SSL/TLS를 이용한 Management Agent(M/A) 구조상의 Network IDS 설계 및 구현

정숙희[0] 김행욱 강진석 강흥식
인제대학교 정보컴퓨터공학부
deliz@hanmir.com[0] hukim@nice.inje.ac.kr
comdol2@netian.com hskang@nice.inje.ac.kr

Design and realization of Network IDS with Management Agent(M/A) through SSL/TLS Protocol.

Suk-Hee Jung[0], Haeng-Uk Kim, Jin-Suck Kang, Heung-Sik Kang
Dept. of Computer Engineering, Inje University

요 약

정보기술의 발전은 디지털 혁명의 근간이 되고 있는 반면, 정보유출·파괴·변조 등의 역기능 또한 내포하기 때문에 이러한 정보화의 역기능에 대한 신속한 탐지 및 능동적인 대응이 절실히 요구되고 있는 실정이다. 그러나 기존의 시스템은 침입에 대한 차단을 중심으로 운영되어 왔고, 침입탐지 시스템이라 할지라도 불법 침입에 대한 대응이 단지 경고나 보고 수준에 머물고 있어 능동적인 대응이 없다는 단점이 있다. 따라서 본 논문에서는 리눅스 시스템에서 실시간 침입탐지 시스템인 NIDS를 설계·구현하고, 동일 네트워크 망에서의 여러 호스트들을 통합적으로 관리하기 위하여 M/A 시스템을 설계·구현하였다. 여기서 NIDS와 M/A 시스템간의 통신은 SSL/TLS 프로토콜을 기반으로 이루어진다. NIDS에서는 Detection Module과 Response Module을 분리하여 구현하였고, 커널 레벨에서 PCAP 라이브러리를 통해 캡처된 패킷이 Detection Module에 의해 침입이라 판단되면 Response Module에서 일정 수준의 대응이 가능하도록 구현하였다.

1. 서론

정보기술(IT)의 발전은 18세기 산업혁명과 비견될 만큼 매우 획기적인 것으로 간주되고 있으며, 특히 물리적 공간을 초월한 사이버 공간이라는 새로운 개념을 탄생시킨 네트워크에 대한 기술발전은 가히 혁명적이라 할 정도로 비약적인 발전을 거듭하고 있다. 이와 같은 네트워크 체계와 정보기술의 발전은 21세기에 들어오면서 '사이버 공간의 글로벌화'라는 개념으로 발전되어 전 세계를 시간과 공간을 초월한 단일 체계로 묶고 있으며, 이를 통한 무한한 변화와 새로운 가능성을 창출해 나가고 있는 등 그 영역이 점차 확대되어 디지털 혁명의 근간이 되고 있다.

정보보호체계는 크게 분류하여 탐지(Detection), 차단(Prevention), 대응(Response)으로 나뉘어져 있으나, 지금까지는 대부분 외부의 알려진 침입에 대한 차단을 중심으로 운영되어 왔다. 그러나 정보기술의 발전은 그 순기능뿐만 아니라 정보유출·파괴·변조 등의 역기능 또한 내포하기 때문에 이러한 정보화의 역기능에 보다 능동적이고 적극적으로 대처하기 위해 신속히 탐지하고 대응할 수 있는 정보보호 관련 기술의 연구개발 및 첨단기술의 도입이 절실히 요구되고 있는 실정이다.

따라서 본 논문에서는 리눅스 시스템에서 실시간 침입탐지 시스템인 Network IDS(NIDS)를 설계·구현하고, 동일 네트워크 망에서의 여러 호스트들을 통합적으로 관리하기 위하여 M/A 시스템을 설계·구현하였다. 여기서 NIDS와 M/A 시스템 사이의 통신은 SSL/TLS 프로토콜을 기반으로 한다. 이에 본 논문의 양식은 다음과 같다. 2장에서는 기존 연구의 문제점을 짚어보고, 3장에서는 본격적으로 본 논문에서 제안하는 SSL/TLS 프로토콜을 이용한 M/A 시스템과 NIDS의 설계와 구현 방안이 서술된다. 이어서 4장에서는 구현된 본 시스템의 실험 결과를 보인 후 마지막으로 5장에서 결론과 향후 발전 방향에 대해서 언급하고 논문을 맺도록 하겠다.

2. 기존 연구의 문제점

현재 오픈 소스로 개발되어 있는 침입탐지 시스템은 여러 가지가 있으나, 그 중에서 대표적인 시스템으로는 네트워크 취약점 검색 공격을 실시간으로 탐지하고 대응할 수 있는 도구인 RTSD와 침입탐지 rule을 내장하여 실시간으로 침입을 탐지하는 Snort를 들 수 있다.

RTSD는 한국정보보호진흥원에서 개발한 시스템으로 스캔 공격을 자동으로 탐지하여 E-mail을 통하여 네트워크 관리자 및 CERTCC-KR에 보고를 해주는 보안도구이다. RTSD는 로그분석에는 효율적이나 침입탐지 기능이 스캔 기능에 국한되어 있고 또한 탐지된 검색 공격에 대한 대응이 단지 E-mail을 통해 보고하는 수준이기 때문에 침입에 대한 직접적인 차단 기능이 부족하다.

Snort는 패킷 수집 라이브러리인 libpcap에 기반을 둔

네트워크 스니퍼인데, 쉽게 정의할 수 있는 침입탐지 rule들에 일치되는 네트워크 트래픽을 감시하고 기록하고 경고할 수 있는 도구이다. Snort는 프로토콜 분석, 내용 검색/매칭을 수행할 수 있으며 오버플로우, Stealth 포트스캔, CGI 공격, SMB 탐색, OS 확인 시도 등의 다양한 공격과 스캔을 탐지할 수 있다. Snort는 630여개 이상의 풍부한 탐지 rule들이 제공되어 막강한 탐지 능력을 가지고 있고, 로그가 각 IP 주소별로 세부적인 패킷이 다 저장되어 관리하기 편리하다. 반면 탐지 기능이 단일 호스트에 국한되어 있고, 불법침입에 대한 대응이 경고 메시지를 저장하는 수준에 머물고 있어 능동적인 대응이 없다는 단점이 있다.

3. 설계 및 구현

그림 1은 전체 시스템의 구성도를 보여준다.

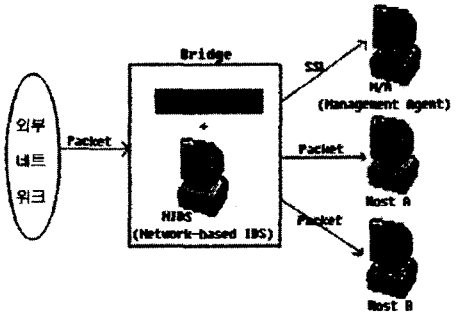

그림 1. 전체 시스템 구성도

Bridge Firewall이 구축된 시스템에 설치된 NIDS는 관리하고 있는 호스트 A와 B로 들어오는 모든 패킷을 복사하여 분석하고 그에 따른 대응을 하게 된다. 그때 생성된 로그는 SSL/TLS 프로토콜을 이용하여 암호화되어 M/A 시스템에 보내어 보여지게 된다. M/A에서는 NIDS가 관리하고 있는 동일 네트워크 망의 모든 호스트들에 대한 로그들을 쉽게 볼 수 있게 wxPython으로 구현되어 있다.

3.1 Bridge Firewall이 구축된 시스템에서의 NIDS

Bridge Firewall은 ip를 가지지 않으면서 허브와 같은 개념으로, 컴퓨터에 랜 카드 2개를 꽂아서 패킷이 랜 카드 1로 들어와서 랜 카드 2로 나가는 방식을 지원한다. 일반적으로 Router 형식으로 동작하는 Firewall은 설치하고자 할 때 Network을 재구성하는 불편함이 있을 수 있으나, Bridge Firewall은 기존 Network에 아무런 설정이나 변화 없이 설치가 가능하고, Routing을 하지 않으므로 처리 속도 면과 안정성 면에서도 일반 Firewall에 비해 뛰어나다고 할 수 있다. 또한 Bridge Firewall은 traceroute로 해도 나타나지 않는다. 리눅스에서는 커널 레벨에서 Bridge를 지원하고 있으므로, 커널 컴파일만으로도 쉽게 구축할 수 있다.

본 논문에서 구현한 NIDS는 이와 같이 여러 장점들을 가진 Bridge Firewall이 구축되어 있는 시스템에 구현하였다. 이는 Bridge Firewall 시스템에서 iptables로 생성된 체인규칙에 의해 외부의 침입으로부터 1차 적으로 방어를 취하며, 침입이 내부로 들어왔을 때 2차 방어수단으로 NIDS가 동작하여 보안을 더욱 강화하였다.

3.2 Detection과 Response Module이 분리된 NIDS

그림 2는 NIDS의 Architecture를 보여준다.

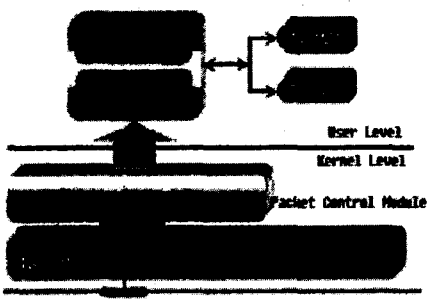

그림 2. NIDS의 Architecture

NIDS는 커널 레벨에서 PCAP 라이브러리를 통해 패킷 캡쳐를 수행하면서 NIDS가 관리하고 있는 동일 네트워크 망의 모든 호스트들의 패킷을 복사한다. 여기서 얻어낸 패킷은 Detection Module에서 분석되고 그 결과 침입이라고 판단되면 Response Module이 일정 수준의 대응을 하게 된다. 여기서 Detection Module과 Response Module은 Object-Oriented Programming을 통해 제작하여, 시스템 관리자가 쉽게 추가·삭제할 수 있다. 또한 커널 레벨에서 패킷 캡쳐가 수행되어지므로 속도 면에서 빠르다.

3.3 wxPython으로 구현된 M/A

wxPython은 Python으로 사용 가능한 여러 개의 GUI 툴킷 중의 하나로 플랫폼에 독립적이므로, M/A 시스템이 반드시 리눅스 머신에 설치되어 있지 않아도 여러 호스트들에 대한 실시간 침입탐지의 로그들을 볼 수 있다. 또한 wxPython은 라이브러리를 직접 불러 사용하므로 수행 속도가 빠르고 메모리 사용 면에서도 효율적이다.

3.4 SSL/TLS 이용한 NIDS에서 M/A로의 로그 전달

SSL/TLS 프로토콜은 클라이언트와 서버 사이에 인증 및 암호화 통신을 위해 사용되는 프로토콜이다. M/A 시

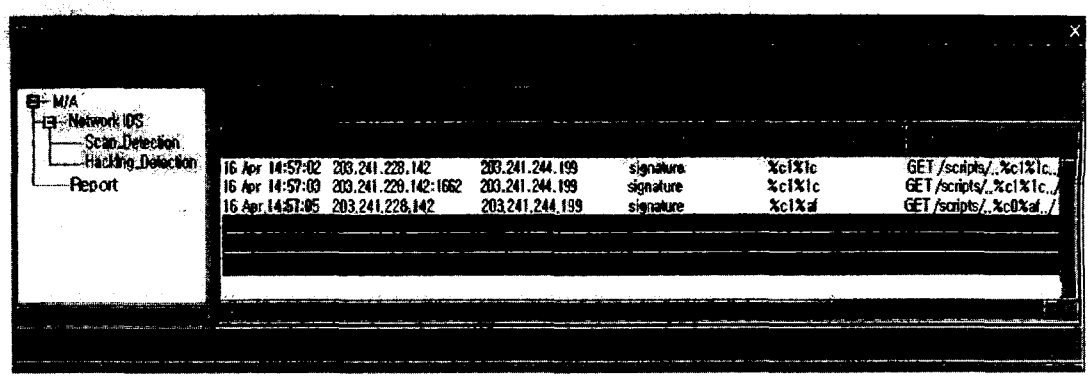

그림 3. M/A 시스템의 실행화면

스템에는 SSL/TLS의 서버 프로그램을 띄워놓고 NIDS에는 클라이언트 프로그램을 실행시켜 암호화된 데이터를 전달한다. M/A 시스템과 NIDS 간의 암호화 통신은 Sniffing을 미연에 방지할 수 있게 한다.

4. 실험

그림 3은 각 공격에 대한 로그가 M/A 시스템에 보여지는 화면이다. 그림 3에서 볼 수 있듯이 M/A 시스템에는 NIDS에서 SSL/TLS 프로토콜을 이용해 전송된 동일 네트워크 망 내의 여러 호스트들에 대한 실시간 침입 탐지 로그들을 보여주고 있다. 로그는 Time, Attack Host, Target Host, Attack Type 등의 필드를 가지며 각각은 공격한 시간, 공격 호스트의 IP 주소, 목적지 호스트의 IP 주소, 공격 유형 등을 말한다. 이와 같이 동일 네트워크 망에 있는 여러 호스트들의 로그를 한 시스템에서 통합하여 관리할 수 있는 이점이 있다.

5. 결론 및 향후 과제

본 연구에서는 기존에 개발된 보안 프로그램들이 실시간 침입탐지에 따른 대응이 효율적이지 못하다는 결론을 내리고 그에 대한 해결 방안으로 본 시스템을 설계·구현하였다. 그리고 지금까지의 보안 프로그램들과는 달리 Bridge Firewall 시스템을 구축하여 1차 적으로 방어를 하며, 2차 방어 수단으로 NIDS가 동작하여 보안을 더욱 강화하였다. 또한 침입탐지가 단일 호스트 기반에 국한되는 것이 아니라 동일 네트워크 망 내의 여러 호스트들에 대한 침입탐지 NIDS를 구현하고, 각 호스트들의 로그를 통합적으로 관리하기 위해 M/A 시스템을 구현하였다. NIDS에서 Detection한 패킷들에 대한 로그는 SSL/TLS 프로토콜을 이용하여 데이터를 안전하게 암호화하여 M/A 시스템으로 전달할 수 있게 하였다. M/A 시스템에서는 여러 호스트들에 대한 공격 시간, 공격 호스트의 IP 주소, 공격 유형 등의 로그를 볼 수 있다.

그러나 본 시스템은 몇 가지 부분적인 개선의 필요성이 있다. 먼저, 본 시스템에는 침입으로 판단된 패킷에 대해 일정 수준의 대응이 가능한 Response Module이 있으나 현재로서는 iptables의 체인규칙에 의해 공격자 IP 주소를 차단하는 수준이기 때문에 차후에는 역추적 시스템과 같은 좀 더 능동적이고 체계적인 대응을 수립할 필요성 있다는 것. 그리고 아울러 룰 기반 시스템의 특성을 효과적으로 활용하기 위해, 새로운 공격 패킷의 패턴이 발견될 경우 이를 실시간으로 분석 및 전송 가능한 시스템과의 유기적인 통합이 선행되어야 한다는 점이다.

참고문헌

[1] http://kldp.org
[2] http://www.certcc.or.kr
[3] http://www.openssl.org
[4] http://woogi.org
[5] http://wxpython.org/
[6] PLUS(포항공대 유닉스 보안 연구회), "Security PLUS for UNIX", 영진.Com
[7] Richard Stevens, "Advanced Programming in the UNIX Environment", Addison-Wesley
[8] Richard Stevens, "TCP/IP Illustrated, Volume1", Addison-Wesley
[9] Richard Stevens, "UNIX Network Programming Vol.1", Prentice Hall
[10] Daniel Pierre Bovet, Marco Cesati, "Understanding the LINUX KERNEL", O`Reilly