BLACKDUCK®

CISA が定義した 6 種類の SBOM から最適なものを見極める

意外と奥が深いソフトウェア部品表（SBOM）

ソフトウェア部品表（SBOM）に含めるべき内容の最小要件についてはかなり前から業界内で合意が形成されていますが、米国サイバーセキュリティ・社会基盤安全保障庁（CISA）はこのたび、SBOM を 6 つの種類に分類しました。6 種類の SBOM のほとんどは、ソフトウェア開発ライフサイクル（SDLC）の特定のフェーズで作成され、その時点で存在するソフトウェアの状態を反映します。

基本的に、SBOM とはある特定のソフトウェアを構成する要素をリストにしたものです。CISA がこれらの SBOM の種類を定義した理由の 1 つは、このようにさまざまな時点でのソフトウェア構成の状態について、組織により多くの情報を提供することにあります。しかしこれに伴い、どのような状況で、どのような目的のためにどの種類の SBOM を作成すべきかを見極めることが必要になってきました。

このガイドでは、完全に機能するアプリケーションのことを「ソフトウェア」と呼び、CISA が定義したこれら 6 つの SBOM を SDLC に即して説明します。

6 種類の SBOM の定義

まず、SBOM の種類およびそれぞれの長所と短所について簡単に説明します。ここでは、全体を組織する原理として SDLC を使用しますが、SBOM の種類によっては SDLC の複数のフェーズで有用なものもあれば、特定のフェーズでしか意味を持たないものもあることに注意してください。また、各種 SBOM で提示されるデータは、ソフトウェアのライフサイクル・フェーズや業界によっても異なることがあります。

デザイン（Design）SBOM

- デザイン SBOM は、新しいソフトウェア成果物のために意図され、計画されたソフトウェア・プロジェクトまたは製品に含まれるコンポーネントを記述したもので、一部のコンポーネントはまだ存在していない可能性もあります。通常、この情報は設計仕様書、RFP、初期コンセプトなどから収集し、人手で SBOM を作成します。
- 通常、この SBOM には最終的なアプリケーションに存在する多くの依存関係が反映されません。しかし、潜在的な問題に早い段階で対処することにより、今後の作業方針を計画するのに役立ちます。

ソース（Source）SBOM

- ソース SBOM は、製品の成果物のビルドに使用する開発環境、ソース・ファイル、および依存関係から直接作成されます。通常、この SBOM はソフトウェア・コンポジション解析（SCA）ツールを使用して生成したものを人手で明確化します。
- この SBOM は、完全にビルドされたアプリケーションや動作中のアプリケーションを可視化して作成したものではないため、この後のライフサイクルで発生する依存関係を含んでおらず、無関係な依存関係を含んでいることさえあります。

ビルド (Build) SBOM

- ビルド SBOM は、ビルド・プロセスの一部としてソース・ファイル、依存関係、ビルド済みコンポーネント、および一時的なビルド・プロセス・データなどのデータから生成されます。この SBOM は、通常の構成アクションに従って完全に自動で生成されます。ビルド SBOM は、サードパーティ SBOM、ソース・ファイル、コード、ビルド・コンポーネントなどすべての利用可能な要素を含み、中間のビルド SBOM とソース SBOM を統合します。
- この SBOM はソース・コードの範囲を超えた依存関係を含んでいるため、デプロイされるアイテムの構成要素を正確に表現します。この段階で作成される SBOM は他の SBOM を含んでいるため、中間のビルド SBOM とソース SBOM を統合して最終リリースの成果物の SBOM を作成できます。このフェーズで SBOM を承認してセキュアなデリバリを行うことも可能です。
- ただし、この SBOM を生成するには、ビルド・ツールとの統合に大がかりな設定作業が必要です。このため、チームによってはビルド・プロセスの調整が必要になることもあります。

解析済み (Analyzed) SBOM

- 解析済み SBOM は、ビルド後の成果物 (実行ファイル、パッケージ、コンテナ、仮想マシン・イメージなど) を解析して生成されます。通常、このような解析にはさまざまなヒューリスティックが必要です。成果物の解析にはサードパーティ・ツールが使用されるため、文脈によってはこの SBOM を「サードパーティ」SBOM と呼ぶこともあります。
- この SBOM を生成するには、自動または手動のバイナリ解析ツールが必要です。ソース・コードやビルド・システムへのアクセスは必要ありません。解析済み SBOM は、内製ソフトウェアへの可視性の確保、またはベンダーやソフトウェア製作者から提供された SBOM の検証を目的として作成されます。また、さまざまなフェーズで実行される他の SBOM 生成ツールでは特定できない依存関係も、この SBOM なら見つけることができます。解析済み SBOM はヒューリスティックとコンテキストに依存するため、バージョン番号が不正確であったり、欠落したりする傾向があります。

デプロイ済み (Deployed) SBOM

- デプロイ済み SBOM は、デプロイされたシステムに存在するソフトウェアの構成要素をリストにしたものです。この SBOM は他の SBOM を組み立てて作成されることがあり、構成オプションの分析と、デプロイ環境 (シミュレーションによるものを含む) での実行時の挙動の検査を併せて考慮したものとなります。
- デプロイ済み SBOM は、システムに存在するインストール済みのソフトウェアを手動で検査して作成します。このように人手による作業が発生するため、チームは SBOM によって提供される情報と成果物の構成情報を考慮した上で、アプリケーションの動作を実行する必要があります。これは、シミュレーション環境でソフトウェア・プロバイダーが実行するか、実際の環境またはシミュレーション環境で運用者が実行します。
- デプロイ済み SBOM にはソフトウェアが実際に動作する環境 を含めることができますが、この情報を正確かつ完全に取得するのは困難なことがあり、アクセスできないコードに多くの依存関係が存在する場合があります。

実行時 (Runtime) SBOM

- 実行時 SBOM は、ソフトウェアを実行しているシステムにインストルメンテーションを施し、システムに存在するコンポーネント、外部呼び出し、動的にロードされるコンポーネントなどを捕捉して生成されます。通常、この SBOM は動的解析ツールを使用して実行中のアプリケーションに対して「ブラック・ボックス」テストを実行して生成されるため、文脈によっては「インストルメンテーション (Instrumented) SBOM」や「動的 (Dynamic) SBOM」と呼ぶこともあります。
- この SBOM はノイズが少なく、どの依存関係を優先して評価すべきかを明確に理解できます。実行中のアプリケーションに対してこのような解析を実行しようとすると、非常に大きなオーバーヘッドが発生し、アプリケーションの機能を完全に露出するには長い時間と多くのテスト・ケースが必要になることがあります。この SBOM の信頼性と正確性を確保するには、アプリケーションの隅々 (まで) を深いレベルまで網羅的に探索する必要があり、これはアプリケーションのアーキテクチャについての知識がないと困難なことがあります。

どの種類の SBOM が最適かを見極める

一般論として、精度と効率のバランスに優れているのはビルド SBOM または解析済み SBOM です。SBOM はソフトウェアの構成要素を明らかにし、アプリケーションの依存関係に存在するリスクを特定することを目的としているため、製作者と利用者の双方にとって正確性が最も重要になります。

多くの場合、ソフトウェア製作者はビルド SBOM を選択します。ビルド SBOM の生成は SDLC に直接統合して自動化できるため、成果物のすべてのバージョンのライフサイクル全体にわたって正確な SBOM を作成できます。

解析済み SBOM も製作者による生成が可能であり、利用者へ出荷するソフトウェアをより洗練された形で可視化できます。解析済み SBOM はソースやビルドの詳細にアクセスできなくても生成できるため、利用者側で生成してアプリケーションの構成を確実に可視化することもできます。このため、製作者と利用者の双方が解析済み SBOM の結果に対して共同で作業し、必要ならそれぞれの SBOM の差異について協議することができます。

また、ビルド SBOM と解析済み SBOM はほとんどの業界要件への適合が容易です。これらの SBOM を組み合わせると、オープンソースの依存関係、プロプライエタリ・コード、ベース・イメージ、ファームウェア、オペレーティング・システム、およびアプリケーションで必要なサードパーティ・ライブラリを含めることができます。これらの SBOM はツールを使用して自動で生成する必要があるため、いつどのように生成するかをカスタマイズできるほか、どのフィールドを含めるか、どのフォーマットを生成するか、そしていつ SBOM を生成するかを指定できます。

SBOM に含めるべき項目の最小要件は、米国電気通信情報庁（NTIA）が定義したものが公共部門以外でも事実上の標準となっていますが、ビルド SBOM と解析済み SBOM は NTIA の定義した最小要件への適合が可能です。これは、ソフトウェア製作者の立場からは、顧客からの要求を満足できることを意味します。また、ソフトウェア利用者の立場からは、ベンダーに対して明確な要件を定義し、自らのソフトウェア・サプライチェーンの統制を確立できることを意味します。

SBOM 管理の始め方

SBOM の種類について理解できたら、次に、これらすべてをどのように管理するかを考える必要があります。

必要なツール

高度な SCA ツールが役立ちます。SBOM の生成と利用に関して組織には多くの要件があり、それらにどのように優先順位を付けるかが課題となります。SBOM はソフトウェア・サプライチェーンを可視化してアプリケーションのリスクを特定および管理できるようにするものであるため、可視性を確保してリスクに対応付けるには、SCA ツールのオンボーディングが役立ちます。

最高水準の SCA ツールなら、アプリケーション、ソース・コード、ファイル、ビルド成果物、コンテナ・イメージ、ライブラリ、ファームウェアなどの依存関係を見つけることができます。SCA は主にオープンソースの依存関係を検出するために使用しますが、ツールによってはプロプライエタリや商用コードの依存関係も開発および認識できることがあります。このような解析の結果として、完全な SBOM が得られます。

SCA ツールからは、依存関係をリスクに紐付けるためのデータ・ソースも提供されるため、これらを評価して以下の三大リスクを検討することもできます。

- セキュリティ
  - 脆弱性の深刻度のしきい値を超えているか。
  - OWASP/SANS コンプライアンスに適合しているか。
  - 脆弱性の悪用可能性、修正可能性、到達可能性はどうか。
- コンプライアンス
  - 各ライセンスでは何が義務付けられているか。
  - 最終アプリケーションのライセンスと競合するライセンスが存在するか。
  - 承認済みリストまたは禁止リストに載っているライセンスはあるか。
- コンポーネントの健全性
  - そのコンポーネントにはアクティブなコントリビューターが存在するか。
  - そのコンポーネントのセキュリティに関する評判はどうか。
  - そのコンポーネントの最新バージョンを使用しているか。

プロセスを確立する

多くのソフトウェア利用者は、自社の顧客に対してソフトウェアを製作しており、SBOM を提供する義務があります。この作業は人手でも可能ですが、SBOM の生成をビルド・システムに組み込み、変更や新規ビルドのたびに再生成される SBOM を API を使用して自動で取得することがベスト・プラクティスとして推奨されます。これにより、機械可読性のある SBOM が生成され、SBOM を SCA ツールにインポートできるようになります。こうすれば、依存関係を迅速かつ継続的に評価して、セキュリティ、ライセンス、品質上のリスクを特定できます。

もう 1 つ重要なのは、SBOM を単なる文書やファイルとして捉えるのではなく、SBOM の作成をプロセスと考えるということです。個々の SBOM はアプリケーションの構成要素をリストにしたものに過ぎませんが、SBOM の作成を 1 つのアプローチとして扱うことにより、サプライチェーンの動的な可視化と上流のリスク管理が可能になります。

SBOM をプロセスとして扱うには、次のことが推奨されます。

- 何を含めるか、どの程度の頻度で SBOM を生成するか、どの技術を使用して SBOM を管理するかに焦点を当てる。
- SBOM のインポート方法を理解する。SBOM は、アプリケーション・セキュリティ態勢管理 (ASPM) ツールや SCA ツールにインポートできるだけでなく、データベースなど、複数の SBOM を集約でき、最終的にポートフォリオ内のすべてのアプリケーションに関して依存関係をリスクへと関連付けることができるなら、どのようなツールにもインポートできます。
- SBOM の利用を実際の行動につなげる。多くの組織がソフトウェア・ベンダーに対して SBOM の提供を要求していますが、これらを評価または使用してリスクの軽減に役立てているのは少数にとどまっています。
- SBOM に対して独自の加工・流通過程の管理 (Chain of Custody) を割り当てる。加工・流通過程の管理によって構築される認証メカニズムは、製品のライフサイクルおよび過程における検証可能な記録としての役割を果たします。
- 重要なステークホルダーと SBOM を安全に共有し、加工・流通過程の管理全体にわたって SBOM の完全性を保護する。
- SBOM に対する検索とクエリーを可能にする。これにより、今後深刻な脆弱性が報じられたときに、その脆弱性が外部に露出しているかどうかを把握できます。

ブラック・ダックのソリューション

SBOM の生成と管理に関して、唯一の万能なアプローチやソリューションは存在しません。利用できるリソースやリスク選好はチームによって千差万別であり、必要な SBOM や生成可能な SBOM の種類もチームにより異なります。これでは、どこから手を付ければ良いのか分からず混乱するのも当然です。このガイドで説明した SBOM 管理アプローチの基本的な構築ブロックに意識を集中することで、チームは正しい方向へと動き始め、その勢いを利用してチームに合った完全な戦略を構築することができます。

ブラック・ダックは SBOM 管理の基本条件を満たした一連のツールとサービスを提供しており、チームはすぐに取り組みを開始することができます。ブラック・ダックの SCA ツールは、アプリケーションの依存関係の特定、ファーストパーティ SBOM の生成、サードパーティ SBOM のインポート、依存関係のリスクの洗い出し、修正ガイダンスなどすべての機能を一貫性のあるユーザー・エクスペリエンスとして提供します。このようにすることで、ソフトウェア製作者と利用者の双方がサプライチェーンを可視化し、リスクの特定と軽減に向けた対策をとることが可能になります。

ただし、ツールはあくまでもソリューションの一部に過ぎないとブラック・ダックは認識しています。SBOM を単なる文書から効率的なプロセスへと転換できるように、ブラック・ダックは高度なノウハウとコンサルティング・サービスを活用し、お客様が置かれた状況を把握しながら SBOM 管理に関する徹底した 戦略の立案をお手伝いします。

詳細はこちら

ブラック・ダックについて

ブラック・ダックは、業界で最も包括的かつ強力で信頼できるアプリケーション・セキュリティ・ソリューション・ポートフォリオを提供します。ブラック・ダックには、世界中の組織がソフトウェアを迅速に保護し、開発環境にセキュリティを効率的に統合し、新しいテクノロジーで安全に革新できるよう支援してきた比類なき実績があります。ソフトウェア・セキュリティのリーダー、専門家、イノベーターとして認められているブラック・ダックは、ソフトウェアの信頼を築くために必要な要素をすべて備えています。詳しくは www.blackduck.com/jp をご覧ください。

ブラック・ダック・ソフトウェア合同会社
www.blackduck.com/jp