침입탐지 시스템을 위한 효율적인 Audit 시스템 설계 및 구현

문제근, 위규범
아주대학교 정보통신 전문대학원
(jkmoon, kbwee)@ajou.ac.kr

The Design and Implementation of Efficient Audit System for Intrusion Detection System

Je-Keun Moon, Kyu-Bum Wee
Graduate School of Information and Communication, Ajou University

요 약

침입탐지 시스템은 Audit 시스템, 침입탐지 엔진, 침입대응 시스템으로 나눌 수 있다. Audit 시스템을 통하여 감사자료를 수집하고, 침입탐지 엔진을 이용하여 침입을 탐지하며, 대응 시스템을 통하여 침입에 대한 대응 및 대책을 세우게 된다. 지금까지 Audit 시스템은 침입탐지 엔진의 일부로 설계, 구현되어 왔다. 이는 침입탐지 엔진에 따라 Audit 시스템도 함께 변경되어야 하는 비효율성을 가져왔다. 본 논문에서는 침입탐지 엔진과 독립적으로 Audit data를 수집하고 관리하며 침입탐지 엔진과 효율적으로 연동될 수 있는 Audit 시스템을 설계 및 구현하였다. 탐지엔진을 위한 알고리즘 개발 및 엔진의 변경시에 본 시스템과 연동하면 별다른 Audit 시스템 구현없이 효율적이고 빠르게 테스트 및 검증할 수 있다.

1. 서 론

침입 탐지가 보안 분야의 중요한 이슈가 되면서 이에 대한 많은 연구가 진행중이며 주로 빠르고 정확한 침입탐지 알고리즘 개발에 많은 연구가 이루어져 왔고 다양한 알고리즘 및 방법론이 제기되고 있다. 그러나 제안되고 있는 많은 방법론들은 대부분 기본적으로 네트웍 패킷[1] 이나 호스트 시스템의 시스템 콜, 프로세스 ID와 같은 Audit 데이터를 기반으로 하고 있기 때문에 이러한 알고리즘을 적용하여 테스트하기 위해서는 필요한 Audit 시스템을 먼저 구축하는 것이 필요했다. 이는 제안된 침입탐지 방법론들을 테스트하기 위해 Audit 시스템을 위해 많은 시간을 소비해야 한다는 단점이 있었다. 또한 대부분 Audit 시스템이 탐지 엔진에 포함되어 있거나 탐지 엔진에 맞게 설계, 구현됨으로써 엔진의 업데이트 및 변경이 있을 시에 새로 구현되어야 한다는 단점이 있었다. 따라서 침입탐지 엔진과는 독립적으로 Audit System을 갖추는 것이 중요하다고 할 수 있다.

물론 인터넷을 통해 쉽게 구할 수 있는 Audit 프로그램들은 많이 있다.(tcpdump[7], iplog[9]등) 그러나 이것은 Audit 자체가 프로그램의 목적이거나 그 하나로 프로그램이 완성되어 있기 때문에 제안된 알고리즘을 테스트하거나 검증하기 위해서는 각 프로그램들의 소스를 수정하든지, Audit 결과를 다시 개발자가 자신의 포맷에 맞게 수정해야 한다는 단점이 있다. 따라서, 본 논문에서는 Audit 시스템을 침입탐지 엔진과 분리하여 침입탐지 엔진에 상관없이 연동할 수 있는 Audit 시스템을 제안하고자 하며 침입탐지 엔진과는 별도로 Audit data의 활용 방안에 대해 연구하고자 한다.

2. 관련 연구

Audit에 관련된 연구는 현재 캐치되는 패킷 데이터를 어떻게 침입탐지에 이용할 수 있을까에 초점이 맞추어져 있다[2]. 대부분의 경우 침입탐지 모델을 어떤식으로 설정하느냐에 따라 이를 이용하는 Audit의 형태가 달라진다. 따라서 주로 Audit 자체에는 큰 의미를 두지 않는 경우가 많다. 그러나 순전히 Audit Data 자체에 의미를 두어서 연구가 진행되는 경우도 있다. 그 예로는 IDS를 위한 Data Mining Approach[3,4]가 있다. 이것은 방대한 Audit에서 침입이라고 간주할 수 있는 특정 규칙을 추출하고자 하는 시도이다. 또한 이러한 규칙이 추출된다면 그것을 가지고 어떻게 침입탐지에 이용할 수 있는가하는 모델 연구가 현재 진행중이다. 이미 Data Mining 기법은 CRM과 같은 다른 분야에서 활발하게 이용되고 있는 기법[6]이다. 이러한 기법을 Security Domain으로 옮겨와서 유용한 정보를 추출하고자하는 시도로서 Mining 기법으로는 Association Rule, Frequency Episode등의 알고리즘을 이용하고 있다. 이러한 접근 방법을 통해 테스트를 한 결과 이미 알려진 공격 방법에 대해서 Mining기법(연관규칙)을 적용했을 때 어느정도 특정 규칙들이 추출된다는 것을 알 수 있다. 위와 같은 테스트를 통해 일단 Mining 기법을 적용하는데에 있어서 검증은 되었다고 할 수 있다. 하지만 이 방법의 문제점은 특정

기준 없이 적용했을 경우 엄청나게 많은 규칙들이 나타날 수 있다는 것과 또한 이를 처리하기 위한 시스템 오버헤드도 많이 걸린다는 점이다. 위 연구의 궁극적인 목적은 Data Mining 기법을 이용한 자동화된 침입탐지 모델이다. 즉, 데이터를 계속 수집하면서 특정 룰들을 추출하고 그 룰들에 기반한 탐지 모델을 설계하고 구현하는 것이다. 시스템이 완성되면 계속해서 새롭게 생겨나는 Attack이나 새롭게 발견되는 룰을 기반으로 사용자에 의한 시스템 유지/보수 없이 시스템의 탐지 성능은 더욱 더 강해지는 것을 의미한다. 그림 1은 현재 연구되고 있는 모델의 구조도이다.

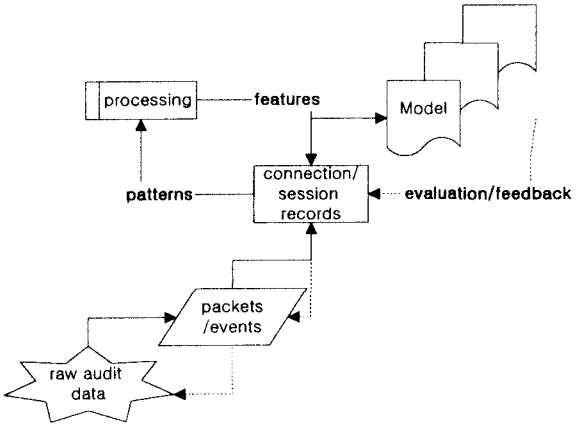

그림 1. Mining Audit Data for Automated Models

3. Audit 시스템 설계

3.1 시스템 개요

제안하는 Audit 시스템은 침입 탐지 모듈이 탐지를 위해 사용할 수 있는 정보를 제공해주는 도구를 설계, 개발하는 것이다. 여기에서 제공하는 정보는 네트웍 패킷 정보, 패킷 통계 정보, 연관 규칙 정보등 다양하게 정의할 수 있다. 이는 지금까지 연구되고 개발되고 있는 많은 침입 탐지 모델 및 알고리즘이 단순한 패킷정보 뿐만 아니라 여러가지 형태의 정보를 필요로 하기 때문이다. 각각의 탐지 모델에 따라 적용하는 Audit들이 다르고 이용하는 정보(IP, Port, Flag등)들이 서로 다르기 때문에 이러한 정보를 체계적으로 제공할 수 있는 Audit 프레임웍은 침입 탐지 모델을 제안하고 알고리즘을 개발하기 위한 좋은 기반이 된다. 침입 탐지 모델이 이용하는 Audit의 형태는 아주 많고 그 Audit들을 어떤식으로 다루느냐는 현재 무수히 많다. 위의 연구[3,4]는 그러한 많은 Audit의 형태중에 Mining기법을 적용하여 추출한 규칙을 가지고 적용할 수 있는 탐지 모델을 연구하는 것이고 이 시스템은 다양하게 많은 Audit의 형태를 침입탐지 모듈의 필요에 맞게 다양하게 제공해줌으로써 편이성을 높이고자하는 것이다. 물론 향후 이를 이용해서 통계나 Mining기법을 통해 특정한 형태의 침입 형태를 파악할 수도 있다.

3.2 시스템 구성

제안하는 audit 시스템은 탐지엔진과의 독립성을 주 목적으로 한다. 따라서 탐지 엔진과는 IPC 및 제공하는 library를 통해서 연동된다. 제안하는 Audit 시스템의 모델은 그림 2와 같다.

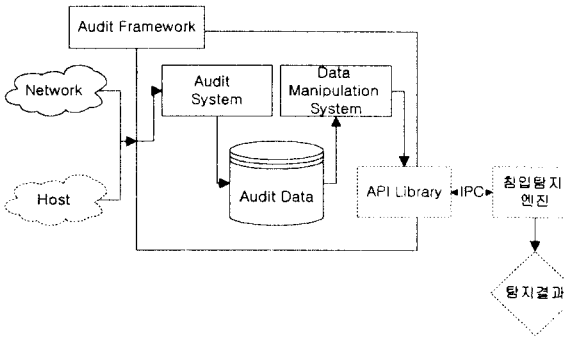

그림 2. Audit 시스템 구성도

3.3 시스템 기능

3.3.1 주요 기능

가) 실시간 Packet Capturing

현재 네트웍에 있는 감시할 수 있는 모든 패킷 및 호스트의 시스템 콜, 실행중인 프로세스 ID 및 상태를 수집한다. 단순히 패킷만을 캡쳐하는 것이 아니라 사용자의 용도에 맞게 캡쳐하는 것을 의미한다. 즉, TCP, UDP, ICMP등 Protocol별 수집, Host 단위별 수집, Port단위별 수집등 다양한 기능을 제공하는 것을 목적으로 한다. 또한 이렇게 수집된 패킷을 파일로 저장함으로서 각각 파일단위로도 가져다 사용할 수 있게 한다.

나) Packet 데이터 변환 및 압축

위에서 캡쳐한 패킷정보를 쉽게 알 수 있도록 IP, PORT, FLAG등으로 변환하고 파일로 저장한다.

다) 통계처리

수집된 패킷정보를 IP, TIME, FLAG등으로 통계처리를 해서 그에 대한 정보를 제공한다. 이러한 정보를 라이브러리를 통해 침입탐지 모듈에서 직접 사용할 수도 있고 이 기능만을 따로 이용하여 침입이나 일반 행위에 대한 통계적 자료로도 이용할 수 있다.

라) Data Mining 기법 적용

많은 데이터중에서 특정 룰 및 특징을 추출해내는 연관 규칙 알고리즘을 적용한다. 이 기법을 적용함으로써 방대하게 많은 데이터에서 비교적 적은 수의 규칙을 뽑아내줌으로써 침입탐지 모델이 빠르게 패킷에 대한 정보를 파악할 수도 있고 특정 행위 즉, 해킹 행위등에서의 특정 규칙을 추출하는 것으로도 이용할 수 있다.

3.3.2 시스템 흐름도

제안하는 Audit 시스템의 흐름도는 그림 3과 같다.

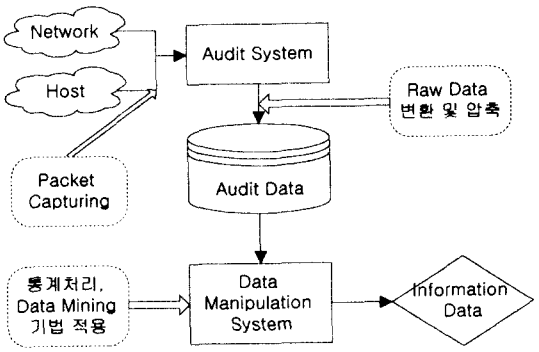

그림 3. 시스템 흐름도

4. 시스템 구현

4.1 개발 플랫폼

Hardware : Sun Ultra-60 Workstation
OS : Solaris 7 이상
Language : gcc 2.95.2 이상
Library : libpcap 0.6.2
Tcpdump : tcpdump 3.6.2

4.2 Audit System

Network packet 및 호스트 시스템의 프로세스 정보를 수집하는 기능을 한다. 네트웍 패킷은 libpcap을 이용하였으며 호스트는 BSM을 이용하였다.

4.3 Data Manipulation System

DMS는 크게 두가지로 나뉘어진다. Audit에 대한 통계 처리와 데이터 마이닝 기법을 이용한 특징 추출이다. 네트웍에서는 주로 다루는 것이 TCP 패킷이기 때문에 TCP Header의 특성을 따를 수 밖에 없다. 또한 호스트는 주로 프로세스 단위이기 때문에 시스템 콜 및 프로세스의 상태를 기준으로 구현된다.

가) 통계 처리

통계 처리를 위한 주요 기준은 다음과 같다.

- IP
- Port 단위
- 패킷 Flag
- 프로세스 상태
- 시스템 콜 단위

나) 데이터 마이닝 기법 적용

- Association Rule [5]

5 구현 결과

[그림 4]는 수집된 네트웍 패킷의 결과를 보여주는 화면이다.

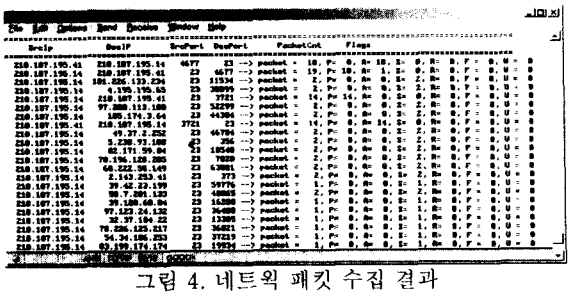

그림 4. 네트웍 패킷 수집 결과

6. 결론 및 향후연구

본 논문에서는 침입탐지 엔진과 독립적으로 작동하여 연동될 수 있는 Audit 시스템을 설계 및 구현하였다. 이로 인해 침입탐지 알고리즘을 적용, 검증할 수 있는 Test Bed를 제공하고, Audit를 가공, 이용하고 데이터에서 특징을 추출하는 데이터 마이닝 기술을 연구 개발함으로써 침입탐지 알고리즘의 손쉬운 개발을 지원할 수 있다.

앞으로의 과제는 이 시스템을 보다 더 범용적으로 사용할 수 있도록 다양한 프로토콜의 Packet Capture 기능을 지원하고 호스트 시스템에서 사용되는 각종 resource 및 침입 탐지에 이용될 수 있는 자원을 수집하며 연결된 패킷, 사용자의 행동의 패턴과 같은 연속적인 데이터를 수집하는 것이다. 또한 이러한 데이터를 바탕으로 다양한 데이터 마이닝 알고리즘을 적용하여 여러 행위의 규칙 및 패턴을 찾아낼 수 있는 시스템으로 발전시키는 것이며 IDS를 위한 Audit데이터 형태의 표준을 마련하는 것이다[8].

참고 문헌

[1] S. McCanne and V. Jacobson, " BSD Packet Filter : A New Architecture for User-level Packet Capture" , *USENIX, January 25-29, 1993. San Diego, CA.*

[2] D. Brent Chanpman, " Network Security Through IP Packet Filtering" , *USENIX UNIX Security Symposium, September, 1992.*

[3] W. Lee, Salvatore J. Stolfo, " Mining Audit Data to Build Intrusion Detection Models." , *USENIX Security Symposium, August 1998.*

[4] W. Lee, S J. Stolfo, " A Data Mining Framework for Adaptive Intrusion Detection" , *Proc. 1999 IEEE Symposium on Security and Privacy,Oakland, CA. 1999.*

[5] Rakesh Agrawal and Ramakrishnan Srikant. " Fast Algorithms for Mining Association Rules" . *Proc of the 20^{h} VLDB Conference 1994*

[6] 마이크로 소프트웨어 2001년 6월호

[7] http://www.tcpdump.org

[8] http://www.ietf.org

[9] http://www.packetfactory.net/