웹 프로그램의 취약점 검색을 위한 소스분석 툴 구현

김성욱*, 황태문**, 김판규*, 박상수*, 이종혁*

*경성대학교 컴퓨터공학과, **대진정보통신고등학교 컴퓨터정보과

Implementation of Source Analysis Tool for Vulnerability Search of Web Program

Soung-uk Kim* · Tae-moon Hoang**· Pan-kyu Kim*· Sang-su Park*· Jong-hyeok Lee*

*Kyungsung University Computer Engineering

E-mail : jhlee@star.ks.ac.kr

요 약

웹용 프로그래밍언어인 PHP, JSP, ASP 등은 기존의 HTML문과 결합하여 좀더 사용자 인터렉티브한 웹 페이지를 가능하게 하였다. 그러나 이런 언어와 프로그램 등 인터넷의 보급과 함께 엄청난 속도로 발전하여 프로그래밍 언어 자체 또한 보안에 취약한 상태로 발전하게 되었고, 이는 그 언어와 연동 가능한 많은 수의 서버 해당 시스템이 외부로 노출되는 빌미를 제공하게 되었다. 본 연구에서는 실제로 웹 프로그램 및 SQL에 대한 해킹 가능성 코드를 분석하는 프로그램을 구현하고 인터넷 해킹에 웹 프로그램의 보안에 대한 중요성을 알리며 해킹방지를 위하여 서버 시스템의 패치 못지않게 웹 프로그램 보안이 효율적인가를 보이고자 한다.

ABSTRACT

The Program language for web, such as PHP, JSP, ASP and so on, make it possible to offer more user interactive web page by using with HTML. These language and program have been developed with great speed, but security part could not catch up with this development. As a result, it has brought a problem which is expose many server systems to the outside. In this research, we implement Web and SQL analysis program which can analysis hacking causing factor. With this analysis program, we will show you how much efficient it has compared with security patch for server system.

키워드

웹 보안, 웹 프로그램, 해킹 방지, 취약점 검색

Ⅰ. 서 론

인터넷 초기의 대부분의 웹 페이지들은 단순한 HTML 코딩만으로 제작이 가능했다. 그러나 인터넷이 보급되고 확산되면서 사이트의 형태는 진화하기 시작하였다. 복잡한 상호 작용 과정인 사용자가 요구하는 정보의 추출 및 전달, 게시판, 회원 가입, 설문, 대화실 등을 구현하기 위해서 웹 프로그래밍이 도입되었다. 웹 프로그래밍이 도입된 페이지에서는 정보 등을 좀더 효과적으로 사용자에게 전달하고, 사용자의 관점에서 보다 친숙하고 조직적으로 보이는 다양한 내용을 사용자에게 제공할 수 있다.

그러나 악의적인 공격자들은 웹 서비스 제공자의 의도와는 상관없이 웹 프로그램에 약간의 보안상 허점이 발견되면 그 허점을 정보 유출의 통로로 이용하기 위하여 쉴 새 없이 공격한다. 웹 프로그램의 서버측 코드 개발자는 의도적으로 허점을 만든 것이 아니고 웹 프로그램의 안정적인 동작에만 집중하므로 이러한 허점이 야기되는 문제에 대해서는 깊이 고려하지 않는 경향이 다분하다는 것이다. [1]-[3]

본 논문은 웹 페이지 또는 웹 프로그램의 보안 취약성을 동적으로 분석하여 침입 가능성을 미연에 방지하는 방법에 관한 연구이다.

본 논문의 분석 대상이 되는 웹 프로그래밍언어는 PHP를 언어이다. PHP언어를 분석 대상 언어로 선택한 이유는 PHP 언어의 호환성과 보편성에 있다. PHP는 현재 가장 많이 사용되는 웹 서비스 어플리케이션인 Apache와 호환성이 뛰어나며 Windows, Linux, Unix 등의 OS에서 두루 사용하고 있는 가장 많이 사용되는 웹용 프로그래밍 언어이다. 또한 ORACLE, MS SQL 2000등 상업용 데이터베이스 및 Mysql 등 무료로 제공되는 데이터베이스와 연동이 가능하다.

이러한 특징으로 인하여 가장 많이 사용되고 있다.

그리고 SQL코드의 해킹 가능성을 알기 위해서 선택한 데이터베이스 또한 PHP와 일반적으로 가장 많이 사용하는 Mysql이다. 실제 검사할 SQL문은 데이터베이스의 특성에 맞게 재구성된 sql이 아닌 Standard sql만을 검사하도록 한다.

본 논문에서 제안하는 해킹 방지를 위한 웹 페이지 보안성 검사프로그램의 실효성 검증을 위하여 실제로 웹 프로그램 및 SQL에 대한 해킹 가능성 코드를 분석하는 프로그램을 구현하고 인터넷 해킹에 웹 프로그램의 보안에 대한 중요성을 알리며 해킹방지를 위하여 서버 시스템의 패치 못지않게 웹 프로그램 보안이 효율적인가를 보이고자 한다.

II. 기존의 연구

2.1 웹서비스

웹 서비스는 회사나 기업의 정보 전달이나 홍보의 목적 외에도, 근래에는 전자상거래나 기업의 마케팅, 개인을 위한 정보 전달을 위해 그 사용도가 높아지고 있다. 또한 국가차원에서 이루어지는 전자 정부 웹 서비스의 중요도는 이루 말할 필요가 없을 것이다. 이러한 웹 서비스를 좀 더 안전하게 만들고자 하는 노력의 필요성도 상대적으로 커지고 있다. 실제 예로 미국의 Computer Security Institute (www.gocsi.com)에서 FBI와 함께 조사하여 발표한 "2002 CSI/FBI Computer Crime and Security Survey" 를 보면 1999년부터 2002년까지 4년 동안 웹 사이트에 대한 질문을 하였는데, 응답자들의 97%가 웹 서비스를 제공하기 위한 사이트를 가지고 있고, 그 중 43.2%가 전자상거래 사이트였다고 한다. 아래의 그림 1을 보면, 지난 12개월동안 웹 사이트에 대하여 허가되지 않는 접근이나 웹 사이트의 오용 때문에 손해를 입었다고 응답한 웹 사이트가 무려 38%임을 알 수 있다. 반면에 실제 이런 공격들이 자신의 웹 사이트에서 발생하는지 여부조차 모른다고 응답한 웹 사이트도 무려 21%에나 이르고 있다.[1]

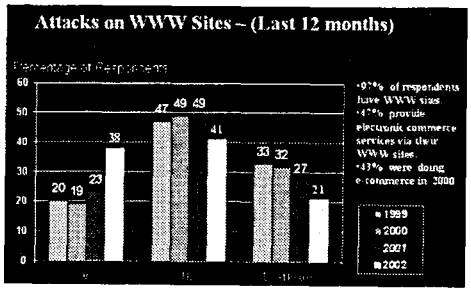

그림 1. 웹사이트 공격통계

2.2. 웹 공격유형분석

1) cross site scripting취약점

Cross-site Scriptiong(이하 XSS) 취약점은 웹어플리케이션이 사용자의 입력으로 받은 악성코드를 필터링하지 않고 그대로 동적으로 생성된 웹 페이지에 포함하여 사용자에게 재전송하는 것이다.[4]

2) sql injection 취약점

최근 웹 프로그램은 자료의 효유적인 저장 및 검색을 위해 PHP, JSP, ASP등의 스크립트언어와 DBMS를 연동하여 사용하는데, SQL Injection은 공격자가 백엔드의 연동된 DBMS를 열람, 수정, 삭제하도록 SQL질의를 수정하여 실행하는 것이다.[3]

3) 버퍼오버플로어취약점

버퍼오버플로우 취약점은 일반적인 응용프로그램 보안 취약점의 하나로써 대표적인 웹 어플리케이션의 취약점에 포함된다. 프로그램ㅇ에서 버퍼의 한계를 점검하지 않고 작성된 코드부분을 이용하여 악의적인 공격자 코드로 리턴 어드레스 등을 덮어 쓰도록 해서 프로그램의 정상적인 동작을 변경하거나 프로그램이 다운되도록 하는 공격방법이다.

4) 파일업로드 취약점

파일업로드 취약점은 공격자가 게시판의 파일첨부를 이용해서 웹 서버의 사용자 쉘을 획득할 수 있는 취약점이다.

2.3 침입탐지시스템

침입탐지란 컴퓨터가 사용하는 자원의 integrity, confidentiality, availability을 저해하는 일련의 행위들의 집합 또는 컴퓨터 시스템의 Security Policy를 파괴하는 행위를 말한다. 이러한 침입탐지를 전문적으로 하기위한 시스템을 침입탐지시스템이라고 하며 이러한 침입탐지 시스템은 컴퓨터 시스템의 비정상적인 사용, 오용, 남용 등을 규정하는 시스템으로 가능한 한 실시간으로 처리하는 시스템, 또는 침입을 시도하거나, 침입행위가 일어나고 있거나, 침입이 발생한 것을 확인하는 절차이다. 침입 탐지 시스템의 본래의 의미는 침입을 탐지 하는 것이고 탐지된 내용을 바탕으로 능동적인 대처까지 해주는 의미도 내포하고 있다.

III. 시스템 설계 및 구현

3.1 시스템 개발 환경

표본 웹 사이트와 테스팅 어플리케이션 시스템으로 구성되고 개발 환경은 다음과 같다.

OS : Windows XP sp1

DB : Mysql 4.0.23

표본 웹 사이트의 웹서버 및 웹서버 모듈

: apache1.3.3, PHP4.3.10, PHPZend2.5.7
테스팅 어플리케이션
: Windows XP sp1, VC++6 MFC.

3.2 시스템 구성도

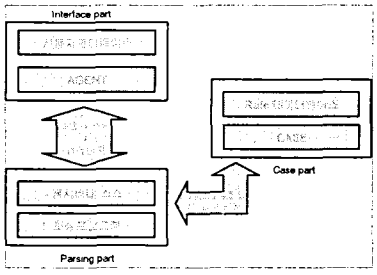

그림 2. 시스템 구성도

본 시스템은 다음과 같은 모듈로 구성되어 있다.
각 모듈은 그림 2에서처럼 사용자 인터페이스에서 사용자로부터 요청을 수용하는 인터페이스 부분과 사용자 요청에 의해서 디렉토리 검색 후 각 디렉토리마다 있는 소스 파일 내 PHP소스를 추출하는 파서 부분 그리고 RuleDB내의 룰을 검색해서 각 룰을 생성하는 CASE생성기, 이런 일련의 작업을 총괄하는 에이전트부분으로 구성된다.

3.3 구현
3.3.1 사용자인터페이스부
인터페이스부는 MFC 기본 함수의 precreatewindow(), 각종 window printing메소드로 기본 인터페이스창을 생성하고 확인, 닫기, 취소 버튼, 찾아보기 버튼 등은 onok(), onclose(), ondir() 등 메소드가 매핑되어 사용자 입력 등을 처리한다. 그림3은 확인 버튼을 클릭하면 호출되는 메소드로

```
void CAgentView::OnOK()
{
m_html=((CButton*)GetDlgItem (IDC_HTML))->
GetCheck(); //html 포함
m_php = ((CButton *)GetDlgItem(IDC_PHP))->
GetCheck(); // php만
((CComboBox*)GetDlgItem(IDC_PATH))->GetWindo
wText(m_path);
m_sub = ((CButton *)GetDlgItem(IDC_SUBDIR))
->GetCheck(); // 서브디렉토리 포함
Parsing(); //파서 함수 호출
pMainFrm->go_Tray(); //실행시키고 트레이로
들어간다.
}
```

그림 3. 확인 버튼 클릭시 실행되는 프로그램

검색할 디렉토리의 path와 검색할 파일 확장자, 서브디렉토리 검색 등에 대한 설정을 멤버 변수에 할당하고 파서 메소드를 호출하여 실제 프로그램을 수행한다. 모든 프로그램이 수행된 후 트레이로 들어가게 된다.

그림 4, 5는 프로그램이 실행전 화면과 수행중인 화면으로 현재 검사 중인 폴더와 파일명을 보여주고 있다. 인터페이스부가 종료되면 destroy되는 것과 윈도우 트레이로 들어가는 두 가지 형식으로 구분되어 질 수 있다.

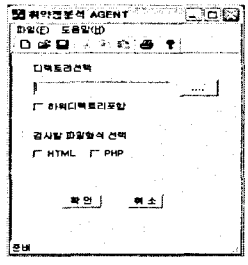

그림 4. 첫 실행화면

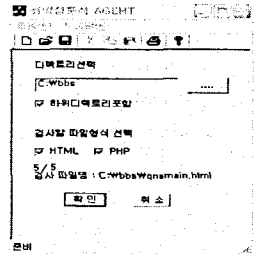

그림 5. 프로그램 수행중

3.3.2 파서부
파서부는 인터페이스부에서 ondir()메소드에 의해 선택된 디렉토리의 파일과 서브 디렉토리를 찾아서 사용자가 선택한 파일 형태(html,php)에의해 해당 파일을 열고 소스코드를 분석하는 역할을 한다.
파싱 메소드는 ondir()에 의해서 호출되며 소스파일을 한라인씩 읽어 들여 해당 라인에 PHP 소스가 있는지 검사 후 있다면 취약점검사를 수행하도록 한다.

3.3.3 CASE생성기
CASE 생성기는 파서부에 의해 호출되며 사용자 입력에 의해 선택된 디렉토리나 파일 또는 관리자가 테스트를 위해 선택한 파일내의 소스와 rule을 비교하여 취약점 존재 여부를 판별한다.
agent테이블 스키마의 ptype은 입력된 값의 타입으로 파라미터와 function으로 구분하여 입력하고 cmd는 취약점을 가지는 이름이다.
데이터 베이스에 접속하여 테이블에서 rule을 추출하여 소스 프로그램과 비교 분석하는 메소드는 check_grama() 메소드다.

IV. 구현 결과

본 프로그램은 시스템 함수를 사용한 취약점과 파라미터 취약점을 분석하여 취약점 가이드를 사용자에게 제공한다.
본 프로그램은 기존 IDS가 웹 서비스 자체프로그램의 취약점에는 둔감한 부분을 웹 프로그램 소스를 직접 분석하여 침입이 가능한 컨텐츠를 찾아내는 것이다.
VC++6.0에서 구현하였고 현재는 윈도우즈 시스템에서만 실행결과를 얻을 수 있다.

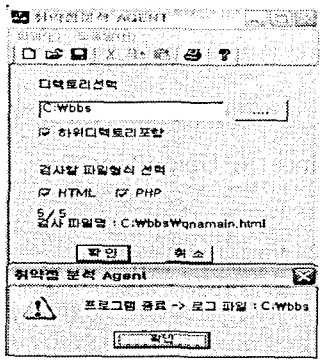

그림 6. 수행완료

수행결과 그림6과 같으며, 현재 개인적으로 사용하는 개인용 방화벽 윈도우즈용과 비교하여 웹 서버에 임의의 파라미터를 가지고 외부에서 직접 접근하였을 경우 해당 방화벽이 얼만큼 접근을 차단하는지에 대한 테스트와 비교하였다. 본 논문에서 제안한 형식과 비슷한 형식으로 나와 있는 프로그램을 아직은 찾지 못해 정확한 성능평가는 얻을 수 없으나 현재 정보보호협회 등에서 내놓은 해킹방지에 대한 지침서나 php.com에서 내놓은 취약점에 대해서는 룰만 적용시킨다면 모든 웹상의 해킹은 막을 수 있다고 사료된다.

위 프로그램의 수행 결과는 해당 웹 프로그램이 있는 루트 디렉토리에 result.log파일로 만들어서 제공하며, 그림 7과 같다.

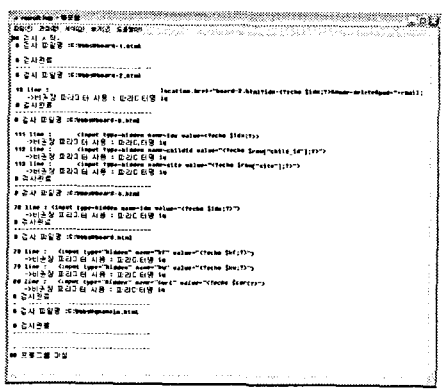

그림 7. 수행결과 로그 파일

표 1은 IDC역할을 하는 개인용 방화벽과 본 프로그램의 수행 능력을 비교한 것을 표로 만들었다.

표 1. 취약점 탐지비교

실험취약점	기존의 IDS	연구의 결과
로그인시 인증회피 (sql injection)	탐지못함	탐지가능
외부에서 서버내 파일 접근(system)	탐지못함	탐지가능
파라미터조작 (parameter)	탐지못함	탐지가능
buffer overflow 공격등	탐지가능	탐지못함

V. 결 론

본 논문은 기존의 IDS가 침입을 탐지해낼 수 있는 부분은 많으나 실제로 웹 서비스에서는 제대로 동작을 하지 않고 특히 웹 프로그램의 취약점에 대해서는 전혀 탐지할 수 없다는 점에서 시작하였다.

IDS처럼 웹서버의 앞단에서 서비스요청에 대한 탐지보다는 소스를 직접 분석함으로써 네트워크에 부하를 주지 않으면서 직관적으로 침입 자체를 막을 수 있었다.

실험대상으로 잡은 소스는 이전에 개발한 적이 있는 업체의 게시판과 로그인 관련 소스를 선택하였고 개인용 방화벽과 본 논문에서 제안한 프로그램으로 비교하여 다음의 결과를 얻었다.

구현결과 기존의 IDC에서는 웹용 프로그램의 취약점에 대해서는 침입탐지 제체가 불가능하게 나온 것을 알 수 있다. 그러나 본 실험에서 작성된 프로그램은 실제 웹 서버에서 수행되는 방식이 아니므로 오진가능성이 높다는 것 또한 문제점이 된다. 그러나 웹 서버 테스팅의 경우 각 컨텐츠별로 세션, 쿠키등으로 인해 실제 테스팅이 이루어지기는 어렵긴 하나 웹 테스트작업을 같이 수행되는 방식이 정확도를 좀더 높일 수 있을 것이다.

참고문헌

[1] 이재동 등, "인터넷 기술과 응용, 사이텍미디어", 2001년 09월

[2] 김성렬 등, "소스코드를 이용한 웹 응용 취약점 분석에 관한 연구", 정보보호학회 학술대회논문집, Vol.13, No2

[3] 김수용 등, "매개 취약점 점검 언어로부터 점검 코드를 자동으로 생성하는 에이전트를 이용한 취약점 관리시스템", 정보보호학회 논문집, Vol.11, No.1

[4] 강동호 등, "악성코드 사이트 탐지 분석기 구현", 정보보호학회지 논문집 Vol12, No.1