취약점 분석을 통한 Web Site 해킹 연구

송진영* · 박대우*

*호서대학교 벤처전문대학원 IT응용기술학과

A Study of Web Site Hacking Through Vulnerability Analysis

Jin-young Song* · Dea-Woo Park*

*Dept. of IT Application Technology, Hoseo Graduate School of Venture

E-mail : *jedisong@naver.com · *prof1@paran.com

요 약

웹사이트를 통한 해커들의 악의적인 웹 취약점 공격으로 개인정보와 개인자산이 유출되고 있다. 일부국가에서는 해커부대가 운용되어, 타 국가의 웹사이트를 통해 기밀정보 및 개인정보를 불법적으로 접근하여 자료를 획득하고 있다. 국내 웹사이트들은 프로그램뿐만 아니라 웹사이트 관리의 문제로 인해 많은 취약점을 갖고 있다. 본 논문에서는 국내뿐만 아니라 전 세계적으로 유행하는 XSS, SQL Injection, Web Shell 공격에 대한 취약점을 분석하고, XSS, SQL Injection, Web Shell 공격을 직접 공격한다. 공격 후에 해킹을 시연한 자료를 수집, 분석을 하여 보안 대응책을 제시한다. 본 연구는 웹사이트 보안과 안전한 웹사이트 관리를 향상 시킬 수 있는 기술연구에 이바지 할 것이다.

ABSTRACT

Personal information being leaked, and personal assets that through a malicious web site for hackers to exploit. Other confidential information via the web site of the country, and your personal information by illegally accessing the data has been obtained who Hacker forces are operating in some countries. Due to the problem of web site management has many vulnerabilities that web sites, as well as programs. In this paper, in the trend world, as well as domestic XSS, SQL Injection, Web Shell analysis of the vulnerability to attacks and XSS, SQL Injection, Web Shell is a direct attack to attack. Security measures are presented what after the attack demonstrated the hack to data collection, analysis. In this study, web site management, web site security and safety can be improved and research will contribute.

키워드

Web Site Security, Hacking, Vulnerability, XSS, SQL Injection, Web Sell

Ⅰ. 서 론

금융거래 정보가 온라인으로 처리되고 있는 상황에서 XSS(Cross Site Script)[1], SQL Injection[2], Web Shell[3] 공격에 대한 취약점은 웹사이트 공격으로 이어진다. 또한 스팸 메일과 웹사이트가 결합된 보안 위협 등 웹사이트 공격의 기술들이 계속 발전하고 있다.

공격자는 최초 이메일로 악의적인 웹사이트 주소를 보내 접속을 유도한다. 사용자가 접속하면 웹브라우저 취약점 공격, 악성코드 설치, 취약점을 가진 문서 파일(PDF, Office 파일 등) 다운로드 등의 공격을 한다. SQL Injection 공격으로 대량의 데이터베이스가 유출되고 있고, 웹사이트 연결된 서버의 상당수가 취약점이 노출되어있는 형국이다.

본 논문에서는 최근 해킹동향과 많은 문제점이 되고 있는 XSS와 SQL Injection 및 Web Shell 공격에 대해서 알아보고 실제로 웹사이트를 공격하고 나서 골격 분석과 취약점 분석 및 취약점을 막는 보안대책에 대해서 연구한다.

Ⅱ. 관련 연구

2.1. 최근 해킹 동향

국제 웹 애플리케이션 보안 연구단체인 OWASP[4]는 2010년 4월 19일 주요 웹 취약점 항목인 OWASP TOP 10 2010 버전을 발표하였다.

OWASP TOP 10은 웹 애플리케이션 개발자와 테스트 담당자, 보안 담당자, 감사자들에게 웹 보안에 관한 우선 실무 가이드라인을 제시하고 있다. 표 1에서 기존 OWASP TOP 10 2007년과 2010년 모두 1위의 취약점으로 Injection 공격에 대한 취약점을 나타내고 있다. Injection 공격에는 SQL Injection과 Web Shell 공격이 포함된다.

표 1. 2010년과 2007년 OWASP TOP 10 비교

2010년 TOP 10	2007년 TOP 10
A1. Injection	A2. Injection Flaws
A2. XSS	A1. XSS

2.2. XSS

크로스 사이트 스크립팅이라고 불리는 XSS 취약점은 WEB어플리케이션에서 발견되는 어플리케이션(HTTP) 관련 취약점으로 방화벽이나 IDS[], 바이러스백신 등과 같은 기존의 보안대책들이 XSS 취약점에 대해서는 거의 감지하지 못한다[5].

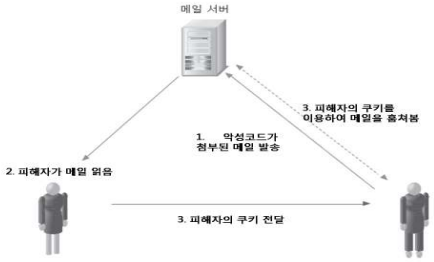

그림 1. XSS를 이용한 공격의 기본원리

그림 1과 같이 XSS의 기본 공격원리는 공격자는 피해자에게 이메일에 평범한 내용과 악성 스크립트 코드를 숨겨서 보낸다. 피해자는 공격자의 메일을 읽는 순간 공격자가 숨겨서 보낸 악성 스크립트 코드는 피해자의 쿠키 정보를 공격자에게 넘긴다. 공격자는 피해자의 쿠키 값을 간단히 조작하여 피해자의 메일을 읽는다[6].

2.3. Web Shell

Web Shell(Web Shell)이란 공격자가 원격에서 대상 웹서버에 명령을 수행할 수 있도록 작성한 웹 스크립트(asp, jsp, php, cgi) 파일이다. 이때 zip, jpg, doc와 같은 데이터 파일종류 이외에 악의적으로 제작된 스크립트 파일인 Web Shell을 업로드하여 웹 서버를 해킹하는 사고가 빈번히 발생하고 있다. 최근에는 파일 업로드뿐만 아니라 SQL Injection과 같은 웹 취약점을 공격한 후 지속적으로 피해시스템을 관리할 목적으로 Web Shell을 생성 한다[7][8].

2.4. SQL Injection

SQL Injection은 정상적인 SQL 질의문을 변조하여 불법 로그인, DB 데이터 열람, 시스템 명령 실행 등을 수행하는 공격이다. 이 공격은 사용자 입력 값 또는 URL 파라미터 값에 대한 적절한 검증작업이 이루어지지 않아 발생된다[9].

일반적으로 데이터베이스에 접근하는 웹 어플리케이션에서 SQL query문에 대한 문자열을 필터링 하지 않고 바로 데이터베이스로 넘기는 것이 가능한 점을 이용한 exploit 방법의 일종이다[10].

Ⅲ. WebSite 취약점 분석

3.1. XSS 취약점 분석

XSS는 사용자의 입력내용을 포함하는 HTTP요청에 대해 동적으로 HTML을 생성하는 어플리케이션(CGI)이 공격대상이 된다.

그림 2. XSS 테스트

그림 2에서 공격할 웹페이지에 XSS 취약점이 있는지 <SCRIPT>alert("hello");</SCRIPT>이 추가된 문자열을 공격할 웹서버에 업로드 시키면 임의의 script를 삽입시킬 수 있는 문제가 있기 때문에 이를 XSS라고 한다.

일반 게시판에 스크립트를 포함시키는 공격과 URL창에 스크립트를 입력한 방법, 이미지파일 등에 악성스크립트를 포함시켜 공격하는 방법 등에 대한 취약점이 있다.

3.2. SQL Injection 취약점 분석

SQL Injection 버그를 가지고 있을 경우에 시스템 레벨의 권한을 획득하는 것이 가능하기도 하고, 데이터베이스 내의 테이블을 조작하거나 내용을 유출 시킬 수 있는 가능성도 존재한다.

SQL Injection은 현재 인터넷 상에서 구동되는 많은 웹 어플리케이션에서 발견할 수 있으며 특히 MS SQL을 데이터베이스로 사용하였을 경우에 데이터베이스 사용자의 권한이 허락한다면 시스템 명령을 바로 실행 시킬 수 있다.

데이터 자체가 중요한 경우 여러 가지 기법을 사용하여 데이터베이스 테이블의 구조를 알아 낼 수도 있고, 데이터베이스의 테이블 자체를 통째로 복사해 올 수도 있다. 또한 프로그래머가 의도적으로 SQL Injection에 대비하지 않는 이상은 막기가 힘들고, 또한 모든 SQL query에 대해서 SQL

Injection을 100% 없앤다는 것 또한 힘들다.

SQL Injection공격은 주로 data, system configuration에 접근하고 조작하고, ackage procedures나 3GL language extensions 등을 사용하여 OS 레벨에 접근이 가능한 취약점이 있다.

3.3. Web Shell 취약점 분석

표 2와 같이 Web Shell 프로그래밍 언어의 함수 취약점에서 나타내고 있다.

표 2. 프로그래밍 언어 취약점

프로그래밍언어	함수	
PHP	require()	include()
	eval()	preg_replace()
	exec()	passthru()
	system()	popen()
Shell Script	모두 실행 가능	
Perl	open()	sysopen()
	glob()	system()
	'' (backticks)	eval()
Java	system.*	
C, C++	system()	exec**()
Python	exec()	eval()
	execfile()	compile()
	input()	

공격자는 Web Shell을 대상 서버에 업로드한 후 웹을 이용하여 시스템 명령어를 수행한다. Web Shell은 웹페이지 소스코드 열람, 악성스크립트인 iframe 등을 삽입, 파일 업로드, 서버 및 데이터베이스 자료 유출 등의 다양한 공격이 가능하다.

Ⅳ. 취약점 분석을 이용한 WebSite 해킹

4.1. 해킹 공격 툴

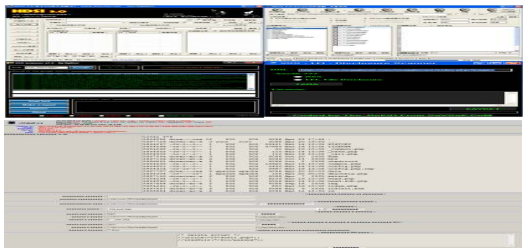

그림 3. SQL Injection/ XSS/ Web Shell Tools

분석된 취약점을 이용한 Web Site 해킹을 위하여 그림 3처럼 XSS툴과 SQL Injection 툴, 웹 쉘 툴을 이용하였다. 중국에서 개발된 SQL Injection 툴은 자동화 도구로 DB획득, 웹 쉘 삽입, 시스템 명령수행, 코드 삽입 등의 기능을 가지고 있다. XSS툴은 자동화 툴은 없지만 웹사이트 취약점을 찾아야 하기 때문에 여러 가지 스캐닝 툴들이 있다.

4.2. XSS 공격

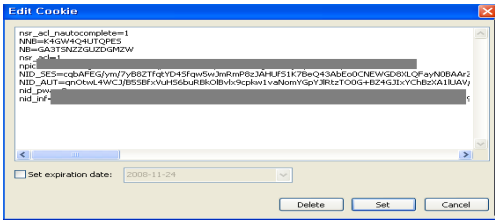

그림 4. XSS를 이용해서 Cookie값 추출

그림 4는 XSS를 이용해서 상대방 쿠키 값을 얻는 그림이다. 해당사이트는 XSS 취약점에 어느 정도 보안을 했지만 Flash 파일에 악성스크립트를 심어서 우회공격을 하였다.

그림 5는 XSS를 이용해서 공격한 장면이다. 왼쪽은 공격자의 정보이고 오른쪽은 공격당한 상대방의 정보이다. XSS를 이용해서 쿠키 값을 얻고 얻은 쿠키 값으로 이용해서 오른쪽 그림과 같이 상대방 정보를 이용해 들어갈 수 가 있다.

그림 5. XSS 공격장면

4.3. SQL Injection 공격

SQL Injection 공격은 파라메타에 특수문자를 삽입하여 500 error를 발생시킨다. response 결과를 가지고 500 error에 대한 Redirection이 설정되어 있는지, 필터링이 되었는지 확인해 볼 수 있다. 주요 Target은 URL창, 페이지나 우편번호의 검색 부분 등 사용자가 입력이 가능한 부분이다. 이 부분은 검색어에 대한 모든 결과를 뿌려주기 때문에 만약 이 곳에서 SQL Injection 공격이 가능하면 원하는 정보를 별도의 추가 작업 없이 받아볼 수 있었다.

그림 6. SQL Injection 공격 장면

그림 6은 SQL Injection공격을 하여 웹사이트

의 관리자로 접속한 그림이다. 먼저 취약점을 찾기 위해 login, password 창에 싱글 쿼트(')가 들어간 문구를 입력한다. 웹서버에서는 select id from member where id='' or '1'='1'-- and pwd = '1'='1'-- 로 명령어를 실행하고 그에 맞는 값을 화면에 보여준다. 보통 자신의 아이디와 패스워드를 입력하지만 '1'='1'-- 를 입력해 내부적으로 문구를 맞는 것으로 처리하여 웹사이트에 관리자로 접속할 수 있다.

4.4. Web Shell 공격

Web Shell 공격 중 한 기법인 RFI(Remote File Include) 공격을 한다. php를 돌리는 서버에서 특수 shell 파일을 실행하도록 함으로서 해당 서버의 원격 실행권한을 획득하는 공격을 하였다.

Web Shell공격은 exec() 같은 함수를 이용하여 시스템 명령어 및 외부 프로그램을 실행할 수 있었다. 그리고 파일을 업로드, 다운로드 및 파일을 조작이 가능하다. 데이터베이스를 열람하거나 조작, 레지스트리 조작을 할 수 있고, 리버스 텔넷을 이용하여 웹서버를 원격으로 제어가 가능하였다.

4.5. 해킹공격에 대한 보안 정책제시

SQL Injection에 대한 보안정책을 제시한다.

■ 웹사이트에 입력이 가능한 모든 부분에 대한 예외처리를 해야 한다.
■ 사용자 입력 값에 체크 루틴을 필수로 한다.
■ 웹서버의 오류메시지를 특정 페이지로 Redirection 처리 한다.
■ 최소화된 권한을 가진 해당 DB에만 권한이 있는 계정을 이용 한다.
■ 웹사이트 요청 필터를 통해서 해킹시도 문자열을 사전에 차단
■ 관련문서를 탐독 및 적용한다.

XSS에 대한 보안정책을 제시한다.

■ 게시판이나 메일 기타 등등에서 HTML을 허용을 금한다.
■ HTML을 허용해야 할 경우에는 꼭 필요한 태그만을 허용
■ 태그의 속성, 이벤트를 꼭 확인하여 그에 대해 적절히 대응
■ IP와 Session을 하나로 묶어 A라는 Session은 B라는 IP에서만 사용가능하게 한다.
■ 주기적으로 취약성 체크 툴을 사용한다.

Web Shell에 대한 보안정책을 제시한다.

■ 운영체제의 모든 명령어들은 웹서버를 통해 실행하지 못하도록 한다.
■ /tmp, /var/tmp 디렉토리에서 웹서버가 스크립트 및 실행파일을 실행하지 못하도록 한다.
■ 웹서버의 업로드 경로에서 실행권한을 사용하지 못하도록 한다.

대부분 보안 대책은 웹사이트 관리자와 개발자가 유의해야 할 사항이다. XSS 같은 경우 일반 사용자가 스팸메일이나 확인되지 않는 정보를 읽어서 피해를 당하는 경우도 있지만, 보안 관리자가 입력 값을 받을 때에는 부적절한 값이나 파일 등을 철저히 필터링 해줘야하고 항상 로그확인이나 경고 이벤트에 주위를 기울여야 한다.

Ⅴ. 결 론

본 논문에서는 서비스되고 있는 웹사이트의 공격의 대표적인 XSS, SQL Injection, 그리고 Web Shell에 대한 취약점 분석과 취약점을 이용하여 실제로 공격을 하여보고 보안 대책을 제시하였다.

해커들의 공격기법들은 변형되어가고 두 가지 이상의 공격기법이 결합된 공격을 하지만 관리자들은 아직 보안에 미숙한 모습을 보이고 있다. 해커들은 서버의 취약점을 이용해 공격을 하기 때문에 위 보안대책은 어디까지나 웹 개발자나 웹 서버관리자를 위한 대책이다.

향후 연구에서는 Blind 나 Mass SQL Injection 등을 연구하고 Injection과 XSS의 결합된 공격기법연구가 필요하다.

참고문헌

[1] K. K. Mookhey, Nilesh Burghate, “Detection of SQL Injection and Cross-site Scripting Attacks”, http://www.security_focus.com/infocus/1768

[2] SQL Injection, http://www.spidynamics.com/papers/SQLInjectionWhitePaper.pdf

[3] 이인용, 조재익, 조규형, 문종섭, "SQL 질의 애트리뷰트 값 제거 방법을 이용한 효과적인 SQL Injection 공격 탐지 방법 연구", 정보보호학회논문지, 제 18권 제 5호, pp.135-147, 2008. 10.

[4] OWASP Top 10 - 2010 Release, 2010.

[5] 김영민, 안준선, "웹 응용 프로그램의 보안 취약성 분석", 프로그래밍언어논문지, 제 21권 제 1호, pp.39-48, 2007. 4.

[6] Scott, D., Sharp, R. “Developing Secure Web Applications.” IEEE nternet Computing, 제 6권 제 6호, pp.38-45, 2002. 11.

[7] 한국인터넷진흥원 해킹대응팀, "휘슬 (WHISTL) 관련FAQ", http://www.krert .or.kr/whistl/Whistl_FAQ.pdf

[8] 황중연, "웹 서버 구축 보안점검 가이드", 한국정보보호진흥원, pp.6-81, 2007. 9.

[9] 장승주, 최은석, "웹 환경을 이용한 보안 취약점 점검 도구 개발에 관한 연구", 한국정보과학회, 제 34권 제 2호, 2007. 10.

[10] 보안관리팀, "공개용 보안프로그램을 활용한 취약성 점검", 한국정보보호진흥원, pp14-24.