

누구를 위한 SBoM 공급망 보안 기술인가, 그 올바른 해결책은?

Who is the SBoM Supply Chain security technology for,
and what is the proper solution?

조용준 기술이사

엘에스웨어(주)

eugene@lsware.com



CONTENTS

- 01 오픈소스 SW, 소프트웨어 공급망, 그리고 SBoM의 등장
- 02 보안 취약점 추적과 SBoM
- 03 현재의 SBoM 취약점 기술 연구방향의 문제점
- 04 사용자 지향 SBoM 기반 선제적 취약점 관리 기술
- 05 결론

개요

- SBoM은 소프트웨어 공급망에서 **정확한 정보를 전달**하기 위해서
- SBoM을 활용하면 취약점을 추적할 수 있다는 점에 주목
- 하지만, 현재 연구는 취약점 추적에는 주목하고 있지 않음
 - SBoM을 만드는데 까지만 주목하거나
 - 물론, 현 상황에서 SBoM을 만드는 기반이 없었다는 점은 고려할 요소
 - 그렇다고 사용 방법을 선제적으로 연구하지 않으면 의미가 없음
 - 개발자 측면에만 주목하고, 사용자 측면을 주목하지 않음
 - 개발자에게도 필요하지만, SBoM은 사용자에게 유용한 정보
 - 특히, 우리나라 및 SI 사업이 많은 국가에서는, 사용자가 취약점을 모니터링 할 수 있어야 함
- SBoM을 제대로 활용할 수 있는 올바른 방향성을 잡아야 함
- 취약점 발견 시, 사용자가 직접 대응할 수 있는 기술을 개발해야

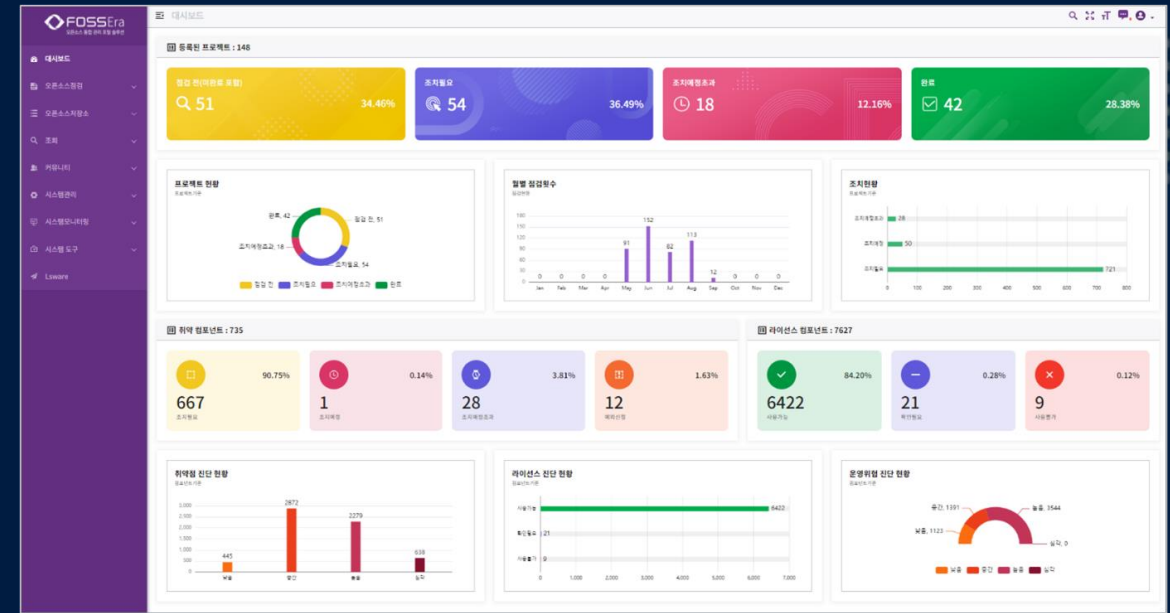
개요

- 현재 이루어지고 있는 연구를 완전히 부정하는 것은 아님
 - 기존 취약점 대응 기술 연구도 오픈소스의 다양한 활용 방식을 고려했을 때, 보완적 기술로써 필요함
 - 하지만, 가장 대표적·절대 다수의 활용 사례에 알맞지 않은 기술
- 하지만, SBoM을 가장 잘 활용할 수 있는 취약점 대응 방식은 아님
- 한국, 관공서 등의 SI 프로젝트 등에서 활용할 수 있는 현실적인 기술이어야 함
 - 연구실 기술이나 대기업 개발사 등에서는 적용해볼 수 있지만, 일반 중소기업에서 활용 불가능한 기술이라면, 가치가 매우 낮음

엘에스웨어(주)

• 포세라 (FOSSEra) 솔루션

- 소프트웨어개발수명주기 별 오픈소스 점검
- 소프트웨어 구성요소 분석 및 SBoM 생성
- 폐쇄망 환경에서 오픈소스 컴포넌트 자동 반입
- CI/CD 연동을 통한 오픈소스 라이선스 자동 점검



• 금융권·대기업 등의 오픈소스 컴플라이언스 시스템 구축중

• 오픈소스 컴플라이언스 컨설팅

- 공공분야 30 여 건, 민간분야 110 여 건 이상

• 오픈체인 인증 획득 (2023년)

- ISO/IEC 5230 Conformant Programs

OPENCHAIN



01

**오픈소스 SW, 소프트웨어 공급망,
그리고 SBoM의 등장**



오픈소스 SW와 소프트웨어 공급망

오픈소스 소프트웨어:
어디에선가 나타난 무료 소프트웨어?

(공급망) 신뢰성 문제

라이선스 파일만 있으면 오픈소스 소프트웨어?
누가 만든 소프트웨어인데?
어디서 가져오고, 어떤 경위로 취득한 소프트웨어인데?



소프트웨어 공급망
(SW Supply Chain)

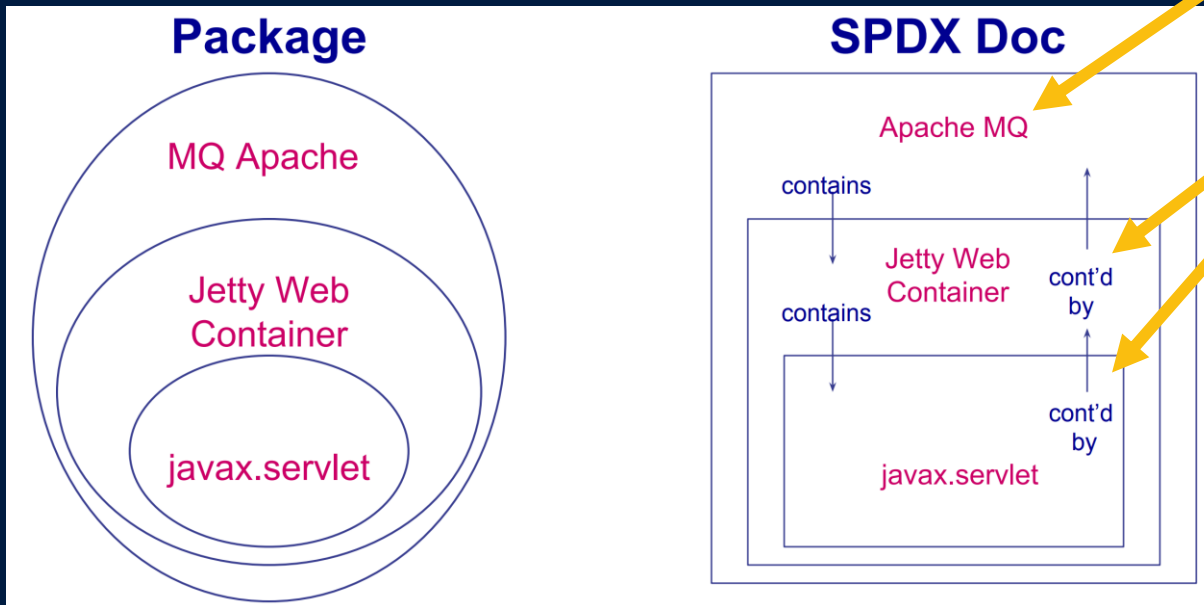


01 소프트웨어 공급망과 SBoM의 등장

- SW를 혼자서 만드는 시대가 아님
 - (작은 library 정도를 제외하고) 다른 SW에 의존하지 않는 SW는 거의 없음
- 해당 SW의 **구성 SW 정보**가 있어야, 해당 SW를 신뢰할 수 있음
 - **소프트웨어 공급망**: 어떤 SW를 구성하는 SW의 공급 구조
 - 이 과정을 최종 산출물 SW를 만들때까지 반복적으로
→ **소프트웨어 공급망 추적**
- 이 소프트웨어 공급망 상에서,
공급망 정보를 제공하기 위해 SBoM(Software Bill of Material) 등장
 - SPDX, CycloneDX, SWID, etc.

예: SPDX (Software Package Data eXchange)

- SBoM의 한 종류
- 구성 요소의 정보를 포괄적으로 포함해서 전달할 수 있음
- 주로 라이선스 정보 전달을 위해 개발



The SPDX Document

SPDX v2.0 File

Document Creation Information

Package Information

File Information

Other Licensing Information

Relationships

Annotations

(SPDX 소개문서: SPDX 2.0 what, why, how & specifics
https://wiki.spdx.org/images/SPDX_2.0_Collab_Presentation.pdf)

다음으로 진행하기 전에, **소프트웨어 공급망 정의**

- A. 소프트웨어 개발을 위한 **SW 개발 공급망**
 - OSS → 컴포넌트 개발사 → ... → 패키지 개발사
- B. 소프트웨어 패키지 공급을 위한 **SW 공급망**
 - OSS → 컴포넌트 개발사 → ... → 패키지 개발사 → 사용자(End-user)
- **SW 공급망 취약점 문제에서,
피해 당사자인 사용자를 빼놓고 이야기 할 수 없음**



02

보안 취약점 추적과 SBOM

Window of vulnerability → Zero-Day Attack

- O.D.: An opportunity to attack something that is at risk
- 발견 뒤 대응할 때 까지의 간격
- 짧을 수록 좋음
 - 제로데이 공격은 최대한 빨리 공격해서 대응하기 까지의 간격을 늘리는 방식
- 지금까지는, 개발자 측면에서만 다루어져 왔음
 - 발견 후 얼마나 빨리 패치를 만들었는가?
- 하지만, 실제 피해는 사용자에게서 일어나기 때문에, 사용자의 대응까지의 간격을 줄이는 데 주목해야 함
 - 패치가 만들어진 후에도 사용자 대응까지 45일에서 5년까지 걸린다는 연구 결과가 있음



(오픈소스) 소프트웨어 취약점 문제

- 문제가 커지는 원인 #1: SW 재사용
 - 오픈소스 SW는 더 재사용하기 좋으니까, 파장도 더 큼
 - 문제가 커지는 원인 #2: 사후 발견 취약점
 - 사용했을 때는 있는지도 몰랐던 취약점
 - 있는 줄 알았으면, 당연히 고쳐서 사용하니까
 - 하지만, 사용하고 난 다음, 한참 뒤에 발견됨
 - 발견된 시점에는 그 취약점이 있는 SW는 수많은 소프트웨어에서 사용되고 있음
- ➔ 해결을 위해서는
(소프트웨어 공급망 상에서) 취약 SW를 추적할 필요가 있음

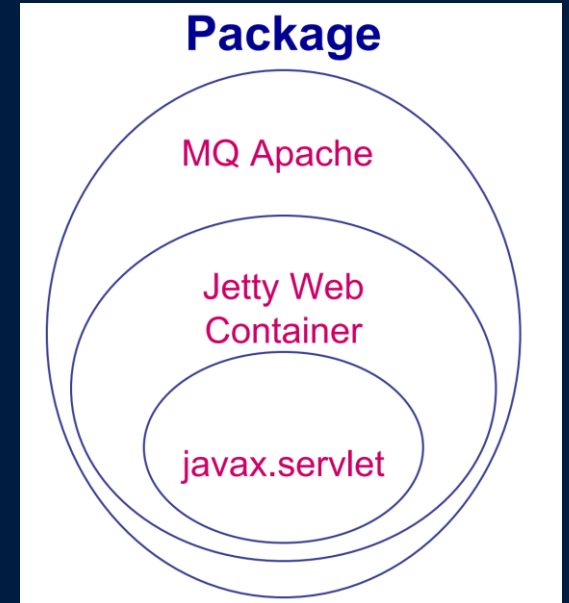


(오픈소스) 소프트웨어 취약점 문제

- Nested Package

- 안에 들어있는 SW를 인지할 수 없는 문제
 - 안에 들어있는 SW는 보통 오픈소스 SW
- 예: 내가 가져다 쓴 소프트웨어가 MQ Apache면, Jetty Web Container에서 취약점이 발견되어도 개발자조차 취약점을 인지하지 못할 수 있음

내가 가져다 쓴 SW는 MQ Apache이지, Jetty Web Container가 아닌데요???



→ 어떤 SW를 구성하는 모든 소프트웨어의 정보가 필요

SBoM이 있으면 가능하지 않은가?



03

현재의 SBoM 취약점 기술 연구방향의 문제점



연구배경

- 한국 및 관공서, 공공기관의 SW 발주 방식: SI
 - 소규모 개발사가 개발
 - 각 개발사가 취약점을 지속적·반복적으로 관리할 수 있는 역량이 부족함
 - → 최소 유지보수 기간(1년)이 지나면 아무도 신경쓰지 못함
 - 어떤 오픈소스를 가져다 썼는지, 어떤 오픈소스에 문제가 발생했는지 등..
 - 사용자(관공서, 공공기관)는 소스코드를 가지고 있지 않음
 - 기존에는 요구하던 관행이 있었지만, 현재는 금지됨

사용자가 직접 소스코드를 분석하는 것도 아니고, 그저 다른 기업에게 재하청줄 뿐이라서 저작권법·공정거래법 위반

➔ 이런 상황에서, 사용자에게 취약점에 대응하기 위한 수단은 있는가?



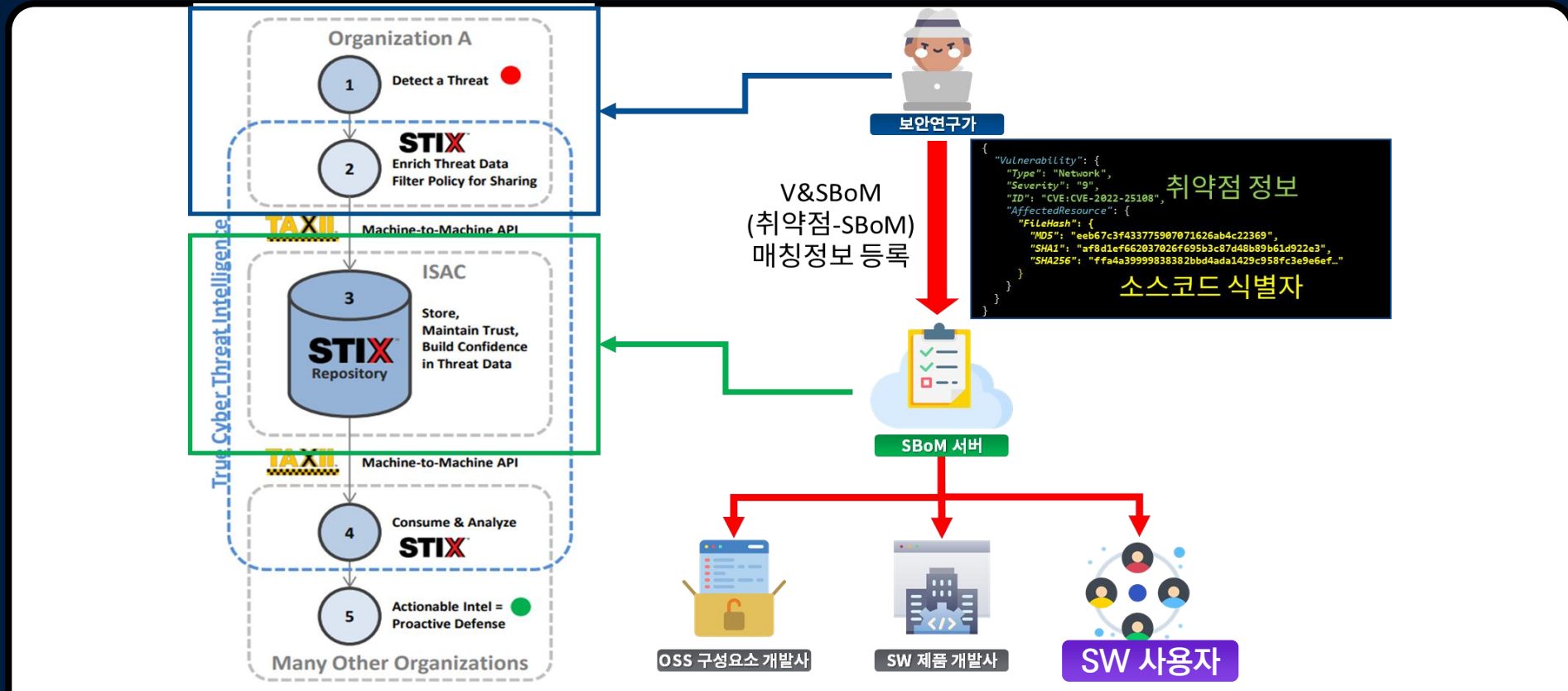
03 소프트웨어 취약점 추적 방법

- SCA(Software Component Analysis)
 - 주로 오픈소스 SW 구성요소 분석에 중점
- 소스코드 유사성 분석 기법
- 동적·정적 분석 기법
 - 전통적인 취약점 분석 기법
 - + DLL, Shared Object 방식으로 배포된 경우, 적용이 어려움
 - 아예 불가능한 건 아닌데, 매우 비싼 솔루션+DB를 구입한 개발사만 가능
 - → 소수의 대기업 개발사나 가능함
- 소스코드를 얻을 수 있는 경우에만 가능 → 사용자는 사용할 수 없는 방법
- 이런 접근 방식의 연구도 SBoM 취약점 추적 방식으로 포장됨

03

소프트웨어 취약점 추적 방법

- SBoM 기반 추적 방식
- SW 구입·공급 시 함께 공급받는 SBoM 정보에 기반하여 취약 SW를 포함하고 있는지 검사하는 기술





04

사용자 지향 SBoM 기반 선제적 취약점 관리 기술



SBoM에 의한 취약점 추적·대응

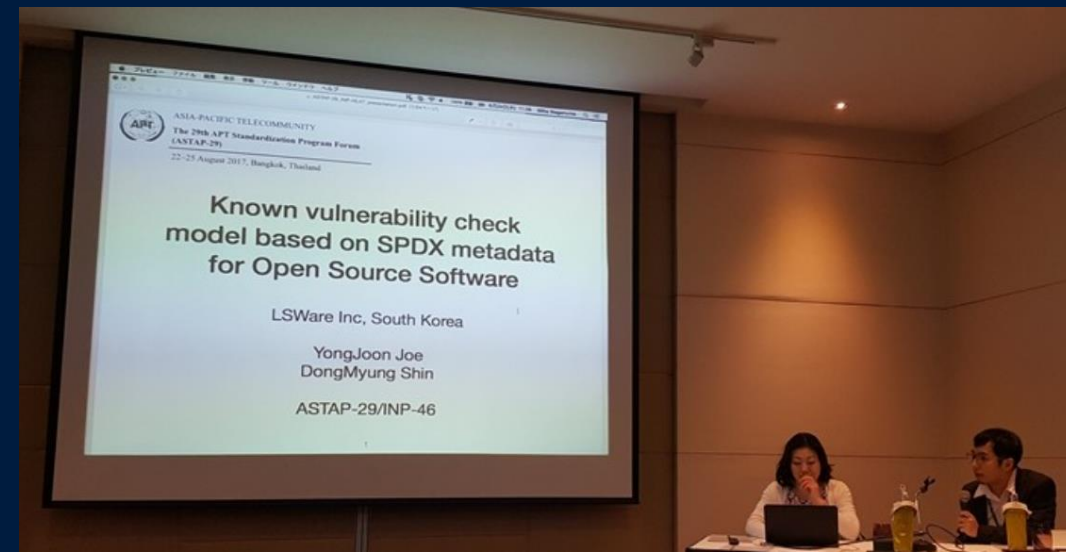
- 취약점 발견자는
취약점이 발견된 Package 정보 / Source Code의 Hash 값을
broadcasting
- 사용자(&개발자)는
 - SBoM에 기록된 Package/Source Code Hash 정보에 바탕하여
자신의 자산 내에 취약점이 발견된 SW가 포함되었는지 추적
 - 취약점이 있는 SW를 발견하면, 자체 보안 정책에 기반하여 대응
 - 알림
 - 네트워크 차단
 - 개발사에 패치 요청
 - Etc.



표준화: SBoM을 이용한 사후 취약점 관리

Security Guideline of Open Source Software (2017~2021, ASTAP-REPT-49)
에서 제안해온 내용

1. 개발사: SBoM을 사용자(end-user)에게 제공
 - SBoM 생성을 위해서는, 개발 공급망 내에서도 각 개발사 사이에 SBoM 상호 교환이 필요
2. 사용자: SBoM monitor에 공급받은(Update된) SW의 SBoM을 등록
3. 취약점 발견자: 발견한 취약점을 SBoM-Vulnerability 매칭 정보 형태로 발신
4. 사용자(≠개발사):
 - SBoM monitor가 등록된 SBoM과 수신한 정보를 매칭
 - 사용자가 자신이 사용하고 있는 SW의 취약점을 **즉각적으로·직접** 발견
 - 사용자 자신이 할 수 있는 대응을 바로 함
 - 개발사에게 패치를 요청





05

결론



결론

- 오픈소스 SW의 취약점 정보는 공개되지만, 어디에 쓰였는지 추적이 안되고 있었음
- 오픈소스 SW 라이선스 추적을 위해서 사용되던 SBoM이 취약점 추적을 위해서 주목되기 시작함
- 특히, 한국의 경우, SBoM에 의한 추적이 매우 유용함
- 하지만,
현재까지의 국내 SBoM 취약점 연구는 SBoM을 활용하는 데에 초점을 맞추고 있지 않음
- 사용자를 위한 SBoM 기반 선제적 취약점 대응을 소개

감사합니다.

누구를 위한 SBoM 공급망 보안 기술인가,
그 올바른 해결책은?

ETRI
OPEN
SOURCE
TECH DAY
2023