파이썬 모듈과 정규표현식을 활용한 웹 취약점 탐색 자동화 봇

김남규* · 김기환** · 이훈재***
*동서대학교 정보보안트랙
**동서대학원 유비쿼터스 IT
***동서대학교 컴퓨터공학부

Browser fuzzing and analysis using known vulnerability

Nam-gue Kim* · Ki Hwan Kim** · Hoon-Jae Lee***
*Dept. of Information and Communication Engineering, Dongseo University
**Dept. of Ubiquitous IT Graduate School of Dongseo University
***Dept. of Computer Engineering, Dongseo University
E-mail : knq512412@gmail.com, ghksdl90@naver.com, hjlee@dongseo.ac.kr

요 약

웹이 보편화되면서 많은 편리함을 가져온 동시에 주 사용자인 기업에는 대외비 문서서, 중요 자원의 유출, 개별 사용자에게는 개인 정보 유출 등의 보안 이슈 또한 가져왔다. 이에 관련 기관들은 웹에 대한 취약점을 정리한 리스트를 제시하였다. 본 지에서는 웹 취약점의 표준으로 널리 사용되는 OWASP top10의 리스트들을 웹에서 자동으로 탐색하고 문서화하는 봇의 구현 모델을 제시한다.

ABSTRACT

Internet technology is universal, news from the Web browser, shopping, search, etc., various activities have been carried out. Its size becomes large, increasing the scale of information security incidents, as damage to this increases the safety for the use of the Internet is emphasized. IE browser is ASLR, such as Isolated Heap, but has been continually patch a number of vulnerabilities, such as various protection measures, this vulnerability, have come up constantly. And, therefore, in order to prevent security incidents, it is necessary to be removed to find before that is used to exploit this vulnerability. Therefore, in this paper, we introduce the purge is a technique that is used in the discovery of the vulnerability, we describe the automation technology related thereto. And utilizing the known vulnerabilities, and try to show any of the typical procedures for the analysis of the vulnerability.

키워드

IE, Internet Explorer, Browser, Fuzzing

Ⅰ. 서 론

웹 취약점에 대한 광범위한 범위의 자동 탐색은 업무 효율을 증대시킨다. 웹은 PC 환경에 국한된 것이 아니라 모바일에서도 널리 사용된다.

2장에서는 봇의 개념 및 웹의 취약점에 대한 표준 중 하나인 OWASP top10에 대해 기술한다. 3장에선 파이썬 웹 파싱 모듈인 beautiful soup, http 모듈, REGEX 등을 사용해서 파싱한 웹 리소스들을 토대로 보편적 취약점을 자동으로 탐색하는 아이디어와 방법을 제시한다.

Ⅱ. 웹 어플리케이션 취약점

2장에서는 봇의 개념 및 취약점 탐색 봇이 찾을 취약점에 대하여 몇 가지 선별하여 기술한다.

2.1 봇

봇은 조직적, 자동화된 방법으로 월드 와이드

웹을 탐색하는 컴퓨터 프로그램이다. 봇에 대한 다른 용어로는 앤트, 자동 인덱서, 웹 크롤러, 웜, 웹 스파이더, 웹 로봇 등이 있다.

2.2 OWASP top 10

OWASP(The Open Web Application Security Project)는 오픈소스 웹 어플리케이션 보안 프로젝트이다. 2013년에 10대 웹 어플리케이션의 취약점에 대해 발표했다. OWASP TOP 10은 웹 어플리케이션 취약점 중에서 빈도가 많이 발생하고, 보안상 영향을 크게 줄 수 있는 것들 10가지를 선정하여 3년 단위로 발표된다.

OWASP Top 10 – 2010 (Previous)	OWASP Top 10 – 2013 (New)
A1 – Injection	A1 – Injection
A3 – Broken Authentication and Session Management	A2 – Broken Authentication and Session Management
A2 – Cross-Site Scripting (XSS)	A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References	A4 – Insecure Direct Object References
A6 – Security Misconfiguration	A5 – Security Misconfiguration
A7 – Insecure Cryptographic Storage – Merged with A9 →	A6 – Sensitive Data Exposure
A8 – Failure to Restrict URL Access – Broadened into →	A7 – Missing Function Level Access Control
A5 – Cross-Site Request Forgery (CSRF)	A8 – Cross-Site Request Forgery (CSRF)
<buried in A6: Security Misconfiguration>	A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards	A10 – Unvalidated Redirects and Forwards
A9 – Insufficient Transport Layer Protection	Merged with 2010-A7 into new 2013-A6

그림 1. OWASP TOP 10

2.2.1 Injection(A1)

SQL삽입, 명령어 삽입, LDAP 삽입과 같은 취약점이 포함되며, 주요 원인은 신뢰할 수 없는 외부 값에 의해 발생되며 명령어 실행 또는 접근이 불가능한 데이터에 대한 접근 등의 취약점을 발생시킨다.

2.2.2 XSS(A2)

신뢰할 수 없는 외부 값을 적절한 검증 없이 웹 브라우저로 전송하는 경우 발생되는 취약점으로, 사용자 세션을 가로채거나, 홈페이지 변조, 악의적인 사이트 이동 등의 공격을 수행할 수 있다.

2.2.3 CSRF(A5)

로그온된 피해자의 웹 브라우저를 통해, 세션 쿠키 및 기타 다른 인증정보가 포함되어 변조된 HTTP 요청을 전송시켜, 정상적인 요청처럼 보이게 하는 기법으로 물품구매, 사이트 글 변조 등의 악의적인 행동을 하는 취약점을 의미한다.

위에 나열한 취약점들은 스크립트 또는 툴을 활용한 탐색이 가능하다. 풋프린팅의 자동화로 효율적인 업무처리가 가능할 것이다.

III. 취약점 탐색 자동화

3장에서는 자동화 탐색에 필요한 도구들을 소개한다. 파이썬 모듈들이 주를 이룰 것이며 해당 도구들을 활용하여 어떤 식의 구현을 요하는지 기술한다. 또한 프로토타입으로 구현한 봇에 어떠한 테스트 절차를 거쳐야 할지 제시한다.

3.1 concept 및 도구 소개

봇 실행 시 root page(부모가 되는 페이지)를 지정한다. 이후 beautifulsoup가 모든 데이터를 긁어서 저장한다. 이 중 url 또는 취약한 부분을 모두 검색한 다음 추출된 url에 http 요청을 날린다. 해당 요청으로 받은 페이지에서 나온 데이터를 파싱한 후 취약한 부분을 찾고 요청을 반복한다. 정규 표현식의 사용은 방대한 데이터 검색 부분을 더 빠르게 처리할 수 있도록 만들어준다.

3.1.1 beautiful soup

게시판이나 웹페이지에 있는 데이터를 긁어 오기 위해 사용할 수 있는 모듈이다. beautiful soup를 활용하여 root page의 모든 데이터를 긁어온다.

3.1.2 REGEX

빠르게 긁어모은 데이터를 검색할 수 있다. 파이썬에서 REGEX 모듈로 지원한다.

3.1.3 http 모듈

파싱한 후 필터링된 url list들에 요청을 할 경우 사용된다.

3.2 봇 테스트

concept대로 구현된 봇의 테스트 방법에 대해 기술한다.

3.2.1 테스트 방법

테스트 시 요청자는 탐색 봇이 되고 수신자는 특정 웹페이지가 될 것이다. 테스터는 서버와 연동된 웹페이지를 구축한 후 취약한 DB 또는 웹페이지 구성을 하여 취약점을 수동으로 체크한 후 해당 취약점이 봇에서도 탐지되는지 체크한다.

예컨대, 로그인 기능을 구현했다면 beautiful soup는 해당 url을 파싱하여 저장할 것이고, 해당 url을 필터링해서 탐색 목록에 추가시킬 것이다.

이후 풋프린팅 및 취약 구문을 request하여 원하는 반응이 나오는지 체크한다. 만약 MsSQL이라면 “ad”+“min”으로 해당 DB의 종류가 MsSQL인지 확인할 수 있다. 이러한 절차를 통하여 해당 웹페이지에 취약한 부분이 있는지에 대한 탐색을 자동화할 수 있다.

Ⅳ. 결 론

본지에선 웹에 대한 취약점에 대한 기술 및 해당 취약점의 탐색을 자동화하는 방법에 대해 제시했다. 정형화된 탐색 방법에 대한 자동화와 이를 토대로 효율적인 취약점 탐색 작업이 가능해진다. 또한 이러한 탐색 방법에 봇의 개념을 적용하여 특정 페이지에서 연결되는 모든 url에 대한 알려진 취약점을 탐색하고 문서화하여 광범위한 탐색을 도모한다.

감사의 글

이 논문은 2013년 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행되었다. (과제번호:2013-071188) 또한 부산광역시에서 지원하는 BB21 과제에서 지원받았음.

참고문헌

[1] 김 지 홍, 김 휘 강, “시스템 취약점 분석을 통한 침투 경로 예측 자동화 기법“, 정보보호학회, 22호 5권, 2012.10

[2] teamcrak, “Taint Analysis 연구분석보고서”, URL : http://teamcrak.tistory.com/328, 2011.01.17

[3] wikipedia, “Symbolic execution”, URL : https://en.wikipedia.org/wiki/Symbolic_execution