소스코드 보호를 위한 보안 웹 브라우저 개발

오창현*, 강정호*, 전문석*
*숭실대학교 컴퓨터학과
e-mail:kky3127@naver.com

Secure Web Browser Development for Source Code Protection

Changhyun Oh*, Jungho Kang*, Moon-Seog Jun*
*Dept of Computer Science & Engineering, Soongsil University

요 약

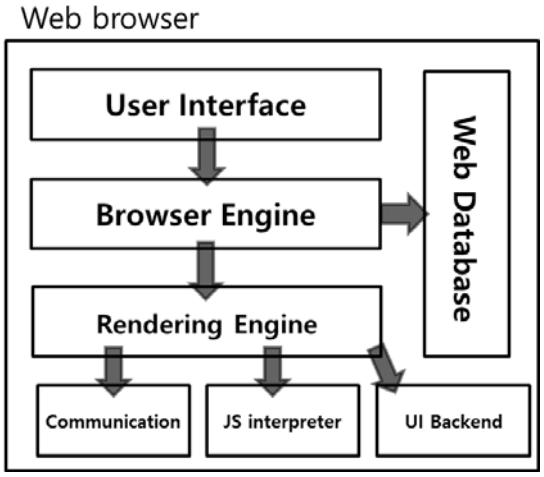
현재 인터넷에서 제공되는 서비스 형태는. 서버에서 서비스를 제공하면 클라이언트는 웹 브라우저를 통해 서비스를 제공 받는다. 이 때 전달받은 소스코드는 평문의 텍스트 형태로 클라이언트에게 노출된다. 개발자 입장에서 해당 소스코드를 보호 하고자 할 경우 조치할 수 있는 방법이 존재하지 않는다. 따라서 소스코드를 보호 할 수 있는 방법에 대한 대책으로 보안 웹 브라우저를 제안한다.

1. 서론

현재 인터넷에서 제공되는 대부분의 서비스 형태는 서버에서 서비스를 제공하면 클라이언트에서는 웹 브라우저를 통해 서비스를 제공 받는 형식이다. 여기서 클라이언트에게 제공되는 서비스는 HTML, XML, DHTML, VRML, XHTML, Java Script, VB Script 등의 클라이언트 사이드 언어로 전달받는다. 클라이언트 사이드 언어란 웹 브라우저에서 해석되어 작동되는 언어를 총칭한다[1]. 여기서 발생하는 문제는 전달받은 소스코드는 평문의 텍스트 형태로 클라이언트의 사용자에게 노출된다는 점이다. 또한 개발자의 부주의로 인해 소스코드 노출의 심각한 문제가 발생할 수 있다. 예를 들면 개발 당시 어플리케이션 프로그램에 대한 디버깅 정보를 HTML 주석을 이용해서 많이 작성한다. 이러한 주석을 개발이 완료된 후 서비스 오픈 하기 전에 모두 삭제해야 하지만 그렇지 않고 종료하는 경우가 빈번히 발생한다. 공격자는 어플리케이션 프로그램에 대한 자세한 정보를 얻기 위해 웹 브라우저에서 소스 보기를 이용하여 기밀정보를 찾는다. 개발자 입장에서 해당 소스코드를 보호 하고자 할 경우 조치할 수 있는 방법이 존재하지 않는다. 본 논문에서는 이에 대한 대책으로 서버로부터 전달받는 소스코드를 암호화 하여 웹 브라우저만 소스코드를 해독하고 서비스는 화면에 출력하면서 안전하게 소스코드는 보호 할 수 있는 보안 웹 브라우저를 제안한다.

2. 관련 연구

2.1 웹 브라우저 구조



(그림 1)웹 브라우저 구조

웹 브라우저는 User Interface, Browser Engine, Rendering Engine, Web Database, Communication, JavaScript Interpreter, UI Backend로 구성되어있다[2]. User Interface는 브라우저에서 사용자에게 보여지는 모든 구성요소이다. Browser Engine은 User Interface와 Rendering Engine사이에서 동작을 제어한다.[3] Rendering Engine은 요청받은 콘텐츠를 표시한다. 예를 들면 HTML을 요청받았을 경우 HTML, CSS를 Parsing하여 화면에 보여준다. Communication은 HTTP요청과 같은 네트워크 호출시 사용된다. 이것은 플랫폼 독립적인 인터페이스로 각 플랫폼 하부에서 실행된다. UI Backend는 기본적인 장

치를 그리는 역할을 수행한다. 플랫폼에서 명시되지 않은 일반적인 인터페이스로 OS인터페이스 체계를 이용한다. JavaScript Interpreter는 자바스크립트 코드를 해석하고 실행한다. Web Database는 자료를 저장하는 계층으로 쿠키 저장과 같이 모든 종류의 자원을 하드디스크에 저장한다.

2.2 프로그래밍 언어

서버사이드(Sever-Side)란 간단히 "웹 서버측에서 하는 작업들"이라고 말할 수 있다. 여기서 말하는 작업이란 구체적으로 웹 브라우저(클라이언트)에서 넘어온 자료를 데이터베이스에 저장 한다든지, 어떤 수학적인 계산을 하여 결과를 만들어 낸다든지 하는 것을 말한다. 이런 작업을 담당하는 것이 웹 프로그램이다. 웹 프로그램의 종류는 PHP, ASP, Perl, Python등이 많이 쓰인다.

클라이언트 사이드란 웹 브라우저(클라이언트 사이드)를 사용하면 서버의 작업량을 줄일 수 있다. 서버가 작업해야 할 부분 중에서 클라이언트가 할 수 있는 작업을 스스로 처리하기 때문에 서버의 작업량을 줄여줄 수 있어 효율적이다. 이렇게 "클라이언트 스스로 일을 처리할 수 있도록 하여 서버의 효율성을 높일 수 있도록 하는 것"이 클라이언트 사이드 언어이다. 클라이언트 사이드 언어로는 자바스크립트와 이외의 대부분의 스크립트 언어가 있다. 플래시 액션 스크립트도 클라이언트 사이드 언어라고 할 수 있다. 이런 스크립트 언어는 웹 서버에서 웹 브라우저로 전송된 후 실행된다.

3. 제안

제안하는 보안 웹 브라우저는 사용자에게 공개되는 Public 모듈과 비공개되는 Secure 모듈로 구분된다.

Public 모듈은 사용자측에게 서비스 요청을 받고 응답을 주는 역할을 수행하며 Secure 모듈은 Server와 암호화된 통신을 주고받는 역할을 수행한다.(그림 2을 참고)

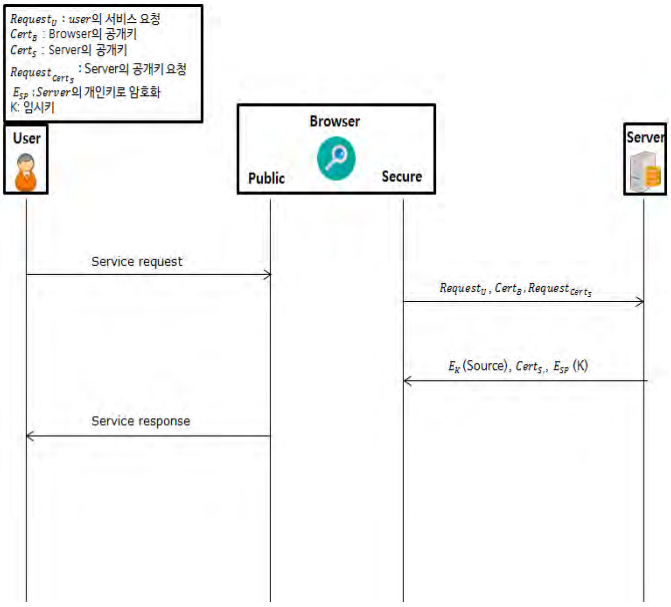
Step1. 사용자가 필요한 서비스에대한 URL을 요청한다.

Step2. Secure 모듈은 먼저 Server에게 User가 보낸 요청, 자신의 인증서, 서버의 인증서 요청을 전송한다.

Step3. 메시지를 전송받은 서버는 서버 인증서와 자신의 개인키로 암호화된 임시키, 임시키로 암호화된 소스코드를 전달한다.

Step4. 사용자는 웹 브라우저에 출력된 서비스를 제공받는다.

웹 브라우저는 서버로부터 전달받은 코드는 브라우저와 서버간에 합의된 키로 암호화 되어있기 때문에 복호화를 먼저 수행한 후 사용자가 볼 수 있도록 브라우저상에서 표현되고 Web Database에 저장되는 자료는 복호화 되는 소스 코드가 아닌 처음에 전달받은 암호화된 코드만이 저장될 수 있도록 한다.



(그림 2) 보안 웹 브라우저 프로토콜

4. 결론

현재 대부분의 사람들은 웹 브라우저를 통해 서비스를 제공받는다. 그러나 기존의 웹 브라우저는 클라이언트 사이드 언어로 쓰여진 소스코드에 대해서는 보호대책이 존재하지 않았다. 본 논문에서 제안된 보안 웹 브라우저를 이용하는 사용자는 기존의 웹 브라우저들과 같이 서비스를 제공받을 수 있으며 소스코드의 암·복호화를 수행하는 보안 웹 브라우저를 설계 하였다. 이를 통해 기대할 수 있는 효과로는 소스코드에 취약점이 존재하더라도 캡슐화하여 감출 수 있고, 개발자나 저작권자의 자산을 안전하게 보호할 수 있다는 점 등 이 있다.

참고문헌

[1] 차재복 "정보통신기술 용어해설" http://www.ktword.co.kr/abbr_view.php?m_temp1=4852&m_search=programming

[2] Tali Garsiel, "How Browsers Work:Behind the scenes of modern web browsers", https://www.html5rocks.com/en/tutorials/internals/howbrowserswork

[3] Gupta, Vineet, "HowBrowsersWork - Part 1 - Architecture", http://www.vineetgupta.com/2010/11/how-browsers-work-part-1-architecture/

[4] Zheng, S., Song, R., Wen, J. R., & Wu, D. (2007, August). Joint optimization of wrapper generation and template detection. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 894-902). ACM.