안전한 오픈소스 소프트웨어를 위한 침해여부 판정 모듈연구

정하규*, 박근일*, 전문석*
*숭실대학교 컴퓨터학과
e-mail:standard@ssu.ac.kr
higa_ps15@ssu.ac.kr
mjun@ssu.ac.kr

A Study on Exploit Vulnerability Decision Module for Secure Open-Source Software

Hague Chung*, Geunil Park*, Moon-Soeg Jun*
*Dept. of Computer Science & Engineering, Soongsil University

요 약

최근 IT 제품의 발달로 무선화, 지능화 등 다양한 분야에서 소프트웨어 활용이 증가하고 있으며, 이에 따라 오픈소스 소프트웨어의 사용이 증가되고 있다. 하지만 공개 오픈소스 소프트웨어는 악의적인 문제 발생 시 일일이 조사, 발견해 낼 수 없는 단점이 있다. 본 논문에서는 안전한 소프트웨어 개발을 위해 정적/동적 취약적 분석을 활용한 취약점 침해여부 판전 모듈 설계를 제안하였다.

1. 서론

본 논문은 오픈소스 SW 취약점 분석기술을 통해 탐지된 취약점이 익스플로잇터블한지 검증하기 위한 “취약점 침해야부 판정 모듈”의 분석 및 설계를 제안한다. 취약점이란 정보 시스템의 기밀성, 무결성, 가용성이 눈에 보이거나 보이지 않게 훼손되는 결과를 초래하거나 초래할 수 있는 일련의 상태라 정의할 수 있다 [1]. 정적 취약점 분석이란 실제 실행 없이 컴퓨터 소프트웨어를 분석하여 취약점을 검출해내는 것을 말한다 [2]. 정적 분석 기법은 완성된 어플리케이션에 대해 내제된 취약점을 탐지하기 위해서 사용되기도 하지만, 어플리케이션을 개발하는 시점에서 취약점을 발생시킬 수 있는 코드로 판명된 패턴, API 등을 검사할 수 있도록 한다. 따라서 완성된 어플리케이션에 대한 유지, 보수비용을 줄일 수 있는 방법으로 많이 사용된다. 정적 분석 기법으로 사용되는 방법들은 패턴 매칭이 대표적인 방법이며, 구문 분석을 이용하는 파싱, 어휘 분석 등이 있다. 동적 분석이란 소스 코드가 없어도 취약점을 분석할 수 있는 방법으로 실제로 실행되고 있는 어플리케이션에 값을 입력하여 도출되는 결과를 통하여 취약점을 분석하는 기법이다 [3]. 동적 분석 시범은 정적 분석 기법이 할 수 없는 기능적인 결함에 대해서 탐지할 수 있다. 정적 분석 기법은 소스코드를 기반으로 취약점을 탐지하기 때문에 실제 실행되었을 때 예상하지 못하는 결과에 대해서는 탐지할 수 없다. 하지만 동적 분석 방법은 더미 퍼징의 경우 입력 가능한 모든 값을 주입하기 때문에 개발자가 예상하지 못하는 입력의 결과에 대해서도 탐지할 수 있으며 정적 분석 기법보다 상대적으로 정탐율이 높은 것이 동적 분석 기법의 장점이다 [4]. 본 논문에서는 정적 취약점 분석과 동적 취약점 분석을 활용한 취약점 침해여부 판정 모듈 설계를 소개한다. 취약점 침해여부 판정 모듈은 기존의 정적/동적 취약점 분석 도구들의 정확성을 향상시키는데 목적이 있다.

2. 관련연구

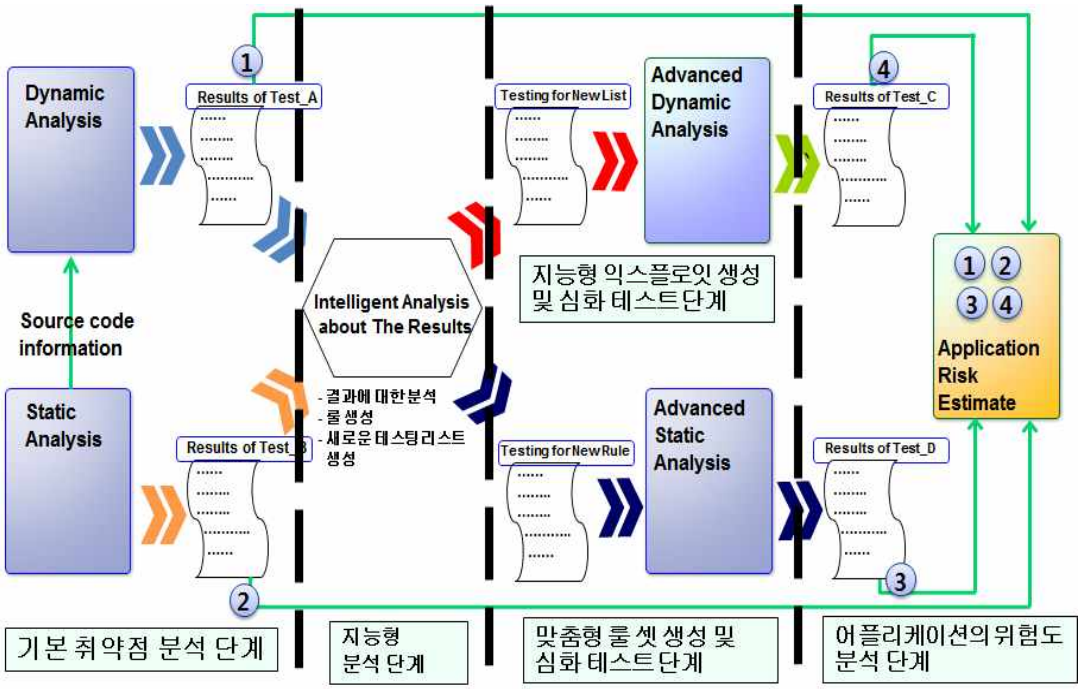
2.1 오픈소스 소프트웨어

오픈 소스 소프트웨어(Open Source Software)는 소스코드를 공개해 누구나 특별한 제한 없이 수정, 이용 등 자유롭게 활용할 수 있는 소프트웨어를 말한다 [5]. 오픈소스는 다양한 분야에서 각광받고 있다. 하지만 공개 오픈소스 소프트웨어는 악의적인 문제 발생 시 해결책 제시에 대한 책임이 불분명하다. 또한 개발과 배포과정에서 일일이 조사, 발견해 낼 수 없는 점 때문에 사용자의 지적재산권 사이에 충돌이 일어날 가능성이 있다.

3. 취약점 분석 절차

제안하는 기법은 크게 4단계를 거치게 되어 있다. 그림1 은 제안하는 취약점 침해 여부 판정 모듈의 전체 구조를 동작에 따라 배치한 것이다.

※ This work was supported by the ICT R&D program of MSIP/IITP. [R0112-14-1061, The analysis technology of a vulnerability on an open-source software, and the development of Platform]



(그림 1) 취약점 분석 절차

Step 1. 기본 취약점 분석 단계로 기존의 정적 분석 도구와 동적 분석 도구를 활용하여 취약점을 탐지하는데, 두 도구들의 상호 작용을 통해 동적 분석 도구의 효율성을 향상시킨다.

Step 2. 기본 취약점 분석 결과를 분석하는 단계로 1단계의 결과를 분석한다. 우선 정적한정 취약점, 동적한정 취약점, 정적-동적 매핑 취약점을 리스트화 한다. 그리고 여기에 속하지 않은 취약점에 대해 정적 취약점의 경우에는 동적 활용 가능한 메타정보 및 취약점 정보를, 동적 취약점의 경우에는 룰 생성이 가능한 메타정보 및 취약점 정보를 3단계와 4단계에서 활용할 수 있도록 구성한다.

Step 3. 맞춤형 룰 셋 생성 및 심화테스트 단계로 2단계에서 생성된 정보를 이용해서 심화된 정적 분석을 수행한다. 이 때 사용하는 정보는 취약점에 대한 자체 정보와, 1단계 기본 취약점 분석 단계의 동적 분석 모듈에서 도출된 결과를 2단계에서 분석한 것이다. 정적 분석은 동적 분석된 결과의 2단계 결과를 바탕으로 해당 취약점을 발견할 수 있는 새로운 룰을 구성하여 검사 후 탐지 결과를 분석하고, 동적 분석은 정적 분석된 결과의 2단계 수행 결과를 바탕으로 해당 취약점을 실행할 수 있는 페이로드를 구성하여 실행 후 로그를 분석한다.

Step 4. 3단계까지 수행한 결과들을 종합하여 해당 어플리케이션에서의 취약점들의 위험도와 어플리케이션 자체의 위험도를 평가한다. 이를 리포팅 가능한 정보로 종합하며, 누적된 결과를 분석하여 분석 결과의 신뢰성을 갱신한다.

4. 결론

이와 같이 정적 분석 모듈과 동적 분석 모듈을 크로스로 체크하여 탐지 정확도를 높이는 것이 본 연구에서 제안하는 주요 기법이며, 각 단계별로 생성되는 결과들을 분석 도구간 상호 작용을 통해 크로스 체크를 위한 정보를 생성하는 것 또한 본 연구의 핵심이다. 또한 탐지된 취약점들의 정보를 이용하여 분석 대상이 된 얼마나 위험한지를 평가할 수 있음을 증명한다.

참고문헌

[1] 박미영, 승현우, 임양미. "가상화 환경 위험도 관리체계화를 위한 취약점 분석." 인터넷정보학회논문지 14.3 (2013): 23-33.
[2] 행전안전부, "소프트웨어 개발보안 가이드" 5월, 2012.
[3] 중소기업청, "시큐어코딩 점검 시스템 개발", 2014. 07.
[4] Korea OSS Promotion Forum. "Business Guide of Open Software", 2014.
[5] 김형주, et al. "융복합 전자정부 서비스를 위한 전자정부 표준프레임워크 기반 시큐어코딩 점검 시스템 설계 및 개발." 디지털융복합연구 13.3 (2015): 201-208.