웹 어플리케이션 보안을 위한 가상화 기반 보안 모델★

양환석* · 유승재**

요 약

네트워크 기술의 빠른 발전과 컴퓨팅 환경의 변화로 인하여 웹 어플리케이션 활용 분야가 최근 몇 년 사이에 광범위하게 넓어졌으며 복잡해졌다. 이러한 웹 어플리케이션이 중요한 서비스에 많이 사용되면서, 이를 대상으로 한 공격도 증가하고 있으며, 그 방법도 다양해지고 지능화되고 있다. 본 논문에서는 웹 어플리케이션의 취약점을 이용한 공격을 막기 위해 가상화 기술을 이용한 보안 모델을 제안하였다. 제안한 모델에서는 클라이언트 요청에 의해 생성되는 세션에 ID를 부여한 후 해당 요청에서 쿼리 유형을 분석하여 해당 가상 웹 서버에 전달함으로써 데이터베이스 서버에서도 쿼리에 대한 요청 정보를 인지할 수 있도록 하였다. 그리고 가상 웹 서버들 사이의 트래픽을 감시하고, Host OS의 자원 낭비를 줄이기 위해 VM-Master 모듈을 구성하였다. 제안한 기법의 공격탐지 및 자원 활용의 우수한 성능은 실험을 통하여 확인할 수 있었다.

A Study on Secure Model based Virtualization for Web Application Security

Hwan Seok Yang* · Seung Jae Yoo**

ABSTRACT

Utilization of web application has been widely spread and complication in recent years by the rapid development of network technologies and changes in the computing environment. The attack being target of this is increasing and the means is diverse and intelligent while these web applications are using to a lot of important services. In this paper, we proposed security model using virtualization technology to prevent attacks using vulnerabilities of web application. The request information for query in a database server also can be recognized by conveying to the virtual web server after ID is given to created session by the client request and the type of the query is analyzed in this request. VM-Master module is constructed in order to monitor traffic between the virtual web servers and prevent the waste of resources of Host OS. The performance of attack detection and resource utilization of the proposed method is experimentally confirmed.

Key words : Web Application, Virtualization, Security, Intrusion Detection

접수일(2014년 5월 30일), 수정일(1차: 2014년 6월 19일), 게재확정일(2014년 6월 27일)

* 중부대학교/정보보호학과
** 중부대학교/정보보호학과(교신저자)

1. 서 론

최근들어 개발되고 있는 대부분의 웹 사이트는 웹 어플리케이션에 의해 생성되는 동적인 웹 페이지 방식을 적용하고 있다. 특히 요즘의 웹 어플리케이션은 단순히 웹 페이지만을 전송하는 것이 아니고 데이터베이스 서버에 접속하여 수많은 데이터를 저장 및 전송을 실행하게 된다. 이러한 특성을 이용한 공격들이 나날이 증가하고 있으며, 더욱 지능화되고 있다. 실제 웹 공격의 70% 이상이 어플리케이션 레벨에서 이루어지고 있으며, 웹 서버의 포트가 많은 공격에 노출이 되어있는 실정이다[1][2]. 따라서 안전한 웹 어플리케이션 서비스 제공과 신뢰성을 향상시키기 위해서는 웹 어플리케이션 취약점 분석 및 보안 모델 개발이 반드시 필요하다.

본 논문에서는 웹 어플리케이션의 안전성과 신뢰성을 향상시킬 수 있는 보안 모델 구조를 제안하였다. 제안한 방법에서는 클라이언트들의 요청들에 의해 생성되는 각 세션에 ID를 부여하여 분리한 후, 세션 내의 웹 요청에 포함되어 있는 쿼리 유형을 분석하게 된다. 분석된 쿼리 유형에 따라 이를 처리하는 가상 웹 서버에 세션을 매핑시키는 기법을 적용하였다. 이렇게 함으로써 데이터베이스 서버에서도 처리되는 쿼리가 단순히 단일 웹 서버에서 수신되는 쿼리가 아니고 이들의 관계를 명확히 구분할 수 있게 된다. 그리고 가상 웹 서버들간의 내부 트래픽을 감시를 통해 비정상행위를 탐지하고, Host OS의 자원 낭비를 줄이기 위하여 VM-Master 모듈을 구성하였다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 어플리케이션의 취약점에 대하여 살펴보고 3장에서는 본 논문에서 제안한 가상화 기반 보안 모델에 대하여 기술하였다. 4장에서는 제안한 보안 모델의 성능 평가를 위해 실험하고 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 웹 어플리케이션의 취약점

웹 어플리케이션은 웹 사이트, 인터넷 쇼핑몰 등과 같은 특정 서비스 제공을 목적으로 만들어진 웹 기반 프로그램을 의미한다. 이러한 웹 어플리케이션이 기본 포트로 사용하는 80, 443 포트를 이용해 악의적인 침투가 가능한 것을 웹 어플리케이션 취약점이라 한다[3]. 이러한 공격에 효과적으로 대응하기 위해서는 웹 서버나 웹 어플리케이션의 취약점을 사전에 점검, 분석하여 차단해야 한다. 웹 어플리케이션의 취약점을 이용한 공격들은 대부분 악성코드 유포, 금전적 이득 등의 목적을 가지고 발생하고 있으며, 그 수법은 갈수록 진화하고 있다. 국제 웹 보안 표준 기구인 OWASP(Open Web Application Security Project)에서 발표한 웹 어플리케이션 취약점 Top 10은 <표 1>과 같다.

<표 1> OWASP 10대 취약점

구분	항목
A1	인젝션(Injection)
A2	클로스 사이트 스크립팅(XSS)
A3	취약한 인증과 세션 관리
A4	안전하지 않은 직접 객체 참조
A5	크로스 사이트 요청 변조 (CSRF)
A6	보안상 잘못된 구성
A7	안전하지 않은 암호 저장
A8	URL 접근 제한 실패
A9	불충분한 전송 계층 보호
A10	검증되지 않은 리다이렉트와 포워드

2.2 공격 유형

인젝션 취약점은 SQL문으로 해석되는 스크립트 구문을 삽입하여 데이터베이스에 접근할 수 있는 취약점을 말하며, 종류로는 SQL Injection, Blind SQL Injection, Malicious Code Injection 등이 있다. 이 공격은 데이터베이스 자료를 검색하여 웹 화면에 표시하거나, 로그인 절차 우회 및 중요 정보 노출 등이 가능하다[4][5].

크로스 사이트 스트립팅(XSS) 취약점은 연결이 유지되지 않는 HTTP 프로토콜의 특징을 이용한다. 즉, 클라이언트에서 실행되는 코드를 사용자 입력으로 주게 되면, 이 코드가 클라이언트측 브라우저에서 수행되는 특징을 이용한 공격이다[6].

악성파일 실행 취약점은 사용자의 컴퓨터에서 악성

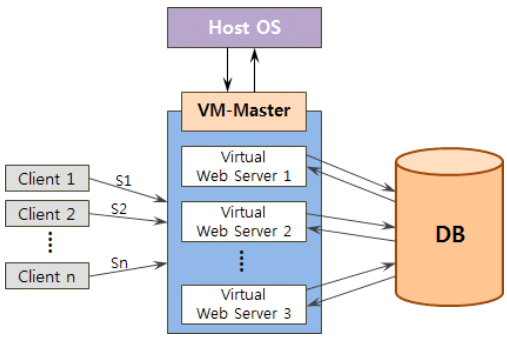
코드를 실행하여 공격자는 원격 코드 실행, 원격 루트킷 설치, 전체 시스템 손상 등을 수행할 수 있게 된다. 이 공격은 PHP, XML, 사용자로부터 파일명이나 파일을 받아들이는 프레임워크에 영향을 준다[7].

3. 제안한 가상화 기반 보안 모델

본 장에서는 신뢰성 높은 웹 어플리케이션 서비스 제공을 위하여 독립된 가상화 보안 모델을 제안하였다.

3.1 보안 모델 구조

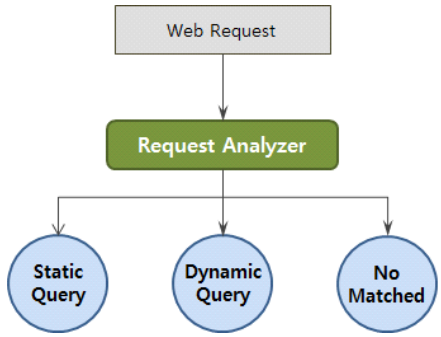
본 논문에서는 3-tier 구조의 웹 어플리케이션 취약점을 향상시킬 수 있는 보안 모델을 제안하였다. 특정 공격에 의해 웹 어플리케이션 서비스가 중단되는 것을 막기 위하여 웹 서버 가상머신을 이용하였으며, 클라이언트들의 요청에 의해 생성되는 각 세션에 ID를 할당하여 독립된 가상 웹 어플리케이션 환공을 제공하였다. 이렇게 함으로써 웹서버와 데이터베이스 트래픽을 고려한 매핑이 이루어져 세션 하이재킹과 같은 공격을 차단할 수 있는 방법을 제공한다. 그리고 클라이언트 요청을 처리하기 위해 생성된 웹 서버 가상머신들간의 모니터링을 위한 VM-Master 모듈을 추가하였다. 이 모듈은 악성코드와 같은 공격으로 인한 피해를 막기 위하여 내부 가상 웹 서버들을 감시하고, Host OS와의 보안 통신을 제공해 준다. <그림 1>은 본 논문에서 제안한 보안 모델의 구조를 보여주고 있다.



(그림 1) 제안한 보안 모델 구조

3.2 세션 매핑 기법

네트워크를 통해 웹 서버로 들어오는 공격 트래픽은 웹 서버의 모든 세션에 영향을 주고, 웹 서버의 자원을 고갈시키게 된다. 따라서 자원 사용의 오버헤드를 줄이면서 시스템의 보안 성능을 향상시키기 위하여 경량 가상화 기술을 적용하였다. 그리고 클라이언트들의 비정상행위를 탐지하기 위하여 요청 정보를 세션으로 구분하였다. 또한 데이터베이스서버에서는 클라이언트들의 요청들에 대해서 세션으로 구분하지 못하고 같은 웹서버로부터 수신한 쿼리로만 인식하게 된다. 따라서 본 논문에서는 데이터베이스 서버에서도 이들간의 관계를 구분할 수 있도록 클라이언트와의 모든 통신은 서로 구분될 수 있는 세션으로 분리하며, ID를 부여한다. 이렇게 분리된 각 n개의 세션내의 웹 요청은 쿼리가 존재하는 경우와 그렇지 않는 경우로 분류할 수 있다. 그리고 쿼리가 존재하는 경우는 다시 하나의 쿼리로 정확한 결과를 보내줄 수 있는 정적인 쿼리, 여러 개의 쿼리 또는 매개변수에 따라 결과가 달라지는 동적인 쿼리, 일치하지 않는 쿼리로 분류된다. <그림 2>는 웹 요청 내에 존재하는 쿼리의 패턴 종류를 보여주고 있다.



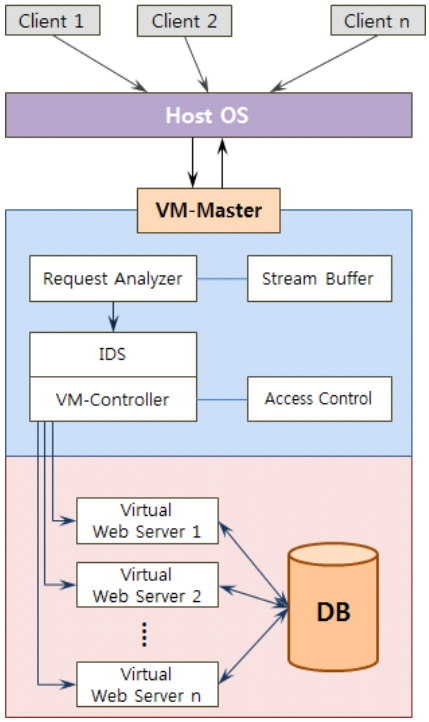
(그림 2) 쿼리의 패턴 종류

먼저, VM-Master에서 클라이언트의 요청을 수신하면 해당 요청에서 쿼리의 포함 여부를 검사한 후, 만약 포함되어 있지 않다면 해당 서비스를 제공해주면 된다. 그러나 요청에 쿼리가 포함되어 있다면 세 가지 종류 중 어디에 해당이 되는지 검사한 후에 해

당 가상 웹 서버에 전달해주면 된다. 이렇게 함으로써 데이터베이스 서버에서도 요청받은 쿼리를 구분할 수 있게 되어 여러 공격에 대응할 수 있게 된다.

3.3 VM-Master 모듈

웹 어플리케이션의 안정된 서비스를 제공하고 가상화 보안을 향상시키기 위하여 VM-Master 모듈을 제안하였다. 만약 Host OS가 보안을 위해 모든 가상화를 제어하는 기법을 적용한다면 많은 시스템 자원이 낭비될 것이다. 따라서 VM-Master 모듈을 적용하여 Host OS의 오버헤드를 줄여주고 게스트 가상머신들에게 제공되는 서비스들의 트래픽을 감시함으로써 침입탐지 기능을 갖게 된다. <그림 3>은 VM-Master의 구조를 보여주고 있다.



(그림 3) VM-Master 구조

<그림 3>에서 Request Analyzer는 각 세션내의 웹 요청에서 쿼리의 포함 여부 및 해당 패턴의 종류를 분석하게 된다. 분석 결과에 따라 해당 패턴을 처리하는 가상 웹 서버에게 쿼리를 전달하게 된다. 이를 수신한 가상 웹 서버에서 데이터베이스 서버에게 쿼리를 전송하여 처리하게 된다. Request Analyzer가 가지고 있는 stream buffer는 클라이언트가 수신한 요청에 공격을 탐지하기 위해서 제공한다. 그리고 VM-Controller에서는 가상 웹 서버에 대한 관리 기능을 수행하며 침입탐지 기능을 가지고 있어 내부 가상 웹 서버들 간의 비정상행위를 탐지하게 된다.

4. 실험 및 결과

4.1 실험 환경

본 논문에서 제안한 가상화 기반 보안 모델에 대한 성능을 평가하기 위하여 <표 2>와 같은 시스템 환경에서 실험하였다. 그리고 실험에 사용할 동적인 웹 사이트의 구축은 Wordpress를 이용하여 구축하였다. 그리고 실험에 사용한 공격은 XSS와 SQL Injection을 5분 동안에 30번의 공격을 시도하였다.

<표 2> 실험 시스템 환경

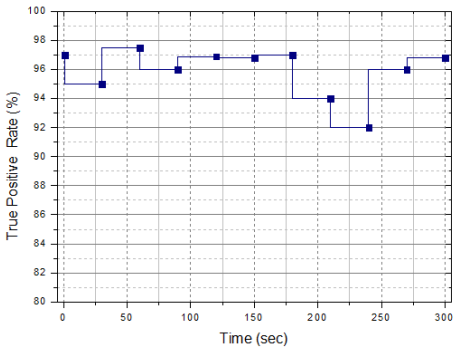
Parameter	Value
CPU	i7-4770K(3.5GHz)
Memory	32GB
Host OS	CentOS 6.5
Database	MySql
Web Server	Apache

4.2 실험 결과 분석

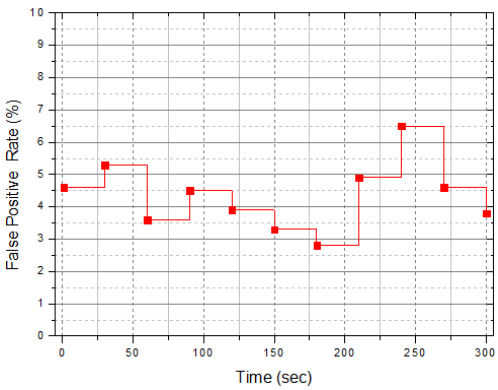
본 논문에서 제안한 보안 모델의 성능 평가를 위한 기준은 침입탐지 성능을 확인하기 위하여 True Positive Rate(TPR)과 False Positive Rate(FPR)을 측정하였으며 웹 요청시의 시스템 오버헤드를 측정하였다. 먼저 TPR과 FPR은 다음 식 1과 식 2로 계산된다.

$$TPR = \frac{TP(True\ \ Positive)}{TP + FN(False\ \ Negative)} \quad (1)$$

$$FPR = \frac{FP(False\ \ Positive)}{FP + TN(True\ \ Negative)} \quad (2)$$

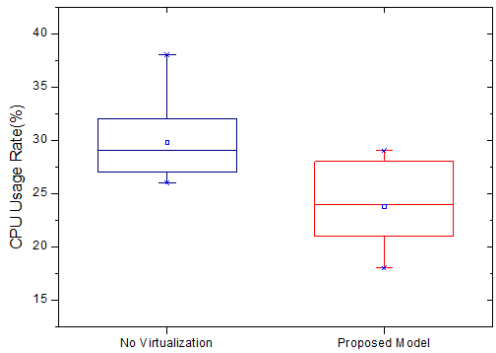


(그림 4) 공격 탐지율(TPR)



(그림 5) 공격 오탐지율(FPR)

공격탐지의 성능은 제안한 보안 모델에서 클라이언트 요청에 의해 생성되는 세션을 ID로 구분하여 쿼리를 분석 및 처리하는 효율성과 VM-Master 모듈의 성능을 확인하게 되는 것이다. <그림 4>와 <그림 5>에서도 확인할 수 있듯이 TPR과 FPR은 우수한 결과를 보여주고 있다.



(그림 6) 자원 사용율 비교

<그림 6>은 웹 요청에 따른 시스템 오버헤드를 측정 결과이다. 가상화 기술을 적용하지 않은 상태에서 요청이 발생했을 때와 가상화를 적용한 후 요청에 대한 처리 오버헤드를 비교하였을 때 제안한 보안 모델이 약 8.6% 우수한 결과를 보여주었다.

5. 결 론

네트워크 기술의 빠른 발전으로 인하여 대부분의 컴퓨팅 환경이 웹 기반으로 통합되어 운영되고 있으며, 이를 활용하는 웹 어플리케이션의 종류 또한 다양해지고 있다. 하지만 웹 어플리케이션의 취약점을 이용한 공격 또한 갈수록 증가하고 있는 실정이다.

본 논문에서는 웹 어플리케이션의 취약점에 대비하고 신뢰성을 향상시키기 위하여 가상화 기술을 적용한 보안 모델을 제안하였다. 웹 요청과 쿼리를 검사하는 메커니즘이 존재한다 하더라도 엄청난 양의 실시간 트래픽을 검사하기가 쉽지 않다. 따라서 네트워크를 통해 들어오는 악의적인 트래픽 검색을 쉽게 하기 위하여 클라이언트의 요청에 의해 생성되는 세션에 ID를 할당하여 각 세션을 구분하였으며 해당 세션내의 요구에 포함되어 있는 쿼리의 유형을 분석하였다. 분석된 쿼리의 유형에 따라 해당 쿼리를 처리하는 웹 서버에 전달되며 이를 처리하는 웹 서버는 가상화 기술을 이용한 가상 웹 서버로 구성하였다. 그리고 VM-Master 모듈을 구성하여 가상 웹 서버들 사이의 비정상행위를 탐지하고, Host OS의 오버헤드를 줄이

는 방법을 사용하였다. 실험을 통해 제안한 기법의 공격탐지 및 자원 활용 면에서 우수한 성능을 확인할 수 있었다.

참고문헌

[1] Adam Kie˙zun, Philip J. Guo, Karthick Jayaraman, Michael D. Ernst, “Automatic Creation of SQL Injection and Cross-Site Scripting Attacks,” ICSE“09, May 16-24, 2009,

[2] Gonzalez, J. M., Paxson, V., and Weaver N., “Shunting: a hardware/so ware architecture for exible, high-performance network intrusion prevention,” In Conference on Computer and Communications Security (CCS), pp.129 - 149, 2007.

[3] Fogla, P. and Lee, W., “Evading network anomaly detection systems: formal reasoning and practical techniques. In Proc of. ACM Conference on Computer and Communications Security (CCS), pp.59-68, 2006.

[4] D. Scott and R. Sharp, "Avstracting Application-Level Web Security," Proc. 11th Int'l Conf. World Wide Web, pp.396-407, 2002.

[5] Kruegel C., Mutz D., Valeur F., and Vigna G., “On the detection of anomalous system call arguments,” In Proc. of European Symposium on Research in Computer Security, pp.326 - 343, 2003.

[6] W. R. Cook and S. Rai, "Safe Query Objects: Statically Typed Objects as Remotely Executable Queries," Proc. 27th Int'l Conf. Software Eng. pp.97-106, 2005.

[7] Klo M., Brefeld U., Dusse, P., Gehl C., and Laskov P., “Automatic feature selection for anomaly detection,” In Proc. of ACM Workshop on Artical Intelligence for Security (AISEC), pp.71-76, 2008.

[저 자 소 개]

양 환 석 (Hwan-seok Yang)

1998년 2월 조선대학교 이학석사
2005년 2월 조선대학교 이학박사
2007년 3월 호원대학교 연구교수
2011년 9월 현재 중부대학교 정보보호학과 조교수

email : yanghs@joongbu.ac.kr

유 승 재 (Seung-jae Yoo)

1988년 2월 동국대학교 이학사
1990년 2월 동국대학교 이학석사
1998년 2월 동국대학교 이학박사
1997년 3월 현재 중부대학교 정보보호학과 교수

email : sjyoo@joongbu.ac.kr