가상화를 이용한 웹 서버 보안시스템 설계 및 구현*

유재형** · 김도형** · 김용호** · 하옥현*** · 김귀남**

요 약

웹 서비스는 기능의 특성상 다른 서비스와는 달리 외부에 노출되어 있고 다양한 어플리케이션들이 웹 서비스와 연동되어 있어서 많은 보안 취약점들이 존재한다. 특히 새로운 웹 기술들이 개발되면서 전에 없던 새로운 형태의 보안 취약점들이 꾸준히 생겨나고 있다. 본 논문에서는 이러한 취약점들을 바탕으로, 가상화 환경을 이용하여 웹서버와 허니웹을 구축함으로써 어떤 공격에 대해서도 시스템의 하드웨어까지 영향을 미치지 않도록 구성되며 허니웹을 통하여 새로운 공격에 대해서도 정보를 수집할 수 있도록 웹 서버 보안시스템을 설계 및 구현 하였다. 이를 통하여 상호 통신의 웹 환경에서 적절한 보안을 제공 할 수 있다.

Design and Implementation of Web Server Security System using Virtualization

Jae Hyung Yoo** · Do Hyung Kim** · Yong Ho Kim**
Ok Hyun Ha*** · Kuinam J. Kim**

ABSTRACT

Web service has many security weekness because it is exposure to outside and connected with various application. Especially, as new technology developed new type of security weakness has occured consistently. In this paper, we construct webserver and honeyweb by using virtual reality on a basis these weakness. So it cannot be influenced by any attack to the hardware of the system. By using honey web, it disigned and embodied web server secutiry system to collect the data about new attack. Through this, it can provide proper secutiry in a web environment of mutual communication.

Key words : Web Server, IDS, honeyweb, Virtualization

* 본 연구는 지식경제부 지역혁신센터사업인 산업기술보호특화센터 지원으로 수행되었음
** 경기대학교 정보보호학과
*** 호남대학교 경찰학과

1. 서 론

WWW(World Wide Web)은 1989년 스위스 제네바에 있는 유럽 원자핵 공동 연구소(CERN)의 팀 버너스리(Tim Bernars-Lee)가 제안한 것으로 하이퍼텍스트(hypertext)라는 기능에 의해 인터넷상에 분산되어 존재하는 온갖 종류의 정보를 통일된 방법으로 찾아볼 수 있게 하는 광역 정보 서비스 및 소프트웨어이다. 웹은 Text기반의 통신에서 벗어나 새로운 방식인 그림, 소리, 영상 등 다양한 형태로 보여줌으로써 새로운 인터넷시대를 열었으며 인터넷 환경을 더욱 친숙하게 만들어 인터넷을 활성화하는데 크게 기여하였다[1]. 웹의 탄생은 인터넷의 발전을 가속화하였으며 웹은 대중화되고 현대 사회에서는 없어서는 안 될 중요한 자리를 차지하게 되었다.

그러나 이러한 편리성과 익명성을 이용한 해킹 및 악성코드 유포와 같은 사이버위협은 계속 증가하고 있으며, 웹의 보안에 대한 이슈는 계속적으로 발생하고 증가하고 있다. 이로 인한 피해는 말로 표현할 수 없을 만큼 크다. 웹 해킹에 의해 기업의 비즈니스가 피해를 입고, 개인정보가 유출이 되어 개인의 프라이버시가 침해가 되고, 금융정보가 유출이 되어 많은 금전적인 피해 등을 입게 된다. 이에 대한 대응으로 웹 방화벽, 웹 스캐너, 웹 프로그래밍 소스 검사툴, 안전한 웹 프로그래밍 가이드 발간 등의 방안 및 활동이 있으나 나날이 발전되는 웹 해킹에 대한 대응에는 아직까지 부족함이 많다. 웹서비스에 있어서 내부 네트워크는 각종 보안장비 도입/운영으로 보호할 수 있으나, 웹 서비스는 외부에 공개되는 특성상 해킹사고 및 공격이 끊이질 않고 있는 실정이다.

본 연구에서는 웹의 발전과 이에 따르는 위험을 살펴보고 웹 서버를 보호하는 웹 방화벽과 호스트 기반의 IDS에서 패턴을 활용한 보안과 가상화를 활용한 새로운 공격 패턴에 대하여 통제 할 수 있도록 honeyweb을 운용한 공격 패턴 수집을 통하여 공격에 대항할 수 있는 보안시스템을 설계 및 구현하고자 한다.

2. 웹과 웹 서버 보안

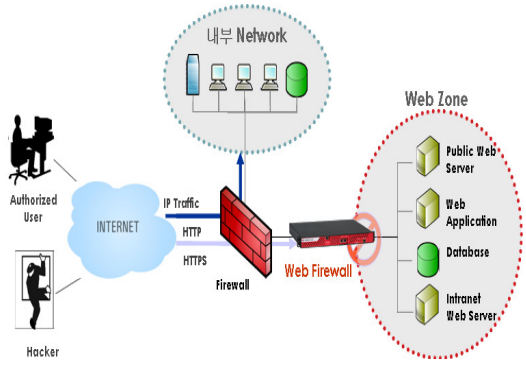
2.1 웹의 발전과 위협

초기의 웹은 단지 웹 서버를 통해 하이퍼텍스트(hypertext)를 보여주는 수준이었다. 그러다가 1990년대에 들어오면서 좀 더 역동적이고 동적인 웹을 보여주기 위해 CGI기술이 도입이 되었다. 1990년대 후반부터는 CGI뿐만 아니라 ASP, Perl, PHP, JSP와 같은 서버측면에서의 다양한 스크립트 기술이 개발이 되면서 웹 기술은 급속도로 발전을 하였다. 2000년대 들어오면서 웹은 서비스 통합형으로 진화하기 시작하였다. 이기종간에 구현되는 다양한 어플리케이션의 일관된 데이터 포맷을 유지시켜주는 XML(Extensible Makeup Language), 내부 및 외부의 여러 어플리케이션 간의 메시지를 전송하는 규약을 정한 SOAP(Simple Object Access Protocol), 협력자와 상호 호환되는 서비스를 정의하고 사용 방법을 기술한 WSDL(Web Service Description Language) 등의 기술을 통해 여러 Application들이 하나의 웹 플랫폼으로 통합되어 웹 기반 통합서비스를 제공할 수 있게 되었다.

웹 서비스는 기능의 특성상 다른 서비스와는 달리 반드시 외부에 노출되어 있어야 하고 방화벽의 보호를 받기 어렵다. 그리고 다양한 어플리케이션들이 웹 서비스와 연동되어 있어서 많은 보안 취약점들이 존재한다. 특히 새로운 웹 기술들이 개발되면서 전에 없던 새로운 형태의 보안 취약점들이 꾸준히 생겨나고 있다. 이러한 웹의 특징들은 해커들에게 끊임없는 흥미거리를 제공하여 웹 해킹의 매력을 높여주고 있다[2].

2.2 웹서버 보안 강화 방안

보안시스템의 발전은 방화벽 및 침입탐지시스템 등 내부 네트워크 및 내부서버의 보호 위주로 발전을 해 왔다. 그러나 웹 서비스가 확산이 되고 방화벽에서 오픈이 되어 있는 웹 서버에 대한 공격이 늘어나면서 방화벽과 침입탐지시스템에서 방어할 수 없는 웹서비스에 대한 방어가 중요해 졌다. 일반적으로 웹에 대한 공격은 방화벽에서 오픈이 되어 있는 HTTP(80)로 이루어 지기 때문에 기존의 보안시스템에서는 속수무책이다. 그래서 이에 대응하기 위해 나온 보안시스템이 웹 방화벽과 웹 스캐너이다. 웹 방화벽의 경우, 방화벽에서 필터링 할 수 없는 어플리케이션 계층까지 필터링이 가능하여 웹 공격에 대해 안전성을 제공한다. 웹 스캐너의 경우, 웹의 취약점을 스캔하여 관리자에게 알려줌으로써 웹 관리자가 웹의 취약점을 제거하고 보안을 강화할 수 있다. 다음 그림은 웹 방화벽을 설명한 그림이다.



(그림 1) 웹방화벽의 기능

웹 서비스가 웹 사이트 중심에서 사용자 중심으로 변하기 때문에 웹 보안도 이에 맞추어 변해야 한다. 기존의 웹 보안이 Inbound에만 치중이 되어 있었다. 사용자 참여 중심이 되고 있는 양방향 통신형태에서는 Outbound 보안에도 신경을 써야 한다. Inbound 웹 트래픽은 물론 Outbound 웹 트래픽에 대해서도 고려해야 한다. 새로운 웹 보안시스템은 다음과 같은 기능을 가지고 있어야 한다.

1) 유입되는 네트워크 패킷에 일반적인 네트워크 및 서버에 대한 공격 뿐만 아니라 웹 어플리케이션 공격까지도 탐지 및 방어해야 한다.
2) 외부로 나가는 네트워크 패킷(웹 서비스 제공)에 대해서도 웹 서비스 자체의 감염에 의한 유해트래픽 발송을 탐지하고 방어해야 한다.

이와 같은 기능을 기반으로 웹 보안시스템은 다음과 같이 고려되어져야 한다.

우선, 사전적 대응으로는 웹 어플리케이션 개발 단계부터 Source Code Review가 이루어져야 한다. 사후적 대응으로 웹 방화벽, 웹 스캐너를 생각할 수가 있는데 새로운 취약점에 대해 계속적으로 탐지 및 대응 패턴을 업데이트 하여 대응을 하도록 하고, 특히 네트워크 트래픽에 대해 Inbound 및 Outbound 트래픽을 모두 모니터링하여 Inbound 트래픽에 대해서는 서버 자체에 대한 공격을 탐지하고 Outbound 트래픽에 대해서는 서버에 감추어진 악성코드 등과 같은 위험요소를 탐지할 수 있도록 한다. 추가적으로 허니웹 시스템을 구축하여 알려진 위협 및 감추어진 위협에 대해 식별이 가능하도록 하여 신규 위협에 대해 분석할 수 있도록 한다.

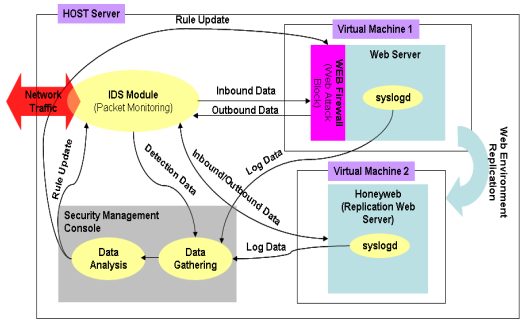
3. 웹서버 보안시스템의 설계 및 구현

본 논문에서는 웹 서버를 중심으로 가상환경을 이용한 보안시스템을 설계해 본다. 보안시스템은 사후적 대응을 중심으로 구현이 되었으며, 크게 3가지 영역으로 구성되어 있다. 우선, 웹 서버를 보호하는 웹 서버 가상화 환경과 웹 방화벽 모듈, 두 번째로 일반적인 네트워크 및 서버의 침입 및 웹 서비스의 유해 트래픽을 탐지하는 IDS Module,

세 번째로는 웹 서버의 복제 서버로 웹에 대한 위험분석을 담당하는 Honeyweb 서버이다.

3.1 웹서버 보안시스템의 설계

본 논문에서 설계하는 보안 시스템은 다음과 같다. 웹 서버 보안시스템은 가상화 환경으로 이루어진다. 우선, 메인 웹 서버를 가상서버로 구성하고 이와 똑같은 웹 서버를 가상서버로 하나 복제하여 허니웹 서버로 구성하고, 물리적인 Host 메인서버에는 IDS Module를 탑재한다. 추가적으로 메인 웹 서버에는 방화벽 모듈을 탑재하여 메인 웹 서버를 보호를 담당한다. 그리고, 웹 보안시스템을 분석하고 모니터링하기 위한 Security Management Console 모듈을 물리적인 Host 메인서버에 구성한다.



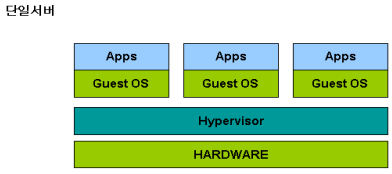
(그림 2) 웹서버 보안 시스템 구조

1) 들어오고 나가는 모든 네트워크 패킷은 IDS Module에 의해 모니터링이 되고 분석되어 진다. IDS Module은 웹 서버로 유입되는 네트워크와 서버에 대한 유해트래픽을 탐지하고 알려준다.
2) Virtual Machine1은 메인 웹 서버로 웹 방화벽에 대해 보호되어지고 있으며 웹 방화벽 및 웹 서버에서 일어나는 모든 Event에 대해 Security Management Console로 보낸다.
3) Virtual Machine2에서는 허니웹 서버로 허니웹 서버에서 일어나는 모든 Event에 대해 Security Management Console로 보낸다.
4) Security Management Console에서는 모든 영역에서 보내오는 Event들을 수집하고 분석하여 관리자에게 보여준다.
5) 수집된 Event들을 분석하여 새로운 공격패턴에 대해서는 IDS Module 및 웹 방화벽 모듈에 업데이트를 한다.

본 시스템은 웹 서버가 가상화 환경으로 구성이 되어 있어 어떤 공격에 대해서도 시스템의 하드웨어까지 영향을 미치지 않도록 구성이 되어 있으며, 웹 서버 자체에 방화벽 모듈이 탑재가 되어 있어 알려진 웹 공격에 대해 대응이 가능하고, IDS Module에 의해 웹 방화벽에서 탐지할 수 없는 부분까지 탐지가 가능하다. 또한 IDS Module에 의해 Outbound 패킷에 대해서도 모니터링 함으로써 웹서버 공격에 대한 탐지뿐만 아니라 웹 서버가 외부로의 공격하거나 악성코드의 유포에 대해서도 탐지함으로써 상호 통신의 웹 환경에서 적절한 보안을 제공한다. 그리고 Honeyweb을 통해 새로운 공격에 대해 정보를 수집하여 IDS Module, Web Firewall에 탐지 및 차단룰을 업데이트 함으로써 신규공격에 대해 대응을 할 수 있다.

3.2 가상화 환경의 구성

Web 보안시스템을 위해 웹 서버를 가상화 환경으로 구성하였다. 가상화 기술은 단일 플랫폼상의 서버 자원(CPU, 메모리, 입출력장치 등)을 사용자가 여러 도메인이나 서버 어플리케이션으로 분할해서 사용할 수 있는 기능이다[7, 10].

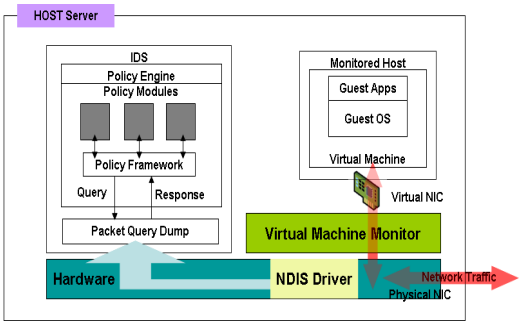


(그림 3) 가상화 서버의 기본구조

구현하는 웹 서버는 하드웨어 메인서버와는 완전 별개인 가상화 환경의 서버이다. 그러므로 하드웨어적인 공격으로부터 자유로워 질 수 있고 물리적인 서버를 통해 보호를 받고 있는 상태가 된다.

3.3 IDS Module 설계

IDS Module은 웹서버 보안시스템의 가장 핵심적인 모듈이라고 할 수 있다. IDS Module을 통해 유입되는 유해 트래픽을 탐지하고 모니터링을 하며 외부로 출력되는 네트워크 트래픽에 대해서도 어떤 유해 트래픽이 없는지 탐지하고 모니터링하는 기능을 한다.



(그림 4) IDS Module의 작동원리

본 Module은 snort로 구성을 하였으며 기본 탐지룰을 탑재하고 있다. 이외에 새로운 공격에 대해서 탐지룰을 추가 또는 수정하여 계속 업데이트를 하도록 한다[6]. NDIS(Network Driver Interface Spectification)의 목적은 Network Interface Cards를 위한 표준 API를 정의 하고, 상위 레벨(TCP/IP)프로토콜 드라이버 뿐만 아니라 MAC driver들에 의해 사용될 수 있는 함수들의 라이브러리를 제공한하며, Microsoft 네트워킹 프로토콜에서는 네트워크 카드 드라이버와 통신에 네트워크 장치 인터페이스 지정(NDIS)을 사용한다[12].

본 Module은 가상머신으로 구성된 메인 웹 서버와 Honeyweb 서버로 유입되는 모든 네트워크 트래픽을 모니터링하고 탐지하고자 한다.

3.4 웹서버 와 방화벽 Module

웹서버는 웹 서비스를 제공하는 메인모듈이다. 여기서 웹서버는 가상화 환경을 이용해 가상머신으로 구성을 한다. 물리적인 서버와는 완전 독립적인 환경을 갖기 위해 네트워크 환경을 Bridged 방식으로 구성을 하여 웹 서버 만의 독립적인 IP를 갖는다. 외부환경에서 보았을 때는 물리적인 서버와는 별개인 완전 독립된 하나의 웹서버로 보인다.

웹 서버에는 웹 방화벽을 구성을 한다. 웹 방화벽은 웹 서버를 안전하게 보호하고 모니터링 할 것이다. 차단 및 탐지되는 모든 정보는 수집이 되어 Syslog Daemon에 의해 Security Management Console로 보내진다. 관리자는 Security Management Console을 통해 웹 서버를 실시간으로 모니터링하고 대응한다. 웹 방화벽의 차단 룰은 Honeyweb Module이나 IDS Module에서 신규로 탐지되는 공격에 대해서는 방어룰을 추가 또는 수정하여 계속 업데이트를 한다. 웹 서버와 방화벽 Module은 웹 서비스를 중단없이 안전하게 제공할 것이다.

3.5 Honeyweb Module

Honeyweb Module은 허니팟(Honeypot)의 개념과 동일한 개념으로 웹 서버와 똑같은 환경을 가지고 있으며, 어떠한 해킹공격이나 악성코드 유입 등을 모두 허용을 함으로써 침입에 대한 탐지와 분석을 담당하는 모듈이다.

본 논문에서는 메인 웹서버의 복제 웹으로 웹서버로 유입되는 모든 네트워크 트래픽 및 외부로 나가는 모든 서비스에 대해 모니터링하고 분석한다. Honeyweb Module 또한 가상화 환경으로 구성함으로써 허니웹 공격에 의한 물리적인 하드웨어 리소스 문제나 하드웨어적인 피해에 대해 자유로워 질 수 있다. Honeyweb Module은 하나의 가상 웹 서버로 완전 분리가 되어 있어 어떤 침입에

의해 물리적인 메인서버 및 메인 웹 서버에는 전혀 피해를 주지 않는다.

Honeyweb Module은 IDS Module에서 탐지하지 못한 침입이나 공격에 대한 정보를 수집하여 Syslog Daemon을 이용해 Security Management Console로 넘겨준다. 관리자는 Security Management Console을 통해 IDS나 Web Firewall에서 탐지하지 못한 침입이나 공격에 대해 알 수가 있다.

Honeyweb Module은 모든 사용자 요청에 대해 정보를 수집하고 알려진 공격 또는 알려지지 않은 공격에 대해 분류를 하고 새로운 위협에 대해 분석을 하고 관리자에게 알려주는 아주 중요한 역할을 할 것이다.

3.6 구현환경

웹서버 보안시스템은 Windows기반의 하드웨어 장비에 Vmware를 통해 가상화 웹서버를 구성하고 Snort와 Modsecurity Firewall, Kiwi Syslog Daemon, HSC(Honeynet Security Console)를 통해 구현이 되었다

〈표 1〉 구현환경

구성요소	상세설명
Host Server	Windows 2000 Server
Virtualization System	VMware Workstation 6.1
Virtual Machine 1	(Web Server) RedHat Linux 8 + Apache 2.2.6 + Modsecurity Web Firewall 2.1.3
Virtual Machine 2	(HOneyweb Module) RedHat Linux 8 + Apache 2.2.6
IDS Module	Snort 2.2 + WinPcap 4.0.2
Security Management Console	HSC(Honeynet Security Console) v2.6.0.4 + mysql 5.0 + KIWI Syslog Daemon 7.1.0

4. 웹서버 보안시스템의 평가

공격테스트는 2007년 OWASP에서 발간된 'OWASP Testing Guide v2.0' 기준으로 시행이 되었다. OWASP Testing Guide는 OWASP에서 웹 어플리케이션의 취약점을 점검하기 위해 발간한 웹 어플리케이션 취약점 점검 가이드북이다[3].

4.1 Inbound 공격

Inbound 공격은 Web Vulnerability Check List 총46개 항목 중 해킹의 가장 첫 번째 단계라고 할 수 있는 정보수집(Information Gathering)단계 중 Application Discovery 항목에 대해서 테스트를 시행하였다.

- Application Discovery(OWASP-IG-002)

Application Discovery는 웹 어플리케이션의 어떠한 숨겨진 취약점을 찾아 정보를 수집하는 것이다. 여기에서는 웹 서버 포트스캔을 통해 http(80), https(443) 포트 이외에 웹 서버에서 사용하는 포트를 찾아 해당 포트를 통해 웹 서버의 정보를 수집하는 공격이다. 포트스캔은 가장 일반적인 스캐너 툴인 nmap을 사용해서 스캔을 하고, 스캔을 통

```
C:\Nmap>nmap -sF -p 1-10000 10.145.21.12

Starting Nmap 4.20 ( http://insecure.org ) at 2008-12-01 20:48 대한민국 표준시
Interesting ports on 10.145.21.12:
Not shown: 9988 closed ports
PORT     STATE         SERVICE
22/tcp   open|filtered ssh
80/tcp   open|filtered http
111/tcp  open|filtered rpcbind
137/tcp  filtered      netbios-ns
420/tcp  filtered      smpte
445/tcp  filtered      microsoft-ds
707/tcp  filtered      unknown
3127/tcp filtered      unknown
5554/tcp filtered      unknown
6000/tcp open|filtered X11
9604/tcp filtered      unknown
9996/tcp filtered      unknown
```

(그림 5) Nmap 포트스캔 공격결과

해 오픈된 포트를 Netcat 툴을 통해 웹 서버의 정보를 수집한다. nmap을 통해 스캔을 할 때는 스텔스 옵션을 주어 스캔 흔적을 최대한으로 감추어 시도하였다. 그리고 TCP포트 1에서 10000사이의 포트스캔을 시도하였다. 결과값은 기존시스템, 제안시스템 두 모델 모두 아래와 같은 같은 값을 보여주었다.

그리고 Netcat을 통해 오픈된 포트들에 대해 정보수집을 한 결과, 다음과 같은 OpenSSH에 대한 정보를 기존시스템, 제안 시스템에서 모두 획득할 수 있었다.

```
C:\>nc 10.145.21.12 22
SSH-1.99-OpenSSH_3.4p1
```

(그림 6) Netcat을 통한 웹 서버의 정보수집

기존 시스템, 제안 시스템 모두 웹 방화벽 로그에 특이사항을 찾을 수 없었지만 제안 시스템의 IDS Module을 통해 포트스캔을 탐지할 수 있었다.

```
[**] [122:1:0] (portscan) TCP Portscan [**]
[Priority: 3]
12/03-20:24:25.468984 10.145.250.200 -> 10.145.21.12
PROTO:255 TTL:0 TOS:0x0 ID:54058 IpLen:20 DgmLen:164 DF
```

(그림 7) IDS Module에서 탐지된 스캔정보

Inbound 공격에 대한 종합결과 웹 서비스에 대한 공격에 대해서는 모두 웹 방화벽에 의한 방어 및 탐지가 가능하였으나 웹 서비스에 대한 공격 외 포트스캔 등에 대해서는 기존시스템은 대응을 할 수 없었으나 제안시스템에서는 내장된 IDS Module을 통해 즉시 탐지가 가능하였다.

4.2 Outbound 공격

Outbound 공격은 웹 서버가 해커에 의해 점령이 되었을 때를 가정하고 시행하였다. 차후 웹서비스에서는 양방향 통신으로 Inbound뿐만 아니라 Outbound에 대한 보안도 중요하기 때문에 Outbound공격에 대해서도 영향을 최소화 하여야 한다. 테스트 방법은 웹 서버가 Netcat에 의해 점령당했다고 가정하고 Netcat을 통한 원격제어를 시도하고 원격제어를 통해 다른 웹 서버의 공격한다. 다른 웹 서버 공격으로는 포트스캔을 시도한다. Attack 시스템, Victim 시스템 모두 Netcat이 설치가 되어 있어야 하며, 각각 다음 명령어를 통해 Attack 시스템이 Victim 시스템을 점령한다.

```
< Victim >
# ./nc -l -p 444 -e /bin/bash
<Attacker>
# ./nc [victim IP] 444
```

(그림 8) Netcat을 통한 Victim 시스템 점령

점령한 시스템을 통해 다른 외부 서버에 대해 포트스캔 공격을 시도한다. Netcat에서 제공하는 기본 포트스캔 명령을 통해 포트스캔을 시도하였다. 이러한 공격 과정들에 대해 기존시스템에서는 어떠한 흔적을 찾을 수가 없었으나 제안시스템의 IDS Module을 통해 Netcat의 포트스캔을 탐지할 수가 있었다.

```
[**] [122:1:0] (portscan) TCP Portscan [**]
[Priority: 3]
12/09-22:38:44.792701 10.145.21.12 -> 10.145.21.102
PROTO:255 TTL:0 TOS:0x0 ID:12951 IpLen:20 DgmLen:158
```

(그림 9) 제안시스템에서 Outbound 공격 탐지 로그

Outbound 공격에 대한 종합결과 Netcat의 Remote Control을 통해 우리의 웹 서버에서 다른 웹

서버로 포트스캔 공격을 할 경우, 대응 및 탐지가 불가하였으나 제안시스템에서는 IDS Module을 통해 즉시 탐지가 가능하였다.

본 논문에서 제안한 시스템은 기존의 웹 방화벽이 탑재된 웹 서버와 비교하여 보면 Inbound 공격에 대해 웹 서비스에 대한 방어는 물론이고 웹 서비스 이외의 공격들(포트스캔 등)과 같은 공격에 대해서도 탐지 및 모니터링 할 수 있었으며 Outbound 공격에 대해서도 탐지 및 모니터링이 가능하여 다른 웹 서버 공격에 대해서도 대응이 가능하였다.

6. 결 론

오늘날 급속도로 발전하는 웹 환경의 보안 취약점 및 정상적인 서비스를 이용한 악의적 공격 기술들은 정보 시스템의 취약점에 대한 공격뿐만 아니라 네트워크의 가용성 위협 및 정보의 신뢰도 저하를 초래하여, 민간 및 공공분야에 급격히 직간접적인 피해를 유발하며 확산되고 있다. 특히, 현대 정보사회에서 정보에 대한 불신감 조장은 정보환경의 근간을 위협하는 심각한 문제로 대두될 것으로 예상되므로 이에 대한 적절한 보안 정책 및 대응방안 연구가 절실히 필요한 실정이다.

이에 본 논문을 통해 웹 서버에 대한 안전한 웹 서비스 환경을 구축하기 위해 웹 서버 보안 시스템을 설계하고 구현함으로써 서비스의 불법적인 접근 차단, 외부 또는 내부 사용자에 의한 악의적 공격 확산 방지, 웹 취약점 보완 및 웹을 이용한 악의적 공격 차단이 가능함을 보여주었다. 또한 새로운 공격에 대해서는 Honeyweb을 통해 분석하고 보안시스템의 탐지 및 방어룰을 업데이트함으로써 이를 탐지 및 방어를 할 수가 있다. 이러한 결과는 별도의 네트워크 상황 변경 없이 웹 서버를 보호할 수 있고, 웹 어플리케이션의 효율성과 신뢰성, 가용성을 보장하고 이용 가능성을 향상시킬 수 있음을 보여주고 있다.

참 고 문 헌

[1] 한국정보보호진흥원, 웹 기술동향과 보안 취약성 분석, 2006.
[2] 한국정보보호진흥원, 웹 서버 보안관리 가이드, 2003.
[3] OWASP, 2007 10대 가장 심각한 웹 어플리케이션 보안 취약점, OWASP, 2007.
[4] 한국소프트웨어진흥원, 가상화(Virtualization) 시장의 경쟁구도와 향후 전망, 2007.
[5] 안창원,김진미, 데이터센터 서버 통합(Consolidation)을 위한 가상화(Virtualization) 기술과 동향, IITA, 2007.
[6] Lionel Litty, Hypervisor-based Intrusion Detection, Univerity of Toronto, 2005.
[7] Tal Garfinkel, Mendel Rosenblum, A Virtual Machine Introspection Based Architecture for Intrusion Detection, Stanford University, 2003.
[8] Breach Security, Inc., ModSecurity Reference Manual, 2007.
[9] Breach Security, Inc., ModSecurity 2.X Changes and Migration Matrix, 2007.
[10] http://www.vmware.com.
[11] Strong, Paul, Enterprise Grid Computing. ACM Queue, 2005.
[12] http://www.microsoft.com.

유 재 형
2007년 인천대학교 컴퓨터 공학과(공학사)
2007년~현재 경기대학교 정보보호학과 석사과정

김 도 형

2003년 경기대학교 정보보호학과(공학석사)
2008년 경기대학교 정보보호학과(이학박사)
1997년~2004년 동부아남반도체
2004년~2008년 동부CNI
2008년~현재 GS홈쇼핑 디앤샵 과장

김 용 호

2002년 광운대학교 정보통신학과(공학석사)
2008년 경기대학교 정보보호학과(정보보호학 박사)
2002년~2007년 경찰청 사이버테러대응센터 연구원

하 옥 현

1978년 성균관대 정치외교학과(정치학사)
1980년 서울대 행정대학원(행정학석사)
1998년 프랑스 사회과학대학원 박사과정(DEA취득)
2005년고려대학교정보보호대학원(공학박사)
2008년 호남대학교 경찰학과 교수(교신저자)

김 귀 남

미국 캔자스대학(공학사)
미국 콜로라도주립대학(공학석사)
미국 콜로라도주립대학(공학박사)
현재 경기대학교 정보보호학과 교수
현재 경기대학교 산업기술보호특화센터장