韓國컴퓨터情報學會 論文誌
第14卷 第12號, 2009. 12.

2009-14-12-3-6

다중 계층 웹 필터를 사용하는 웹 애플리케이션 방화벽의 설계 및 구현

장 성 민*, 원 유 헌**

Design and Implementation of a Web Application Firewall with Multi-layered Web Filter

Sung-Min Jang*, Yoo-Hun Won**

요 약

최근 인터넷 상에서 빈번하게 발생하는 내부 정보와 개인 정보 유출과 같은 보안 사고들은 보안을 고려하지 않고 개발된 웹 애플리케이션의 취약점을 이용하는 방법으로 빈번하게 발생한다. 웹 애플리케이션의 공격들에 대한 탐지는 기존의 방화벽과 침입 탐지 시스템들의 공격 탐지 방법으로는 탐지가 불가능하며 서명기반의 침입 탐지 방법으로는 새로운 위협과 공격에 대한 탐지에 한계가 있다. 따라서 웹 애플리케이션 공격 탐지 방법에 대한 많은 연구들이 웹 트래픽 분석을 이용하는 비정상행위 기반 탐지 방법을 이용하고 있다. 비정상행위 탐지 방법을 사용하는 최근의 웹 방화벽에 관한 연구들은 웹 트래픽의 정확한 분석 방법, 패킷의 애플리케이션 페이로드 검사로 인한 성능 문제 개선, 그리고 다양한 네트워크 보안장비들의 도입으로 발생하는 통합관리 방법과 비용 문제 해결에 중점을 두고 있다. 이를 해결하기 위한 방법으로 통합 위협 관리 시스템이 등장 하였으나 부족한 웹 보안 기능과 높은 도입 비용으로 최근의 애플리케이션 공격들에 대해 정확한 대응을 하지 못하고 있는 현실이다. 본 연구에서는 이러한 문제점들을 해결하기 위해 웹 클라이언트의 요청에 포함된 파라미터 값의 길이에 대한 실시간 분석을 이용하여 공격 가능성을 탐지하는 비정상행위 탐지방법을 제안하고, 애플리케이션 데이터 검사로 발생하는 성능 저하 문제를 해결할 수 있는 다중 계층 웹 필터를 적용한 웹 애플리케이션 방화벽 시스템을 설계하고 구현하였다. 제안된 시스템은 저가의 시스템이나 레거시 시스템에 적용 가능하도록 설계하여 추가적인 보안장비 도입으로 야기되는 비용 문제를 해결할 수 있도록 하였다.

Abstract

Recently, the leakage of confidential information and personal information is taking place on the Internet more frequently than ever before. Most of such online security incidents are caused by attacks on vulnerabilities in web applications developed carelessly. It is impossible to detect an attack on a web application with existing firewalls and intrusion detection systems. Besides, the signature-based detection has a limited capability in detecting new threats. Therefore, many researches concerning the method to detect attacks on web applications are employing anomaly-based detection methods that use the web traffic analysis. Much research about anomaly-based detection through the normal web traffic analysis focus on three problems - the

• 제1저자 : 장성민
• 투고일 : 2009. 08. 20, 심사일 : 2009. 09. 09, 게재확정일 : 2009. 12. 24.
* 어울림정보기술(주) 기술연구소 책임연구원 ** 홍익대학교 컴퓨터공학과 교수
※ 이 논문은 2008년 홍익대학교 교내연구비 지원으로 연구되었음.

method to accurately analyze given web traffic, system performance needed for inspecting application payload of the packet required to detect attack on application layer and the maintenance and costs of lots of network security devices newly installed. The UTM(Unified Threat Management) system, a suggested solution for the problem, had a goal of resolving all of security problems at a time, but is not being widely used due to its low efficiency and high costs. Besides, the web filter that performs one of the functions of the UTM system, can not adequately detect a variety of recent sophisticated attacks on web applications. In order to resolve such problems, studies are being carried out on the web application firewall to introduce a new network security system. As such studies focus on speeding up packet processing by depending on high-priced hardware, the costs to deploy a web application firewall are rising. In addition, the current anomaly-based detection technologies that do not take into account the characteristics of the web application is causing lots of false positives and false negatives. In order to reduce false positives and false negatives, this study suggested a realtime anomaly detection method based on the analysis of the length of parameter value contained in the web client's request. In addition, it designed and suggested a WAF(Web Application Firewall) that can be applied to a low-priced system or legacy system to process application data without the help of an exclusive hardware. Furthermore, it suggested a method to resolve sluggish performance attributed to copying packets into application area for application data processing. Consequently, this study provide to deploy an effective web application firewall at a low cost at the moment when the deployment of an additional security system was considered burdened due to lots of network security systems currently used.

▸ Keyword : 웹 애플리케이션 방화벽(web application firewall), 서명기반탐지(signature-based detection), 비정상행위기반탐지(anomaly-based detection), 통합위협관리시스템(unified threats management system)

Ⅰ. 서 론

최근의 애플리케이션과 웹 애플리케이션을 대상으로 하는 다양하고 고도화된 인터넷 공격들과 웹 서버 취약성을 이용하는 공격들은 기존에 네트워크 레이어에 집중되어있던 공격탐지 및 차단 방법을 애플리케이션 레벨까지 수행하도록 하고 있다[1,3,4,7,14]. 또한 기존의 도입된 많은 네트워크 보안장비들과 함께 추가적으로 애플리케이션 레이어의 처리만을 수행하는 웹 애플리케이션 방화벽을 출현시키는 요인이 되었다. 웹 애플리케이션에 대한 공격의 탐지와 방어를 목적으로 출현한 웹 애플리케이션 방화벽은 공격의 탐지를 위해서 기존의 서명기반 탐지 방법과 비정상행위기반 탐지 방법을 사용한다[1,3,4,5,7].

서명기반 탐지 방법은 새로운 위협과 공격에 대하여 미탐지 비율이 높은 문제점을 가지고 있다. 비정상행위기반 탐지 방법은 오탐지 문제점을 가지고 있지만, 새로운 공격을 탐지할 수 있는 장점 때문에 많은 시스템에서 적용하고 있다[1,2,3,8]. 이러한 비정상행위 기반의 탐지에 대한 연구들은 다음의 중요한 이슈들을 가지고 있다. 첫째, 주어진 웹 애플리케이션 트래픽에 대한 정확한 분석 방법이다[5,8]. 둘째, 애플리케이션 공격을 탐지하기 위해 수행되는 패킷의 애플리케이션 페이로드 분석과 처리로 발생하는 시스템의 성능 문제이다. 셋째로 많은 수의 네트워크 보안장비들의 도입으로 야기되는 통합관리와 비용문제이다. 네트워크 상의 하나의 시점에서 모든 네트워크 레벨의 보안을 제공하고자 하는 통합위협관리 시스템이 제안되고 있으나 기능의 효율성과 고가의 비용으로 추가적인 부담이 되고 있는 것이 현실이다[10,18]. 또한, 다양한 웹 공격을 정확하게 탐지하고 차단할 수 있는 하드웨어 일체형 웹 애플리케이션 방화벽에 관한 연구들이 진행되고 있다[4,5,7,10]. 이러한 연구들은 애플리케이션 처리로 야기되는 시스템 성능 저하 문제를 고가의 하드웨어를 사용하는 방법으로 패킷 처리속도 개선에 집중하고 있어 현실적으로 고가의 웹 애플리케이션 방화벽 도입은 추가적인 비용의 부담을 야기한다.

본 논문은 이러한 문제점을 개선하기 위해 보다 정확한 웹 애플리케이션의 실시간 분석 방법과 저가의 시스템에서 구현 가능한 웹 애플리케이션 방화벽을 제안한다. 웹 클라이언트의

요청을 애플리케이션 특성에 맞는 실시간 분석 방법을 제공하고 패킷 복사로 야기되는 시스템 성능 저하의 문제를 해결할 수 있는 다중 웹 필터를 제안한다. 또한 저가의 시스템과 레거시 시스템들에 쉽게 적용할 수 있는 구조로 설계하여 웹 애플리케이션의 도입으로 요구되는 비용 부담을 해결한다. 제안된 시스템을 실제 운영환경에 적용하여 효용성을 검증하고, 저 사양의 하드웨어 시스템에서도 효율적으로 웹 애플리케이션 방화벽을 적용할 수 있는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴보고 3장에서는 본 연구에서 제안한 시스템 구조를 설명한다. 4장에서는 실험과 결과를 분석하고 5장에서는 결론을 기술한다.

II. 관련 연구

침입 탐지는 컴퓨터 시스템이나 네트워크에 발생하는 사건들을 모니터링하는 행위와 보안정책 위반 그리고 보안 위협과 같은 보안사고 징후를 분석하는 행위들로 정의할 수 있다[12].

침입 탐지 시스템(Intrusion Detection System)은 침입 탐지 행위를 자동으로 수행하는 시스템으로 정의되며 침입 방지 시스템은 모든 침입 탐지 기법들을 이용하여 보안 사고를 야기할 수 있는 시도들에 대한 탐지와 방어를 수행하는 시스템으로 정의할 수 있다[12]. 웹 애플리케이션 방화벽은 이러한 침입 방지 시스템의 행위 중에서 특별히 웹 애플리케이션에 대한 침입과 불법 접근을 방지하기 위해 제안된 시스템으로 최근에 빈번하게 발생하고 있는 웹 애플리케이션에 대한 침해사고로 인하여 등장한 시스템이다[4,5,7,10].

침입 탐지는 다양한 형태로 수행되고 있지만 크게 서명기반탐지 방법(signature-based detection)과 비정상행위기반 탐지 방법(anomaly-based detection)으로 분류되고 있다[2,9]. 서명기반탐지 방법은 이미 잘 알려진 공격 패턴을 서명으로 정의하고 서명과 동일한 형태를 가지는 패킷이 수신될 경우에 공격으로 판단하는 방법이다[2]. 따라서, 서명기반탐지 방법은 잘 알려진 공격 탐지에는 효과적이지만, 서명 값이 존재하지 않는 새로운 공격에 대해서는 효과적이지 못하다[12]. 반면, 비정상행위탐지 방법은 네트워크와 시스템의 정상적인 행위와 비교하여 현저하게 차이를 나타내는 현상이 발생하였을 경우 공격 가능성으로 판단하는 방법이다. 비정상행위탐지 방법을 사용하는 침입탐지 시스템은 사용자 정보, 서버 정보, 네트워크 트래픽 흐름, 애플리케이션 정보 등의 프로파일(profile)로 정상적인 행위를 정의하여 사용한다. 비정상행위탐지 기법의 가장 큰 장점은 알려지지 않은 새로운 위협을 탐지할 수 있는 것이다. 하지만, 비정상행위에 대한 정확한 정의가 어려워 공격탐지에 빈번한 오탐을 야기할 수 있다[1,2,3].

웹 애플리케이션 방화벽은 애플리케이션 계층의 데이터 분석을 통하여 애플리케이션과 애플리케이션 프로토콜에 대한 위협적인 행위를 탐지하여 기존의 방화벽과 침입 탐지 시스템으로는 탐지할 수 없는 웹 애플리케이션에 대한 공격들을 방어하는 시스템으로 정의할 수 있다. 또한 URL에 따른 접근 제어 기능과 암호화된 SSL(Secure Socket layer) 통신을 검사하는 기능 등을 포함한다. 웹 애플리케이션 방화벽은 공격을 탐지하는 방법으로 포지티브 보안 모델과 네거티브 보안 모델로 구분할 수 있는데 포지티브 보안 모델은 안전하다고 정의된 것만 접속을 허용하는 방법이고, 네거티브 보안 모델은 위험하다고 미리 정의된 것만 접속을 거부하고 나머지는 모두 허용하는 방법이다[12,18,21]. 침입 탐지 시스템이 서명기반 탐지와 비정상행위 탐지를 상호 보완하여 사용하듯이 대부분의 웹 애플리케이션 방화벽도 위의 두 개의 모델을 함께 적용하여 구현한다. 웹 애플리케이션 방화벽은 그 적용 방법에 따라 네트워크 기반과 웹서버 기반으로 구축이 가능하다. 네트워크 기반의 웹 애플리케이션 방화벽은 네트워크 기반의 침입 탐지 시스템으로 웹 서버의 앞에서 웹 서버로 향하는 웹 애플리케이션 트래픽 분석을 수행한다. 따라서 환경을 구성할 때 웹 서버에 의존적이지 않으며 웹 서버에 별다른 영향을 주지 않고 침입 탐지 기능을 수행한다. 웹 서버 기반은 호스트 기반의 침입 탐지 시스템으로 웹 서버와 동일한 시스템으로 구성되어 웹 서버에 의존적일 수 있다. 애플리케이션 기반의 비정상행위 탐지는 애플리케이션 계층의 데이터 페이로드 같은 패킷의 데이터 정보가 공격탐지 대상이다. 즉 HTTP, FTP 등과 같은 서비스 대상 공격을 탐지하기 위해서는 데이터 페이로드 분석을 수행해야 한다. 특정 서비스마다 페이로드 구조 및 트래픽 패턴이 다르기 때문에 서비스마다 정상 트래픽에 대한 프로파일 정보가 필요하다[1,2,8]. 비정상 행위의 정확한 탐지를 위해서는 애플리케이션의 정확한 분석이 필요하다[4,6,9,12].

III. 다중 웹 필터를 가지는 WAF

3.1 웹 애플리케이션 방화벽 설계

웹 애플리케이션 방화벽의 주요 기능을 세 가지 주요 모듈로 구성한다.[3,4,5,6,7]. 첫째, 네트워크 중계 모듈이다. 네트워크 중계 기능은 HTTP/HTTPS 서비스를 중계하기 위한 모듈로 사용자의 접속부터 웹 서버와 애플리케이션의 응답처

리까지의 모든 연결을 중계하는 기능을 제공한다[10,18,21]. 서비스를 중계하기 위한 프로세스를 생성하고 서비스에 대한 네트워크 정보를 관리한다. 그리고, 데이터베이스에 저장되어 있는 환경 설정 데이터를 이용하여 서비스를 초기화한다. 클라이언트의 요청이 들어오면 HTTP 헤더의 데이터를 분석한다. 보안 정책 모듈에서 분석된 데이터에 보안정책을 적용한다. 보안 정책의 적용에서 정상적인 클라이언트의 요청으로 판단되면 웹 서버로 응답을 요청하고, 비정상적인 요청이나 공격으로 판단되면 클라이언트의 요청을 차단하고 로그를 발생한다. 정상적인 경우에는 웹 서버의 응답을 클라이언트에 전송하고 로그를 저장한다. 둘째, 보안 필터링 시스템이다. 분석 시스템에 수신된 클라이언트의 요청과 웹 서버의 응답은 필터링 모듈로 전달된다. 보안 필터링 모듈은 등록된 웹 요청 패킷에 보안 정책을 적용하여 공격 여부를 판단하고 처리하는 핵심 기능을 담당한다. 필터링 모듈은 기본적으로 클라이언트 요청에 대해서 OWASP 10대 취약점[23]을 처리할 수 있어야 하며 장애 시에 대처할 수 있도록 바이패스(bypass) 기능을 제공하여야 한다[17,22,23]. 서버 응답의 보안 필터링은 개인 정보에 대한 유출 탐지 및 차단, 홈페이지 위변조 방지, 서버 정보 숨김 등의 기능을 제공하도록 설계한다. 셋째, 보안 관리 시스템 기능이다. 보안 관리 시스템 기능은 관리자 프로그램에서 보안 필터링 모듈과 분석 모듈에 접근할 수 있는 관리자 계정을 생성 및 관리할 수 있으며 접근 통제와 권한 설정을 통해 허가된 관리자에게만 시스템 접근 권한을 제공하고 관리 할 수 있도록 하는 기능이다. 시스템의 설정 변경, 모니터링 기능 지원, 감사 로그 기능을 제공한다.

3.2. 웹 애플리케이션 방화벽 구조

본 연구에서 제안하는 웹 애플리케이션 방화벽 구조는 그림 1에서 보는 바와 같이 드라이버 레벨, 커널 레벨, 그리고 애플리케이션 레벨로 구분된다. 드라이버 레벨의 모듈은 리눅스 시스템에서 제공하는 드라이버를 하드웨어에 맞도록 수정하여 사용한다. 웹 애플리케이션 방화벽의 주요 모듈은 커널 레벨과 애플리케이션 레벨에서 구성되도록 한다.

커널 레벨에서는 방화벽 모듈, IPS 모듈, 그리고 본 연구에서 제안된 웹 컨텐츠 필터인 LAWCF(Lightweight Advanced Web Content Filter) 모듈로 구성한다.

3.2.1 커널 레벨 웹 방화벽 모듈

커널 레벨의 웹 방화벽 모듈은 다시 세부 모듈로 네트워크 수준 방화벽 모듈, IPS모듈, LAWCF 모듈로 구성된다. 본 연구에서는 일반적인 방화벽 모듈과 Snort기반 IPS 모듈을 사용하고 웹 필터 중심으로 설계한다[22].

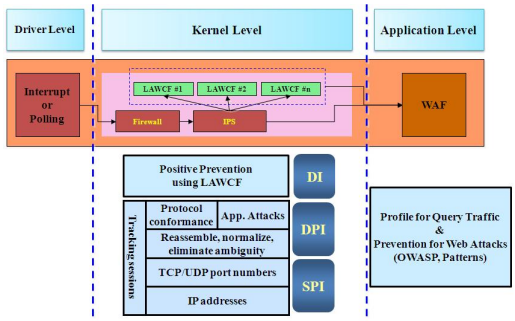

그림 1. 웹 애플리케이션 방화벽 구조 및 기능
Fig. 1. WAF Structure and Functions

3.2.1.1 LAWCF 모듈

HTTP 트랙픽에 대한 애플리케이션 데이터 처리를 위해서는 패킷의 애플리케이션 레벨로의 복사는 필수적이고 애플리케이션으로의 데이터 복사는 데이터 처리에 많은 부하를 야기한다. 이러한 문제를 해결하기 위해서 데이터 복사 횟수를 감소시킴으로써 패킷처리의 효율성을 가지도록 하는 것이 LAWCF의 목적이다. LAWCF는 특정 시점의 웹서버 트래픽 프로파일 정보를 참조하여 상대적으로 접속 비율이 높은 클라이언트 요청에 대한 웹 트래픽을 커널 레벨에서 처리하도록 한다. LAWCF는 웹 서버의 정상적인 트래픽 프로파일 정보를 이용하여 포지티브 보안 모델을 적용하여 공격 탐지를 수행한다[3,4,5,6,7]. 목적지 주소를 참조하고 이에 대한 HTTP 패킷 분석을 수행하는 LAWCF가 처리할 수 있도록 대기 버퍼에 쌓아놓는다. 각각의 LAWCF는 폴링하면서 해당 패킷의 정보를 분석하고 프로파일 정보를 이용하여 비정상행위를 탐지한다. LAWCF는 커널 쓰레드로 구동되며 사용자가 지정한 프로세서에 바인딩 될 수 있도록 한다. 커널 레벨에서 패킷의 처리 순서는 그림 2와 같다.

3.2.2 애플리케이션 레벨 웹 방화벽 모듈

애플리케이션 프락시(application proxy) 형태의 애플리케이션 레벨 웹 방화벽 기능 모듈은 웹 애플리케이션에 트래픽에 대한 프로파일 데이터 생성, 관리, 갱신 기능과 LAWCF로의 프로파일 정보 전송 기능, LAWCF에서 처리되는 트래픽보다 상대적으로 낮은 우선순위를 가지는 패킷에 대한 포지티브 보안 모델을 적용한 탐지 및 차단 기능, 서명기반 공격탐지 및 차단 기능, 관리시스템으로의 로그 전송 및 리포팅 기능을 제공한다[5].

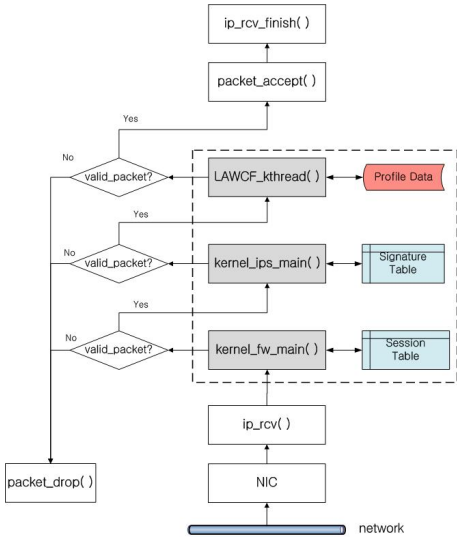

그림 2. 커널 수준에서 패킷 처리 흐름도
fig. 2. Packet Processing Flow Chart in Kernel Level

3.3 웹 클라이언트 요청 프로파일

실시간으로 웹 트래픽에 대한 프로파일 정보를 생성하고 적용할 수 있는 방법을 제안한다. 또한 보다 정확한 프로파일 정보를 생성하기 위하여 클라이언트의 주소 각각에 대한 URI(Uniform Resource Identifier) 파라미터 분석을 수행한다[15]. 클라이언트의 특정 URL 접속 요청 수에 대한 비정상 행위의 오탐률과 미탐률을 낮추고, 적은 비용으로 시스템 성능을 최대한 높일 수 있도록 설계한다.

3.3.1 트래픽 프로파일 정보

프로파일 정보는 일정기간의 실시간 웹 트래픽을 분석하여 테이블로 생성된다. 이렇게 만들어진 프로파일 테이블은 클라이언트의 주소와 웹 서버 주소, 그리고 URI 분석을 통하여 얻어지는 값으로 구성된다. 본 연구에서는 클라이언트의 요청(GET/POST 메서드)에 포함된 파라미터 값에 대한 길이를 이용한 탐지 방법을 이용한다. 각각의 파라미터 길이에 대한 실시간 프로파일을 만들고 이를 정상적인 트래픽의 임계치로 설정한다. 이때 클라이언트의 요청의 파라미터 길이가 임계치를 초과하면 비정상적인 클라이언트의 요청으로 탐지하는 방법을 사용한다. 보다 정확한 프로파일 정보를 생성하기 위한 방법으로 웹서버의 응답이 OK응답(HTTP 응답코드가 200~206일 때 즉, 웹 요청 성공)인 OK 응답만을 프로파일로 생성한다. 즉, 서버응답이 OK가 아니면 프로파일을 수행하지 않는다. 프로파일 기능이 활성화되어 있으면 서버로부터 해당 OK응답(HTTP OK응답)을 여부를 확인하고, 웹 요청 테이블에 유지하고 있던 요청들을 프로파일 테이블에 저장한다. 이때 프로파일 테이블에 이미 저장하려는 파라미터 이름과 동일한 길이가 존재하면 파라미터 정보에 히트 수를 증가한다. 파라미터 이름이 존재하지 않으면 파라미터 이름, 파라미터 값의 길이, 히트 수를 추가한다. 끝으로 클라이언트에 응답을 전달한다. 그림 3은 프로파일 수행의 과정을 보여준다.

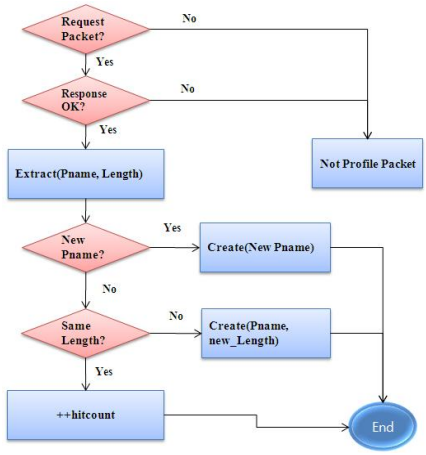

그림 3. 프로파일 생성 과정
fig. 3. Profile Generation Process

3.3.2 비정상 트래픽 탐지를 위한 임계치 설정

클라이언트가 요청한 파라미터 값의 길이별로 히트 수를 계산하여 정상적인 트래픽으로 판단하는 임계치로 사용한다. 히트 수가 큰 요청에 대해서는 커널 레벨에서 비정상행위를 탐지 및 차단을 위해 사용한다. 그 이유는 특정 URL의 히트 수가 다른 URL의 히트 수에 비해서 상대적으로 크다는 것은 해당 웹 서버에서 클라이언트의 요청이 다른 URL보다 상대적으로 많고 정상적인 트래픽으로 판단할 수 있기 때문이다. 상대적으로 높은 히트 수를 가진 클라이언트의 요청의 공격탐지를 위해서 모든 패킷을 애플리케이션 레벨로 복사하는 것은 시스템 처리 성능을 저하시키는 가장 주요한 원인이 되기 때문에 작은 히트 수를 나타내는 사용자 요청에 대해서만 애플리케이션 레벨에서 처리하여 전체적인 시스템 처리 성능을 높인다.

3.3.3 보안 관리 시스템

웹 애플리케이션 방화벽의 설정과 관리기능을 담당하는 보안 관리 시스템은 시스템 콘솔에서 CLI(Command Line Interface)를 지원하는 형태로 구현하는 방법과 독립된 단일

시스템 형태로 구현할 수 있다. 전자의 경우는 시스템에서 많은 정보를 유지해야하고 로그와 감사 기록 관리에 추가적으로 시스템 자원을 필요로 하기 때문에 실시간 공격 탐지 및 차단 성능에 영향을 줄 수 있다. 따라서 저 사양의 시스템을 전제로 한 본 연구에서는 독립된 보안 관리 시스템으로 설계한다. 보안 관리 시스템은 관리기능과 모니터링 기능의 두 가지 기능을 중심으로 설계한다. 관리기능은 웹 애플리케이션 방화벽의 기본 시스템 설정, 보안정책 설정, 공격패턴 설정, 프로파일 생성 및 적용, 웹 필터 설정 기능들로 구성된다. 모니터링 기능은 크게 로그와 실시간 모니터링 기능으로 구성한다. 시스템의 모든 동작과 이벤트에 대한 정확한 기록과 실시간 공격 탐지 및 차단 정보에 관련된 모니터링 기능은 최근의 모든 보안 시스템에서 필수 기능이다[5,10,11].

IV. 실험 및 평가

이 장에서는 제안된 웹 애플리케이션 방화벽에 대한 구현 방법과 실험 환경을 기술하고 실험 결과 분석을 통하여 제안된 시스템의 효용성을 검증한다. 제안된 시스템의 효용성을 검증하기 위해서 WASC(Web Application Security Consortium)의 웹 애플리케이션 방화벽 평가항목[6]으로 실험을 수행하였고, 특별히 구현된 시스템이 본 연구의 목적에 맞도록 구현되었는지 여부를 테스트하기 위해 다음 세 가지 항목을 실험하고 분석한다.

① 클라이언트의 요청 트래픽의 오탐률 측정
② 공인된 웹 성능 분석도구를 이용한 성능 측정
③ 리눅스 커널 기반의 레거시 시스템에 대한 적용 가능성

4.1 구현 환경

웹 애플리케이션 방화벽을 구현하기 위하여 리눅스 환경의 GNU CC 개발 툴을 이용하여 구현하였다[13]. 본 연구에서 제안한 웹 애플리케이션 방화벽을 구현한 시스템은 저 비용, 저 사양으로 리눅스 커널이 운영 가능한 하드웨어와 새로운 장비의 도입으로 사용되고 있지 않은 3개의 전용 방화벽 장비와 PC를 사용하였다. 시스템의 사양은 실험에서 상세하게 기술한다. 리눅스 커널 버전은 Kernel-2.6.0을 사용하였다. 관리시스템은 윈도우즈 운영체제와 로그 및 감사 기록 데이터의 관리를 위하여 MS-SQL 서버를 사용하였다.

4.2. 실험 및 결과

4.2.1 프로파일 데이터 생성

실험에 사용할 프로파일 데이터를 생성하기 위해 실제 운영환경에서 클라이언트의 요청 정보를 이용한다. 프로파일 데이터 생성의 환경 구성은 표 1과 같다.

표 1. 실험 환경
Table 1. Experimentation Environment

실험구성	운영환경	설명
클라이언트	Microsoft Windows	48대(48명)
웹 서버	5개의 사용자 게시판	IIS 서버, 오라클 서버
프로파일기간	2주(14일)	1일 시간: 8:30~ 17:00

4.2.2 오탐률 측정

오탐률 측정은 프로파일 기간의 데이터의 정확성을 판단하기 위해서 위에서 생성한 2주(14일)간의 프로파일 데이터를 사용하였다. 프로파일에 포함되지 않는 클라이언트의 모든 요청을 비정상 행위로 판단하였다. 제안된 시스템의 효용성을 증명하기 위해 오탐률 측정을 두 가지 설정 방법으로 수행하였다. 첫 번째 방법은 모든 프로파일 정보를 설정하고 오탐률을 측정하였고, 두 번째는 상위 10% 히트 수의 프로파일 만을 설정하고 오탐률을 측정하였다. 이를 통해서 성능을 보장하면서 정확하게 비정상 행위를 탐지하는가를 실험하였다. 오탐률은 1일 동안 발생한 총 히트 수에서 비정상으로 탐지된 요청 중 정상적인 요청의 비율로 계산하였다. 오탐률 측정은 모든 클라이언트의 요청이 애플리케이션 수준의 웹 애플리케이션 방화벽에서 처리되도록 설정하였다. 오탐률 계산식은 다음과 같다.

$$R_{fp} = \frac{C_n}{C_h} * 100 \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

R_{fp} = 1일 오탐지 율
C_h = 1일 총 히트 수
C_n = 1일 비정상 탐지 요청 중 정상 요청 수

그림 4는 모든 프로파일 정보를 가지고 탐지한 오탐률이다. 2주간 클라이언트의 정상적인 요청에 대한 오탐률을 측정한 결과, 평균 오탐률은 1.43%를 나타내었다.

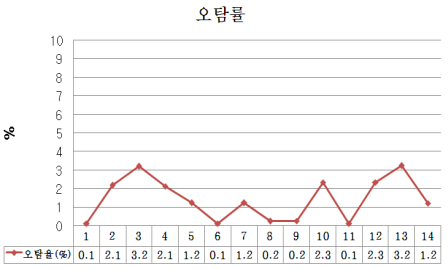

그림 4. 모든 프로파일을 적용한 오탐률(%)
fig. 4. False Positive for All Profile(%)

그림 5는 상위 10%의 프로파일 데이터만을 참조한 오탐률을 나타낸다. 평균 오탐률은 6.45% 로 나타났다.

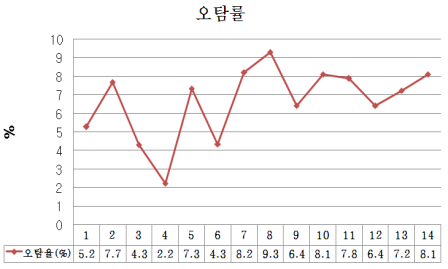

그림 5. 상위 10% 히트 수의 프로파일을 적용한 오탐률(%)
fig. 5. False Positive for Top 10% Profile (%)

4.2.3 시스템 성능 측정

세 가지 형태의 시나리오로 네트워크를 구성하여 제안된 시스템 성능을 측정하였다. 앞 절에서 생성된 프로파일 정보를 기반으로 하여 네트워크 시나리오 각각의 성능을 측정하였다. 성능측정 도구는 IXIA Aptixia IxLoad를 사용하였다. IxLoad는 웹 요청 패킷을 생성하여 성능을 측정 대상인 웹 애플리케이션 방화벽으로 전송한다. 웹 애플리케이션 방화벽의 처리 후에 IxLoad로 다시 돌아오는 응답 패킷의 비율로 시스템 성능을 측정하는 방법을 사용한다[19].

본 실험에서는 실제 환경과 동일한 네트워크 환경을 설정하여 HTTP 클라이언트로 구성하고 GET/POST 메서드 명령만을 수행하도록 설정하였다. 각각의 시나리오 별로 10회의 성능 측정을 수행하여 신뢰할 수 있는 측정값을 얻도록 하였다.

4.2.3.1 실험 1 - 저사양 PC

실험 1은 저 사양의 PC를 사용하여 제안된 모든 기능을 구현한 독립된 하나의 시스템으로 수행하였다. 표 2는 실험 1의 시스템 구성 및 기본 방화벽 성능을 나타낸다. 아무런 설정 없이 네트워크 방화벽 모듈만을 실행되도록 설정하고 측정한 값이다.

표 2. 실험 1의 시스템 구성과 기본 방화벽 성능
Table 2. System Architecture and Basic Firewall Performance of Experimentation 1

하드웨어 구성	방화벽 성능 (bypass설정)	비고
Intel Cerelon L420, 1GB RAM, 80GB HDD, 2*10/100MB NIC	53Mbps 2000 세션지원	PC 1대 모든 기능 구현

그림 6은 실험 1 환경에 대한 환경 구성을 보여주고 있다. 제안된 시스템을 웹서버와 클라이언트 사이에서 인라인 모드로 구성한다.

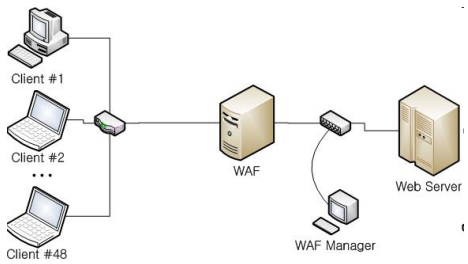

그림 6. 실험 1 환경
fig. 6. Environment of Experimentation 1

실험방법은 시스템을 4 가지 설정으로 구분하여 IXIA의 IxLoad로 수행하였다. 그림 7은 실험 1의 결과를 보여주고 표 3은 실험 1의 성능측정값의 평균값을 나타낸다.

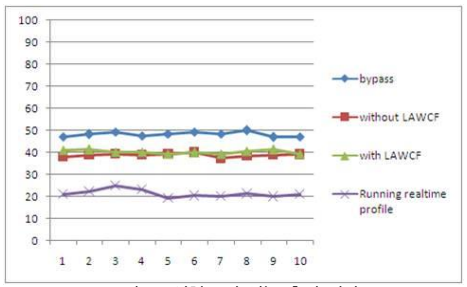

그림 7. 실험 1의 성능 측정 결과
fig. 7. Result of Performance for Experimentation 1

표 3. 실험 1의 성능 측정 값
Table 3. Value of Performance for Experimentation 1

WAF 설정	성능평균 (Mbps)	표준편차	비고
bypass 설정	48.23	0.96	룰 및 프로파일 설정없음
LAWCF 실행설정 무	38.84	0.70	
LAWCF 실행설정 유	40.25	0.75	상위 10% LAWCF에서 필터링
실시간프로파일실행	21.36	1.61	

표 3의 결과를 보면 실험 1에서 웹 애플리케이션 방화벽 모듈을 실행할 경우에 기존 bypass보다 9%의 성능 저하를 보여준다. 커널 수준의 웹 필터인 LAWCF를 사용하도록 설정한 경우 설정하지 않은 경우보다 약 4%의 성능 향상을 나타내었다. 실시간 프로파일의 적용을 설정한 경우에는 많은 성능 저하를 나타내었다.

4.2.3.2 실험 2 - 레거시 시스템

실험 2는 리눅스 기반의 레거시 시스템에 적용 가능하도록 본 연구에서 제안한 시스템을 구현하고 포팅하여 실험하였다. 표 4는 실험 2에 사용한 레거시 방화벽의 하드웨어 구성과 기본적인 방화벽 성능이다.

표 4. 실험 2의 시스템 구성과 기본 방화벽 성능
Table 4. System Architecture and Basic Firewall Performance of Experimentation 2

하드웨어 구성	방화벽 성능 (bypass)	비고
PPC 266MHz, 128MB RAM, 128 CF Memory, 4*100MB NIC	62 Mbps 20000 세션 지원	방화벽 장비1대 웹 방화벽의 모든 기능 포팅

그림 8은 실험2의 환경 구성을 그림 9는 실험 2의 실험결과 값을 나타내고 있다.

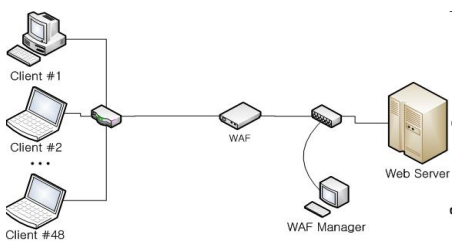

그림 8. 실험 2 환경
fig. 8. Environment of Experimentation 2

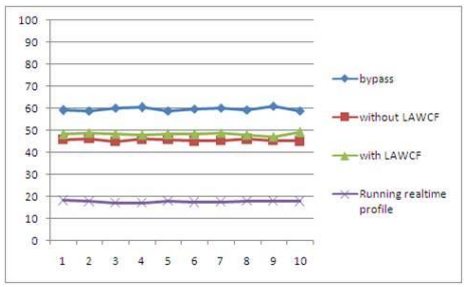

그림 9. 실험 2의 성능 측정 결과
fig. 9. Result of Performance for Experimentation 2

표 5는 실험 2의 10회 성능측정 실험의 평균값을 보여준다. 보안 정책이 없는 경우(bypass)는 거의 동일한 성능을 나타내었다. LAWCF룰 사용하도록 설정했을 때 설정하지 않은 경우보다 약 6%의 성능 향상을 나타내었다. 실시간 프로파일을 적용하였을 경우에 실험 1에 비하여 더 낮은 성능을 나타내는 이유는 실험 1에서는 프로파일 생성과 관리에 데이터베이스 시스템을 사용할 수 있었지만 실험 2에서는 시스템 특성상 데이터베이스 시스템을 사용할 수 없는 제약 때문이다.

표 5. 실험 2의 성능 측정 값
Table 5. Value of Performance for Experimentation 2

WAF 설정	성능평균 (Mbps)	표준 편차	비고
bypass 설정	59.70	0.70	룰 및 프로파일 설정없음
LAWCF 실행설정 무	45.61	0.43	
LAWCF 실행설정 유	48.31	0.54	상위 10% LAWCF에서 필터링
실시간프로파일실행	17.67	0.35	

4.2.3.3 실험 3 - 클러스터링 구성

실험 3은 여러 개의 레거시 시스템을 이용하여 가장 히트 수가 높은 URL을 분배하여 처리하도록 구성하여 실험하였다. 그림 10과 같이 방화벽에서 패킷을 분배하여 LAWCF에 전송하도록 하고, 애플리케이션 수준의 방화벽은 레거시 시스템에서 처리하지 않는 모든 요청에 대하여 필터링을 수행하도록 설정하였다. 표 6은 하드웨어 구성과 기본적인 방화벽 성능을 보여준다. 네트워크 방화벽에서는 클라이언트의 웹 요청에 대해서 세 개의 LAWCF를 기능을 수행하는 시스템으로 라운드 로빈 방법으로 분배한다. LAWCF는 모두 동일한 프로파일로 설정하였다. 본 실험에서는 히트 수의 전체 상위 순위 10%의 프로파일 테이블로 트래픽 필터링을 수행하였다.

표 6. 실험 3의 시스템 구성과 기본 방화벽 성능
Table 6. System Architecture and Basic Firewall Performance of Experimentation 3

하드웨어 구성	방화벽 성능 (bypass)	비고
Octeon 3120, 1GB RAM, 256 CF, 4*10/100TX	198 Mbps 100,000 세션	네트워크방화벽 기능 실행
PPC 266MHz, 128MB RAM, 128 CF, 4*10/100TX	59 Mbps 2,0000 세션	전용장비 LAWCF#1실행
PPC 300MHz, 128MB RAM, 128 CF 4* 10/100TX	59 Mbps 20,000 세션	전용장비 LAWCF#2실행
PPC 333MHz, 256MB RAM, 128 CF 4* 10/100TX	63Mbps 50,000 세션	전용장비 LAWCF#3실행
Intel Core2Duo 2.33Ghz, 2GB RAM, 80GB HDD, 4*10/100MB NIC	187Mbps 100,000 세션지원	애플리케이션 수준 방화벽 실행

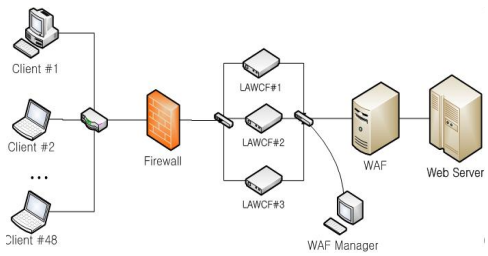

그림 10. 실험 3 환경
fig. 10. Environment of Experimentation 3

그림 11은 실험 3에 대한 성능 그래프이며 표 7은 성능 측정값에 대한 평균값을 나타낸다.

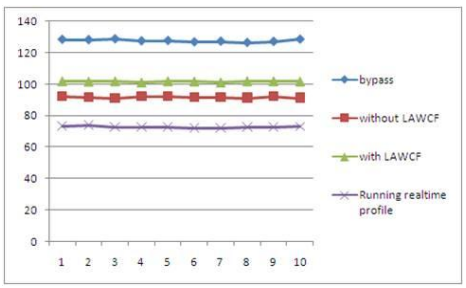

그림 11. 실험 3의 성능 측정 결과
fig. 11. Result of Performance for Experimentation 3

표 7. 실험 3의 성능 측정 값
Table 7. Value of Performance for Experimentation 3

WAF 설정	성능평균 (Mbps)	표준 편차	비고
bypass 설정	127.68	0.76	룰 및 프로파일 설정없음
LAWCF 실행설정 무	91.76	0.39	
LAWCF 실행설정 유	101.82	0.30	상위 10% LAWCF에서 필터링
실시간프로파일실행	72.95	0.49	

LAWCF를 적용하였을 경우에 미적용시 보다 약 9% 의 성능 향상을 보여주었다. 실시간 프로파일을 수행했을 경우에는 실험 1, 실험 2와 비교하여 현저한 성능 저하를 나타내었다.

4.2.3.4 실험 4 - 프로파일 변화에 따른 성능 측정

실험 1,2,3은 프로파일 테이블의 상위 히트수 10%만을 커널 수준의 LAWCF에서 필터링하도록 실험하였다. 실험 4에서는 커널에서 처리하는 프로파일 데이터를 10%씩 증가하면서 성능 측정을 수행하였다. 시스템 구성은 그림 10의 실험 2와 동일하게 설정하였다. 표 8은 커널에서 처리하는 프로파일 증가에 따른 성능측정값을 보여준다.

표 8. 실험 4의 성능 측정 값
Table 8. Value of Performance for Experimentation 4

LAWCF 프로파일(%)	애플리케이션 프로파일(%)	성능평균 (Mbps)	표준편차
0	100	45.61	0.42
10	90	48.31	0.53
20	80	48.34	0.49
30	70	48.51	0.48
40	60	49.09	0.50
50	50	49.89	0.49
60	40	49.19	0.45
70	30	50.11	0.49
80	20	47.23	0.51
90	10	46.90	0.47
100	0	46.12	0.52

LAWCF에서 처리하는 프로파일 테이블이 증가한다고 해서 항상 성능이 향상되지는 않았다. 실험 4에서는 70%의 프로파일을 커널에서 처리하는 것이 가장 우수한 성능을 나타내었다.

4.3 실험 분석

본 연구에서 제안된 프로파일 방법은 100건의 클라이언트 요청에 대해서 2주간의 모든 프로파일을 적용하였을 경우 평균 1~2건의 오탐을 보여주었으며, 상위 10%의 프로파일만을 적용하였을 때는 평균 6~7건의 오탐을 보여주었다. 현재 비정상행위 기반의 탐지를 사용하는 웹 애플리케이션 방화벽의 평균 오탐률에 대한 공식적인 통계가 없기 때문에 제안된 프로파일 기법의 오탐률을 정확하게 평가할 수는 없다. 하지만 2주 동안의 프로파일 기간 동안에 한 번도 접속하지 않은 정상적인 요청들에 대해서만 비정상 행위로 탐지하기 때문에 프로파일 기간을 어느 정도 연장한다면 보다 낮은 오탐률을 제공할 것으로 기대된다. 시스템 성능 면에서도 애플리케이션 수준으로 모든 웹 요청을 복사해야 하는 문제점을 해결하기 위해 제안된 LAWCF는 적용하지 않은 시스템보다 평균 6%의 성능 향상을 나타내었다. 또한 프로파일 정보 중에서 상위 70%를 커널 수준에서 처리하도록 설정하여도 시스템 성능을 보장할 수 있음을 보여주었다. 마지막으로, 레거시 시스템을 이용하여 웹 애플리케이션 방화벽을 구현하여 저 비용으로 우수한 성능을 제공할 수 있음을 보여주었다. 현재 대부분의 네트워크 장비들이 리눅스기반의 커널 구조를 가지고 있는 점을 감안할 때 쉽게 이식할 수 있을 것으로 기대된다. 따라서 저사양의 시스템에서 상위 10% 프로파일을 적용하여 성능을 보장하면서 효과적으로 비정상행위를 탐지할 수 있을 것으로 예상된다. 그러나 동적으로 실시간 프로파일을 적용하도록 설정하면 클라이언트의 실시간 요청을 반영한 프로파일 정보를 적용할 수 있었지만 프로파일 정보의 데이터 처리로 발생하는 성능저하로 전체 시스템 성능을 보장할 수 없었다.

V. 결론

본 연구에서는 다양해진 웹 공격과 침해사고로 큰 관심의 대상이 되고 있는 웹 애플리케이션의 방화벽을 제안하였다. 제안된 시스템은 정상적인 클라이언트 요청에 대한 분석으로 만들어진 프로파일 기반의 비정상행위 탐지를 수행하는 웹 애플리케이션 방화벽이다. 클라이언트의 웹 요청을 정확하게 분석하기 위해 정상적인 트래픽에 포함된 URI 파라미터 값의 길이에 대한 실시간 분석을 이용하였다. 이를 기반으로 포지티브 보안 모델을 적용한 비정상행위 탐지를 수행하였다.

제안된 프로파일 방법의 효용성을 증명하기 위해 실제 운영되고 있는 환경에 적용하여 실험하였다. 그 결과 제안된 프로파일 기법이 1일 트래픽의 1.43%의 오탐률을 나타내는 것을 알 수 있었다. URI 파라미터 값의 길에 대한 프로파일 정보 중에서 히트 수가 높은 웹 요청에 대하여 빠르게 처리할 수 있도록 커널 레벨의 LAWCF(Lightweight Advanced Web Content Filter)를 제안하였다. 히트 수가 높은 웹 요청은 애플리케이션 레벨까지 패킷을 복사하지 않고 커널 레벨에서 LAWCF가 처리하도록 하였다. 웹 애플리케이션 방화벽 성능 측정 도구를 이용한 성능 테스트에서 LAWCF를 사용하지 않은 경우보다 6%의 성능 향상을 보여주었다. 하지만 실시간 프로파일 적용 시에는 안정적인 성능을 제공하지 못했다. 또한, 리눅스 커널 기반의 시스템 설계와 구현을 통하여 기존의 리눅스 기반의 레거시 시스템들에 쉽게 이식할 수 있음을 보여주었다. 레거시 시스템을 클러스터링 형태로 구성하여 성능의 향상을 가져올 수 있도록 하였다. 이는 저가의 시스템과 레거시 시스템들에 쉽게 웹 필터를 적용하여 소규모 웹 서비스를 제공하는 네트워크와 기존 장비로 추가적인 장비 도입에 부담이 되는 비용 문제를 해결할 수 있도록 하였다.

참고문헌

〔1〕 Christopher Kruegel, Giovanni Vigna, "Anomaly detection for Web-based attacks," Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS'03), ACM Press, pp. 251-261. 2003.

〔2〕 InSeon Yoo, "Protocol Anomaly Detection and Verification," Proceedings of the 2004 IEEE Workshop on Information Assurance and Security, pp. 30-37, 2004.

〔3〕 황순길, 김관진, "*웹 해킹 패턴과 대응*," 사이텍미디어, 2005.

〔4〕 이병일, "*네트워크 트래픽 추이 분석 웹기반 비정상행위 탐지 시스템의 설계 및 구현* 홍익대학교 대학원 석사논문, 2005년 12월.

〔5〕 Web Application Firewall Evaluation Criteria, Web Application Security Consortium , Jan. 2006.

〔6〕 장성민, 김효남, 원유헌, "HTTP 트래픽 기반의 비정상행위 탐지 시스템," 한국정보과학회 한국컴퓨터종합학술대회 논문집C, 33권, 313쪽, 2006년 6월.

〔7〕 Jianning Mai , ChenNee Chuah , Ashwin Sridharan ,Tao Ye, Hui Zang, "Is Sample Data Sufficient for Anomaly Detection," IMC'6, Oct. 2006.

〔8〕 정보보호21C 4월호, Infothe Media Group, 46쪽, 2007년 4월
〔9〕 장성민, 오형준, 안현태, 원유헌, "내부자 감사를 위한 클라이언트-서버 구조의 네트워크 트래픽 포렌식 시스템의 설계," 한국정보과학회 한국컴퓨터종합학술대회 논문집 A, 34권, 113쪽, 2007년 6월.
〔10〕 Karen Scarfone, Peter Mell, "Guide to Intrusion Detection and Prevention Systems," Recommendations of the National Institute of Standards and Technology, Feb. 2007.
〔11〕 Richard Reiner, "Secure Software Development Methodology and Implementation," FISCON 2007, Oct. 2007.
〔12〕 Sung-Min Jang, Yoo-Hun Won, "Web Anomaly Detection System for Mobile Web Client," IEEE CS Conference FGCN2007, Dec. 2007.
〔13〕 Hypertext Transfer Protocol - HTTP/1.1, R. Fielding et al., RFC 2008, June 1997.
〔14〕 Angela Orebaugh, Gilbert Ramirez, and Jay Beale, Snort 2.1 Intrusion Detection, Second Edition, Syngress, May 2004.
〔15〕 SANS Website, *http://www.sans.org*
〔16〕 정보보호21C 3월호, Infothe Media Group, 56쪽-58쪽, Mar. 2008.
〔17〕 IXIA Aptixia IxLoad pages, *http://www.ixiacom.com*
〔18〕 Alan Murphy and Ken Salchow, "Applied Application Security- Positive and Negative Efficiency," F5 Networks, Oct. 2007.
〔19〕 정보보호21C 6월호, Infothe Media Group, 54쪽-62쪽 June 2008.
〔20〕 Snort Home pages, *http://www.snort.org*
〔21〕 OWASP's Ten Most Critical Web Application Security Vulnerabilities, *http://www.owasp.org*

저 자 소 개

장 성 민

1997: 충주대학교 이학사.
1999: 홍익대학교 이학석사.
2008: 홍익대학교 이학박사
2007-현재: 어울림정보기술(주) 기술연구소책임연구원
관심분야 : 웹보안, 네트워크보안, 정보보호시스템, DDoS보안, 차세대 네트워크보안

원 유 헌

1972: 성균관대학교 이학사.
1975: KAIST 이학석사.
1985: 고려대학교 이학박사
1976-현재: 홍익대학교 컴퓨터공학과 교수
관심분야 : 프로그래밍언어, 보안프로그래밍, 정보보호, 차세대 네트워크보안