실시간 웹 무결성 검증 시스템

한재홍*, 김대영*, 김상진**, 오희국*

*한양대학교 컴퓨터공학과, **한국기술교육대학교 인터넷미디어공학부

Real-time Web Intigrity Verification System

Jaehong Han*, Daeyoung Kim*, Sangjin Kim**, Heekuck Oh*

*Department of Computer Science and Engineering, Hanyang University
**School of Internet Media Engineering, Korea University of Technology and Education

요 약

인터넷을 통한 고도의 정보 통신망 구축은 제한적이던 네트워크의 침입에 대한 위험성을 증대시켰으며, 그에 따라 안전성 있는 웹 서비스 구축이 무엇보다 중요한 문제로 자리 잡고 있다. 안전성 있는 인터넷 서비스를 구성하기 위한 하나의 일환으로, 본 논문에서는 사용자에게 안전한 웹 서비스 제공을 위한 실시간 웹 무결성 검증 시스템을 제안하고자 한다. 프록시 서버와 웹 서버의 연동 보안 기술로, 데이터 무결성 검증을 위한 SHA-1과 N-gram의 효율적인 hybrid 방식 기법을 이용한 이 시스템은 XSS 취약점 공격, SQL 삽입 공격, 파일 업로드 취약점 공격 등을 방어하고 정부기관 및 기업뿐만이 아닌 홈네트워크 환경에서도 안전한 보안 서비스를 제공할 수 있을 것이다.

I. 서론

컴퓨터와 데이터 통신 기술의 급속한 발전은 고도의 정보 통신망 구축을 가능하게 하였을 뿐 아니라, 정부기관을 비롯하여 금융권, 기업체 그리고 개인에 이르기까지 각종 정보를 공유하게 함으로써 사회 전반에 큰 변화를 가져오게 하였다. 그러나 이러한 인터넷 등의 컴퓨터 통신망을 통해 예전까지는 제한적이던 네트워크 침입의 위험성이 크게 높아졌으며, 그로 인해 정보가 위조 또는 변조 되고 허락 없이 유출되는 등 각종 불법 행위가 빈번하게 발생함으로써 정보화로 야기되는 역기능의 폐해가 심각한 지경에 이르고 있다.

이 같은 위험을 방지하고 인터넷 상에서 안전한 웹 서비스를 제공하기 위하여 대부분의 공공기관이나 기업, 개인들은 인터넷 보안 대책 방향으로 방화벽(firewall)이나 침입탐지 시스템(Intrusion Detection System) 등 각종 네트워크 보안 시스템을 설치, 운용하여 안전성 있는 웹 서비스를 제공하기 위해 노력하고 있다.

안전성 있는 인터넷 서비스를 구성하기 위한 하나의 일환으로 본 논문에서는 사용자에게 안전한 웹 서비스를 제공하기 위한 실시간 웹 무결성 검증 시스템을 제안하고자 한다. 국정홍보처와 정보통신수출진흥센터 홈페이지 변조 사건이나 민간 외교 사절단 '반크' 영문 홈페이지가 해킹당하는 등 이전 해킹에 대한 피해 사례를 살펴보면 각 정부기관 및 민간 단체, 기업 등의 웹 홈페이지가 변조되면서 국가적 차원의 피해를 입는 경우가 대부분이다. 이러한 문제점을 보완하여 사용자가 웹 서비스 요청 시 각 데이터에 대한 무결성을 검증하여 공격자에 의한 해킹 시에도 사용자에게 안전한 웹 서비스를 제공할 수 있는 실시간 웹 무결성 검증 시스템을 설계 및 구현하였다. 제 2장에서는 실시간 웹 무결성 검증 시스템의 정의와 시스템을 개발하기 위한 관련연구내용을 기술하였고, 제 3장에서는 본 논문에서 제시하는 시스템의 설계 및

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터(ITRC) 지원 사업의 결과로 수행되었음.

* 이 연구에 참여한 연구자는 '2단계 BK21 사업'의 지원을 받았음.

구성과 사용 시나리오를 설명한다. 제 4장에서는 시스템의 전체적인 구현에 대해 기술하고, 마지막 제 5장에서 본 논문의 결론을 맺는다.

II. 관련연구

1. 실시간 웹 무결성 검증 시스템의 정의

실시간 웹 무결성 검증 시스템은 방화벽, SSL/TLS 등 기존의 보안 시스템에 추가되어 사용자에게 좀 더 안전한 웹 서비스를 제공하기 위한 목적으로 설계되었다. 이 시스템은 사용자에게 웹 서비스를 제공하기 전에 요청한 웹 페이지에 대한 데이터의 무결성을 검증하여, XSS 취약점 공격, SQL 삽입(injection) 공격, 파일 업로드 취약점 공격 등의 직접적인 홈페이지 변조 공격에 대비하고 있다. 만약, 해킹에 의해 웹 서비스 데이터가 변조되어 무결성 검증 시 오류가 나타나면 웹 서버를 통해 원본 데이터를 가져와 사용자에게 항시 문제없는 웹 서비스를 제공하게 된다.

실시간 웹 무결성 검증 시스템은 프록시 서버와 웹 서버에서 SHA-1과 N-gram 기반 색인 기법을 이용하여 무결성 데이터를 생성하고 검증하게 된다.

2. Proxy Server

워크스테이션 혹은 PC 사용자와 인터넷의 사이에서 중개 역할을 수행하는 서버로서, 그 기관의 보안 및 관리 제어와 캐시(cache) 서비스를 가능하게 한다. 프록시 서버가 사용자로부터 Web 페이지 요청과 같은 인터넷 서비스 요청을 받으면, 하나의 캐시 서버로서 이전에 다운 로드된 Web 페이지들의 로컬 캐시를 탐색한다. 만일 요청한 페이지가 발견되면, 인터넷에 요청을 내보낼 필요 없이 그 페이지를 사용자에게 제공한다. 만일 그 페이지가 캐시에 없으면, 프록시 서버는 요청한 사용자를 대신해서 하나의 클라이언트(client)로써 자신의 IP 주소들 중의 하나를 사용해서 인터넷 상의 서버에 그 페이지를 요청한다. 요청한 페이지가 수신되면, 프록시 서버는 이것을 원래의 요청으로 연관시켜 사용자에게 전송하게 된다.

본 논문에서 제안하는 실시간 웹 무결성 검증 시스템 설계 시 빠른 인터넷 서비스를 위해 가장 많이 사용하는 스퀴드 프록시 서버를 사용하며, 프록시 서버에서 무결성 데이터를 검증함으로서 이미 검증한 웹 페이지에 대해서는 요청 시 바로 사용자에게 제공하여 프록시 서버의 사용 이점을 최대한으로 이용하였다. 또한, 프록시 서버를 역방향으로 설계하여 데이터 변조 시 원본 데이터를 제공하는 웹 서버를 방화벽 내부에 두어 더욱 안전한 보안 시스템을 구성토록 하였다[1][2].

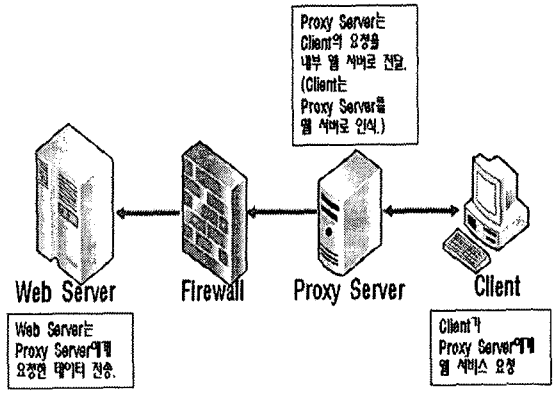

그림 2 : 역방향 프록시 서버 구성도

3. SHA-1

SHA-1(Secure Hash Algorithm-1)는 1993년 NIST에서 설계한 SHA의 취약부분을 수정하여 개발된 SHA의 새로운 버전이다. SHA-1은 MD4에 기반한 160비트의 해쉬 함수로, MD4라는 공통된 기반을 가지고 있기 때문에 많은 부분이 MD5와 비슷하다. SHA-1은 5개의 32비트 워드로 구성된 160비트 상태(state)를 가지고 있고, MD5와 같이 32비트 워드를 섞는 4개의 라운드를 가지고 있으며, 각각의 메시지 블록을 4번 이용하는 대신, SHA-1은 메시지 블록의 16워드를 필요로 하는 80워드로 확장(stretch)하기 위해서 선형 재귀(linear recurrence)를 이용한다. SHA-1에서 이용되는 선형 재귀가 각 메시지 비트가 적어도 12번 혼합 함수에서 사용될 수 있도록 보장함으로서, 충돌 범위는 2^{80}단계로 늘릴 수 있다. SHA-1은 MD5와 비교하여 속도는 2~3배 정도 속도가 느리지만, 그만큼 더 안전하다는 평가 속에 널리 이용되고 있다[3].

본 논문에서 SHA-1은 HTML 문서나 이미지, 그리고 각 웹 스크립트 파일 같은 각 웹 서비스 데이터에 대한 무결성 데이터를 생성하고 검증하는데 이용된다.

4. N-gram

문장·음성의 인식이나 이해를 하려면 문법 처리

가 필요하다. 그런데 자연 언어(한국어, 영어 등)의 문법은 매우 복잡하기 때문에 인간의 발화(發話)는 정규 문법에 따르지 않는 경우가 많다. 종래의 문장 인식에서는 음소·단어 인식을 한 후에 규칙에 따라 기술된 문법 처리를 적용하고 최후에 오류를 수정하는 방법이 주류였으나, 최근에는 단어열을 확률적으로 취급하는 언어 모델이 성행되어 연구, 실용화되고 있다. N-gram은 이와 같은 확률적 언어 모델의 대표적인 것으로서, n개 단어의 연쇄를 확률적으로 표현해 두면 실제로 발성된 문장의 기록을 계산할 수 있다[4][5].

N-gram은 HTML 문서에 대해 더욱 정확한 무결성 검증을 위해 사용된다. 예를 들어 웹 페이지에 대해 문서의 위치가 바뀌어도 문제가 없는 경우에 대해 SHA-1의 해쉬 값을 통한 무결성 검증 시 데이터 변조에 의한 오류가 발생하였다고 검증하게 된다. 이러한 HTML 문서의 경우는 N-gram으로 추출한 결과값으로 무결성을 한번 더 검증함으로서 해쉬값으로 무결성 검증 오류 시에도 올바른 웹 데이터로 판정하여 부정확한 무결성 검증에 대한 문제를 방지할 수 있다.

표 1 : N-gram 기반 색인법의 예

문서/질의 : 이번 자료부터 음절색인방법을 적용한다	
(1) 모든 어절 추출	이번, 자료부터, 음절색인방법을, 적용한다
(2) 불용어 제거	자료부터, 음절색인방법을, 적용한다
(3) 비색인 음절 절단	자료, 음절색인방법, 적용
(4) 2-gram방법 적용	자료, 음절, 절색, 색인, 인방, 방법, 적용

III. 실시간 웹 무결성 검증 시스템 설계

본 장에서는 사용자에게 안전한 웹 서비스 제공을 위한 실시간 웹 무결성 검증 시스템의 설계 및 구성, 그리고 사용 시나리오에 대해 설명한다. 실시간 웹 무결성 검증 시스템은 아파치 웹 서버에 연동된 무결성 데이터 생성 모듈을 이용하여 DB에 저장된 원본 데이터에 대한 무결성 데이터를 생성하고, 스퀴드 프록시 서버에 연동된 무결성 검증 모듈을 이용하여 원본 데이터와 무결성 데이터를 비교, 데이터의 무결성을 검증하게 된다.

1. 실시간 웹 무결성 검증 시스템의 구성도

실시간 웹 무결성 검증 시스템은 프록시 서버와 아파치 웹 서버의 연동 보안 기술로, 아파치 웹 서버에는 무결성 데이터 생성 모듈, 스퀴드 프록시 서버에는 무결성 검증 모듈이 각각 연동된다. 무결성 데이터 생성 모듈은 웹 서비스에 이용되는 모든 데이터에 대한 무결성 데이터를 생성하여 DB에 저장하게 되며, 무결성 검증 모듈은 사용자가 웹 서비스 요청 시 원본 데이터와 무결성 데이터를 DB에서 가져와 원본데이터에 대한 무결성을 검증한다.

그림 3은 실시간 웹 무결성 검증 시스템의 세부 모듈에 대한 구성도를 나타낸다.

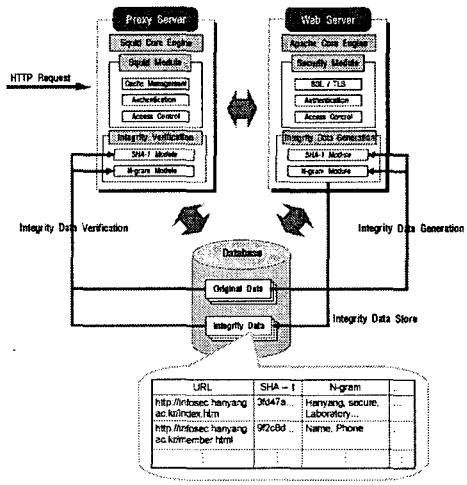

그림 3 : 실시간 웹 무결성 검증 시스템 모듈의 구성도

1) 무결성 데이터 생성 모듈

아파치 웹 서버에 연동된 무결성 데이터 생성 모듈은 SHA-1과 N-gram을 이용하여 해쉬값으로 무결성 데이터를 생성한다. 이미지, 그리고 PHP나 JSP의 웹 스크립트 파일 등은 DB에 있는 원본 데이터를 이용하여 무결성 데이터를 생성, DB에 저장하게 된다. 또한 HTML 문서는 해쉬 값만이 아닌 N-gram을 통해 각 명령어를 추출한 무결성 데이터를 하나 더 만들어 DB에 저장한다. 웹 페이지 업데이트 실시간으로 웹 파일에 대한 무결성

데이터를 바로 생성함으로서 사용자가 원하는 웹 서비스 이용 시 즉각적으로 무결성 검증을 할 수 있도록 한다.

1) 무결성 검증 모듈

스퀴드 프록시 서버에 연동된 무결성 검증 모듈은 원본 데이터와 무결성 데이터를 비교하여 데이터의 무결성을 검증한다. HTML 문서는 해쉬 값과 N-gram 추출 데이터를 이용하여 무결성을 검증하고, 각 웹 페이지에 맞는 이미지 및 웹 스크립트 파일의 원본 데이터와 무결성 데이터를 DB에서 가져와 무결성을 검증한다. 데이터에 이상이 없을 시는 바로 사용자에게 웹 서비스를 제공하며, 데이터가 변조되었다고 판단 시는 웹 서비스에 연결된 원본 데이터 DB를 통해 가져온 원본 데이터로 복구하여 사용자에게 안전한 웹 서비스를 제공하게 된다.

2. 사용 시나리오

1) 공공기관에서 사용하는 실시간 웹 무결성 검증 시스템

그림 4에서와 같이 웹 보안 서비스 제공 시나리오는 다음과 같은 절차에 따라 수행된다.

Step 1. 관리자는 웹 서버에 연동된 무결성 데이터 생성 모듈을 이용하여 원하는 웹 페이지에 대한 무결성 데이터를 생성하여 DB에 저장한다.

Step 2. User는 공공기관의 홈페이지에 접속하여 웹 서비스를 요청한다.

Step 3. 프록시 서버에 연동된 무결성 검증 모듈은 사용자가 요청한 웹 페이지에 대해 무결성을 검증한다. 이상이 없을 시는 원문 페이지를 그대로 송신하며, 만약 데이터가 변조되었다고 판단되면 원문 페이지로 복구하여 사용자에게 웹 페이지를 송신한다.

Step 4. 만약 홈페이지 업데이트 시 관리자는 실시간으로 무결성 데이터를 생성하여 웹 서비스에 차질이 없도록 한다.

그림 4는 공공기관에서 실시간 웹 무결성 검증 시스템을 이용한 웹 보안 서비스 제공 시나리오이다.

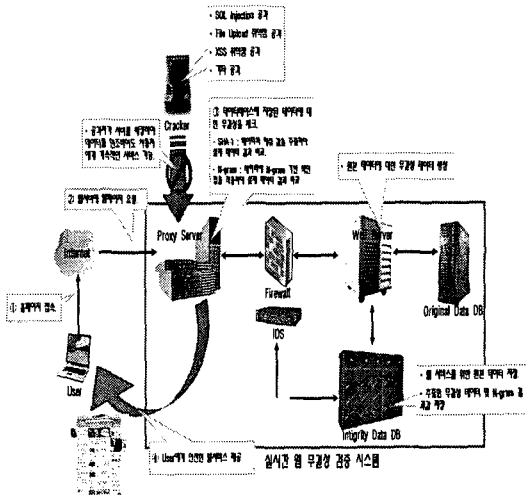

그림 4 : 공공기관에서의 웹 보안 서비스 제공 시나리오

2) 홈네트워크에서 사용하는 실시간 웹 무결성 검증 시스템

실시간 웹 무결성 검증 시스템은 OSGi를 이용한 홈네트워크 미들웨어 구축을 통해 집안 상태 정보 등을 사용자가 안전하게 서비스 받을 수 있도록 할 수 있다.

웹 기반 무결성 검증 홈네트워크 보안 서비스 시나리오 절차는 다음과 같은 절차에 따라 수행된다.

Step 1. 홈네트워크 웹 서버는 실시간으로 각 모듈의 상태 정보에 대한 무결성 데이터를 생성하여 DB에 저장한다.

Step 2. 사용자는 PDA 단말기를 통해 원격 서버에 접속하여 홈네트워크 인터페이스를 요청한다.

Step 3. 무결성 검증 플랫폼을 통해 각 상태 정보에 대한 데이터의 무결성을 검증하여 사용자에게 홈네트워크 서비스를 제공한다. 홈네트워크 웹 서버가 생성한 무결성 데이터를 이용하여 상태 정보 업데이트 시 실시간으로 데이터의 무결성을 검증, 사용자가 안전하고 빠른 홈네트워크 서비스를 이용할 수 있도록 한다.

그림 5는 웹 기반 무결성 검증 홈네트워크 보안 서비스 구성도를 보여준다.

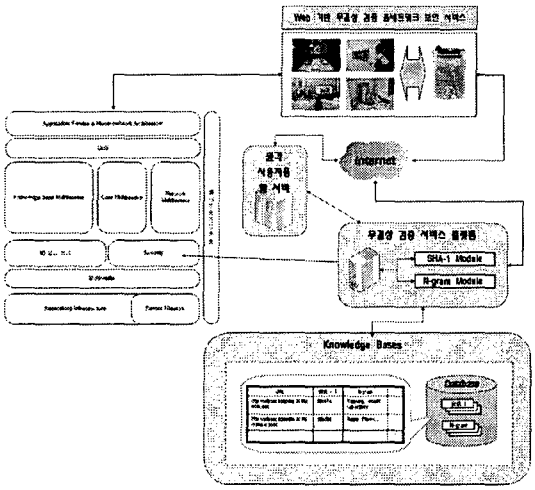

그림 5 : 웹 기반 무결성 검증 홈네트워크 보안 서비스 구성도

IV. 시스템 구현

본 절에서는 실시간 웹 무결성 검증 시스템의 구현에 대해 기술한다. 기본적으로 실시간 웹 무결성 검증 시스템은 Open SSL 기반으로 구현하였으며, 스퀴드 프록시 서버와 아파치 웹서버에 SHA-1와 N-gram 기반 색인 기법을 이용한 생성 및 검증 모듈을 연동하였다. DB는 MySQL을 이용하여 무결성 검증을 위한 무결성 데이터 DB와 원본 데이터 복구를 위한 원본 데이터 DB를 각각 구성하였다. 기본적으로 SSL 및 방화벽의 보안 시스템을 구성하여 홈페이지 변조 공격 외 다른 해킹 공격 역시도 방어할 수 있도록 하였다.

1. 웹 서버 및 무결성 데이터 생성 모듈

웹 서버는 DB에 있는 웹 페이지 데이터를 가져와 무결성 데이터를 생성한다. 관리자는 무결성 검증을 원하는 웹 페이지를 설정하여 자신이 원하는 페이지에 대해서만 무결성 데이터를 생성하도록 설정할 수 있다. 기업 홈페이지의 경우 관리자는 무결성 생성 모듈을 실시간으로 실행하여 홈페이지 업데이트 시 무결성 데이터를 실시간으로 생성하여 사용자가 웹 서비스 요청 시 웹 무결성 검증에 차질이 없도록 할 수 있다.

2. 프록시 서버 및 무결성 검증 모듈

사용자가 웹 서비스를 요청하면 프록시 서버는 요구한 웹 페이지에 대해 무결성 검증을 실행한다. HTML 문서에 대한 무결성 검증 시 각 웹 페이지에서 로드하는 웹 스크립트 파일, 이미지 등 DB에 저장된 데이터를 불러와 따로 무결성 검증을 실행한다. 기본적으로 무결성 검증 시 SHA-1를 이용한 해쉬 값을 산출하여 무결성 데이터와 비교하며, HTML 문서와 같이 해쉬 값만으로 검증이 어려운 데이터는 N-gram 결과값을 이용하여 무결성을 검증한다.

V. 결론

정부 기관 및 금융권, 기업체에서의 인터넷 활용을 통한 각종 정보의 공유와 비즈니스 및 개인 상호 작용이 활발히 이루어지고 있는 가운데 사용자에 대한 안전한 웹 서비스의 제공이 무엇보다 중요한 문제로 각광받고 있다. 이러한 상황에서 각 기관이나 기업의 홈페이지 변조와 악용은 국가적 차원의 문제로 발전되고 있는 상황이다. 이러한 문제를 해결하기 위하여 본 논문에서는 사용자에게 안전한 웹 서비스 제공을 위한 실시간 웹 무결성 검증 시스템을 설계하고 구현하였다.

방화벽, SSL/TLS 등 기존의 보안 시스템에 실시간 웹 무결성 검증 시스템을 포함해 더욱 더 강건한 보안 서비스를 구축함으로서 정부기관이나 기업들의 웹 서버 데이터를 삭제, 또는 변조하거나 웹 페이지의 링크를 바꾸는 등 홈페이지에 대한 직접적인 공격을 방어할 수 있다. 아울러 OSGi를 이용한 웹 기반 무결성 홈네트워크 보안 서비스를 구축하여 사용자에게 집안의 상태 정보 등을 안전하게 서비스함으로서 더욱 안전한 홈네트워크 보안기술 개발에 도움이 될 것이라 생각한다.

참고문헌

[1] T.Squid, "SQUID Frequently Asked Questions," http://www.squid-cache.org/Doc/FAQ, 2004.

[2] O. Pearson, "Squid : A User's Guide," http://squid-docs.sourceforge.net/latest/html/book1.html, 2003.

[3] NIST, FIPS 180-1 "Secure Hash Standard," 1995.

[4] M.K .Brown, A. K.ellner, and D. Raggett, "Stochastic Language Models (N-Gram)

Specification,"
http://www.w3.org/TR/2001/WD-ngram-spec-20010103/, 2001.
[5] P.F. Brown, V.J. Della Pietra, P.V. deSouza, J. C. Lai, and R.L. Mercer, "Class-based n-gram models of natural language," Computational Linguistics, vol. 18, pp. 467-479, 1992.
[6] B. Laurie, P. Laurie, 권순선 역, "Apache : The Definitive Guide," 한빛미디어, 1999.
[7] 홍석범, "리눅스 서버 보안관리 실무," 수퍼유저코리아, 2005.