최신보안동향 - SW 공급망 보안 [19반]정민석_7000

본 고는 과학기술정보통신부, 국가정보원, 디지털플랫폼정부위원회가 합동으로 발간한 '소프트웨어 공급망 보안 가이드라인 1.0'에 근거하여 SW 공급망 보안이 대두되는 환경적 변화, 실제 피해 사례, 대응책을 정리하는 것을 목적으로 합니다.

SW 공급의 주체가 늘어나고 분업화되는 추세입니다. 기존에는 SW 생산 주체가 최종 생산자 밖에 없었다면, 현재는 분업화된 여러 생산자가 있습니다. 분업된 생산자에는 오픈소스 개발자, 상용 SDK 생산 업체, SI 업체 등이 있을 수 있겠습니다. 이렇게 분업된 생산자는 최종 생산자와 연결되며, 최종생산자의 SW는 분업된 생산자의 SW에 의존합니다. 즉, 분업된 생산자의 SW의 보안 취약점이 발생한다면, 최종생산자의 SW의 보안 취약점으로 귀결될 수 있습니다.

실제 공급망의 취약성으로 인한 피해사례를 살펴보겠습니다. 먼저 Log4Shell(CVE-2021-44228)의 사례가 있습니다. JAVA 로깅 라이브러리에 원격 코드 실행을 허용 수 있는 취약점으로 인하여, 해당 라이브러리를 사용하는 최종 생산자의 SW 또한 취약해지는 사례가 있었습니다. 이 점은 JAVA 로깅 라이브러리라는 분업된 생산자 SW에서의 취약점으로 인하여, 최종 생산자의 SW 또한 취약해질 수 있다는 점을 시사합니다.

최종 생산자의 SW 뿐만 아니라 생산자의 생산 인프라에도 취약점이 발생할 수 있습니다. XZ Utils의 liblzma 라이브러리에서 발생한 취약점(CVE-2024-3094)은 특정 리눅스 필수 패키지에 포함되어 있었던 오픈소스 라이브러리입니다. 해당 취약점 사례는 오픈소스 개발자로 인하여 발생하였습니다. 해당 오픈소스 개발자는 2021년부터 오픈소스 프로젝트에 참여하며, XZ 프로젝트 관련자로 등극하였습니다. 그리고 2024년 악의적으로 SSH 백도어를 심었습니다. 이 사례는 소프트웨어 공급의 분업화가 생산 인프라의 취약점으로 이어질 수 있다는 것을 상기시킵니다. 더하여 악의적인 오픈소스 SW 개발자로 인하여 발생할 수 있는 취약점 또한 고려해야 한다는 점을 시사합니다.

이러한 공급망 취약점에 대한 대응책에 대해 살펴보겠습니다. 먼저 '소프트웨어 공급망 보안 가이드라인 1.0'에서는 SBOM을 제시합니다. SBOM은 SW 재료의 목록을 명세한 것입니다. 다시말해 타사의 어떤 SW를 사용했는지, 어떤 SW에 의존하는지를 명세한 것입니다. SBOM을 통하여 알려진 보안취약점을 관리하고, 리스크를 관리하는 것에 이점을 가지게 됩니다.

저는 오픈소스 SW 공급망 취약점에 대하여 주목하려 합니다. 그 중에서도 오픈소스 SW의 신뢰성을 어떻게 파악할지에 대한 기준을 제시하고 싶습니다. 최종 생산자가 오픈소스 SW를 사용할 때에 있어 그 신뢰도를 판단할 수 있는 기준을 제시하고, 신뢰도 높은 오픈소스 SW를 사용하도록 유도한다면, 최종 생산자의 취약성은 줄어듭니다. "신뢰할 수 있는 개발자인지?", "오픈소스 SW 자체의 의존성 목록은 취약성이 없는지?" 등을 종

합적으로 판단하는 기준을 제시하여 안전한 보안/개발 생태계에 기여하고 싶습니다. 다만 저는 아직 SW 신뢰도를 판단하는 정책을 학습하지 못하였습니다. 기존의 정책을 학습하여 오픈소스 SW의 신뢰도를 판단할 수 있는 기준을 제시할 수 있도록 노력하겠습니다.

[참고 자료]

과학기술정보통신부, 국가정보원, 디지털플랫폼정부위원회(2024.05.12) 소프트웨어 공급망 보안 가이드라인 1.0
(https://www.msit.go.kr/bbs/view.do?sCode=user&mId=113&mPid=238&bbsSeqNo=94&nttSeqNo=3184474)