규칙기반 다단계 침입 탐지 시스템

민욱기*, 최종천*, 조성제*
*단국대학교 컴퓨터과학 및 통계학과
e-mail:wearemin@naver.com, godofslp@dankook.ac.kr,
sjcho@dankook.ac.kr

A Rule-based Intrusion Detection System with Multi-Level Structures

Uk-ki Min*, Jong-cheon Choi*, Seong-je Cho*
*Dept. of Computer Science and Statistics, Dankook University

요 약

본 논문에서는 보안 정책 및 규칙에 기반을 둔 네트워크 포트 기반의 오용침입 탐지 기능 및 센서 객체 기반의 이상침입 탐지 기능을 갖춘 리눅스 서버 시스템을 제안 및 구현한다. 제안한 시스템은 먼저 시스템에서 사용하는 보안 정책에 따른 규칙을 수립한다. 이러한 규칙에 따라 정상적인 포트들과 알려진 공격에 사용되고 있는 포트 번호들을 커널에서 동적으로 관리하면서, 등록되지 않은 새로운 포트에도 이상탐지를 위해 공격 유형에 대하여 접근제어 규칙을 적용하여 이상 침입으로 판단될 경우 접근을 차단한다. 알려지지 않은 이상침입 탐지를 위해서는 주요 디렉토리마다 센서 파일을, 주요 파일마다 센서 데이터를 설정하여 센서 객체가 접근될 때마다 감사로그를 기록하면서, 이들 센서 객체에 대해 불법적인 접근이 발생하면 해당 접근을 불허한다. 본 시스템은 보안 정책별 규칙에 따라 다단계로 구축하여 특정 침입에 대한 더욱 향상된 접근제어를 할 수 있다.

1. 서론

침입 탐지 시스템(IDS: Intrusion Detection System)은 네트워크 또는 호스트에서 일어나는 사건 및 사용자 행위들을 감시하면서 침입 여부를 파악하기 위해 그 사건들을 분석하고 침입에 대응하는 시스템이다[1]. 즉, IDS는 시스템의 비밀성, 무결성, 가용성, 인증 등을 위협하는 모든 상황을 탐지하는 것을 목표로 하며, 침입자의 불법적인 사용뿐만 아니라 합법적인 사용자의 오용이나 남용도 발견할 수 있다[2]. 보통 IDS는 원시 데이터의 소스(탐지 영역)에 따라 호스트 기반 방식과 네트워크 기반 방식으로 분류되며, 침입탐지 방법(유형)에 따라 오용탐지(misuse detection)와 이상탐지(anomaly detection)로 분류된다[3].

우리는 이전에 리눅스 서버 상에서 효율적인 센서객체(센서 파일 및 센서 데이터 포함)라는 덫을 이용한 이상침입탐지시스템과 네트워크 포트기반의 오용침입 탐지 기능을 가진 다단계 구조를 가진 침입 탐지 및 방어 시스템을 제안하였다.

본 논문에서는 위의 제안된 시스템을 바탕으로 단계별 보안 정책과 규칙을 적용하여 각 단계마다 정상과 비정상 동작에 대하여 정의하고 정의된 탐지 대상에 따라 더욱 세분화되고 안전한 방어 기법을 제안하였다. 또한, 여러 가지 실험에 의한 안전성 검증과 접근제어라는 새로운 단계를 설정하여 보안강도를 높이도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대하여 알아보고, 3장에서는 시스템의 설계 및 구현에 대해 기술한다. 4장에서는 규칙기반 다단계 침입시스템에 대한 단계적 실험내용을 살펴보고, 5장에서는 결론을 짓고 향후 연구에 대해 언급한다.

2. 관련연구

2.1 Signature-based IDS (pattern-matching IDS : 오용 침입 탐지)

잘 알려진 공격 유형(a known attack type)에 대응하는 패턴을 찾기 위해 단순한 패턴 매칭을 수행하고 상황을 보고한다. 그러나 시그너처를 수집하고 관리하는 방법 자체가 문제가 되는 단점이 있다. 또, 시그너처 DB에 없는 새로운 유형의 공격에는 무방비하다[5].

2.2 Heuristic Intrusion Detections(이상 침입 탐지)

기본적으로 정상에서 벗어난 행위(behavior)를 찾고, 정상 및 비정상 행위를 찾은 다음, 정상 및 비정상 행위를 이해하는데 도움이 되는 특성을 3가지 부류인 normal, suspicious, unknown중의 하나로 분류한다. IDS가 학습하면서 어떤 동작들이 수용 가능한지 아닌지 구분하게 되면, 특정 동작들은 위 부류 중의 하나에서 다른 부류로 이동된다. 패턴 매칭 기법에서처럼, heuristic intrusion detection도 해당 시스템이 유지하는 정보(동작들이 올바른 부류로 분류되었는지의 정보)의 양 및 현재 동작들이 각 분류에 적합한 정도에 따라 제한을 받는다[4,6].

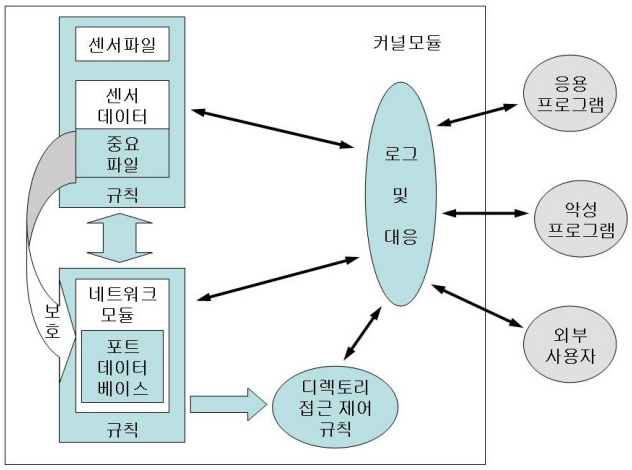
2.3 specification-based detection(규칙기반탐지)

규칙 기반을 둔 탐지에서 위험한 객체들의 정확한 동작은 객체들의 잘 만들어진 보안 규칙들로서 수동으로 추출하고, 실제 동작과 비교한다. 침입은 보통 비정상적인 동작 객체에서 일으킨다. 그에 관한 정확한 지식 없이 탐지 할 수 있다. 규칙기반을 둔 탐지는 비정상 탐지에서 합법이지만 보이지 않는 동작에 기인하는 오경보의 높은 비율을 극복한다. 그러나 상세한 규칙의 개발은 시간낭비이다[7,8].

3. 시스템 설계 및 구현

3.1 시스템 구성

본 논문에서는 리눅스 시스템에서 적은 오버헤드로 다양한 공격 유형에 좀 더 효과적으로 대응할 수 있는 "다단계 구조를 가진 침입 탐지 및 방어 시스템의 구현"에서 제안된 시스템을 확장한다. 제안한 시스템의 구성은 (그림 1)과 같다.



(그림 1) 시스템 구성도

알려지지 않은 유형의 이상침입을 탐지하기 위해 주요 디렉토리와 파일을 센서파일과 센서 데이터로 설정하여 센서의 접근을 제어한다. 이에 추가로 네트워크 포트 관리 모듈이 통합되어 있어, 포트가 할당될 때마다 관련 정보를 로그로 남기며, 악의적인 프로그램에서 사용되는 포트를 차단한다. 이와 같은 방법으로 악성프로그램이 주요 파일을 불법적으로 접근할 때, 효과적으로 대응, 예방할 수 있다.

(그림 1)의 포트 데이터베이스에는 알려진 악성 프로그램들이 사용하는 비정상 포트번호(공격자가 주로 사용하는 포트)와 정상포트 번호들을 분류하여 관리하며, 데이터베이스 또한 수정/삭제되지 말아야 하므로 센서 데이터에 보호를 받도록 구성하였다[3]. 이에 추가로 접근제어 규칙을 새로 구성하여 어떤 규칙으로 접근 제한을 받는다.

본 논문에서는 로그 기록을 위해 리눅스 시스템 로그를 사용하고 있으며, 7단계별 위험수준에 따른 기록이 가능하므로 관리자 설정에 따라 기록을 설정할 수 있다.

3.2 보안정책 및 규칙

본 논문에서는 다음과 같은 보안 정책을 사용하여 단계적 규칙을 책정하고 침입을 탐지 및 차단한다.

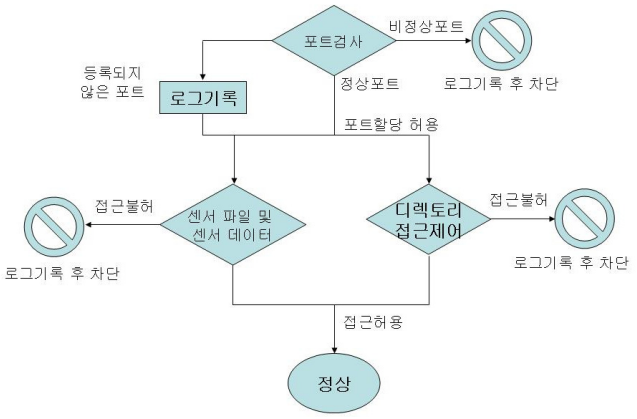
- 센서 객체는 센서 파일이나 센서 데이터를 말한다.
- 센서 객체를 설치하거나 제거, 변경은 시스템 관리자만 실행할 수 있는 특정 명령(프로세스)에 의해서만 가능함.
- 특정 명령을 제외한 어떤 명령도 센서 파일을 접근할 수 없다.
- 센서 데이터는 호스트 내부 또는 Intranet 내에서는 접근 가능하나, 외부 Intranet으로 유출되지 않아야 한다. (즉, 센서 데이터가 설치된 파일은 네트워크를 통해 외부에서 접근될 수 없다.)
- 센서 객체로의 모든 접근은 로그로 기록된다.
- 센서 객체의 명칭 및 개수는 시스템 및 관리자에 따라 다르게 설정될 수 있다.
- 센서 객체의 명칭 및 개수는 동적으로 변경 가능하다.
- 공격이 가능한 비정상 포트에 대한 접근을 차단한다.
- 정상포트에 대해 별도의 접근제어 규칙을 적용하여 허가되지 않은 데이터에 대한 접근을 차단한다.
- 정상포트에 대해 센서파일로 정의되지 않은 다른 디렉토리 보호 목적으로 사용되는 디렉토리 이외의 접근, 유출을 차단한다.
- 포트에 대한 모든 접근에 대하여 접근기록을 로그로 기록한다.

3.3 규칙 기반의 접근 제어

3.2의 보안 정책에 기반 하여, 시스템은 다음의 action을 침입으로 판단한다.

- 특정 명령이 아닌 다른 명령/프로세스가 센서 파일을 접근하는 경우는 침입으로 간주하고, 이는 어떤 악의적인 프로그램이 critical directory 내에 있는 파일들을 무작위로 유출하려고 할 때 유용하다.
- 센서 데이터가 external network, or Internet으로 유출되는 경우는 침입으로 간주하며, 어떤 악의적

인 프로그램이 특정 secret file만을 선별하여 유출하려고 할 때 유용하다.
- 비정상 포트의 할당 요청 시 침입으로 간주하여, 로그를 남기고 차단시킨다. 트로이 목마, 백 도어, 웜과 같은 악의적인 프로그램의 차단에 유용하다.
- 등록되지 않은 새로운 포트의 할당은 잠정적인 위협으로 간주하고 로그를 남긴다. 포트할당으로 접근을 허용하면 다른 접근제어 규칙에 제한을 받게 된다.
- 정상포트는 할당을 허용하고, 정상적인 디렉토리의 접근은 허용하나 인가되지 않은 디렉토리의 접근은 차단한다.



(그림 2) 단계별 보안 정책

본 논문에서는 (그림 2)와 같이 단계별 보안 정책을 적용하여 규칙기반의 침입탐지를 수행한다.

3.4 시스템 구현

본 시스템을 구현하기 위한 환경으로 펜티엄4 1.7Ghz, Linux Redhat 9 시스템에서 실험하였다. LKM으로 구현되었으며, 커널 버전은 2.4.22를 사용하였다. 시스템 콜 가로채기(hooking)방법을 이용하였다.

각 프로세스의 센서 접근을 감시하고 정보의 접근제어를 위해 sys_open(), sys_read(), sys_close(), sys_fork(), sys_unlink() 등의 커널 내부 함수를 수정하였으며, “센서”를 접근했을 때 로그를 남기게 된다. 이 때 로그는 실행된 명령의 절대경로, PID, 사용자ID, 접근시간 등을 기록한다. “센서파일”은 루트권한의 프로세스라도 접근을 했다면 로그를 기록한 다음, 인가된 프로세스가 아니면 해당 프로세스를 종료시킨다. 센서데이터가 접근되면, 해당 프로세스 정보를 로그로 남기고 네트워크 기반 침입탐지 모듈과 협력하여 대응한다.

서버에 접근하기 위해서는 포트 할당이 필요하다. 서버에서 포트 할당을 차단하기 위해 sys_bind() 함수를 수정하였다. TCP와 UDP 모두 sys_bind()를 사용하므로 sys_socketcall()을 후킹하여 사용하였다.

포트 데이터베이스 파일은 커널 수준에서 관리되며, 정상적인 포트번호와 알려진 공격에 사용되고 있는 포트번호들을 동적으로 유지한다. 데이터베이스에 등록되어 있지 않은 포트 번호가 요청될 경우, 잠정적인 위협으로 간주하고 로그를 남겨 관리자에 통보하도록 구성되어 있다.

허가된 정상포트를 통해 디렉토리 접근을 하고자 할 경우 해당 서비스에서 허용하는 디렉토리가 아닌 임의의 허가되지 않은 디렉토리에 대한 접근을 차단하는 기능을 디렉토리 접근제어라고 하여 구현 중에 있다.

4. 실험

4.1 실험개요

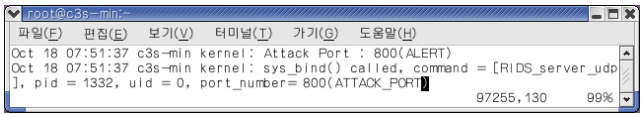
본 논문에서의 실험은 중요한 몇 가지 포트기반의 접근과 센서기반 침입 탐지 모듈과 네트워크 포트기반 모듈의 연동으로 접근제어 실험 중심으로 실험하였다. SIDS에서 센서파일과 센서데이터를 설치한 후 네트워크 포트기반 모듈에서 비정상포트와 정상포트를 등록하여 실험하였다.

4.2 실험내용

다음 실험을 위해 2개의 프로그램(RIDS_server_udp, RIDS_client_udp)을 작성하였다. RIDS_server_udp는 서버 프로그램으로 udp로 포트를 할당하고 client의 요청을 기다리는 프로그램이며. RIDS_client_udp 파일은 udp로 파일을 유출하는 프로그램이다. 네트워크 비정상 포트 할당과 센서 객체 접근제어 2가지 방법으로 실험하였다. 여기서 비정상 포트번호로 800번을 등록하였고, 500번 포트는 등록되지 않은 새로운 포트를 의미한다.

4.2.1 네트워크 비정상 포트 실험

다음은 네트워크 포트 기반의 할당 제어 실험이다.



(그림 3) 비정상 포트 할당 시도 시 시스템 로그

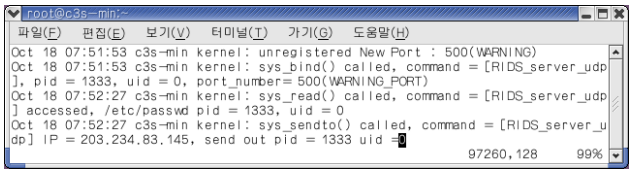
(그림 3)은 비정상 포트(800)번호로 포트할당을 시도 하였을 때 남겨진 로그파일이다. 로그는 시간, 호출된 시스템콜 명, process, pid, uid, 포트번호(포트유형)가 기록된다. 등록되지 않은 새로운 포트의 접근은 로그를 (그림 4)와 같이 로그를 남기고 포트 할당을 허용한다. 정상 포트 할당 시에는 불허함으로써 접속이 종료된다. 이 실험은 주로 트로이목마, 백 도어, 웜과 같은 악성 코드 방어에 대해 효과적이다.

4.2.2 센서 객체 접근제어 실험

4.2.1의 실험과 같이 네트워크 포트기반 모듈은 잘 알려진 비정상포트를 미리 정의하여 네트워크 연

결 제한으로 접근을 제어한다. 이 방법은 일반적인 방화벽과 차이점은 내부에서 외부로의 접속 시도 차단, SIDS와의 연동으로 접근제어, 접근제어의 확장 등 다양한 상황에서 예방, 대응할 수 있다.

본 시스템에서는 새로운 포트(등록되지 않은)번호로 접근 시도 시 잠정적인 위협으로 간주하고 로그를 남기게 되고 접근을 허용하고 있다. 등록되지 않은 포트(500)번호로 접속하여 센서 데이터(/etc/passwd)로 정의된 파일을 유출하는 네트워크 프로그램을 작성하여 실험해 보았다.



(그림 4) 새로운 포트 할당으로 센서 데이터 유출 시 시스템 로그

(그림 4)는 새로운 포트로 접속하여 센서 데이터로 지정된 파일을 유출 시도하여 시스템 로그에 기록된 화면이다. sys_socketcall(BIND)는 포트할당에 대한 로그이다. 시간, 시스템 콜 명, process, pid, uid, port number가 기록된다. sys_read()는 센서 데이터에 대한 접근 기록이며 시간, 시스템 콜 명, process, sensor data file name, pid, uid가 기록된다. sys_socketcall(SENDTO)는 센서 데이터가 유출 시도에 대한 로그이다. 시간, 시스템 콜 명, process, IP, pid, uid가 기록된다. 이번 실험과 같이 새로운 포트로 센서 데이터를 유출 시도 하였을 경우, 관리자는 이 로그를 분석하여 정상포트, 비정상 포트로 분류한다. 이러한 센서 데이터의 접근은 남겨진 데이터를 이용하여 공격 경로를 추적, 포렌식에 이용될 수 있다.

4.2.3 성능평가

본 시스템에서 침입탐지 및 방어 모듈로 인해 유발되는 성능 상의 성능평가는 <표 1>과 같다. 센서기반 침입 탐지 시스템의 성능평가는 실험환경과 동일한 시스템에서 측정하였으며, 네트워크 포트기반 탐지 모듈의 경우 10000개의 포트가 정의된 시스템에서 평균 bind함수 수행시간은 약30us 이내로 약간의 성능저하를 보인다.

<표 1> 네트워크 포트기반 탐지 모듈 평균시간

시간 \ 포트갯수	1000개	10000개
평균 bind함수 완료시간(us)	19us	27us
원래 시스템에서 bind()함수 완료시간(us)	4us	4us
오버헤드(us)	15us	23us

5. 결론

본 논문에서는 보안정책과 규칙을 기반으로 하는 다단계 침입 탐지 시스템의 설계 및 구현을 보였으며, 센서기반 침입탐지시스템과 네트워크 포트기반 오용침입 탐지와의 연동으로 접근제어와 실험을 통한 실제방어 기법에 대해 기술하였다. 새로운 정책을 적용한 단계를 추가해 나감으로서 향후 점차 발전하는 해킹기술에 효과적으로 대응하며 최소한의 성능저하로 탐지의 범위를 늘려 탐지 율을 높이는 방법에 대한 연구가 수행되어야 할 것이다. 또한 무선 환경에서의 구현방법도 연구할 계획이다.

참고문헌

[1] 장철연, 조성제, 최종무, “LKM을 이용한 센서기반 침입 탐지 및 보호 시스템, 한국정보과학회 2003 가을 학술발표논문집(I), 제 30권 2호, pp.694-696, 2003. 10.

[2] 장철연, 조성제, “LKM을 이용한 센서기반 침입 탐지 시스템”, 단국대 석사학위논문, 2003.

[3] 민욱기, 장혜영, 최종천, 조성제, “다단계 구조를 가진 침입 탐지 및 방어 시스템의 구현”, 한국정보과학회 2005 한국컴퓨터종합학술대회 논문집(A), pp.136-138, 2005. 7.

[4] Understanding Heuristics: Symantec's Bloodhound Technology, Symantec White Paper Series Volume XXXIV

[5] S. Kumar and E.H. Spafford. "A Pattern Matching Model for Misuse Intrusion Detection", Proceedings of the 17th National Computer Security Conference, 1994.

[6] Salvatore J. Stolfo, et al., "Anomaly Detection in Computer Security and an Application to File System Accesses", Proceedings of 15th International Symposium of Foundations of Intelligent Systems, 2005.

[7] R. Sekar, et al., "Intrusion Detection: Specification-based anomaly detection: a new approach for detecting network intrusions", Proceedings of the 9th ACM conference on Computer and communications security, 2002.

[8] R. Sekar and P.Uppuluri, "Synthesizing Fast Intrusion Prevention/Detection Systems from High-Level Specifications", Proceedings of USENIX Security Symposium, 1999.