特集
Special Feature

［DX（デジタル・トランスフォーメーション）時代のサプライチェーン・セキュリティ］

7 ソフトウェア部品表（SBOM）に基づくリスク管理

基 応 専 般

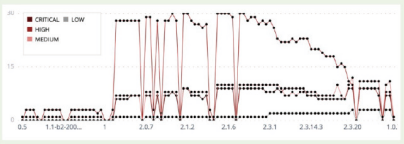
―オープンソース・ソフトウェア（OSS）および商用ソフトウェアのリスク管理のための SBOM ―

松岡正人 | 日本シノプシス合同会社

ソフトウェアのリスク管理のための SBOM

米国 NTIA の Allan Friedman 博士は，2019 年の Black Hat で「Transparency in Software Supply Chain : Making SBOM a reality」と題する講演を行い「工業製品は保守のために BOM が用意されているのに，なぜソフトウェアにはこれがないのか?」と述べた．

■表 -1　SBOM 情報の例
※各項目のうち，推移や一覧，ほかのソースへのリンクについては参考例として弊社製品の画面およびリンク先の情報を貼り込んでいる．

項目	情報
URL	https://github.com/apache/tomcat
ライセンス	Apache License 2.0 http://www.apache.org/licenses/LICENSE-2.0
脆弱性情報（CVE）	CVE-2019-0221，CVE-2019-0232，CVE-2019-10072，CVE-2019-17569，CVE-2020-1935，CVE-2020-1938，CVE-2020-9484，CVE-2020-11996，CVE-2020-13935，CVE-2020-13934 ※各項目をクリックすることでMITREなどのCVEのソースを参照可能であることが望ましい
バージョン	3.3.x (archived), 4.1.x (archived), 5.5.x (archived), 6.0.x (archived), 7.0.x, 8.0.x (superseded), 9.0.x, 10.0.x ※各バージョンをクリックすることで，詳細なバージョンの履歴と，関連する実行環境などの技術情報などが参照できることが望ましい
最終更新日	2020/10/16
コミット数の推移	
発見されたバグの推移	
開発者数の推移	
依存関係	
従属関係	使用した場所 プロジェクト / バージョン / リリース済み / フェーズ WebGoat/WebGoat / 6.0 / なし / 開発中 WebGoat/WebGoat / v8.0.0.M21 / なし / 開発中

ソフトウェア部品表（SBOM）は，ソフトウェアの構築に使用されるさまざまなコンポーネントの詳細とサプライチェーンの関係を記載したモノである．再帰的に利用されたソフトウェアコンポーネントを構成する成分のリストで構成され，ソフトウェアコンポーネント，それらのコンポーネントに関する情報，およびそれらの間の関係を識別して一覧表示したものである（表 -1）．

本稿では SBOM を用いたリスク管理方法について概論し，個別の課題について併記する．

どのようなリスクを管理するか

ソフトウェア開発では，自ら書いたものと第三者が書いたものとの 2 種類のソフトウェアを組み合わせることが一般的である．これらのプログラム・コードには，バグやセキュリティ上の問題となる「脆弱性」と呼ばれるものも含まれる可能性がある．CVE（Common Vulnerabilities and Exposures）は，米国政府の支援を受けた非営利団体 MITRE 社が個別製品中の脆弱性を対象として採番している識別子である．日本では JPCERT/CC☆1 や IPA☆2 が研究者などが発見した脆弱性をソフトウェアや製品の製造元とともに脆弱性の特定と修正を行い，CVE として公開している．CVE を付与することにより，ある組織が発行する脆弱性対策情報と，ほかの組織が発行する脆弱性対策情報が同じ脆弱性に関する情報であるかを判断するのに活用できる．また情報同士の相互参照や関連付けに利用できる．

オープンソース・ソフトウェア（以下 OSS）の数は増

☆1 https://www.jpcert.or.jp/
☆2 https://www.ipa.go.jp/

え続けており，一方でコミュニティが活動しなくなることで開発が放棄されるものも増えつつある．OSS の脆弱性に伴うリスクをどのように管理するかが大きな課題となってきている．2020 年オープンソース・セキュリティ & リスク分析（OSSRA）レポート[☆3]では，調査した 1,253 本の商用ソフトウェアの 99% に OSS が含まれ，コード全体の 70% を占めていたと報告されており，OSS の脆弱性は商用ソフトウェアの脆弱性に大きな影響を与えている点で「リスク因子」である．

別のリスク因子は「OSS のライセンス」である．ライセンス違反の事例についてはここでは詳細に述べないが，いくつかの事例が IPA（（独）情報処理推進機構）のサイトなどで参照できる．

SBOM 管理の実際

SBOM 管理のための標準化や関連する情報をコンポーネントに埋め込むための手法が開発されても，管理自体を手作業で行うのでは効率が悪すぎる．現実的には SCA（Software Composition Analysis：ソフトウェア・コンポジション解析）ツールを利用する必要があり，SCA ツールは商用で提供されている．

SCA ツールは，公開されている脆弱性情報やライセンス情報だけでなく,さまざまな調査データも含めてデータベースに格納してある．管理対象となるソースコードやバイナリーを専用のツールを使って解析し，内包されているソフトウェアコンポーネントをデータベースと照合することで峻別することができる．特定できたコンポーネントには，その由来やサプライチェーン上のリスクとなり得る情報を紐づけることができる．

しかし, 多くの場合問題となるのは「Snippet(スニペット)」と呼ばれる，コードの一部だけをコピーする場合である．スニペットによって起こり得るリスクとは「ライセンス違反」が主であり，訴訟の対象となることで製品の販売差し止めや損害賠償に至る場合もある．しかし，それらのコードの断片がどのコンポーネント由来のものであるのかを突き止めるために 100% 合致する解析結果を得ることは非常に困難であることから，いわゆる「誤検知」が発生する可能性がある．これを回避するにはソースコード管理を厳密に行い，由来の不明なスニペットを排除する必要がある．

バイナリーで流通するコンポーネントの扱いは技術的な難易度が非常に高いため，現時点で利用可能なツールはほとんど存在していない．Black Duck Binary-Analysis[☆4], Clarity[☆5], CodeSentry[☆6] などがあるが，URGENT/11[☆7] などの組込み機器の開発で問題となる脆弱性を検知することができるツールは限定されるため，用途によっては期待する分析結果を得られないこともある．

仮想化によって高可用性や高いスケーラビリティを実現しているクラウド環境では，アプリケーションと実行環境をコンテナ化することで高度で柔軟な運用を可能にしている．このコンテナイメージもバイナリーであり，開発と運用とを一体で管理する DevOps を実現する重要な技術として利用が拡大しているが，コンテナそのものもイメージごと解析する必要が出てきており，いくつかの SCA ツールで解析できる．

このように，開発と運用，言語と実行環境，システムの構成と利用されているコンポーネント，これらの組合せにより，開発プロセスの各フェーズで SBOM の情報を解析し必要に応じて脆弱性の修正や，緩和策の導入などを行う体制が必要なのは言うまでもない．

（2020 年 10 月 22 日受付）

☆3 https://www.synopsys.com/blogs/software-security/ja-jp/2020-ossra-findings-infographic/

☆4 https://www.synopsys.com/ja-jp/software-integrity/security-testing/software-composition-analysis.html

☆5 https://www.insignary.com/

☆6 https://www.grammatech.com/codesentry-sca

☆7 https://www.fda.gov/medical-devices/safety-communications/urgent11-cybersecurity-vulnerabilities-widely-used-third-party-software-component-may-introduce

■松岡正人　masato.maysuoka@synopsys.com

長岡工業高等学校電気科卒業．組込み含むソフトウェア開発を 10 年余り経験後，日本ラショナルソフトウェア，日本マイクロソフト，カスペルスキーなどを経て 2019 年より日本シノプシス JNSA IoT Security WG リーダー．ASTER 理事．