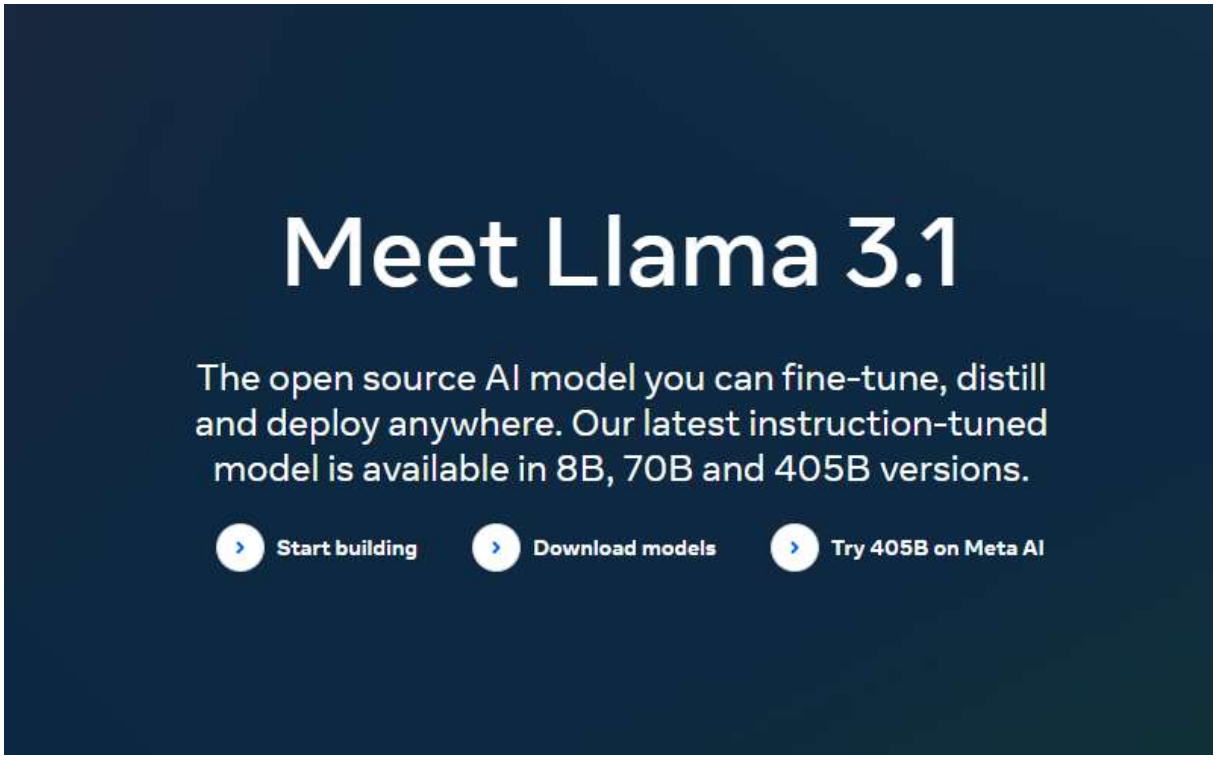
Headline

SBOM(Software Bill of Materials)을 활용한 오픈소스 SW 관리 방안

금융컨설팅 1 팀 이해범 수석

■ 개요

1990 년대, 기업들은 IT 시스템을 구축하며 관련 사업을 시작했다. 당시 소프트웨어(SW) 개발은 전문 지식을 가진 종사자나 독학으로 연구하는 소수의 매니아들이 할 수 있는 영역이었다. 그러나 2000 년대 중반에 들어서면서 IT 산업이 확장되고, 인터넷과 모바일 기술로 전 세계가 연결되면서 약간의 지식이나 관심만 있으면 누구나 소프트웨어를 만들 수 있는 세상이 열렸다. 2024 년 현재는 AI 가 소프트웨어를 자동으로 생성하는, 마치 SF 영화에서나 볼 법한 일이 현실이 되었다. 이 모든 것을 가능하게 만든 핵심은 바로 '오픈소스 SW'다.

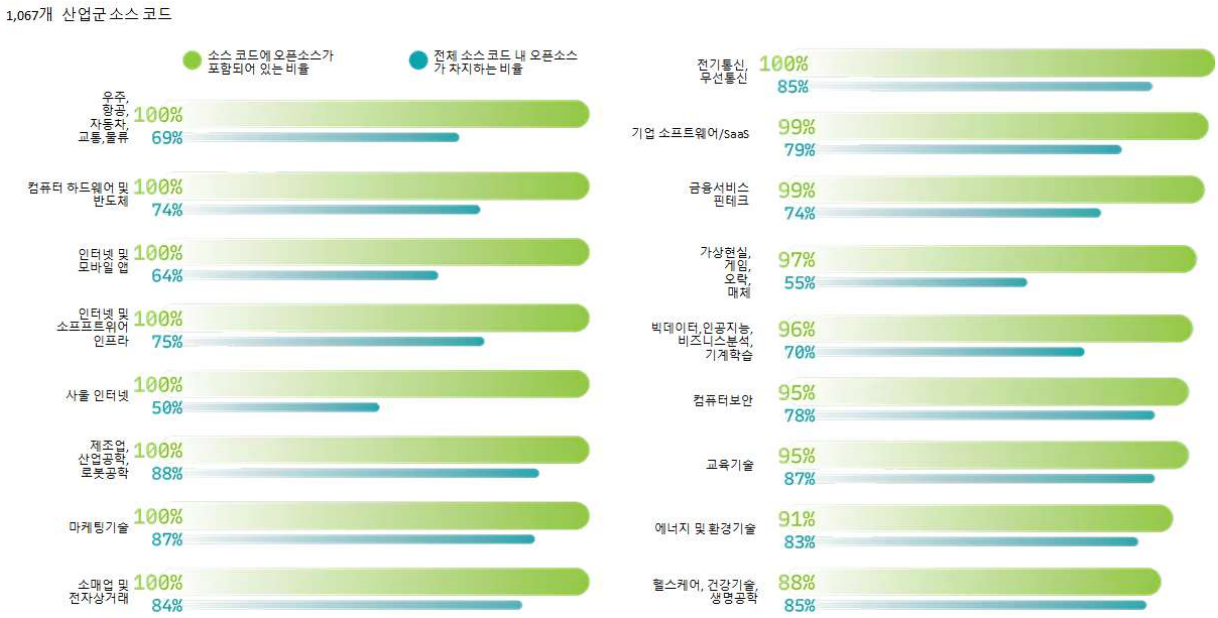


* 출처: Meta Liama

그림 1. 오픈소스 AI 언어 모델 Llama 3.1

■ 소프트웨어 공급망과 오픈소스 SW

과거, 소프트웨어 구매 비용이 매우 부담스러웠던 시절이 있었다. 그러나 IT 는 특정 기업이나 개인의 독점이 되어서는 안 된다는 신념을 가진 초창기 선구자들이 있었다. 이들은 자신들의 재능을 활용해 소프트웨어를 개발하고, 이를 무료로 사용할 수 있도록 공개했다. 이후 인터넷을 통해 오픈소스 커뮤니티가 발전하면서 손쉬운 교육, 빠른 개발, 그리고 엄청난 비용 절감 효과를 제공하게 되었다. 그 결과, 오늘날 대부분의 사업에서 오픈소스 SW 가 널리 사용되고 있다.



* 출처: 2024 오픈소스 보안 및 위험 분석 보고서 (https://www.synopsys.com)

그림 2. 오픈소스 사용률 추이

전통적으로 소프트웨어 공급방식은 시작코드부터 종료코드까지 단일 또는 소수의 조직이나 기업에서 만드는 방식이었다. 이후 오픈소스 SW 가 발전하면서 소프트웨어 공급망 관리(Software Supply Chain Management)라는 개념이 등장하게 되었다.

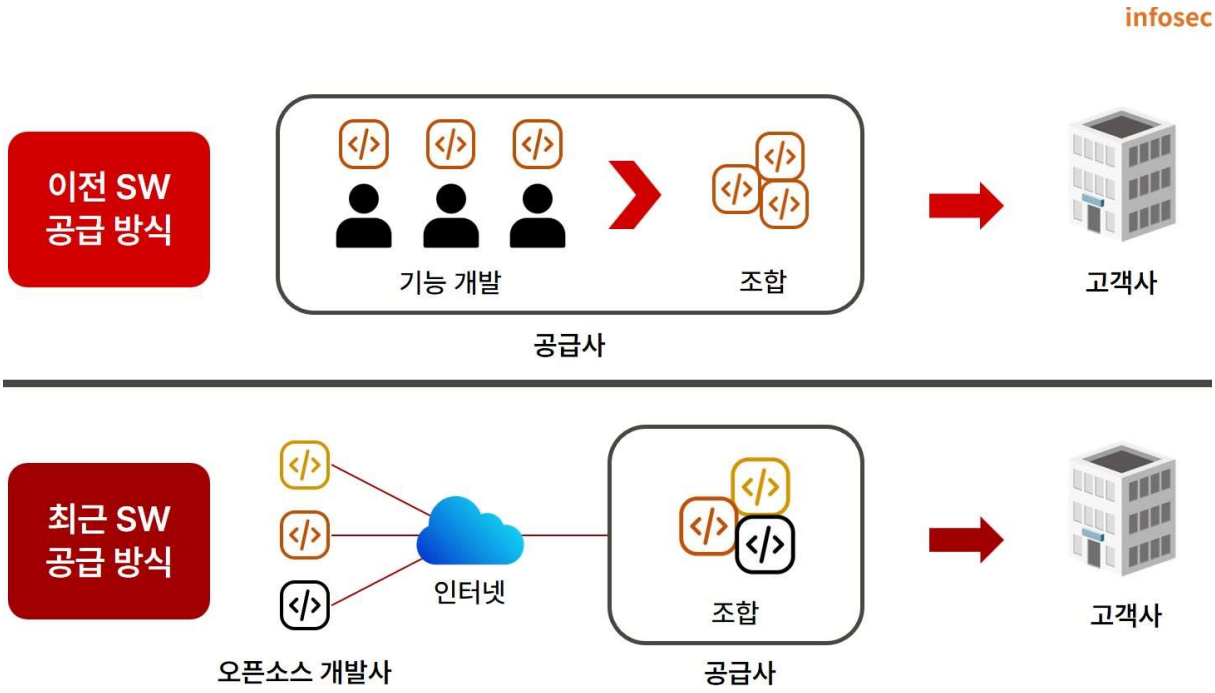
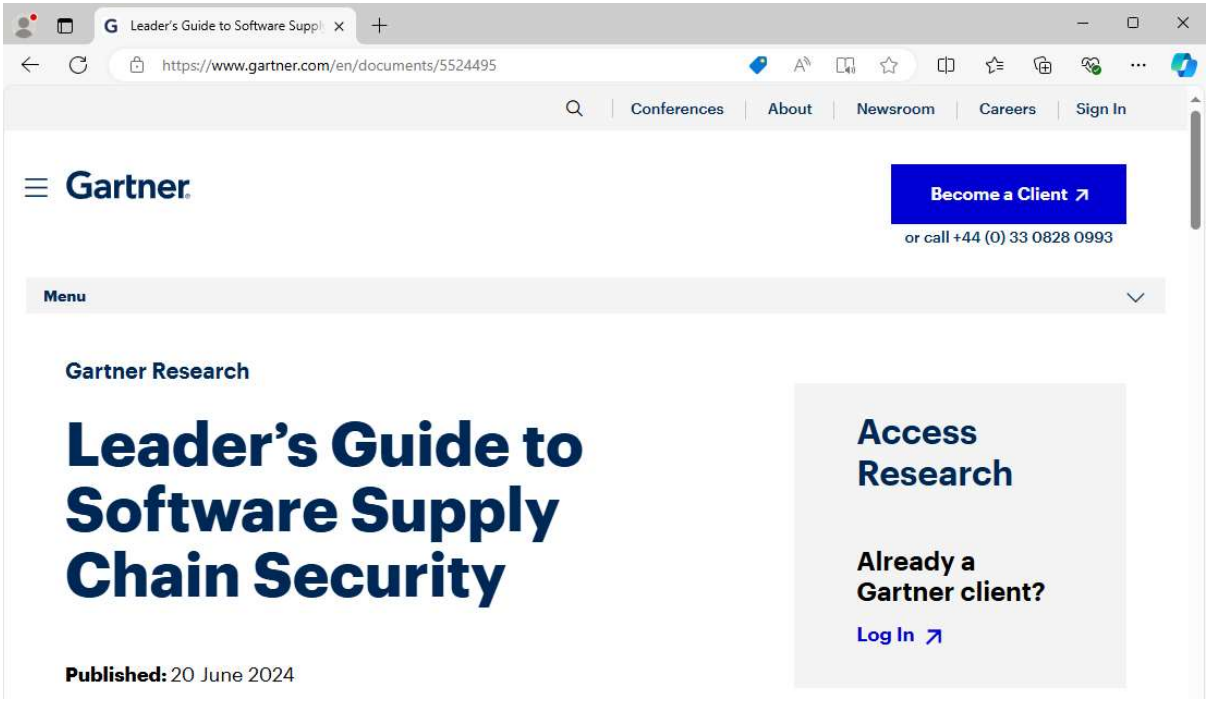


그림 3. 소프트웨어 공급망 개념도

소프트웨어 공급망에 참여하는 참가자와 소프트웨어가 증가하면서 관리의 난이도가 점점 높아지고 있다. 최근에는 오픈소스 SW 의 보안 취약점을 이용한 침해사고 건수도 지속적으로 증가하는 추세다. 2024 년 6 월 가트너의 '소프트웨어 공급망 보안을 위한 리더 가이드' 자료에 따르면 오픈소스 SW 사용 증가에 따라 다음과 같은 문제점이 발생할 것으로 전망되고 있다.

1. 소프트웨어 공급망 공격으로 인한 피해액이 2023 년 460 억 달러에서 2031 년 1,380 억 달러로 증가할 것으로 추정

2. 기업, 기관이 사용하는 소프트웨어의 90% 이상이 오픈소스 종속성을 포함하고 74%는 고위험 종속성 포함



* 출처: 가트너 홈페이지

그림 4. 소프트웨어 공급망 보안을 위한 리더 가이드

■ 소프트웨어 공급망 보호를 위한 노력

이미 광범위하게 사용되고 있는 오픈소스 SW 로 인해 소프트웨어 공급망에 대한 공격은 더이상 한 개인이나 기업의 문제가 아니다. 이러한 공격은 해당 오픈소스 SW 를 사용하고 있는 정부, 기업, 조직, 산업 등 국가와 사회 전반에 걸쳐 영향을 끼치고 있다.

공격명 (발생 연도)	사고 내용 및 피해 현황
SolarWinds (2020)	러시아 기반 해킹 그룹의 공격으로 IT 소프트웨어 공급사의 소프트웨어 개발환경 및 배포 시스템이 해킹되어 18,000 개 이상의 기관이 피해를 입음
CodeCov (2021)	컨테이너 이미지 보안취약점을 악용해 소스코드 검증을 위한 배포 환경의 인증 정보가 유출, 전 세계 2 만 9 천여 개 고객사에 영향을 끼침
Colonial Pipeline (2021)	송유관 관리사의 IT 시스템이 랜섬웨어에 감염되어 미국 남동부 8,900km 일대 공급이 중단, 지역 비상사태 선포와 약 50 억 원의 랜섬머니(Ransom Money) 지급 발생
Log4Shell (2021)	Log4j 의 제로데이 보안취약점과 공개된 개념 증명 코드를 악용하여 악성코드를 심고, 전 세계 취약 서버를 대상으로 대량의 해킹 공격 발생
Kaseya (2022)	클라우드 기반 IT 원격 관리 솔루션 서버를 해킹하고, 업데이트 파일로 위장한 랜섬웨어를 고객사에 배포. 17 개국 1,500 여 개 조직이 피해
3CX (2023)	악성코드가 삽입된 X-트레이더 금융 소프트웨어를 다운받은 PC 를 감염시켜 60 만 명 이상의 고객과 1,200 만 개 조직으로 전파
이니세이프 (2023)	금융 보안인증 소프트웨어의 보안취약점을 악용해 PC 해킹 및 악성코드 유포로 국내 61 개 기관, 총 207 대의 기관, 기업, 개인 PC 해킹 피해

* 출처: 과학기술정보통신부 SW 공급망 보안 가이드라인(2024.05)

표 1. 소프트웨어 공급망 주요 침해사고

전 세계 국가들은 보안의 심각성을 인지하고 소프트웨어 공급망 보안을 위한 규정, 지침, 가이드 등을 수립하고 준수하도록 제도를 마련하고 있다.

국가	정책, 제도 현황
미국	[21.05] 바이든 정부 행정명령(EO 14028, '21.5 월) • 연방정부에 납품되는 소프트웨어에 대한 SBOM 제공 [23.03] 식품의약국(FDA) 의료기기 사이버보안 강화 • FDA 에 승인을 요청하는 모든 제조사는 장치의 공개 소프트웨어 및 상용 소프트웨어의 구성요소 목록을 포함하는 SBOM 제공
유럽연합	[22.09] 사이버 복원력 법(Cyber Resilience Act, 이하 CRA) 제정안 발의 • 역내에 공급(유통)되는 디지털기기의 SBOM 제출을 의무화
일본	[22.05] 경제산업성 내 SW TF 설치 • 경제산업성에 SW TF 를 설치하고 의료·자동차·소프트웨어 분야에 걸친 SBOM 실증(PoC)사업 진행

* 출처: 과학기술정보통신부 SW_공급망_보안_가이드라인(2024.05)

표 2. 소프트웨어 공급망 보호를 위한 각국의 정책, 제도 현황

국가는 다르지만 특정 문제점에 대해 사람들이 생각하는 방식은 비슷하다. 앞의 자료에서 언급한 것을 살펴보면 한가지 **공통적인 해결책은 바로** SBOM(Software Bill of Materials)이다.

■ SBOM 의 등장

SBOM 에 대한 표준연구는 국외에서 이미 활발하게 진행중에 있으며, 국내에서도 SBOM 표준이 수립되어 있다. 대부분의 항목이 유사하며 어떤 표준이 가장 좋다고 단정하기는 어렵기 때문에, 각 기업별 상황에 맞는 표준을 선택하고 활용할 수 있다.

표준 포맷	설명
SPDX®(Software Package Data Exchange	2011 년 리눅스 재단에서 개발해 2021 년 국제표준(ISO/IEC 5962:2021)으로 등록 • 오픈소스 라이선스 관리와 SBOM 포맷 활용에 용이하며, 소프트웨어 패키지와 관련된 컴포넌트, 라이선스, 저작권 및 보안 정보를 전달
CycloneDX(CDX)	OWASP 커뮤니티에서 개발, 공급망 기능을 지원하는 풀스택 BOM 산업 표준을 지향하며, '17 년도에 초기 프로토타입 공개를 시작으로 현재 1.4 버전까지 공개. • 처음부터 SBOM 포맷으로 특화설계, SaaS BOM 을 포함한 다양한 사양을 지원
SWID (Software Identification)	NIST 가 2009 년에 공개하여 2015 년도에 국제표준(ISO/IEC19770-2:2015)으로 등록 • 소프트웨어 제품의 특정 릴리즈에 대한 정보를 포함하고 있으며, 소프트웨어 정보에 대한 태그를 생성하여 장치에 설치된 상용 및 오픈소스 SW 인벤토리를 지원
TTAK.KO-11.0309	한국정보통신기술협회(TTA)에서 제정한 공개 소프트웨어 공급망 관리를 위한 소프트웨어 목록 구성(SBOM) 속성 규격 • 다양한 소프트웨어 공급망과 사용목적에 따른 가변적인 소프트웨어 구성요소목록 관리를 위한 15 가지의 소프트웨어 구성요소 관리 항목을 제시

* 출처: 한국전자통신연구원 SW 공급망 관리 및 SBOM 동향 (2023.08)

표 3. 국내외 SBOM 표준

표준별로 SBOM 을 표시하는 방식이나 명칭은 조금씩 다르다. 미국 전기통신정보관리국(NTIA) 에서는 SBOM 구성에 필요한 최소항목과 각 표준과의 관계를 제시했다.

속성	SPDX	CycloneDX	SWID	TTAK.KO-11.0309
Author Name	(2.8) Creator:	metadata/authors/ author	<Entity> @role(tagCreator), @name	ComponentAuthor:
Timestamp	(2.9) Created:	metadata/ timestamp	<Meta>	ReleaseDate:
Supplier Name	(3.5) PackageSupplier:	Supplier Publisher	<Entity> @role (softwareCreator/ publisher), @name	ComponentSupplier:
Component Name	(3.1) PackageName:	name	<softwareIdentity> @name	ComponentName:
Version String	(3.3) PackageVersion:	version	<softwareIdentity> @version	ComponentVersion:
Component Hash	(3.10) PackageChecksum: (3.9) PackageVerification Code:	Hash "alg"	<Payload>/../<File> @[hash-algorithm]:hash	FileChecksum:
Unique Identifier	(2.5) SPDX Document Namespace (3.2) SPDXID:	bom/serialNumber component/bom-ref	<softwareIdentity> @tagID	FormatID:
Relationship	(7.1) Relationship: DESCRIBES CONTAINS	(Inherent in nested assembly/subassembly and/or dependency graphs)	<Link> @rel, @href	IncludeComponent, ImportComponent

* 출처: Framing Software Component Transparency: Establishing a Common Software Bill of Materials (SBOM) (2021.10)

* 출처: 한국전자통신연구원 SW 공급망 관리 및 SBOM 동향 (2023.08)

표 4. SBOM 표준 간 데이터 속성

■ 오픈소스 SW 관리를 위한 SBOM 의 필요성

일부 자료에서는 SBOM(소프트웨어 자재 명세서)을 식품에 부착된 성분표시 정도로 설명하기도 하지만, 사실은 이보다 더 중요한 개념이다. 이를 이해하려면, SBOM 을 마치 인기 있는 식당의 비법 소스를 만드는 레시피에 비유할 수 있다. 이 레시피가 경쟁 식당에 유출되면 그 식당의 영업에 심각한 타격을 줄 만큼 중요한 정보다.

어느 날, 그 식당의 손님들이 비법 소스를 먹고 배탈이 나기 시작해, 식당의 신뢰도와 매출이 하락한다. 이에 사장은 비법 소스에 사용된 재료의 목록과 구매처를 점검하던 중, 몇 일 전 정전이 발생한 공장에서 공급받은 재료가 문제가 있음을 발견한다. 이 문제를 해결하기 위해 그는 즉시 다른 공급업체로부터 동일한 재료를 받아 다시 소스를 만들고, 고객의 신뢰를 회복할 수 있었다.

과거의 소프트웨어는 일반적으로 단일 또는 소수의 개발자나 업체가 처음부터 끝까지 전부 개발하였지만, 오늘날에는 오픈소스 SW 의 사용이 보편화되었다. 이로 인해, 보안 사고가 발생할 경우 어디에서 문제가 생겼는지 신속하게 찾아내고 보안 대책을 수립하는 일이 더 이상 부가적인 업무가 아닌 필수적인 업무가 되었다. 소프트웨어 공급망 관리 측면에서, 보안 취약점을 해결하고 고객 신뢰를 확보하기 위해 SBOM 관리가 필수적이다.

오픈소스 SW 입장에서는 억울할 수도 있지만, 전 세계적으로 관리와 감시의 대상으로 여겨지는 이유는 다음 세 가지로 요약될 수 있다.

1. 보안 취약점 이슈
상용 소프트웨어는 보안 취약점 발생 시 제조사나 공급사가 친절하게 보안 패치나 조치 방법을 제공하지만, 오픈소스 SW 는 사용자 스스로 문제를 찾아 해결해야 한다.

2. 라이선스 이슈
오픈소스 SW 가 무료로 제공되지만, 이를 무료로 사용하려면 사용자가 직접 라이선스 조건을 확인하고 이에 맞춰 조치를 취해야 한다.

3. 평판 이슈
때로는 특정 집단에서 개발되었다는 이유만으로 보안성을 의심받는 경우도 있다.

조직이나 기업에서 사용중인 소프트웨어 또는 애플리케이션의 SBOM 을 관리하면, 오픈소스 SW 사용 시 발생할 수 있는 문제들을 어느 정도 해결할 수 있다.

1. 신속한 보안 취약점 확인 및 대응
상용 소프트웨어의 경우, 보안 취약점이 발생하면 공급사가 이를 자체적으로 분석하고 고객에게 해결 방안을 제공한다. 반면, 오픈소스 SW 는 사용자가 직접 보안 취약점 여부를 확인하고 해결해야 한다. SBOM 을 체계적으로 관리하면 사용 중인 오픈소스 SW 를 신속하게 식별하고, 보안 취약점에 대한 대책을 빠르게 수립할 수 있다.

2. 라이선스 식별 및 적절한 조치
상용 소프트웨어는 구매 및 계약 단계에서 비용을 지불하고, 조직의 IT 운영 환경에 맞는 라이선스를 획득하기 때문에 라이선스 관련 문제가 드물다. 그러나 오픈소스 SW 는 무료로 제공되더라도 복잡한 라이선스 조건을 준수해야 한다. 담당자가 라이선스를 잘못 해석하거나 식별하지 못하면 법적 소송에 휘말릴 수 있다. SBOM 을 통해 오픈소스 SW 의 라이선스 정보를 관리하면, 소프트웨어 개발 초기 단계에서 법적 문제를 사전에 해결할 수 있다.

3. 개발사 또는 공급사에 대한 보안성 검토 및 사용지속 여부 결정

2024 년 4 월, 미국은 인기 동영상 공유 플랫폼인 '틱톡'의 사용을 법적으로 금지했다. 이는 단지 개발사가 중국 업체이기 때문에, 중국 정부가 개인정보를 요구할 가능성이 있다는 우려 때문이었다. SBOM 에 개발사와 공급사에 대한 정보를 포함해 관리하면, 이러한 이슈 발생 시 적절한 보안 대책을 세우고, 필요에 따라 소프트웨어를 교체하거나 사용을 중단하는 등의 결정을 신속하게 내릴 수 있다.

SBOM 관리는 소프트웨어의 투명성을 높이고 보안, 법적 문제, 공급망 리스크 등을 효과적으로 관리하는 데 중요한 역할을 한다.

■ 오픈소스 SW 관리체계 수립을 위한 몇 가지 방안

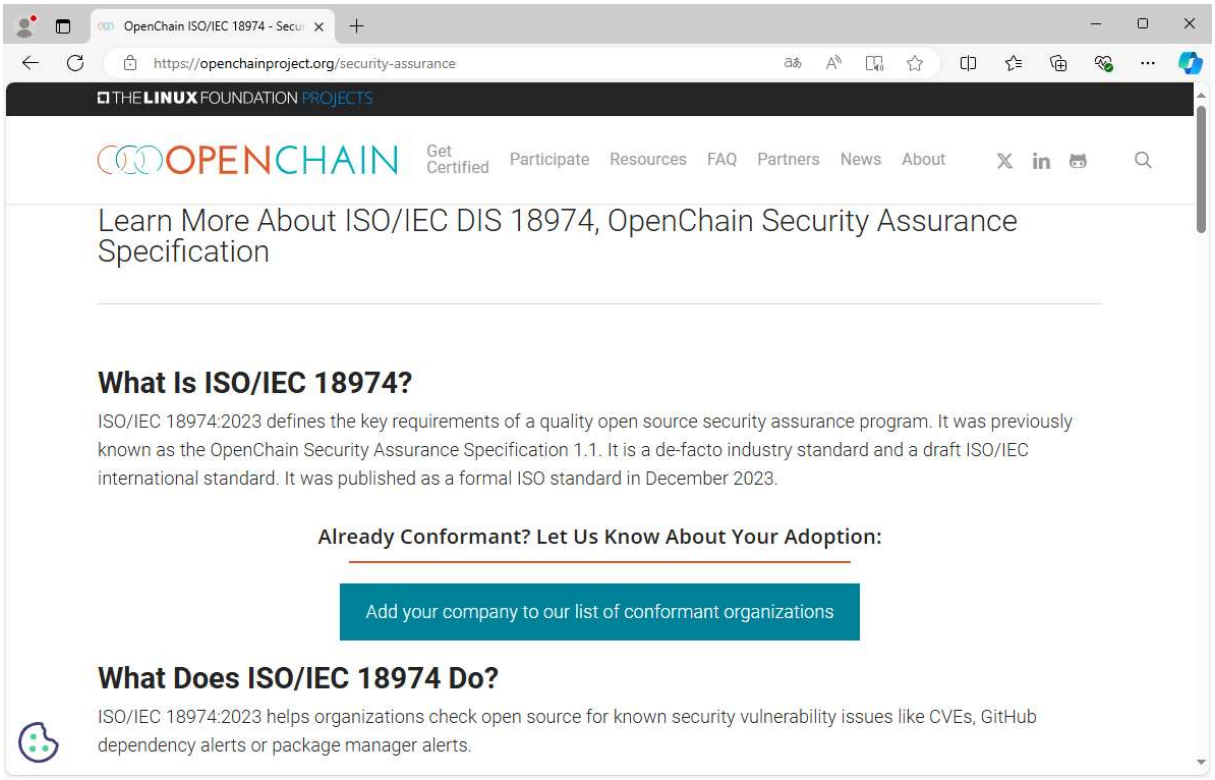
최근 우리나라에서도 소프트웨어 공급망 보안과 오픈소스 SW 관리를 위한 활동이 활발히 진행되고 있다. 금융당국은 오픈소스 SW 활용 가이드를 발간하고 있으며, 금융회사 담당자를 대상으로 소프트웨어 공급망 보안을 위한 주기적인 포럼을 개최하고 있다. 또한 은행기관을 중심으로 SBOM 관리를 위한 자동화 시스템을 도입하거나 검토하는 움직임이 확산되고 있다.

그러나 자동화 시스템을 도입한다고 해서 모든 문제가 해결되는 것은 아니다. 시스템마다 고유한 특성이 있어 오탐지나 탐지 누락 문제가 발생할 수 있으며, 오픈소스 SW 의 특성상(정보보호와 법률 준수 측면에서) 관리 주체와 업무 담당자를 명확히 정해야 한다. 또한 위험 평가의 방법론도 수립해야 하는 과제가 있다. 이러한 문제들을 통합적으로 관리하기 위해서는 오픈소스 SW 관리체계의 수립이 필수적이다.

여기에서는 오픈소스 SW 관리체계에 대한 국제 표준인 ISO/IEC 18974:2023 의 주요 내용을 소개하고, 실제로 오픈소스 SW 관리체계를 도입할 때 고려해야 할 몇 가지 사항을 공유하고자 한다.

1. ISO/IEC 18974:2023

ISO/IEC 18974:2023 은 고품질 오픈소스 보안 보증 프로그램의 주요 요구 사항을 정의한 산업 표준이자 ISO/IEC 국제 표준 초안으로 2023 년 12 월에 공식 ISO 표준으로 발표되었다.



* 출처: ISO/IEC 18974:2023

그림 5. ISO/IEC 18974:2023

ISO/IEC 18974:2023 에서는 다음의 3 가지 항목에 대한 관리방안을 제시하고 있다.

① 보안 프로세스를 갖추어야 하는 핵심 장소
② 역할 및 책임을 할당하는 방법
③ 보안 프로세스의 지속 가능성을 보장하는 방법

이 사이트 에서는 ISO/IEC 18974:2023 관리체계 도입을 희망하는 기업들을 위해 ISO 표준에서 필요로 하는 요구사항에 대해 오픈소스 자체 인증 체크리스트를 제공하고 있다. 오픈소스 SW 관리체계 수립이 필요한데 방법을 고민하고 있는 조직이 있다면 아래 체크리스트를 이용해 보기를 권장한다.

섹션	요구사항
4.1.3	• 프로그램 참가자는 오픈 소스 보안 보증 정책과 이를 찾을 수 있는 위치를 알고 있습니다. • 프로그램 참가자는 관련 오픈 소스 목표를 알고 있습니다. • 프로그램 참가자는 프로그램의 효과를 보장하기 위해 예상되는 기여에 대해 알고 있습니다. • 프로그램 참가자는 프로그램 요구 사항을 따르지 않을 경우의 결과를 알고 있습니다.
4.1.4	• 당사는 프로그램의 범위와 한계를 명확하게 정의하는 서면 진술서를 가지고 있습니다. • 프로그램 성과를 측정하기 위한 일련의 메트릭이 있습니다.

	• 우리는 지속적인 개선을 입증하기 위해 각 검토, 업데이트 또는 감사에서 문서화된 증거를 가지고 있습니다.
4.1.5	• 당사는 제공된 소프트웨어에 대한 구조적, 기술적 위협을 식별할 수 있는 방법을 가지고 있습니다. • 당사는 제공된 소프트웨어에서 알려진 취약점의 존재를 탐지하는 방법을 가지고 있습니다. • 식별된 알려진 취약점에 대한 후속 조치를 취할 수 있는 방법이 있습니다. • 당사는 보증이 필요한 경우 식별된 알려진 취약성을 고객 기반에 전달할 수 있는 방법을 가지고 있습니다. • 당사는 공급 소프트웨어의 릴리스 이후 새로 게시된 알려진 취약성에 대해 공급 소프트웨어를 분석하는 방법을 가지고 있습니다. • 당사는 출시 전에 공급된 모든 소프트웨어에 대해 지속적이고 반복적인 보안 테스트를 적용하는 방법을 가지고 있습니다. • 당사는 제공된 소프트웨어가 출시되기 전에 식별된 위험이 해결되었는지 확인할 수 있는 방법을 가지고 있습니다. • 당사는 식별된 위험에 대한 정보를 적절하게 제 3 자에게 내보낼 수 있는 방법을 가지고 있습니다.
4.2.1	• 당사는 제 3 자가 알려진 취약점 또는 새로 발견된 취약점에 대해 문의할 수 있는 방법을 가지고 있습니다(예: 프로그램 참가자가 모니터링하는 이메일 주소 또는 웹 포털을 통해). • 당사는 제 3 자의 알려진 취약성 또는 새로 발견된 취약성 문의에 응답하기 위한 문서화된 내부 절차를 가지고 있습니다.
4.2.2	• 우리는 프로그램과 관련된 사람, 그룹 또는 기능을 문서화했습니다. • 우리는 식별된 프로그램 역할에 적절한 인력이 배치되고 적절한 자금이 제공되었는지 확인했습니다. • 우리는 식별된 알려진 취약성을 해결하기 위해 사용 가능한 전문 지식을 보장했습니다. • 보안 보증에 대한 내부 책임을 할당하는 문서화된 절차가 있습니다.
4.3.1	• 당사는 제공된 소프트웨어에 사용된 모든 오픈소스 SW 가 제공된 소프트웨어의 수명 주기 동안 지속적으로 기록되도록 하는 문서화된 절차를 가지고 있습니다. 여기에는 제공된 소프트웨어에 사용된 모든 오픈 소스 소프트웨어의 아카이브가 포함됩니다. • 당사는 제공된 소프트웨어에 대한 오픈소스 구성 요소 기록을 보유하고 있으며, 이는 문서화된 절차가 적절하게 준수되었음을 보여줍니다.
4.3.2	• 당사는 제공된 소프트웨어의 오픈소스 SW 구성 요소에 대한 알려진 취약성의 탐지 및 해결을 처리하기 위한 문서화된 절차를 가지고 있습니다. • 당사는 제공된 소프트웨어에 대한 오픈소스 구성 요소 기록을 보유하고 있으며, 이를 통해 식별된 알려진 취약성 및 취한 조치(조치가 필요하지 않은 경우에도 포함)를 추적할 수 있습니다.
4.4.1	• 프로그램이 이 사양의 모든 요구 사항을 충족함을 확인하는 문서가 있습니다.
4.4.2	• 지난 18 개월 이내에 프로그램 적합성을 검토했음을 확인하는 문서가 있습니다.

* 출처: ISO/IEC 18974 온라인 자체인증 체크리스트

표 5. ISO/IEC 18974 온라인 자체 인증 체크리스트

2. 오픈소스 SW 관리체계 도입 검토 시 고려사항

1) 조직의 구성

오픈소스 SW 를 도입하는 기업들의 조직 및 서비스 목표는 매우 다양하다. 대외 서비스용과 내부 업무시스템에 따라 관리 목표가 다르며, 성공적인 결과를 위해 목표를 가장 잘 관리할 수 있는 조직에서 관리해야 한다. 그러나 조직으로만 관리체계 운영의 성공을 보장하기는 어렵다. 성공적인 관리체계 운영을 위해서는 각 조직에서 적극적으로 역할을 수행하는 것이 필요하며, 협의체 또는 의사소통 채널을 만들어 각 조직 간 협력이 이뤄져야 한다.

오픈소스 SW 를 내부 업무용으로 사용하는 경우, 일반적으로 폐쇄적인 환경에서 운영되기 때문에 보안 취약점이나 라이선스 이슈가 발생할 가능성은 상대적으로 낮다. 따라서 신속한 개발과 요구사항 반영이 중요한 이 환경에서는 오픈소스 SW 를 직접 다루는 개발 조직이 관리하는 것이 더 효과적일 수 있다. 반면, 외부 서비스용으로 오픈소스 SW 를 사용하는 경우에는 보안 취약점이 노출되어 직접적인 공격을 받을 위험이 높고, 프로그램 노출에 따른 라이선스 이슈가 발생할 수 있다. 이러한 경우에는 IT 기획 조직에서 관리체계를 운영하는 것이 적절하다.

서비스 유형	관리목표	조직	역할
내부 업무용	빠른 업무요구사항 반영	개발조직	관리체계 운영, SBOM 관리
		정보보호조직	CVE 취약점 관리
		법무조직	라이선스 관리
		IT 기획조직	검토
외부 서비스용	안정적 서비스 제공	IT 기획조직	관리체계 운영, SBOM 관리
		정보보호조직	CVE 취약점 관리
		법무조직	라이선스 관리
		개발조직	평판 관리

표 6. 서비스 유형별 관리 목표 및 조직 역할

2) 시스템 도입

소규모 서비스 조직의 경우, 오픈소스 SW 관리를 어디서부터 시작해야 할지 막막할 수 있다. 하지만 국내 여러 사이트에서 오픈소스 SW 와 관련된 다양한 정보를 제공하고 있으며, 보안 취약점이나 라이선스 문제에 대한 자료를 검색하면 충분한 도움을 받을 수 있다.

사이트명	URL	서비스
공개 SW 포털	www.oss.kr	• 오픈소스 SW 보안 취약점 정보 검색
오픈소스 SW 라이선스 종합정보시스템	www.olis.or.kr	• 오픈소스 SW 라이선스 정보 검색

표 7. 오픈소스 SW 정보를 제공하는 사이트

오픈소스 SW 가 포함된 서비스의 수가 많거나 이해관계자가 다양한 경우, 자동화된 관리시스템 도입이 필요할 수 있다. 오픈소스 SW 관리를 위한 몇 가지 자동화 시스템이 있으며 아래에 소개한 시스템 외에도 다양한 시스템이 있으니 여러분의 조직과 서비스에 맞는 시스템으로 선택하면 된다.

시스템 명	특징
Black Duck	• 오픈소스 SW 를 사용하는 동안 발생하는 라이선스와 취약점, 소스 코드 품질 관리가 가능한 포괄적인 솔루션 • 소프트웨어 공급망과 애플리케이션 라이프사이클 전반에 걸쳐 오픈소스 SW 의 라이선스와 보안 관리 가능
LABRADOR	• 오픈소스 SW 의 구성요소를 담은 SBOM 제공으로 소프트웨어 공급망 보안 관리 지원 • 오픈소스 SW 의 라이선스 및 취약점 리스크를 탐지하고 패치 할 수 있는 소프트웨어 안전관리 플랫폼

FOSSID	• 오픈소스 라이선스 및 보안취약점 관리 솔루션으로서 소스 코드 내 컴포넌트를 탐지하여 각 컴포넌트의 라이선스 및 보안취약점을 식별 • 방대한 오픈소스 데이터베이스 및 자동 데이터 수집 기술, AI 를 통한 향상된 탐지 성능 등을 제공
White Source	• 방대한 데이터베이스에 기반을 둔 라이선스 준수 및 취약점 관리 서비스를 제공하며, 컨테이너 및 서버리스 등 다양한 환경을 지원
Sparrow SCA	• 오픈소스 SW 라이선스 식별 및 보안취약점 진단 도구 • 소스코드, 바이너리 파일 분석 및 오픈소스 SW 소스코드 일부만 가져오는 스니팻 분석 지원 기능 제공

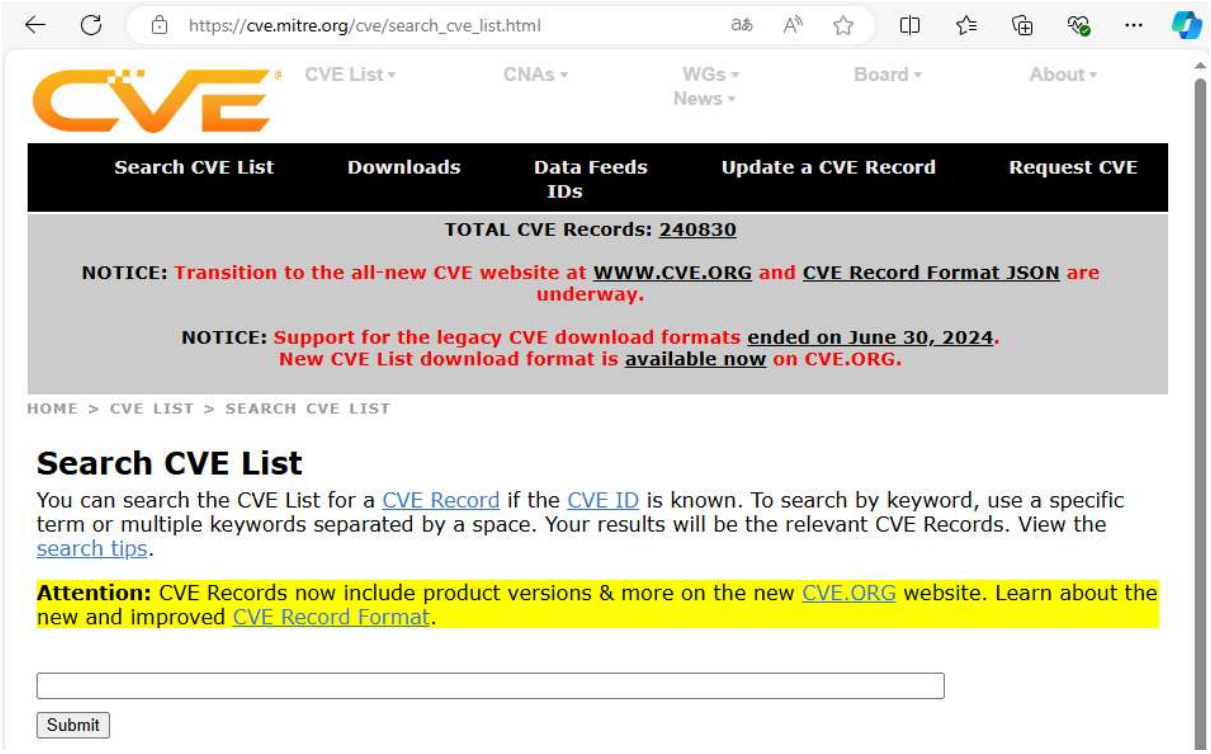
* 출처: 금융감독원 금융분야 오픈소스 소프트웨어 활용·관리 안내서(2022.12)

표 8. 오픈소스 SW 자동화 관리 시스템

3) SBOM 관리 항목

A. 보안취약점 관리를 위한 항목 선정 및 활용 (CVE 검색 방법)

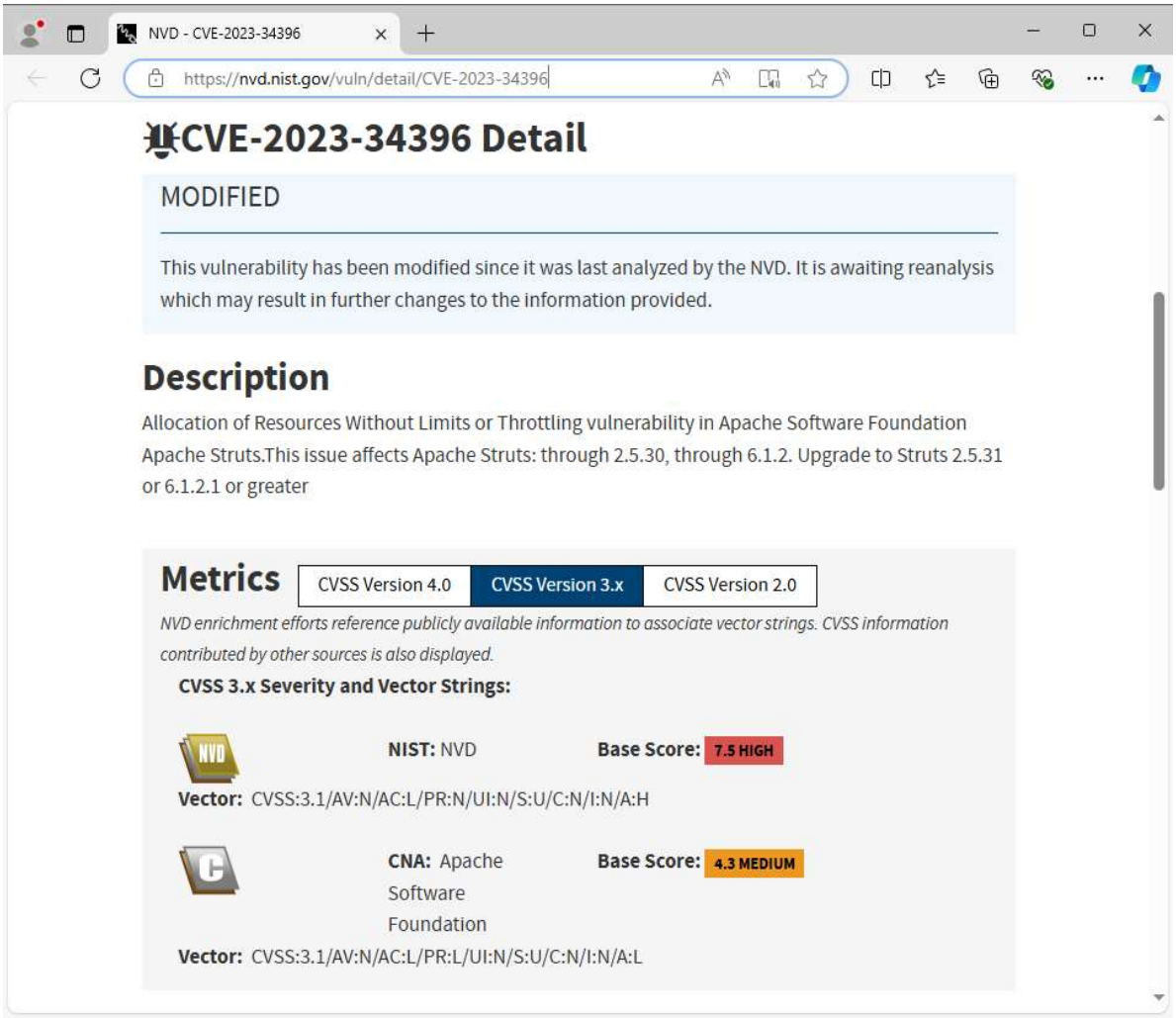
오픈소스 SW 의 기술적 취약성을 확인하는 것이 가장 중요한 업무이지만, 실제 해당 취약점으로 인한 사고가 발생하지 않으면 조치 방법이 어려워 이를 간과하거나 인지하지 못하고 지나가는 경우도 있다. 보안취약점 발생 여부를 점검하는 가장 쉬운 방법은 미국 비영리 단체인 MITRE 에서 CVE(Common Vulnerabilities and Exposures)를 검색하는 것이다.



* 출처: MITRE

그림 6. CVE 검색 페이지

검색 결과에서 CVE ID를 클릭하면 취약점 개요와 참조 링크가 제공된다. 참고로 미국 국가 취약성 데이터베이스(NVD, National Vulnerability Database) 링크를 통해 해당 취약점에 대한 공통 취약점 등급 시스템(CVSS, Common Vulnerability Scoring System) 점수를 확인할 수 있으며, 해당 보안 취약점의 심각성을 평가할 수 있다.



* 출처: NIST

그림 7. CVSS 검색 페이지

Key Point - 보안취약점 관리를 위한 SBOM 항목: 이름, 버전정보

B. 라이선스 관리를 위한 항목 선정 및 활용 (라이선스별 조치 방법)

라이선스는 기술적인 보안 취약점 항목은 아니다. 그러나 관리가 되지 않을 경우, 보안사고 수준의 금전적 손실이 발생할 수 있는 중요한 항목이다. 일반적으로 라이선스 정보는 해당 오픈소스 SW를 개발한 단체의 홈페이지에 게시되어 있다. 오픈소스 SW 라이선스의 종류와 사용조건은 매우 다양하기

때문에 정확한 라이선스를 식별해야 한다. 또한, 특정 라이선스의 경우 해당 오픈소스와 결합된 모든 소스코드의 공개를 요구해야 하는 조건이 있어 사용결정에 대해 신중해야 한다.

분류	라이선스 종류	주요 사용 조건
Permissive (허용)	Apache-2.0, BSD-2-Clause, BSD-3-Clause MIT	• 저작권 표시, 라이선스 고지
Weak Copyleft (약한 제약)	LGPL-2.1, LGPL-3.0, EPL-2.0, MPL-2.0	• 오픈소스가 사용된 해당 소스코드 부분 공개
Copyleft (강한 제약)	GPL-2.0, GPL-3.0	• 오픈소스와 결합된 모든 소스 코드 공개

출처: 과학기술정보통신부 SW 공급망 보안 가이드라인(2024.05)

표 9. 라이선스 분류 및 주요 사용조건

Key Point - 라이선스 관리를 위한 SBOM 항목: 라이선스 종류 명

C. 평판 관리를 위한 항목 선정 및 활용

기술적인 관점에서 벗어나, 사용하는 오픈소스 SW 가 신뢰할 수 있고 지속적으로 사용해도 괜찮은지를 평가하는 것은 주관적인 영역에 속한다.

우리가 일상생활에서 상품을 구매할 때 일반적으로 유명 제조사에 대한 신뢰, 널리 사용되는 제품, 고장이 거의 없는 제품, 지속적인 AS 여부 등을 기준으로 선택한다. 오픈소스 SW 를 선택할 때도 이와 유사한 기준을 적용할 수 있다. 즉, 유명 IT 업체나 재단에서 관리하는 소프트웨어, 많은 개발자와 활발한 커뮤니티가 지원하는 소프트웨어, 신뢰할 만한 국가에서 관리하는 소프트웨어 등을 기준으로 삼아 선택하는 것이 바람직하다.

Key Point - 평판 관리를 위한 SBOM 항목: 제조사, 국가

4) 위험 평가

현실적으로 모든 보안 취약점을 완벽하게 조치하는 것은 불가능하다. 오픈소스 SW 를 어느 정도의 수준에서 관리할 것인지 객관적으로 판단하고 결정하기 위해서는 보안 취약점에 대한 위험평가 기준을 마련해야 한다.

앞에서 관리하기로 정한 보안취약점, 라이선스, 평판 관리를 위해서 다음과 같이 취약도 평가기준을 마련하여 관리할 수 있다.

평가항목	평가 기준 예시	비고
보안취약점	CVE 가 존재하며 CVSS 9.0 ~ 10.0 사이	CVSS v3.x 계산결과에 따른 심각도 분류 기준 "Low", "Medium", "High", "Critical"에 따라 기준 수립
	CVE 가 존재하며 CVSS 7.0 ~ 8.9 사이	
	CVE 가 존재하며 CVSS 4.0 ~ 6.9 사이	

	CVE 가 존재하며 CVSS 0.1 ~ 3.9 사이	(출처: https://nvd.nist.gov)
라이선스	Copyleft(강한 제약) 라이선스 사용 조건	라이선스 사용 조건 충족을 위한 조치의 기술적 난이도에 따라 기준 수립
	Weak Copyleft(약한 제약) 라이선스 사용 조건	
	Permissive(허용) 라이선스 사용 조건	
평판	관리 주체/개인 등을 신뢰할 수 없는 경우	관리하는 주체의 규모, 커뮤니티 접근성, 신뢰도 등에 따라 기준 수립
	관리 단체는 없으나 커뮤니티가 활성화되어 있는 경우	
	글로벌 IT 기업, IT 재단 및 주요 국내 IT 기업 등에서 관리하는 경우	

표 10. 취약도 평가 기준 예시

각 기업이 수립하고 있는 위험평가 방법론에 따라 취약도 평가기준에 대한 점수 산정 및 위험도 산출 공식을 선정하면 일관성 있게 위험평가를 수행할 수 있다.

모든 위험을 조치하는 것은 현실적으로 어려움이 있다. 기본적인 위험조치 방법론에 따라 조치 가능한 위험을 정하기 위해 감당할 수 있는 수용가능위험수준(DoA, Degree of Assurance)을 선정하고 이를 초과하는 위험에 대해 조치한다면 보다 효과적인 위험조치가 가능하다.

일반적으로 위험조치계획을 위험감소, 위험전가, 위험회피, 위험수용 등 4 가지 유형으로 분류하여 관리한다. 위험조치를 위해 각 취약점 별 조치 방안을 수립한다.

조치 방안	보안취약점	라이선스	평판
위험감소	• 보안 패치 적용	• 라이선스 사용조건 반영 -저작권 표시, 관련소스 공개 등	• 글로벌 IT 기업, IT 재단에서 관리하는 오픈소스로 교체
위험회피	• 관련취약점이 없는 오픈소스 SW 로 교체	• Permissive(허용) 조건의 오픈소스로 교체 • 상용 소프트웨어로 대체	• 글로벌 IT 기업, IT 재단에서 관리하는 오픈소스로 교체
위험전가	• 보안사고 관련 보험 가입	• 보안사고 관련 보험 가입	• 보안사고 관련 보험 가입
위험수용	• 보완 통제 적용(예: 방화벽)	-	-

표 11. 오픈소스 SW 위험 조치 방안

5) 오픈소스 SW 저장소 관리의 필요성

오픈소스 SW 는 인터넷만 있으면 어디서든 다운로드할 수 있지만, 공식 사이트가 아닌 출처에서 파일을 다운로드할 경우에는 항상 의심해야 한다. 악성코드가 포함된 오픈소스 SW 를 개발 프로젝트에 사용하면, 그 자체로 취약점이 발생할 수 있기 때문이다. 오픈소스 SW 를 안전하게 사용하기 위해서는 저장소를 구축하고 비인가자의 접근을 통제하며, 파일 반입에 대한 절차를 엄격히 적용해야 한다. 또한, 오픈소스 SW 에 대한 무결성 검증 값(HASH)을 SBOM 에 추가로 관리하면 파일의 위·변조에 대비할 수 있다.

그림 8. 오픈소스 저장소 반입 흐름도

■ 맺음말

현대 사회의 모든 산업에서 IT 기술은 더 이상 단순한 보조 수단이 아니라 가치 창출을 위한 핵심 요소로 자리잡고 있다. AI 의 발전 덕분에 사람의 고유 영역으로 여겨지던 예술과 감정 분야까지 IT 기술이 접목되고 있다. 이제 IT 보안 사고가 발생하면, 단순히 한 조직이나 서비스에 금전적 손실을 미치는 것이 아니라 전 세계적인 영향을 미친다.

SBOM(소프트웨어 자재 명세서) 관리는 IT 보안 사고를 완전히 예방할 수는 없지만, 사고 발생 후 회복을 위한 가장 빠르고 합리적인 방법이라고 확신한다. 보안이나 IT 기술에 대한 전문적인 지식이 없어도, 현재 우리 회사가 어떤 오픈소스 SW 를 사용하고 있는지 확인하는 것으로 시작할 수 있다. 결심한 후에는 SBOM 을 도입하겠다는 결정을 내리면 된다.

SK 쉴더스는 오픈소스 SW 관리 체계 구축과 관련한 컨설팅을 제공하고 있다. 자세한 내용은 SK 쉴더스 홈페이지나 문의하기를 통해 확인할 수 있다.