(profile)를 특성화한 다음, 정상적인 형태와 분명하게 차이가 나는 패턴을 침입으로 탐지한다.

<기법-2> 오류에 의한 탐지(Misuse Detection)
먼저 알려진 침입(known attack)의 특징들(collection of signatures)을 규정한 다음, 그러한 패턴과 일치하는 활동

(activities)을 침입으로 탐지한다.



[그림. 1] IDS에서 Misuse Detection과 Anomaly Detection

[그림 1]은 이론적인 영역과 현실적인 영역에 대한 IDS의 모형을 나타낸 것이다. 개념적으로 <기법-1>은 비정상적인 행동(bad behavior)을 기반으로 하고, <기법-2>는 정상적인 행동(normal behavior)을 기반으로 한다. 이들 두 가지 기법은 완전하지 않기 때문에 서로 보완적이여야 한다. 왜냐하면 모든 침입 가능성을 파악하기도 어렵고, 모든 동작은 사용된 패턴에 따라 변경되기 때문에 비정상적인 동작을 정상적인 동작으로 탐지되거나 정상적인 동작을 침입으로 판단하는 탐지오류가 발생할 수 있기 때문이다[5][6].

본 연구에서 제시한 정책중심의 침입탐지 기법은 규정(rule)에 의해 정상적인 동작과 비정상적인 동작을 구분하는 경계를 정의한다. 이 규정에는 프로그램 실행에서 침입이라고 판단되는 불법적인 시도들을 구체적으로 열거한다. 예를 들면 프로세스를 관리하는 규정은 프로세스가 시스템의 어느 자원에 어떠한 방법으로 접근할 수 있는지를 정확하게 정의함으로서 침입탐지를 판단한다.

논문의 구성은 2장에서 규정의 적용범위(scope)가 무엇인가를 제시하고, 프로그램에서 정확한 실행동작을 이해할 수 있도록 규정을 정의한다. 따라서, 이러한 규정은 프로그램에 대한 template, audit trail, configuration, semantics 등을 이용하여 정의하며, 프로그램이 실행될 때 이들 규정이 커널에 load 되고, 규정의 시행은 프로세스가 프로그램의 내용을 실행할 때 프로세스의 동작이 규정에 따라 실행과정을 서술한다. 3장에서는 제안한 CB(Check-Box) IDS에 포함된 여러 가지 정책의 내용들에 대하여 기술하였다. 특히, CB IDS의 정책기반 기법은 서명(signature)기반이나 윤곽(profile)기반의 고전적인 공격을 극복하는데 많은 장점이 있다. CB에 포함된 보안의 범위는 시스템 자원의 사용 의도에 따라 명확하게 정의하며, 보안규정은 프로그램의 동작을 탐지하고 분석하는 기반이 된다. 따라서, 이러한 보안규정의 활용은 다음과 같은 2가지 장점을 제공한다. i) 알려지지 않은 공격을 탐지할 수 있다. ii) 합법적인 동작이 공격으로 잘못 판단된 것을 탐지할 수 있다.

CB의 정책기반 기법에 대한 가장 중요한 장점은 탐지가 실시간으로 이루어지는 것으로서 인증되지 않는 접근이나 실행을 쉽게 방지할 수 있다.

2. CB에 대한 정책의 설정과 규정의 세부사양

정보시스템에 대한 침입은 의도적이지 않은 것처럼 시스템의 자원에 접근하기 때문에 CB에 침입탐지 정책을 설정하고, 시스템 자원에 대한 프로세스의 접근을 제어하는 세부적인 규정을 이용하여 시행한다. 이러한 규정에 의해 시행되지 않은 접근은 침입으로 판단한다.

다음 [표 2]는 시스템의 자원과 각 자원에 대한 접근형태들을 영역별로 구분한 것이다[7].

[표 2] Types of Resources and Access

Resoures	Types of Access
File system objects	create, open, read, write, removal, link-to, range of access permissions,
File systems	change of ownership
Identities	mount, unmount, types of mount
Processes(address spaces, signals, ..)	acquire, release, inherit
CPU cycles, process scheduling priority	read, write, deliver
System clock	raise
System/kernel memory	set, read
IPC objects: pipes, semaphores, message queues, shared memory,...	read, write
Devices, network	create, open(attach), read, write
Privileges	create/attach, open, read, write, io-control, removal, link-to, range of access permission, change of ownership
	acquire, release, raise, lower

CB에서 시행 규정에 대한 세부사항의 특성은 다음과 같다.
- 파일 시스템의 object에 대한 접근 허가 권한
- 파일 시스템에 대한 접근 방법
- uid와 gid에 대한 허가권의 변환
- send, receive, block, ignore, handle 할 수 있는 Signals
- scheduling priority를 수정할 수 있는 프로세스 특징
- 다른 시스템 자원에 대한 제어(ioctl call, socket)

정책과 규정에 대한 세부사항을 잘 표현하고, 여러 형태의 침입에 대하여 효과적으로 제어하기 위해서는 시스템 호출 리스트가 제공되어야 한다. 시스템의 자원은 시스템 호출을 통하여 접근되기 때문에 시스템 호출이 허용되지 않으면 자원에 접근할 수 없다. 본 논문은 Linux 커널에서 시스템 호출을 구현하기 위하여 다음과 같은 시스템 호출을 규정하였으며, [표 3]은 이들을 나열한 것이다[8].

[표 3] Harmless System Calls

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, 상속규정은 프로그램들이 규정을 공유함으로 중복성을 제거하고, 매번 컴파일을 하지 않기 때문에 효과적으로 구현할 수 있다. 프로그램에 대한 정책과 규정을 구현하기 위해서는 여러 기법(mechanism)과 도구(tool)들이 이용되는데, 본 연구에서는 Intended Semantics, Audit Trail, Configuration, Existing Templates 등을 중심으로 규정을 설정하고, 이들 규정을 Linux 운영체제의 커널 기반 참조 모니터에 적용하였다. 다음 [표 4]는 프로그램의 동작을 제어하는 최근의 기법들을 나타낸 것이다.

[표 4] 프로그램의 동작을 제한하는 기술

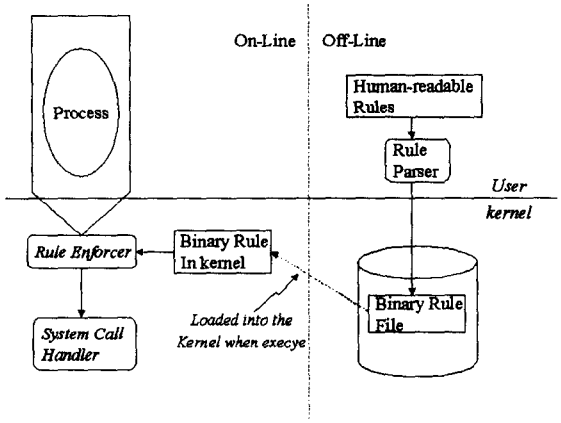
정 책	기 법
언어 기반	Program Correctness-Based Mechanism Program Typed-Based Mechanism
패턴 기반	System Call
커널 기반	LIDS(Linux Intrusion Detection System) LSM(Linux Security Module) Sub-Domains Sandboxing Systems Domain-and-Type-Enforcement Based System

3. 정책의 기술적 세부사항

이 장에서는 정책이 어떻게 정의되고 수행되는지를 나타내는 CB 시스템의 구조를 기술한 다음, 정책의 세부사항을 자세하게 논의함으로서 커널에서 CB 시스템의 효율성에 대하여 기술한다.

3.1 CB 시스템의 구조

다음 [그림 2]는 CB 시스템의 구조를 나타낸 것이다.



[그림 2] Check-Box System Architecture

<정책의 세부사항과 구문분석> 프로그램의 실행에 대한 CB 정책은 편집기에서 사용된 것처럼 사람이 읽을 수 있는 형태로 자세히 기술한 다음, 구문분석 프로그램에 의해 2진 파일로 분석한다. 이 부분은 프로그램이 실행되기 전 오프라인으로 행해진다.

<정책의 적재와 시행> CB 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 각각 자신의 특정한 구문과 의미를 갖는다. 각각의 자원과 자원에 관련된 의미를 규정한 구문의 기술은 CB 시스템에서 모든 가능한 규정을 서술한 것으로서 그 내용은 Rules of File System Objects, Rules of Identities, Rules of Signals, Socket Rules, Device Rules 등 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 등 이다.

3.4 커널에 의한 영향력

본 연구에서 시스템에 대한 가장 중요한 설계의 기준은 커널에 대한 영향력을 최소화하는 것이다. 기능의 배치는 커널에 대한 영향력을 감소시키기 위하여 실행한다. Linux에서 침입회피 구현에 대한 참조는 시스템 호출 통로정과 프로세스생성과 종료에 대한 커널코드에서 제2의 시점에서 가로채기한다. 커널 자원에 대한 전체적인 영향(충격)은 10 또는 20라인으로 제한한다. 프로세스 시행의 나머지는 에모리에 할당하고 설치 규정은 독립적 모듈이다.

4. 결 론

본 연구는 프로세스에 의해 시스템 호출 가로채기 모듈을 이용하여 시스템에 대한 침입을 탐지할 수 있는 CB(CheckBox)시스템을 제안하였다. 제안한 CB기법은 시스템 호출을 시행할 때 프로그램의 실행에 대한 점검을 규정시행 모듈을 이용하여 실시하는 간단하고 이해하기 쉬운 방법이다. 점검을 규정하는 구문과 의미는 침입을 탐지하기에 간단하고 효과적이다. 실험결과의 성능은 설계의 요소와 밀접하기 때문에 성능보다는 보안보장을 기준으로 하였다.

본 연구에서는 서명기반 시스템과 통계적 윤곽기반 시스템의 공격을 결합하여 분석함으로서 더욱 효과적인 보안을 이룩할 수 있었다.

참 고 문 헌

[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.