내부 센서를 이용한 침입 탐지 시스템에 관한 연구

장정숙, 전용희

대구가톨릭대학교 컴퓨터 · 정보통신 공학부

A Study on the Intrusion Detection System Using Internal Sensors

Jung-Suk, Jang, Yong-Hee, Jeon

School of Computer and Information Communications Engineering,
Catholic University of Daegu

요 약

효율적인 네트워크의 보호를 위해 네트워크를 경유한 공격에 대하여 알려진 공격과 새로운 공격에 대한 빠른 탐지와 적절한 대응을 할 수 있는 침입 탐지 시스템(Intrusion Detection System: IDS)에 대하여 최근 관심이 증대되고 있다. 기존의 침입 탐지 시스템들은 다양한 침입에 대한 능동적인 탐지에 어려움이 있다. 본 논문에서는 기존의 침입 탐지 시스템이 가지고 있는, 성능(fidelity) 문제, 자원 사용 문제 및 신뢰성 문제를 해결하기 위하여, 호스트에서 내부 센서를 사용하는 메커니즘에 대하여 고찰하고 분석한다. 그리고 침입 탐지 프레임워크를 구축하기 위한 내부 센서의 개념, 특징 및 능력에 대하여 기술한다.

Ⅰ. 서론

기존의 침입 탐지 시스템들은 다양한 침입을 능동적으로 탐지하는데 어려움이 많다. 향상된 침입 탐지는 나아가 외부적인 침입뿐만 아니라 내부적인 침입까지 효율적인 탐지가 가능한 구조를 가져야한다. 최근 몇 년 동안에 IDS는 상업적인 그리고 교육적인 분야에서도 활발히 전개되고 있다. 기존의 IDS의 대부분은 집중적 데이터 분석 엔진 혹은 호스트 당 데이터 수집과 분석 컴포넌트를 사용함으로써 확장성, 신뢰성, 그리고 공격에 대한 저항을 제한하는 문제를 가진다. 이를 해결하기 위한 방안의 하나로 소스 코드 계측(instrumentation) 기반의 내부 센서(internal sensors)로 불리는 모니터링 기술을 분석한다.

이 기술을 통하여 프로그램 안에서 데이터와 행위의 밀접한 관찰을 할 수 있다. 이는 거의 실시간으로 수행되는 침입 탐지 시스템을 구현하기 위하여 또한 사용될 수 있으며, 그리하여 공격에 저항적이고, 메모리와 CPU 이용률 측면에서 호스트에서 상당히 낮은 오버헤드를 부과한다. 본 논문에서는 호스트 레벨에서의 침입 탐지 프레임워크를 구축하기 위한 내부적 센서를 사용하는 개념을 기술하고 그들의 구조와 특성을 분석한다.

본 논문의 나머지는 다음과 같다. 제 II 절에서는 침입 탐지의 데이터 수집 구조에 대하여 기술하고, 제 III 절에서는 침입 탐지의 데이터 수집 메커니즘, Ⅳ절에서는 내부 센서 기반 침입 탐지 구조에 대하여 기술하며, 마지막 Ⅴ절에서는 맺음

말로 본 논문의 끝을 맺는다.

Ⅱ. 침입탐지의 데이터 수집구조

컴퓨터 시스템 자원의 무결성(integrity), 기밀성(confidentiality), 가용성(availability)을 위협하는 시도를 침입(intrusion)이라 하고, 그러한 행위를 식별하는 것을 침입 탐지(Intrusion Detection)라 하며, 침입 탐지를 수행하는 컴퓨터 시스템을 침입탐지시스템(Intrusion Detection System)이라 한다. IDS가 요구하는 일반적인 특성은 다음과 같다[1]:

- 최소한의 감독으로 연속적인 실행
- 결점 허용성
- 전복(subversion)에 대한 저항
- 시스템에 최소 오버헤드 부과
- 구성 가능성
- 배치의 용이성
- 시스템 변화에 대한 적응성
- 공격 탐지 가능성

재래의 운영체제에 의해 제공되는 감사 데이터(audit data)는 오용 탐지(misuse detection)를 위해 유용한 내용이 부족하다고 알려져 있다. 데이터가 어떤 방법으로 얻어 지는가가 침입 탐지 시스템의 개발에 중요한 설계 사항이다. 따라서 침입 탐지를 위하여 적합한 데이터 수집구조와 데이터 수집 메커니즘에 대하여 기술할 필요가 있다.

본 논문에서는 데이터 수집 구조를 집중(centralized) 형태와 분산(distributed) 형태로 분류하고, 집중형 침입 탐지 시스템과 분산형 침입 탐지 시스템으로 나누어 기술한다. 일반적으로 침입 탐지를 위한 데이터 수집 구조 분류는 데이터를 수집하는 위치에 따라 분류되며, 다음과 같다[2]:

- 집중형 침입 탐지 시스템(centralized IDS)

침입 탐지 시스템에 사용하는 데이터의 분석은 다수의 위치에서 수행되고, 위치의 수는 고정이고 모니터 되는 컴포넌트 수와는 독립적이다.

- 분산형 침입 탐지 시스템(distributed IDS)

침입 탐지 시스템에 사용하는 데이터의 분석은 다수의 위치에서 수행되며, 위치의 수는 모니터 되는 컴포넌트의 수에 직접 비례한다.

이 정의는 모니터 되는 컴포넌트의 수에 기반 한 것이다. 여기서, 위치(location)는 수행 코드의 인스턴스(instance)로 정의된다. 분산 및 집중 침입 탐지 시스템 모두 호스트 혹은 네트워크 기반 데이터 수집 방법, 혹은 그것들의 결합을 사용할 수 있다. 위의 두 방법들은 신뢰성, 결함 허용성, 부가된 오버 헤드, 확장성, 서비스의 우아한 종료, 역동적 구성에 대하여 각각 다른 특성들을 가지게 된다.

Ⅲ. 침입탐지의 데이터 수집 메커니즘

1. 직접 모니터링과 간접 모니터링

침입 탐지 시스템이 물리적 현상으로부터 직접 데이터를 획득하여 모니터 되는 컴포넌트에서 조건 혹은 행위를 측정 할 때 이것을 직접 모니터링(direct monitoring)이라 하고, 침입 탐지 시스템이 정보를 획득하기 위해서 분리 메커니즘 혹은 도구에 의존 할 때 이것을 간접 모니터링(indirect monitoring)이라 한다. 즉, 직접 모니터링은 객체의 특성을 측정 혹은 관찰하는 것이고, 간접 모니터링은 그 특성을 가지는 객체의 영향을 측정하거나 관찰하는 것이다. 예를 들면, 유닉스 호스트에서 CPU 부하를 관찰하는 ps 명령어의 사용은 직접 모니터링으로 고려된다. 그것은 ps가 커널에서의 해당 데이터 구조로부터 부하 데이터를 추출하기 때문이다. 또한 만약 CPU 부하가 로그 파일에서 기록되어 후에 읽혀진다면, 이것은 간접 모니터링으로 고려한다. 그것은 관찰을 위하여 분리 메커니즘(로그 파일, 네트워크 패킷)에 의존하기 때문이다. 이와 같이, 데이터 수집 방법에 따라 직접 혹은 간접 모니터링으로 분류된다.

직접 모니터링은 신뢰성(reliability), 완전성(completeness), 부피(volume), 확장성(scalability), 그리고 적시성(timeliness)에서 간접 모니터링보다 더 나은 탐지 능력을 보인다. 그러나, 직접 모니터링은 생성 정보의 형태와 모니터 되는 컴포넌트에 대하여 보다 특정한 방법에서 설계되어야 하는 메커니즘으로 구현이 복잡하다는 단점을 가진다.

2. 호스트 기반과 네트워크 기반

실제적으로, 데이터 수집 방법은 다음 정의에 의하여, 일반적으로 호스트 기반 혹은 네트워크 기반으로 분류된다.

- 호스트 기반(host-based) 데이터 수집

시스템의 상태, 메모리의 내용, 혹은 로그파일을 호스트에 상주하는 소스로부터 데이터를 획득하는 것.

- 네트워크 기반(network-based) 데이터 수집

패킷이 네트워크를 통하여 지나갈 때 패킷을 포획함으로써 데이터를 획득하는 것.

호스트 기반 데이터 수집이 타당한 이유로는 다음과 같은 것이 있다[3]: 정확한 데이터 수집, 발생하는 모든 사건에 대한 보고가능, 네트워크 기반에서 발생하는 삽입 및 속임수(evasion) 공격 문제가 없음, 데이터 수집과 통일에 대한 문제가 없음, 그리고 호스트 내부의 활동(action)을 관찰 가능. 네트워크 기반 데이터 수집은 기존의 네트

워크에 IDS가 전개되므로 호스트에 아무런 변경을 주지 않는다. 이런 이유로 많은 상용 침입 탐지 시스템은 네트워크 기반 데이터 수집을 사용한다. 또한 다른 호스트에서 완전히 보이지 않음으로써, 네트워크 상에서의 행동을 관찰하기 위한 편리한 장점을 제공한다.

"네트워크 기반"으로 통상 간주되는 침입탐지 시스템은 간접/네트워크 기반 모니터링 메커니즘에 해당하며, 간접/호스트 기반과 모든 직접 모니터링 메커니즘은 "호스트 기반" IDS에 해당한다. 최근에는 완전한 모니터링을 위해서 호스트 기반과 네트워크 기반 컴포넌트 둘 다를 사용하는 추세이다.

3. 외부 센서와 내부 센서

모든 직접 모니터링 방법은 호스트 기반이다. 호스트의 직접 모니터링은 다음 정의에 따라 외부 혹은 내부 센서를 사용하여 수행된다[1].

• 외부 센서(external sensor)

호스트에서, 한 조각의 소프트웨어가 컴포넌트(하드웨어, 소프트웨어)를 관찰하여 IDS에게 유용한 데이터를 보고한다. 컴포넌트와 코드는 분리하여 구현된다.

• 내부 센서(internal sensor)

호스트에서, 한 조각의 소프트웨어가 컴포넌트(하드웨어, 소프트웨어)를 관찰하여 IDS에게 유용한 데이터를 보고한다. 외부 센서와 다른 점은 컴포넌트와 코드는 통합하여 구현된다.

내부센서는 소프트웨어 혹은 하드웨어 컴포넌트 안에 구축될 수 있다. 예를 들면, Unix 커널 안에 내장된 프로세스-정보를 수집하는 컴포넌트일 수도 있고 혹은 네트워크 인터페이스 카드의 펌웨어에서처럼 하드웨어 컴포넌트 안에도 구축이 가능하다. 내부 센서는 모니터 되는 컴포넌트의 소스 코드의 부분이다. 내부센서는 이미 존재하는 프로그램에 추가될 수 있고, 이 경우 소스 코드 계측의 경우로 고려된다. 이상적으로, 내부 센서는 변경을 하고 에러를 고치는 비용과 노력이 적게 드는 프로그램의 개발동안 추가되어야 한다. 프로그램의 어떤 부분도 IDS에 의하여 사용되는 데이터를 제공하는 한 내부 센서로 고려된다. 직접 데이터 수집을 위한 외부 센서와 내부 센서는 다른 강점과 약점을 가진다. 그러므로, 침입 탐지 시스템에서 함께 사용될 수 있다. 다음 표 1과 표 2는 내부 센서와 외부 센서의 장단점을 기술한 것이다[1].

표 1: 내부 센서의 장점과 단점.

내부 센서	
장점	단점
• 정보의 생성과 사용 사이에서 최소의 지연 • 침입자의 추적을 은폐하기 위한 데이터 수정이 불가능 • 분리 처리되지 않기 때문에 쉽게 불능화 혹은 수정 불가능 • 네트워크 트래픽과 처리 부하의 감소 • 단일 호스트에 많은 센서를 반영 가능 • 센서는 모니터링하는 프로그램의 부분으로 구현되기 때문에, 필요한 어떤 정보에도 접근 가능	• 구현은 모니터 되는 프로그램 소스 코드에 접근이 요구됨 • 모니터 되는 프로그램에 대하여 수정을 요구하기 때문에, 구현이 어려움 • 모니터 할 프로그램과 같은 언어에서 구현 될 필요 • 만약 부정확하게 구현 혹은 설계된다면, 속하는 프로그램의 기능 혹은 성능을 심하게 손상 가능 • 다른 운영체제에서 갱신, 수정 그리고 이식이 어렵고, 혹은 같은 프로그램의 다른 버전에서조차 어렵다 • 감소된 이식성

표 2: 외부 센서의 장점과 단점.

외부 센서	
장점	단점
• 호스트로 부터 쉬운 변경, 추가 혹은 삭제 • 어떤 프로그래밍 언어에서도 구현 가능	• 데이터의 생성과 사용 사이에는 지연이 존재 • 센서가 정보를 획득하기 전에 침입자에 의해 변경 가능 • 침입자에 의해 불능화 혹은 수정 가능. • 연속적으로 수행되기 때문에 성능에 영향 • 정보에 대한 제한된 접근(유닉스 명령어, 시스템 호출)

Ⅳ. 내부센서 기반 침입탐지 구조

내부 센서는 호스트 모니터링과 침입 탐지를 수행하기 위해서 적절한 다수의 특성을 가진다. 이 절에서는 침입 탐지를 위한 주요 특징과 구조를 기술한다. 먼저 주요 특징은 다음과 같다:

• 내부 센서는 주요한 데이터 수집 컴포넌트이다.
• 임베디드(내장) 탐지기의 사용을 통하여 분산, 지역 데이터 감소를 제공한다.
• 데이터 처리와 더 높은-단계 동작을 필요로 할 때, 외부 센서의 존재를 또한 고려한다.

1. 임베디드(내장) 탐지기

ESP(Embedded Sensors Project)라고 불리는 침입 탐지 구조는 지역 데이터 감소를 위한 메커니즘으로 임베디드 탐지기(embedded detector)를

사용한다[4].

• 임베디드 탐지기(Embedded detector)

내부 센서로서, 특정한 공격을 검사하고 그들의 발생을 보고한다.

임베디드 탐지기는 공격 순간에 이용 가능한 데이터의 사용으로, 공격이 탐지되는 위치에서 코드 내에 존재해야한다. 만약 정확하게 구현된다면, 탐지기는 단순한 검사 수행으로 공격이 발생하고 있는지를 결정할 수 있다. 임베디드 탐지기들의 탐지 능력으로 지역 데이터 감소를 수행하기 위한 메커니즘이다. 이것은 시스템에서 생성된 데이터 양의 감소를 위해 분산 IDS에서 특히 중요하다.

2. 내부센서와 임베디드 탐지기와의 관계

내부 센서와 임베디드 탐지기의 차이점은 센서가 프로그램 안에서 임의의 조건을 관찰하고 이것의 현재 상태 혹은 값을 보고한다는 것이다. 반면에 탐지기는 공격의 구체적인 흔적을 찾는다(그림 1 참조). 임베디드 탐지기들은 내부 센서들의 특별한 형태이다.

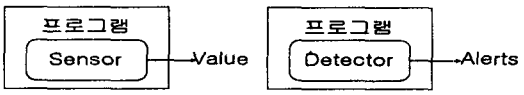

그림 1: 센서와 탐지기의 생성 차이

개념적으로, 임베디드 탐지기는 공격을 탐지하는 논리가 첨가된 내부 센서로 생각할 수 있다(그림 2 참조). 어떤 경우에, 내부 센서는 코드 안에서 명확하게 차이가 난다. 예를 들어, 센서에서 포트 스캔을 위한 탐지는 포트에 대한 연결을 추적하고 그리고 그들의 수와 소스들을 보고한다.

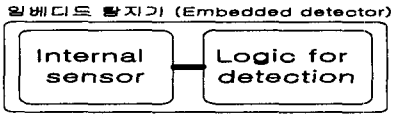

그림 2: 임베디드 탐지기의 개념적 구조

다른 경우에, 내부 센서는 임베디드 탐지기에서 묵시적으로 구축되고 그리고 이것의 값은 즉시 결정을 취하는데 사용된다. 예를 들면, Ping-of-death 공격을 위한 탐지기는 어떤 임계값과 한 변수를 비교하여, 핑 패킷의 크기를 검사할 수 있고, 만약 값이 크다면 경고(alert)를 발생한다. 이 경우에 개념적인 "센서"는 그 변수 값을 읽는 행위일 것이고, 그리고 "탐지기" 부분은 임계값에 대하여 그 값을 비교하는 것이다. 임베디드 탐지기의 데이터 수집 부분과 데이터 분석 부분과의 차이는 실제 클 수 있다. 그림 3(a)에서는 하나의 센서가 다중 탐지기들에게 다른 공격을 조사하도록 데이터를 제공하는 것을 보여주고, 그림 3(b)는 다중 센서들이 단일 탐지기에게 데이터를 제공하는 것을 보여준다.

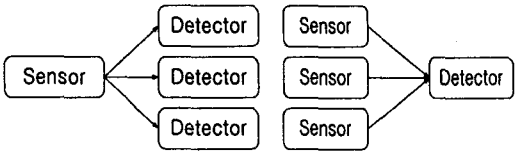

그림 3(a): 하나의 센서가 다중의 탐지기에 데이터 제공. 그림 3(b): 다중의 센서가 단일 탐지기에 데이터 제공.

그림 3: 센서와 탐지기사이의 데이터 제공

침입 탐지 과정의 데이터 수집과 데이터 처리 단계는 가능한 한 시간과 공간에서 함께 밀접해야 한다. 따라서, 대부분 임베디드 탐지기들은 센서와 탐지기가 강하게 함께 결합된 형태여야 한다.

3. 임베디드 탐지기의 강점과 약점

임베디드 탐지기는 탐지기가 수행되는 호스트에 실제 존재하는 취약성과는 독립적으로, 취약성 공격에 대한 탐지가 가능하다. 그림 4는 임베디드 탐지기 삽입 전과 후 버퍼 오버플로우에 취약하지 않는 코드의 예를 보여준다[5].

```
1  char buf[256];                    1 char buf[256];
2 strcpy(buf,getenv("HOME"),         2 {
        sizeof(buf));                3 if (strlen(genenv("HOME"))>255) {
                                     4  log_alert ("buffer overflow");
                                     5   }
                                     6 }
                                     7  strcpy(buf, getenv("HOME"),
                                             size(buf));
```

탐지기 삽입 전 코드 탐지기 삽입 후 코드

그림 4: 임베디드 탐지기 이용 버퍼 오버플로우 코드

그림 4의 코드가 버퍼 오버플로우에 취약하지 않는 것은 strncpy 함수가 사용되기 때문이다. 탐지기는 그림 4의 오른쪽에서 보여주는 것처럼 2-6 줄이 코드에 추가된다. 이 예는 탐지기가 조사하는 공격에 취약하지 않는 코드에까지 임베디드 탐지기가 존재할 수 있다는 것을 보여준다. 임베디드 탐지기의 다른 장점은 그들이 모니터 되는 컴포넌트(예를 들면, 만약 프로그램이 악성 행위를 이미 보았다면)에 존재하는 방어 메커니즘을 사용할 수 있고 이것들을 탐지와 결합하여 사용 가능하다 것이다.

V. 맺음말

침입 탐지 시스템 설계는 탐지를 위한 알고리즘과 구조와 함께 변화되어 왔다. 호스트-기반으로 시작하여, 네트워크 기반으로 진화되었으며, 최근에는 두 개를 혼합한 분산 형태에 대한 관심이 증대되고 있다. 이런 변화에서도, 침입 탐지 시스템에 의하여 사용되는 정보 소스는 감사 기록과 네트워크 트래픽으로 남아 있다. 대부분 기존 침입 탐지 시스템은 간접 모니터링으로 간주되는 시스템의 행위를 반영하고 있다. 이들 데이터 소스들은 그들이 제공하는 데이터의 적시성, 완전성, 정확성, 및 확장성에 제한을 가지고 있다. 또한 서비스 거부를 포함하여 몇 가지 측면에서 공격에 대하여 취약하게 만든다.

이 논문에서는 프로그램 코드로서, 내부 센서 사용을 기반으로 하는 구조를 고찰하고 기술하였다. 침입 탐지 시스템을 구축하기 위한 임베디드 탐지기의 응용에 대하여도 분석하였다. 이와 같은 방법을 통하여 기존의 침입 탐지 시스템이 가지고 있는 문제점을 다소 해결할 수 있고, 침입 탐지 시스템이 요구하는 특성을 다수 만족시킬 수 있음을 알 수 있었다.

또한 본 논문에서는 침입 탐지 시스템의 내부 센서의 구조와 특성에 대해 기술했으며, 그리고 임베디드 탐지기들을 위한 데이터 소스 형태의 분류에 대하여 알아보았다.

앞으로, 국내에서도 침입 탐지에 센서를 이용한 구조적 프레임워크에 대한 연구가 필요하다고 하겠다.

※참고문헌

[1] Diego Martin Zamboni, Using Internal Sensors for Computer Intrusion Detection, Ph. D. dissertation, Purdue University, CERIAS TR 2001-42, August 2001.

[2] Eugene H. Spafford and Diego Zamboni, "Intrusion detection using autonomous agents", Computer Networks, 34(4), pp.547-570, Oct. 2000.

[3] Thomas E. Daniels and Eugene H. Spafford. "Identification of host audit data to detect attacks on low-level IP vulnerabilities", Journal of Computer Security, 7(1), pp.3-35, 1999.

[4] Eugene Spafford and Diego Zamboni. Data collection mechanisms for intrusion detection systems. CERIAS Technical Report 2000-08, CERIAS, Purdue University, 1315 Recitation building, West Lafayette, IN, June 2000.

[5] Florian Kerschbaum, Eugene H. Spafford, and Diego Zamboni. "Using embedded sensors for detecting network attacks", In Proceedings of the 1st ACM Workshop on Intrusion Detection Systems. ACM SIGSAC, November 2000.