침입 탐지를 위한 정책 기반의 보안 관리

조수형[0] 김정녀
한국전자통신연구원
(shjo[0], jnkim)@etri.re.kr

Policy-bsed Security Management for Intrusion Detection

Su-Hyung Jo[0] Jeong-Nyoe Kim
Secure Operating System Research Team, Electronics and Telecommunications Research Institute

요 약

VPN, 전자상거래 등의 인터넷 서비스들이 인터넷을 통해 빠르게 퍼져가고 있지만, 인터넷이 가지고 있는 보안 취약성 때문에 항상 해킹의 위험에 노출되어 있다. 이러한 해킹의 피해를 최소화하고 동적으로 침입을 감지할 수 있는 침입 탐지 시스템과 같은 보안 솔루션이 필요하다. 그리고, 보안 정책이 없는 관리 시스템은 보안 환경의 변화에 민첩하게 대처하지 못하고 통합된 관리 방법을 제시하지 못한다. 이 논문에서는 표준화된 보안 정책과 분석, 유지, 복구 기능을 가지고 정책을 기반으로 동작하는 보안 관리 시스템을 설계하였다. 보안 관리 시스템은 정책에 따라 관리 상태를 설정하고, 정책의 통신을 위해 COPS를 이용한다. 그리고, 네트워크상의 패킷을 필터링하고 침입을 탐지하며 불법 침입을 통보한다.

1. 서 론

최근의 컴퓨팅환경은 그 영역이 점점 거대해지고 있으며, 인터넷의 보급과 간편하고 편리한 네트워크 접속, 사용을 제공하는 다양한 서비스로 인하여 그 형태가 복잡해지고 있다. 이러한 요구에 부응하기 위해 LDAP, E-Commerce [1], 분산 컴퓨팅 등의 기술이 등장하고 있다. 그러나, 스니핑, 스푸핑, 바이러스와 같은 많은 보안 취약성 때문에 인터넷은 항상 해킹의 위험에 노출되어 있다. 이러한 인터넷 보안문제를 해결하기 위해 바이러스 백신, 방화벽, 통합 보안 관리, 침입탐지 시스템 [2] 등이 필요하다. 침입탐지시스템 (IDS: Intrusion Detection System)은 방화벽이 침입의 격리를 실패한 경우에도 해킹의 피해를 최소화하고 동적으로 침입에 대응할 수 있는 차세대 보안 솔루션이다. IDS는 네트워크 기반의 IDS와 호스트 기반의 IDS가 있으며, 두 가지를 혼합한 하이브리드 IDS가 있다.

보안 정책 [3, 4]이 없는 보안 관리 시스템은 보안 환경의 변화에 민첩하게 대처하지 못하고 통합된 관리 방법을 제시하지 못한다. 이러한 문제를 해결하기 위해 표준화된 보안 정책을 가지고 시스템을 분석하고 유지 보수할 수 있는 정책 기반의 보안 관리가 필요하다. 정책을 기반으로 한 통합된 보안 관리 기술은 인터넷의 보안 문제를 관리하고 침입을 감지하며 이에 대응할 수 있다.

본 논문에서는 침입 탐지를 위한 정책 기반의 보안 관리 시스템을 설계하였다. 보안 관리 시스템은 네트워크 라우팅을 수행하는 네트워크 노드와 일반 호스트, 네트워크를 관리하는 관리 노드로 구성된다. 네트워크 노드는 패킷 필터링, 침입 탐지 및 분석, 침입 대응, 정책 수행을 하는 보안 라우터이다. 관리 노드는 정책을 기반으로 네트워크 노드와 일반 호스트를 관리하고, 정책의 결정과 수행을 위해 COPS를 사용한다. 그리고, 사용자 인터페이스인 웹 브라우저와 웹 서버간에 SSL을 바탕으로 통신한다.

본 논문의 구성은 다음과 같다. 2장에서는 침입 탐지 시스템에 대하여 알아보고 3장에서는 구현을 위한 COPS와 JSP에 대하여 알아본다. 4장에서는 관리 노드를 설계하고, 구현을 위해 필요한 사항에 대하여 설명한다. 그리고, 5장에서 결론을 맺는다.

2. 침입탐지 시스템 (IDS)

침입탐지 시스템(IDS)은 네트워크와 시스템의 정보를 침해하는 침입을 탐지하여 네트워크 관리자에게 효과적으로 네트워크를 관리할 수 있도록 판단하여 결정의 자료를 제공하는 시스템이다. 침입탐지 시스템은 자료 수집 및 축약, 침입 판단, 통보 및 대응 모듈로 구성된다. IDS는 네트워크 기반의 IDS (N-IDS)와 호스트 기반의 IDS (H-IDS)가 있다. 네트워크 기반의 IDS는 네트워크 노드에 설치되어 네트워크 노드로부터의 패킷을 분석한다. N-IDS는 시스템의 OS에 제한을 받지 않으며 H-IDS에 비하여 설치와 구동에 더 적은 비용이 든다. H-IDS는 네트워크를 감시하고자 하는 서버에 설치되어 운용단계에서 감사나 시스템 콜에 의해 저장된 시스템 로그 파일을 이용한다. 그러나, H-IDS는 N-IDS에 비하여 구현하기가 어렵고 설치와 구동 면에서 비용이 많이 든다. 현재 IDS는 정상 행위를 침입으로 판단하는 false positive error와 침입임에도 불구하고 정상행위로 간주하는 false negative error 문제를 가진다. 이를 해결하기 위해 침입을 판단할

때 사용하는 판단 규칙을 좀 더 세분화 할 필요가 있다. 그리고, 암호화된 패킷에 대한 침입 탐지와 우회경로를 통한 침입 탐지에 대한 기술 개발이 필요하다.

침입 탐지와 관련된 표준화 연구가 DARPA/ITO 나 IETF IDWG [5]에서 수행되고 있다. IDWG는 다양한 IDS 제품들 사이에 호환이 가능하도록 수집된 자료를 표현하는 데이터 형식과 내용에 대한 표준화 작업을 진행하고 있다.

3. 구현을 위한 기술

3.1 COPS (Common Open Policy Service)

COPS는 라우터, 스위치 등과 같은 네트워크 구성 요소들의 정책을 기술하고, 각 구성요소들끼리 정책을 교환하고 협상하는 프로토콜이다. IETF RFC 2748에서 COPS를 정의하고 있다. COPS는 단순 질의/응답 프로토콜로 정책을 교환하거나 설정하는 프로토콜이다. 다양한 영역으로 기능을 쉽게 확장 가능하고, TCP를 기반으로 설계된 프로토콜로 클라이언트와 서버 간의 동기화가 필요하다.

COPS를 확장한 프로토콜로 대표적으로 COPS-RSVP(COPS Usage for Resource Reservation)와 COPS-PR (COPS Usage for Policy Provisioning)이 있다. COPS-RSVP는 Outsourcing 모델로 PEP(Policy Enforcement Point)가 정책을 적용할 일이 있을 때마다 항상 PDP(Policy Decision Point)에게 정책 요청을 하고 PDP가 정책을 결정해서 PEP에게 준다. RFC 2749에 정의되어 있다. COPS-PR은 COPS-RSVP와 달리 PEP가 필요한 정책을 미리 가지고 있으면서 정책을 적용하고, 정책을 추가하거나 수정이 필요할 때 PDP에게 요청하여 새로운 정책을 받는다. RFC 3084에 정의되어 있다. " provisioning" 은 여러 가지 의미로 사용되는데, 여기에서는 " providing" 의 의미와 " configuring" 의 의미로 사용한다.

인텔에서 COPS를 구현하여 IntelR COPS Client Software Development Kit 3.1을 프리웨어로 제공하고 있으며, IntelR COPS Server를 판매하고 있다 [6]. 오픈 소스 단체인 Vovida.org에서 C++로 COPS를 구현하여 COPS stack 1.2.0을 제공하고 있다 [7]. 그리고, Lulea 대학의 석사논문에서도 COPS를 구현하였다 [8].

3.2 JSP (Java Server Page)

JSP (Java Server Page)는 서블릿으로 웹 페이지의 내용이나 형태를 제어하는 기술이다. JSP는 웹 개발자와 디자이너가 웹 페이지를 빠르게 개발할 수 있고, 쉽게 유지보수 할 수 있도록 도와준다. 그리고, 플랫폼에 관계없이 손쉽게 웹 기반의 응용을 개발 할 수 있도록 해준다. JSP는 웹 페이지 내용을 만들 때 Java를 이용하여 XML과 같은 태그나 scriptlet을 사용한다. JSP는 자바 서블릿을 확장한 기술로 서블릿은 서버에서 돌아가는 작은 프로그램을 말한다. 서블릿은 웹 개발자에게 웹 서버 기능을 확장할 수 있는 단순하고 일정한 매커니즘을 제공한다. 현재 많은 개발자들이 서블릿을 이용하여 웹 응용을 개발하고 있다.

4. 시스템 설계

4.1 시스템 프레임워크

보안 관리 시스템은 네트워크 라우팅을 수행하는 네트워크 노드와 일반 호스트, 네트워크를 관리하는 관리 노드로 구성된다. 네트워크 노드는 패킷 필터링, 침입 탐지 및 분석, 침입 대응, 정책 수행을 하는 보안 라우터이다. 관리 노드는 정책을 기반으로 네트워크 노드와 일반 호스트를 관리하고, 정책의 결정과 수행을 위한 통신 프로토콜로 COPS를 사용한다. 그리고, 사용자 인터페이스인 웹 브라우저와 웹 서버간에 SSL을 바탕으로 통신한다. 그림 1은 보안 관리 시스템의 프레임워크를 나타낸 그림이다.

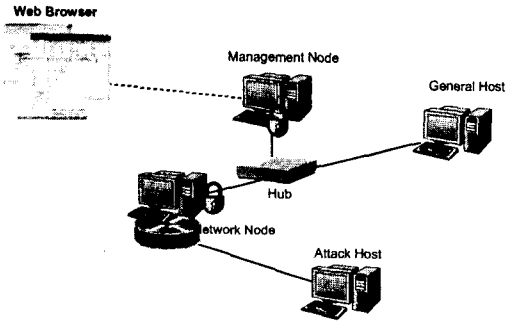

그림 1. 보안 관리 시스템의 프레임워크

4.2 관리 노드 구조

관리 노드는 정책을 기반으로 네트워크 노드와 일반 호스트를 관리한다. 관리 노드는 management servlet, network sensor, policy decider, policy parser, audit manager, audit handler와 데이터베이스로 구성된다. 그림 2는 관리 노드의 세부 구조이다. Management servlet은 로그인에 대한 처리, 패킷 모니터링과 네트워크와 정책에 관한 정보를 웹 브라우저로 디스플레이하는 모듈이다. Network sensor는 호스트 정보, 라우팅 정보, 패킷 정보와 같은 네트워크 정보를 수집하여 Network DB에 저장한다. Policy decider는 네트워크 노드와 호스트에 필요한 정책을 결정하고, Policy parser 정책을 변환한다. Audit manager와 Audit handler는 불법 접근, bad packet, 침입의 경보를 처리한다. Policy DB는 접근 제어 규칙, 필터링 규칙, 침입 탐지 규칙의 정책에 대한 데이터베이스이다.

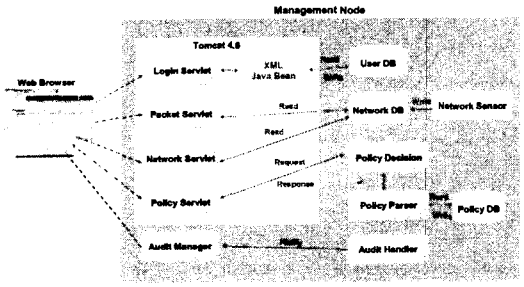

그림 2. 관리 노드 구조

톰캣 4.0 [9]은 Management servlet을 위한 servlet engine을 제공하는 프로그램이다. 톰캣은 Management servlet을 실행하고 관리 정보를 웹 브라우저로 보여주는 웹 서버 역할까지 한다.

4.3 구현을 위해 필요한 사항

Linux Kernel 2.4.18 (Red Hat 7.3)에서 정책 기반의 보안 관리 시스템의 개발하고 있다. JSP 서버 프로그램으로 톰캣 4.0.4과 자바 컴파일러로 JDK 1.3.1[10]을 사용한다. 톰캣 4.0 버전은 JDK 1.4 버전과 호환성 문제가 발생할 수 있으므로 JDK 1.3.1을 사용한다. 아파치와 톰캣 4.0을 연결하는 모듈로 mod_webapp.so을 사용한다. 이 모듈을 "/usr/local/apache/libexec/"에 복사한다. 아파치를 실행할 때, "mod_webapp.so"를 호출하기 위해 conf/httpd.conf 파일에 "LoadModule webapp_module libexec/mod_webapp.so"을 추가한다. 윈도우와 Solaris 환경에서 인터넷 익스플로러나 넷스케이프 같은 웹 브라우저를 실행하여 보안 관리 시스템 접속한 후 사용한다. 웹 브라우저에서 웹 서버로의 접속은 HTTP 통신을 하고 사용자 인증을 거친 후에 일어나는 다른 모든 통신은 SSL을 기반으로 한 TCP 통신을 한다.

5. 결론

정책 기반 보안 관리는 방화벽, 침입탐지시스템, 접근제어시스템을 제어하는 보안 정책을 이용하여 네트워크를 관리하고, 보안 정책에 의해 변경된 사항을 네트워크에 적용하며 네트워크의 특정 문제에 대해 자동화된 구성과 제어 솔루션 이용하여 네트워크의 보안을 제공하는 것이다. 본 논문에서는 침입 탐지를 위한 정책 기반 보안 관리에 대하여 설계하고 구현 방법에 대하여 알아보았다. 정책 수행을 위한 통신 프로토콜로 COPS를 이용하여 정책 기반 보안 관리를 개발하고, JSP를 이용하여 플랫폼에 관계없이 웹 기반의 인터페이스를 개발할 수 있다. 향후에는 암호화된 패킷에 대한 침입 탐지와 우회경로를 통한 침입 탐지 방법에 대한 연구가 필요하다.

참고문헌

[1] Richard Kee, Roger Walton, Henning Dransfeld, and Nick Harman, Ovum Forecast the Internet and E-commerce, Ovum, July 2000.
[2] Stephen Northcutt and Judy Novak, Network Intrusion Detection. An Analyst's Handbook, 2nd ed. New Riders, 2001.
[3] IETF Policy Working Group
http://www.ietf.org/html.charters/policy-charter.html
[4] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, The Common Open Policy Service Protocol: RFC 2748, January 2000.
http://www.ietf.org/rfc/rfc2748.txt
[5] IETF Intrusion Detection Working Group
http://www.ietf.org/html.charters/idwg-charter.html
[6] Intel COPS SDK
http://www.intel.com/labs/manage/cops/
[7] Vovida.org, http://www.vovida.org/
[8] Lulea University
http://epubl.luth.se/1402-1617/2000/125/
[9] Tomcat, http://jakarta.apache.org/
[10] Java, http://java.sun.com/
[11] S. H. Jo, J. H. Nah, & S. W. Sohn, Internet Security Management System for IPsec, Proceeding of NordU2002/USENIX Conference, Helsinki, Finland, February 2002.
[12] 조수형, 김정녀, "인터넷 패킷보호 보증 플랫폼의 보안 관리 시스템 설계 및 구현", 통신정보 합동학술대회(JCCI 2002), April 2002.