HTML5 로 구현한 웹 어플리케이션 보안 취약성 개선

김광수*, 장영수, 최진영
*고려대학교 컴퓨터 정보통신대학원 소프트웨어공학과
e-mail : kskim@korea.ac.kr, jyskkh@chol.com, choi@formal.korea.ac.kr

The Soft Security Improvement of HTML5 With WEB Application

Kwang Su Kim*, Young Su Jang, Jin Young Choi
*Dept. of Software Engineering, Graduate School of Computer & Information Technology, Korea University

요 약

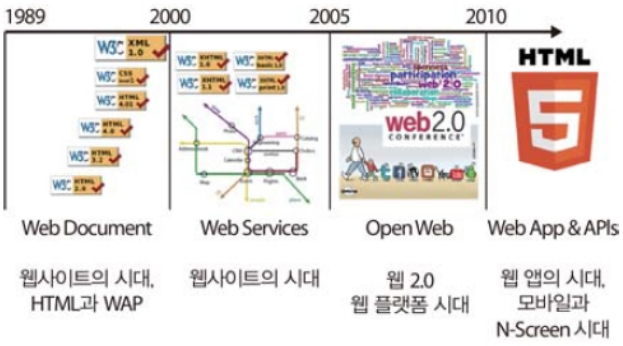
HTML5 는 웹 문서를 작성하기 위한 HTML(Hyper Text Markup Language)의 차세대 웹 표준 이다. HTML5 는 아직 개발 중에 있으며 2014 년 하반기에 최종표준이 발표 될 것으로 전망 된다. HTML5 는 이전 버전의 HTML 과 호환성을 유지하면서 개발자에게 동영상, 위치정보, 소켓통신 및 다양한 미디어 서비스 을 별도의 플러그인 없이 HTML5 의 확장된 표준 태그로 Dynamic 한 기능을 구현할 수 있게 한다. 그러나 HTML5 에 새롭게 추가된 일부 표준 태그 에서 웹 어플리케이션(Web application) 서비스의 데이터 보안 취약점이 발견되었다. 본 논문에서는 HTML5 로 웹 어플리케이션 소프트웨어 개발 과정에서 발견된 표준 태그 및 API 보안 취약점을 분석하고 공격 대상이 되는 소스코드 의 취약점을 개선 하였다. 보안에 취약한 소스코드 취약점을 개선하여 외부 공격자의 위협으로 부터 보안 취약점을 예방 할 수 있는 대응방법을 제안한다.

1. 서론

W3C(World Wide Web Consortium) 의 대표적 Markup Language 표준인 HTML(Hypertext Markup Language) 은 인터넷 통신기술과 함께 활용 범위가 급속도로 증가 하였다. 하지만 활용범위가 확대되고 이에 따른 늘어나는 기능 요구사항 에 비해 해결할 수 없는 문제점들이 존재 하였다. HTML4 버전은 오랜 기간 동안 성능 확장 없이 사용 되어 오고 있었기 때문에 다양한 비즈니스 요구사항을 충족시켰어야 했던 브라우저 업체들은 HTML4 의 제한된 기능 등이 주요 문제사항으로 대두 되었다. 이러한 문제 인식 은 W3C 가 아닌 브라우저 업체들로부터 시작되었으며, Apple, Google, Mozilla, Opera 등 웹 Browser Vendor 가 2004 년도에 WHATWG(Web Hypertext Application Technology Working Group) 표준화 기구를 만들어 HTML 규격을 확장하기 시작한 것이 HTML5 의 등장 배경이라고 할 수 있다. 후에 W3C 도 이러한 움직임에 대한 타당성을 인정하고, 2007 년도에 W3C 는 HTML WG 활동을 통해 WHATWG 의 규격들을 수용하는 방향으로 공식적인 W3C 표준으로서 HTML5 표준화 작업을 시작하였다 [1]. HTML5 는 새로운 웹의 표준 이라고 할 수 있으며 스마트미디어 환경을 위한 플랫폼 기술로 주목을 받고 있다.

2. HTML5 의 발전과정 및 특징

웹 기술의 진화 발전 과정을 살펴보면(그림 1) 과 같이 4 단계로 구분할 수 있다. 1 단계는 1989~1999 년까지의 기간으로 HTML, URL(Unified Resource Locator), HTTP(Hypertext Transfer Protocol) 세 가지 기술에 기초해 웹 기술이 제안되고 보다 나은 인간 중심의 정보처리 및 지식공유를 목표로 하는 단계였다. 2 단계는 2000~2004 년까지의 기간으로 XML(Extensible Markup Language) 에 기반하여 인간 중심의 정보 처리뿐 아니라 다양한 Device, Service, Multimedia 를 연결하는 것을 목표로 하는 단계였다. 3 단계는 2005~2009 년까지로 Google, Amazon, Wikipedia 등의 성공과 함께 웹 산업을 제 2 의 전성기로 이끌며 다양한 신규 서비스가 등장할 수 있는 기반을 만들었다. 마지막 4 단계는 2010 년 이후부터 현재까지로, Smart Phone, Tablet 등 다양한 Mobile 기기 들을 대상으로 HTML5 와 웹 API 를 통해 한 단계 진화된 웹 응용 환경을 제공하며, 위치 및 Social 정보 등을 결합하는 통합 응용 플랫폼으로서 웹이 자리 잡아 가는 단계라 할 수 있다[2].



(그림 1) 웹 기술의 진화 발전 과정

HTML5 는 웹 Markup 기술을 기반으로 웹 어플리케이션 기술, 그리고 웹 플랫폼 기술로 진화하고 있다.

2-1. HTML5 API 특징 및 지원현황

HTML5 표준은 Semantic Markup 부분과 API 부분으로 크게 나누어 진다. Semantic Markup 부분은 기존 HTML4 버전 보다 정확한 의미 표현이 가능하도록 새로운 Markup 태그가 추가되었다. 이를 통해서 검색엔진 이나 웹 Contents 를 기반으로 하는 다양한 서비스는 향상된 기능을 제공할 수 있게 되었다. API 의 경우 웹 어플리케이션 기반의 개발을 지원 하기 위해 HTML4 까지는 존재하지 않았던 새로운 기능으로 추가된 부분이다. HTML5 에 추가된 많은 기능이 있으나 주요 특징들은 <표 1> 과 같이 정리해 볼 수 있다[3]. 또한 <표 2> 는 웹 Browser Vendor 별 HTML5 주요기능에 대한 브라우저 지원 현황을 보여준다[4].

<표 1> HTML5 의 주요기능 및 관련 표준

주요기능	설명
Web Form	사용자의 입력정보를 받기 위해 사용되는 입력형태에 대한 정의에 사용되는 마크업, 애트리뷰트와 이벤트
Canvas	웹에서 즉시모드(immediate mode)로 2차원 그래픽을 그리기 위한 API와 Canvas 내 각종 객체를 회전, 변환하고 그레디언트, 이미지 생성 등 각종 효과를 주는 기능에 대한 API
SVG	XML 기반의 2차원 백터 그래픽을 표현하기 위한 언어
Video/Audio	Video는 비디오 또는 영화를 보여주기 위해 사용되는 미디어 엘리먼트이며, Audio는 사운드나 오디오 스트림 을 표현하기 위한 미디어 엘리먼트
Geolocation	디바이스의 지리적 위치 정보를 제공하는 API 표준
Offline Web Application	인터넷 연결이 지원되지 않는 경우에도 웹 응용이 정상적으로 수행될 수 있도록 지원 하는 기능으로 응용에 대한 캐싱과 데이터에 대한 캐싱으로 구성
Web SQL Database	다양한 표준 SQL을 사용해 질의할 수 있는 데이터베이스 기능에 대한 API
Local Storage	기존의 쿠키의 기능을 개선하기 위한 목적으로 개발된 기능으로 웹 클라이언트에서 키 와 값이 쌍으로 구성된 데이터를 영구적으로 저장하는 기능
Web Socket	웹 응용이 서버 측의 프로세스와 직접적인 양방향 통신을 위한 API
Web Worker	웹 응용을 위한 쓰레드(Thread) 기능에 대한 API

<표 2> 브라우저별 HTML5 지원 현황

기능	IE	Chrome	Firefox	Safari	Opera
Canvas	○	○	○	○	○
Video	○	○	○	○	○
SVG	○	○	○	○	○
Geolocation	○	○	○	○	○
Web Socket	X	○	X	○	X
Web Worker	X	○	○	○	○
Web SQL	X	○	X	○	○

2-2. HTML5 가 웹 환경에 미치는 영향

표준화된 웹 환경의 확산은 멀티미디어를 비롯한 다양한 기능들을 제공하기 위해서 사용 되고 있는 비표준 인터넷 웹 환경 (ActiveX, Flash, Silverlight 등 별도 프로그램 설치) 이 점차 감소 될 것이다.
또한 인터넷 상에서 다양한 어플리케이션을 구현하고, 이를 누구나 브라우저로 접근하여 사용 할 수 있게 함으로서 Apple(iOS) 및 Google(Android) 등 OS Platform 에 대한 의존도가 감소한다[5]. 사용자의 경우 인터넷에만 접속하면 Smart Phone, Tablet, 개인 PC 등 다양한 기기 에서는 물론 Apple 이나 Google 등의 Vendor 에 상관없이 Software 나 Contents 등을 이용 가능할 것이다[6].

3. 시큐어 코딩(Secure Coding) 기법

소프트웨어의 개발과정에서 개발자의 지식 부족이나 논리적 오류, 실수 또는 프로그래밍 언어별 고유한 약점 등 다양한 원인으로 발생할 수 있는 취약점을 최소화 하기 위하여, 설계 단계부터 보안을 고려하여 코드를 작성하는 제작방식을 의미한다[7]. 실제로 해킹의 75%가 소프트웨어의 취약점을 악용해 이루어지고 있다[8]. 시큐어 코딩 기법을 적용하여 소프트웨어를 제작하게 되면 취약점이 대폭 줄어들어 보안성이 향상될 뿐만 아니라, 보안에 대한 고려 없이 개발된 소프트웨어의 유지보수 및 수정비용은 시큐어 코딩에 드는 비용의 수십 배에 달하기 때문에 시큐어 코딩은 비용적으로도 매우 효과적인 개발방식으로 평가 받고 있다[9].

<표 3> S/W 개발단계별 결함 수정비용 분석

구분	설계단계	코딩단계	통합단계	베타제품	제품출시
설계과정	1 배	5 배	10 배	15 배	30 배
코딩과정	-	1 배	10 배	20 배	30 배
통합과정	-	-	1 배	10 배	20 배

이와 같이 개발 초기 단계 에서부터 보안에 초점을 맞추어 소프트웨어를 제작 한다면 잠재적으로 발생할 수 있는 보안 취약점의 가능성을 최소화 할 수 있고 외부 공격자의 위협으로부터 비교적 안전한 소프트웨어를 만들 수 있다.

3-1. CERT 코딩 기법

카네기 멜론 대학교의 소프트웨어 공학 연구소에서 관리하는 CERT(Computer Emergency Response Team)는 시큐어 코딩에서도 활발한 활동을 하고 있는데, 특히 코딩 규칙 및 가이드(Coding Rules/Guide) 의 표준화 작업을 수행하고 있다. CERT 는 Secure Coding Standard 를 프로그래밍 언어별 특징을 기준으로 분류하여 안전한 코딩을 언어가 갖는 특징 별로 구분하여, 각 언어의 사용자 및 학습자가 수월하게 접근하도록 하고 있다. CERT 에서 제공하는 “Secure Coding Standard for Java” 는 2013.9 월 기준으로 19 개의 카테고리로 구성되어 있고 각 항목은 지속적으로 갱신되고 있다[10].

3-2. CWE 코딩 기법

CWE(Common Weakness Enumeration)는 미국토안보부(The U.S. Department of Homeland Security) 에서 관리하고 있으며, 소프트웨어의 취약성을 사전식으로 분류하여 프로그래머가 쉽게 접근할 수 있도록 구성되어 있다. 이것은 수집된 취약점 항목을 뷰, 카테고리, 취약점, 복합요소를 기준으로 분류하여 살펴볼 수 있는 특징을 가지고 있으며, 취약점 항목에 대한 갱신은 현재에도 지속적으로 이루어지고 있다[11].

4. HTML5 API 보안 위협요소

웹 환경에서 Dynamic 한 웹 어플리케이션 개발을 지원하기 위한 차세대 웹 표준 기술인 HTML5 의 사용이 점차 증가 하고 있다. HTML5 기술은 웹 환경에서 플러그인 없이 다양한 미디어 서비스를 구현할 수 있게 한다. Microsoft, Google, Facebook 등 IT 기업들은 HTML5 기술을 이용하여 다양한 웹 어플리케이션 및 모바일 서비스를 구축 하고 있다. 하지만 HTML5 의 일부 API 에서 데이터 보안 취약점이 발견되었다. HTML5 의 API 보안 위협 요소는 다음과 같다.

4-1. HTML5 의 API 보안위협 요소

4-1-1. Web Storage 보안위협

Web Storage 의 보안 이슈는 저장되어 있는 데이터에 대한 불법 적인 접근을 사용자 측에서 인지할 수 없다는 것이다. Web Storage 에 대한 모든 접근 및 제어는 자바스크립트를 통해 이루어지며, 도메인이 XSS(Cross-Site-Scripting) 등 스크립트 기반 취약점을 가지고 있을 경우 공격자는 사용자의 브라우저에 있는 모든 Web Storage 데이터들을 사용자 모르게 조작하거나 가져올 수 있다[12].

4-1-2. CDM (Cross Document Messaging) 보안위협

기존의 웹 표준에서는 보안을 위해 Cross Domain 간에 웹 페이지나 데이터의 요청을 할 수 없도록 금지함으로써 악의적인 사이트가 합법적인 사이트로부터 데이터를 가로채는 것을 막는다. HTML5 에서는 postMessage API 를 통하여 서로 다른 도메인간 에 HTTP 요청을 할 수 있다. 하지만 검증된 도메인만 허용하지 않을 경우 보안상의 문제점으로 나타날 수 있다[13].

4-2. 보안 위협 요소 취약점 대응방안

4-2-1. Web Storage 취약점 개선

HTML5 에서 지원하는 로컬 스토리지인 Web Storage API 는 기존 쿠키(Cookie) 의 단점을 보완할 수 있고 네트워크 사용량 절약 및 성능 개선을 위해 도입된 기술이다. 그러나 XSS 등의 보안 취약점을 가지고 있을 경우 자바 스크립터에 의해 데이터 조작이나 변경이 가해질 수 있다. <표 4> 는 Web Storage API 를 이용하여 데이터를 클라이언트에 저장 하는 소스코드의 개선 전 코드와 개선 후 코드의 예 이다.

<표 4> Web Storage API 개선 전 코드와 개선 후 코드

개선 전 코드
window.localStorage.setItem('value', content.value); window.localStorage.getItem('value');
개선 후 코드

<script src=”http://crypto-js.googlecode.com/svn/tags/3.1.2/build/rollups/
tripledes.js” />
var key = “Deskey”;
var value = window.localStorage.getItem('value');
CryptoJS.DES.decrypt(value, key);
var enValue = CryptoJS.DES.encrypt(content.value, key)
window.localStorage.setItem('value', enValue);

Web Storage API 를 이용하여 오프라인 로컬 저장소에 비밀데이터나 민감한 데이터를 저장 하는 경우 데이터를 암호화 하는 것을 권장한다. 개선 후 코드 에서는 Google Code 의 Triple DES 알고리즘을 이용하여 데이터를 저장 하였다. 암호화 하여 데이터를 저장할 경우 클라이언트(Client) 측에 키(key)값을 저장하면 안되고, 서버(Server) 측에 저장해야만 데이터를 안전하게 보호 할 수 있다.

4-2-2. CDM (Cross Document Messaging) 취약점 개선

HTML5 는 서로 다른 도메인 URL 상의 웹 페이지끼리 메시지를 비동기 송/수신 할 수 있는 CDM(Cross Document Messaging) API 를 제공한다. CDM API 를 이용하면 서로 다른 도메인간의 정보를 재 가공 하거나 인터넷 상에 쉽게 노출 시킬 수 있다. 하지만 요청한 도메인에 대한 검증이 되지 않았을 경우악의적

인 공격에 노출될 위험성 커지게 된다. <표 5> 는 CDM API 를 이용하여 개발된 소스코드의 개선 전 코드와 개선 후 코드의 예 이다.

<표 5> CDM API 개선 전 코드와 개선 후 코드

개선 전 코드

```
window.postMessage(data, '*')
window.addEventListener('message', receiver, false);
function receiver(e) {
   document.getElementById('text').innerHTML
   = e.data
}
```

개선 후 코드

```
window.postMessage(data, targetOrigin, [ports])
window.addEventListener('message', receiver, false);
function receiver(e) {
   if (e.origin === 'http://www.example.com:8080') {
      document.getElementById('text').innerHTML
      = e.data
   }
}
```

CDM API 를 통해 데이터를 처리할 경우 송신자의 도메인 주소를 확인하고 신뢰 할 수 있는 도메인만의 요청만을 받아야 한다. 또한 데이터 처리 전 Dom-MessageEvent-Origin 속성 과 전송포트를 통하여 검증된 도메인으로부터의 요청인지 확인하고 수신된 데이터의 입력 값을 검증 해야만 안전한 소프트웨어를 만들 수 있다.

5. 결론 및 향후 연구

HTML5 는 아직 표준화 작업이 완료된 기술이 아닌 진행 중인 기술이며 표준을 위한 작업 이 진행되고 있다. HTML5 에는 새로운 태그(Audio / Video / Canvas 등) 와 API 가 추가 되었다 이로 인해 어플리케이션의 다양한 멀티 미디어 조작 과 Data Handling 및 구현이 가능하게 되었다. 하지만 이러한 기술들을 통하여 다양한 이점을 얻을 수 있는 반면 동시에 악의적인 용도로 남용할 수 있는 공격의 범위도 앞서 설명한 것처럼 증가하게 되었다. 결국 HTML5 를 기반으로 하는 모든 웹 어플리케션 은 새로운 보안 위협에 직면하게 될 것으로 예측된다. 이와 관련된 공격을 예측하고 대응방법을 개발하는 연구가 필요하다. 본 논문에서는 Web Storage API 에 대해서는 데이터 저장 시에 plain text 데이터를 Triple Des Symmetric-key Algorithm 을 사용하여 데이터 암호화 방법을 제안하였다 또 CDM API 에서는 원격지의 서로 다른 도메인간 데이터 송/수신 시 Dom-MessageEvent-Origin 속성 사용하여 검증된 도메인만 접근이 허용하도록 제안 하였다. 위에 언급한 것과 같이 API 의 취약점 개선을 통하여 웹 어플리케이션 구축 시 안정적으로 데이터를 보호 할 수 있을 것으로 예상 된다

참고문헌

[1] 이승윤·박기식(2012), "HTML5 와 스마트미디어 플랫폼," 한국통신학회논문지 29(10) 25-29.

[2] 전종홍·이승윤(2012), "HTML5 기반의 웹 플랫폼 기술 표준화 동향," 전자통신동향분석, 27(4) 83-95.

[3] 이원석(2011), "HTML5 와 모바일 웹," TTA Journal 128 50-54.

[4] 남지혁·서창갑, "HTML5 를 이용한 모바일 웹사이트 구현," 디지털정책연구, 11(1) 165-172.

[5] 인터넷 글로벌 경쟁력 강화를 위한 차세대 웹 표준 확산 추진계획, 방송통신위원회, 2012. 7.

[6] 이은민(2011), "HTML5 가 웹환경에 미치는 영향," 정보과학회지, 29(6) 55-60.

[7] "소프트웨어 개발 보안가이드", 행정안전부, 2012.5

[8] "Now is the time for security at Application Level" Gartner, Dec. 2005

[9] "The Economic Impacts of Inadequate Infrastructure for Software Testing", 2002.5, NIST

[10] The CERT Sun Microsystems Secure Coding Standard for Java. http://www.securecoding.cert.org/confluence/display/java/

[11] CWE(Common Weakness Enumeration). http://cwe.mitre.org/

[12] 강석철(2013), "HTML5 신규웹서비스환경에서의 보안이슈," Internet & Security Focus, 12.

[13] 김현순(2012), 모바일 기술자경력관리시스템의 HTML5 보안취약점분석에 관한 연구, 숭실대학교 정보과학대학원 석사학위 논문.