SYNOPSYS®

[2023]
オープンソース・セキュリティ & リスク分析レポート

[目次]

[はじめに]

「2023 オープンソース・セキュリティ & リスク分析レポート」と CyRC について

2023 年で 8 回目の発行を迎えた「オープンソース・セキュリティ & リスク分析（OSSRA）レポート」は、毎年、商用ソフトウェアに含まれるオープンソースのリスクについての現状をセキュリティ、コンプライアンス、ライセンス、およびコード品質の面から詳しく分析しています。シノプシスは、セキュリティ、法務、リスク、開発チームがオープンソースのセキュリティおよびライセンスのリスク状況への理解を深めていただけるように、これらの調査結果を公開しています。このレポートのデータは、シノプシス サイバーセキュリティ・リサーチセンター（CyRC）から提供されています。CyRC は、セキュリティ・アドバイザリおよびリサーチの提供と発行を通じ、組織における高品質なソフトウェアの開発と利用を促進することを使命としています。

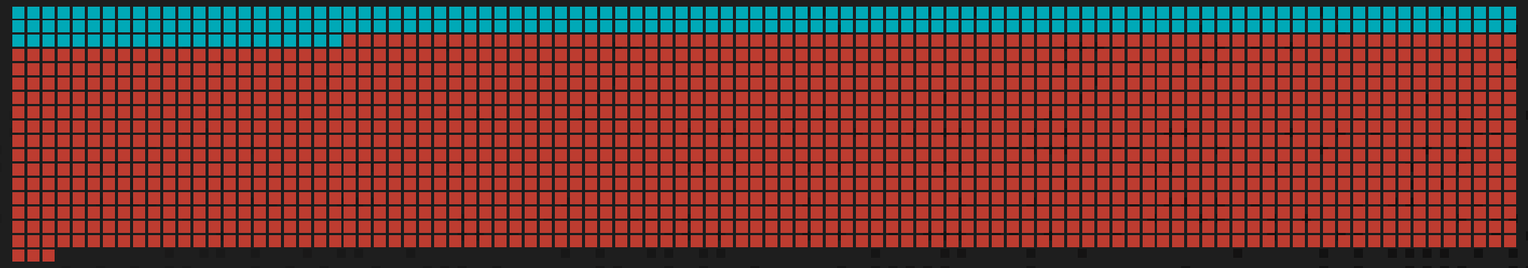
OSSRA の年次レポートは、前年のデータを用いて CyRC が調査し、得られた結果をまとめたものです。したがって 2023 年版レポートは 2022 年のデータに基づいています。2022 年は CyRC は 17 の業種で 1,700 を超える商用コードベースから収集した匿名化データを分析しました。CyRC の監査サービス・チームは、主に顧客の合併・買収（M&A）取引において幅広いソフトウェア・リスクを特定することを目的として、毎年数千ものコードベースを監査しています。2022 年は経済見通しの不透明感が強まり、テクノロジー企業の M&A が低調だったにもかかわらず、監査の件数は堅調を維持しています。

シノプシスの Black Duck® ソフトウェア・コンポジション解析（SCA）製品チームおよび CyRC の監査サービス・チームは、これまで約 20 年にわたって世界中のセキュリティ、開発、法務チームに対し、セキュリティおよびライセンス・コンプライアンス・プログラムの強化を支援してきました。Black Duck SCA により、組織はオープンソース・コードを特定および追跡し、既存の開発環境全体でオープンソース・ポリシーを自動で適用できるようになります。通常、M&A 取引の一環として実施される Black Duck 監査は、ソフトウェアのリスクを全面的に分析します。また、Black Duck 監査では、組織で使用しているアプリケーションに含まれるオープンソース、サードパーティ・コード、web サービス、および API を網羅した、最新かつ非常に正確なソフトウェア部品表（SBOM）も提供しています。この監査サービス・チームは、ライセンス・コンプライアンスおよびセキュリティの潜在的リスクを Black Duck KnowledgeBase ™のデータに基づいて特定しています。Black Duck KnowledgeBase には、28,000 を超えるフォージおよびリポジトリから CyRC が収集、整理した 610 万を超えるオープンソース・コンポーネントについてのデータが登録されています。

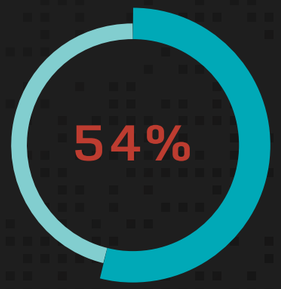
業種を問わず、また組織のセキュリティとリスクに関してどのような役割を担っているかにかかわらず、ビジネスの原動力としてのオープンソースの存在感は高まる一方であり、オープンソースの効果的な管理ができていないと大きな落とし穴が待っていることは、OSSRA が常に強調している点です。このレポートで毎年指摘しているように、オープンソースは、私たちが現在使用しているすべてのアプリケーションの基盤となっています。したがって、オープンソースを効果的に特定、追跡して管理することは、ソフトウェア・セキュリティ・プログラムの成功に欠かせません。このレポートでは、オープンソースの開発者と利用者の両方がオープンソースのエコシステムへの理解を深め、オープンソースの責任ある管理をできるように、いくつかの重要な提言と知見を示していきます。

業種を問わず、ビジネスの原動力としてのオープンソースの存在感は高まる一方であり、オープンソースの効果的な管理ができていないと大きな落とし穴が待っていることは、OSSRA が常に強調している点です。

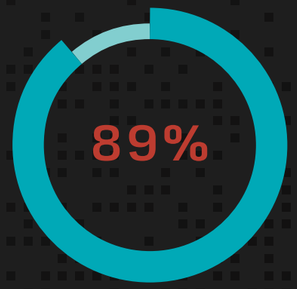
[概要]



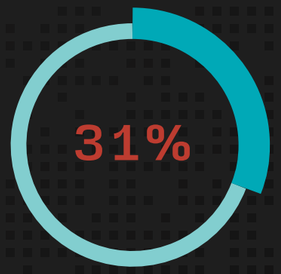
2022 年にスキャンした **1,703** のコードベースのうち、**87%** がセキュリティ / 運用リスク診断も実施



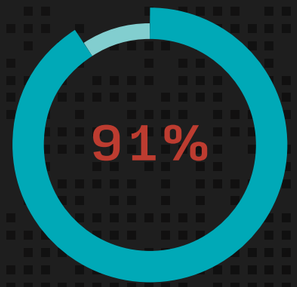
ライセンスの競合が見つかった
コードベースの割合



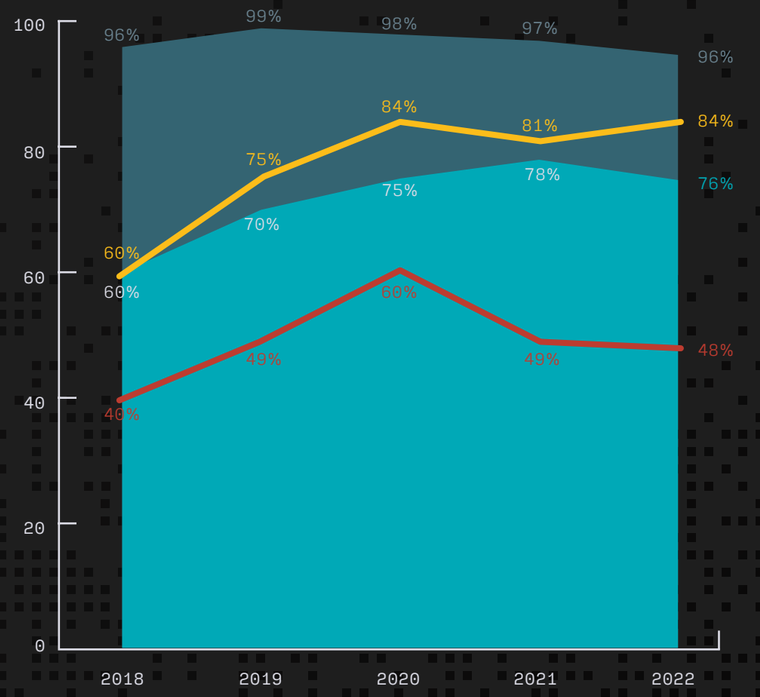
4 年以上前の旧バージョンの
オープンソースを使用している
コードベースの割合



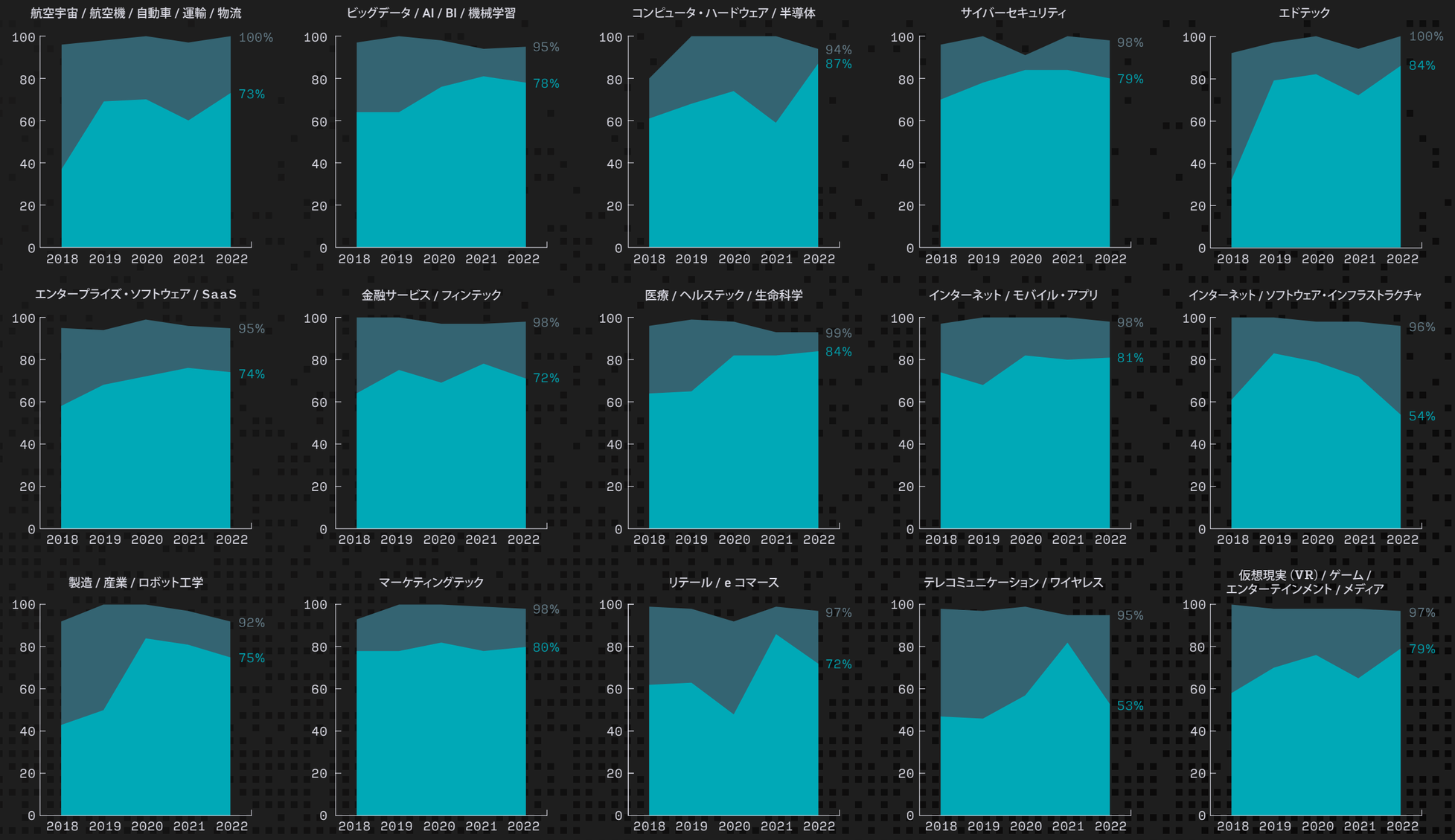
ライセンスのない、または
カスタム・ライセンスを使用している
コードベースの割合



過去 2 年間に新たな開発活動
実績のなかったコンポーネントを
含むコードベースの割合



業種別に見た
オープンソースの使用状況



用語

コードベース

アプリケーションまたはサービスを構成するコードおよび関連ライブラリ。

バイナリ解析

静的解析の一種。ソースコードを入手できないアプリケーションの内容を特定します。

Black Duck Security Advisory（BDSA）

CyRC セキュリティ・リサーチ・チームが調査および分析した、オープンソース脆弱性に関するタイムリーで一貫性のある詳細な情報。BDSA は、シノプシスの顧客に向けてオープンソース脆弱性およびアップグレード / パッチ・ガイダンスの補足情報をいち早く提供します。BDSA はデータ量、完全性、正確性の面で National Vulnerability Database（NVD: 情弱性情報データベース）よりも充実した内容となっており、早期の警告と包括的な知見が得られます。BDSA では、即日通知、具体的な軽減策のガイダンスと回避策の情報、深刻度スコア、参考情報など多くのものが提供されます。

ソフトウェア・コンポーネント

開発者がソフトウェアの部品として追加できる作成済みコード。カレンダーなどのユーティリティの場合もあれば、アプリケーション全体をサポートする包括的なソフトウェア・フレームワークの場合もあります。

依存ファイル

あるソフトウェアが別のソフトウェア・コンポーネントを使用する場合、そのソフトウェアの動作は使用するコンポーネントに依存することになるため、そのコンポーネントを依存ファイルと呼びます。1 つのアプリケーションまたはサービスには多くの依存ファイルが存在することがあり、それらの依存ファイルがさらに別のコンポーネントに依存していることもあります。

オープンソース・ライセンス

オープンソース・コンポーネント（またはコンポーネントのコード・スニペット）をソフトウェアで使用する場合、そのコンポーネントの使用や再頒布の方法など、エンドユーザーが従うべき義務を記述した条文。ほとんどのオープンソース・ライセンスは、次の 2 つのカテゴリのいずれかに分類されます。

寛容型ライセンス

使用に関する制限が少ないものを寛容型ライセンスと呼びます。
一般に、この種のライセンスでは元のコードの作者に帰属する著作権を表示することが主な条件となります。

コピーレフト・ライセンス

一般に、コピーレフト・ライセンスには、改変および拡張したバージョンも元のコードと同じ条件でリリースする必要があり、要請があった場合は変更を含むソースコードを提供する必要があるという互恵型の義務が含まれます。コピーレフト・ライセンスのオープンソースをソフトウェアに組み込んで使用すると、コードベース全体の知的財産権が疑問視されることがあるため、営利企業からは敬遠されます。

ソフトウェア・コンポジション解析（SCA）

アプリケーション・セキュリティ・ツールの一種で、オープンソース・ソフトウェアの管理プロセスを自動化するために使用します。SCA ツールをソフトウェア開発ライフ・サイクルに統合すると、コードベースで使用されているオープンソースの特定、リスク管理および緩和のための推奨事項の提示、ライセンスへのコンプライアンス・チェックなどが可能です。

ソフトウェア部品表（SBOM）

コードベースに含まれるソフトウェア・コンポーネントと依存ファイルを網羅したインベントリ（目録）のこと。多くの場合、ソフトウェア・コンポジション解析（SCA）ツールによって生成されます。米国商務省電気通信情報局（NTIA）は、「SBOM にはソフトウェア・コンポーネントと依存ファイルに関する機械可読な目録、これらコンポーネントに関する情報、およびそれらの階層関係を含めるべき」としています。SBOM は企業間やコミュニティ間での共有を想定しているため、内容とフォーマット（人間可読性と機械可読性の両方を備えたもの）に一貫性が求められます。米国政府のガイドラインでは現在、SPDX（Software Package Data Exchange）と CycloneDX の 2 つが承認済みの標準フォーマットとして指定されています。

大統領令 EO 14028

2021 年 5 月、バイデン米大統領は「Improving the Nation's Cybersecurity」（EO 14028）を発令し、連邦政府のさまざまな機関に対し、連邦政府と取引のある企業に向けたソフトウェア・セキュリティ・ガイドラインの作成を指示しました。この大統領令には具体的な活動のタイムラインが示されていますが、本稿作成時点では、これらは契約上の義務とはされていません。しかし、強制ではないものの、この大統領令を契機に多くの組織が自社のセキュリティ・プラクティスの再点検を開始し、ソフトウェア・セキュリティ・リスクのレベルを精査するようになっています。ソフトウェア部品表（SBOM）はソフトウェアの開発者側と利用者側の間でソフトウェア・サプライチェーンの情報伝達を円滑化するものであり、SBOM の使用は EO 14028 が推進する重要な要素の 1 つとなっています。

Apache Log4j2 の脆弱性（BDSA-2021-3887、CVE-2021-44228、他）

Apache Log4j2（一般的には「Log4j」）は、アプリケーションにログ機能を実装するために Java コミュニティで広く使用されているオープンソース・コンポーネントです。Log4j には、リモート・コード実行（RCE）、サービス拒否（DoS）、LDAP の脆弱性など、複数の脆弱性が存在することが確認されています。

ゼロデイ脆弱性

標的とされるソフトウェアのベンダーや作者など、脆弱性の緩和に関与する人がまだ知らない脆弱性、または知っていてもそれを修正するパッチが存在しない脆弱性のこと。

OpenSSL の脆弱性

2022 年 11 月、人気のあるオープンソースのコマンドライン・ツールである OpenSSL は、深刻度「Critical」の 2 つの脆弱性（CVE-2022-3602、CVE-2022-3786）に関するアドバイザリ警告をリリースしました（これらの深刻度は、その後「High」に引き下げられています）。これらは、証明書の検証処理を通じてバッファー・オーバーフロー / オーバーランが発生する脆弱性です。前者の脆弱性を悪用されると、クラッシュが発生して任意コードを実行される可能性があります。後者の脆弱性を悪用されると、メモリ破損の問題につながる可能性があります。

バッファー・オーバーフロー / オーバーランの脆弱性

バッファーは、アプリケーション実行時に一時的なメモリ・ストレージとして使用されます。バッファー容量を超える大きさのデータをバッファーに書き込むと、バッファー・オーバーフロー / オーバーランが発生し、それによってクラッシュ、メモリの問題、またはその他の予期しない挙動が引き起こされます。攻撃者はこの脆弱性を悪用してファイルの改ざんや機微な情報へのアクセスなどを実行します。

[脆弱性とセキュリティ]

オープンソースの脆弱性とセキュリティ

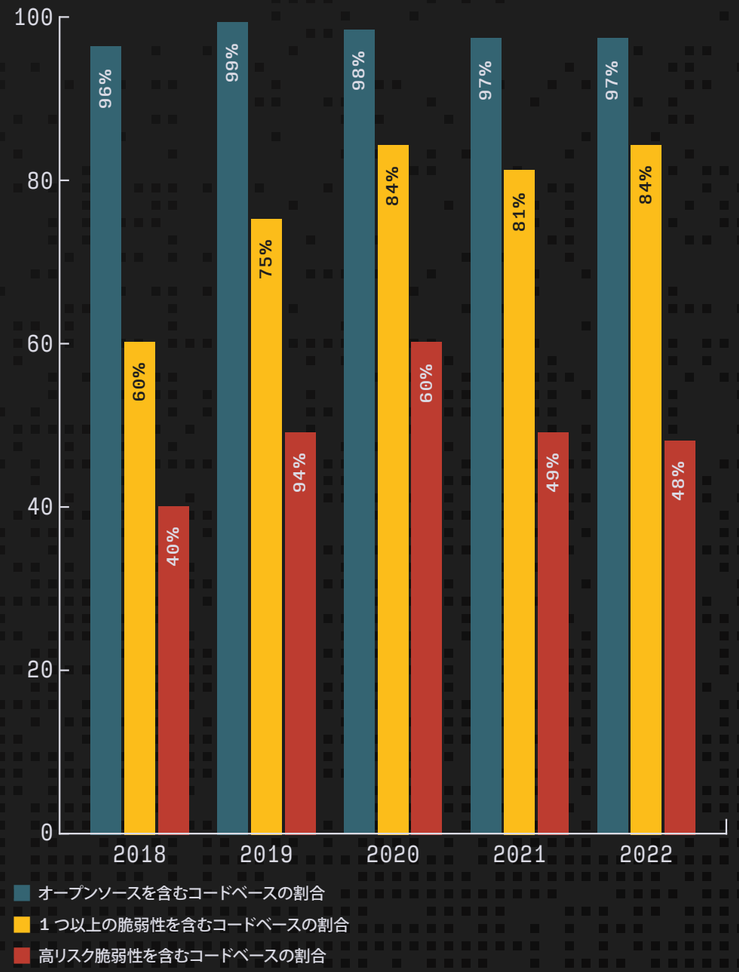
今回 Black Duck 監査サービス・チームが分析した 1,703 のコードベースのうち、96% にオープンソースが含まれていました。また、スキャンした全コードベースの 76% がオープンソースでした。これは、今回調査したコード全体の 76% がオープンソース・コードであるという意味です。

1 つのアプリケーションに含まれるオープンソース・コンポーネントの数は、平均で 595 個でした。これほどの規模になると、脆弱性を監視してセキュリティのメンテナンスを行うのは大変であり、コンポーネントの数が少ない場合に通用していた手法は事実上使えなくなります。そこで必要になってくるのが、SCA などの自動ソリューションです。

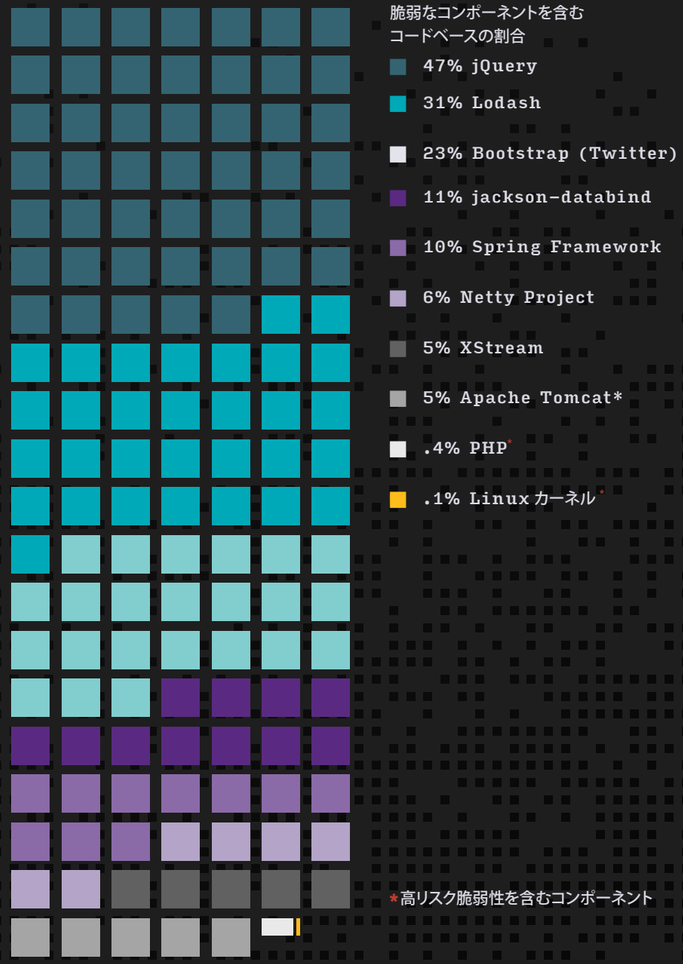
既知のオープンソース脆弱性を 1 つ以上含んでいたコードベースの割合は 84% で、2022 年版 OSSRA レポートから約 4% 増加しています。また、スキャンしたコードベースのうち、高リスク脆弱性を含んでいたものの割合は 48% で、これは昨年から 2% とわずかに減少しています。高リスク脆弱性とは、実際に攻撃を受けたことがあるか、概念実証コード（エクスプロイト）が存在する、またはリモート・コード実行の脆弱性に分類されるものを言います。

Black Duck 監査では、すべてのコードベースに対してオープンソースのライセンス・コンプライアンス検査を実施していますが、セキュリティおよび運用リスクの診断は任意（オプトアウト方式）で実施しています。2023 年版のレポート作成にあたり、Black Duck 監査サービス・チームは 1,703 件の監査を実施しましたが、このうち 87%（1,481 件）が脆弱性 / 運用リスクの診断も受けています。2023 年版 OSSRA レポートの「脆弱性とセキュリティ」および「オープンソースのメンテナンス」のセクションに示したデータは、リスク診断も受けた 1,481 のコードベースを母集団としていますが、「ライセンス」のセクションに示したデータは全 1,703 のコードベースを母集団としています。

脆弱性と高リスク脆弱性



脆弱性のあるコンポーネント



ゴルディアスの結び目：オープンソース・ソフトウェアのリスクとサプライチェーンのセキュリティ

最近、シノプシスと Enterprise Strategy Group（ESG）が共同で作成したレポート「社会的な規範を守る：GitOps とシフト・レフト・セキュリティ」では、最近の市場で懸念されている問題と、組織が現在のセキュリティ課題にどのように対処しているのかを調査しました。このレポートで調査を受けた組織の 73% が「最近発生したソフトウェア・サプライチェーン攻撃をふまえ、オープンソース・ソフトウェア、コンテナ・イメージ、およびサードパーティ・ソフトウェア・コンポーネントのセキュリティ対策を大幅に強化している」と回答しています。憂慮されるのは、過去 12 カ月間に「オープンソース・ソフトウェアに存在する既知の脆弱性を悪用された」とする回答者が 34% にものぼっていることです。

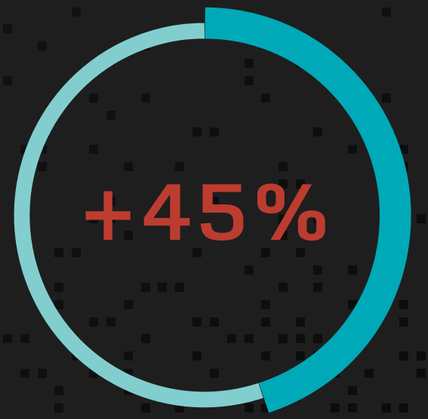
今の時代、少しでもソフトウェア・セキュリティに関わっている人であれば、誰でもソフトウェア・サプライチェーンの問題に関心を向けています。ソフトウェア・サプライチェーンのセキュリティ問題はニュースでも連日のように報じられており、業種を問わずあらゆる組織に影響を与えています。しかしバイデン大統領令（EO 14028）から 2 年近くが経った現在でも、組織は自社のソフトウェア・サプライチェーンの広がりを理解し、自社が使用しているソフトウェアを可視化し、自社が頒布および販売しているソフトウェアに対する透明性確保への要求の高まりに対処するといった、サプライチェーンの基本的な取り組みに苦慮しています。

では、何をすれば良いのでしょうか。ソフトウェア・サプライチェーンのセキュリティ対策の第一歩は、使用しているアプリケーションに含まれるオープンソースおよびサードパーティ・コードを管理することから始まります。

オープンソースおよびサードパーティ・ソフトウェアのセキュリティを効果的に管理および徹底できていなければ、サプライチェーンのセキュリティに関して他にどのような取り組みを行おうとも、徒労に終わるでしょう。このソフトウェアを管理するには、使用している依存ファイルを完全に可視化し、これらコンポーネントによって混入するリスクに関する情報を容易に収集できるようにしておく必要があります。このリスクを特定できたら、リスクの管理、優先順位付け、修正のためのツールと手法を整備することが必要となります。

また、リスク管理活動およびイニシアティブを可視化し、それに対する支持を取り付けるために、特定したリスクを主要なステークホルダーに通知することも重要になってきます。さらに、これらすべての機能と手法を既存の開発パイプラインに統合し、可能な限り自動化を活用することも推奨されます。

複雑に聞こえるでしょうか。それは、実際に複雑なので仕方ありません。サプライチェーンの最終製品とそのユーザーは、その製品の作成に関わったすべてのコンポーネント、人、活動、材料、手続きによる影響を受けます。サプライチェーンとその膨大な入力を完全に可視化してこそ、初めてそのセキュリティ対策が可能になります。そしてこの可視性を確保する取り組みの第一歩となるのが、自分が本当にセキュアなのかどうかを確認するという作業です。「信頼せよ、されど検証せよ」というロシアの古いことわざが今ほど当てはまる時代はありません。オープンソースおよびサードパーティ・ソフトウェアを管理するということは、そのセキュリティを検証済みであるということを意味します。もし未検証であれば、それはサプライチェーンの一番弱い鎖の輪に根拠のない信頼を置いていることになります。



世界中で 2025 年までに
ソフトウェア・サプライチェーン攻撃を
経験すると予想される組織の割合

–Gartner

オープンソース・ソフトウェアのリスクとサプライチェーンのセキュリティは強く結びついています。

では、何をすれば良いのでしょうか。ソフトウェア・サプライチェーンのセキュリティ対策の第一歩は、使用しているアプリケーションに含まれるオープンソースおよびサードパーティ・コードを管理することから始まります。

業種別に見た脆弱性

オープンソースを含むコードベースの割合は毎年増加しています。今年のデータを見ると、最も割合の小さい業種（製造 / 産業 / ロボット工学）でさえコードベースの 92% にオープンソースが含まれており、過去最高の割合となっています。

しかし、オープンソースの使用状況のグラフに脆弱性のグラフを重ね合わせると、気になる点が見えてきます。特に注目すべきは航空宇宙 / 航空機 / 自動車 / 運輸 / 物流業界です。この業種では、検査したコードベースのすべて（100%）にオープンソースが含まれており、コード全体に占めるオープンソース・コードの割合は 73% でした。そして、コードベースの 63% に高リスク（深刻度スコア 7 以上）に分類される脆弱性が含まれていました。

エネルギー / クリーンテックでも同様に、コード全体に占めるオープンソース・コードの割合は 78% で、高リスク脆弱性を含むコードの割合は 69% でした。また、オープンソースを含むコードベースの割合は 95% でした。

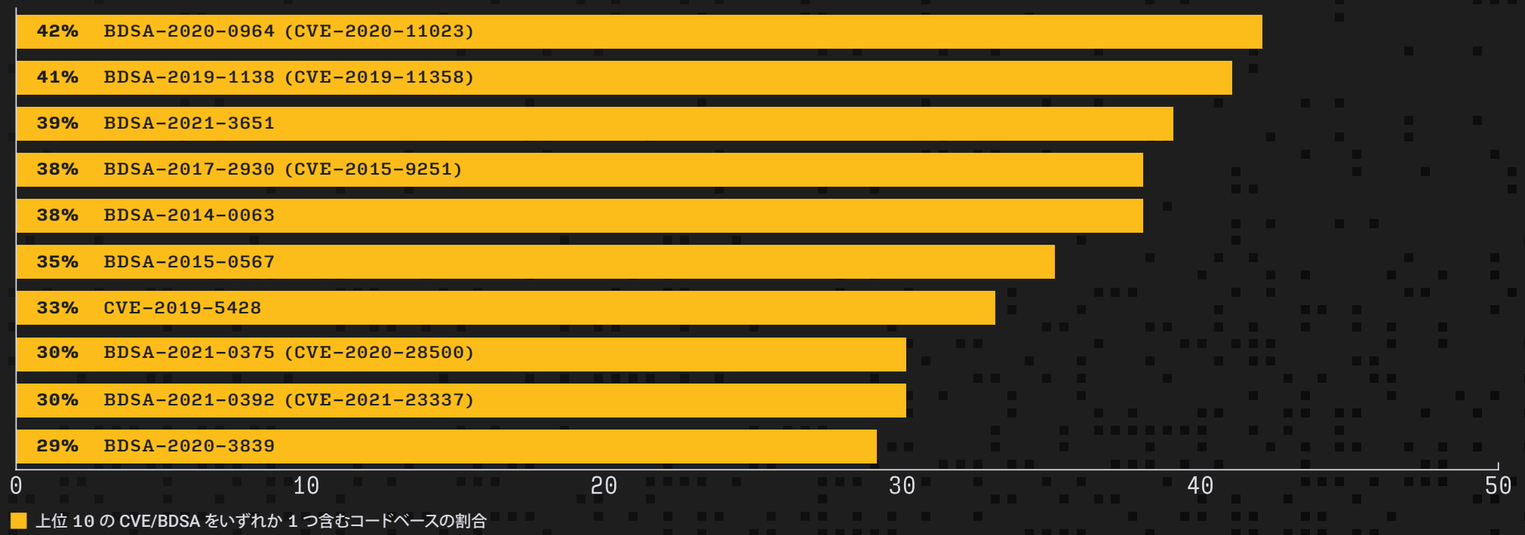
これほど顕著ではないにせよ、同様の調査結果が多くの業種に見られていることにセキュリティ・チームは注意する必要があります。今年監査したコードベースのほとんどすべてにオープンソースが含まれていました。業種を問わずコードベースの大半がオープンソースで構成されており、それらオープンソースには既知の脆弱性が驚くほど多く存在しており、組織はそれらにパッチを適用できておらず、悪用に対して脆弱な状態のままです。オープンソースはそれ自体が高いリスクを内包しているのではなく、オープンソースを正しく管理できていないことがリスクを生むということをしっかりと理解する必要があります。

業種別に見たオープンソースの使用状況と脆弱性

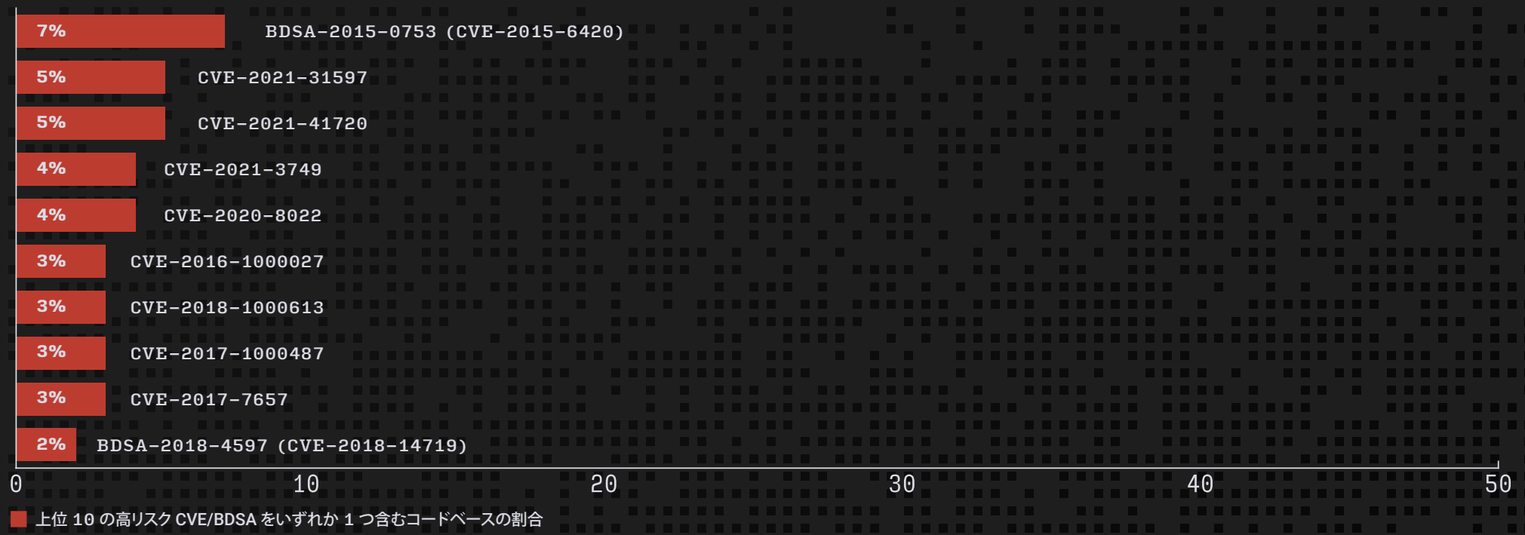
脆弱性の深刻度スコアリング

シノプシス脆弱性深刻度スコアリング・システム（SVSS）は、可能な限り多くの変数を収集してスコアを決定しています。これらのスコアは、シノプシスの Black Duck Security Advisory（BDSA）の一部として使用されます。BDSA は FIRST.org の規定に準じて CVSS スコアリング・システムを利用し、CVSS に従って深刻度スコアを割り当てていますが、SVSS は単に NVD が発行している内容を踏襲するのではなく、CyRC が独自に割り当てています。スコアを割り当てる際、BDSA は悪用可能性など多くの要因を考慮することで、CVSS スコアの精度を最大限に高めています。また、NVD などのソースとは異なり、BDSA は現状評価基準も考慮してスコアを決定しています。このように細部まで調整して最高精度のスコアを提供することにより、顧客がトリアージの優先順位を正確に決定できるように支援することを目的としています。

CVE/BDSA



高リスク CVE/BDSA



5 年間の振り返り

今年の OSSRA レポートでは、過去 5 年間のデータを振り返り、顕著なトレンドを特定してみます。以下がその分析結果です。

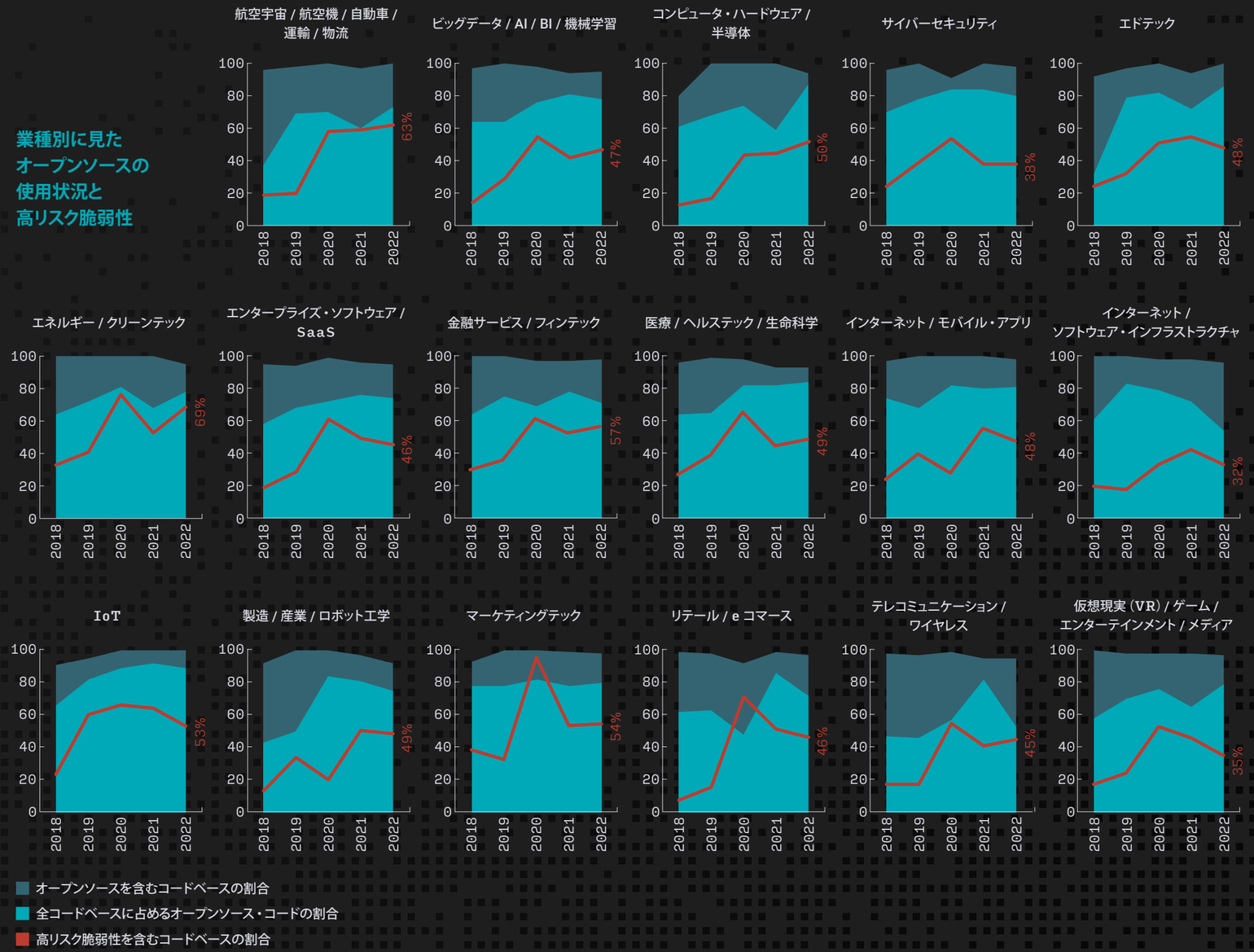
分析結果 1：オープンソースの採用状況は業種ごとに異なる

このレポートでは、オープンソースの成長と採用が続いていることを毎年指摘していますが、この 5 年間を振り返ってみると、採用状況は業種によってさまざまです。先に述べたように、オープンソースを含んでいるコードベースの割合は、すべての業種で 92% 以上でした。しかし、全コードベースに占めるオープンソース・コードの割合を過去 5 年間にわたって観察すると、業種によって大きな違いがあります。

2018 年から 2022 年にかけて、スキャンした全コードベースに占めるオープンソース・コードの割合はエドテックで 163%、航空宇宙 / 航空機 / 自動車 / 運輸 / 物流で 97%、製造 / 産業 / ロボット工学で 74% それぞれ増加しています。エドテックでオープンソースの利用が爆発的に増えているのは、パンデミックの影響と考えられます。教育の場がオンラインに移行し、ソフトウェアが重要基盤としての役割を果たすようになったことを考えると、この増加には納得がいきます。オープンソースは（適切な管理が必要ではあるものの）無償であり、特に予算の制約が強い業界で短期間に大幅な機能強化を果たすには魅力的な存在です。エドテック・システムは内製のものが多く、有志によってメンテナンスが行われています。このため、エドテックの分野では最新テクノロジーの基盤としてオープンソースが活用されているケースが多いと考えられます。

一方、航空宇宙 / 航空機 / 自動車 / 運輸 / 物流業界はオープンソースの採用があまり進んでいません（ただし去年からは 22% と大幅に伸びています）。これは、これらが規制の多い業種であることに関係していると考えられます。これらの業界では、オープンソースの効果的なセキュリティ対策に必要なだけのリソースも能力もなく、過去には組織が意図的にオープンソースを避けてきたこともあったと考えられます。

業種別に見たオープンソースの使用状況と高リスク脆弱性



シノプシスで以前、政府 / 重要インフラ・プログラム担当ディレクターを務めていた Joe Jarzombek は次のように延べ、この数字についてもう 1 つの見方を示しています。「低品質なソフトウェアに起因する技術的負債は、新機能のデリバリの妨げとなります。他の業種や政府と同様に、防衛産業も先取的、創造的、予防的な作業よりむしろ技術的負債の修正に多くの時間と労力を費やしています」。ただし近年、サイバーセキュリティ成熟度モデル認証（CMMC）が始まったことにより、「多くの業界でセキュリティを意識したデータ保護の懸念が緩和され」（Jarzombek）、これらの業界は革新性と俊敏性を高めることが可能になっています。

つまり、オープンソースの採用が遅々として進まないと、オープンソースの利点である革新、スピード、俊敏性が立ち遅れ、旧式の手法がいつまでも残ることになります。しかし技術的負債や重い規制は少しずつ解消されつつあり、オープンソースの利用が増えつつあります。

分析結果 2：高リスク脆弱性が放置されている

次に、高リスク脆弱性を含むコードベースの割合を業種ごとに比べてみました。ここでも、程度の差こそあれ、すべての業種で同じような傾向が見られました。2018 年から現在までで、高リスク脆弱性を含むコードベースの割合の増加率が最も小さかったのは、マーケティングテックの 42% でした。リテール /e コマースでは、高リスク脆弱性を含むコードベースの割合が 2018 年から 557% も増えており、大きな懸念材料となっています。

再び航空宇宙 / 航空機 / 自動車 / 運輸 / 物流業界に目を向けてみると、高リスク脆弱性を含むコードベースの割合は 232% と大きく増えています。これは、単にリスクを軽減する必要がなかったのかもしれません。脆弱性を修正するかどうかの判断は、悪用の可能性や悪用された場合の影響の大きさによって大きく左右されます。これら業界で使用されるソフトウェアやファームウェアの多くは閉じたシステム内で動作するため、悪用の可能性は低く、パッチ適用の緊急性を感じていないとも考えられます。しかも、脆弱性を軽減する方法はコンポーネントをアップデートする以外にいくつもあります。

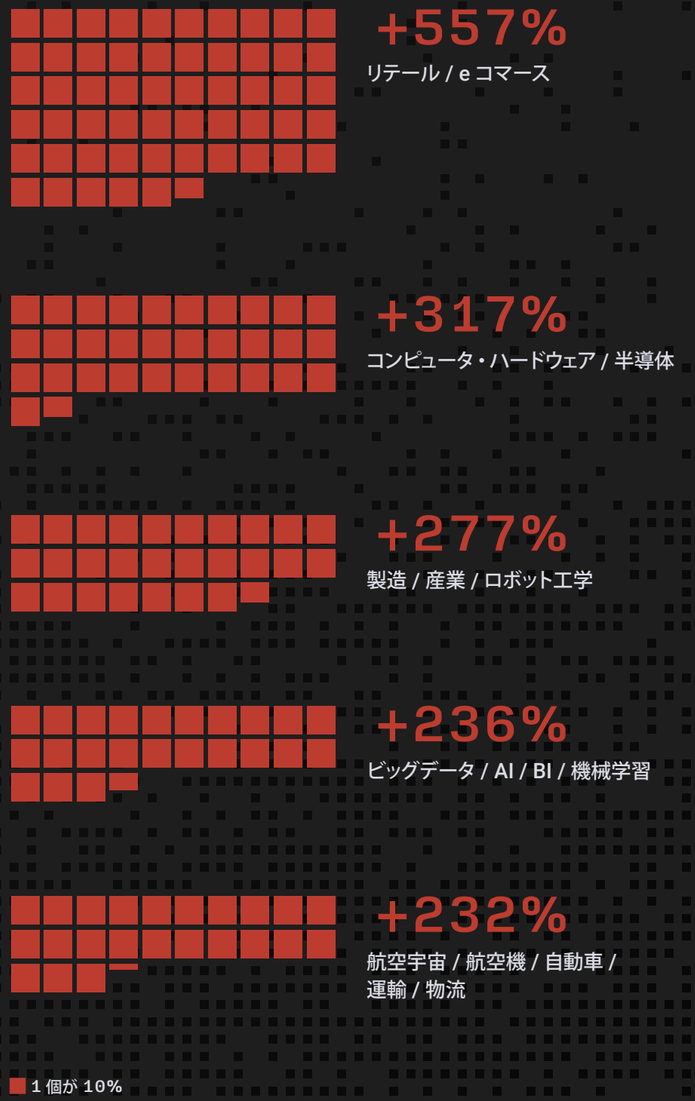
また、この業界におけるソフトウェアのデプロイ、配布、運用のあり方を考えると、高リスク脆弱性を含むコードベースの割合が大きく増えていることのもう 1 つの理由が見えてきます。組み込みソフトウェアおよびファームウェアが動作するハードウェアの多くは、ネットワークに接続されておらず、SaaS アプリケーションのようにアップデートを自動で簡単に配布することができません。パッチを適用するには新しいバージョンをダウンロードしてインストールしなければならない、あるいは USB ドライブを物理的に機器に挿入しなければならないといった環境では、当然パッチの適用頻度は少なくなり、パッチ未適用の脆弱性が増える結果となります。

しかも、この業種のソフトウェアはエンタープライズおよび SaaS アプリケーションを開発している独立系ソフトウェア・ベンダー（ISV）や企業とは異なる標準や手法を使用して開発されています。オープンソース・ソフトウェアの高リスク脆弱性に対処できていないのは、おそらくこの違いによるものと思われます。

IoT（Internet of Things）業界に目を向けてみると、また違った状況が見えてきます。2020 年以降、スキャンしたコードベースの 100% にオープンソースが含まれています。各コードベースに含まれるオープンソースの量も増加しており、コード全体に占めるオープンソース・コードの割合は、2018 年から 35% 増加して 89% となっています。IoT は、オープンソースの利点が活かされる業界の典型と言えます。Ring、Amazon、Nest などスマート・デバイスを開発している IoT 企業は、新しいソフトウェアを一刻も早く開発すべく熾烈な競争を繰り広げています。このように競争の激しい業種では、オープンソースを使用することで高い俊敏性を得ることができます。オープンソースがなければ、猛烈な開発ペースについていくことはできないでしょう。しかしその負の側面として、脆弱性が存在します。

IoT 業界は高リスク脆弱性を含むコードベースの割合が 2018 年から 130% 増加しており、今年は監査したアプリケーションの 53% に高リスク脆弱性が含まれていました（業種全体では比較的多い部類に属します）。IoT デバイスの有用性を考えると、これは深刻な数字です。私たちは、日常生活のさまざまなシーンでネットワークに接続した IoT デバイスを使用しており、そのこと自体の安全性を信頼しきっています。自動スケジュールで照明をオン・オフする IoT デバイスには、私たちがいつ自宅にいて、いつ外出しているのかに関するデータが入っています。カメラには家の中の写真が記録されています。これ以外にも、玄関ドアのスマートロックや、赤ちゃんを見守るベビーモニターなどもあります。

過去 5 年間で高リスク脆弱性を含むコードベースの割合が最も大きく増加した業種



[ライセンス]

オープンソース・ライセンス

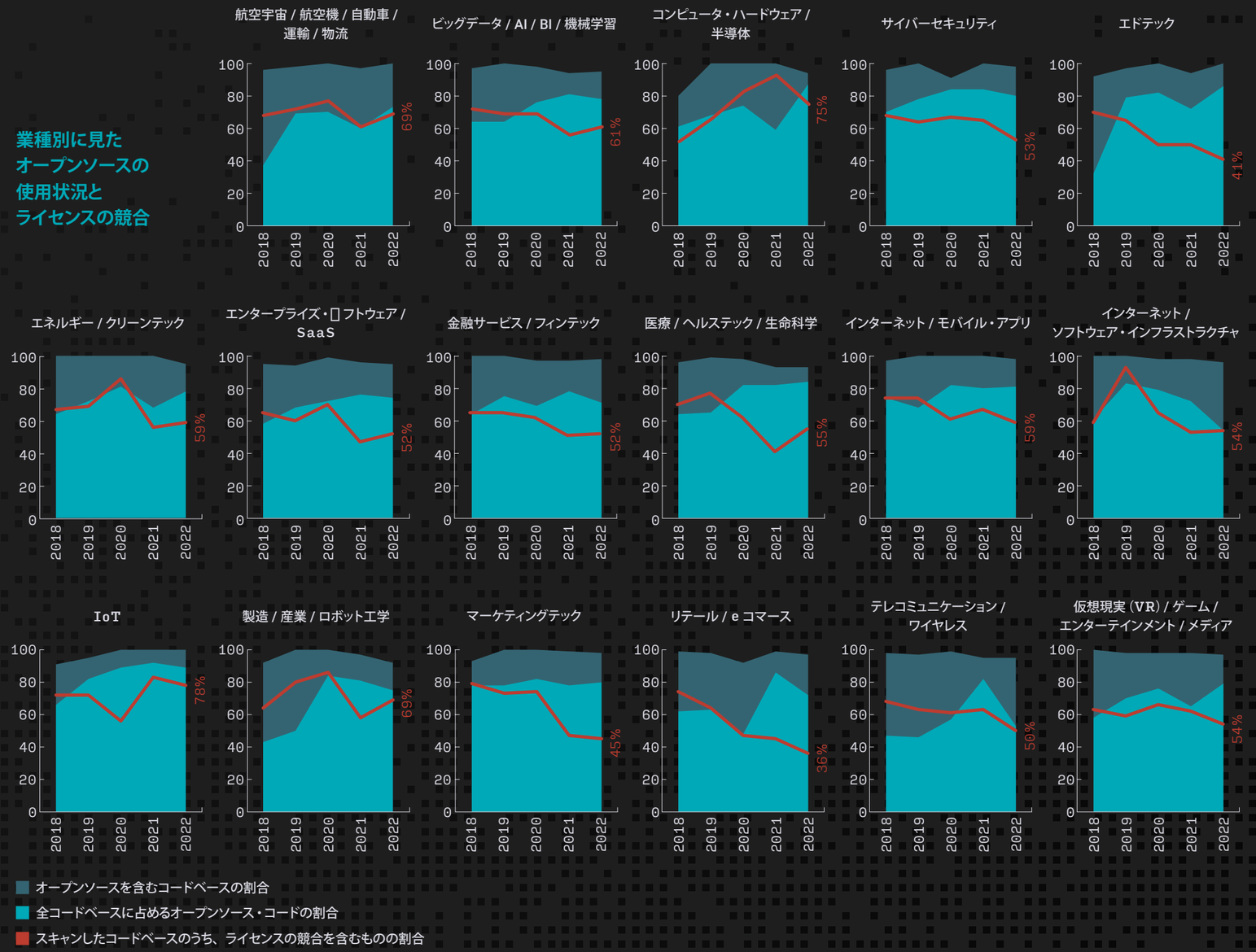
Black Duck 監査サービス・チームが 2022 年に監査したコードベースのうち、ライセンスの競合があるオープンソースを含んだものが 54% ありました。これは昨年から 2% の微増ですが、2020 年 (65%) からは 17% の大幅な減少となっています。

ライセンスの競合の原因として今年最も多かったのが、Creative Commons ShareAlike 3.0 (CC BY-SA 3.0) ライセンスで、監査したコードベースの 22% にこのライセンスとの何らかの競合が見つかりました。CC BY-SA 3.0 でライセンスの競合が多いという事実は、オープンソース・ライセンスに関して見落としがちな問題を示唆しています。商用であれオープンソースであれ、開発者はコード・スニペット、関数、メソッド、および動作コードを組み込んでソフトウェアを作成します。これらのコードは、全体的なソフトウェアがそれに依存して動作するため、「依存ファイル」と呼ばれます。したがって、オープンソース・プロジェクトにはプロジェクト全体とは異なるライセンスでライセンスされたサブコンポーネントが含まれることがよくあります。これらのライセンスがプロジェクト本体のライセンスを上回る条件を規定している場合に競合が発生します。

Black Duck 監査サービス・チームが見つけたライセンスの競合のうち 85% は、技術知識の発見と共有を目的としたオンライン Q&A プラットフォームの Stack Overflow からプルしたコンテンツが関与しています。Stack Overflow の人気の高まり、および CC BY-SA 3.0 ライセンスは頒布型および SaaS 型いずれのデプロイ・モデルでも競合の要因となることが指摘されている事実を考えると、このライセンスが最も多くの競合を引き起こしているのは不思議なことではありません。

業種別では、オープンソース・ライセンスの競合が劇的に減少したケースも見られます。昨年、コンピュータ・ハードウェア / 半導体業界でオープンソース・ライセンスの競合が見つかったコードベースの割合は 93% でしたが、今年は 75% まで減少しています。全体的に見て、ほとんどの業種で同様の改善傾向が見られます。今年、オープンソース・ライセンスの競合が最も多かったのは IoT の 78% でした。このように全体的にオープンソース・ライセンスの競合が改善しているのは、経済の見通しが不透明になったこと、そしてサプライチェーンのリスクに関して全般的な不安があることが原因とも考えられます。

業種別に見たオープンソースの使用状況とライセンスの競合



ライセンス・リスクを理解する

米国など多くの裁判管轄地では、ソフトウェアなどの著作物は無方式主義により独占的に著作権保護されます。著作者からライセンスという形で明示的に権利を付与されない限り、他者がソフトウェアを使用、複製、頒布、改変することは違法となります。最も寛容なオープンソース・ライセンスでさえ、そのソフトウェアの使用と引き換えにユーザーが負う義務を規定しています。

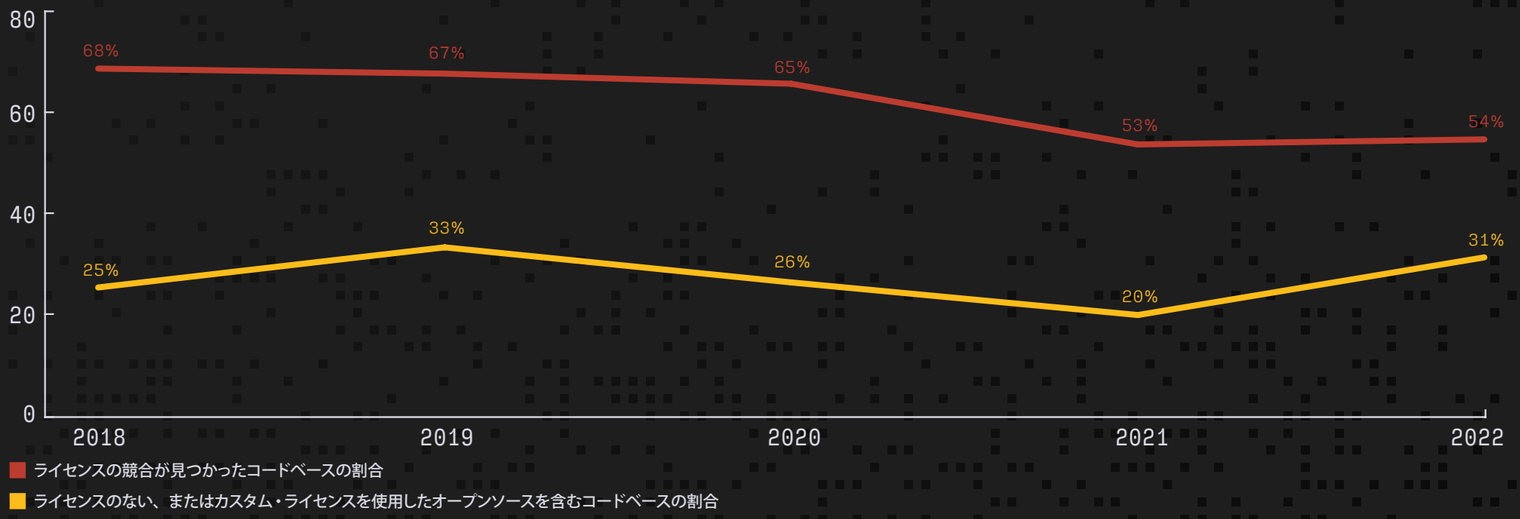
コードベースに含まれるオープンソースのライセンスが、コードベース全体のライセンスと競合するような場合、ライセンス・リスクが生じる可能性があります。オープンソース・プロジェクトに適用されるコピーレフト・ライセンスとして最も一般的なのが GNU General Public License（GPL）です。商用のクローズドソース・ソフトウェアに GPL でライセンスされるコードが含まれていると、競合が発生します。

2022 年に監査したコードベースのうち、ライセンスの存在を確認できない、またはカスタム・ライセンスを使用したものは前年から 55% 増えて 31% になりました。これには 2 つの理由が考えられます。1 つは、スニペットや部分的なコンポーネントを開発者が手動でコードベースに追加した可能性です。このような場合、開発者はスニペットに関連するライセンスを含めることを忘れてしまうことがよくあります。もう 1 つは、ただ単にメンテナーが数ある既知の標準ライセンスとは異なるライセンスや条項を作成しているという可能性です。表面的には問題ないかもしれませんが、ライセンスの影響を完全に理解していないことはリスクとなるため、そのようなことはなるべく避ける必要があります。

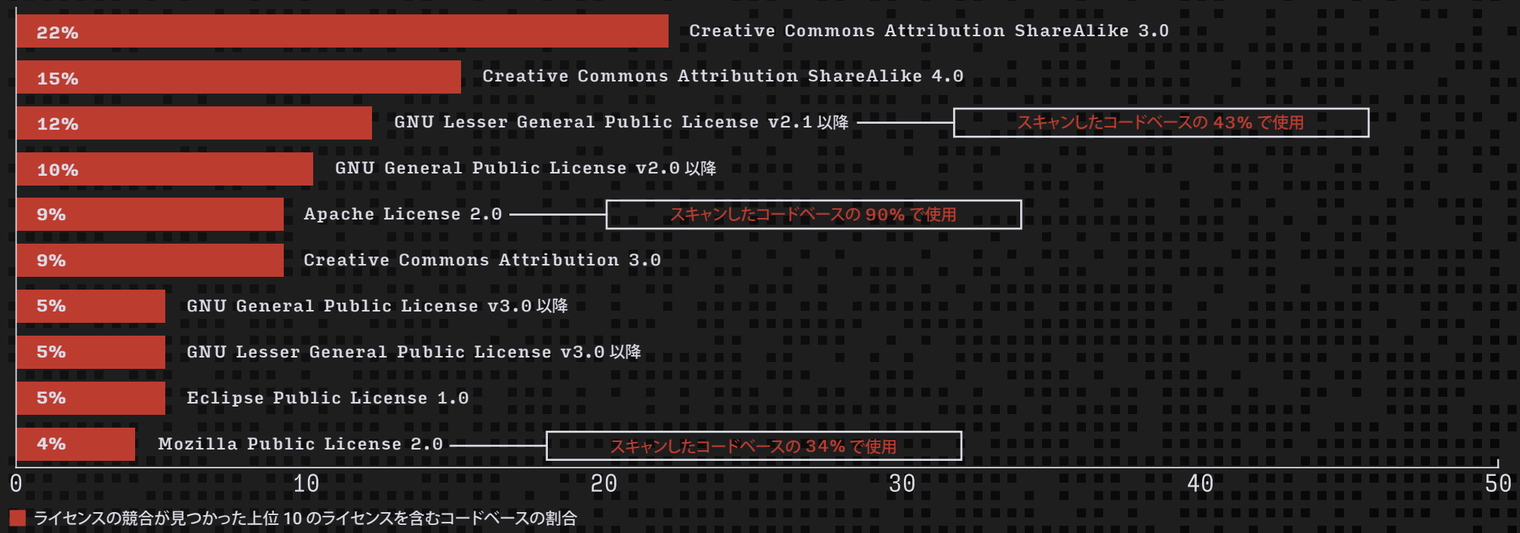
ライセンスのないコードは GitHub リポジトリにも存在します。これは、開発者がライセンス表記をコード内、テキスト・ファイル内、あるいはプロジェクトに関連するメタデータ内に含めずにコードを公開した場合に起こります。あるいは、ブログや web サイトにコードが掲載されているのを開発者が見つけ、ソースコードが公開されているのなら利用しても大丈夫だろうと考えてコピーすることもあります。それはもっともなことのように思えるかもしれませんが、著作権法では認められません。

さらに、標準的なオープンソース・ライセンスではなく、その派生ライセンスや一部をカスタマイズしたライセンスの場合、ライセンシーにとって望ましくない条件が課せられたり、知的財産権（IP）の問題やその他の影響について法的な立場からの評価が必要になることがあります。カスタマイズしたライセンスの代表例と言えるのが、JSON ライセンスです。JSON ライセンスは、ベースとなっている寛容型の MIT ライセンスに「このソフトウェアは善い目的に使用されるべきで、邪悪な目的に使用してはならない」という文言を追加しています。この曖昧な文言はさまざまな意味に解釈できるため、特に M&A に関係する場合は、多くの弁護士が JSON ライセンスのソフトウェアを使用しないように助言しています。

ライセンスの問題



競合の原因となっているライセンス



[オープンソースのメンテナンス]

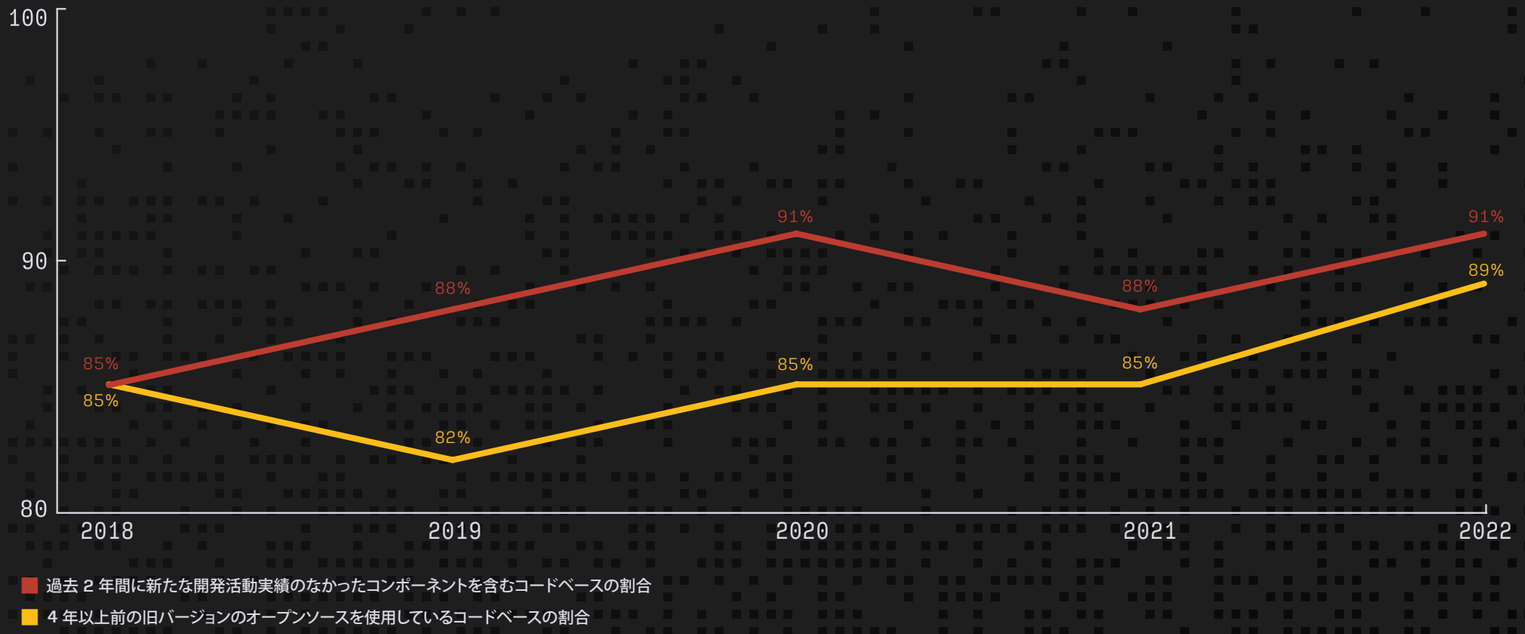
オープンソースの開発者によるメンテナンス

開発者がオープンソース・コンポーネントを選択する際は、活発なコミュニティに支えられているものを選ぶのが理想です。例えば、Linux は数百の団体に属する数千人もの開発者によって日々改善が続けられています。しかし、Black Duck 監査サービス・チームがオプションのリスク診断（p.7 参照）を実施した 1,481 のコードベースのうち、過去 2 年間に新たな開発活動実績のなかったオープンソース（過去 24 カ月にわたり一度も機能のアップデート、コードの改良、セキュリティ問題の修正が行われていないもの）を含むものが 91% ありました。これは、特に小規模なプロジェクトの場合、プロジェクトのメンテナンスが終了しているものと考えられます。

定義上、オープンソース・プロジェクトとは、不特定多数のコントリビューターやメンテナーによる成果物です。このようにしてオープンソースは共同作業として成り立っていますが、コントリビューターに保守活動をしてもらうインセンティブが存在しないことがオープンソースの課題でもあります。Kubernetes などの重要なプロジェクトでは健全なサポートが行われていることがほとんどですが、ごくわずかな人数によってメンテナンスされているプロジェクトも多く存在します。

オープンソース・プロジェクトに関する最近の論評にもあるように、「重要な、そして多くのプロジェクトが依存しているプロジェクトでさえ、正当な評価をほとんど受けずに保守作業に当たっている人が少なくありません」。Microsoft、RedHat、Google など一部の組織はオープンソース・プロジェクトへの参加やそのメンテナンスを奨励するインセンティブ・プログラムを実施していますが、そのような組織はごくわずかです。この結果、私たちが日々使用するソフトウェアを基盤として支えているオープンソースが、時には何年間も手つかずのまま放置されるという事態を生んでいます。

コードベースのサステナビリティ



放棄されたオープンソース・プロジェクトの例としてここ数カ月で大きな話題となったのが Twitter です。同社でオープンソース・リードを務めていた Will Norris 氏は次のように述べています。「Twitter はオープンソース開発に背を向けました。Twitter でオープンソース開発に携わっていたキーパーソンは、ほとんどが社を去っています。私が一緒にオープンソース開発をしていたエンジニアも、誰一人残っていません。経営陣が刷新され、ビジネスの優先度が変わったことで、オープンソース・プロジェクトの優先度が下がり、より重要と見なされている新しいイニシアティブが優先されるようになっています」。

Twitter で起こりつつあるこうした状況で大きな問題となるのが、メンテナンスが止まることによる悪影響です。プロジェクトのメンテナンスをする人がいなくなると、良くて技術的負債、最悪の場合はセキュリティ・リスクが生じます。

既知のリスクを超えて

プロジェクトの放棄以外に、悪意によるオープンソース・プロジェクトの妨害行為という問題もあります。サプライチェーンのセキュリティに関する話題として、組織が既知の脆弱性を特定して軽減しているという話はよく聞きますが、サプライチェーンの分野で注目を集めつつある新しい形のオープンソース・ソフトウェア・リスクについては対処できているでしょうか。悪意のあるオープンソース・ソフトウェア・パッケージ、「プロテストウェア」、依存関係かく乱攻撃、タイポスクワッティングなどはいずれも、オープンソースのセキュリティに対する新種の脅威として懸念されています。

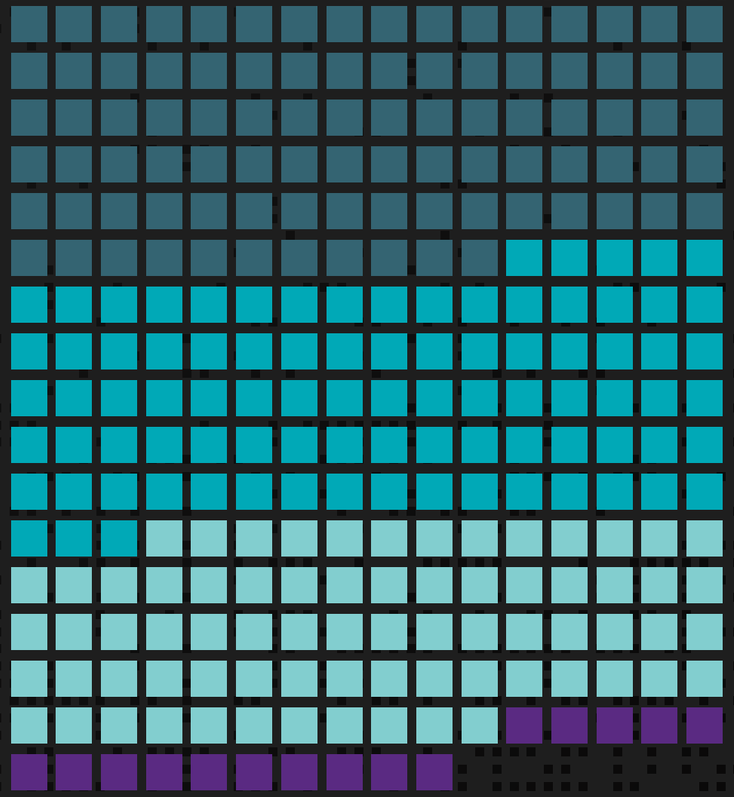
過去 5 年間ほどに起こった侵害および注目される脆弱性を振り返ったときに 1 つの共通するテーマとなるのが、「信頼」です。組織およびエンドユーザーは、程度はともあれ、自分が使用するソフトウェア、およびその開発と供給に携わった人々を信頼しているに違いありません。つまり、企業は自社のサプライチェーンのすべてのノードがセキュリティと品質に関して自社と同じ対策をとっていると信頼しています。しかし、これは危険な仮定です。

最近、シノプシスと Consortium for Information and Software Quality（CISQ）が共同で作成したレポート「米国における低品質ソフトウェアのコスト」では、2022 年のサイバー犯罪で最も顕著だったトレンドとして、ランサムウェア、クリプトジャッキング、IoT および OT 攻撃、サプライチェーン攻撃が挙げられています。ここでも、テーマとして見えてくるのが「信頼」です。私たちは、オープンソースのメンテナーや開発者が脆弱性を見つけて修正してくれるだろうと信頼しています。私たちは、オープンソースのコントリビューターが自分と同じ考え方や動機を持っているだろうと信頼しています。しかし残念ながら、常にそうとは限りません。ここで肝に銘じておきたいのは、運用リスクは既知の脆弱性だけにとどまらないということです。運用リスクを軽減するということは、将来の脅威を予測し、それを防ぐための対策をとるということでもあります。

2022 年夏、npm（JavaScript ランタイム環境 Node.js のデフォルトのパッケージ管理ツール）に悪意あるパッケージがいくつか見つかりました。これらは、モバイル・アプリケーションや web サイトに埋め込まれたフォームから機微なデータを収集します。これらのパッケージは、信頼できる有名なパッケージに似せた名前を付けており、数千回ダウンロードされています。これらのパッケージ（Icon-package、Ionicio、Ajax-libs など）は、どのバージョンも脆弱で悪意があると考えるべきです。

この例を読んで、あなたは現在自社で npm を使用していることに思い当たったでしょうか。あなたの組織のセキュリティ責任者はそれを適切に検査し、安全であることを確認したでしょうか。そうでなければ、あなたの信頼は過信かもしれません。使用するプロジェクトの評判と、誰がそのメンテナンスを行っているのかを明確に理解することは欠かすことのできない作業です。というのも、彼らがあなたと同じ動機でメンテナンスを行っているとは限らないからです。悪意を持ったメンテナーはサプライチェーンに対するあなたの信頼につけ込み、デュー・ディリジェンスを怠ればアプリケーションに悪意あるコードを挿入してくる可能性があります。

バージョン管理の問題



91% ―最新バージョンでないコンポーネントを含んでいたコードベースの割合

88% ―過去 24 カ月に活動実績のなかったコンポーネントを含んでおり、そのコンポーネントの最新バージョンを使用していなかったコードベースの割合

72% ―過去 24 カ月に活動実績のなかったコンポーネントを含んでおり、そのコンポーネントの最新バージョンを使用していたコードベースの割合

15% ―同じコンポーネントの異なるバージョンを 11 種類以上使用していたコードベースの割合

オープンソースの利用者によるメンテナンス

Black Duck 監査サービス・チームがオプションのリスク診断（p.7 参照）を実施した 1,481 のコードベースのうち、旧バージョンのオープンソース・コンポーネント（アップデートやパッチが存在するにもかかわらず、それらが適用されていないもの）を使用していたものが 91% ありました。

ソフトウェアが最新の状態に維持されていないのには、理由があります。DevSecOps チームは、新バージョンを適用することで得られるメリットよりも、意図しない結果を招くリスクの方が大きいと判断することもあります。組み込みソフトウェアであれば、外部ソースからしか混入経路のない脆弱性はそれほど大きなリスクにならないと考えることもできます。あるいは、時間やリソースの不足が原因の場合もあります。多くのチームは新規コードの開発とテストで既に手一杯で、特に深刻度の高い問題でなければ、既存のソフトウェアのアップデートを後回しにすることがあります。

しかしこの 91% のうち大部分は、おそらく DevSecOps チームがオープンソース・コンポーネントの新バージョンが利用可能になっていることに気付いていないか、そのコンポーネントが使われていることすら知らないか、そのどちらかと思われます。多くの組織が慣れ親しんでいる商用ソフトウェアでは、パッチやアップデートは自動でプッシュ配信されるため、自分でアップデートの提供状況をチェックする必要がありません。しかしオープンソースでは事情が大いに異なります。オープンソースを使用する以上、ユーザーが責任を持ってコンポーネントのセキュリティと安定性に気を配って管理すること、そしてコンポーネントの新バージョンやパッチが提供されていないかを自分で調べる姿勢が求められます。

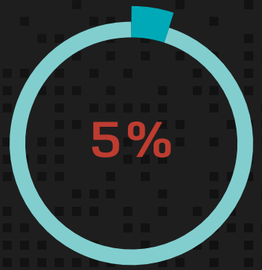
パッチの適用漏れ

発見から 1 年以上が経過した Log4Shell は、息の長い脆弱性の好例と言えるでしょう。メディアで大きな話題となった上に、コードベース内での存在を確認して修正する方法が数多くあるにもかかわらず、今年の監査でもこの脆弱性が検出されました。スキャンした全コードベースの 5% で脆弱なバージョンの Log4j が見つかっており、Java コードベースに限って言えば、11% で見つかっています。

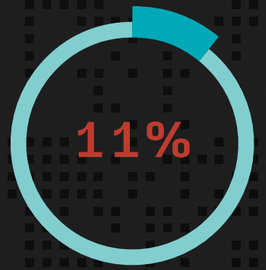
脆弱なバージョンの Log4j が今も見つかっているのは、そもそも Log4j を使用していることを把握できていないことが最大の理由と考えられます。Log4j は基盤的なコンポーネントであるため、セキュリティ・チームからは見えにくい存在となりがちです。また、基盤的コンポーネントへのパッチ適用は、表面的なクライアントサイド・コンポーネントへのパッチ適用よりも複雑であることが多いため、Log4j の存在には気付いているものの、まだパッチの適用に至っていないというケースもあるでしょう。

組織の基本的なコンポーネントに Log4j のような脆弱性が見つかるという状況は今後も変わりそうにありません。その理由としてまず挙げられるのは、オープンソースに対して組織が本質的な信頼を置いているという点です。つまり、オープンソースも商用ソフトウェアと同等の手法で同等のセキュリティ・チェックを受けているだろうという信頼です。次に挙げられるのは、組織がアプリケーションに含まれるオープンソース・コンポーネントを特定できていないという点です。これがセキュリティ・プラクティスの欠如によるものか、能力の欠如によるものか、その両方なのかは断定できません。

オープンソース、商用、独自開発コードに起因するビジネス・リスクの軽減に向けた第一歩は、その頒布元や取得方法にかかわらず、組織で使用しているすべてのソフトウェアの網羅的な目録（インベントリ）を作成することから始まります。完全かつ信頼できる SBOM という目録があって初めて、セキュリティ・チームは Log4Shell のような新しい脆弱性が開示された場合のリスクへの対処方法を含め、前進への行動計画を立てることができます。



監査したコードベースのうち、脆弱なバージョンの Log4j を含むものの割合



脆弱なバージョンの Log4j を含む Java コードベースの割合

[まとめ]

「信頼せよ、されど検証せよ」

戦時下に生まれたとされる「信頼せよ、されど検証せよ」という言葉は、サプライチェーン攻撃が相次ぐ現代に妙に当てはまります。自分が使用する、または開発するソフトウェアを信頼すること自体に問題があるわけではありません。問題なのは、そのソフトウェアが十分なセキュリティ分析を受けたかどうかを検証しないことです。

Gartner は「2025 年には世界中でソフトウェア・サプライチェーン攻撃を受けたことのある組織の割合は 2021 年の 3 倍に増加し、45% に達するだろう」と予測し、サプライチェーンのセキュリティに関して適切な戦術を導入することの緊急性を強調しています。このサイバー戦争の時代において、組織は予測不能な事態にどのように備えれば良いのでしょうか。

信頼にまつわる問題

前述したように、アプリケーション（またはそこに含まれるコード）のセキュリティを過信することは、セキュリティ侵害などの壊滅的な結果を招く要因となります。このような性善説に立った信頼を捨て、攻撃者のように考える姿勢を身につけることが重要です。脅威アクターとは、本質的に日和見主義者であり、常に最小の労力で最大の成果が得られるものを標的とします。使用しているソフトウェアがセキュアであることを確認もしていないのに、自社が格好の標的でないと言い切ることなどできるでしょうか。

攻撃の経路をすべて予測することは不可能ですが、ビジネス・リスクへの意識を組織全体で高めていけば、脅威を和らげることはできます。現在、あらゆる組織がソフトウェアに依存しており、その意味ではすべての企業がソフトウェア企業であると言えます。このような視点で自社のセキュリティを考え直してみると、使用しているソフトウェアがセキュアであると単純に信頼してしまうことのリスクがよくわかるはずです。外部から継承したものであれ、社内で開発したものであれ、自分で使用するソフトウェアのセキュリティは、自らの責任で確保する必要があります。

SBOM による検証

ソフトウェア・サプライチェーン攻撃に対抗する武器となるのが SBOM です。SBOM の概念はもともと製造業で生まれたものです。製造業では、ある製品を構成するすべての部品の情報を詳細な目録として記録したものを BOM と呼んでいます。BOM があれば、部品に欠陥が見つかった場合にメーカーはどの製品が影響を受けるかを把握し、修理や交換の作業を開始できます。同様に、オープンソース・コンポーネントの目録情報を正確に記録した SBOM を常に最新の状態に維持しておくことは、コードの品質、コンプライアンス、およびセキュリティを万全にする上で欠かせません。製造業の BOM の場合と同様に、オープンソース・コンポーネントの SBOM があれば、リスクのあるコンポーネントをピンポイントで即座に特定し、優先度の高いものから修正していくことができます。包括的な SBOM には、使用するアプリケーションに含まれるすべてのオープンソース・コンポーネントだけでなく、これらコンポーネントのライセンス、バージョン、およびパッチのステータスも記録されるため、サプライチェーン攻撃に対する理想的な防御策となります。

SBOM の目的は、オープンソースおよびサードパーティ・コードの使用を管理し、アプリケーションの「成分」を可視化し、この情報を標準的な方法で伝達できるようにすることにあります。ただし、SBOM は単に文書または記録としての基本的な役割だけに注目するのではなく、管理システム、あるいはツール、手法、手続きの集合体として見ることを推奨します。オープンソース・コンポーネントをその来歴も含めて特定した上で、これらのコンポーネントを脆弱性データに紐付けられるようにすることが重要です。効果的なオープンソース管理ができるようになると、サプライチェーン・リスク管理を成功させるための戦略および必要なセキュリティ・プログラムの改善についての意見交換も容易になります。

2022 年現在、商用コードの 96% にオープンソースが含まれています。したがって、使用しているアプリケーションに含まれるコンポーネントを可視化することは、現代の DevSecOps プログラムにとって必要最小限の要件と言えます。ビジネス・リスクと全体的なセキュリティへの見通しがきくようになれば、自社がセキュアであることを信頼するのではなく、確認できるようになります。

推奨文献

- ホワイトペーパー：社会的な規範を守る：GitOps とシフト・レフト・セキュリティ
- 記事：M&A Activity Looks Anemic Heading to Year-End as LBOs Shrivel
- 記事：Global M&A market slows in 2022 first half—but shows signs of strength
- Wikipedia：Trust, but verify
- プレスリリース：Strategic Direction for Cybersecurity Maturity Model Certification (CMMC) Program
- web ページ：Improving the Nation's Cybersecurity: NIST's Responsibilities Under the May 2021 Executive Order
- インタビュー：1:1 with Joe Jarzombek
- 記事：Twitter turns its back on opensource development

シノプシスの特色

シノプシスのソフトウェア・インテグリティ・グループが提供する統合型ソリューションは、ソフトウェア開発とデリバリのあり方を根底から変革し、ビジネス・リスクに対処しながらイノベーションを加速することを可能にします。シノプシスのソリューションにより、開発者はスピードを落とすことなくセキュアなコードを作成することができます。開発および DevSecOps チームはスピードを犠牲にすることなく、開発パイプライン内でテストを自動化できます。セキュリティ・チームは先手を打ったリスク管理が可能となり、組織にとって最も重要な問題の修正に集中できます。シノプシスは業界随一のノウハウを活かし、最適なセキュリティ・イニシアティブの立案と実行をご支援します。信頼性の高いソフトウェアの構築に必要なものをワンストップでご提供できるのは、シノプシスだけです。

SYNOPSYS®