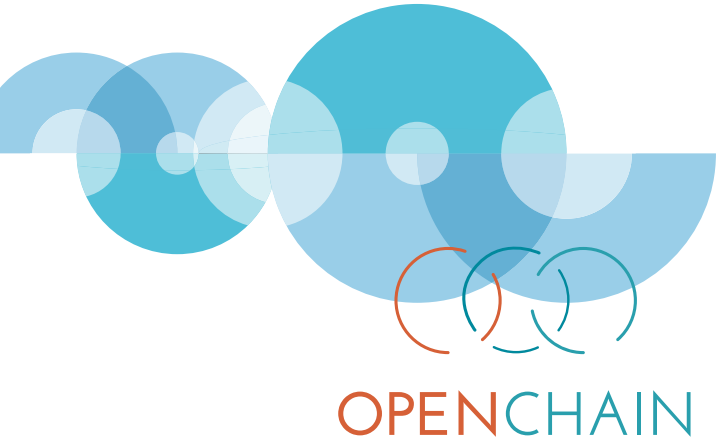


nipa 정보통신산업진흥원
National IT Industry Promotion Agency
27872 충북 진천군 덕산면 정통로 10
http://www.nipa.kr

기업 공개SW 거버넌스 가이드

OpenChain 2.0 해설

nipa 정보통신산업진흥원
National IT Industry Promotion Agency

기업 공개SW 거버넌스 가이드

OpenChain 2.0 해설

과학기술정보통신부
Ministry of Science and ICT

nipa 정보통신산업진흥원
National IT Industry Promotion Agency

기업 공개SW 거버넌스 가이드

OpenChain 2.0 해설

- **주관연구기관** : 정보통신산업진흥원
  송주영 본부장
  한복미 팀장
  김진기 책임

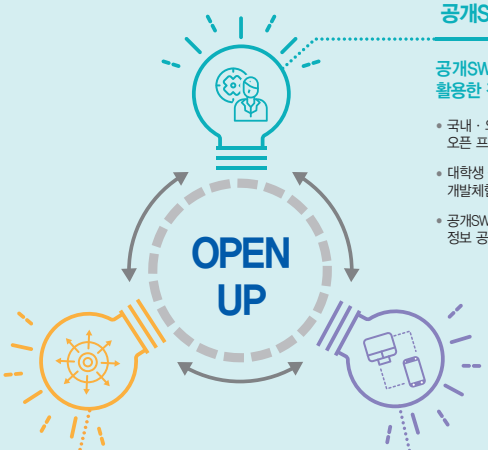
- **연구수행기관** : 공개SW역량프라자
  김병선 전문위원
  김혜영 부장

- **외부전문위원** : LG전자 장학성 책임
  SKT 박철웅 부장

정보통신산업진흥원 공개SW 사업

정보통신산업진흥원의 Open UP은 기업, 개발자, 커뮤니티, 공공기관을 대상으로 공개SW의 활용 · 기술 지원, 인력 양성, 창업 지원, 저변 확대 등 공개SW 통합 지원 역할을 수행하고 있습니다.

오픈소스로 성장하는 Open UP



공개SW 개발자 지원

공개SW 참여, 공유, 협업 문화를 활용한 전문인력 양성 및 저변확대

- 국내 · 외 공개SW프로젝트를 리딩하는 오픈 프런티어(전문개발자) 발굴 및 지원
- 대학생 등 예비 개발자 대상 공개SW 개발체험 및 재직자 대상 전문 교육 제공
- 공개SW 개발 · 커뮤니티 간의 기술교류 및 정보 공유를 위한 기회 · 환경 제공

공개SW 기업 지원

공개SW 기반 최신 기술개발, 안전한 활용, 원스톱 창업 지원

- AI, Bigdata, Cloud등 최신 기술 분야 공개SW기술개발 지원 및 기업육성
- 안전한 공개SW 활용을 위한 라이선스/보안 검증, 컨설팅, 기업 육성
- 공개SW 기반 스타트업, 예비창업자 대상 기술개발, 사업화, 글로벌 진출 지원

공공 공개SW 활용 지원

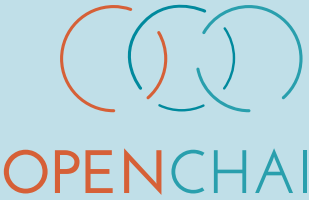
공공부문 공개SW도입 및 개방형 OS 선도 적용을 위한 기술지원 서비스 제공

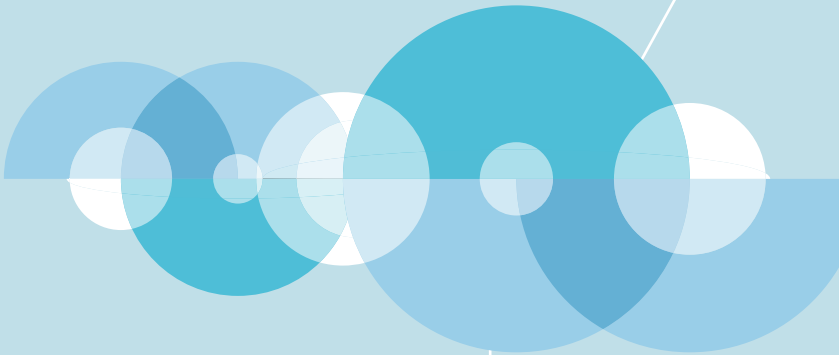
- 공공부문 정보화사업 공개SW 적용사례 발굴 및 전환 컨설팅, 교육지원
- 정부 '디지털 정부혁신 추진계획'에 따른 개방형 OS 선도적용 기술 지원

과학기술정보통신부 Ministry of Science and ICT
nipa 정보통신산업진흥원 National IT Industry Promotion Agency
T. 02.561.0551
H. http://www.oss.kr

OPENCHAIN

목 차





Ⅰ. OpenChain Project란?

1. OpenChain Specification
2. OpenChain Conformance (준수)
3. OpenChain Curriculum

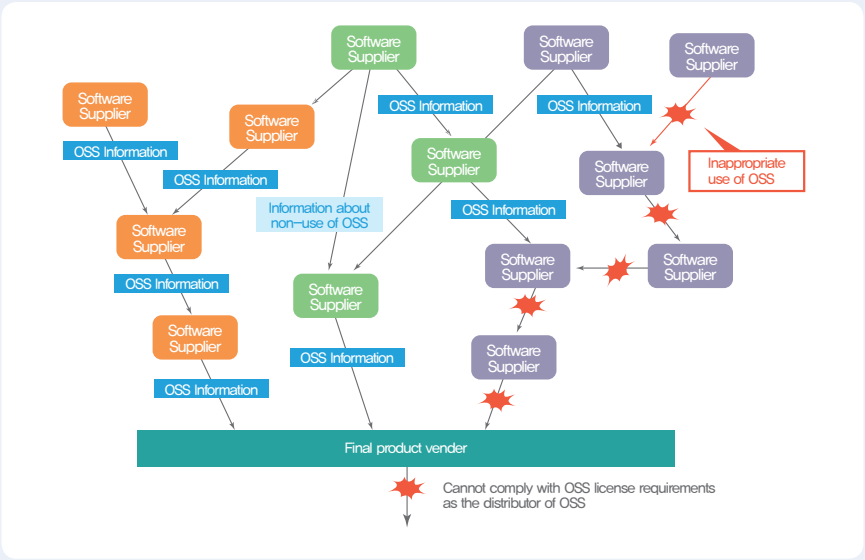
OPENCHAIN

Ⅰ. OpenChain Project란?

오늘날 소프트웨어는 갈수록 그 규모와 복잡도가 커지고 있다. 하나의 소프트웨어를 개발하기 위해서는 자체 개발하는 소프트웨어뿐 아니라 오픈소스, 3rd party Software, 반도체 벤더의 SDK 등 소프트웨어 공급망에 걸친 다양한 소프트웨어가 사용될 수 있기 때문이다.

이러한 복잡한 소프트웨어 공급망의 조직 중 한 곳이라도 라이선스 의무를 준수하지 않거나, 올바른 오픈소스 정보를 제공하지 못한 경우, 최종 소프트웨어를 배포하는 기업은 라이선스 준수에 실패하고 이로 인해 제품 판매가 중단되는 상황이 발생할 수 있다. 실제로 2009년 12월, Busybox라는 오픈소스 관련된 소송이 있었다. Busybox는 임베디드 시스템에 광범위하게 사용되고 있는 GPL-2.0 라이선스가 적용된 오픈소스인데, 두 곳의 국내 회사를 포함하여 총 14개 회사가 소송 대상이 되었다. 이 사례에서 주목할만한 점은 이 중에는 제품을 직접 개발하지 않고 배포만 한 회사도 소송을 당했다는 점이다.

이와 같은 복잡한 소프트웨어 공급망 환경에서는 어느 한 기업이 아무리 훌륭한 프로세스를 갖추고 있다고 해도 자체적으로 완벽한 오픈소스 컴플라이언스를 달성하는 건 매우 어렵다. 결국 소프트웨어를 최종 배포하는 기업이 오픈소스 컴플라이언스를 제대로 이행하기 위해서는 소프트웨어 공급망의 모든 구성원이 라이선스 의무를 준수하고 올바른 오픈소스 정보를 제공하여 공급망 전체에 신뢰가 구축되어야 한다.



[OpenChain Open Source Software License Compliance General Public Guide]
(https://www.openchainproject.org/resources)

Linux Foundation의 OpenChain 프로젝트는 기업이 오픈소스 컴플라이언스를 위해 준수해야 할 활동을 더 간단하고 일관성 있게 만들어 소프트웨어 공급망 전체에 신뢰를 구축할 수 있도록 해준다.

OPENCHAIN

(https://www.openchainproject.org/)

2016년 유럽의 한 오픈소스 콘퍼런스에서 퀄컴의 오픈소스 변호사인 데이브 머(Dave Marr)는 한 기업의 오픈소스 컴플라이언스 수준을 높이기 위해서는 소프트웨어 공급망 내의 모든 구성원이 오픈소스 컴플라이언스 수준을 높이는 것이 중요함을 강조한 바 있다.

아울러 이를 위해서는 오픈소스를 충분히 이해하고, 정책 및 프로세스를 앞서 구축하고 있는 기업들이 자신들의 자산과 노하우를 공개해 누구나 이를 참고할 수 있게 해야 한다는 의견을 제시했다. 콘퍼런스 참석자들은 "오픈소스 컴플라이언스는 기업의 이익을 차별화할 수 있는 분야가 아니다. 기업은 최소한의 리소스를 투입하여 적정한 수준의 리스크 관리를 원하기 때문에 기업들이 가진 자산을 공유하면 할수록 적은 비용으로 모두 함께 컴플라이언스를 달성 할 수 있다" 는 아이디어에 공감했다. OpenChain 프로젝트(당시에는 Work Group)는 그렇게 시작됐고, Qualcomm, Siemens, Wind River, ARM, Adobe 등 다수 글로벌 기업들이 참여했다.

1 OpenChain Specification

OpenChain 프로젝트는 곧 OpenChain Specification 1.0을 제작하여 배포했다. OpenChain Specification은 오픈소스 컴플라이언스를 위한 핵심 요구사항을 정의한 12페이지 분량의 표준 규격으로, 기업의 규모나 업종에 관계없이 모든 분야의 회사에 적합하도록 고안되었다. 2019년 4월에는 버전 2.0의 Specification이 배포됐으며, 기업이 오픈소스 컴플라이언스 달성을 위해 꼭 수행해야 할 여섯 가지 주요 요건에 대한 설명과 이를 수행하고 있음을 입증하기 위한 검증 자료 목록을 정의하고 있다.

1. 오픈소스 컴플라이언스를 관리하기 위한 프로그램
2. 효과적인 리소스 제공을 위한 업무 정의 및 지원
3. 오픈소스 검토 및 승인을 관리하는 프로세스
4. 컴플라이언스 결과물 생성 및 제공을 위한 프로세스
5. 오픈소스 커뮤니티 참여를 이해하고 관리하기 위한 정책
6. OpenChain Specification 요건 준수

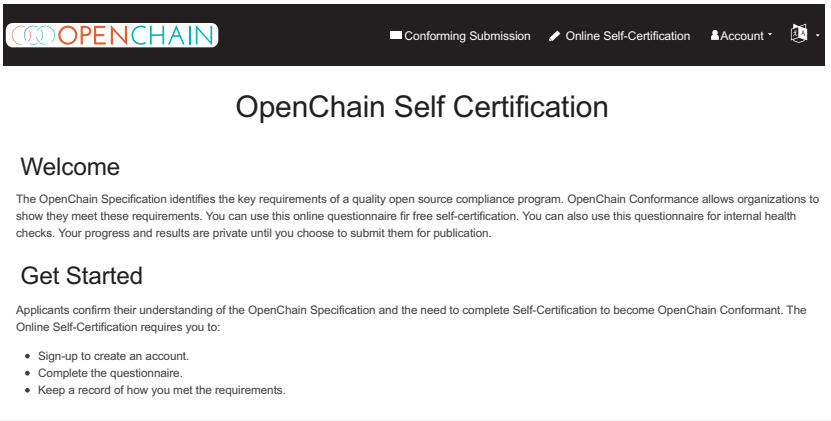
오픈소스 컴플라이언스를 처음 시작하는 기업이라면 이와 같은 OpenChain Specification의 요건을 하나씩 충족해가면서 수준을 향상시키는 것이 좋은 전략이다.

Contents

(https://wiki.linuxfoundation.org/_media/openchain/openchainspec-2.0.pdf)

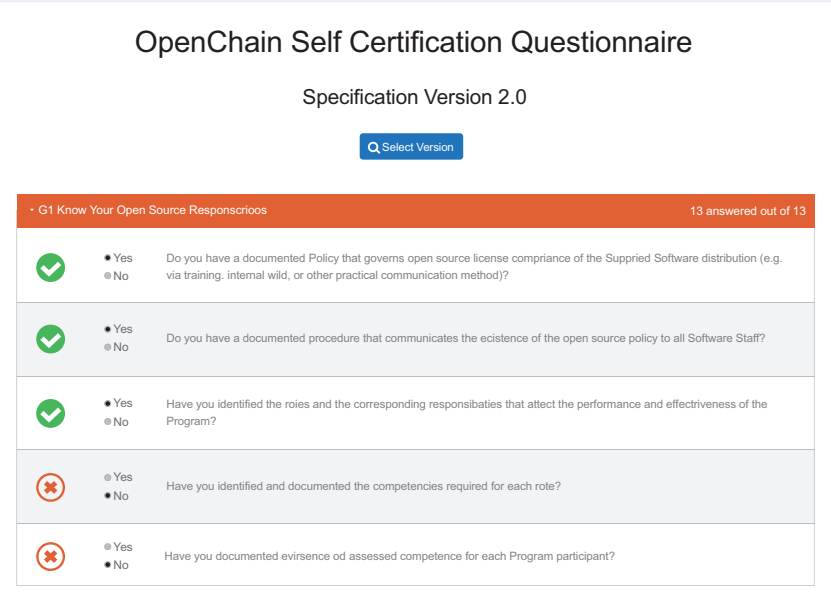
2 OpenChain Conformance (준수)

OpenChain Project는 기업이 OpenChain Specification을 충족하는지 자체적으로 확인할 수 있도록 온라인 자체 인증 웹사이트를 제공한다.



(https://certification.openchainproject.org/)

기업의 오픈소스 담당자는 OpenChain 자체 인증 웹사이트에 가입해 온라인 자체 인증을 시작할 수 있으며, Yes/No 질문에 답변하는 방식으로 진행된다.



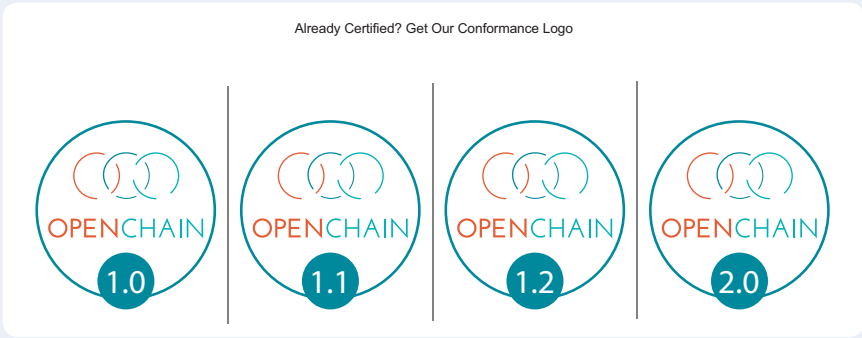
(https://certification.openchainproject.org/)

자체 인증을 통해 부족한 부분이 무엇인지, 추가로 필요한 활동이 무엇인지 판단할 수 있다.

만약, 오픈소스 컴플라이언스 체계가 잘 구축된 기업이 OpenChain 자체 인증 질문지의 모든 항목을 Yes로 대답할 수 있다면 이 결과를 웹사이트상에 제출할 수 있다(Conforming Submission). 그러면 OpenChain 준수(Conformant) 기업으로 인정됨과 동시에, OpenChain 프로젝트의 웹사이트에서 OpenChain 준수 프로그램을 갖춘 기업 목록에 기업의 로고를 등록할 수 있게 된다.

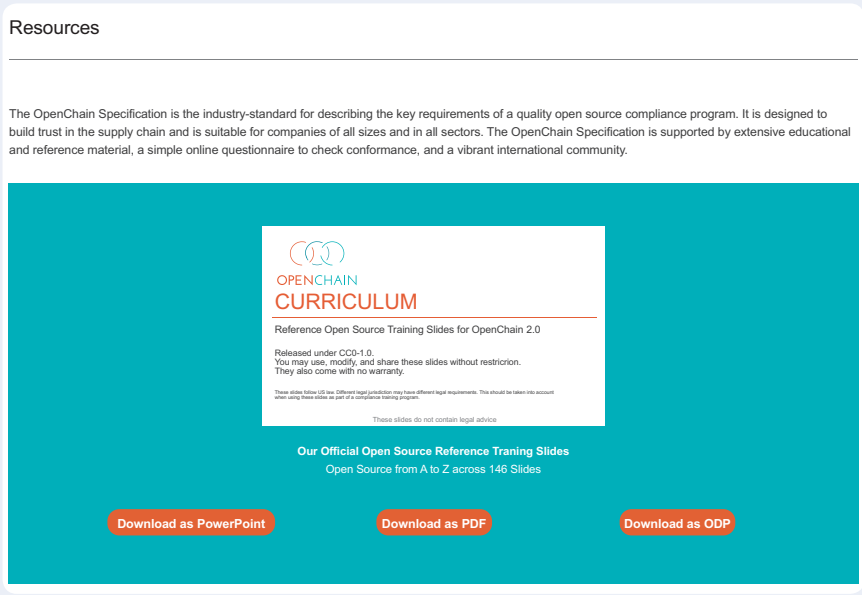
[Organizations with a publicly announced OpenChain Conformant Program]

OpenChain 준수 기업에게는 OpenChain 로고를 사용할 수 있는 자격이 주어진다. 이렇게 OpenChain 준수 프로그램을 갖췄다고 인정받은 기업은 소프트웨어 공급망 내에서 오픈소스 컴플라이언스를 충실하게 수행하고 있음을 보여줄 수 있다.



3 OpenChain Curriculum

OpenChain 프로젝트에서는 기업이 컴플라이언스 프로그램을 구축하는데 필요한 정책 문서 템플릿, 교육 자료 등 다양한 참고자료를 제공한다. 이 자료들은 OpenChain Specification 및 일반적인 오픈소스 컴플라이언스 활동을 지원하기 위해 고안됐으며, 누구나 자유롭게 사용할 수 있도록 Public Domain으로 제공된다.



(https://www.openchainproject.org/resources)

OPENCHAIN

II. OpenChain Specification 준수 방법

1. 프로그램 설립 (Program Foundation)
2. 관련 업무 정의 및 지원 (Relevant Tasks Defined and Supported)
3. 오픈소스 콘텐츠 검토 및 승인 (Open Source Content Review and Approval)
4. 컴플라이언스 결과물 생성 및 전달
5. 오픈소스 커뮤니티 참여에 대한 이해
6. 설명서 요건 준수

OPENCHAIN

Ⅱ. OpenChain Specification 준수방법

OpenChain Specifiation에서는 오픈소스 컴플라이언스를 위한 핵심 요구 사항을 정의한다. OpenChain Specification을 준수한다고 인정받은 기업은 소프트웨어 솔루션을 배포하는 조직간에 신뢰를 제공할 수 있게 된다. 여기에서는 기업들이 OpenChain Specification을 준수하기 위해 충족해야 하는 여섯가지 주요 요건과 그 방법을 세부적으로 설명한다.

1 프로그램 설립 (Program Foundation)

1) 1.1 정책 (Policy)

오픈소스를 이용하여 소프트웨어를 개발하고 배포하는 기업이라면 오픈소스를 관리하기 위한 정책과 프로세스를 구축하고, 이를 위한 인력과 자원을 할당해야 한다. OpenChain에서는 이러한 일련의 활동을 관리하는 체계를 오픈소스 프로그램이라고 부르고, OpenChain Specification을 준수하기 위한 첫번째 요건은 바로 이 프로그램을 설립해야 하는것이다. 여기서 오픈소스 프로그램이란 정책, 프로세스, 인원 등 기업이 오픈소스 컴플라이언스 활동을 수행하기 위한 일련의 관리 체계를 의미한다.

OpenChain Specification에서는 이를 입증하기 위한 자료로 우선 문서화된 오픈소스 정책을 요구한다. 이 안내서에서는 참고를 위해 OpenChain Specification의 요건을 충족하는 오픈소스 정책 문서 예시를 “[부록 01] 오픈소스 정책 for OpenChain 2.0 (예시)” 에서 제공한다. OpenChain Specification은 이어지는 장에서 오픈소스 프로그램이 갖춰야할 요건들을 설명하고 있다.

OpenChain Specification

1.1 정책

공급 대상 소프트웨어의 오픈소스 라이선스 컴플라이언스를 관리하는 문서화 된 오픈소스 정책이 존재한다. 정책은 내부적으로 전달되어야 한다.

입증 자료:

1.1.1 문서화된 오픈소스 정책

1.1.2 소프트웨어 공급 담당자가 오픈소스 정책의 존재를 인식하도록 하는 문서화 된 절차 (교육, 내부 위키, 혹은 기타 실질적인 의사소통 방법 등)

1.1 Policy

A written Open Source policy exists that governs Open Source license compliance of the Supplied Software. The policy must be internally communicated.

Verification Material(s):

1.1.1 A documented Open Source policy

1.1.2 A documented procedure that makes Software Staff aware of the existence of the Open Source policy (e.g., via training, internal wiki, or other practical communication method)

오픈소스 프로그램은 소프트웨어 공급담당자가 이 오픈소스 정책 문서의 존재를 알고, 필요한 활동을 할 수 있도록 교육, 내부 위키 등 실질적인 수단을 제공해야 한다. 여기서 소프트웨어 공급담당자(Software Staff)란 기업이 소프트웨어를 개발하고 배포, 기여하는데 관여하는 모든 직원을 의미하며, 소프트웨어 개발자, 배포 엔지니어, 품질 엔지니어 등을 포함한다.

많은 기업들은 오픈소스 정책 문서를 사내 위키 사이트를 통해 공개하여 직원 누구나 필요한 사항을 확인할 수 있게 한다. 또한, 신규 채용인원의 입사 연수 시 오픈소스 정책에 대한 교육을 의무화하고, 소프트웨어 공급담당자를 대상으로 매년 혹은 2년에 한번씩 주기적인 교육을 제공함으로 모든 소프트웨어 공급담당자가 오픈소스 정책의 존재를 인식하도록 할 수 있다. 이러한 방법들을 오픈소스 정책 문서에 구체화하여 포함시켜야 한다.

2) 1.2 역량 (Competence)

OpenChain Specification

1.2 역량

조직은 다음 사항을 수행해야 한다: (The organization shall:)
- 프로그램의 성능 및 효과에 영향을 미치는 역할과 해당 역할에 대한 책임을 확인한다;
- 각 역할을 수행하는 인원의 필요한 역량을 파악한다;
- 해당 인원이 적절한 교육, 훈련 및 경험을 바탕으로 자격을 갖춘 자임을 보장한다;
- 해당되는 경우, 필요한 역량을 확보하기 위한 조치를 취한다;
- 적절히 문서화된 정보를 역량의 증거로 보유한다.

입증 자료:
1.2.1 프로그램 내 여러 참여자에 대한 문서화된 책임과 역할 목록
1.2.2 각 역할에 대한 역량을 확인하는 문서
1.2.3 각 프로그램 참여자에 대해 역량을 평가한 문서화된 증거

1.2 Competence

The organization shall:
- Identify the roles and the corresponding responsibilities of those roles that affects the performance and effectiveness of the Program;
- Determine the necessary competence of person(s) fulfilling each role
- Ensure that these persons are competent on the basis of appropriate education, training, and/or experience;
- Where applicable, take actions to acquire the necessary competence; and
- Retain appropriate documented information as evidence of competence.

Verification Material(s):
1.2.1 A documented list of roles with corresponding responsibilities for the different participants in the Program.
1.2.2 A document that identifies the competencies for each role.
1.2.3 Documented evidence of assessed competence for each Program participant.

오픈소스 프로그램이 올바르게 구축되고 운영될 수 있도록 역할과 책임(R&R)을 정의해야 한다. 각 역할을 수행할 담당자가 갖춰야 할 역량을 정의하고, 지정된 담당자가 해당 역할을 수행할 수 있는 역량을 갖추었는지 파악해야 한다. 해당 인원이 교육, 훈련 및 경험을 바탕으로 맡은 역할을 수행할 수 있는 자격을 갖추었음을 보장해야 한다. 이를 위해 각 인원이 필요한 역량을 갖출 수 있도록 교육을 제공한다.

이를 입증하기 위해 기업은 프로그램 내 여러 참여자에 대한 책임 및 역할 목록과 각 역할을 수행하는 담당자가 갖춰야할 역량을 정의하여 문서화 한다. 이 안내서에서는 참고를 위해 오픈소스 프로그램의 각 참여자의 역할과 책임 및 필요한 역량을 정의한 샘플 문서를 "[부록 01] 오픈소스 정책 for OpenChain 2.0(예시)의 4. 역할, 책임 및 역량" 에서 제공한다.

그리고, 기업은 각 참여자가 역량을 갖추고 있는지 평가하고, 이를 보관한다. 이를 위해 기업은 각 참여자들이 필요한 역량을 보유할 수 있도록 교육을 제공한다. 교육 내용을 기반으로 평가하고, 그 결과는 기업의 교육 시스템 혹은 HR 부서에서 보관해야 한다. 소프트웨어 공급담당자가 수천명 이상이어서 교육 제공이 쉽지 않을 경우, 기업의 온라인 교육과 평가 시스템을 이용하는 것도 좋은 방법이다.

OPENCHAIN

CURRICULUM

Reference Open Source Training Slides for OpenChain 2.0

Released under CC0-1.0.
You may use, modify, and share these slides without restricrion.
They also come with no warranty.

These slides follow US law. Different legal jurisdiction may have different legal requirements. This should be taken into account when using these slides as part of a compliance training program.

These slides do not contain legal advice

* 원문(English) : https://github.com/OpenChain-Project/Reference-Material/raw/master/Training-Slides/Official/en/2.0/openchain-curriculum-for-2.0-en.pptx
* 한국어 번역 : https://github.com/OpenChain-Project/Reference-Material/raw/master/Training-Slides/Official/en/2.0/openchain-curriculum-for-2.0-en.pptx

OpenChain Project에서는 참고용 교육 자료와 평가 문제를 제공하고 있어서 이를 참고하여 각 기업에 맞는 교육 자료를 만들 수 있다.

3) 1.3 인지도 (Awareness)

OpenChain Specification

1.3 인지도

조직은 프로그램 참여자가 다음 사항을 알고 있음을 보장해야 한다:

a) 오픈소스 정책;

b) 오픈소스 관련 목표;

c) 프로그램의 효과에 대한 기여;

d) 프로그램의 요건 미준수의 의미.

입증 자료:

1.3.1 각 프로그램 담당자에 대해 프로그램의 목표, 프로그램에 기여, 그리고 프로그램 미준수의 의미를 포함하는 인지도를 평가한 문서화된 증거.

1.3 Awareness

The organization shall ensure that Program participants are aware of:

a) The Open Source policy;

b) Relevant Open Source objectives;

c) Their contribution to the effectiveness of the Program; and

d) The implications of not following the Program's requirements.

Verification Material(s):

1.3.1 Documented evidence of assessed awareness for each Program personnel including the Program's objectives, ones contribution within the Program, and implications of Program non-conformance.

프로그램 참여자가 오픈소스 정책, 기업의 오픈소스 관련 목표, 오픈소스 프로그램이 효과적일 수 있도록 참여자의 기여 방법, 그리고 프로그램 요건을 준수하지 않았을때의 발생할 수 있는 위험에 대해 인식하도록 한다.

이를 위해 오픈소스 정책은 프로그램 참여자가 오픈소스 정책 등의 주요 내용을 인식할 수 있도록 다음의 내용을 포함해야 한다.

- 먼저, 오픈소스를 사용, 배포, 기여하는 일련의 활동을 수행하는 목표를 포함한다. 예를 들어, "오픈소스를 이용하여 제품을 만들때 오픈소스 컴플라이언스 리스크를 최소화하고, 오픈소스 커뮤니티에 참여하고 기여함으로 최고의 가치를 창출한다"와 같은 형태로 목표를 수립할 수 있다.
- 그리고, 프로그램 참여자들이 자신의 역할에 대한 책임을 완수함으로써 오픈소스 프로그램의 효과가 증대될 수 있음을 알린다.
- 또한, 오픈소스 프로그램의 요건들을 준수하지 않았을 때 어떠한 위험이 발생하는지에 대해서도 알린다.

대표적인 위험 요소는 다음과 같다.

- 사용한 코드의 저작권자로부터 법적 클레임
- 의도하지 않은 기업 독점 코드의 공개
- 라이선스 의무 위반으로 인한 벌금
- 평판 손실
- 수익 손실
- 공급업체 및 고객과의 계약 위반

각 프로그램 담당자가 프로그램의 목표, 프로그램에 기여 방법, 프로그램 미준수의 의미에 대해 올바르게 인식할 수 있도록 교육을 제공하고, 이를 평가한다. 평가한 결과는 문서화하여 보관한다. 1.2장에서 언급한 교육 및 평가 시 이에 대한 내용을 포함하면 될 것이다.

4) 1.4 프로그램 적용 범위 (Program Scope)

OpenChain Specification

1.4 프로그램 적용 범위

서로 다른 프로그램들은 서로 다른 수준의 범위까지 적용될 수 있다. 예를 들어, 하나의 프로그램이 하나의 제품 라인, 전체 부서 또는 전체 조직을 관리할 수 있다. 각 프로그램별로 범위 지정이 이루어질 필요가 있다.

입증 자료:

1.4.1 프로그램의 적용 범위와 한계를 명확하게 정의한 문서화된 진술.

1.4 Program Scope

Different Programs may be governed by different levels of scope. For example, a program could govern a single product line, an entire department or an entire organization. The scope designation needs to be declared for each Program.

Verification Material(s):

1.4.1 A written statement that clearly defines the scope and limits of the Program.

오픈소스 프로그램은 반드시 기업 전체에 적용해야 하는 것은 아니다. 기업 내 각 조직의 특성에 따라 프로그램의 적용 범위를 달리 할 수 있다. 예를 들어, 소프트웨어를 전혀 배포하지 않는 조직이라면 오픈소스 프로그램의 적용 범위에 해당하지 않을 수 있다. 따라서, 기업의 오픈소스 정책은 오픈소스 프로그램의 적용 범위와 한계를 명확히 정의해야 한다.

예를 들어, "이 오픈소스 정책은 회사가 외부에 배포하는 모든 제품에 적용한다. 향후 배포하는 제품의 형태에 따라 프로그램의 구성과 적용 범위가 달라질 수 있으며, 이에 대해서는 오픈소스 팀이 OSRB와의 협의를 통해 결정한다."와 같은 형태로 프로그램 적용 범위를 정의할 수 있다.

5) 1.5 라이선스 의무 (License Obligations)

OpenChain Specification

1.5 라이선스 의무

각 라이선스에 의해 부여된 의무, 제한 및 권리를 결정하기 위해 식별된 라이선스를 검토하는 프로세스가 존재한다.

입증 자료:

1.5.1 각 식별된 라이선스에 의해 부과되는 의무, 제한 및 권리를 검토하고 문서화하기 위한 문서화된 절차.

1.5 License Obligations

A process exists for reviewing the Identified Licenses to determine the obligations, restrictions and rights granted by each license.

Verification Material(s):

1.4.1 A written statement that clearly defines the scope and limits of the Program.

오픈소스의 사용 가능 여부를 판단하기 위해서는 먼저 오픈소스의 라이선스가 무엇인지 식별하고, 라이선스가 요구하는 의무사항을 검토하고 확인해야 한다. 오픈소스 프로그램은 소프트웨어 개발팀에서 오픈소스 라이선스가 부여하는 의무, 제한 및 권리를 검토할 수 있도록 오픈소스 라이선스 의무 요약 자료를 제공하는 것이 좋다. NIPA에서 제공하는 "공개SW 라이선스 가이드" (https://www.oss.kr/oss_license)에서는 주요 오픈소스 라이선스의 의무, 제한 및 권리를 자세히 설명한다.

오픈소스를 사용하기에 앞서 라이선스 검토하고 이를 문서화하는 절차는 "[부록 02] 오픈소스 컴플라이언스 프로세스 (예시)" 절차의 오픈소스 식별 단계에 해당한다.

2 관련 업무 정의 및 지원 (Relevant Tasks Defined and Supported)

1) 2.1 접근성 (Access)

OpenChain Specification

2.1 접근성

외부 오픈소스 문의에 효과적으로 대응할 수 있는 프로세스를 유지한다. 제 3자가 오픈소스 컴플라이언스 문의를 할 수 있는 방법을 공개적으로 밝힌다.

입증 자료:

2.1.1 제 3자가 오픈소스 컴플라이언스 문의를 할 수 있게 공개적으로 알려진 방법 (공개된 연락처 이메일 주소, 또는 Linux Foundation의 Open Compliance Directory 등).

2.1.2 제 3자의 오픈소스 라이선스 컴플라이언스 문의에 대응하기 위한 내부의 문서화된 절차.

2.1 Access

Maintain a process to effectively respond to external Open Source inquiries. Publicly identify a means by which a third party can make an Open Source compliance inquiry.

Verification Material(s):

2.1.1 Publicly visible method that allows any third party to make an Open Source license compliance inquiry (e.g., via a published contact email address, or the Linux Foundation's Open Compliance Directory).

2.1.2 An internal documented procedure for responding to third party Open Source license compliance inquiries.

배포한 제품에 사용된 오픈소스에 대해 고객 및 오픈소스 저작권자가 기업에게 오픈소스 관련 문의, 요청 및 클레임을 제기하는 경우가 있다. 소송까지 당하지 않기 위해서는 이러한 외부 문의에 가능한 빠르고 정확하게 대응하는 것이 중요하다. 따라서 기업은

외부에서 기업에게 오픈소스 관련 문의를 할 수 있는 연락 방법을 공개적으로 밝히고, 외부 오픈소스 문의를 접수하였을 때 빠르고 효과적으로 대응 할 수 있는 프로세스를 갖추고 있어야 한다.

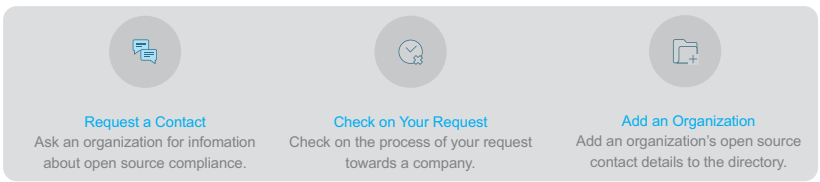
외부에서 기업에게 오픈소스 관련 문의를 할 수 있는 연락 방법은 회사의 오픈소스 담당자의 이메일 주소를 공개하거나, Linux Foundation의 Open Compliance Directory를 이용하는 것이다.

오픈소스 개발자들이 기업의 오픈소스 컴플라이언스 관련 이슈를 논의하기 위해 기업 담당자에게 연락하고 싶어도 연락 방법을 찾지 못하다가 결국 법적 클레임까지 제기하는 경우가 있다. Linux Foundation은 이러한 경우를 최소화 하기 위해 기업들에게 오픈소스 관련 문의를 받을 수 있는 연락처를 공개할 수 있도록 Open Compliance Directory라는 공간을 마련하였다

Open Compliance Directory

(https://compliance.linuxfoundation.org/references/open-compliance-directory/)

이를 통해 오픈소스 개발자들은 원하는 기업의 컨택 포인트 정보를 쉽게 확인할 수 있고, 법적 클레임까지 제기하기 이전에 기업의 오픈소스 담당자와 오픈소스 컴플라이언스 이슈를 논의하여 문제를 해결할 수 있다. 기업의 오픈소스 담당자는 Open Compliance Directory에 기업 정보 및 연락 방법을 등록하는 것이 소송 리스크를 줄일 수 있는 방법 중 하나이다.



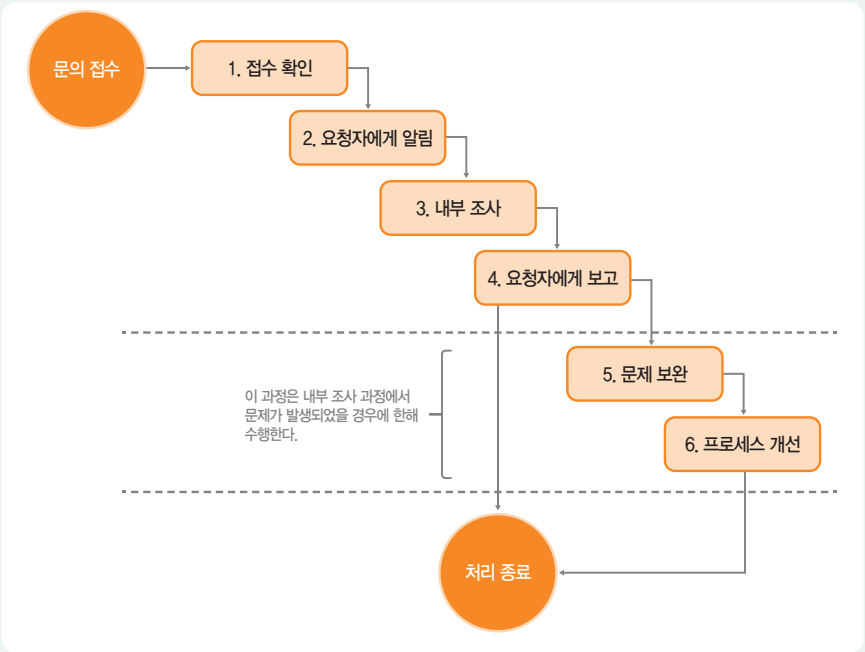
(https://www.linuxsources.org/content/open-compliance-directory-add-organization-request)

또한 기업은 제품의 사용자 매뉴얼에 오픈소스 관련 안내문을 추가하고, 여기에 연락 가능한 이메일 정보를 제공함으로 외부에 오픈소스 관련 기업의 연락처를 제공할 수 있다.

외부 문의 및 요청의 주된 내용은 다음과 같다.

- 특정 오픈소스가 제품 및 서비스에 사용되었는지 확인 요청
- Written Offer에서 언급된 GPL, LGPL 등의 라이선스 하의 소스 코드 제공 요청
- 오픈소스 고지문에 명시되지 않았지만 제품에서 발견된 오픈소스에 대한 해명 및 소스 코드 공개 요청
- GPL, LGPL 등의 의무로 공개된 소스 코드에 누락된 파일 제공 요청
- Copyright 표시 요청

외부로부터의 이러한 오픈소스 컴플라이언스 문의에 신속하고 정확하게 대응한다면 소송까지 진행되는 위험을 크게 줄일 수 있다. 따라서, 기업은 외부의 오픈소스 컴플라이언스 문의에 대응하기 위한 절차를 갖고 있어야 한다. 컴플라이언스 문의를 대응하기 위한 일반적인 절차는 다음과 같다.



① 접수 확인 (Acknowledge)

문의를 받으면 즉시 응답하여, 문의가 제대로 접수되었음을 알린다. 이때 조치 예정일을 함께 알린다. 요청자의 의도가 무엇인지 정확히 파악하는 것이 중요하기 때문에 문의가 불명확한 경우 추가 설명을 요청한다.

② 요청자에게 알림 (Inform)

요청자에게 오픈소스 컴플라이언스를 충실히 수행하고 있음과 요청자의 문의에 대해 조사하고 있음을 알린다. 내부 조사 진행사항이 업데이트되면 알리는 것이 좋다.

③ 내부 조사 (Investigate)

문의에 대해 내부 조사를 진행한다. 문제가 된 제품의 버전에 대하여 컴플라이언스 프로세스가 적절하게 수행되었는지 BOM 및 문서화 된 검토 이력을 통해 확인한다

④ 요청자에게 보고 (Report)

요청자에게 통보했던 조치 예정일 내에 내부 조사를 마치고, 이에 대한 내부 기록을 남긴 후 요청자에게 결과를 알린다.

⑤ 처리 종료 (Close Inquiry)

요청자의 문의가 오해로 인한 잘못된 지적이나 요청이었다면 추가 조치 없이 요청자에게 이를 알리고 처리를 종료한다.

⑥ 문제 보완 (Rectify)

내부조사에서 실제 컴플라이언스 문제가 발견되면 해당 조직은 제품 또는 서비스의 컴플라이언스 문제를 해결하기 위해 필요한 모든 절차를 수행한다. 예상되는 완료 일자를 요청자에게 다시 한번 알린다. 즉, 해당 오픈소스 라이선스의 의무를 이행하기 위한 정확한 방법과 시기를 알려야 한다. 문제를 해결한 후에는 즉시 요청자에게 알리고 문제가 해결되었음을 확인할 수 있는 최선의 방법을 제공한다.

⑦ 프로세스 개선 (Improve)

컴플라이언스 문제가 있었던 경우, OSRB 미팅을 통해 사례를 검토하고, 문제가 발생한 경위를 파악하여, 문제가 재발하지 않을 수 있도록 프로세스를 개선한다.

2) 2.2 효과적인 리소스 제공 (Effectively Resourced)

OpenChain Specification

2.2 효과적인 리소스 제공

프로그램 업무를 확인하고 리소스를 제공하라:
- 프로그램 업무를 성공적으로 수행할 수 있도록 책임을 할당하라.
- 프로그램 업무를 위해 충분한 리소스가 제공된다:
  - 업무를 수행할 시간이 할당되었다;
  - 적절한 자금이 할당되었다.
- 정책 및 지원 업무를 검토하고 업데이트하는 프로세스가 존재한다;
- 오픈소스 라이선스 컴플라이언스와 관련된 법률 가이드를 필요로 하는 인원이 법률 전문 지식을 이용할 수 있다;
- 오픈소스 라이선스 컴플라이언스 문제를 해결하기 위한 프로세스가 존재한다.

입증 자료:
2.2.1 확인된 프로그램 역할의 담당자 이름, 그룹 또는 기능이 기재된 문서
2.2.2 확인된 프로그램 역할이 적절하게 충원되었고 적합하게 자금이 제공되었다
2.2.3 오픈소스 라이선스 컴플라이언스 문제를 해결하기 위해 내부 또는 외부의 전문 법률 지식을 이용할 수 있는 방법의 확인.
2.2.4 오픈소스 컴플라이언스에 대한 내부 책임을 할당하는 문서화된 절차
2.2.5 미준수 사례의 검토 및 시정을 규정하는 문서화된 절차

2.2 Effectively Resourced

Identify and Resource Program Task(s):
- Assign accountability to ensure the successful execution of Program tasks.
- Program tasks are sufficiently resourced:
  - Time to perform the tasks have been allocated; and
  - Adequate funding has been allocated.
- A process exists for reviewing and updating the policy and supporting tasks;
- Legal expertise pertaining to Open Source license compliance is accessible to those who may need such guidance; and
- A process exists for the resolution of Open Source license compliance issues.

Verification Material(s):
2.2.1 Document with name of persons, group or function in Program role(s) identified.
2.2.2 The identified Program roles have been properly staffed and adequate funding provided.
2.2.3 Identification of legal expertise available to address Open Source license compliance matters which could be internal or external.
2.2.4 A documented procedure that assigns internal responsibilities for Open Source compliance.
2.2.5 A documented procedure for handling the review and remediation of non-compliant cases.

기업은 오픈소스 프로그램이 원활하게 기능을 수행할 수 있도록 리소스를 충분하게 제공해야 한다.

- 프로그램 참여자들이 업무를 수행할 수 있는 시간과 자금을 할당하고, 주기적으로 오픈소스 정책을 검토하여 기업의 소프트웨어 전략에 맞추어 업데이트해야 한다.
- 프로그램 참여자들이 컴플라이언스 이슈 해결을 위한 프로세스가 구축되어야 하고, 이슈 해결을 위해 법적인 검토가 필요할 경우 법무 자문을 요청할 수 있는 방법이 제공되어야 한다.

오픈소스 프로그램이 기능을 수행하기 위해서는 각 역할 별 담당자가 지정되어야 한다.

- 각 역할 별 담당자 혹은 담당 조직을 지정하고, 누구나 이를 참고할 수 있도록 문서화하여 공유한다.
- 각 조직의 책임자는 프로그램 내의 각 역할별 담당자가 적절히 충원되었는지, 업무를 수행하는데 필요한 자금이 적절하게 제공되었는지를 확인한다.

만약, 프로그램 참여자가 자신의 역할을 수행하는데 리소스나 자금 지원이 부족하다고 판단한다면, 반드시 기업의 오픈소스 책임자에게 문제를 제기하여 해결해야 한다. 문제가 효과적으로 해결되지 않을 경우, 오픈소스 이사회에 보고하고, 이사회는 필요한 의사결정을 수행하여 적절한 자원이 할당 될 수 있도록 해야 한다.

기업은 프로그램 참여자가 이슈 해결을 위해 법률적인 검토가 필요할 경우, 이에 대해 법률 자문을 요청할 수 있는 방법을 제공해야 한다. 회사 내의 법무팀을 통해 우선 제공하고, 이슈가 첨예한 경우, 오픈소스 전문 변호사를 보유한 외부 법무 법인을 이용할 수 있다. OpenChain Project에서는 파트너 프로그램을 통해 오픈소스 관련 자문을 제공하는 글로벌 법무법인 리스트를 제공한다.

(https://www.openchainproject.org/partners)

OpenChain 파트너로 등록된 법무법인은 OpenChain Project에서 요구하는 요건을 충족한 곳들이며, 대한민국에서는 법무법인 태평양이 등록되어 있다.

오픈소스 책임자는 기업의 오픈소스 컴플라이언스 활동을 위한 기업 내부의 역할과 책임을 할당해야 한다. 오픈소스 정책 문서에는 오픈소스 책임자가 오픈소스 컴플라이언스 이슈 해결을 위해 담당해야 할 역할에 대해 기술한다.

컴플라이언스 미준수 이슈가 제기된 경우, 기업은 이를 신속히 검토하고 대응하기 위한 절차를 문서화해야 한다. 이에 대한 자세한 내용은 2.1장에서 외부 문의 대응에 대한 프로세스 설명 부분을 참고할 수 있다.

3 오픈소스 컨텐츠 검토 및 승인 (Open Source Content Review and Approval)

1) 3.1 BOM (Bill of Materials)

OpenChain Specification

3.1 BOM

공급 대상 소프트웨어를 구성하는 각 오픈소스 컴포넌트(및 식별된 라이선스)를 포함하는 BOM을 작성하고 관리하는 프로세스가 있다.

입증 자료:

3.1.1 공급 대상 소프트웨어를 구성하는 오픈소스 컴포넌트 모음에 대한 정보를 식별, 추적, 검토, 승인 및 보관하는 문서화된 절차

3.1.2 공급 대상 소프트웨어에 대해 문서화된 절차가 적절히 준수되었음을 입증하는 오픈소스 컴포넌트 기록.

3.1 Bill of Materials

A process exists for creating and managing a bill of materials that includes each Open Source component (and its Identified Licenses) from which the Supplied Software is comprised.

Verification Material(s):

3.1.1 A documented procedure for identifying, tracking, reviewing, approving, and archiving information about the collection of Open Source components from which the Supplied Software is comprised.

3.1.2 Open Source component records for the Supplied Software that demonstrates the documented procedure was properly followed.

오픈소스 컴플라이언스 활동의 가장 기본은 바로 공급 대상 소프트웨어에 포함된 오픈소스 현황을 파악하는 것이다. 공급 대상 소프트웨어에 포함된 오픈소스와 그 라이선스를 식별하여 그 정보를 담고있는 BOM을 작성하고 관리하는 프로세스를 구축해야 한다. 공급 대상 소프트웨어마다 어떤 오픈소스가 포함되어 있는지 알고 있어야 소프트웨어를 배포할 때 각 라이선스가 요구하는 의무 사항을 준수할 수 있기 때문이다. 모든 오픈소스는 배포 대상 소프트웨어에 통합하기 전에 검토 및 승인되어야 한다. 오픈소스의 기능, 품질 뿐만 아니라 출처, 라이선스 요건을 충족하는지 검토가 되야 한다. 이를 위해 검토 요청 → 리뷰 → 승인 과정이 필요하다. [부록 02]에서는 기업의 오픈소스 컴플라이언스를 위한 프로세스 전과정에 대해 설명하고 있다. 식별부터 등록까지의 과정을 통해 BOM을 작성하고 관리하게 된다.

공급 대상 소프트웨어에 포함된 오픈소스 목록은 문서화하여 보관해야 한다. Eclipse 재단에서 후원하는 오픈소스 프로젝트인 SW360(https://projects.eclipse.org/proposals/sw360)은 공급 대상 소프트웨어별로 포함하고 있는 오픈소스 목록을 트래킹할 수 있는 기능을 제공한다. SW360 사용 방법은 [부록 03]을 참고할 수 있다.

오픈소스 컴플라이언스 프로세스의 모든 과정과 결과는 문서화가 되어야 한다. 이메일을 사용하는 것 보다는 Jira, Bugzilla 등의 이슈 트래킹 시스템을 이용하는 것이 이러한 과정을 효율적으로 문서화 할 수 있다.

2) 3.2 라이선스 컴플라이언스

OpenChain Specification

3.2 라이선스 컴플라이언스

프로그램은 공급 대상 소프트웨어에 대해 소프트웨어 공급 담당자가 접하게 되는 일반적인 오픈소스 사용 사례를 관리할 수 있어야 하며, 다음과 같은 사례가 포함될 수 있다(이 목록이 완전한 것은 아니며, 모든 사용 사례가 적용되어야 하는 것은 아니다).:

- 바이너리 형태로 배포;
- 소스 형태로 배포;
- Copyleft 의무를 발생시킬 수 있는 다른 오픈소스와 통합;
- 수정한 오픈소스를 포함;
- 공급 대상 소프트웨어 내에서 상호 작용하는 다른 컴포넌트와 호환되지 않는 라이선스 하의 오픈소스 또는 기타 소프트웨어를 포함;
- 저작자 표시 요건이 있는 오픈소스를 포함.

입증 자료:

3.2.1 공급 대상 소프트웨어의 오픈소스 컴포넌트에 대해 일반적인 오픈소스 라이선스 사용 사례를 처리하기 위한 문서화된 절차.

3.2 License Compliance

The Program must be capable of managing common Open Source license use cases encountered by Software Staff for Supplied Software, which may include the following use cases (note that the list is neither exhaustive, nor may all of the use cases apply):

- distributed in binary form;
- distributed in source form;
- integrated with other Open Source such that it may trigger copyleft obligations;
- contains modified Open Source;
- contains Open Source or other software under an incompatible license interacting with other components within the Supplied Software; and/or
- contains Open Source with attribution requirements.

Verification Material(s):

A documented procedure for handling the common Open Source license use cases for the Open Source components of the Supplied Software.

오픈소스 라이선스를 제대로 준수하기 위해서는 오픈소스 라이선스 별로 요구하는 사항에 대해 정확히 알고 있어야 한다. 개별 소프트웨어 개발자가 이를 일일이 파악하는 것은 어렵기 때문에 오픈소스 책임자는 자주 사용되는 오픈소스 라이선스 들에 대해 일반적인 사용 사례별 요구사항/주의사항을 정리하여 회사 내부에 공유하는 것이 좋다. 오픈소스 책임자는 자주 사용되는 오픈소스 라이선스별로 일반적인 사용 사례에 대한 의무 요약 자료를 제공한다. 오픈소스 라이선스에 대한 일반적인 가이드와 라이선스 의무 요약 자료는 NIPA에서 제공하는 "공개SW 라이선스 가이드"를 참고할 수 있다. (https://www.oss.kr/oss_license) [부록 2] 오픈소스 컴플라이언스 프로세스 (예시)의 오픈소스 컴플라이언스 프로세스의 식별, 검사, 문제해결, 리뷰, 승인 단계를 통해 공급 대상 소프트웨어의 오픈소스 컴포넌트에 대해 일반적인 오픈소스 라이선스 사용 사례를 처리할 수 있다.

식별 및 검사 단계에서는 소스코드 스캔 도구를 사용할 수 있다. 소스코드 스캔 도구는 무료로 사용할 수 있는 오픈소스 기반 도구부터 상용 도구까지 다양하게 있다. 각 도구들은 특장점 들이 있지만 어떤 하나도 모든 문제를 해결할 수 있는 완벽한 기능을 제공하지 않는다. 따라서 기업은 제품의 특성과 요구사항에 맞는 적합한 도구를 선택해야 한다. 많은 기업들이 이러한 자동화된 소스 코드 스캔 도구와 수동 검토를 병행하여 이용한다. Linux Foundation의 FOSSology Project는 오픈소스로 공개된 소스 코드 스캔 도구로서 기업들이 손쉽게 무료로 사용할 수 있다. 사용 방법은 [부록 03] 오픈소스도구 (FOSSology, SW360)를 참고할 수 있다.

(https://www.fossology.org/)

4 컴플라이언스 결과물 생성 및 전달

1) 4.1 컴플라이언스 결과물 (Compliance Artifacts)

OpenChain Specification

4.1 컴플라이언스 결과물

공급 대상 소프트웨어에 대한 컴플라이언스 결과물 세트를 생성하는 프로세스가 존재한다.

입증 자료:

4.1.1 식별된 라이선스에서 요구하는 대로 컴플라이언스 결과물을 준비하고 공급 대상 소프트웨어와 함께 배포하기 위한 프로세스를 설명하는 문서화된 절차.

4.1.2 공급 대상 소프트웨어의 컴플라이언스 결과물 사본을 보관하기 위한 문서화된 절차 – 보관 파일은 공급 대상 소프트웨어의 마지막 제공 이후 적절한 기간(혹은 식별된 라이선스가 요구하는 기간 (둘 중 더 긴 시간)) 동안 보관되어야 한다. 절차가 올바르게 지켜졌음을 입증하는 기록이 존재한다.

4.1 Compliance Artifacts

A process exists for creating the set of Compliance Artifacts for the Supplied Software.

Verification Materials(s):

4.1.1 A documented procedure that describes the process under which the Compliance Artifacts are prepared and distributed with the Supplied Software as required by the Identified Licenses.

4.1.2 A documented procedure for archiving copies of the Compliance Artifacts of the Supplied Software – where the archive is planned to exist for a reasonable period of time since the last offer of the Supplied Software; or as required by the Identified Licenses (whichever is longer). Records exist that demonstrate the procedure has been properly followed.

3.1장에서 오픈소스 컴플라이언스 활동의 가장 기본은 공급 대상 소프트웨어에 포함된 오픈소스 현황을 파악하는 것이라고 하였다. 이는 바로 오픈소스 컴플라이언스의 핵심인 오픈소스 라이선스의 의무를 파악하여 요건들을 충족하기 위해서이다. 즉, 공급 대상

소프트웨어에 포함된 것으로 식별한 오픈소스에 대한 컴플라이언스 결과물 세트를 생성하는 프로세스가 구축되어야 한다.

컴플라이언스 결과물은 크게 두가지로 구분된다.

1. 오픈소스 고지문 : 오픈소스 라이선스 전문과 Copyright 정보 제공을 위한 문서
2. 공개할 소스코드 패키지 : GPL, LGPL 등 소스 코드 제공을 요구하는 오픈소스 라이선스 의무 이행을 위해 공개할 소스코드를 취합한 패키지

컴플라이언스 결과물은 공급 대상 소프트웨어를 배포할 때 함께 제공해야 한다. "[부록 02] 오픈소스 컴플라이언스 프로세스(예시)" 의 고지, 확인, 배포 단계를 통해 컴플라이언스 결과물을 생성하여 배포한다.

공급 대상 소프트웨어를 배포 시, 공개할 소스코드 패키지를 동봉하는 것이 곤란할 경우, 최소 3년간 소스코드를 제공하겠다는 서면 약정서(Written Offer)를 제공하는 것으로 대신할 수 있다. 일반적으로 서면 약정서는 제품의 사용자 매뉴얼을 통해 제공하며, 예시는 다음과 같다.

> The software included in this product contains copyrighted software that is licensed under the GPL. A copy of that license is included in this document on page X. You may obtain the complete Corresponding Source code from us for a period of three years after our last shipment of this product, which will be no earlier than 2011-08-01, by sending a money order or check for $5 to:
>
> GPL Compliance Division
> Our Company
> Any Town, US 99999
>
> Please write "source for product Y" in the memo line of your payment.
> You may also find a copy of the source at http://www.example.com/sources/Y/.
> This offer is valid to anyone in receipt of this information.

(https://www.softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.html#appendix-1-offer-of-source-code)

따라서, 컴플라이언스 결과물은 3년 이상 보관해야 하며 이를 위한 프로세스가 구축되어야 한다. 기업들은 자체적인 웹사이트(예: http://opensource.lge.com/)를 구축하여 외부 고객들이 공급 대상 소프트웨어에 대한 오픈소스 고지문과 공개할 소스코드 패키지를 언제든지 다운받을 수 있도록 편의를 제공한다.

5 오픈소스 커뮤니티 참여에 대한 이해

1) 5.1 기여

OpenChain Specification

5.1 기여

조직이 오픈소스 프로젝트에 기여를 고려한다면
- 오픈소스 프로젝트에 대한 기여를 관리하는 문서화된 정책이 존재한다;
- 이 정책이 내부적으로 전달되어야 한다;
- 정책을 구현하는 프로세스가 존재한다.

입증 자료:

조직이 오픈소스 프로젝트에 대한 기여를 허용한다면 다음이 존재해야 한다:

5.1.1 문서화된 오픈소스 기여 정책;

5.1.2 오픈소스 기여를 관리하는 문서화된 절차;

5.1.3 모든 소프트웨어 공급 담당자가 오픈소스 기여 정책의 존재를 인식하도록 하는 문서화된 절차 (교육, 내부 위키, 또는 기타 실질적인 의사소통 방법 등).

5.1 Contributions

If an organization considers contributions to Open Source projects then
- a written policy exists that governs contributions to Open Source projects;
- the policy must be internally communicated; and
- a process exists that implements the policy

Verification Materials(s):

If an organization permits contributions to Open Source projects then the following must exist:
- 5.1.1 a documented Open Source contribution policy;
- 5.1.2 a documented procedure that governs Open Source contributions; and
- 5.1.3 a documented procedure that makes all Software Staff aware of the existence of the Open Source contribution policy (e.g., via training, internal wiki, or other practical communication method).

글로벌 소프트웨어 기업들은 오픈소스를 사용하여 제품을 만들고 서비스를 하는 것 뿐만 아니라 오픈소스 프로젝트에 기여하며 얻을 수 있는 전략적 가치도 중요하게 여긴다. 그러나 오픈소스 프로젝트 생태계와 커뮤니티 운영방식에 대한 충분한 이해와 전략 없이 접근한다면 예기치 않게 회사의 명성이 손상되고 법적 위험이 발생할 수 있다. 따라서 기업은 오픈소스 프로젝트로의 참여 및 기여를 위한 전략과 정책을 만드는 것이 중요하다.

[부록 01] 오픈소스 정책 for OpenChain 2.0(예시)의 8장 오픈소스 기여 정책을 참고할 수 있다.

6 설명서 요건 준수

1) 6.1 준수 (Conformance)

OpenChain Specification

6.1 준수

프로그램이 OpenChain을 준수한다고 간주되려면 조직은 프로그램이 이 설명서에 제시된 요건을 충족하는지 확인해야 한다.

입증 자료:

6.1.1 요건 1.4에 명시된 프로그램을 확인하는 문서는 이 설명서의 모든 요건을 충족한다.

6.1 Conformance

In order for a Program to be deemed OpenChain Conformant, the organization must affirm that the program satisfies the requirements presented in this specification.

Verification Materials(s):

6.1.1 A document affirming the Program specified in requirement 1.4 satisfies all the requirements of this specification.

기업이 OpenChain을 준수하는 오픈소스 프로그램을 가지고 있다고 선언한다는 것은 OpenChain Specification의 모든 요건을 충족한다는 것이다. 어느 하나의 요건이라도 충족하지 못한다면 OpenChain을 준수한다고 할 수 없다.

OpenChain Specification의 모든 요건을 충족한다면, [부록 01] 오픈소스 정책 for OpenChain 2.0(예시)의 9장에서와 같이 OpenChain을 충족하고 있음을 문서상에 명시할 수 있다.

2) 6.2 기간(Duration)

OpenChain Specification

6.2 기간

이 설명서 버전에 대한 OpenChain 준수 프로그램은 준수한다고 확인이 이루어진 날로부터 18개월동안 지속된다. 준수 확인 등록 절차는 OpenChain 프로젝트의 웹사이트에서 확인할 수 있다.

입증 자료:

6.2.1 준수한다는 확인이 이루어진 후 18개월 이내에 이 설명서 버전(2.0)의 모든 요건을 충족하는 것을 확인하는 문서.

6.2 Duration

A Program that is OpenChain Conformant with this version of the specification will last 18 months from the date conformance validation was obtained. The conformance validation registration procedure can be found on the OpenChain project's website.

Verification Materials(s):

6.2.1 A document affirming the Program meets all the requirements of this version of the specification (version 2.0), within the past 18 months of obtaining conformance validation.

오픈소스 프로그램이 OpenChain을 준수한다고 선언한 이후에도 계속해서 준수하는 활동을 유지하는 것이 중요하다. OpenChain Specification 2.0의 6.2.1조에서는 OpenChain을 준수한다고 선언한 이후에도 최소 18개월 이상은 변함없이 OpenChain Specification 2.0의 모든 요건을 준수하고 있어야 함을 요구한다.

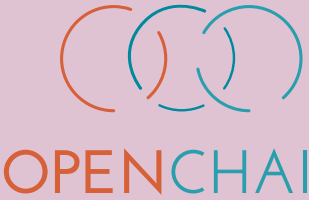
기업은 오픈소스 프로그램이 OpenChain을 준수함을 선언한 이후 적어도 18개월 이상 계속해서 준수하는 상태를 유지하여야 하며, 그렇게 하고 있다면, [부록 01] 오픈소스 정책 for OpenChain 2.0(예시)의 9장에서와 같이 OpenChain을 18개월 이상 계속하여 충족하고 있음을 문서상에 명시할 수 있다.

부 록

1. [부록 01] 오픈소스 정책 for OpenChain 2.0(예시)
2. [부록 02] 오픈소스 컴플라이언스 프로세스(예시)
3. [부록 03] 오픈소스 도구 사용 방법(FOSSology, SW360)

References

1. OpenChain Project Website : https://www.openchainproject.org/
2. OpenChain Specification 2.0 : https://wiki.linuxfoundation.org/_media/openchain/openchainspec-2.0.pdf
3. OpenChain Open Source Policy : https://www.openchainproject.org/news/2019/01/17/openchain-open-source-policy-template-now-available
4. Open Source Compliance In The Enterprise / Ibrahim Haddad : https://www.linuxfoundation.org/compliance-and-security/2018/12/open-source-compliance-in-the-enterprise/



[부록 01] 오픈소스 정책 for OpenChain 2.0 (예시)

[부록 02] 오픈소스 컴플라이언스 프로세스 (예시)

[부록 03] 오픈소스 도구 (FOSSology, SW360)

OPENCHAIN

[부록 01]
오픈소스 정책 for OpenChain 2.0 (예시)

이 오픈소스 정책 for OpenChain 2.0(예시))은 다음 두가지 자료를 참고하여 작성하였다.

1. OpenChain Open Source Policy Template : https://github.com/OpenChain-Project/Reference-Material/blob/master/Policy-Templates/Official/en/2.0/Open-Source-Policy-Template-2.0-en.xlsx
2. Linux Foundation Generic FOSS Policy : https://github.com/todogroup/policies/blob/master/linuxfoundation/lf_compliance_generic_policy.pdf

○○회사 오픈소스 정책

1 목 적

이 정책은 오픈소스를 사용하는 조직 전체가 오픈소스 컴플라이언스 활동을 수행하도록 수립되었다. 또한 이 정책은 직원들이 오픈소스의 가치를 이해하게 하고, 오픈소스 커뮤니티에 기여하기 위한 방법을 제공한다.

〈○○회사〉의 직원은 이 정책의 근거와 내용을 이해하고 필요한 활동을 충실히 수행함으로써 정책의 효과 및 회사의 컴플라이언스 수준 향상에 기여한다.

이 정책을 준수하는 것은 중요하다. 준수하지 않을 경우 다음과 같은 상황을 초래할 수 있다.

- 사용 중인 코드에 대한 저작권 또는 기타 지식재산권 보유자의 법적 클레임
- 고객으로부터의 클레임
- 회사 독점 코드의 의도치 않은 공개
- 라이선스 의무 위반으로 인한 벌금 부과
- 평판 손실
- 수익 손실
- 공급업체 및 고객과의 계약 위반

이러한 이유로 〈OO회사〉는 코드 침해를 심각하게 간주하며, 코드를 침해하는 개인은 회사의 징계 절차에 처해질 수 있다.

2 적 용

이 오픈소스 정책은 [회사가 외부로 제공하거나 배포하는 모든 제품]에 적용된다. 오픈소스를 내부 사용 목적으로만 사용하는 것은 이 정책의 범위에 포함되지 않는다.

또한 이 정책은 〈OO회사〉의 직원이 오픈소스 프로젝트에 기여하거나 〈OO회사〉의 코드를 오픈소스로 공개할때 적용한다.
〈OO회사〉의 오픈소스 정책은 [LINK]에서 확인할 수 있다.

3 용 어

"오픈소스" – Open Source Initiative(OpenSource.org)에서 발표한 Open Source Definition 혹은 Free Software Foundation에서 발표한 Free Software Definition을 충족하는 라이선스, 혹은 유사한 라이선스가 하나 이상 적용된 소프트웨어.

4 역할 , 책임 및 역량

이 정책의 효과적인 수행을 보장하기 위해 다음과 같이 필요한 역할 및 책임과 각 역할의 담당자가 갖추어야 할 역량을 정의한다.

〈OO회사〉의 소프트웨어 개발 및 배포를 담당하는 최고 임원은 각 역할 및 책임을 위한 담당자가 지정되고, 역할을 수행할 적절한 자금과 시간이 할당되도록 보장해야 한다.
각 역할의 담당자는 자신의 역할에 대해 적절하게 지원이 되지 않는다면 반드시 오픈소스 책임자를 통해 문제를 해결해야 한다. 적절하게 해결되지 않는다면, 오픈소스 운영위원회를 통해 문제를 제기할 수 있다.

가) 오픈소스 책임자

오픈소스 책임자는 오픈소스가 사용된 〈OO회사〉 제품의 컴플라이언스를 보장할 책임과 함께 다음 사항에 대한 책임이 있다.

- 오픈소스 정책을 검토, 개선 및 전파한다.
- 효율적인 오픈소스 정책 수행을 위해 회사 내부의 역할 및 책임을 검토하고 할당한다.
- 오픈소스 컴플라이언스 관련 이슈에 대한 교육과 평가를 검토하고 구현한다.
- 오픈소스 운영위원회의 의장을 맡아서 활동을 지휘한다.
- 소프트웨어 개발팀이 오픈소스 정책과 프로세스를 이해하고 준수하도록 안내하는 역할을 하고, 필요할 경우 경영진에게 문제를 제기한다.
- 외부로부터의 오픈소스 사용 및 컴플라이언스에 대한 문의에 답변한다.

오픈소스 책임자는 업무 수행을 위해 오픈소스 관련 IP 리스크, 개발 프로세스를 이해하고, 커뮤니케이션 스킬에 대한 역량을 갖춰야 한다.

2020년 1월 현재 OOO팀의 OOO가 오픈소스 책임자 역할을 담당한다.

나) 오픈소스 센터

오픈소스 센터는 오픈소스 컴플라이언스를 위한 전문 센터이며, 컴플라이언스를 효과적으로 달성하기 위한 프로세스를 정의한다. 오픈소스 책임자가 리더 역할을 수행하고, 센터의 구성원들은 오픈소스 책임자가 원활하게 책임을 수행할 수 있도록 돕는 역할을 맡는다. 오픈소스 센터는 다음과 같은 역할을 수행한다.

- 컴플라이언스 실무 교육을 개발 및 제공한다.
- 컴플라이언스 도구를 선택 / 개발 및 배포한다.
- 코드 검사 및 자동 스캔을 수행하여 〈OO회사〉 제품 내 오픈소스 포함 여부를 식별한다.
- 오픈소스 사용 요청을 검토하고 승인한다.
- 오픈소스 사용 목록에 관한 기록을 유지한다.
- 오픈소스 고지 및 소스코드 공개를 위한 웹 사이트를 개발하고 유지 관리한다.

다) 소프트웨어 개발팀

소프트웨어 개발팀은 소프트웨어 개발에 사용할 오픈소스를 식별하고 오픈소스 센터에 오픈소스 사용 승인 요청을 제출한다.
소프트웨어 개발팀은 소프트웨어 개발에 사용한 오픈소스에 적용되는 오픈소스 라이선스의 의무를 이행할 책임이 있다.
소프트웨어 개발팀은 오픈소스 정책 및 프로세스와 소프트웨어 아키텍쳐를 이해한다.

라) 법무팀

법무팀은 오픈소스 라이선스와 의무를 해석한다. 이러한 의무를 이행하기 위한 가이드를 소프트웨어 개발팀에 제공한다. 호환되지 않는 오픈소스 라이선스로 인한 충돌을 포함하여 라이선스 및 지식재산권 문제에 대해 자문을 제공한다. 필요할 경우 오픈소스 사용 검토 및 승인 결정에 참여한다.
오픈소스 프로젝트로의 기여를 위한 검토 요청에 의견을 제공한다.

5 교육 및 평가

소프트웨어 배포에 관여하는 〈OO회사〉의 모든 직원은 교육 및 평가를 통해 오픈소스 정책을 숙지한다.

이 정책을 수행하는 모든 대상자는 자신의 역할에 필요한 역량을 다루는 최소한의 기본 교육을 수강하고 평가를 받는다.

교육 및 평가 프로그램은 〈OO회사〉 오픈소스정책의 목표, 컴플라이언스 수준 향상에 기여하기 위한 참여자의 역할 및 컴플라이언스 미준수 시 회사 및 개인에 미치는 영향 등에 대해 다룬다.
평가 기록은 최소 3년동안 유지한다.

6 오픈소스 사용 정책

오픈소스를 사용하기 위해서는 먼저 오픈소스 라이선스가 무엇인지 식별하고, 라이선스가 요구하는 의무 사항을 검토하고 확인한다. 그렇게 공급 대상 소프트웨어에 포함된 오픈소스와 라이선스 의무사항을 식별하고, 소프트웨어를 배포 시 라이선스 의무사항을 준수하기 위한 활동을 한다.
이를 효과적으로 수행하기 위해 〈OO회사〉 오픈소스 컴플라이언스 프로세스를 준수한다.

오픈소스 라이선스 준수를 위한 과정에서 의문사항이 있는 경우 [오픈소스 책임자]는 법무팀에게 문의 할 수 있다.

오픈소스 사용 결정 결과 및 관련 근거는 오픈소스 이슈 추적 시스템에 기록한다.

7 외부 문의 대응 정책

〈OO회사〉에서 배포한 소프트웨어에 대해 외부에서 오픈소스 관련한 문의 및 요청을 할 수 있도록 공개된 연락처를 제공한다. 이를 위해 소프트웨어 배포 시 오픈소스 센터의 이메일 주소를 제공하고,
Linux Foundation의 Open Compliance Directory (https://compliance.linuxfoundation.org/references/open-compliance-directory/)에 〈OO회사〉의 연락처를 등록한다.

외부로부터 오픈소스 관련 문의를 받은 사람은 누구나 오픈소스 책임자에게 문의한다. 오픈소스 책임자는 문의를 처리하고 회사 내 적절한 개인 또는 조직에 할당한다. 오픈소스 책임자는 문의를 할당하고 처리하는 것에 대한 전반적인 책임이 있다.

〈OO회사〉에서 배포한 소프트웨어에 대해 외부로부터 컴플라이언스 미준수 이슈가 제기될 경우, 오픈소스 책임자는 다음과 같이 처리한다.

1. 질의 접수 승인 및 적절한 해결 시간을 명시한다.
2. 질의가 진짜 문제인 것인지를 확인한다. (아니라면 영업일 기준 3일 이내에 질의자에게 응답한다.)
3. 이슈가 진짜 문제라면, 3일 이내에 적절한 대응 방법을 결정하고, 질의자에게 대응 계획에 대해 응답한다.

4. 결정한 방법에 따라 30일 이내에 대응하고, 질의자에게 문제가 해결되었음을 알린다.
5. 이상의 사항을 오픈소스 이슈 추적시스템에 기록한다.

8 오픈소스 기여 정책

〈OO회사〉는 오픈소스에서의 비즈니스 가치 창출을 위해 외부 오픈소스 프로젝로의 참여와 기여를 권장한다. 그러나 의도하지 않은 지식 재산의 노출 혹은 침해를 주의해야 한다.

회사의 업무와 관련이 있는 오픈소스 프로젝트에 기여하기 위해서는 먼저 SW개발팀 리더에게 승인을 받아야 한다.

그리고 오픈소스 프로젝트의 오픈소스 라이선스와 특허 조건을 검토한다. 또한 기여하고자 하는 오픈소스 프로젝트가 요구하는 DCO (Developer Certificate of Origin), CLA (Contributor License Agreement)등의 문서 서명에 대해 검토해야 한다. 필요할 경우 법무팀에 검토를 요청할 수 있다.

9 OpenChain 준수

〈OO회사〉는 소프트웨어 공급망에서의 오픈소스 컴플라이언스 수준 향상을 위해 Linux Foundation의 OpenChain 프로젝트의 정신을 지지하며 적극적으로 참여한다.

〈OO회사〉의 오픈소스 정책은 OpenChain Specification 2.0을 준수하도록 설계되었다.

〈OO회사〉는 〈OO회사〉의 오픈소스 정책을 포함하는 오픈소스 프로그램이 OpenChain Specification 2.0의 모든 요건을 준수하고 있음을 확약한다.

〈OO회사〉는 〈OO회사〉의 오픈소스 정책을 포함하는 오픈소스 프로그램이 OpenChain Specification 2.0의 모든 요건을 준수하고 있음을 확약한 이후 18개월 동안 여전히 모든 요건을 준수하기 위한 활동을 수행하고 있음을 확약한다.

OPENCHAIN

[부록 02]
오픈소스 컴플라이언스 프로세스(예시)

오픈소스 컴플라이언스의 주요 두가지 목적은 다음과 같다.

1. 의무 파악 : 공급 대상 소프트웨어가 포함하고 있는 오픈소스를 식별하고 각 오픈소스 라이선스가 요구하는 의무를 파악한다.
2. 의무 사항 이행 : 식별한 의무 사항을 이행한다.

이를 위해 기업은 공급 대상 소프트웨어를 배포하는 시점에 오픈소스 라이선스 의무사항을 준수할 수 있도록 오픈소스 컴플라이언스 프로세스를 구축해야 한다. 여기서는 일반적인 오픈소스 컴플라이언스 프로세스의 구성요소와 각각의 기능 및 역할을 포함하는 프로세스(예시)를 제안한다.

이 오픈소스 컴플라이언스 프로세스 (예시)는 다음 자료를 참고하여 작성하였다.

- Open Source Compliance In The Enterprise / Ibrahim Haddad : https://www.linuxfoundation.org/compliance-and-security/2018/12/open-source-compliance-in-the-enterprise/

〈○○회사〉 오픈소스 컴플라이언스 프로세스

〈OO 회사〉의 오픈소스 정책에 근거하여 오픈소스를 사용하기 위해서는 먼저 오픈소스 라이선스가 무엇인지 식별하고, 라이선스가 요구하는 의무 사항을 검토하고 확인해야 한다.
그렇게 공급 대상 소프트웨어에 포함된 오픈소스와 라이선스 의무사항을 식별하고, 소프트웨어를 배포 시 라이선스 의무사항을 준수하기 위한 오픈소스 컴플라이언스 활동을 해야 한다.

〈OO회사〉의 오픈소스 컴플라이언스 프로세서는 공급 대상 소프트웨어에 사용되는 오픈소스를 관리하는 일련의 과정을 정의한다. 이 과정에는 다음 사항이 포함된다.

1. 공급 대상 소프트웨어에 사용된 모든 오픈소스 식별
2. 식별한 오픈소스에 의해 발생하는 모든 의무를 식별하고 추적
3. 모든 의무를 충족하기 위한 활동

이를 효과적으로 수행하기 위해 〈OO회사〉의 모드 소프트웨어 공급관리자는 다음 10단계를 수행한다.

Step 1. 오픈소스 식별 (Identification of Open Source)

오픈소스 식별 단계는 오픈소스 컴포넌트를 식별하기 위한 검토 단계이다. 자체 독점 소프트웨어인지, 제3자 소프트웨어인지 여부에 관계 없이 공급 대상 소프트웨어에 포함된 오픈소스를 모니터링한다. 오픈소스 식별 방법은 다음과 같다.

- 오픈소스 사용 요청 접수 : SW개발자는 특정 제품에 오픈소스를 사용하고자 함을 오픈소스 책임자 또는 오픈소스 센터에 알리고, 검토 및 승인을 위한 오픈소스 패키지의 용도에 관한 정보를 제공한다.
- 회사 개발 소프트웨어 검사 (Auditing) : 개발자가 오픈소스의 소스코드를 복사해서 가져와 소프트웨어를 개발할 수 있기 때문에 회사가 개발한 소프트웨어에 대해서도 검사를 수행한다.
- 제3자 소프트웨어 실사 (Due diligence)

식별 단계 시작 조건	식별 단계 결과
• 개발자로부터 특정 오픈소스 사용 요청 접수 • 개발 프로세스 상 소프트웨어 검사 단계 • 제3자 소프트웨어 입수 및 개발소프트웨어로의 통합	• 오픈소스에 대한 컴플라이언스 기록 생성 (Jira 등 활용) • 소스코드 스캔 대상 선정 및 요청

Step 2. 소스 코드 검사 (Auditing Source Code)

소스 코드 검사 단계에서는 소스 코드 분석 도구를 사용하여 소스 코드를 스캔하여 오픈소스를 발견한다. 소스 코드 스캔도구는 FOSSology를 이용한다. GPL-3.0 등 정책적으로 사용할 수 없는 오픈소스 라이선스가 적용된 오픈소스 혹은 라이선스 충돌로 양립할 수 없는 오픈소스가 발견될 경우 문제로 식별하여 개발팀에 보완을 요청한다.

소스 코드 검사 단계 시작 조건	소스 코드 검사 단계 결과
• 소스 코드 스캔 요청 (Jira ticket 생성)	• 소스 코드 스캔 결과 생성 (오픈소스 출처, 라이선스 등 정보 포함) • 식별된 문제에 대해 개발팀에 보완 요청 (Jira ticket 생성)

Step 3. 문제 해결 (Resolving Issues)

소스 코드 검사 단계에서 식별된 모든 문제를 해결한다. 문제 사항은 Jira Ticket으로 생성하여 개발팀에 할당되고, 오픈소스 책임자는 모든 문제가 적절하게 해결되었는지 확인한다.

문제 해결 단계 시작 조건	문제 해결 단계 결과
• 소스 코드 스캔 완료 및 결과 생성 • 문제 식별	• 식별된 문제를 모두 해결

Step 4. 검토 (Reviews)

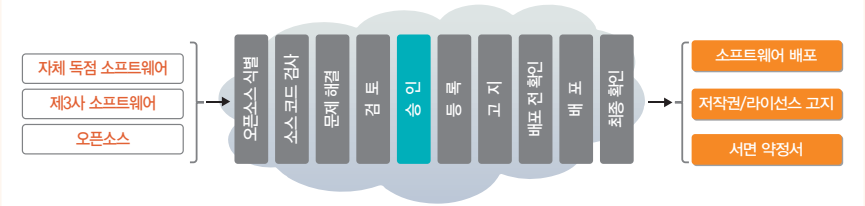
식별된 모든 문제가 해결되면 검토 단계로 이동한다. 검토 단계의 절차는 다음과 같다.

1. 소프트웨어 PL : 소프트웨어에 포함된 오픈소스에 대한 사용 승인 요청서를 제출한다.
2. 오픈소스 책임자 : 사용 승인 요청서를 접수하면 모든 정보가 누락없이 포함 되었는지를 확인하고, Jira ticket을 생성하여 검토 절차를 진행한다.

3. 소스코드 검사 담당자: Jira ticket이 생성되면 소스코드 검사를 수행하여 문제가 모두 해결되었는지 확인한다.
4. 법무팀 : 라이선스 이슈를 검토한다.

검토 단계 시작 조건	검토 단계 결과
• 식별된 모든 문제 해결	• 오픈소스 책임자, 소스코드 검사 담당자, 법무팀 등의 검토를 완료하여 승인 준비가 된 상태

Step 5. 승인 (Approval)



검토가 완료되면 Jira ticket은 승인 단계로 이동한다. OSRB는 오픈소스의 사용을 승인하거나 거절한다. 거절시에는 이유에 대한 설명과 수정 방법을 제안한다. OSRB가 오픈소스 구성요소의 사용을 승인하면 개발팀은 라이선스 의무를 이행하기 위한 준비를 시작한다.

승인 단계 시작 조건	승인 단계 결과
• 검토가 완료된 상태	• OSRB는 오픈소스의 사용을 승인하거나 거절함 • 거절 시에는 이유에 대한 설명과 수정 방법 제안

Step 6. 등록 (Registration)

사용이 승인된 오픈소스 구성요소는 오픈소스 사용을 추적하는 BOM (소프트웨어 인벤토리)에 추가한다. BOM에는 오픈소스 구성요소 이름, 버전, 관리 담당자 이름, 이를 사용하는 제품 이름, 제품 버전, 제품 릴리즈 번호 등의 정보를 포함한다. BOM을 관리하는 도구는 SW360을 사용한다.

등록 단계 시작 조건	등록 단계 결과
• OSRB가 오픈소스 사용을 승인	• 오픈소스 구성요소를 BOM에 등록

Step 7. 고지 (Notices)

오픈소스를 사용할 때 주요 의무 중 하나는 고지 의무이다. 이를 위해 다음 사항을 수행한다.

- 저작권, 라이선스 고지를 제공한다.

- 라이선스 사본을 제공한다.
- (해당되는 경우) 소스 코드 사본을 얻을 수 있는 방법을 최종 사용자에게 알린다.

고지 단계 시작 조건	고지 단계 결과
• 오픈소스를 BOM에 등록	• 저작권, 라이선스 고지를 준비하고, 이를 제품에 포함되도록 관련 부서로 전달

이와 같은 사항을 제품 배포 시 포함시킬 수 있도록 관련 부서에 전달한다. 화면이 있는 제품이면 사용자가 메뉴 〉 오픈소스 고지 정보에서 오픈 소스 고지 내용을 확인할 수 있게 한다. 제품에 화면이 없을 경우, 사용자 매뉴얼에 오픈소스 고지 내용을 포함시킨다.

Step 8. 배포 전 확인 (Pre-Distribution Verifications)

이 단계에서는 다음 사항을 보장하기 위한 확인을 수행한다.

- 오픈소스 라이선스가 요구하는 공개할 소스 코드를 취합한다.
- 취합한 소스 코드는 제품에 탑재된 바이너리와 매치되어야 한다.
- 소스 코드 내 부적절한 주석을 제거한다.
- 적절한 고지문이 제품에 포함되었다. 여기에는 최종 사용자가 소스 코드를 받을 수 있는 방법 (Written Offer)도 함께 제공한다.

배포 전 확인 단계 시작 조건	배포 전 확인 단계 결과
• 모든 오픈소스 구성요소가 BOM에 등록	• 고지 의무를 이행할 수 있도록 조치 • 공개할 소스 코드 취합 • 소스 코드 제공 방법 결정 • 배포 전 확인 수행 완료

Step 9. 배포 (Distribution)

배포 전 확인이 완료되면 공개할 소스 코드 패키지를 오픈소스 배포사이트에 업로드한다. 오픈소스 배포사이트에는 제품 및 버전별로 등록할 수 있다. 최종 사용자는 자신이 원하는 제품의 버전에 해당하는 소스 코드 패키지를 오픈소스 배포사이트에서 검색하여 다운로드 받을 수 있다.

배포 단계 시작 조건	배포 단계 결과
• 모든 배포 전 확인 완료	• 특정 제품의 버전에 대한 공개할 소스 코드 패키지를 오픈소스 배포사이트에 업로드

Step 10. 최종 확인 (Final Verifications)

공개할 소스 코드 패키지를 오픈소스 배포사이트에 업로드 후 패키지가 올바르게 업로드 되었고, 외부에서 오류 없이 다운로드 및 압축 해제가 되는지 확인한다. 라이선스에 따라 빌드하여 바이너리 생성까지 보장을 요구하는 경우, 외부에서 다운받은 소스 코드가 README의 안내대로 오류 없이 빌드하여 바이너리가 생성되는지, 생성된 바이너리가 제품에 탑재된 바이너리와 동일한지 확인한다.

최종 확인 단계 시작 조건	최종 확인 단계 결과
• 공개할 소스 코드가 오픈소스 배포사이트에 게시	• 외부에서 다운로드가 이상없이 수행되는지, 제품과 동일한 버전의 바이너리와 매치가 되는지 확인

OPENCHAIN

[부록 03]
오픈소스 도구(FOSSology, SW360)

오픈소스 컴플라이언스 활동을 위해서는 정책, 프로세스나 교육자료뿐만 아니라 소스코드 스캔, Dependency 분석, 오픈소스 BOM 관리 등을 위한 다양한 도구와 시스템도 요구된다. 때문에 다수의 기업이 이러한 도구와 시스템을 도입하고 활용하는데 많은 리소스를 투입하고 있다. 특히 오픈소스 컴플라이언스를 처음 시작하는 기업은 프로세스뿐 아니라 비용 측면에서도 어려움을 겪고 있다.

이런 어려움을 해결하기 위해, 2019년 6월, OpenChain 프로젝트에 참여하고 있는 지멘스, 보쉬, 도시바, 후지쓰, 히타치 등의 오픈소스 컴플라이언스 도구 전문가들을 주축으로 OpenChain Tooling Work Group이 시작되었다.

OpenChain Tooling Work Group은 여러 기업의 오픈소스 전문가들이 이슈를 함께 해결하고 결과물을 공유해 오픈소스 컴플라이언스 비용을 절감하고 양질의 컴플라이언스 결과물을 만들어 내기 위해 구성되었다.

구체적으로는 FOSSology, SW360, Software Heritage, ClearlyDefined, SPDX 등의 기존 오픈소스 프로젝트를 활용하여 통합(turn-key) 오픈소스 툴 체인을 만들고, 모든 기업이 이를 자유롭게 사용할 수 있도록 하는 것을 목표로 삼고 있다.
(https://groups.io/g/oss-based-compliance-tooling)

여기서는 FOSSology와 SW360에 대해 소개 및 간단한 사용 방법에 대해 알아본다.

오픈소스 컴플라이언스를 위해 소프트웨어 내에 포함된 오픈소스와 라이선스 정보를 검출하기 위해 소스코드 스캔 도구를 사용할 수 있다. Linux Foundation의 FOSSology 프로젝트는 이러한 스캔 도구를 개발하고 오픈소스로 공개해 누구나 자유롭게 사용할 수 있게 한 도구다.
(https://www.fossology.org/)

주요 특징

FOSSology는 웹기반의 프로그램으로 사용자는 웹사이트에 로그인하여 개별 파일 혹은 소프트웨어 패키지를 업로드할 수 있다. FOSSology는 업로드된 파일 내에 라이선스 텍스트와 Copyright 정보를 검출한다. 개발자는 사용하고자 하는 오픈소스의 라이선스가 무엇인지, Copyright은 어떻게 되는지에 대한 정보를 확인하고자 할때 FOSSology를 이용하는 것이 좋다. FOSSology는 개발자가 업로드한 오픈소스 패키지 내의 모든 파일을 스캔하여 각 파일 내 라이선스 관련 텍스트와 Copyright 정보를 자동으로 검출하고, 이를 리포트로 생성한다. FOSSology 주요 특징에 대한 자세한 내용은 다음 페이지를 참고할 수 있다. : https://www.fossology.org/features/

설 치

기업 내에서 FOSSology를 사용하기 위해서는 사내에 FOSSology 서버를 구축해야 한다. 이를 위해 리눅스 기반의 서버 시스템에 FOSSology를 설치해야 한다. FOSSology는 다음 세가지 방법으로 설치할 수 있다.

1. Docker 사용
2. Vagrant와 VirtualBox 사용
3. Source build하여 설치

여기서는 가장 간편한 방법인 Docker를 사용하는 방법에 대해 설명한다.
FOSSology는 컨테이너화된 Docker 이미지를 Docker Hub (https://hub.docker.com/) 를 통해 공개하고 있다. : https://hub.docker.com/r/fossology/fossology
Pre-built된 Docker 이미지는 다음 명령어를 사용하여 실행할 수 있다

```
$ docker run -p 8081:80 fossology/fossology
```

Docker 이미지는 다음 URL과 계정 정보로 사용할 수 있다.

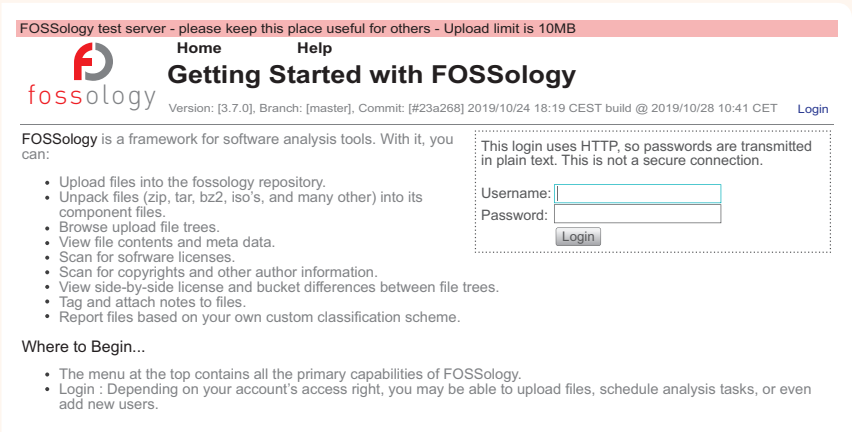
- http://IP_OF_DOCKER_HOST:8081/repo
- Username : fossy
- Passwd : fossy

설치와 관련한 자세한 내용은 다음 페이지를 참고할 수 있다.
(https://github.com/fossology/fossology/blob/master/README.md)

테스트 서버

FOSSology를 설치할 수 있는 시스템 구축이 곤란한 상황이라면, FOSSology Project에서 제공하는 테스트 서버를 이용할 수 있다. FOSSology 프로젝트에서는 테스트를 위한 환경을 제공한다. (테스트 서버는 예고없이 중단될 수 있다.) 사용자는 다음 계정으로 FOSSology 테스트 서버에 접속하여 FOSSology 기능을 시험해볼 수 있다.

- 테스트 서버 URL: http://83.169.21.23/fossology/
- Username: testuser
- Password: test



Basic Workflow

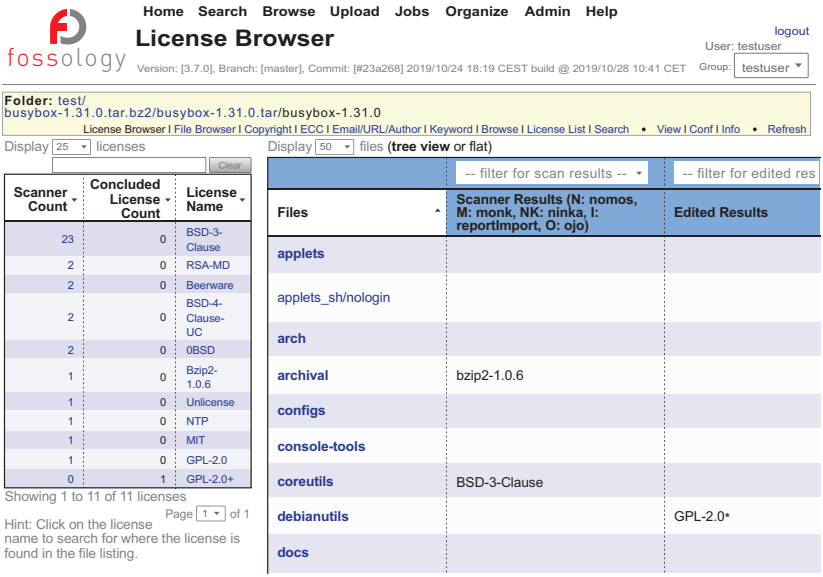
FOSSology의 기본 사용 절차는 다음과 같다.

- 사용하고자 하는 오픈소스의 라이선스와 Copyright 정보를 확인하기 위해 오픈소스 파일 혹은 패키지를 FOSSology에 업로드 한다.
- 메뉴 〉 Upload 〉 From File을 선택한다.

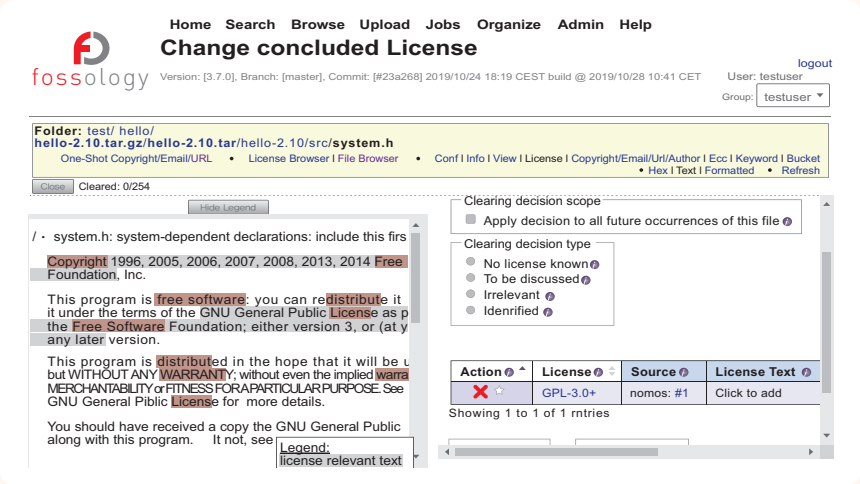
- 업로드할 파일을 선택하고 Upload 버튼을 클릭한다.
- 업로드가 완료되면 Job Agent에 의해 자동으로 분석을 수행한다.
- 메뉴 〉 Jobs 〉 My Recent Jobs에서 분석중인 Status를 확인할 수 있다.

	Wget (Example3(Upload From VCS))				
Job / Dependency	Status	wget			Average items / sec
27512	Completed	wget_agent	1 item	2019-07-08 14:47 - 2019-07-08 14:48	0.06 items/sec
27513 / 27512	Completed	ununpack	1,115 items	2019-07-08 14:48 - 2019-07-08 14:48	139 items/sec
27514 / 27513	Completed	sdj2nest	1,105 items	2019-07-08 14:48 - 2019-07-08 14:48	1108 items/sec
27515 / 27514	Completed	copyright	617 items	2019-07-08 14:48 - 2019-07-08 14:48	206 items/sec
27516 / 27514	Completed	ecc	839 items	2019-07-08 14:48 - 2019-07-08 14:48	420 items/sec
27517 / 27514	Completed	keyword	863 items	2019-07-08 14:48 - 2019-07-08 14:48	863 items/sec
27518 / 27514	Completed	mimetype	519 items	2019-07-08 14:48 - 2019-07-08 14:48	130 items/sec
27519 / 27514	Completed	monk	525 items	2019-07-08 14:48 - 2019-07-08 14:48	263 items/sec
27520 / 27514	Completed	nomos	525 items	2019-07-08 14:48 - 2019-07-08 14:48	16 items/sec
27521 / 27514	Completed	pkgagent	2 items	2019-07-08 14:48 - 2019-07-08 14:48	0.00 items/sec

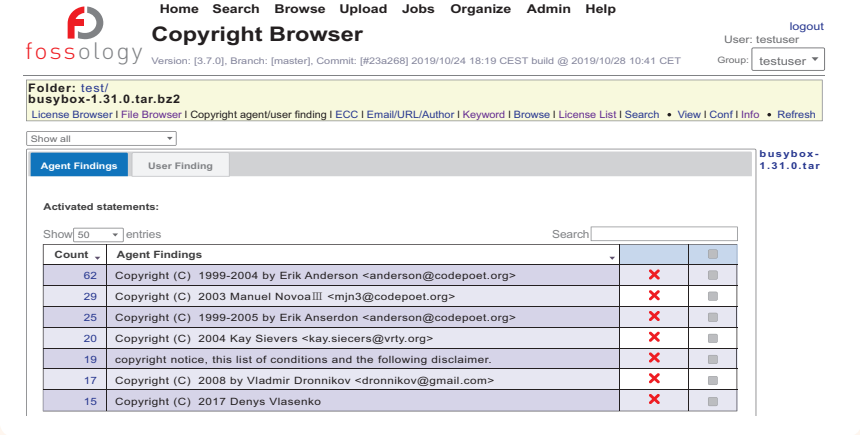
• 분석이 완료되면 메뉴 〉 Browse에서 분석 결과를 확인할 수 있다.



• 개별 파일을 선택하면 FOSSology가 검출한 라이선스 관련 텍스트가 무엇인지 확인할 수 있다.



• 메뉴 〉 Browser 〉 파일 혹은 디렉토리를 선택 〉 Copyright/Email/Url/Author에서는 FOSSology가 검출한 Copyright/Email/Url/Author 정보를 보여준다.



사용자는 FOSSology는 이렇게 분석한 결과가 유효한지 여부에 대해 확인 후 잘못 검출된 항목에 대해서는 분석결과에서 제외시키는 작업을 할 수 있다. FOSSology는 이를 Clearing 과정이라고 설명하며, 자세한 사항은 다음 페이지를 참고할 수 있다. : https://www.fossology.org/get-started/basic-workflow/

위와 같은 방법으로 사용하고자 하는 오픈소스의 라이선스는 무엇인지, Copyright 정보는 어떻게 되는지를 확인할 수 있다.

SW360

오픈소스를 포함하는 제품을 개발하고 배포하는 기업이라면 각 제품과 릴리스 버전마다 사용한 오픈소스의 버전, 라이선스 등의 정보를 수집하고 추적해야 한다. 이를 통해 기업은 올바른 오픈소스 컴플라이언스 활동을 수행할 수 있다.

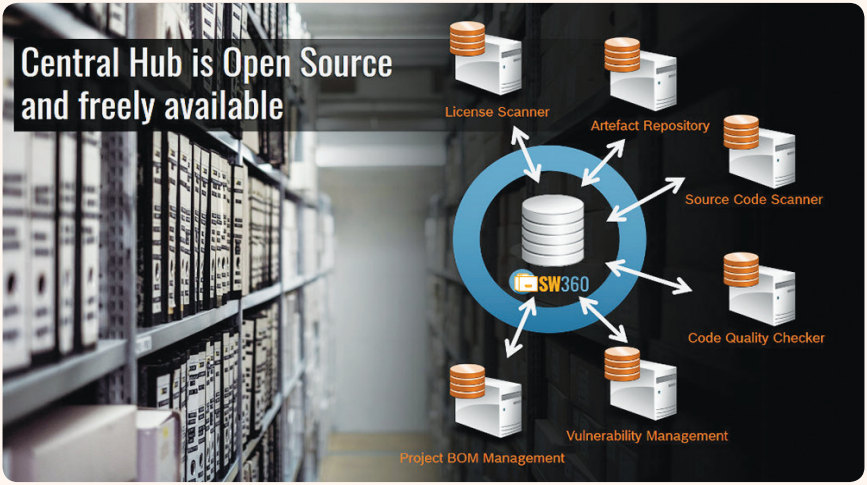
특히, NVD (https://nvd.nist.gov/vuln)에서 특정 오픈소스 버전에 보안 취약점이 보고되었을때, 해당 버전을 사용하고 있는 제품이 무엇인지 추적을 할 수 없다면, 그 기업은 어느 제품에 보안 패치를 적용해야 할 지 알 수 없는 상황에 처하게 되고, 그 기업의 제품들은 보안취약점에 그대로 노출이 될 수 밖에 없다.

이렇듯, 오픈소스 정보를 추적하는 활동은 꼭 필요하다. 기업들은 이를 위해 자체 시스템을 구축하거나, 상용 서비스를 구매하여 사용하기도 한다. SW360은 Eclipse 재단에서 후원하는 오픈소스로서 소프트웨어 BOM에 대한 정보를 수집 및 추적하기 위한 웹 애플리케이션 및 저장소를 제공한다. (https://www.eclipse.org/sw360/)

주요 특징

SW360은 웹기반의 UI를 제공하며 주요 기능은 다음과 같다.

- 제품에 사용된 컴포넌트 추적
- 보안 취약점 평가
- 라이선스 의무 관리
- 고지문 등 법적 문서 생성



(https://www.eclipse.org/sw360/)

설 치

SW360은 다음과 같이 구성된다.

- Frontend : Liferay-(Tomcat-)based portal application
- Backend : Tomcat-based thrift service
- Database : CouchDB

Project 구조와 설치를 위해 요구되는 소프트웨어 등 자세한 내용은 README에서 확인할 수 있다. : https://github.com/eclipse/sw360/blob/master/README.md

SW360은 다음 세가지의 설치 방법을 제공한다. 사용자는 이 중 하나를 선택하여 설치할 수 있다.

1. Vagrant (https://www.vagrantup.com/) 기반 설치 : Vagrant는 가상화 인스턴스를 관리하는 도구로서 sw360vagrant에서는 SW360을 한번에 Deploy하기 위한 환경을 제공한다. : https://github.com/sw360/sw360vagrant
2. SW360의 구성요소를 개별적으로 설치할 수 있다. : https://github.com/eclipse/sw360
3. Docker를 통해 Deploy할 수 있다. : https://github.com/sw360/sw360chores

여기서는 CentOS 7.6 시스템에 Vagrant 기반으로 설치하여 Deploy하는 방법을 소개한다. 자세한 사항은 README를 참고한다. : https://github.com/sw360/sw360vagrant/blob/master/README.md

1) 사전 설치

vagrant box 에 SW360을 설치하기 위해서는 openjdk, VirtualBox 및 Vagrant를 설치해야 한다.

- 먼저 openjdk 1.8.0을 설치한다.

```
$ yum install java-1.8.0-openjdk
$ java -version
openjdk version "1.8.0_191"
OpenJDK Runtime Environment (build 1.8.0_191-b12)"
OpenJDK 64-Bit Server VM (build 25.191-b12, mixed mode)
```

- VirtualBox를 설치한다.

```
$ sudo wget https://download.virtualbox.org/virtualbox/rpm/el/virtualbox.repo -P /etc/yum.repos.d
$ sudo yum install VirtualBox-5.2
```

- CentOS 7에서 VirtualBox 설치 시, "kernel module is not loaded" 에러가 발생할 경우, kernel-devel을 설치하여 해결한 후 VirtualBox를 재설치한다.

```
$ sudo yum install https://centos7.iuscommunity.org/ius-release.rpm
$ sudo yum install dkms
$ sudo yum install kernel-devel
# reboot
$ sudo /sbin/vboxconfig
$ systemctl status vboxdrv
```

· vboxdrv.service - VirtualBox Linux kernel module
Loaded: loaded (/usr/lib/virtualbox/vboxdrv.sh; enabled; vendor preset: disabled)
Active: active (exited) since Wed 2020-02-19 09:06:02 KST; 20min ago

• Vagrant와 vagrant-aws plugin을 설치한다.

```
$ sudo yum install https://releases.hashicorp.com/vagrant/2.2.6/vagrant_2.2.6_x86_64.rpm
# vagrant-aws plugin 설치
$ vagrant plugin install vagrant-aws
```

• 그리고, sw360vagrant 코드를 Clone해온다.

```
$ git clone https://github.com/sw360/sw360vagrant.git
```

2) Dependency 다운로드

• Vagrant box를 빌드하는 시간을 줄이기 위해 Dependency Package들을 미리 다운로드 받는다.

```
$ cd sw360vagrant
$ ./download-packages.sh
```

• 그러면 다음의 package들이 ./shared/package 폴더 안에 다운로드 된다.

- Liferay 7.2.1 CE GA2 with Tomcat (9.0.17)
- Postgresql-42.2.9 ODBC client for Java as *.jar file
- SW 360에서 필요한 11개의 *.jar 파일
- Thrift 0.11
- A box images from the Ubuntu 16.04 LTS (xenial-server-cloudimg-amd64-vagrant.box)

3) Base box 생성

• 이제 다음 명령어로 Base box를 생성한다.

```
$ cd generate-box
$ ./generate_box.sh
```

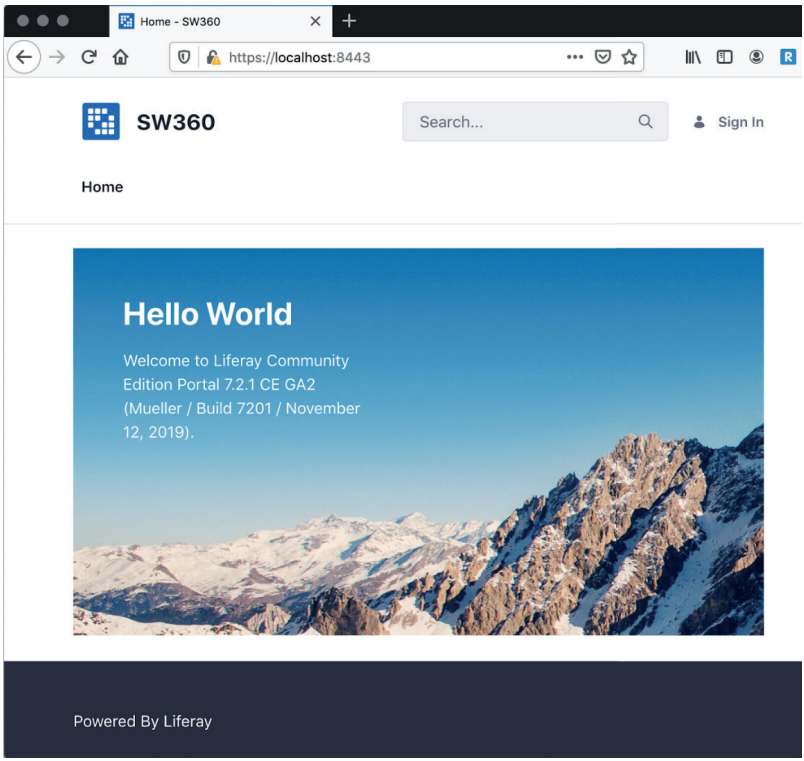
• 이 작업은 시간이 수십분 소요될 수 있다.

4) Box 실행

다음 명령어로 Box를 실행한다.

```
# If you have built a vagrant box from this directory earlier, you will have to destroy it first via
$ vagrant destroy
$ cd ../sw360-single
$ vagrant up
```

- Box를 실행하면 liferay, postgresql 및 couchdb가 구성된다. 이상없이 실행이 될 경우, https://localhost:8443/ 로 Liferay 화면에 접근할 수 있다.



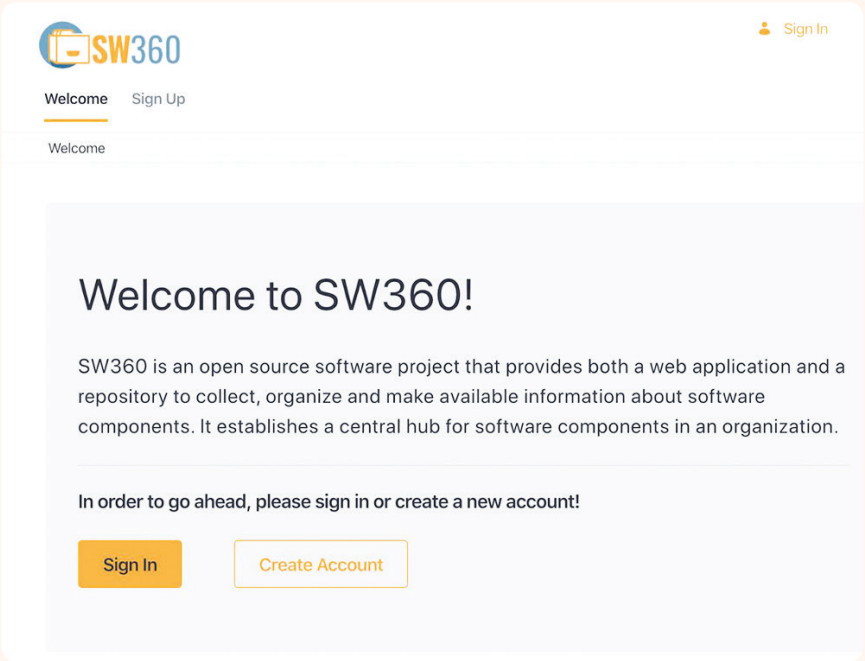
5) SW360 Layout Deploy

마지막 단계는 Liferay에서 SW360의 Layout을 Deploy하는 것이다. 이 작업은 아직 자동화가 되지 않아 관리자가 수동으로 수행해야 한다. https://localhost:8443/에 접근하여 다음 계정으로 로그인한다.

- id : setup@sw360.org
- pw : sw360fossy

이후에는 다음 사이트의 안내에 따라 Layout deploy를 수행한다. : https://github.com/eclipse/sw360/wiki/Deploy-Liferay7

Deploy가 완료되면 다음과 같은 화면을 볼 수 있다.



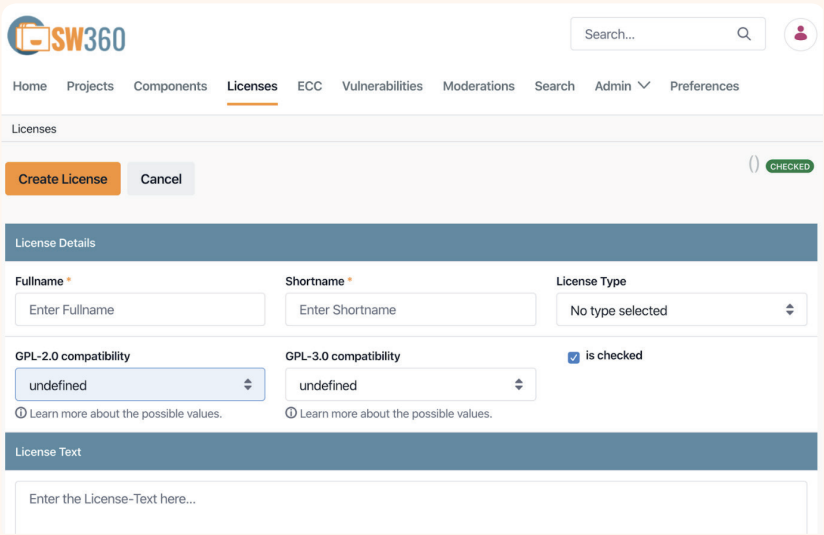
Basic Workflow

1) License 등록

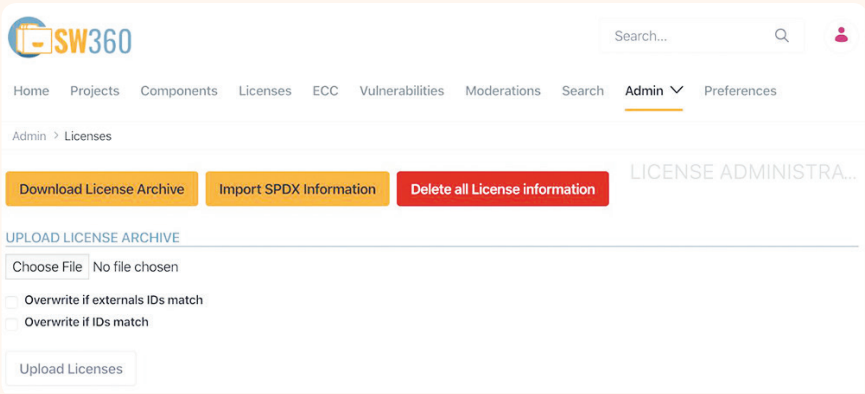
SW360을 처음 설치하면 자주 사용하는 오픈소스 라이선스 들을 등록해야 한다. 라이선스 등록 시에는 다음과 같은 정보를 포함한다.

- Full Name
- Short Name
- License Type
- GPL-2.0 Compatibility (예: yes, no)
- License Text

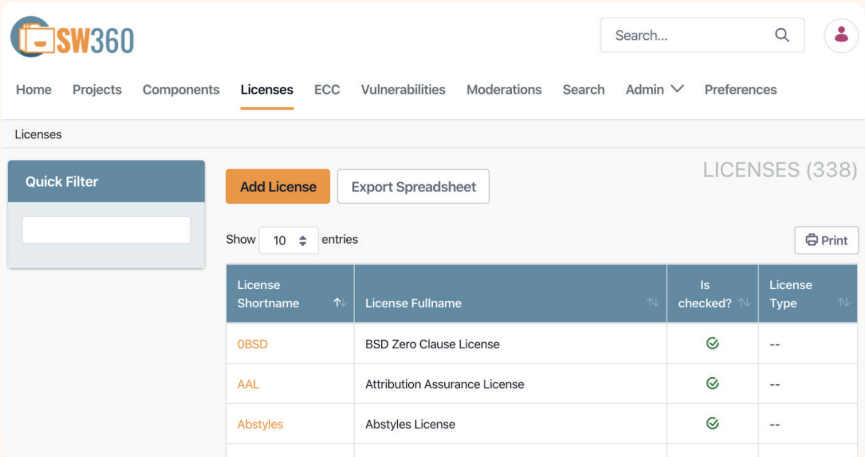
메뉴 〉 Licenses 〉 Add License를 선택하면 다음과 같이 Create License 화면으로 진입한다.



이와 같이 라이선스를 하나씩 수동으로 등록하는 일은 상당히 수고스러울 수 있는데, 다행히 SW360은 SPDX License List를 한번에 Import하는 기능을 제공한다. 메뉴 〉 Admin 〉 Import SPDX Information을 클릭한다.



그러면, 곧 SPDX License List가 자동으로 등록된다. 메뉴 〉 Licenses 에서 338개의 License가 등록된 것을 확인할 수 있다.



2) Component 및 Release 등록

SW360에서 Component는 하나의 소프트웨어 단위이다. 여기에는 다양한 형태의 소프트웨어가 해당할 수 있으며, 그 예는 다음과 같다.

- 오픈소스 소프트웨어
- 라이브러리
- 3rd party 소프트웨어

Component는 다음과 같은 정보를 포함한다.

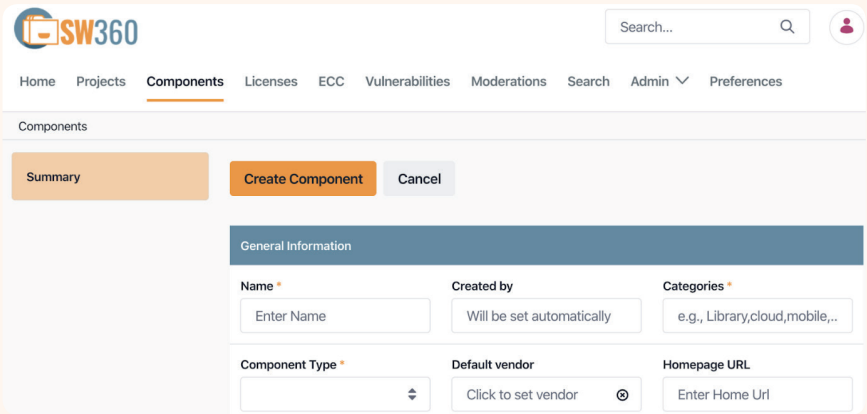
- Component Name
- Main Licenses
- Categories (예: Library, Cloud, Mobile, ...)
- Component Type (예: OSS, Internal, InnerSource, Service, Freeware)
- Default Vendor
- Homepage URL

Release는 Component에서 하나의 Version을 가리키는 단위이다. 따라서 하나의 Component는 여러개의 Release를 가질 수 있다. Release는 하나의 Component 하위에 생성되어 관리된다.

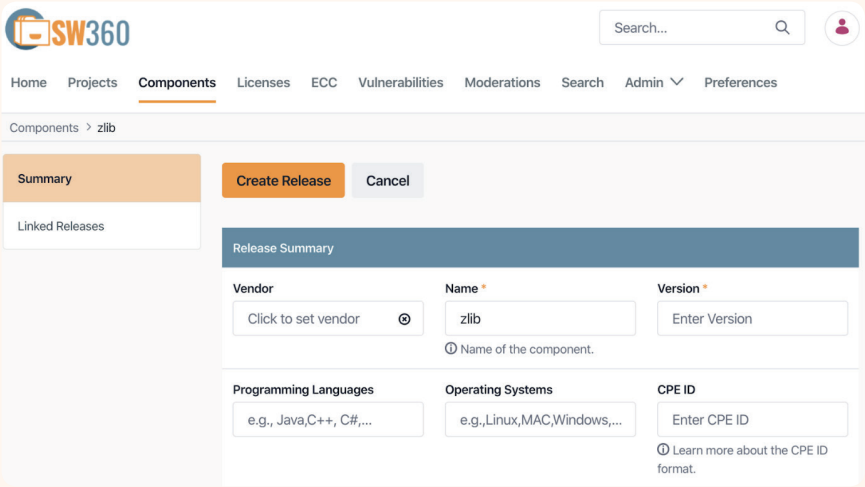
Release는 다음과 같은 정보들을 포함한다.

- Component Name
- Version
- License
- Download URL
- CPE ID (예: cpe:2.3:a:apache:maven:3.0.4)

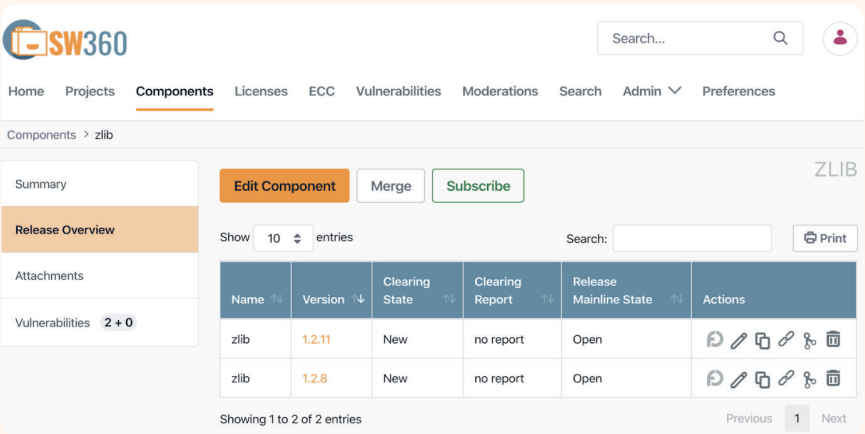
예를 들어, zlib-1.2.8을 등록해야 한다면, 먼저 Component에 zlib을 먼저 등록한 후, Release에 zlib 1.2.8을 등록한다. Menu 〉 Components 〉 Add Component를 선택하면 Create Component 화면으로 진입하여 zlib에 대한 정보를 등록할 수 있다.



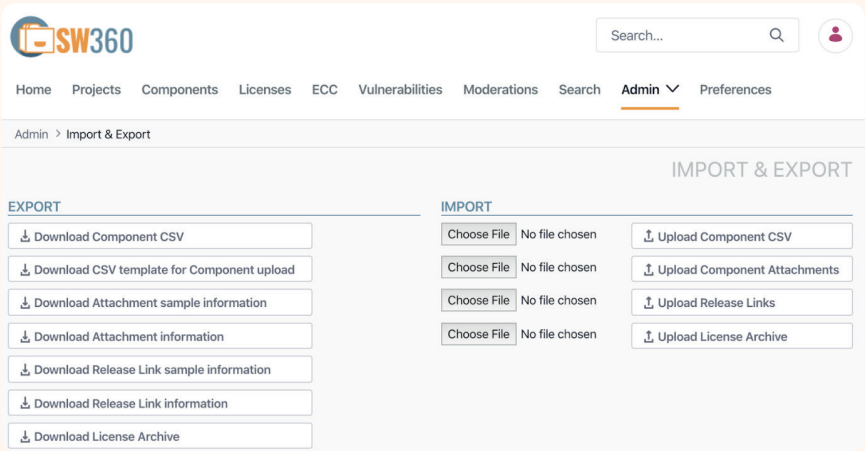
Component를 생성하면, Components 〉 Releases 〉 Add Release에서 zlib-1.2.8 version에 대한 정보를 등록할 수 있다.



하나의 zlib이라는 Component에 1.2.8과 1.2.11 version을 각각의 Release로 등록하였을때, Release Overview 화면에서 다음과 같이 2개의 Release가 존재하는 것을 볼 수 있다.



SW360은 다수의 Component 정보를 Import시키기 위한 기능을 제공한다. 메뉴 〉 Admin 〉 Import / Export에 CSV template에 등록을 원하는 Component 정보를 입력 후 Import 할 수 있다.



단, 이 기능은 2020년 2월 기준 아직 안정적으로 동작하지 않을 수 있다.

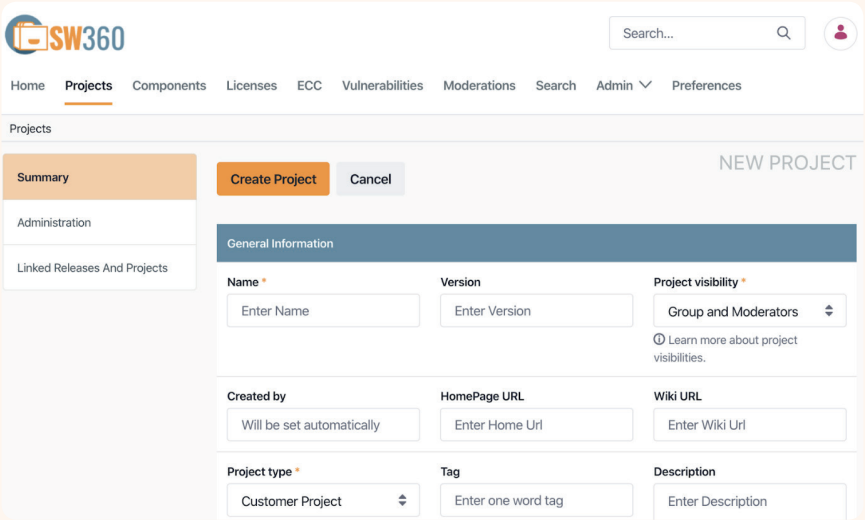
3) Project 생성

Project는 하나의 제품을 가리킨다. 사업 유형에 따라 제품일수도 있고, 서비스 혹은 소프트웨어 일수도 있다. Project에는 제품에 사용된 Component/Release를 등록하여 관리한다.

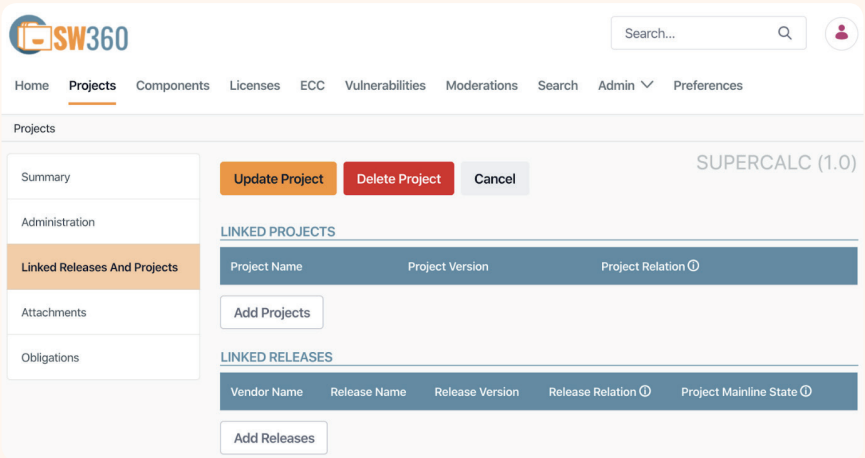
Project 생성 시에는 다음과 같은 정보를 등록한다.

- Project Name
- Version
- Project type (예: Product, Customer Project, Service, Intenal Project, InnerSource)

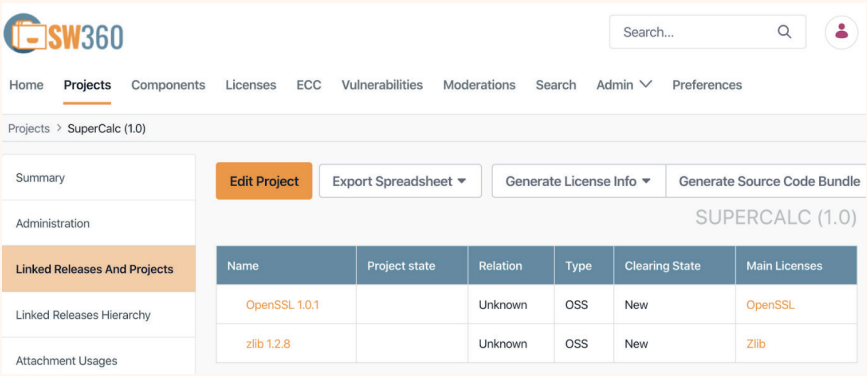
메뉴 〉 Projects 〉 Add Project를 통해 Project를 생성할 수 있다.



Project를 생성하고 나면, 포함하는 Release나 하위 Project를 등록한다. 메뉴 〉 Projects에서 해당 Project를 선택하면 "Linked Releases and Projects" 에서 Linked Projects와 Linked Releases를 등록할 수 있다.

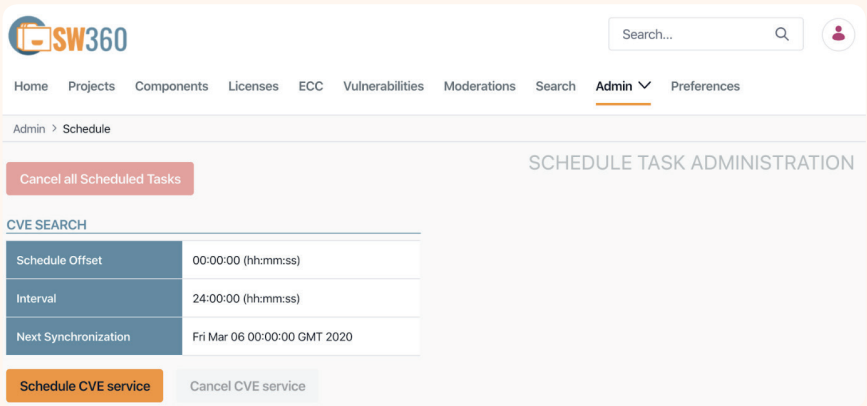


다음은 SuperCalc라는 Project에 OpenSSL 1.0.1과 zlib 1.2.8을 Linked Releases로 등록한 이후의 화면이다.

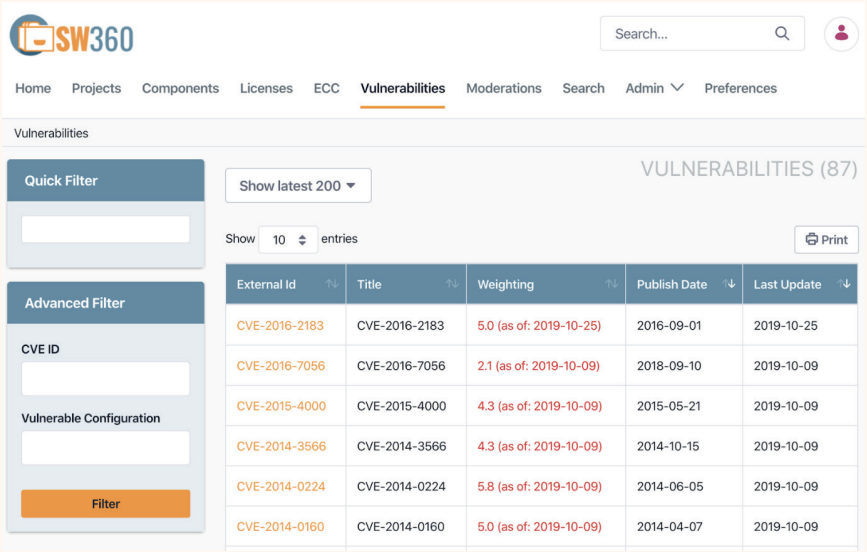


4) 보안취약점 관리

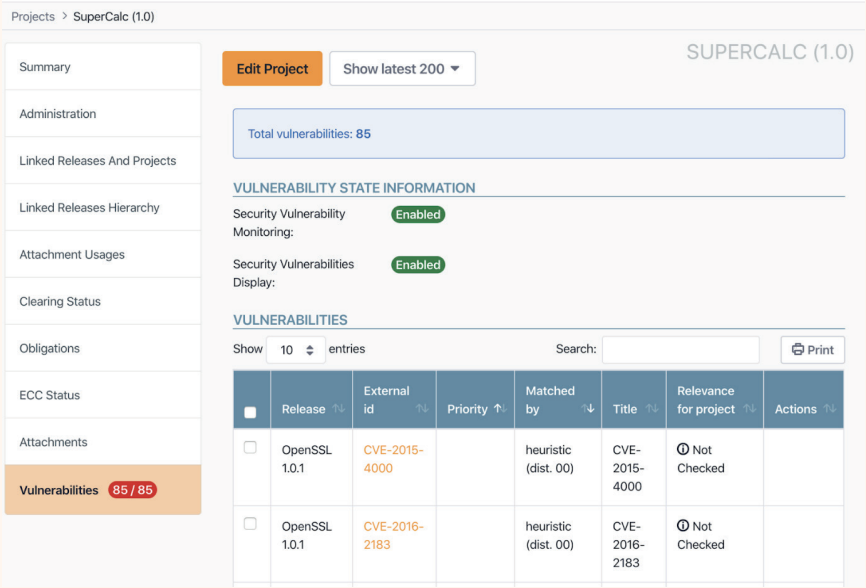
SW360은 등록된 Release에 대해 보안취약점이 있는지 자동으로 확인할 수 있다. 이를 위해 SW360은 CVE 정보를 주기적으로 수집하도록 스케쥴링하는 기능을 제공한다. 메뉴 〉 Admin 〉 Schedule 에서 CVE SEARCH 정보를 24시간마다 수집하도록 스케쥴링을 설정할 수 있다.



이렇게 스케쥴링을 설정하면 SW360은 정해진 시간에 CVE Search 사이트(https://cve.circl.lu/)에서 CVE 정보를 수집한다. 수집한 CVE 정보는 메뉴 〉 Vulnerabilities 에서 확인할 수 있다.



이렇게 Vulnerabilities 정보가 수집된 이후에는 생성한 Project에 보안취약점이 있는지 조회할 수 있다. 위에서 생성한 SuperCalc Project에서는 85개의 보안취약점이 보고된 것을 확인할 수 있다.



이와 같은 방법으로 기업에서 개발/배포하는 소프트웨어를 SW360에 등록하여 관리한다면, 오픈소스 컴플라이언스 뿐만 아니라 보안취약점에 대해서도 리스크를 최소화 할 수 있는 형태로 관리가 가능하다.

또한 SW360은 위와 같은 Web Interface 뿐만 아니라 대부분의 기능을 REST API로 제공하여서 FOSSology 등의 다른 도구와의 연동이 가능하다. : https://github.com/eclipse/sw360/wiki/Dev-REST-API
즉, 소스 코드 스캐닝 도구의 분석 결과를 SW360에 Import 시키는 등의 방법으로 DevOps에 Integration 시켜서 Project, Release 등록을 자동화시켜서 관리한다면 효율성이 크게 증가될 것이다.

기업 공개SW 거버넌스 가이드

OpenChain 2.0 해설

2020년 1월 20일 1판 1쇄

발행인 김 창 용

발행처 정보통신산업진흥원
27872 충북 진천군 덕산면 정통로 10
TEL. 043-931-5000 FAX. 043-931-5129

디자인·인쇄 (사)한국장애인유권자연맹 인쇄사업부
TEL. 02-325-1585