한국시뮬레이션학회 논문지
Vol. 20, No. 1, pp. 29-37 (2011. 3)

쿠키파일의 보안성 향상을 위한 저장 방식

심원태[1] · 최요한[2] · 서희석[2†] · 노봉남[3]

A Storage Method to Enhance Cookie File Security

Won-Tae Sim · Yo-Han Choi · Hee-Suk Seo · Bong-Nam Noh

ABSTRACT

Cookie file can be properly protected by designing security zone to enhance the safety of cookie file vulnerable to cyber attacks. In this paper, the model, in which cookie file is stored in the security area and the current visiting page is closely linked with cookie, is proposed to help users utilize in the same existing way, as well as enhance the security of user cookie files. Even if attacker tries to compromise web browser's cookie folder, the security of other cookies can be preserved. It is possible since the folder has the only cookie for the current web page where user is visiting.

Key words : Web, Cookie, Security, HTML

요 약

공격자의 공격으로부터 취약한 쿠키 파일의 안전성 향상을 위한 보안영역을 설계하여 공격자의 공격으로부터 사용자의 쿠키 파일을 안전하게 관리 할 수 있다. 본 논문에서는 쿠키 파일을 보안영역에 저장함으로써 사용자가 현재 방문하고 있는 웹 페이지와 유기적으로 연결되어 웹 페이지에서 쿠키 정보를 기존의 방식과 같이 사용 할 수 있도록 도와주는 동시에 사용자의 쿠키 파일의 안전성을 향상 시킬 수 있는 모델을 제안한다. 공격자가 사용자가 사용하는 웹 브라우저의 쿠키 폴더에 대한 공격을 시도 한다할지라도 사용자가 현재 방문 중인 웹 페이지에 대한 쿠키 파일만 존재하기 때문에 다른 쿠키 파일에 대한 안전성을 유지 시킬 수 있다.

주요어 : 웹, 쿠키, 보안, HTML

1. 서 론

정보 통신의 비약적인 발전은 인터넷을 통한 개인간, 기업간, 국가간의 정보 교류를 증진 시켰고, 이러한 현상은 더욱 가속화, 복잡화되고 있으며 스마트폰의 등장으로 개인은 언제 어디서나 웹에 접근이 가능하다[1].

웹 접근성의 향상은 개인의 웹서비스 활용도를 증가시켰으며, 사용자의 요구를 충족시키고, 그 대가로 금전적 이득을 추구하는 전자상거래시스템 등을 자리 잡게 되였다.

이러한 시스템들은 사용자의 신용카드 및 개인정보를 요구하고 이를 시스템에서 이용하기 때문에 개인 정보 관련 보안요구사항을 만족시키기 위해 많은 노력을 기울이고 있으나, 현재 보안에 취약한 응용프로그램들로 인해 개인정보 침해사고가 빈번하게 발생하고 있다.

특히, HTTP 프로토콜의 단순하고 간결성으로 인해 발생 되는 문제를 해결하기 위해 도입된 Cookie 시스템은 사용자의 개인정보를 단순한 텍스트 형태로 저장한다. 그리고 이를 보안적 요소가 고려되지 않은 파일 시스템에 저장함으로서 사용자에게 편의성을 제공한다. 하지만 Cookie 시스템의 이러한 단순성으로 인해서 저장된 개인정보의 침해 위협의 가능성이 대두되고 있다.

보안이 취약한 쿠키 파일에 대한 해결방안으로 현재는 쿠키파일의 내용을 암호화 방법을 연구하고, 이를 통해 보안 위협을 극복하려는 시도가 이루어지고 있다. 하지만 이러한 방식은 오늘날의 항상 된 컴퓨팅 환경으로 암호화 기법에 대한 정보와 암호화 시 사용된 키 값 등의 정보를 공격자가 획득한 경우 쉽게 쿠키파일에 저장된 값을 열람 할 수 있다는 위험이 존재한다.

접수일(2010년 12월 20일), 심사일(1차 : 2011년 1월 13일, 2차 : 2011년 3월 14일), 게재 확정일(2011년 3월 16일)
[1] 한국인터넷진흥원
[2] 한국기술교육대학교 컴퓨터공학부
[3] 전남대학교 시스템보안연구센터
주 저 자 : 심원태
교신저자 : 서희석
E-mail; histone@kut.ac.kr

이 뿐만 아니라 현재 웹 브라우저에서 기본적으로 지정한 쿠키폴더를 사용자가 변경하지 않고 그대로 사용하는 경우가 많다. 이러한 경우 쿠키 파일을 공격자의 서버로 전송하는 악성코드에 사용자의 컴퓨터가 감염되었을 경우 현재 사용자가 웹 페이지를 방문 중이지 않을 때에도 사용자의 컴퓨터에 저장되어 있는 쿠키 파일 전체가 위험에 노출 될 수 있다.

본 논문은 사용자 컴퓨터에 저장되는 쿠키 파일에 대한 보안성을 향상시키기 위해 이전까지 쿠키파일이 저장하기 위해 사용된 폴더와는 별도의 쿠키파일 보안영역을 설계함으로써 관련 보안 위협을 줄이고자 한다.

2. 쿠키파일의 필요성

2.1 HTTP 프로토콜의 단점

Hyper Text Transfer Protocol은 웹서비스를 이용하기 위한 통신 규약으로 현재 가장 많은 트래픽을 발생시킬 만큼 폭넓게 사용되고 있다. 사용자들에게 정적인 페이지만을 제공하던 초기버전에서, 증가하는 사용자들의 다양한 욕구를 충족시킬 수 있도록 하는 동적인 페이지를 제공하기 위한 기술이 개발되었다.

이러한 기술의 변화는 Cookie 시스템을 도입하였으며, Cookie 시스템은 사용자의 개인정보를 클라이언트에 텍스트 파일 형태로 저장하여 사용자가 이전에 방문한 내용을 담고 있다.

사용자가 웹 페이지를 다시 방문 하였을 경우에 쿠키파일에 담겨 있는 정보를 토대로 사용자에게 맞추어진 웹 페이지를 제공할 수 있게 되었다.

2.2 사용자에게 맞추어진 웹 제공 가능

Cookie 시스템을 사용하는 웹 페이지를 방문하는 경우 사용자는 웹 페이지에서 자신에게 최적화된 기능을 제공받을 수 있다. 이는 사용자가 이전에 방문 했던 기록을 쿠키 파일에 저장하고, 재방문할 경우 쿠키파일의 내용을 기반으로 하여 ID와 PW를 입력하는 과정 없이 로그인된 하면을 제공하거나, 이전에 받았던 서비스의 연속성을 제공받을 수 있기 때문이다.

3. 쿠키파일의 취약성

Cookie 시스템은 매우 단순한 알고리즘을 통해 생성된다. 사용자가 웹사이트에서 입력한 아이디와 패스워드 혹은 웹 사이트를 종료하기 전에 마지막으로 방문하고 있던 페이지의 내용을 사용자 컴퓨터에 텍스트 파일로 저장하여 편리함을 제공하지만, 생성된 쿠키파일은 보안에 매우 취약하다.

3.1 단순한 쿠키 파일 열람 절차

현재 서비스되고 있는 많은 웹 페이지들은 HTML만으로 서비스하기 어려운 기능을 JavaScript를 이용하여 구현하여 서비스하고, 쿠키파일을 생성, 관리, 활용하고 있다.

이처럼 웹에서 많이 사용되고 있는 JavaScript를 이용하면 손쉽게 클라이언트에 저장되어 있는 쿠키 파일 정보를 확인 할 수 있다.

다음 JavaScript를 URL 입력창에 입력하면 클라이언트에 저장되어 있는 쿠키파일의 정보를 확인 할 수 있다.

그림 1은 위의 JavaScript를 이용하여 현재 서비스 중인 웹 사이트에서 작성된 쿠키파일의 내용을 확인결과이다. 이와 유사한 JavaScript로 팝업창을 통해서 해당 사이트의 쿠키파일의 내부 정보를 확인 할 수 있는 JavaScript가 존재한다.

팝업창을 통해서 쿠키파일의 내부 정보를 획득할 수 있는 Script의 경우에는 팝업창을 보여 주는 경로를 지정

표 1. Cookie 파일을 확인하는 JavaScript 1

javascript:document.cookie

그림 1. 쿠키 파일 내부 정보

표 2. 쿠키 파일을 확인하는 JavaScript 2

javascript:alert(document.cookie)

표 3. 쿠키 파일을 전송하기 위한 JavaScript

```
<script language=javascript>
  window.open("공격자의
웹 사이트 주소?cook="+document.cookie);
</script>
```

하여 공격자가 원하는 곳으로 쿠키파일의 정보를 전송 할 수 있어 더욱 큰 보안적 위협 요소가 된다.

그림 1과 그림 2는 위에서 소개한 JavaScript를 이용하여 몇몇 웹 사이트에서는 쿠키파일 내부 정보를 확인해 보았다.

이 사이트들은 쿠키파일의 내부 정보를 보호하기 위해서 암호화를 하였다. 하지만 이러한 경우에도 표본이 될 수 있는 다수의 쿠키 파일을 공격자가 습득하고, 암호화에 사용된 알고리즘을 찾아낸다면 현재의 컴퓨팅 능력으로 암호화되기 이전의 평문내용을 습득할 수 있다.

3.2 쿠키 파일이 저장되는 위치

쿠키 파일이 저장되는 위치는 대부분 사용자가 사용하는 브라우저가 지정하고 있는 기본 경로에 저장 된다.

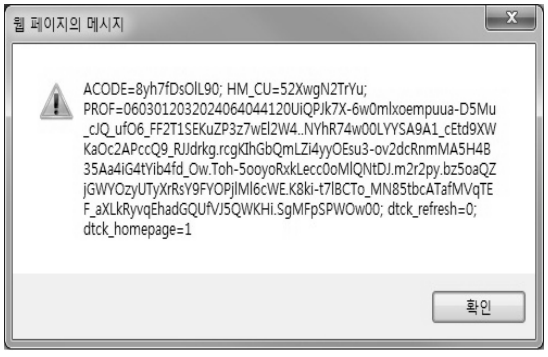

그림 2. 쿠키 파일 내부 정보

① Netscape 사용자의 경우 탐색기에서 “cookies.txt”라는 파일을 찾으면 쿠키가 저장되어 있는 위치를 찾을 수 있다.

② Microsoft internet Explorer에서는 쿠키를 하나의 텍스트 파일로 만들어 저장하고 있으며, Windows 2000의 경우에는 “Documents and Settings\(user 계정)\cookies\”에 저장된다.

사용자의 Cooke파일을 공격자의 서버로 전송하는 악성코드에 사용자의 PC가 감염된 경우를 가정해 보자. 이 악성코드는 각 브라우저가 기본적으로 사용하는 쿠키파일 저장 위치를 확인 하는 것만으로도 감염자 대부분의 쿠키파일을 획득할 수 있을 것이다.

3.3 JavaScript가 포함된 게시글 열람

현재 서비스 되고 있는 많은 웹 페이지들은 사용자가 글쓰기 화면에서 HTML코드를 이용하여 사용자가 글을 작성 할 수 있는 기능을 제공하고 있다[2].

사용자가 원하는 형태의 모든 글을 작성할 수 있도록 웹 사이트에서 제공하기는 어렵다. 또한 HTML코드에 익숙한 사용자는 직접 HTML코드를 이용하여 글을 작성하고 싶어 하는 요구에 충족시켜 주기 위해서 HTML글쓰기 기능을 제공하고 있다[4].

하지만 이러한 HTML글쓰기 기능에서 JavaScript기능 사용제한을 하지 않는 경우에는 아래와 같은 JavaScript를 통해서 글을 읽는 사용자의 쿠키파일 내부의 정보를 공격자가 원하는 곳으로 전송 할 수 있는 위험이 있다[3].

이러한 코드를 삽입한 게시글을 읽을 경우 사용자의 쿠키파일이 공격자에게 전송되는 과정은(그림 4) 같이 도식화 할 수 있다.

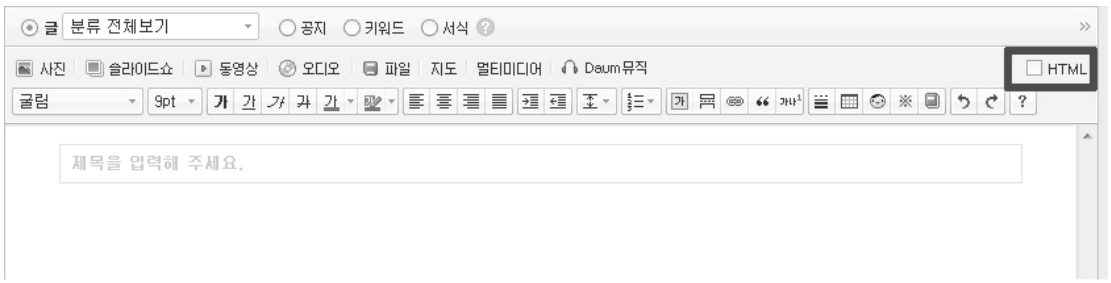

그림 3. HTML을 통한 게시 글 작성

그림 4. 쿠키 파일 노출 결로

3.4 기타 쿠키 파일 취약점

- 쿠키 생성 시 사용되는 데이터 및 알고리즘의 단순성으로 인한 취약성
- 웹 서버와 사용자 간 네트워크(인터넷 등)를 통한 쿠키 정보를 교환 하는 과정에서 발생 가능한 취약성으로 네트워크 스니핑으로 쿠키 정보를 빼낼 수 있다.
- 사용자의 부주의로 로그인 후 자리를 비웠을 때 타인이 저장 장치를 이용하여 쿠키 정보를 복사하여 재상용 가능성[5]
- 신뢰 관계를 맺고 있는 서버 간 세션 관리를 위하여 사용자 정보를 교환할 수 있으며, 이 구간(네트워크)에서 정보를 도청 할 수 있다.
- 사용자 PC에 백도어 등의 악성프로그램이 깔려 있는 경우 쿠키 정보뿐만 아니라 모든 정보를 해커에게 빼앗길 수 있다[6].

4. 현재의 Cookie 파일 보호 방법

Cookie 파일이 사용자의 컴퓨터에 일반적인 텍스트 파일로 작성되어 보안상에 많은 위험이 존재 하고 있다는 것은 앞에서 충분히 설명 하였다고 생각한다. 이처럼 공격에 취약한 Cookie 파일을 보호하기 위해 현재 여러 가지 방법이 제시 되었다[7].

4.1 인증된 사용자만 Cookie에 접근

Cookie파일은 평문 형태로 웹 서비스를 이용하는 클라이언트 측에 저장된다. 쿠키가 사용되어질 때 공격자도 쉽게 쿠키의 내용을 가져올 수 있다. 따라서 클라이언트에 저장된 Cookie파일에 접근을 인증과정을 통해 Cookie를 생성한 사용자나 서비스를 제공해줄 정상적인 웹 사이트만 사용할 수 있도록 한다면 쿠키는 보호될 수 있다.

4.2 Cookie 데이터를 암호화 하여 저장

사용자에게 쉽게 노출 될 수 있는 쿠키 정보를 보호할 수 또 다른 방법은 쿠키정보를 암호화하여 저장하는 것이다. 암호화를 통해 저장한다면 쿠키 정보가 평문이 아니므로 노출되어도 암호화 키를 알아내거나 암호화된 정보를 복호화 못하는 이상 이 쿠키의 정보는 무용지물이다. 하지만 이 방법만으로 사용자에 대한 신뢰를 제공하지 못한다. 또한 쿠키의 정보를 암호화 하게 되면 암·복호화 시간이 추가로 필요하다.

4.3 사용자가 직접 Cookie삭제

용자가 자신의 컴퓨터에 만들어져 있는 쿠키파일을 주기적으로 직접 삭제 하는 방법이다. 이는 쿠키파일을 공격자가 훔쳐가는 것을 차단 할 수 있지만, 다음 삭제까지의 주기가 길어질 경우 그 사이에 축척 된 쿠키 파일에 대해 위험이 존재하고 사용자가 직접 삭제해야 하기 때문에 사용자의 의식에 큰 영향을 받게 된다.

4.4 현재 Cookie 파일 보호 방법의 문제점

앞에서 살펴본 Cookie파일 보호 방법은 현재의 웹 페이지의 구조를 변경하거나 웹 페이지 이용자 스스로 취해야 하는 방법들이라는 점이다.

웹 페이지 이용자의 연령층이 다양하듯이 이용자들의 컴퓨터 활동능력 또한 다양하고, 컴퓨터 활용 능력이 높다고 하더라고 사용자 스스로 보안에 관한 인식을 가져야 한다는 점에서 그 활용도가 높다고 말 할 수 없다[7].

5. 쿠키 파일의 보안성 향상을 위한 보안영역

쿠키 파일은 앞에서 설명한 다양한 위험에 노출 되어 있다. 그 중 쿠키 파일이 웹 브라우저의 기본 저장 위치에서 생성되고 관리됨으로 인해 사용자의 쿠키 파일이 다량으로 유출 되어 개인정보가 침해되는 위험이 크다.

쿠키파일을 생성되고 관리되는 위치를 옮기거나 분산되어 저장되기만 해서 한 번에 모든 쿠키파일이 유출 되는 상황을 막을 수 있을 것이다.

사용자의 웹 사이트 방문 실태를 살펴보면 사용자는 한 번에 3개 이상의 사이트를 방문하는 빈도가 낮다. 즉, 클라이언트에 저장되어 있는 쿠키 파일 중 3개 이상의 파일이 한번에 사용되는 경우는 드물다. 대부분의 웹 사이트에서 쿠키파일을 생성하는데 실제 사용자가 사용할 때에는 모든 쿠키 파일이 웹 브라우저에서 저장한 쿠키 폴더에 존재 하지 않아도 사용자는 웹 사이트를 이용하는데 불편을 느끼지 않는다[6].

5.1 보안영역의 기본 개념

클라이언트에 생성된 쿠키파일을 안전한 보안영역에서 관리를 한다. 만약 사용자가 웹 브라우저를 통해서 웹 사이트 방문을 요청할 경우 웹 브라우저는 보안영역에서 관리되고 있는 쿠키파일 중 사용자가 요청한 웹 사이트에 대한 쿠키 파일이 존재 하는지 여부를 먼저 확인하다.

만약 해당 웹 사이트에 대한 쿠키파일이 존재할 경우 보안영역에서 웹브라우저가 사용하는 쿠키 폴더로 쿠키 파일을 이동 시키고 웹 사이트에 페이지를 요청 한다.

웹 브라우저는 웹 페이지 정보를 받아 오고 옮겨온 사용자의 쿠키파일을 활용해 사용자에게 최적화 된 웹 페이지를 사용자에게 제공한다.

이러한 과정을 도식화 하면(그림 5)와 같다.

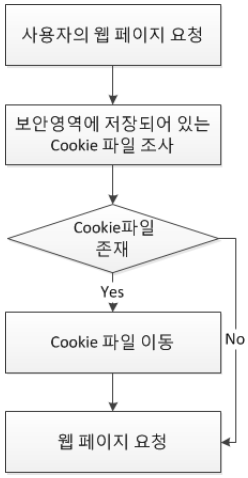

그림 5. 보안영역을 통한 쿠키 파일 관리 절차

5.2 보안영역의 기능

보안영역은 각 인터넷브라우저에 따라서 서로 다른 장소에 저장되는 쿠키 파일을 하나의 장소에서 안전하게 보관하기 위한 다음과 같은 기본적인 기능이 있어야 한다.

보안영역은 쿠키파일을 안전하게 보관 및 관리하기 위한 목적으로 설계되었기 때문에 보안영역에 대한 접근 권한이 엄격히 이루어져야 한다. 또한 공격자가 사용자PC에 설치한 악성코드나 백도어 등의 프로그램으로부터 보안영역 접근권한에 대한 API Hooking이 불가능해야 할 것이다.

또한 웹 브라우저의 상태를 지속적으로 확인해 현재 방문 중이지 않은 웹페이지에서 작성한 쿠키파일이 쿠키 디렉토리에 존재하지 않도록 관리된다.

6. DEVS 모델링

4장에서 설명하고 있는 새로운 쿠키 파일 저장 시스템에 대해서 설명하고자 한다.

현재의 쿠키 파일 시스템 보다 안전한 관리를 위한 시스템의 모델링 구조는(그림 6)과 같다. Cookie Storage System은 Input System과 Cookie Transfer System으로 구성된다.

Input System 사용자가 웹 브라우저 주소 입력창에서 입력하는 주소, 사용자가 웹 페이지 문서 내에 존재하는 페이지 이동 태그를 클릭했을 때에 웹 브라우저 변화를 관리하기 위한 시스템이다. EF모델을 통해서 사용자가 직접 입력하는 URL혹은 HTTP문서 내의 페이지 이동 태그를 클릭했을 때, URL값을 입력 받기 위해서 사용된다. Input System은 EF모델을 통해서 받아온 URL값을 관리하기 위해서 사용되는 모델이다.

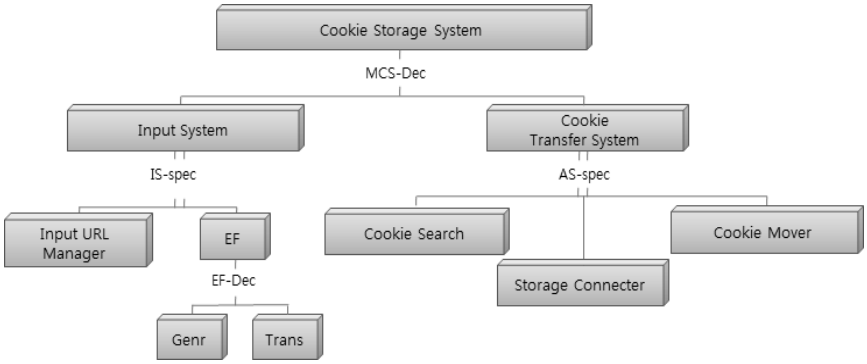

그림 6. 쿠키 파일 저장 시스템 모델링 구조

Cookie Transfer System은 Cookie 파일 이동에 관여하는 시스템이다. 내부에는 Cookie Search, Storage Connecter, Cookie Mover로 구성된다. Storage Connecter은 쿠키 파일이 관리되고 있는 보안영역에 접근을 담당한다. Cookie Search는 보안영역 내에 존재하고 있는 쿠키 파일을 검색하는 일을 수행한다. Cookie Mover는 쿠키 파일을 보안영역으로부터 웹 브라우저가 지정한 쿠키폴더로의 이동, 쿠키폴더로부터 보안영역으로 쿠키파일 이동을 관리하게 된다.

6.1 Input System

Input System은 사용자가 입력하는 URL을 관리하기 위해서 사용되는 시스템이다. 시스템 내부에는 Input URL Manager와 EF모듈이 존재한다.

6.1.1 Input URL Manager

사용자가 입력하는 URL, 웹 페이지 내에 존재하는 페이지 태그 클릭 시에 Input URL Manager에서 URL을 관리한다.

URL Parser은 사용자가 요청하는 페이지의 URL을 활용하게 된다. URL 중에서 쿠키파일을 검색하기 적당한 문자열로 URL을 분리하게 된다.

클라이언트에서 생성되는 쿠키 파일은 “Cookie:User Name@도메인명/”로 구성이 되는데 사용자가 요청하는 페이지의 URL은 웹 서비스 하위 디렉토리나 파일을 요청하는 경우가 빈번이 발생함으로 요청하는 URL전체를 활용해서 쿠키파일을 검색할 경우 그 효율적인 면에서 성능이 떨어지기 때문에 URL을 재구성 한다.

Page Request Module은 사용자가 방문하기를 원하는 웹 페이지를 웹 서버에 요청하기 위한 모듈이다.

6.2 Cookie Transfer System

Cookie Transfer System에는 Cookie Search와 Storage Connecter, Cookie Transfer가 존재한다.

Storage Connecter은 쿠키파일이 저장되어 있는 보안영역에 접근하기 위해 사용된다. 보안영역은 단순한 디렉토리로 구성할 수도 있지만, 디렉토리로 보안영역을 구성할 경우 큰 보안성 향상을 기대 할 수 없다. 따라서 새로운 보안영역을 구성하야 하게 되는데, 이 보안영역에 대한 접근을 제한해야 하고 이러한 보안영역에 접근하기 위해서 Storage Connecter을 사용하게 된다.

6.2.1 Cookie Search

Cookie Search는 보안영역에서 관리되고 있는 쿠키파일 중에서 현재 필요로 하는 쿠키파일이 존재 하는지 여부를 확인하기 위해서 사용된다.

보안영역에서 관리 되고 있는 쿠키파일을 검색하기 위해서 보안영역의 쿠키파일의 파일명과 URL Parser로부터 생성된 URL을 활용하게 된다.

String Compare Module은 File Name와 URL 두 String 값을 이용하여 쿠키파일을 검색하는 모듈이다.

만약 보안영역에 필요로 하는 쿠키파일이 존재 하지 않을 경우에는 쿠키 파일이 없다는 것을 웹 브라우저에 알려 새로운 쿠키파일을 생성하도록 한다.

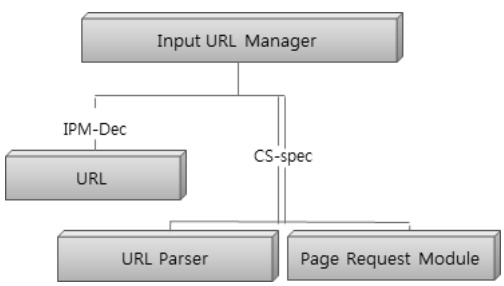

그림 7. Input URL Manager

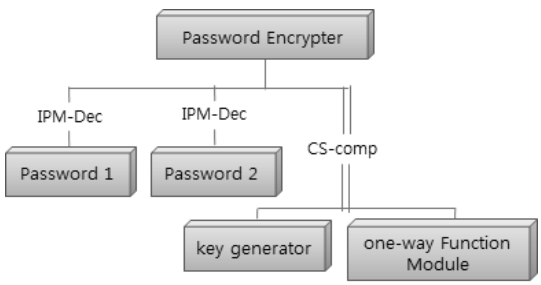

그림 8. Cookie Search

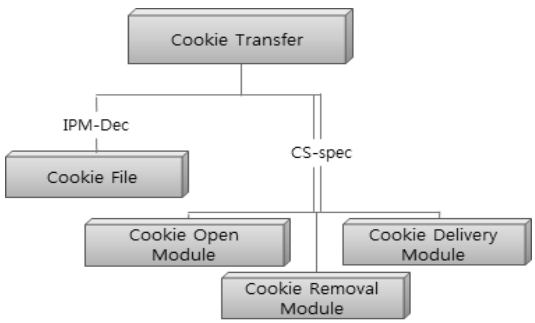

그림 9. Cookie Transfer System

6.2.2 Cookie Transfer

Cookie Transfer은 보안영역에서 저장되어 있는 쿠키파일을 웹 브라우저에서 설정 되어 있는 쿠키디렉토리로 옮기는 역할 수행하는 모듈이다.

Cookie Open Module은 쿠키파일의 내용을 열기 위해서 사용하는 모듈로 쿠키파일 내용을 웹 브라우저가 지정한 쿠키파일 디렉토리로 옮기기 위한 준비 작업을 수행하게 된다.

Cookie Delivery Module은 보안영역에 저장되어 있는 쿠키파일을 웹 브라우저에 설정되어 있는 쿠키디렉토리로 옮기는 실질적인 역할을 수행하는 모듈이다.

또한, 사용자가 웹 페이지 방문을 끝냈을 때 쿠키디렉토리에서 보안영역으로 다시 옮기는 역할 수행하게 된다. 이때 Cookie Open Module을 한번 더 사용하여 쿠키파일의 내용이 변경된 경우 최신의 쿠키파일을 보안영역에 저장하게 된다.

Cookie Removal Module는 사용자가 웹 페이지 방문을 마쳤을 때, 웹 브라우저에 설정되어 있는 쿠키디렉토리에서 쿠키파일을 삭제해 악성코드로부터 쿠키디렉토리의 내용이 유출되는 것을 방지하게 된다.

6.3 보안영역 적용 시 예측

위에서 설계한 보안영역은 사용자의 웹페이지 이동 요청을 웹브라우저가 수행하기 이전에 이루어진다. 사용자가 요청한 웹페이지의 주소는 Input URL Manager를 통해서 어떠한 도메인의 웹 페이지 인지를 확인하고 이를 Cookie Transfer로 도메인값을 전달하게 된다.

Cookie Transfer은 Cookie Delivery Module을 통해서 웹브라우저에 사용하는 쿠키디렉토리 파일로 옮겨진다.

웹브라우저를 통해 로드 된 웹페이지는 Cooker Delivery Module를 통해서 옮겨진 쿠키파일의 정보를 이용하여 사용자에게 적합한 웹페이지 결과를 제공한다.

사용자의 웹페이지 방문이 끝났을 때, Cookie Transfer의 Cooker Open Module은 보안영역에 저장된 쿠키파일과 쿠키디렉토리에 저장된 쿠키파일을 서로 비교하여 내용이 변경된 경우 두 쿠키파일 중 최신 쿠키 정보가 저장되 쿠키일을 선택해 Cooker Delivery Module에게 알려주고, Cooker Delivery Module은 이 정보를 바탕으로 쿠키파일을 다시 보안영역으로 가지고 오게 된다.

Cooker Removal Module은 Cookie Delivery Module의 작업이 끝나면 쿠키디렉토리에 있는 쿠키파일을 삭제하여 악성코드로부터 쿠키파일이 유출되는 것을 방지한다[9].

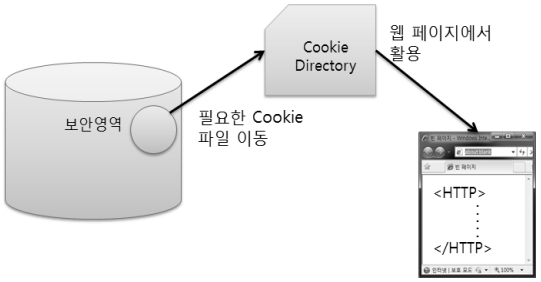

그림 10. 보안영역을 활용한 쿠키파일 사용 예

7. 기대 효과

현재의 쿠키 파일 관리 방법에 비해서 보안영역을 통해서 관리 할 경우 공격자가 획득할 수 있는 사용자의 쿠키파일의 수가 크게 줄어들게 된다. 하지만 보안영역을 활용한 쿠키파일 저장방식은 현재 웹 페이지에서 활용하기 위해 쿠키 디렉토리로 이동한 쿠키파일의 보안까지 책임질 수 없다는 단점이 존재한다[9].

하지만(그림 10)에서 볼 수 있듯이 사용자의 중요한 개인정보가 담겨져 있는 쿠키파일을 보안영역에서 저장해 현재 시점에서 필요한 쿠키파일만 쿠키 디렉토리로 이동시켜 활용할 경우에는 악성코드로 인한 쿠키 디렉토리에 저장되어 있는 쿠키파일의 보안적 취약점을 크게 감소시킬 수 있기 때문에 보안 영역이 필요하다.

8. 결 론

쿠키파일은 사용자의 개인정보와 관련된 매우 중요한 정보를 포함하고 있다. 그러나 쿠키파일의 보안성 향상을 위해 쿠키파일의 내용의 암호화 등에 대한 연구가 주로 진행되어 왔다. 하지만 쿠키 파일의 위치가 노출됨에 따라 발생하는 문제점 또한 중요하고 연구되어야 하는 부분이다.

본 논문에서는 악성코드 등으로부터 쿠키 파일이 저장되는 디렉토리에 대한 보안성 향상을 위해 보안영역 활용 방안을 제안하고, 보안영역을 활용한 쿠키파일 저장 방식에 대한 모델링을 제시하였다.

보안영역을 활용한 쿠키파일 저장 방식은 클라이언트나 웹 브라우저에서 구현 되어지는 기술이기 때문에 웹 서버 관리자는 보안기능을 활용하기 위해서 웹 페이지 코드를 변경 할 필요가 없다. 본 논문에서 제시된 쿠키파일 저장 방식은 웹 페이지 코드의 변경이 없이도 기존의 쿠

키기능을 활용 하면서 쿠키정보를 보호할 수 있기 때문에 효율적인 쿠키 보호 방법으로 이용될 수 있다.

추후 지속적인 연구와 쿠키저장방식의 구현과 다른 시각에서의 쿠키 파일을 안전성을 향상시킬 수 있는 연구를 통해 지금 보다 쿠키 파일의 안전성을 향상 시킬 수 있는 연구가 지속되어야 할 것이다.

참 고 문 헌

1. 최향창, 최은복, 노봉남, “쿠키 보호 시스템 설계”, 정보보학회 봄 학술발표논문집, 29(1), pp. 904-906, 2002년 4월.
2. 홍봉화, 정윤돈, 김은원, “DHTML 편집기를 이용한 블로그 사이트에서 쿠키보안에 관한 연구”, 전자공학회 논문지, 42(2), pp. 28-36, 2005년 6월.
3. http://seobangnim.com/zbxe/4030
4. 안철수 연구소 전문가 칼럼 “쿠키의 취약성 및 요구되는 보안성” 2005.
5. 임대호, “악성코드에 의한 HTTP Cookie 유출 문제점 및 대책”, CERTCC-KR 권고문, 2000년 1월.
6. 양종필, 이경헌, “인증서 기반의 개선된 보안 쿠키의 설계와 구현”, 한국통신학회논문지, vol. 27, no. 11C, 2002년 11월.
7. 서진원, 서희석, 곽진, “웹서비스 공격정보 분류 방법 연구”, 한국시뮬레이션학회 눈문지, 19(3), pp. 99-108, 2010년 4월.
8. 서희석, 최중섭, 조필환, “운도우 악성코드 분류 방법론의 설계”, 정보보호학회논문지, 19(2), pp. 83-92, 2009년 4월.
9. 김태경, 서희석, 이동영, “분산 네트워크에서 안전한 전송을 위한 알고리즘에 관한 연구”, 한국시뮬레이션학회 논문지, 18(1), pp. 35-40, 2009년 3월.

심 원 태 (wtsim@kisa.or.kr)

1986 서울대학교 계산통계학과 졸업(학사)
1988 KAIST 전산학과 졸업(석사)
1987～2000 (주)데이콤 부장
2000～2003 (주)인젠 연구소장
2003～현재 한국인터넷진흥원 팀장, 침해사고대응단장

관심분야 : 시스템과 네트워크 보안, 클라우드 보안

최 요 한 (ahluiyoo@kut.ac.kr)

2008～현재 한국기술교육대학교 컴퓨터공학부 컴퓨터공학과

관심분야 : 악성코드, 네트워크, 모바일 보안

서 희 석 (histone@kut.ac.kr)

2000 성균관대학교 산업공학과 학사
2002 성균관대학교대학원 전기전자 및 컴퓨터공학과 석사
2005 성균관대학교대학원 전기전자 및 컴퓨터공학과 박사
2005～현재 한국기술교육대학교 컴퓨터공학부 조교수

관심분야 : 모델링&시뮬레이션, 네트워크보안, 보안 시뮬레이션, USN

노 봉 남 (bbong@jnu.ac.kr)

1978 전남대학교 수학교육과 졸업(학사)
1982 KAIST 대학원 전산학과 졸업(석사)
1994 전북대학교 대학원 전산과 졸업(박사)
1983～현재 전남대학교 전자컴퓨터공학부 교수
2000～현재 전남대학교 시스템보안연구센터 소장

관심분야 : 디지털 포렌식, 시스템과 네트워크 보안