오픈소스기반의 웹서비스 취약점 진단도구에 관한 분석

유정석*, 홍지훈**, 정준권**, 정태명***
*성균관학교 컴퓨터공학과
**성균관대학교 전자전기컴퓨터공학
***성균관대학교 정보통신대학
e-mail:nogury88@skku.edu, {jhhong88,jkjung}@imtl.skku.ac.kr, tmchung@ece.skku.edu

A Study on Analysis of Open Source Analysis Tools in Web Service

Jeong-Seok Yoo*, Ji-Hoon Hong**, Jun-Kwon Jung**, Tai-Myoung Chung***
*Department of Computer Engineering, Sungkyunkwan University
**Department of Electrical and Computer Engineering, Sungkyunkwan University
***College of Information and Communication Engineering, Sungkyunkwan University

요 약

최근 인터넷이 발전함에 따라 월드와이드웹(World Wide Web) 기반의 웹 서비스가 급격한 발전을 이루었다. 또한 이 웹 서비스를 바탕으로 다양한 컨텐츠들과 이를 이용하는 사용자의 수도 함께 증가하였다. 그러나 이와 같은 웹 서비스의 보편화가 증대될수록 이를 악용하려는 사이버 범죄 또한 비례하여 증가하고 있다. 최근에는 공격자들이 스마트폰을 대상으로 악성코드를 전파하기 위한 방법으로 웹 서비스를 활용하기 시작하면서 웹 서비스의 보안에 대한 중요성이 더욱 강조되고 있다. 이러한 웹 서비스 보안의 필요성을 인지하고, 많은 사람들이 무료로 쉽게 웹 서비스 보안취약점을 진단 할 수 있도록 여러 오픈소스 기반의 보안 취약점 진단도구가 연구, 개발되고 있다. 하지만 웹 서비스의 보안약점을 진단하는 도구의 적합성 평가 및 기능 분류가 명확하지 않아서 진단도구를 선택하고 활용함에 있어 어려움이 따른다. 본 논문에서는 OWASP에서 위험도에 따라 선정한 웹 서비스의 보안 취약점 Top 10 항목과 소프트웨어 보안약점 진단가이드 등을 통해 웹 서비스 보안 취약점을 진단하는 도구에 대한 분석 기준을 제시한다. 이후 오픈소스로 공개된 테스트 기반 취약점 탐지도구와 소스 기반 취약점 진단도구들에 대해 제시한 기준을 이용하여 분석한다. 본 논문의 분석결과로 웹 서비스의 안전성을 평가하기 위해 활용할 수 있는 진단 도구에 대한 분석정보를 제공함으로써 보다 안전한 웹 서비스의 개발과 운영에 기여할 것으로 기대한다.

1. 서론

최근 인터넷이 발전함에 따라 월드와이드웹(World Wide Web) 기반의 웹 서비스가 급격한 발전을 이루었다. 또한 이 웹 서비스를 바탕으로 다양한 컨텐츠들과 이를 이용하는 사용자의 수도 함께 증가하였다.

그러나 이와 같은 웹 서비스의 보편화가 증대될수록 이를 악용하려는 사이버 범죄 또한 비례하여 증가하고 있다. 최근에는 공격자들이 스마트폰을 대상으로 악성코드를 전파하기 위한 방법으로 웹 서비스를 활용하기 시작하면서 웹 서비스의 보안에 대한 중요성이 더욱 강조되고 있다.

이러한 웹 서비스 보안의 필요성을 인지하고, 많은 사람들이 무료로 쉽게 웹 서비스 보안취약점을 진단 할 수 있도록 여러 오픈소스 기반의 보안 취약점 진단도구가 연구, 개발되고 있다. 하지만 웹 서비스의 보안약점을 진단하는 도구의 적합성 평가 및 기능 분류가 명확하지 않아서 진단도구를 선택하고 활용함에 있어 어려움이 따른다.

본 논문에서는 OWASP(Open Web Application Security Project)에서 위험도에 따라 선정한 웹 서비스의 보안 취약점 Top 10 항목과 소프트웨어 보안약점 진단가이드 등을 분석기준으로 활용하여 취약점 진단에 적합한지에 대한 탐지능력을 조사한다[1].

분석 대상은 크게 1)테스트 기반 취약점 탐지도구와 2) 소스 기반 취약점 진단도구로 분류 할 수 있다. 테스트 기반 취약점 탐지도구는 동작중인 웹 서비스를 대상으로 동적테스트를 수행하여 취약점을 점검하는 도구이며, 소스 기반 취약점 진단도구는 SW실행 없이 소스코드를 대상으로 분석하여 보안 취약점을 점검하는 도구이다.

본 논문은 웹 서비스를 개발할 때에 활용할 수 있는 오픈소스기반의 진단도구에 대한 분석 정보를 제공함으로써 보다 안전한 웹 서비스의 개발과 운영에 기여할 것으로 기대한다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 서비스의 취약점 진단도구를 평가하기 위한 기준을 설명하고, 3장에서는 본 연구에서 분석하고자 하는 오픈소스기반의 취약점 진단도구들을 살펴본다. 4장에서는 각 진단도구에 대해 취약점 점검 능력을 분석하며, 마지막으로 5장에서 결론을 맺는다.

2. 진단도구 분석 기준

안전한 웹 서비스의 제공을 위해서는 진단도구를 활용하여 선제적으로 예방을 해야 한다. 따라서 진단도구들은 알려진 취약점에 대해 적합한 진단기능을 제공하는지 점검이 필요하다. 이를 위해 OWASP에서 발표한 웹 서비스 취약점 10가지에 따라 도구들의 적합성을 분석한다.

이 기준은 테스트 기반 취약점 탐지도구가 적합한 기능을 제공하는지에 대해 평가 할 수 있었으나, 소스 기반 취약점 진단도구에 대해서는 적절한 기준이 되지 못했다. 따라서 소스 기반 취약점 진단도구에 대한 분석은 행정안전부에서 발간한 소프트웨어 개발보안 가이드와 소프트웨어 보안약점 진단가이드에 따라 기준을 제시하고 분석한다[2,3].

2.1 테스트 기반 취약점 탐지도구에 대한 분석 기준

안전한 웹 서비스 보안을 위하여 OWASP에서 발표한 웹 서비스 취약점 10가지는 다음과 같다.

<표 1> OWASP Top 10 - 2013

A1 : 인젝션 취약점
A2 : 인증 및 세션 관리 취약점
A3 : 크로스사이트 스크립팅(XSS)
A4 : 취약한 직접 객체 참조
A5 : 보안 설정 오류
A6 : 민감 데이터 노출
A7 : 기능 수준의 접근통제 누락
A8 : 크로스 사이트 요청 변조(CSRF)
A9 : 알려진 취약점이 있는 컴포넌트 사용
A10 : 검증되지 않은 리다이렉트 및 포워드

1) A1 - 인젝션 취약점

신뢰할 수 없는 데이터가 명령어 혹은 질의문의 일부분으로써 웹 애플리케이션의 인터프리터로 보내질 때 발생한다. 공격자가 악의적으로 삽입한 데이터로 인해 의도하지 않은 명령어의 실행, 혹은 데이터의 변경이 일어날 수 있다.

2) A2 - 인증 및 세션 관리 취약점

인증 및 세션 관리와 관련된 기능이 정확하게 구현되어 있지 않을 때 발생한다. 공격자는 패스워드 또는 세션토큰을 공격하여 다른 사용자로 가장할 수 있다.

3) A3 - 크로스사이트 스크립팅(XSS)

웹 서비스가 암호화나 검증 절차 없이 사용자가 제공하는 신뢰할 수 없는 데이터를 가져온 후, 제한 없이 웹 브라우저로 보낼 때 발생한다. 공격자는 피해자의 브라우저에 스크립트를 실행하여 사용자 세션 탈취, 웹 사이트 변조 등을 할 수 있다.

4) A4 - 취약한 직접 객체 참조

개발자가 파일, 디렉토리, 데이터베이스 키와 같이 내부 구현객체를 참조하는 대상의 정보를 노출시킬 때 발생한다. 공격자는 노출된 참조를 조작하여 허가 받지 않은 데이터에 접근할 수 있다.

5) A5 - 보안 설정 오류

프레임워크, 웹 서버, 데이터베이스 서버 및 플랫폼에 적절한 보안설정이 정의되거나 적용되지 않을 때 발생한다. 기본으로 제공되는 값을 그대로 이용할 경우 안전하지 않을 수 있다.

6) A6 - 민감 데이터 노출

웹 서비스에서 신용카드, 개인 식별 정보와 같은 중요한 데이터에 대해 암호화와 같은 보호조치를 취하지 않을 때 발생한다. 공격자는 약하게 보호된 데이터를 훔치거나 변경해서 신용카드 사기, 신분 도용 등을 할 수 있다.

7) A7 - 기능 수준의 접근통제 누락

웹 서비스의 UI에서 해당 기능이 보이게 하기 전에 기능 수준의 접근권한을 확인하지만 서버에서는 동일한 접근통제 검사를 수행하지 않을 때 발생한다. 공격자는 요청 정보를 위조하여 허가 받지 않은 기능에 접근할 수 있다.

8) A8 - 크로스 사이트 요청 변조(CSRF)

로그온 한 희생자의 브라우저가 사전에 승인된 요청을 취약한 웹 서비스에게 보낼 경우 발생한다. 공격자는 희생자의 브라우저가 공격자에게 이득이 되는 행동을 수행하도록 할 수 있다.

9) A9 - 알려진 취약점이 있는 컴포넌트 사용

알려진 취약점이 존재하는 컴포넌트, 라이브러리, 프레임워크를 사용할 때 발생한다. 공격자는 심각한 데이터 손실을 발생시키거나 공격 가능한 범위를 활성화 시키도록 영향을 미칠 수 있다.

10) A10 - 검증되지 않은 리다이렉트 및 포워드

사용자를 다른 페이지로 리다이렉트 하거나 포워드 할 경우, 적절한 검증 절차가 없이 신뢰할 수 없는 데이터를 사용할 때 발생한다. 공격자는 피해자를 승인되지 않은 페이지에 접근하도록 전달할 수 있다.

2.2 소스 기반 취약점 진단도구에 대한 분석 기준

행정안전부에서 발간한 소프트웨어 개발보안 가이드와 소프트웨어 보안약점 진단가이드를 기반으로, 웹 서비스를 대상으로 한 공격들 중 빈번하게 발생하는 공격에 대한 취약점에 대해 9가지 항목을 도출했다. 본 논문에서는 이 9가지의 항목을 소스 기반 취약점 진단도구를 분석하는 기준으로 삼는다.

<표 2> 소스 기반 진단도구 분석 기준

1. SQL Injections
2. Cross-Site Scripting
3. Command Injection
4. Path Traversal
5. XPath Injection
6. XML Injection
7. LDAP Injection
8. Remote Code Execution
9. HTTP Response Splitting

1) SQL 인젝션

데이터베이스와 연동된 웹 서비스에서 입력된 데이터에 대한 유효성을 검증하지 않을 경우, 공격자가 SQL문을 삽입하여 정보를 열람하거나 조작할 수 있다.

2) Cross-Site Scripting

검증되지 않은 외부 입력이 동적 웹페이지 생성에 사용될 경우, 전송된 웹페이지를 열람하는 접속자의 권한으로 부적절한 스크립트를 수행할 수 있다.

3) Command Injection

검증절차를 거치지 않은 사용자 입력값이 운영체제 명령어의 일부 또는 전부로 구성되어 실행되는 경우, 의도하지 않은 시스템 명령어가 실행될 수 있다.

4) Path Traversal

외부 입력 값을 검증하지 않고 시스템 자원에 대한 식별자로 사용하는 경우, 공격자는 입력 값 조작을 통해 시스템이 보호하는 자원에 임의로 접근할 수 있다.

5) XPath Injection

외부 입력값을 적절한 검사과정 없이 XPath 쿼리문 생성을 위한 문자열로 사용할 경우, 공격자는 쿼리문의 의미를 왜곡시키고 임의의 쿼리를 실행할 수 있다.

6) XML Injection

공격자가 서버로부터 전달되는 XML문을 중간에 가로채서 코드를 수정할 수 있는 보안약점이다.

7) LDAP Injection

외부 입력을 적절한 처리없이 LDAP 쿼리문이나 결과로 사용할 경우, 공격자가 LDAP 쿼리문의 내용을 임의로 변경할 수 있다.

8) Remote Code Execution

원격의 파일을 로컬에 있는 파일처럼 인식하여 실행이 가능할 경우, 악성코드 삽입 및 시스템 내부 명령을 실행할 수 있다.

9) HTTP Response Splitting

HTTP요청에 들어있는 파라미터가 HTTP응답헤더에 포함되어 사용자에게 다시 전달될 때 입력 값에 개행 문자가 존재하면 HTTP응답을 분리하고 악의적인 코드를 주입할 수 있다.

3. 취약점 진단도구 분석 대상

3.1 테스트 기반 취약점 탐지도구

테스트 기반 탐지도구는 동작중인 웹 서비스를 대상으로 동적테스트를 수행하여 취약점을 점검하는 도구이다. 테스트 기반 취약점 탐지도구로 널리 알려진 ZAP, Open VAS, WATOBO에 대해서 조사 분석 한다.

ZAP은 Java를 기반으로 제작되어 크로스 플랫폼 환경에서 모두 사용할 수 있으며 GUI기반의 프로그램 형태로 실행된다[4].

OpenVAS는 Linux환경에서 사용할 수 있으며, 테스트를 수행하는 중앙서버와 GUI기반의 클라이언트 프로그램으로 구성되어 있다[5].

WATOBO는 Ruby를 기반으로 제작되어 크로스 플랫폼 환경에서 모두 사용할 수 있으며 GUI기반의 프로그램 형태로 실행된다[6].

3.2 소스 기반 취약점 진단도구

소스 기반 취약점 진단 도구는 SW실행 없이 소스코드를 대상으로 분석하여 보안 취약점을 점검하는 도구이다. 소스 기반 취약점 진단도구로 널리 알려진 Yasca와 LAPSE+에 대해서 조사 분석 한다.

Yasca는 C/C++, HTML, JavaScript, ASP, ColdFusion, PHP, COBOL, .NET에 대해 진단이 가능하며 CLI환경에서 실행된다[7].

LAPSE+는 JAVA에 대해 진단이 가능하며 이클립스 Plugin으로 실행된다[8].

4. 취약점 진단도구 분석 결과

4.1 테스트 기반 취약점 탐지도구 분석 결과

분석 결과 ZAP의 경우 여섯 가지 항목에 대해 취약점 탐지 기능을 가지고 있었으며, OpenVAS의 경우 세 가지 항목에 대한 탐지 기능을, WATOBO의 경우 두 가지 항목에 대해 탐지 기능을 보유하고 있었다. 분석 대상이 된 취약점 탐지도구 중에서 기준 항목 모두를 탐지하는 기능을 제공하는 도구는 없었으며, 보안 취약점 항목별로 활용할 수 있는 진단도구마다의 특성을 확인할 수 있었다.

<표 3> OWASP Top 10에 따른
테스트 기반 취약점 탐지도구의 분류 목록

Web Application Risk	테스트 기반 취약점 탐지도구
A1 : injection 취약점	ZAP
A2 : 인증 및 세션 관리 취약점	ZAP
A3 : 크로스사이트 스크립팅(XSS)	ZAP
A4 : 취약한 직접 객체 참조	ZAP
A5 : 보안 설정 오류	OpenVAS, WATOBO
A6 : 민감 데이터 노출	WATOBO
A7 : 기능 수준의 접근통제 누락	OpenVAS
A8 : 크로스 사이트 요청 변조(CSRF)	ZAP
A9 : 알려진 취약점이 있는 컴포넌트 사용	OpenVAS
A10 : 검증되지 않은 리다이렉트 및 포워드	ZAP

소스 기반 취약점 진단도구의 경우, 소스 레벨에서 취약점 진단이 가능한 인젝션 취약점, 크로스사이트 스크립팅 항목에 대해 Yasca, LAPSE+ 모두가 진단 기능을 보유하고 있음을 확인할 수 있었다.

두 취약점 항목의 경우, 소스 레벨에서 발생되는 취약점의 종류가 다양하므로 좀 더 세부적인 항목에 대한 분석이 필요하다. 이는 2.2장의 소스 기반 취약점 진단도구에 대한 분석 기준을 기반으로 다음 장에서 조사 분석하였다.

<표 4> OWASP Top 10에 따른
소스 기반 취약점 진단도구의 분류 목록

Web Application Risk	소스 기반 취약점 진단도구
A1 : injection 취약점	Yasca, Lapse+
A2 : 인증 및 세션 관리 취약점	X
A3 : 크로스사이트 스크립팅(XSS)	Yasca, Lapse+
A4 : 취약한 직접 객체 참조	X
A5 : 보안 설정 오류	X
A6 : 민감 데이터 노출	X
A7 : 기능 수준의 접근통제 누락	X
A8 : 크로스 사이트 요청 변조(CSRF)	X
A9 : 알려진 취약점이 있는 컴포넌트 사용	X
A10 : 검증되지 않은 리다이렉트 및 포워드	X

4.3 소스 기반 취약점 진단도구의 기능 분석

각각의 취약점 항목마다 다양한 유형의 소스코드가 존재한다. 그렇기 때문에 한 가지의 경우라도 보안 취약점에 대해 정상적으로 진단하는 경우 해당 기능을 가지고 있다고 판단하고 진단가능(O 표시)으로 표기하였다. 그렇지 않은 경우 진단불가능(X 표시)로 표기하였다.

분석 결과, Yasca의 경우, 다양한 언어를 대상으로 진단기능을 제공하는 장점이 있지만 일부의 취약점 항목에 대해서만 진단할 수 있었다. LAPSE+의 경우, Java 언어만을 대상으로 취약점 진단기능을 제공하지만 선정된 9가지의 항목에 대해서 취약점 모두를 진단 할 수 있음을 알 수 있었다.

<표 5> 소스 기반 취약점 진단 도구의 분류 목록

	Yasca	LAPSE+
1. SQL Injections	O	O
2. Cross-Site Scripting	O	O
3. Command Injection	O	O
4. Path Traversal	X	O
5. XPath Injection	X	O
6. XML Injection	X	O
7. LDAP Injection	X	O
8. Remote Code Execution	O	O
9. HTTP Response Splitting	X	O

5. 결론

본 논문의 분석결과를 보면, 오픈소스 기반의 취약점 진단도구는 OWSAP Top 10에서 권고하는 모든 보안요소를 진단하는 기능이 제공되는 것이 아니라 진단도구별로 일부분의 기능만 제공되고 있었다.

또한 동작중인 웹 서비스를 대상으로 동적테스트를 수행하여 탐지할 수 있는 취약점과 소스 레벨에서 진단할 수 있는 취약점이 서로 상이하고, 그 취약점을 진단할 수 있는 진단도구마다의 특성이 달랐다. 하지만 웹 서비스에 대한 보안 취약점 공격은, 한 가지 취약점의 노출로도 웹 서비스 전체가 위험 환경에 놓일 수 있다. 따라서 웹 서비스 프로바이더 및 개발, 운영자들은 테스트 기반 탐지도구와 소스 기반 진단도구 모두를 활용하여 다양한 경우의 취약점 공격에 대비를 해야 한다. 또한, 향후에 알려지지 않은 취약점에 대해서도 끊임없이 연구/분석하여 더 안전한 웹 서비스 환경을 만들어야 한다.

본 논문에서는 이러한 안전한 웹 환경을 구축해 나가기 위한 초석으로, 웹 서비스 취약점을 탐지/진단 할 수 있는 도구에 대한 분석정보를 제공함으로써 안전한 웹 서비스의 개발과 운영에 기여할 것으로 기대한다.

참고문헌

[1] OWASP, “OWASP Top 10-2013 : The Ten Most Critical Web Application Security Risks”, OWASP, 2013
[2] 행정안전부, “소프트웨어 개발보안 가이드”, 2012.
[3] 행정안전부, “소프트웨어 보안약점 진단가이드”, 2012.
[4] ZAP, https://code.google.com/p/zaproxy/
[5] OpenVAS, http://www.openvas.org/
[6] WATOBO, http://sourceforge.net/apps/mediawiki/watobo/
[7] Yasca, http://www.yasca.org/
[8] LAPSE+, https://code.google.com/p/lapse-plus/