사례연구를 통한 Cross Site Script 공격 및 방어대책

손강원*, 김지훈*, 조두산**, 윤종희*
*영남대학교 컴퓨터공학과,**순천대학교 전자공학과
e-mail : kwkw91@yu.ac.kr, f13521@naver.com, dscho@sunchon.ac.kr, youn@yu.ac.kr

A Case Study of Cross Site Script attack and defence measures

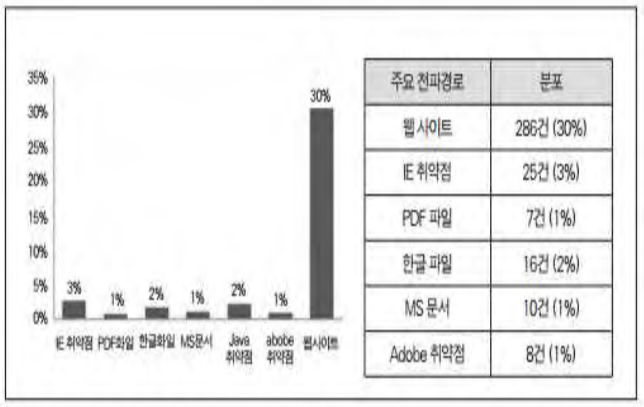
Kang-Won Son*, Ji-Hun Kim*, Doosan Cho**, JongHee Youn*
*Dept of Computer Engineering, YeungNam University
**Department of Electronics Engineering, Sunchon National University

요 약

최근 인터넷 기술 발전에 따라 웹 시장이 커지고 웹 사이트 범죄가 급증하고 있다. 통계에 따르면 악성코드 주요 전파경로로 웹 사이트가 가장 높은 비율을 차지 할 정도로 조심해야 할 전파경로 중 하나이다. 그 중에서 이 논문은 웹 사이트 관련 공격 기술 중 하나인 크로스 사이트 스트립트(Cross Site Script, XSS)취약점을 알아본다. 먼저 XSS에 대해 알아보고, 실제 공격 예시를 살펴본다. 그리고 XSS를 막기위한 방어대책으로 화이트 리스트와 필터링에 대해 알아본다.

1. 서론

최근 인터넷 기술이 발전되고 웹 시장이 커짐에 따라 그에 따른 웹 사이트 범죄가 증가하고 있다. 웹 사이트 피해의 사례로는 2010년 트위터가 Cross Site Script(XSS) 공격을 당해 음란물 유포 사이트로 변한 사례가 있고[1], 2012년에는 XSS 취약점을 바탕으로 야후 계정에 접속해 메시지를 읽거나 전송할 수 있었던 사례[2] 등 현재까지 많은 수의 XSS 공격이 이루어지고 있다.



주요 전파경로	분포
웹 사이트	286건 (30%)
IE 취약점	25건 (3%)
PDF 파일	7건 (1%)
한글 파일	16건 (2%)
MS 문서	10건 (1%)
Adobe 취약점	8건 (1%)

그림 1 악성코드 주요 전파경로별 분류

그림 1은 2012년 1월에서 2013년 3월까지 한국인터넷진흥원에서 조사한 악성코드 주요 전파경로별 분류[3]이다. 이 결과를 분석해보면 웹 사이트로 인한 전파경로가 286건으로 30%, IE 취약점으로 인한 전파경로가 25건으로 3%, 한글 파일과 MS문서에 의한 취약점이 각각 16건과 10건으로 2%와 1%이다. 이 분석결과로 웹 사이트의 취약점을 통해 악성코드 배포가 가장 많은 비중을 차지하고 있고, 또 인터넷 이용자들이 가장 조심해야할 전파경로임을 알 수 있다. 그 중에서 이 논문에서는 웹 기반 취약점 중 하나인 크로스 사이트 스크립트(Cross Site Script)의 취약점에 대해 알아볼 것이다. 2장에서는 XSS와 XSS 실제 공격내용, 3장에서는 그에 따른 방어대책, 마지막 4장은 결론으로 마무리 지을 것이다.

2. XSS 공격

2.1 XSS

XSS[4]는 영문명칭 cross-site scripting의 약어로 웹 사이트의 관리자가 아닌 사용자가 웹 페이지에 악성 스크립트를 심어 이를 악용하는 취약점이다. 사용 예로는 게시판을 통해 악성 스크립트를 심거나, 메일에 악성코드를 심어 보내 수신자의 정보를 탈취하는 것을 예로 들 수 있다. 이 취약점으로 다른 이의 정보를 탈취할 수 있고, 사이트의 비정상적인 실행을 야기할 수 있다.

그림 2 XSS 공격방식

그림 2는 XSS의 공격방식이다. 해커는 사용자에게 자신의 얻고자하는 정보를 얻기 위해 취약점이 있는 웹에 악성코드(악성 URL)를 심어놓고 사용자의 클릭을 유도한다. 사용자가 악성코드가 들어있는 글을 클릭하게 되면 악성코드가 사용자의 PC에서 실행되고, 그 결과로 해커는 사용자의 정보를 얻게 된다. 현재 세계 시스템 공격 방법 통계에서 XSS는 SQL 인젝션, OS 명령어 인젝션, 오버플로우에 이어 4위에 해당하는 높은 위험도[5]를 가지고 있다.

2.2 XSS 공격 실습

최근 한 A 사이트 자유게시판에 간단한 코드의 XSS공격을 시도했다. 이 코드는 <IMG SRC=/ onerror=window.open("http://해커IP/attack.php?data="+document.cookie)></img>이다. 이 코드는 이미지파일이 오류가 났을 경우 스크립트가 실행되는 코드다. 해커서버에 attack.php라는 이름의 php파일을 열어주고 그 php에 victim의 쿠키값을 넘겨주는 스크립트이다. attack.php의 핵심 코드는 $cookie=$_GET['data']; 이다. 이 코드는 스크립트에서 data로 넘겨준 쿠키값을 cookie라는 변수에 저장시켜주는 역할을 한다. 위의 스크립트 코드를 게시판 글 등록과 함께 게시했고, victim이 그 글을 읽었다. 그 결과로 victim의 쿠키 정보를 얻을 수 있었다. 이렇게 획득한 쿠키 정보를 탈취해 쿠키값 변조 프로그램을 사용해 해당 교육기관 사이트에 세션 하이재킹해 로그인을 할 수 있었고, 로그인 이후 학생정보 조회 및 수정이 가능하고, 더 나아가 그 학생의 학적상태를 변경하는 등 모든 기능들을 사용할 수 있게 되었다. 그리고 또 다른 악성코드를 통해 같은 A기관의 웹 메일에 악성스크립트를 심은 메일을 보냈다. 그 결과로 그 메일을 읽은 회원의 쿠키정보를 탈취할 수 있었다. 이렇게 간단한 코드를 사용함에도 A교육기관 사이트에 XSS공격이 가능 한 것은 XSS에 대한 보안이 제대로 세워지지 않았기 때문이다.

3. XSS 방어대책

3.1 화이트 리스트

화이트 리스트는 블랙 리스트의 반대말로, 허용되는 코드만 지정해주고 나머지는 모두 제한하는 것이다. 화이트 리스트는 지정해 주지 않으면 모두 제한되기 때문에 강력한 보안성을 가지고 있다. 화이트 리스트는 많은 내용을 담지 않는 웹에서 유용하다. 사용할 코드만 화이트 리스트에 등록해놓으면 되기 때문에 많은 기능을 사용 할 수 없는 단점이 있지만 보안적인 측면에서는 강력한 힘을 발휘할 수 있다.

3.2 필터링

XSS 방어대책 중 또 다른 방법은 필터링이다. 필터링은 스크립트를 사용하는데 있어 스크립트를 모두 사용하지 못하게 되면 안 되기 때문에 그 대신에 사용하는 방법이다. 2.2의 스크립트의 필터링을 위한 방법으로는 REPLACE함수가 있다. REPLACE함수는 지정한 문자 수에 따라 텍스트 문자열의 일부를 다른 텍스트 문자열로 바꿔주는 함수로, 함수의 구문은 REPLACE(old_text, start_num, num_chars, new_text)이다. old_text는 바꿀 문자열, start_num은 old_text의 문자위치, num_chars는 바꿀 문자 수, new_text는 old_text를 바꿀 텍스트를 각각 의미한다. 2.2의 스크립트는 onerror문구를 통해 실행된다. 그래서 REPLACE('onerror', 1, 0, “”); 하면 스크립트문구 중 onerror는 공백으로 처리되어 이 코드의 실행을 막을 수 있다. 대부분의 대형포탈에서는 방어대책으로 필터링을 사용하는 만큼 잘 활용하면 많은 XSS공격들을 막아 낼 수 있다. 하지만 XSS공격에 완벽한 방어방법은 없기 때문에 뚫리게 되면 보안 패치를 수시로 업데이트 해주어야 한다.

4. 결론

현재 많은 사이트들은 웹 공격에 취약하다. 이 논문에서 공격을 시도한 A교육기관의 예에서도 자유게시판이 취약하다면 같은 웹의 다른 게시판이나 텍스트박스 또한 취약할 수 있다. 보안전문가들이 웹 공격에 대비하기 위해 많은 수의 방어기법을 사용하고 있다. 이 논문에서 소개한 방어기법 외에도 많은 기법들이 존재한다. XSS 공격은 사실상 완벽히 방어하는 것은 무리가 따른다. 스크립트를 사용하지 않게 하는 것이 가장 좋은 방법이지만 그렇게 되면 웹 사이트의 훌륭한 기능들을 다 사용할 수 없게 된다. 현재 보안은 가장 중요한 분야로 떠오르고 있다. 많은 연구와 기법 개발을 통해 XSS 공격 방어에 힘 써야 한다.

참고문헌

[1] 트위터 XSS공격, http://www.zdnet.co.kr/news/news_view.asp?artice_id=20100922074501&type=det

[2] 야후 계정 탈취, http://www.boannews.com/media/view.asp?idx=33862

[3] KISA 수집 PC악성코드분석동향, http://www.kisa.or.kr/uploadfile/201306/2013062114540030 67.pdf

[4] XSS, http://ko.wikipedia.org/wiki/사이트_간_스크립팅

[5] SANS/CWE 가장 위험한 25大 소프트웨어 오류, The Web Application Security Consortium