오픈소스 보안취약점 관리를 통한 안전한 오픈소스 사용 방안

류원옥, 조수형, 이승윤
한국전자통신연구원 오픈소스센터
woryoo@etri.re.kr, shjo@etri.re.kr, syl@etri.re.kr

A study on the safe use of open source through open source security vulnerability management

Ryoo Won Ok, Su Hyung Jo, Lee Seung Yun
ETRI Open Source Center

요 약

전 세계적으로 오픈소스 사용은 매년 20% 이상씩 증가하고 있으며, 동시에 오픈소스 보안취약점 또한 매년 약 2,000개에서 3,000개씩 꾸준히 증가하고 있는 추세이다. 즉, 오픈소스의 공개는 동시에 보안취약점도 함께 공개되어 해킹의 표적이 되고 있다. 따라서 오픈소스 사용의 증가에 따라 해킹에 의한 위험도도 높아지고 있다는 것을 알 수 있다. 본 고에서는 지속적으로 증가하고 있는 오픈소스를 안전하게 사용하기 위한 보안취약점 현황 및 관리의 필요성을 살펴보고 안전한 오픈소스 사용 방안을 제시한다.

Ⅰ. 서 론

ICT 기술의 발전과 더불어 지속적으로 증가하고 있는 오픈소스를 안전하게 사용하기 위한 오픈소스 보안취약점 관리 방안을 알아본다. 전세계적으로 오픈소스 사용은 매년 20% 이상씩 증가하고 있고, 오픈소스 보안취약점은 매년 약 2,000개에서 3,000개씩 꾸준히 증가하고 있는 추세이다. 오픈소스 공개는 보안취약점도 함께 공개되어 해킹의 표적이 되고 있다. 따라서 오픈소스 사용의 증가에 따라 해킹의 위험도도 높아지고 있다는 것을 알 수 있다.

또한 시놉시스 조사에 따르면 96%의 소프트웨어가 오픈소스를 사용하고, 1000개 이상의 파일로 구성된 소프트웨어는 99%가 오픈소스를 포함하고 있다. 오픈소스 취약점 54%가 고위험군에 속하며, IoT 애플리케이션 77%가 오픈소스를 사용하고 평균 667개의 보안취약점을 갖고 있다.[1] 오픈소스의 경우 상용 소프트웨어와 달리 소스코드가 모두 공개되기 때문에 취약점이 쉽게 노출된다. 미국 국립표준기술연구소가 공개한 정보에 의하면 보안취약점 1위에 안드로이드 OS가 선정되었다고 한다.[2] 본고에서는 오픈소스 보안취약점의 현황 및 관리의 필요성을 살펴보고, 안전한 오픈소스 사용 방안을 제시하였다.

Ⅱ. 오픈소스 보안취약점

오픈소스 보안취약점은 라이선스 컴플라이언스와 함께 관리되고 있는 주요 대상이다. 그러나 대부분의 오픈소스 사용자들은 사용한 오픈소스의 라이선스 캄플라이언스에 대한 관리는 중요하게 생각하고 있으나 보안취약점에 대해서는 관리가 소홀한 편으로 많은 소프트웨어 개발자들은 2014년 발견된 하트블리드(HeartBleed) 취약점을 갖는 OpenSSL 버전을 현재에도 사용하고 있다. 이런 상황은 다양한 프로그램에서 발생하고 있다. 4년 전에 처음 사용했던 오픈소스에서 2년 전에 보안취약점이 발견되어 새 버전을 배포했으나 4년 전 소스는 수정되지 않아 보안취약점에 노출되어 있는 경우도 있다.[3]

오픈소스에서 발생하는 보안취약점은 각국에서 다양하게 관리하고 있고, 대표적으로 미국에서 사용하는 분류 체계는 다음과 같다.[4]

① CWE(Common Weakness Enumeration)
소프트웨어 취약점을 사전식으로 분류해 쉽게 찾아 볼 수 있도록 정보를 제공한다.

② CWSS(Common Weakness Scoring System)
소프트웨어 취약점의 중요도를 평가하는 평가체계로 CWE 프로젝트의 일부로 수행되고, CWE와 CWSS의 특징은 안전한 소프트웨어의 개발과 보안 유지에 책임이 있는 당사자들인 정부, 학계, 산업체들이 모여서 만든다는 것이다.

③ CVE(Common Vulnerabilities and Exposures)
시간대 별로 발생된 보안취약점 또는 위험 노출을 정리한 목록을 제공한다. CVE-[해당년도]-[일련번호]로 표현한다.

④ CVSS(Common Vulnerability Scoring System)
평가된 취약점의 우선순위를 부여하여 관리하기 위해, IT 취약점에 대한 영향과 특성을 표현하는 공통 프레임워크를 제공한다.

⑤ NVD(National Vulnerability Database)
보안 콘텐츠 자동화 프로토콜(SCAP)을 사용하여 표현된 미국 정부의 표준 기반 취약점 관리 데이터 저장소로, NVD는 보안 체크리스트, 소프트웨어의 결함, 잘못된 구성, 제품 이름, 평가 메트릭과 관련된 보안 데이터베이스를 포함하고 있다.

대표적인 오픈소스 보안취약점 관리 업체인 WhiteSource가 2020년에 공개한 Top10 보안취약점 소스, 관련 CVE/CVSS 정보 및 버젼 정보는 다음과 같다.[5]

순위	오픈소스명	CVE/CVSS	영향받는 버전
1	Lodash	CVE-2020-8203 CVSS: 7.4 High	4.17.2 이전
2	FasterXLjackson-databind	CVE-2020-24616 CVSS: 8.1 High	2.9.10.6 이전 2.x
3	HtmlUnit	CVE-2020-5529 CVSS: 8.1 High	2.37.0 이전
4	Handlebars	CVE-2019-20920 CVSS: 8.1 High	3.0.8 이전 및 4.x, 4.5.3 이전
5	http-proxy	WS-2020-0091	1.18.1 이전

		CVSS: 7.5 High	
6	decompress	CVE-2020-12265 CVSS: 9.8 Critical	4.2.1 이전
7	XStream	CVE-2020-26217 CVSS: 8.8 High	1.4.14 이전
8	Netty	CVE-2020-11612 CVSS: 9.8 Critical	4.1.46 이전 4.1.x
9	Spring Framework	CVE-2020-5398 CVSS: 7.5 High	5.2.3 이전 5.2.x, 5.1.13 이전 버전 5.1.x 및 5.0.16 이전 버전 5.0.x
10	PyYAML	CVE-2020-1747 CVSS: 9.8 Critical	5.3.1 이전

시놉시스가 펴낸 2020년 Open Source Security and Risk Analysis 보고서에서는 조사 대상 1,250개의 상업용 소프트웨어의 99%는 한 개 이상의 오픈소스를 포함하고 있고, 사용된 오픈소스의 82%는 4년 이상 된 구버전이다.

III. 오픈소스 보안취약점 관리

오픈소스를 안전하게 관리하기 위한 주요요소 및 소스코드 관리 시스템(GitHub/GitLab)에서 제공하는 의존성 검사 도구 및 주요 IT 기업의 보안취약점 관리 방안을 살펴본다.

오픈소스를 안전하게 관리하기 위해 주요 요소는 다음과 같다.[6]

1. **내 소프트웨어 알기** - 소나타입(Sonatype) 2020년 설문에 따르면, 대부분 기업이 기업의 소프트웨어 애플리케이션에 사용되는 모든 오픈소스 구성요소와 각각 해당하는 취약점에 대한 완벽한 시야를 갖추지 못하고 있어 오픈소스 프로젝트에서 취약점이 발표되면 그 오픈소스 구성요소를 사용한 적이 있는지, 있다면 어디인지, 이 두 가지를 즉시 확인해야 한다고 조언하고 있다.
2. **종속성 문제 해결** - 베라코드(Veracode)의 2020년 소프트웨어 보안 현황 보고서에는 흔히 볼 수 있는 소프트웨어 보안 문제를 지적한다. 개발자 스스로가 아닌, "상호연결된 종속성"으로 인해 애플리케이션 내에 잠재적인 위험이 간접적으로 유입되고, 이 위험이 대부분의 개발자 시야에서 벗어나 방치된다는 것이다.
3. **코드 스캔 자동화를 통해 알려지지 않은 불확실한 요소 찾기**
   깃허브와 같은 리포지토리 운영업체는 정기적인 오픈소스 프로젝트 스캔은 숨겨진 취약점과 버그 외에도, 개인 키 및 인증 정보가 기여자에 의해 우발적으로 공개되는 경우와 같은 데이터 유출 신호를 잡아낸다. 자동화 툴을 사용해 광범위한 보안 감사를 하면 문제의 구성요소가 공급망으로 흘러 들어가기 전에 오픈소스 생태계 내의 신뢰와 무결성 문제에 대처하는 데 도움이 된다.
4. **라이선싱 위험에 주의**
   시놉시스(Synopsys)의 2020년 오픈소스 보안 및 위험 분석 보고서는 "공식적인 라이선스 충돌은 코드 베이스에 포함된 오픈소스 구성요소의 라이선스가 코드 베이스 전체에 대한 라이선스와 상충할 때 발생한다."고 한다. 블랙덕(Black Duck) 보고서에 따르면, 2019년에 감사된 코드 베이스의 67%에 라이선스가 충돌하는 구성요소가 포함됐다. "GPL은 널리 사용되는 오픈소스 라이선스 중 하나이며, 다양한 GPL 버전은 코드 베이스의 다른 코드와 라이선스 충돌을 일으킬 수 있다. 실제로 충돌 요소가 있는 상위 10개 라이선스 중 5개가 GPL 및 그 변형이다"라고 분석했다.

주요 소스코드 관리 시스템(Github/Gitlab)에서 제공하는 분석 도구 중에는 의존성 검사 도구로 Github에는 Dependabot, Gitlab에는 Dependency Scanning 이 있다.

- Dependabot은 사용하는 오픈소스에 대한 업데이트 유무를 자동으로 확인해주는 도구로, 사용하는 오픈소스가 최신 버전이 아닌 경우 개별 PR(Pull Request)을 생성하여 사용자가 최신 버전의 merge 여부를 결정할 수 있도록 한다.
- Dependency Scanning은 오픈소스에서 사용된 종속성의 버전과 함께 어떤 취약점이 있는지 자동으로 알아볼 수 있어, 종속성 취약점 탐지에 용이하다.

보안에 강력하다고 알려진 소프트웨어들은 일반적으로 최종 릴리즈 전에 보안취약점 분석 단계를 필수로 도입하고, 보안취약점 분석에 투자하고 있다.

- 마이크로소프트는 Semmle의 취약점 분석 기술을 Github에 통합시켜 자사 소스코드의 취약점을 검사한다.
- 구글은 세계 최고의 취약점 분석 팀(Project Zero)을 운영하여 오픈소스 프로젝트를 포함한 유명 소프트웨어의 취약점을 분석하여 공개한다.
- 삼성은 AVAS(Automated Vulnerability Analysis System)을 개발하여 오픈소스의 취약점 분석을 자동화하고 있다.

V. 결론

오픈소스 관리자를 대상으로 한 스니크(Snyk)의 설문조사에 따르면, 오픈소스 관리자의 44%가 오픈소스에 대한 취약점 분석을 수행하지 않는다고 한다.[5] 오픈소스는 무료이고 보안에 안전하다는 인식, 그리고 보안취약점 분석에 필요한 역량 부족으로 보안취약점 분석 없이 무분별하게 사용하고 이후에 소홀하게 관리하는 경우가 많다.

소프트웨어를 개발하는 기업은 오픈소스 사용에 따른 보안 위험 및 취약점 피해를 최소화하기 위해서는 다음과 같은 관리가 필요하다.

① 오픈소스 전담부서를 통한 체계적인 관리시스템 구축
② 사용하는 오픈소스에 대한 명확한 정보(라이선스, 버전, 저자 등) 및 의존성 및 종속성 관리
③ 오픈소스 라이선스 및 특허에 대한 충돌 및 보복조항 관리
④ 오픈소스 보안취약점에 대한 주기적인 스캔닝 및 소스 수정 관리

ACKNOWLEDGMENT

본 연구는 한국전자통신연구원 내부연구과제의 일환으로 수행되었음. [21YR1200, ETRI 오픈소스 거버넌스 고도화 및 개방형 R&D 활동 지원].

참 고 문 헌

[1] 데이터넷, 웹·이메일 보안③ 오픈소스 라이선스·보안취약점 관리, 2020.10.12. (http://www.datanet.co.kr/news/articleView.html?idxno=151523)
[2] 기태현, 오픈소스 보안 관리의 중요성, 2021, https://www.oss.kr/oss_guide/show/6f523bc2-99b6-4b60-ad11-4c252872f0aa
[3] 강태진, 3자 공급 코드의 위협, 2021, https://www.oss.kr/oss_guide/show/713f9f9d-df31-4886-825f-9038dae33d33?page=1
[4] Top 10 Open Source Vulnerabilities In 2020, https://www.whitesourcesoftware.com/resources/blog/top-security-open-source-vulnerabilities-2020/
[5] 내부문서, 코드 패턴 검색을 이용한 오픈소스 보안취약점 분석 연구, 2021.11.27.
[6] ITWord, 오픈소스 코드 속 보안취약점을 피하는 실무 팁 4가지(Ax Sharma, CSO), 2020.08.20.