□특집□

월드 와이드 웹(WWW) 보안기술

강 신 각† 박 정 수††

◈ 목 차 ◈

1. 서 론

WWW(World Wide Web)을 한마디로 요약해서 말한다면 분산 하이퍼미디어 정보검색 시스템이라고 표현할 수 있다. 여기서 하이퍼미디어란 용어는 하이퍼텍스트와 멀티미디어의 개념이 통합된 것으로, WWW을 이용하면 세계 각지에 분산되어 존재하는 텍스트, 그래픽, 영상, 그리고 사운드 등으로 구성되는 멀티미디어 정보를 동적으로 검색할 수 있다.

웹의 역사는 1989년 3월 유럽의 물리학연구소인 CERN에서 여기저기 산재해 있는 다양한 정보들을 효율적으로 사용할 목적으로 Tim Berners Lee에 의해 시작된 프로젝트로 부터 유래한다. 1993년 6월 웹 서비스를 위한 사용자 인터페이스인 모자익(Mosaic)이 개발, 발표되면서 웹은 급격히 주목 받기 시작하였고, 그 후 넷스케이프 브라우저가 세계 브라우저 시장을 석권하였으며, 뒤이어 마이크로소프트사에서 윈도우에 웹 브라우저인 익스플로러를 기본으로 제공하면서 현재 세계 브라우저 시장을 양분하고 있다.

초창기에 웹은 공개적으로 게시된 정보의 검색 서비스에 주로 이용되었으나 웹의 편리함과 효용성이 알려지면서 전자메일, BBS, 전자상거래 등 광범위한 분야로 이용이 확대되고 있다. 특히 인터넷에서의 전자상거래는 시장 잠재력이 무한한 분야로써, 많은 회사에서 관심을 가지고 적극적으로 사업화에 나서고 있어 현재 인터넷 상에는 수많은 전자쇼핑몰이 출현하고 있다. 이와 같이 웹이 다양한 응용 서비스로 그 이용이 확대되면서 웹 서비스의 보안성 문제가 제기되기 시작하였다. 애당초 웹은 다른 대부분의 인터넷 응용과 마찬가지로 보안을 별로 염두에 두지 않고 설계된 응용으로 보안이 요구되는 민감한 분야에서는 사용이 적합치 않은 것이 사실이었다. 그러나 웹이 단순한 정보검색 뿐 아니라 신용카드 정보와 같은 타인에게 노출되어서는 안될 중요한 정보의 전송과 같이 용도가 다양해 짐에 따라 웹의 보안 문제는 필수사항이 되게 되었다[1][2].

본 고에서는 웹의 보안 요구사항, 제안된 보안 기법들, 그리고 웹 브라우저와 서버에서 현재 실제로 구현되어 일반적으로 사용되고 있는 웹 보안 사례에 대해 살펴본다.

2. 웹 보안 요구사항

보안 서비스가 요구되는 대표적인 웹 응용을 보면 먼저, 특정 그룹 구성원 사이에서만 민감한

† 정회원 : 한국전자통신연구원 책임연구원
†† 정회원 : 한국전자통신연구원 연구원

정보를 공유하고자 하는 응용 형태가 있다. 이 경우에는 서버의 정보에 접근하는 클라이언트를 제어하기 위해 클라이언트에 대한 인증(Authentication)이 요구된다. 둘째로는 중요한 정보를 신뢰성 있게 교환하고자 하는 응용 형태가 있다. 즉, 구매 주문이나 중요한 공문서를 발행하고자 하는 경우 클라이언트 뿐만 아니라 서버 역시 정당한 서버인지를 인증할 수 있어야 하며, 교환되는 메세지 또는 문서 자체에 대한 인증도 요구된다. 셋째로는 웹을 이용하여 교환되는 정보 자체가 타인에 노출되지 않기를 바라는 통신의 비밀보장 서비스가 요구될 수 있다. 그리고 넷째로 웹을 이용하는 상거래 응용의 경우에는 웹 상에서 전자지불 기능을 요구하게 되며, 이 경우 판매자의 정당성 인증과 구매자의 지불이 안전하게 이루어질 수 있도록 하는 보호기능이 요구된다.

이러한 보안 요구사항을 정리해 보면 웹 클라이언트 인증, 웹 서버 인증, 웹 서버에 있는 문서 정보에 대한 접근제어(Access Control), 서버와 클라이언트 사이에 일어나는 트랜잭션 데이타의 인증, 무결성(Integrity), 그리고 기밀성(Confidentiality)이 요구된다[3].

3. 웹 보안 기법

3.1 개요

웹은 하이퍼미디어 문서를 작성할 수 있도록 하는 표준언어인 HTML(HyperText Markup Language)과 HTML로 작성된 문서를 클라이언트와 서버 사이에 전송할 수 있게 해주는 통신 프로토콜인 HTTP(HyperText Transfer Protocol), 그리고 웹 시스템이 인지할 수 있는 주소지정 문법인 URL(Uniform Resource Locator)을 주요 구성요소로 한다. 이밖에도 클라이언트에서 비디오, 사운드, 그래픽 정보 등을 사용자에게 표현해 주는 다양한 외부 표시기(External Viewer) 및 플러그 인 소프트웨어가 웹의 기능을 더욱 강력하게 해 주고 있다.

웹에 보안 기능을 제공하기 위한 대부분의 연구는 전송 프로토콜인 HTTP에 기존의 암호 메커니즘을 어떻게 적용할 것인가에 초점이 맞추어져 있다. 이 HTTP는 정보를 가진 웹 서버로의 접근 프로토콜과 HTML 문서 교환을 위한 전송 메시지 형식 제공 기능을 가지고 있다. 이 두 가지 기능을 기반으로 웹 보안 기술은 일반적으로 세가지로 분류된다[4].

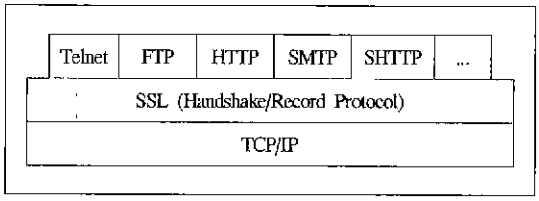
첫번째는 Kerberos나 PGP와 같이 안전성이 인정되고 있는 기존 암호 시스템과 연계하여 HTTP의 상위계층에서 보안 기능을 구현해 주는 내용기반(Content-based) 방식으로, 이 방식에서는 일반적으로 HTTP 메시지 전체를 암호화 한다.

두번째는 HTTP와는 독립적으로 존재하여 HTTP 메시지의 바디 영역만을 암호화하는 메시지기반(Message-based) 방식이 있다. 이러한 방식에 따라 전송되는 트랜잭션 메시지별로 보안 기능을 제공하는 기술로 제안된 것이 SHTTP(Secure-HTTP)이다[5]. S-HTTP는 전자상거래 응용 등에 사용하기에 적합하고 다른 방식에 비해 우수한 기능을 제공하여 제안 당시 주목을 받았었으나 이미 넷스케이프 브라우저에 구현되어 사용되고 있던 SSL에 밀려 실제 확산에는 실패하였다.

세번째는 HTTP 계층과 TCP 계층 사이에 존재하여 HTTP 메세지가 전달될 TCP 연결에 대해 암호화 기술을 적용하는 방식이 있다. 이 경우, 동일 TCP 연결상의 모든 HTTP 메시지에 대해 동일한 암호화 서비스를 제공하게 되는데 이를 채널기반(Channel-based) 방식이라 하며 SSL(Secure Socket Layer)이 대표적인 프로토콜이다. 이와 같은 여러 가지 방식 중에서 채널기반 방식인 SSL이 넷스케이프 브라우저의 보급에 힘 입어 사실상의 웹 보안 표준으로 정착되었다[6]. IETF에서는 SSL 3.0을 기반으로 Transport Layer Security (TLS) Protocol 1.0을 개발하였다[7].

3.2 채널기반 웹 보안 방식

대표적인 채널기반 방식인 SSL은 넷스케이프사에서 개발되었으며, 특정 응용을 위한 보안 프로토콜이 아닌 일반적인 인터넷 보안 프로토콜로 사용되고 있다. (그림 1)에서 처럼, 인터넷 응용과 TCP/IP 통신 프로토콜 계층 사이에 존재하는 프로토콜이므로 웹을 위한 HTTP뿐만 아니라 Telnet, FTP 등 다른 응용들에도 적용될 수 있다. 이와 같은 측면이 사실상의 웹 보안 표준으로 자리잡게 했지만, 새롭게 부각되고 있는 인터넷 전자상거래에서 필수적으로 요구되고 있는 전송 문서별 디지털 서명과 같은 기능을 제공하지 못하고 있는 약점은 보완해야 할 과제이다.



(그림 1) SSL의 계층 모델

3.21 SSL의 구성

SSL은 두 통신 응용들간에 신뢰할 수 있는 채널을 통해 기밀성 서비스 제공을 목표로 하며, 개념적으로 레코드 프로토콜(Record Protocol)과 핸드쉐이크 프로토콜(Handshake Protocol) 계층으로 구분된다. TCP 상위에 존재하는 레코드 프로토콜은 여러 가지 상위 계층의 프로토콜들을 캡슐화하기 위해 사용된다. 이는 데이터 압축 및 암호화 메커니즘을 통해 실제적인 암호문을 생성하는 과정을 의미한다. 이 과정을 통해 캡슐화 되는 응용 중의 하나인 핸드쉐이크 프로토콜은 서버와 클라이언트간의 상호 인증 서비스를 제공하고 암호 알고리즘과 암호키를 협상한다. 이 외에도 SSL Change Cipher Spec과 SSL Alert Protocol이 레코드 계층을 통해 캡슐화 된다. SSL 프로토콜은 두 통신 응용간의 기밀성 서비스, 클라이언트와 서버 인증 서비스, 메시지 무결성 서비스, 암호 알고리즘과 암호키 등의 협상 서비스를 제공한다.

3.2.2 SSL의 동작

SSL이 제공되면 하부 망을 통해 전달되는 메시지 전체에만 암호화를 적용하기 때문에 웹 클라이언트와 서버 응용은 일반적인 HTTP를 통해 접속이 이루어진다. 이와 같은 암호화 통신을 위한SSL의 동작은 크게 두 가지로 분류할 수 있다. 먼저 핸드쉐이크 프로토콜에 의해 통신하고자 하는 응용 간에 공개키 암호화 기술을 이용하여 안전한 통신 채널을 설정하고, 그 후에 레코드 프로토콜에 의해 안전한 통신 채널로 응용 간에 정보가 교환된다. 첫번째 과정에서 클라이언트와 서버 간의 상호 인증 과정과 세션 키 교환을 수행하며, 두 번째 과정에서 공유된 세션 키를 이용하여 응용 실체 사이에 대칭 키 방식의 암호통신을 하게 한다.

3.3 내용기반 웹 보안 방식

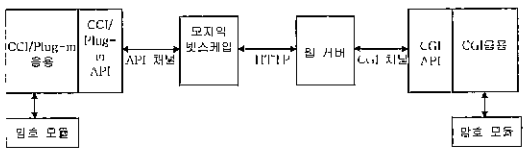
3.3.1 외부 프로그램을 이용한 방법

내용기반 웹 보안방식 중의 한 방법으로 서버와 브라우저의 외부에서 수행되는 프로그램을 따로 설치하는 방법이 있다. 이 방법은 기존의 웹 시스템에 전혀 수정을 요구하지 않으면서 웹 보안 기능을 구현해 준다. S-HTTP가 기존의 HTTP와 서버, 브라우저 프로그램을 수정해 주어야 한다는 부담 때문에 실패한 것과 비교해 보면, 이 특징은 큰 장점으로 생각할 수 있다.

(그림 2)와 같이 독립적으로 암호 기능을 수행하는 외부 프로그램이 서버와 클라이언트에 연결되어 있으며, 이 외부 프로그램은 HTTP 요청 및 응답 메시지의 암/복호화에 이용된다. 메시지가 서버와 브라우저를 떠나 전송되기 직전에 각 외부 프로그램에 보내지고, 암호화된 메시지들은 다

시 서버와 브라우저로 보내져서 인터넷을 통해 전송된다. 이와 같은 외부 프로그램은 기존의 CGI 프로그래밍 기법을 사용하여 서버와 연결되며, 클라이언트는 브라우저에 따라 여러 가지 방법이 이용된다. 넷스케이프인 경우, 플러그 인 또는 외부 표시기 기법을 이용하고, 모자익의 경우는 NCSA에서 제공하는 CCI 라이브러리를 이용한다. 익스플로러는 SSPI를 이용할 수 있다[8].

이 방법의 가장 큰 장점은 기존의 웹 시스템에 아무런 수정을 요구하지 않는다는 것이다. 또한, PGP 등 암호 모듈은 웹 이외에도 여러 응용 프로그램들에게 보안 기능 지원을 위해 이용될 수 있으므로, 한 시스템에 여러 가지 암호 프로그램을 설치하지 않아도 된다는 장점이 있다. 그렇지만, 결국 웹 응용 외부에 위치하기 때문에 불필요한 처리 절차를 수행하게 됨으로써 수행시간이 길어진다는 단점이 있다.

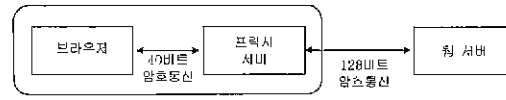


(그림 2) 외부 프로그램을 이용한 방법 예

3.3.2 프락시 서버를 이용하는 방법

40비트의 암호 키를 사용하는 SSL의 단점을 극복하기 위한 기술로, 브라우저가 설치되어 있는 컴퓨터에 프락시 서버를 설치하고 이 프락시 서버로 하여금 웹 서버와 128비트 암호문 통신을 하게 하는 방법이다. 40비트의 암호키를 사용하는 제한은 브라우저에만 적용되고 서버에는 128비트도 허용된다는 점에서 착안한 기술이다.

다시 말해, 브라우저와 프락시 서버는 기존의 40비트를 사용하는 SSL 통신을 그대로 사용하고, 프락시 서버가 가상의 SSL 서버 역할을 해 주는 것이다. 프락시 서버는 브라우저가 전송한 40비트의 암호문을 128비트 암호문으로 재 생성하여 실제로 브라우저가 통신하려고 하는 웹 서버로 전송한다. 이때 웹 서버는 128비트의 키를 사용하도록 지정되어 있어야 한다. 이 방법을 사용하면 모든 통신이 프락시 서버를 통해서 송/수신되어야만 하고 브라우저에서 프락시 서버를 지정해야 한다. 만약 어느 시점에서 사용하지 않겠다면 브라우저 옵션을 일일이 변경해 주어야 하는 번거로움이 있다. 그러나 현재는 브라우저도 128비트까지 허용되고 있으며, 이러한 경우에는 이 방식을 사용할 필요가 없게 된다.



(그림 3) 프락시 서버를 이용한 통신 예시

4. 웹 브라우저 보안

대부분의 인터넷 사이트는 권한이 없는 사람들이 서버에 접근하여 정보를 보내지 못하도록 대책을 마련해 두고 있다. 이러한 사이트를 보안 사이트라고 한다. 대표적인 웹 브라우저인 넷스케이프와 익스플로러에서는 보안 사이트가 사용하는 보안 프로토콜들을 지원하기 때문에 마음 놓고 안전하게 보안 사이트와 정보를 주고받을 수 있다. 이들 브라우저는 SSL을 기반으로 전송 메시지에 대한 보안 서비스를 제공하고 있다. 웹 브라우저들은 다음과 같은 보안 요구사항을 기술적으로 극복해야 한다.

- 클라이언트 응용이 임의의 서버로 부터 전송되는 데이터를 해석하므로 생기는 위협으로 트로이 목마의 한 형태이다.
- 클라이언트 응용이 데이타와 프로그램을 다운로드하여 생기는 위협으로 핫 자바의 경우가 전형적인 예이다.
- 악의의 서버가 클라이언트 시스템에 침입할 수 있다.

- 웹 서버는 임의의 사용자에 의해 쉽게 설정 정보가 변경될 수 있다.
- 사용자가 웹 서버로부터 가져온 특정 문서를 외부 표시기로 보려고 시도할 때 외부 표시기 자체가 보안 구멍을 가질 수 있으며 이로 인해 문서가 유출될 수 있다.

4.1 넷스케이프의 보안기능

브라우저와 서버가 동시에 SSL 기능을 제공하면 HTTP 메시지는 암호화되어 전송된다. 예를 들어, 브라우저가 핸드쉐이크 프로토콜을 통해 교환된 키를 이용해서 RC4 알고리즘을 사용하여 암호화하여 전송하면 서버는 이를 복호화 한다. 이때, 일반적인 연결과 보안 연결을 구분하기 위해 "http" 대신에 "https"가 사용된다. 또한 브라우저의 왼쪽 아래에 있는 보안 아이콘으로 보안 채널로 연결되어 있는 지를 확인할 수 있다. 파란색 배경의 열쇠 아이콘은 보안 채널로 연결되어 있음을, 회색 배경에 부러진 열쇠는 보안이 되지 않았음을 의미한다. 열쇠 아이콘의 모양 중에서도 두개의 이빨을 가진 것은 고급 암호화를 의미하고 하나의 이빨은 중급 암호화를 의미한다. 이때 보안 문서는 세션간에 디스크에 캐시 되지 않지만 저장할 수 있으며 HTML 문서 소스도 볼 수 있다. 즉, 보안 기능은 문서 조작에 영향을 주지 않고 문서의 전송에만 영향을 준다. 현재 넷스케이프 4.7에 정의된 보안 기능으로는 패스워드를 통한 시스템 보안, SSL 기반의 전송 프로토콜 보안, S/MIME 기반의 전자우편 보안, Java/JavaScript 보안, 인증서 및 보안 모듈 제공 기능 등이 있다.

현재 SSL 버전 2.0과 3.0 을 동시에 지원하며, 버전 3.0에서는 다음과 같은 암호화를 위한 PKCS#11 보안 모듈을 기반으로 블록 암호 알고리즘과 인증 기능을 위한 해쉬 함수들을 지원한다. 또한, HTML 문서에 포함된 Java Applet이나 JavaScript를 이용하면 망을 통해 시스템에 접근할 수 있기 때문에 프로그램 코드에 대한 강력한 접근 제어를 수행한다.

- 40비트 지원 RC2
- 40비트, 56비트 및 128비트 지원 RC4
- 56비트 지원 DES(CBC 모드)
- FIPS 140-1 호환, 168비트 지원 삼중 DES
- MD5와 SHA-1 MAC

Netscape 브라우저가 전자우편 도구로 사용되는 경우 전자우편 암호화와 서명, 뉴스그룹 메시지에 대한 서명 기능을 고려해야 한다. 디지털 서명되어 전송되는 모든 메시지는 인증서를 포함하게 될 것이며, 이를 통해 수신자는 다시 송신자에게 암호화된 메시지를 전송하는 것이 가능하게 된다. 현재, 전자우편 보안을 위해 S/MIME이 사용되며, 40비트 RC2와 56비트 DES암호화 알고리즘을 지원한다. 넷스케이프 브라우저는 인증서를 자신의 인증서, 타인의 인증서, 웹 사이트 인증서, 인증서 서명자의 인증서 등의 그룹으로 분류하여 관리한다. 이들 각 인증서들은 브라우저를 통해 획득될 수 있다.

4.2 익스플로러의 보안기능

강력한 보안 서비스를 두 통신 당사자간에 제공받고자 한다면 두 당사자간에 상호 신뢰해야만 한다. 즉, 통신 상대가 악의적인 사용자인지를 먼저 파악해야 하는 것이다. 이와 같은 기능을 익스플로러에서는 인증서를 통해 수행하는데, 통신 상대방의 인증서에 서명한 서명자를 신뢰함으로 동시에 통신 상대도 믿는 것이다. 일반적인 인증서는 통신 상대방의 공개 키에 대해 서명자가 디지털 서명하는 것을 의미한다. 여기에 공개 키와 쌍을 이루는 개인 키까지 포함하여 "디지털 ID"라 부르며, VeriSign이나 GlobalSign 등으로부터 유료 또는 무료로 개인 인증서와 웹사이트 인증서를 발급받을 수 있다. 이와 같은 인증서는 DER이나

Base64로 인코딩된 X.509와 PKCS #7 인증서 형태로 저장 관리되며 클라이언트 및 서버 인증, 프로그램 코드 서명, 안전한 전자우편 및 타임스탬핑, MS 신뢰 목록 서명 및 MS 타임스탬핑, IPsec 종단시스템과 IPsec 터널 종단 및 IPsec 사용자에 대한 인증, 파일 시스템 암호화, 윈도우 하드웨어 드라이버 검증, 윈도우 시스템 컴퍼넌트 검증과 같은 용도로 사용된다.

위와 같은 인증서의 기능 중에서, 클라이언트와 서버 인증은 인터넷을 통해 잘 알 수 없는 수많은 사이트에 접근해야 한다는 측면에서 볼 때 특히 중요하다. 이 클라이언트와 서버 인증을 효과적으로 수행하기 위해 익스플로러에서는 인터넷 사이트를 인터넷 영역, 로컬 인트라넷 영역, 신뢰할 수 있는 사이트 영역, 제한된 사이트 영역이라는 4개의 영역으로 구분해 주고, 사용자가 각 영역별로 적절한 보안 수준을 지정할 수 있도록 하고 있다. 이 정보는 브라우저의 상태표시줄 오른쪽에 나타나 현재 웹 페이지의 보안 영역을 표시한다.

또한, 인증서의 중요한 용도 중 하나인 프로그램 코드 서명 기능은 인터넷에서 프로그램을 전송하거나 실행할 때 원본을 믿을 수 있는 것인지 확인하기 위해 사용한다. 이 때문에 인터넷에서 사용자 시스템으로 프로그램을 전송할 때 익스플로러가 MS Authenticode 기법을 사용하여 프로그램의 ID를 확인하고, 프로그램의 인증서 유효성을 검사한다. 즉, 소프트웨어 공급업체의 ID와 인증서가 일치하는지, 인증서의 유효 기간이 끝나지 않았는지 등을 확인한다. 그러므로, 인터넷을 통해 프로그램을 배포하고자 하는 사용자는 VerSign과 같은 디지털 ID 발급자로부터 Active-X 응용 프로그램을 디지털 서명하기 위한 Authenticode 인증서를 먼저 획득해야 하는 것이다. 이 기법은 불안전한 프로그램을 전송하거나 시스템에서 실행하는 것을 방지하지는 못하지만 고의로 해를 주려고 만든 프로그램이 잘못 유포되는 것은 방지할 수 있다.

5. 웹 서버 보안

5.1 Apache 서버

Apache는 "A PatCHy server"라는 의미이며, 기존에 존재하는 코드와 여러 개의 패치 파일들로 구성되었기 때문에 붙여진 이름이다. 즉, NCSA httpd 1.3과 같은 대중화된 HTTP 서버내에 포함된 코드와 개념에 바탕을 두고 개발되었으며, 모듈화 되어 있어 다른 기능을 쉽게 첨가할 수 있는 장점을 가지고 있다. 최근 Apache 그룹은 CERT에서 지적된 CSS(Cross Site Scripting) 보안 문제를 해결한 Apache 1.3.12 버전을 발표했다[9][10]. 주요 개선점을 살펴보면, DSO(Dynamic Shared Object)를 지원하여 Apache 모듈들이 런타임시에 서버 처리 메모리로 로딩될 수 있으며, 전체 메모리 사용량이 현격히 줄어든다. FreeBSD, Linux, Solaris, SUNOS 등의 OS를 지원하며, Windows NT와 Windows 95/98, NetWare 5.x를 실험적으로 지원한다. Apache 소스파일을 재구조화했으며, Configuration 파일내의 "Module" 라인이 "AddModule"로 변경되었다. 이는 모듈을 사용자가 쉽게 첨가하게 하기 위한 것이다[11].

5.2 Apache-SSL

Apache-SSL은 Apache 서버와 SSLeay/OpenSSL을 기반으로 개발된 안전한 웹 서버로 비상업적인 목적으로 자유롭게 사용할 수 있다. SSLeay는 SSL을 호주의 Eric Young이 구현하여 공개한 것이며, SSL 2.0과 3.0, TLS 1.0을 지원한다. 현재 SSLeay-0.9.0까지 나와 있는 상태이다. OpenSSL은 SSLeay를 기반으로 암호 라이브러리와 SSL 2.0과 3.0, TLS 1.0을 지원하는 공개된 툴킷이다. 이 Apache-SSL은 완전한 소스코드가 제공되고, 클라이언트 인증 서비스와 128비트 기반 암호화

메커니즘이 제공된다. 2000년 1월 현재 apache_1.3.11+ssl_1.38 버전이 제공되고 있다[10][12].

Apache-SSL의 보안 측면을 좀더 살펴보면, 서버 인증서를 이용하여 사용자가 자신의 정보를 올바른 상대방에게 전달한다는 것을 확신할 수 있게 해주며, 암호화 통신을 지원하기 때문에 통신 중에 정보가 유출될 위험이 없다. 또한 서버상의 환경설정 파일을 이용하여 서버의 보안 정책을 수립할 수 있도록 하며, httpd.conf 파일 내에 디렉터리 별 접근 제어와 같은 인증 서비스에 대해 설정해 놓으면 구동 시 읽어서 이에 따라 서비스하게 한다. 이와 같은 보안 서비스를 올바르게 제공하기 위해 Apache-SSL 서버의 인증서가 필요한데, 공인된 인증기관으로부터 웹을 통해 받아올 수도 있으며 시험용으로 제공되는 더미 인증서를 직접 만들어서 사용할 수도 있다. 이는 SSL 설치 디렉토리에서 "make certificate"를 통해 쉽게 수행된다. 다시 말해, 안전한 웹 서버로 동작하기 위해서는 웹 서버 자신의 인증서와 클라이언트들에게 인증서를 생성해 주기 위한 CA용 인증서를 발행해야 한다[13].

6. 결 론

본 고에서는 초기 정보검색 도구로 사용되던 웹이 다양한 인터넷 비즈니스 도구로 확산되면서 요구되고 있는 웹 보안 요구사항 및 보안 기법, 그리고 실제 사용되고 있는 웹 브라우저 및 웹 서버에서 구현되어 제공되는 보안 기능에 대해 살펴 보았다. 웹 보안 필요성이 요구되던 초기에 다양한 방식 들이 제안되었으나 시장 논리에 의해 넷스케이프가 개발한 SSL이 웹 보안 프로토콜 표준으로 정착되어 현재 일반적으로 사용되고 있다. 인터넷 표준을 제정하는 IETF에서는 SSL을 기반으로 하는 TLS 1.0 규격을 개발하여 표준으로 권고하고 있는 상황이다.

현재 사용되고 있는 웹 브라우저 및 서버의 경우, 특별한 보안 취약성이 아직 보고되지 않고 있으나 웹 브라우저에 다양한 응용을 위한 플러그인 프로그램 및 외부 표시기 프로그램이 개발 보급되고 있으므로 이러한 프로그램에 보안 허점이 존재하지 않는지 여부를 늘 주의 깊게 살펴 보아야 할 것이다. 최근 인터넷의 급속한 보급 및 활용에 힘 입어 전자상거래 서비스와 같은 보안 기능을 요구하는 새로운 서비스가 도입되면서 웹 보안의 중요성이 더욱 강조되고 있으므로 향후 지속적인 연구 개발이 요구된다.

참고문헌

[1] 강신각, Web Security and Payments, WWW-KR 워크샵, 자료집 4-2호, 1996.11.

[2] 박정수, 강신각, 박성열, 월드 와이드 웹 보안 기술 및 동향, 정보과학회지, 한국정보과학회, 1997.4.

[3] G. Bossert, et al., Considerations for Web Transaction Security, RFC 2084, 1997.1.

[4] R. Fielding, J. Gettys, J. Mogul, H. Frystyk and T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, draft-ietf-http-v11-spec-07.txt, 1996.8.

[5] E. Rescorla, et al., The Secure HyperText Transfer Protocol, RFC 2660, 1999.8.

[6] A. Freier, P. Karlton, and P. Kocher, The SSL Protocol Version 3.0, http://www.netscape.com/eng/ssl3/3-spec.ps. 1996.3.

[7] T. Dierks, et al., The TLS Protocol 1.0, RFC 2246, 1999.1.

[8] J. Weeks, etc, CCI-Based Web Security : A Design Using PGP, WWW Journal 95, 1995.

[9] "The Netcraft Web Server Survey," http://www.netcraft.com/survey/, 2000.2.

[10] T. Hudson and E. Young, " SSLeay and SSLapps FAQ", http://www2.psy.uq.edu.au/~ftp/Crypto/, 1998.9.

[11] "Apache-SSL", http://www.apache-ssl.org/, 2000.2

[12] Cert Advisory CA-2000-02, "Malicious HTML Tags Embedded in Client Web Requests: Cross Site Scripting Info," http://www.cert.org/advisories/CA-2000-02.html, 2000.2.3.

[13] "SSL Certificate Overview," http://www.thawte.com/certs/server/request.html.

강 신 각

1984년 충남대학교 전자공학과 (공학사)
1987년 충남대학교 전자공학과 (공학석사)
1998년 충남대학교 전자공학과 (공학박사)
1984년-현재 한국전자통신연구원 표준연구센터 책임연구원
1997년-현재 ITU-T Q.13/7 Rapporteur
관심분야 : 멀티미디어통신, 차세대인터넷, 정보보호

박 정 수

1992년 경북대학교 전자공학과 (공학사)
1994년 경북대학교 전자공학과 (공학석사)
1994년-현재 한국전자통신연구원 표준연구센터 연구원
관심분야 : 정보보호, 차세대인터넷