Journal of the Korea Academia-Industrial cooperation Society
Vol. 18, No. 11 pp. 46-52, 2017

https://doi.org/10.5762/KAIS.2017.18.11.46
ISSN 1975-4701 / eISSN 2288-4688

소스코드의 취약점 이력 학습을 이용한 소프트웨어 보안 취약점 분석 시스템

이광형[1*], 박재표[2]
[1]서일대학교 소프트웨어공학과, [2]숭실대학교 정보과학대학원

A Software Vulnerability Analysis System using Learning for Source Code Weakness History

Kwang-Hyoung Lee[1*], Jae-Pyo Park[2]
[1]**Division of Software Engineering, Seoil University**
[2]**Graduate School of Information Sciences**

요 약 최근 ICT 및 IoT 제품의 활용 분야가 다양화 되면서 오픈소스 소프트웨어의 활용 분야가 컴퓨터, 스마트폰, IoT 디바이스 등 다양한 기기와 환경에서 활용되고 있다. 이처럼 오픈소스 소프트웨어의 활용분야가 다양해짐에 따라 오픈소스의 보안 취약점을 악용하는 불법적인 사례가 지속적으로 증가하고 있다. 이에 따라 다양한 시큐어 코딩을 위한 도구나 프로그램이 출시되고 활용되고 있지만 여전히 많은 취약점들이 발생하고 있다. 본 논문에서는 안전한 오픈 소스 소프트웨어 개발을 위해 오픈 소스의 취약점 분석 결과에 의한 이력과 패턴을 지속적으로 학습하여 신규 취약점 분석에 활용할 수 있는 방법을 제안한다. 본 연구를 통해 취약점 이력 및 패턴 학습기반의 취약점 분석 시스템을 설계하였으며, 프로토타입으로 구현하여 실험을 통해 시스템의 성능을 평가하였다. 5개의 취약점 항목에 대해 평균 취약점 검출 시간은 최대 약 1.61sec가 단축되었으며, 평균 검출 정확도는 약 44%point가 향상된 것을 평가결과에서 확인할 수 있었다. 본 논문의 내용 및 결과는 소프트웨어 취약점 연구 분야에 대한 발전과 소프트웨어 개발자들의 취약점 분석을 통한 시큐어 코딩에 도움이 될 것을 기대한다.

Abstract Along with the expansion of areas in which ICT and Internet of Things (IoT) devices are utilized, open source software has recently expanded its scope of applications to include computers, smart phones, and IoT devices. Hence, as the scope of open source software applications has varied, there have been increasing malicious attempts to attack the weaknesses of open source software. In order to address this issue, various secure coding programs have been developed. Nevertheless, numerous vulnerabilities are still left unhandled. This paper provides some methods to handle newly raised weaknesses based on the analysis of histories and patterns of previous open source vulnerabilities. Through this study, we have designed a weaknesses analysis system that utilizes weakness histories and pattern learning, and we tested the performance of the system by implementing a prototype model. For five vulnerability categories, the average vulnerability detection time was shortened by about 1.61 sec, and the average detection accuracy was improved by 44%. This paper can provide help for researchers studying the areas of weaknesses analysis and for developers utilizing secure coding for weaknesses analysis.

Keywords : Open source, Secure coding, Secure weakness, Weakness history, Weakness learning,

1. 서론

최근 ICT 및 IoT 기술의 발달로 오픈 소스 소프트웨어가 컴퓨터뿐만 아니라, 스마트 단말 및 IoT 디바이스 등 다양한 기기와 다양한 산업 분야에서 활용되고 있다 **[1-7].** 이처럼 오픈소스 소프트웨어의 활용분야가 다양

본 논문은 2017학년도 서일대학교 학술연구비에 의해 연구되었음.
*Corresponding Author : Kwang-Hyoung Lee(Seoil Univ.)
Tel: +82-2-490-7226 email: dreamace@seoil.ac.kr
Received October 23, 2017 Revised November 2, 2017
Accepted November 3, 2017 Published November 30, 2017

해지면서 오픈소스 소프트웨어의 보안 취약점을 악용한 불법적인 공격에 대한 위험도 커지고 있다. 이에 따라 오픈 소스 소프트웨어의 취약점을 근본적으로 해결하기 위한 시큐어 코딩의 필요성이 제기 되었다**[8]**.

현재 시큐어 코딩의 프로그램의 경우 형상자원의 변경이력 관리를 위해 관련 솔루션이 출시되고 있지만 보안 취약점을 이용한 해커들의 공격과 보안 이슈들을 해결하기 위한 시큐어 코딩이 소프트웨어의 소스코드에 적용되고 있지는 있다. 따라서 이전 버전 소스코드의 취약점이 수정되는 등의 버전 업이 수행되지 않고 있으며, 형상관리 솔루션을 이용한 배포관리 기능에서 보안 취약점이 해결되지 않는 상태에서 기존 소스코드의 이용으로 인해 새로운 취약점에 의한 피해가 발생하고 있다**[9]**.

또한, CMS(Configuration Management System)는 다양한 정보, 데이터, 그리고 콘텐츠를 체계적으로 관리하기 위한 솔루션으로 다양한 관리 기능과 효율적인 UI를 제공하지만, 개인 개발자가 개발한 API 모듈에 대해서는 소스코드의 취약점을 완벽하게 해결할 수 없으므로 보안에 취약한 API 모듈의 불법적인 공격이 가능하다는 문제점을 가지고 있다**[10]**.

본 논문에서는 오픈 소스의 시큐어 코딩을 위해 취약점 분석 시스템에 취약점 분석 이력 및 패턴 학습 기능을 지원하는 취약소스 형상관리 엔진을 연동하고 오픈소스의 취약점의 이력과 패턴을 학습하여 그 결과를 신규 오픈소스의 취약점 분석에 활용할 수 있는 취약 소스 형상관리 기반의 취약점 이력 및 패턴 학습을 이용한 취약점 분석 시스템을 제안한다. 2장에서는 관련연구로서 글로벌 취약점 정보와 취약소스 형상관리에 대해 고찰하고 3장에서는 제안한 취약점 이력 및 패턴 학습기반 취약점 분석 시스템의 설계에 대해 설명한다. 5장에서는 프로토타입 구현 및 성능 평가 결과에 대해 논하고, 5장에서는 결론을 맺는다. 제안한 시스템은 프로토타입으로 구현하여 취약점 분석 실험을 통해 성능을 검증하였다.

2. 관련 연구

2.1 글로벌 취약점 관련 정보

CWE(Common Weakness Enumeration)는 미국 국토안보부 내의 국가사이버보안국의 지원으로 MITRE에서 다양한 관점에서 소프트웨어의 약점을 세부적으로 분류하여 정의한 것이다. CWE는 국제 공공 부문에 무료로 제공되며, 소프트웨어의 취약점 정의와 취약성의 형식 또는 패턴과 함께 예제 코드를 기술하여, 취약점을 수정, 완화, 해결할 수 있는 방법을 설명하고 있다**[11-14]**.

CVE(Common Vulnerabilities and Exposures)는 시간에 따라 탐지된 보안 취약점을 분류 및 정리한 리스트로 CWE는 일반적이고 기본적인 소프트웨어 보안 취약점의 분류체계라고 한다면 CVE는 기 발견된 소프트웨어 보안 취약점의 히스토리에 대한 기록이라 할 수 있다**[15]**.

CAPEC(Common Attack Pattern Enumeration and Classification)는 소프트웨어 보안 취약점을 분류하는 것이 아니라 보안 취약점을 악용하는 불법적 공격 패턴을 분류한다. 미국 국토안보부에서 지원하고 있고, Cigital사가 주도로 리스트를 관리하고 불법적 공격패턴의 형식와 실제 사례, 소프트웨어의 불법적 공격패턴에 대한 취약점 점검 테스트 및 탐지 방법, 불법적 공격에 대응하는 보안 방법 등을 정리하고 있으며, 이러한 정보들을 활용하여 소프트웨어의 보안 취약점을 해결하는데 도움을 주고 있다**[16,17,22,23]**.

2.2 형상관리

형상관리는 소프트웨어의 생명주기 동안에 소프트웨어의 각 구성항목의 식별과 정의, 기본 라인의 설정, 구성항목 수정 및 공표의 통제, 구성항목 상태, 수정 요청의 기록 및 보고, 구성항목의 완전성, 일치성 및 정확성 보장, 구성항목의 저장 및 취급, 인도를 위한 관리, 기술절차를 체계적이고 효율적으로 유지하고 관리함으로써 소프트웨어의 가시성과 추적 가능성을 제공하며, 소프트웨어를 주기적으로 관리할 뿐만 아니라 소프트웨어 품질 보증을 제공하기 위한 관리 기법이다**[18]**.

또한 형상관리는 소프트웨어의 개발에 있어서 생명주기 전반에 정의되는 형상 항목과 이와 관련된 형상물들을 그룹핑하여 소프트웨어의 형상을 생성하고 이에 대한 변경을 체계적이고 효율적으로 관리하기 위한 소프트웨어 개발관리 기법 및 프로세스이다**[19]**.

2.3 취약소스 형상관리

취약(vulnerable) 소스 형상관리는 소프트웨어의 소스코드를 취약점 분석을 통해 취약점이 존재하는 소스코드의 취약한 부분 즉, 모듈에 대한 정보들과 취약점의 패턴을 검출한 후 검출 및 분석 결과들을 지속적으로 학습한다. 학습된 정보는 DB에 저장한 후 신규 소스코드의 취

약점을 점검할 때 그 결과 값을 이용하여 더욱 효율적으로 취약점을 신속하고 정확하게 검사할 수 있는 관리 기법이다[20,21].

3. 취약점 이력 및 패턴 학습기반 취약점 분석 시스템

3.1 시스템 구성

본 논문에서 제안하는 취약점 이력 및 패턴 학습기반 취약점 분석 시스템은 취약점 분석 엔진과 취약 소스 형상관리 엔진으로 구성된다. 시스템의 구성도는 그림 1과 같다.

취약 소스 형상관리 엔진에서는 취약점 분석 엔진으로부터 오픈소스의 취약점 분석 정보를 받아 취약점 학습 및 취약점 수정 이력 학습, 취약점 검출 가중치 계산, 취약점 학습 프로파일 생성을 수행하고, 생성된 취약점 학습 프로파일을 취약점 분석 엔진으로 전송하여 2차 점검부터 취약점 학습 프로파일링 기반으로 취약 소스 부문에 대해 취약점을 수행한다.

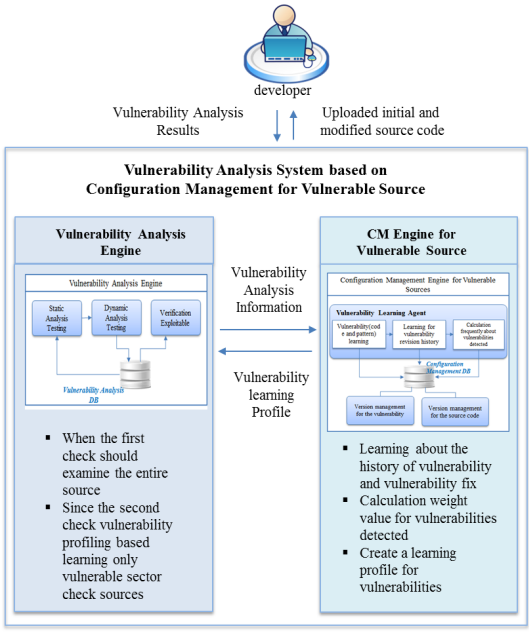

Fig. 1. Proposed System Architecture

3.2 취약소스 형상관리 엔진

취약점 관리 엔진에서는 업로드된 소스코드의 취약점 분석을 위해 정적분석, 동적분석 및 익스플로잇터블 검증을 수행하고 분석된 정보는 취약점 분석 DB를 통해 취약소스 형상관리 엔진에 전송된다.

취약 소스 형상관리 엔진에서는 취약점 분석엔진으로부터 취약점 분석 결과를 전송 받아 취약점 학습 에이전트를 통해 취약 코드와 취약 패턴과 취약점 수정 이력을 학습한 후 취약점 검출 가중치를 계산한다. 학습되고 계산된 정보는 취약점 형상관리 DB에 저장되고 관리되며, 취약점 프로파일을 생성하여 취약점 분석 DB로 전송하여 계산된 취약점 가중치에 따라 취약점 관리 리스트를 업데이트 한다. 이후 오픈 소스의 시큐어 코딩이 이루어진 후 재검사 시 패턴학습 결과에 따라 취약 패턴만 재검사한다. 그림 2는 취약점 분석 엔진 및 취약소스 형상관리 엔진의 구성도이다.

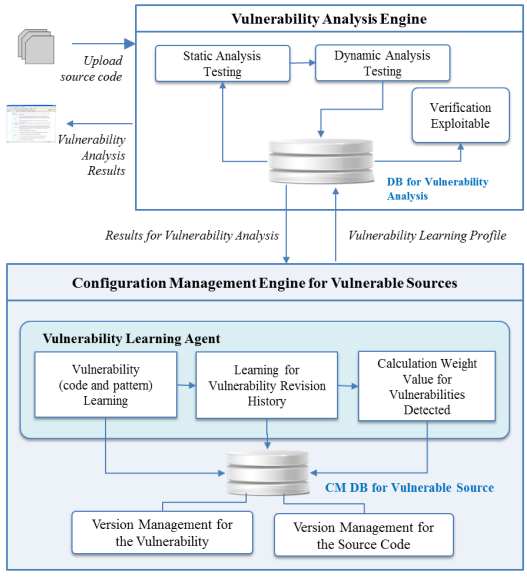

Fig. 2. Architecture of Vulnerability Analysis and Configuration Management Engine

3.3 취약 소스 형상관리 처리 프로세스

취약점 분석 엔진에서 오픈 소스가 신규 코드인지 또는 업데이트 버전 소스 코드인지를 판단하여 수정된 업데이트 버전 소스 코드일 경우 그림 3과 같이 취약 부문의 정적분석, 동적분석 및 익스프로이터블 검증을 수행한 후 분석 결과를 취약점 분석 DB에 저장한다.

2차 분석 결과 취약점이 다시 발견되면 취약 소스 형상관리 엔진에 분석 결과를 전송하여 해당 취약점의 코드와 패턴을 학습한다. 또한 취약점 분석 결과에 의해 개발자에 의해 수정된 취약점 이력을 학습한 후 취약점 가중치를 재계산하여 그 결과 값을 취약점 학습 DB에 저장한다. 이후, 취약점 학습 DB에 저장된 정보를 기반으로 취약점 분석 프로파일을 생성하여 취약점 분석 DB에 전송한 후 취약점 분석 항복 리스트를 업데이트 한다. 그림 3은 취약 소스 형상관리 처리 프로세스를 나타낸다.

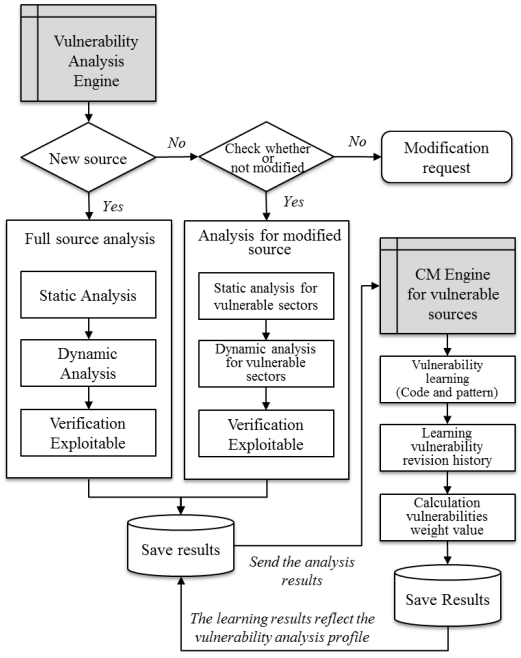

Fig. 3. Process of Configuration Management for Vulnerability Source

3.4 취약점 패턴 학습 및 취약점 가중치 계산

취약점 패턴은 국제적으로 활용하고 있는 소프트웨어 보안 취약점에 대한 가이드라인인 CWE와 소프트웨어 보안 취약점 분석에 사용되는 레퍼런스 코드인 CVE를 사용하여 이를 통해 기본적인 취약점 패턴을 정의한다.. CWE와 CVE에서 정의되지 않은 취약점 패턴은 신규 취약점 패턴으로 정의하여 취약 패턴에 새로 추가한다.

오픈소스의 취약점 패턴을 학습하기 위해서는 우선 오픈소스를 프로젝트 단위로 시스템에 등록한 후 취약점 분석 엔진을 통해 취약점 분석을 수행한다.

취약점 분석 후 분석 결과로 도출된 취약점이 CWE 항목과 비교하여 매칭되는 항목이 있으면 해당되는 CWE ID를 선택하여 매핑한 후 그 정보를 저장한다. 이후 해당되는 CWE 취약점 패턴 항목과 CVE 취약점 패턴이 매칭되는 항목이 있으면 해당되는 CVE ID를 선택하여 서로 매핑한 후 그 정보를 저장한다. 이후 취약점 분석에 의해 도출된 취약점 패턴은 CWE의 ID와 CVE의 ID와 매핑된 후 본 시스템에서 사용될 최종 취약점 패턴 ID로 정의되어 저장된다. 그림 4는 취약점 패턴을 학습하는 프로세스이다.

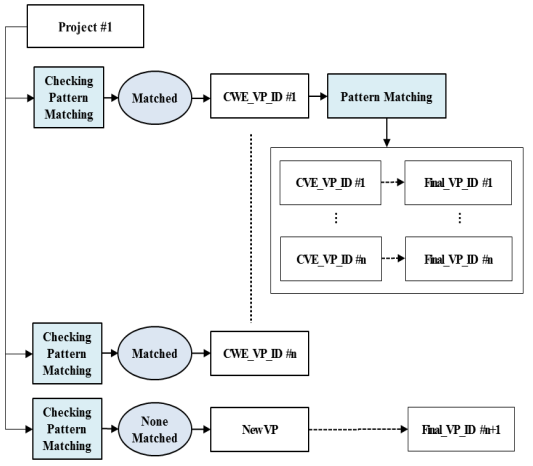

Fig. 4. Process of Vulnerability Pattern Learning

이 때, 취약점 분석 결과로 도출된 취약점 패턴이 CWE의 취약점 항목과 매칭되는 항목이 없으면 신규 취약점 패턴으로 등록한다. 취약점 분석 엔진을 통해 취약점 분석이 이루어질 때 마다 이러한 취약점 분석 프로세스를 통해 취약점 패턴이 학습되고 학습 결과 및 해당 정보가 지속적으로 업데이트 된다. 취약점 패턴에 대한 가중치 계산은 다음 수식과 같이 정의된다.

$$W_{Vulner(Final\_VP\_ID(K))} = \frac{N_{Hit\_Vulner(Final\_VP\_ID(k))}}{\sum_{i=1}^{n}(N_{Vulner(Final\_VP\_ID)(i)}}$$

(식 1)

취약점 패턴에 대한 가중치는 취약점 분석이 수행될 때 마다 취약점 분석에 의해 결과로 도출되는 취약점 ID 개수를 기존에 CM DB에 저장되어 있는 최종(final) 취약점 패턴 ID의 전체 개수로 나누어 계산한다. 따라서

취약점 분석이 지속적으로 수행될 때 마다 해당 취약점 패턴의 가중치는 계속 업데이트된다.

4. 프로토타입 구현 및 성능 평가

제안한 취약점 이력 및 패턴 학습을 이용한 취약점 분석 시스템은 프로토타입으로 설계되고 개발되었다. 그림 5와 그림 6은 프로토타입으로 개발된 취약점 이력 및 패턴 학습을 이용한 취약점 분석 시스템의 취약점 분석 결과 화면이다.

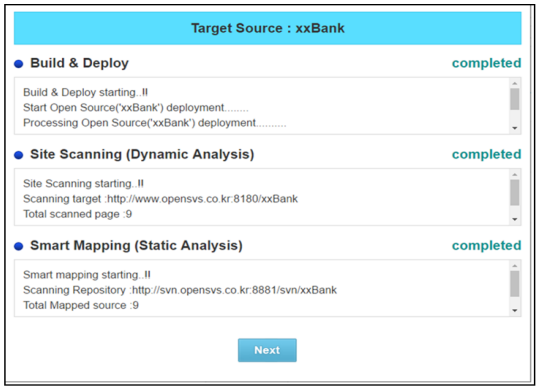

Fig. 5. The vulnerability analysis result for Scanning and Smart Mapping

그림 5에서 보이듯이 소스코드에 대한 스캐닝과 매핑을 통해 동적 분석과 정적분석을 수행한 결과를 확인할 수 있다.

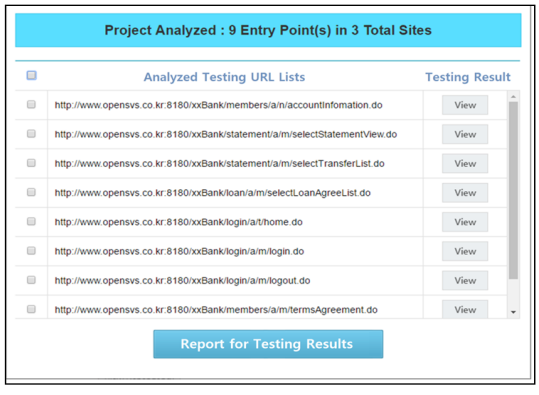

Fig. 6. The vulnerability analysis result for Analyzed Testing URL

그림 6에서는 취약점 분석에 의해 점검되고 분석된 오픈소스 URL 정보를 확인할 수 있다. 그림 7는 취약 소스 형상관리 엔진을 통한 취약점 패턴 분석 화면이다. 시스템에 입력된 오픈소스의 취약점 분석을 위해 동적 분석 및 동적 분석을 한 결과이다. 취약점 패턴 분석 화면에서는 취약점 패턴 ID, 취약점 패턴 정보 그리고 취약 패턴에 대해 계산된 가중치 값을 확인할 수 있다.

취약점이 발견된 해당 URL을 선택하여 상세 분석 결과를 확인할 수 있으며, 종합적인 취약점 분석 레포트를 조회할 수 있다.

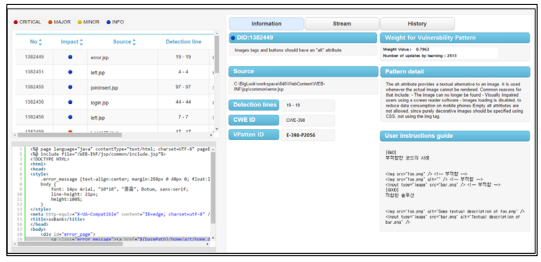

Fig. 7. The vulnerability analysis result

취약점 이력 및 패턴 학습을 이용한 취약점 분석 시스템의 성능을 평가하기 위해 오픈소스 소스코드에 의한 성능 실험을 통해 취약점 검출에 대한 속도와 정확도를 측정하였다. 시스템에 의해 취약점 분석이 이루어질 때 마다 취약점 패턴 학습 및 취약점 가중치 계산이 지속적으로 수행되고, 이를 통해 취약점 검출에 대한 속도와 정확도를 실험하여 측정하였으며, 그 실험 결과는 표 1과 같다.

Table 1. Test results for vulnerability detection accuracy of proposed system

Number of Tests / Security Vulnerabilities	1th		10th		20th		50th	
	DT (sec)	DA (%)	DT (sec)	DA (%)	DT (sec)	DA (%)	DT (sec)	DA (%)
Race Condition	2.34	35.9	2.82	43.8	1.68	52.8	1.12	73.1
Divide by 0	3.26	41.8	2.61	55.5	2.10	65.5	1.25	87.5
Not Reachable	3.69	24.2	3.23	32.9	2.84	49.5	1.49	77.2
Null Pointer Exception	2.22	44.2	2.48	54.3	1.82	64.6	0.72	86.4
Assertion Errors	2.01	34.9	2.15	45.6	1.91	67.8	0.89	76.8

* Juliet codes : http://samate.nist.gov/SARD/testsuite.php
* DT: Detection Time
* DA: Detection Accuracy

실험 대상 오픈 소스 파일은 356개의 juliet code(java)를 이용하였다. 표 1에서 보이듯이 제안한 시스템에 의해 취약점 분석이 이루어질 때 마다 취약점 검출에 대한 시간은 단축되었으며, 취약점 검출에 대한 정확도는 향상되는 것을 확인할 수 있었다.

5. 결론

본 논문에서는 안전한 오픈소스 소프트웨어 개발을 위해 취약점의 이력 및 패턴 학습기반의 취약점 분석 시스템을 제안하였다.

제안한 시스템은 프로토타입으로 구현하여 실제 활용되고 있는 오픈소스의 취약점 분석을 통해 취약점 검출 시간과 검출 정확도가 얼마나 향상되는지 실험하였다. 실험결과 지속적인 취약점 이력과 취약점 패턴 학습을 통해 취약점 검출에 대한 정확도가 향상되는 것을 확인하였다. 제안 시스템은 모든 java 기반의 오픈소스에 활용가능하며, 취약점 분석 결과에 따라 개발자의 오픈 소스가 수정된 후 재검사 시 취약 패턴 학습 결과에 의해 업데이트된 취약점 패턴 정보를 이용하여 해당 취약점만을 재검사함으로써 취약점에 대한 점검 및 분석 시간을 현저히 단축할 수 있는 장점이 있다.

향후 레포팅 모듈 기능과 전체적인 서비스 웹을 모두 구현한 후 상용 취약점 분석 시스템과의 비교실험을 통해 성능을 평가할 계획이다.

제안한 취약점의 이력 및 패턴 학습기반의 취약점 분석 시스템을 통해 기존의 취약점 분석 시스템이 가지고 있던 취약점 검출에 대한 정확성의 한계를 극복하고 자동화된 방안으로 더욱 정확하고 효율적으로 취약점을 검출할 수 있기를 기대한다.

References

[1] Jin-Hyeon Chang, “Improvement of The National Technical Qualifications System from ICT point of view”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 16, No. 2, pp. 189-199, Apr. 30, 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.2.189

[2] Mi-Hee Youn, Dongwon Kim, “A study of Development and Management on ASEAN Women’s ICT Development Index and Measurement”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 16, No. 4, pp. 181-187, Aug. 31, 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.4.181

[3] Young-Jun Jeon, Hee-Joung Hwang, “Design of Dynamic Buffer Assignment and Message model for Large-scale Process Monitoring of Personalized Health Data”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 15, No. 6, pp. 187-193, Dec. 31, 2015.
DOI: http://dx.doi.org/10.7236/JIIBC.2015.15.6.187

[4] Jee-Hyun Kim, Young-Im Cho, “A Study on National ICT Competency Model, The Journal of The Institute of Internet”, Broadcasting and Communication (IIBC), Vol. 15, No. 6, pp. 275-281, Dec. 31, 2015.
DOI: http://dx.doi.org/10.7236/JIIBC.2015.15.6.275

[5] Young-Jun Jeon, Seok-Jin Im, Hee-Joung Hwang, “Design of a Data Grid Model between TOS and HL7 FHIR Service for the Retrieval of Personalized Health Resources”, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 16, No. 4, pp. 139-145, Aug. 31, 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.4.139

[6] Gee-Hyun Hwang, “The Relationship among TQM Practices, Employee Satisfaction and Employee Loyalty in ICT Customer Service and Retail Distribution Organizations”, Journal of Society of Korea Industrial and Systems Engineering, Vol.38, No.1, pp. 188-198, 2015.
DOI : https://doi.org/10.11627/jkise.2014.38.1.188

[7] Eunhye Kim, Ju-Won Park, “Runtime Prediction Based on Workload-Aware Clustering”, J. Soc. Korea Ind. Syst. Eng, Vol. 38, No. 3, pp. 56-63, Sep. 2015.
DOI: http://dx.doi.org/10.11627/jkise.2015.38.3.56

[8] H. H. Chae, J. K. Lee, K. H. Lee, "A Study on The Security Vulnerability Analysis of Open an Automatic Demand Response System", Journal of digital Convergence , vol. 14, no. 5, pp. 333-339, 2016.
DOI: http://dx.doi.org/10.14400/JDC.2016.14.5.333

[9] H. J. Lee, O. C. Na, S. Y. Sung, H. B. Chang, “A Design on Security Governance Framework for Industry Convergence Environment ”, Journal of the Korea Convergence Society, vol. 6, no. 4, pp. 33-40, 2015.
DOI: https://doi.org/10.15207/JKCS.2015.6.4.033

[10] S. K. Choi, T. J. Hwang, Y. B. Park, “2011 CWE/SANS Top 25 Dangerous Software Errors-based Vulnerability analysis and Secure Coding of the Hadoop's MapReduce Framework,” Korea Computer Congress, 2013.

[11] CAPEC : Comon Attack Pattern Enumeration and Classification, http://capec.mitre.org/. Date accessed: 20/06/2016.

[12] 2011 CWE/SANS Top 25 Most Dangerous Programming Errors, http://cwe.mitre.org/top25/. Date accessed: 20/06/2016.

[13] Ji Hoon Kyung, Chong Su Kim, A Study on Measurements of IT Security Service Quality:Feasibility of Quantitative Measures, J. Soc. Korea Ind. Syst. Eng Vol. 38, No. 4, pp. 30-38, Dec. 2015.
DOI: http://dx.doi.org/10.11627/jkise.2015.38.4.30

[14] Hee-Ohl Kim, Dong-Hyun Baek, Prioritize Security Strategy based on Enterprise Type Classification Using

Pair Comparison, J. Soc. Korea Ind. Syst. Eng, Vol. 39, No. 4, pp. 97-105, Dec. 2016.
DOI: http://dx.doi.org/10.11627/jkise.2016.39.4.097

[15] Common Vulnerability Scoring System, http://www.first.org/cvss/. Date accessed: 20/06/2016.

[16] Common Weakness Enumeration, http://cwe.mitre.org. Date accessed: 20/06/2016.

[17] S. W. Cho, W. J. Jang, H. W. Lee, "mVoIP Vulnerability Analysis And its Countermeasures on Smart Phone", Journal of the Korea Convergence Society, vol. 3, no. 3, pp. 7-12, 2012.

[18] Y. J. Moon, A study on program configuration management methodology based on the configuration management practices of CMMI and SPICE, Master's Thesis, Dept. of Computer Engineering, Yonsei University, 2006.

[19] CODE Community, IT Framework, InforDream, 2006.

[20] J. H. Lee, S. J. Kim, J. P. Park, "A Development of Smart Fuzzing Tool Combined with Black and White Box Testing," Asia Pacific International Conference on Information Science and Technology(APIC-IST) 2016.

[21] S. S. Shin, J. I. Kim, J. J. Youn, "Vulnerability Analysis of the Creativity and Personality Education based on Digital Convergence Curation System", Journal of the Korea Convergence Society, vol. 6, no. 4, pp. 225-234, 2015.
DOI: https://doi.org/10.15207/JKCS.2015.6.4.225

[22] Myongyeal Lee, Jaepyo Park, Analysis and Study on Invasion Threat and Security Measures for Smart Home Services in IoT Environment, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 16, No. 5, pp. 27-32, Oct. 31, 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.5.27

[23] Ho-Yong Lee, Dong-Hoon Lee, Security of Ethernet in Automotive Electric/Electronic Architectures, The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 16, No. 5, pp. 39-48, Oct. 31, 2016.
DOI: http://dx.doi.org/10.7236/JIIBC.2016.16.5.39

이 광 형(Kwang-Hyoung Lee) **[종신회원]**

- 1998년 2월 : 광주대학교 컴퓨터공학과 졸업(공학사)
- 2002년 2월 : 숭실대학교 컴퓨터공학과 (공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 서일대학 소프트웨어공학과 부교수

<관심분야>
멀티미디어 데이터 검색, 영상처리, 멀티미디어 보안, DRM, USN, 학습콘텐츠

박 재 표(Jae-Pyo Park) **[정회원]**

- 1998년 8월 : 숭실대학교 일반대학원 컴퓨터학과(공학석사)
- 2004년 8월 : 숭실대학교 일반대학원 컴퓨터학과 (공학박사)
- 2008년 9월 ~ 2009년 8월 : 숭실대학교 정보미디어기술연구소 전임연구원
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

<관심분야>
물리적 보안, 컴퓨터통신, 보안정책, 디지털포렌식