THE LINUX FOUNDATION

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

ホワイトペーパー

OSS セキュリティのための動員プラン

(The Open Source Software Security Mobilization Plan)

翻訳：工内隆・伊達政広 (The Linux Foundation Japan)

目次

エグゼクティブ サマリー

現代のソフトウェア サプライチェーンは、基本的なコンポーネントもそのオペレーションもオープンソース ソフトウェア（「OSS」）に、あまねく依存しています。組織（企業や政府を含む）が、より速く、より高いレベルの品質で革新を促進させる能力は、多くの場合、OSS コンポーネントの採用と関係しています。ソフトウェアの「スタック」の約 70 ～ 90%は、OSS で構成されています[1,2]。それは OSS のメリットを共有していると同時に、OSS コンポーネントの脆弱性にさらされるというリスクも共有しているということです。

行政サービス、インフラストラクチャ プロバイダー、非営利団体、および大多数の民間企業は、ソフトウェアに依存して機能しており、広く使用されているソフトウェアの脆弱性と弱点は、現代社会全体のセキュリティと安定性に脅威をもたらしています。

ソフトウェアのサプライチェーンは複雑であり、物理的なサプライチェーンと同じように混乱や破損の影響を受けやすくなっています。民間部門は、物理インフラストラクチャ（港、送電網、電気通信ネットワークなど）と同様に、標準、共有サービス、およびソリューションを介してソフトウェアサプライチェーンの保護に絶えず投資し、不注意によるエラーと意図的な攻撃の両方のリスクを軽減しています。公的部門もシステムを堅牢にするために役割を果たすことができ、また果たすべきです。公的部門と民間部門は、これらの脅威に対処する上で、より大きな課題に直面しており、市民とステークホルダー共通のセキュリティと安全性のニーズに応えるために、ともに協力しなければなりません。

OSS サプライチェーンをセキュアにするための継続的な取り組みはかなり進んできていますが、満足できるレベルの回復力とリスク軽減を達成するためには、セキュリティ対策を現状の事後対応から事前対応へと移行するためのより包括的な一連の投資が必要です。私たちの目標は、より高度なセキュリティと信頼性を確保するために、OSS サプライチェーンで使用されるシステムとプロセスを進化させることです。

本稿では、ソフトウェア サプライチェーンを強化するための 3 つの基本的な目標に対して、すぐに取り組むことができる 10 のイニシアチブの包括的なポートフォリオを提案しています。行政サービス、インフラストラクチャ プロバイダー、非営利団体、および大多数の民間企業は、ソフトウェアに依存して機能しており、広く使用されているソフトウェアの脆弱性と弱点は、現代社会全体のセキュリティと安定性に脅威をもたらしています。

OSS を使用すると、OSS に関連したリスクを直接的、かつ体系的に軽減できます。そのようなことは、プロプライエタリ ソフトウェアではあり得ません。これは、使用しているソフトウェアのソースコードが利用可能であること、ほとんどの開発チームがオープンに作業していること、およびプロジェクト間で非常に多くのコンポーネントやコンセプトが再利用されていることに起因しています。

---

1 “2020 Open Source Security and Risk Analysis Report” by Synopsys: https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/2020-ossra-report.pdf and “2020 State of the Software Supply Chain” by Sonatype: https://www.sonatype.com/resources/white-paper-state-of-the-software-supply-chain-2020

2 “2022 Open Source Security and Risk Analysis Report” by Synopsis: https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html

OSS の開発、再結合、配布、および展開の方法に対する体系的な強化、および特に再利用頻度の高い「クリティカル」コンポーネントに対するスマートな投資は、すべてのダウンストリーム ユーザーのリスクを軽減するための、汎用性が高く、費用対効果の高い方法です。これには、OSS を組み込んでいる多くのプロプライエタリ ソフトウェアやカスタム ソフトウェア ソリューションも含まれます。

すべてのソフトウェアに欠陥があると想定する必要があります。すべてのソフトウェア開発チームは、最善の努力にもかかわらず、不注意でコードの欠陥を見逃してしまうことがあります。オープンソース プロジェクトを運営しているチームと協力することは、前述のいずれの目標を達成させるためにも非常に重要です。OSS プロジェクトのマネージャーは、通常「メンテナー」と呼ばれます。彼らは、より広い「コントリビューター」のカテゴリとは違う役割として、オープンソース プロジェクトのロードマップとリリースのタイミングを管理しています。メンテナーは、不正なコードをプロジェクトに挿入しようとする悪意のある試みを阻止したり、彼らのソフトウェアに暗黙的または明示的に組み込まれた他の OSS プロジェクトの変更がユーザーに害を及ぼさないように保護したりする、最前線の防衛線でもあります。

したがって、あらゆる形態の「投資」や「介入」は、OSS メンテナーに新しい価値を提供することに焦点を当てる必要があります。これには、彼らの作業のセキュリティと整合性を強化する手法を採用しやすくしたり、多くのプロジェクトがに苦労しているサードパーティによるコードレビューなどの資金調達活動までを含みます。開発者に追加の負担をかけたり、コード開発作業に対して、個人的または専門的な責任を増大させたり、資金の裏付けなく、命令や指示を出すようないかなる「投資」や「ポリシー」も、採用が進まないだけでなく、オープンソース ソフトウェアのさらなる発展の妨げになります。成功するアプローチは、単に助言するだけでなく、教育、サポート、および技術リソースも提供するといった、協力的なものでなければなりません。これは、変化への強力な推進力を生み出すことと、（提案の）採用を推進するために OSS コミュニティからの信頼を獲得することを、うまく両立させる上で非常に重要なものになります。

OSS を広範に利用している企業は、ずいぶん前から、サプライチェーンを通してどのような OSS が流入してくるかを知ることが、最優先課題であることに気づいていました。当初、これはリーガル面、およびソフトウェア ライセンスのコンプライアンスのためでしたが、現在では、次第に脆弱性を持つソフトウェア資産の追跡のためのものになっています。これらの企業は、利用している OSS プロジェクトへの関与を体系的に管理したり、あるいは企業のコードを専門的・技術的基準を守ってアップストリーム化したり、企業自身でリリースしたりするために、オープンソース プログラム オフィス（OSPO）を設置するようになりました。これらの企業は、セキュリティのベストプラクティスを重視し、セキュアなソフトウェアの開発の方法について従業員をトレーニングし、彼らが依存している OSS に対するセキュリティ監査に資金を提供し、さらには問題になる前に、問題や懸念のあるプロジェクトを特定するためのツールに投資する傾向があります。

これらのソフトウェア セキュリティのベストプラクティスをソフトウェア エコシステム全体に適用するときが来ました。世界中のほとんどの組織が、一般的に使用されている同じ OSS コンポーネントに依存しているため、OSS エコシステムは、適用を進めるために非常に重要な場となっています。

動員プラン：
全体目標とアクティビティ ストリーム（活動の流れ）

ホワイトハウスで 2022 年 1 月に開催された、民間部門、米国政府の専門家、および OSS ファウンデーション間の会議[3]では、3 つの包括的な目標が議論されました。

- **セキュアな OSS の作成：**第一に、コードとオープンソース パッケージのセキュリティ上の欠陥や脆弱性の防止に焦点
- **脆弱性の検出と修復の強化：**欠陥の検出、それらを修正するプロセスの強化
- **エコシステムのパッチ応答時間を短縮：**修正の配布、実装のための応答時間を短縮

本稿は、これら 3 つの目標のそれぞれを達成するために提案する一連の「アクティビティ ストリーム」を示すための有用な枠組みを提供すると考えています。各目標のストリームの概要を以下に示します。各ストリームを実装するための計画は、このドキュメントの最後にある付録でさらに詳しく説明されています。

このドキュメントで説明されている各ストリームの計画は、OpenSSF コミュニティの専門家の小さなチームによる数週間の取り組みから得られた成果です。今後、これらのストリームや全体計画を、以下のように進化させるつもりです。1）各ストリームの戦略に対して、さらなるデューデリジェンスを実行します。2）新しい取り組み、または私たちが気付いていなかった既存の取り組みを探します。3）一部の作業を自主的に実施できる可能性のある個人または組織を特定します。4）計画がより広く知られるようになるにつれて、他の要因も変更されます。また、これら計画のさらなる進化のために、より広範な OSS コミュニティ（個々の開発者、オープンソースのファウンデーション、および、いろいろな形態でオープンソースコードを利用している組織）の参加を切望しています。各ストリームに資金が確保され、計画をスタートさせる時点で、最終的に、利用可能な資金に合わせて、短期的な目標が調整される場合があります。本計画をレビューする際には、これらの「謙虚なスタート」と「変更に対するオープン性」を考慮して下さい。

これらの計画のいずれかが、OpenSSF に新しい技術的活動の立ち上げを求めたり、または OpenSSF から配布され成果物を生み出したりするようになると、OpenSSF の技術ガバナンス プロセス、特に技術諮問委員会（TAC: Technical Advisory Council）による監督・管理は、一貫した品質基準と他の OpenSSF の活動との整合性の確保のために有用な支援を行います。このドキュメントの内容は、特定の取り組みのための TAC への提案、またはそのガバナンスを迂回するための提案として解釈されるべきではありません。これらのストリームの一部は、他の既存組織、または新たな組織から、より効率的な代替案が提供されることもあるでしょう。

3 https://www.whitehouse.gov/briefing-room/statements-releases/2022/01/13/readout-of-white-house-meeting-on-software-security/

目標 1：セキュアな OSS の作成

セキュアなソフトウェアを作成するのは簡単ではありません。いくつかの異なる、重複する課題が問題を引き起こしています。

- 公式、非公式のコンピューター サイエンス教育カリキュラムは、通常、セキュリティ技術をカバーしていない
- セキュリティ研究者は、数十年前から存在しているプログラミング言語やシステムであっても、このエコシステムで新しい形態の脆弱性を定常的に検出
- 既存のセキュリティ ツールはセキュリティの問題を特定するのに役立つが、コード全体の品質とセキュリティの確保に実際に役立つには、トレーニングと専門知識を要する

多くの場合、セキュリティは最終状態ではなく、プロセス[4]として特徴付けられます。したがって、ここでの投資は、認識と教育の改善、クリティカルなコードのメンテナーが用いるプロセスの改善、およびソフトウェア開発のためにすべての開発者が使用するツールの改善に焦点を当てるべきです。

ストリーム 1：基本基準を満たすセキュアなソフトウェア開発の教育と認定をすべての人に提供

大学で教育を受けていたり、「ブートキャンプ」やコーディング アカデミーなどの他のフォーラムで教育を受けていたり、あるいは独学であったりしても、セキュアなソフトウェアを作成するための正式なトレーニングを受けているソフトウェア開発者を見つけることは、まずありません。適度な量のトレーニング（少なくとも 10 時間、理想的には 40 ～ 50 時間）が、開発者のパフォーマンスに大きな違いをもたらすでしょう。OpenSSF Secure Software Fundamentals などに、無料で利用できるトレーニング モジュールがあります。小さなチームを結成して、このようなトレーニング資料が業界標準と見なされるまで、繰り返し改善し、しかるのちに、コーディング アカデミーやキャリアクセラレーターを含め、あらゆる種類の教育機関や、あるいは従業員をこれでトレーニングし、また求職者が認定を取得していることを要件とする主要雇用者とのパートナーシップを通じて、これらのコースと認定の需要を喚起することを提案します。

費用：初年度は 450 万ドル、それ以降は年間 345 万ドル。

ストリーム 2：上位 10,000（またはそれ以上）の OSS コンポーネントを対象とした、ベンダーに中立で客観的指標に基づくリスク評価用パブリック ダッシュボードの開設

このストリームは、特定の OSS コンポーネントとそれを開発しているメンテナー チームがエンドユーザーのリスク軽減のために、プラクティスと手法に、どの程度適切に対応しているかを評価するために、さまざまなオープンソース評価データ（OpenSSF Best Practices Badge や Security Scorecard など）を収集します。このようなプラットフォームは、ソフトウェア構成分析（SCA：software composition analysis）によって、依存関係にあるコンポーネントの脆弱性を追跡し、アップストリームのコンポーネントのバグが他のコンポーネントにどのように影響するかを明らかにします。これは、OSS を導入する組織には「気づき」を提供し、また、リスクを軽減することにより、より多くのユーザーを引き付けたいプロジェクトには明確なガイダンスを提供します。

費用：初年度は 350 万ドル、それ以降は年間 390 万ドル。

4 https://www.schneier.com/essays/archives/2000/04/the_process_of_secur.html

ストリーム 3：ソフトウェア リリースでのデジタル署名の採用を加速

デジタル署名は、ソフトウェアの作成者からエンドユーザーまで、エコシステムのすべての関係者が、使用するコンポーネントが、実際に、使用を意図していたコンポーネントであることを確認するために重要なものです。署名はすでに配布先のエンドポイントで比較的一般的に使用されていますが、上流の開発プロセスではそれほど一般的ではありません。このストリームは、既存の署名ツールとインフラストラクチャの改善を推進し、オープンソース プロジェクトによるデジタル署名の採用を促進します。メンテナー向けのトレーニングと直接のソースコードのコントリビューションを通じて、デジタル署名されたソースのリリースを達成することが目標です。目標は、上位 200 のプロジェクトのうち 50 が、上位 10,000 のプロジェクトのうち 1000 が、相互運用可能なソフトウェア署名を使用するようになることです。

費用：初年度は 1300 万ドル、それ以降は年間 400 万ドル、初年度以降、1 回限りの 1,000 万ドル追加投資が必要。

ストリーム 4：メモリセーフでない言語を置き換えることで、多くの脆弱性の根本原因を排除

C や C++ などの一部のプログラミング言語は、メモリ安全性に脅威を与えており、ソフトウェアの欠陥を検出、除去することを困難にすることがあります。対照的に、Go や Rust などのほとんどのプログラミング言語は、デフォルトでメモリ管理や、その他のセキュリティにセンシティブなタスクを安全に処理するため、開発者は脆弱性のカテゴリ全体を簡単に回避することができます。現代のインターネットソフトウェア インフラストラクチャの多くは、C で記述されたソフトウェアで構築されており、毎年、多数の脆弱性が検出されています。2006 ～ 2018 年の Microsoft の脆弱性の約 70%はメモリ安全性の問題によるものであり[5]、2020 年に Google は Chrome の脆弱性の 70%がメモリ管理とメモリ安全性の問題によるものであると報告しています[6]。メモリセーフな言語で書き直すのに適した、小さくて、自己完結型のクリティカルなコンポーネントを特定し、書き直しを促進することで、このような弱点のカテゴリ全体を排除することができます。このストリームは、新しいコードベースを業界標準にするために、その開発作業、および関連するプロモーション、採用促進、コミュニティ開発活動にリソースを提供します。

費用：初年度は 550 万ドル、それ以降は年間 200 万ドル。

5 “Microsoft: 70 percent of all security bugs are memory safety issues” by Catalin Cimpanu, 2019-02-11, https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues

6 “Chrome: 70% of all security bugs are memory safety issues” by Catalin Cimpanu, 2020-05-22, https://www.zdnet.com/article/chrome-70-of-all-security-bugs-are-memory-safety-issues

目標 2：脆弱性の検出と修復方法の強化

現在、かなりの脆弱性の検出が、バグ報奨金からダークネットで取り引きされる「ゼロデイアタック」まで、高額な報酬を伴った状況で行われています。多くの場合、OSS のメンテナーは、十分なリソースを持った敵と競争して、自分のコードの深い箇所に潜むバグを見つけるためのリソースを持っていません。十分な資金、自動化、スケールメリットが出せれば、この不均衡に対処できます。

さらに、Eric Brewer (Google VP of Infrastructure & Google Fellow) は、「欠陥を検出することはそれほど難しいことではありませんが、欠陥を修正するのには、いつも、苦労している」と述べています。十分なリソースのある OSS 活動でも、既存のメンテナーが、たとえわずかな数であっても、新しい潜在的な欠陥を常にフォローできると決めてかかることはできません。検出するだけでは不十分なのです。修正のためのリソースも一体となって必要になります。ただし、まだ公開されていない脆弱性の修正は機密性が高く、慎重に取り組むべき作業であり、注意深いアプローチが必要です。

ストリーム 5：セキュリティの専門家が、脆弱性に対応するクリティカルな時に、オープンソース プロジェクトを支援するために、OpenSSF Open Source Security Incident Response Team（OpenSSF オープンソース セキュリティ インシデント対応チーム）を設立

OSS メンテナー コミュニティは、深刻なセキュリティ通知を受けると、その影響の大きさや複雑さに圧倒され、適切な方法で、迅速にバグを修正するのに大変苦戦することがあります。このような時に、セキュリティに精通し、その OSS プロジェクトで使用されている言語やフレームワークの詳細についてトレーニングを受けている熟練したソフトウェア開発者の小さなチームが非常に役立ちます。彼らには、非常に急な通知でコンタクト可能でなければならないし、作業はフルタイムで、数日または数週間に及ぶ可能性もあり、また、情報が開示されないように、雇用主などから機密保持のコミットメントを得なければなりません。一連のよく使われるプログラミング言語やフレームワークをカバーし、非常時に 2 ～ 3 人、そのような専門家を配置できるようにするには、30 ～ 40 人の専門家を抜擢し、トレーニングし、彼らをマネージメントする必要があります。

費用：初年度は 275 万ドル、それ以降は年間 305 万ドル。

ストリーム 6：高度なセキュリティ ツールと専門家のガイダンスを通じて、メンテナーと専門家による新しい脆弱性の検出加速

ほとんどのオープンソースのメンテナーは、脆弱性を早期に検出できるセキュリティ スキャンツール（ソフトウェア構成、静的分析、ファジング テストなど）を利用できません。多くの場合、ツールはプロプライエタリ製品であるだけでなく、それを繰り返し実行するためのサーバーのコストだけでも、ほとんどの人が負担できる額を超えるでしょう。このストリームは、主要なソースコード リポジトリ ホスト、セキュリティ ツール ベンダー、クラウド プラットフォーム プロバイダー、および OSS メンテナーの協調を実現させて、ベンダー中立な統一プラットフォームでスキャン・監視ツールを利用可能にさせます。また、それらのメンテナーは、潜在的な脆弱性と提案された修正を検証するために、有償の、セキュリティに精通し選抜された専門家と協力して作業を行います。これにより、メンテナーはコード内の新しい脆弱性を、より迅速に検出し、修正し、協調的開示を行うことができます。最終的には、上位 10,000 の OSS コンポーネントをカバーすることを目標としています。

費用：初年度は 1500 万ドル、それ以降は年間 1100 万ドル。

ストリーム 7：1 年に 1 度、最大 200 の最もクリティカルな OSS コンポーネントのサードパーティ コードレビュー（および必要な修復作業）の実施

セキュアなソフトウェアを作成するのは難しいことです。これは、非常に困難な作業であり、実際、世界で最も高く評価されているテクノロジー企業やオープンソース プロジェクトでさえ、後に、修正のために、ソフトウェアの更新を必要とするセキュリティ上の欠陥があるコードを作成することがあります。これらの欠陥は、コードを実際の環境（「本番環境」）に導入する前に、劇的に減らすことができるかもしれませんが、それでも脆弱性を見つけるために使用される多くの技術ツールが、最もクリティカルで複雑なバグのいくつかを見逃します。それは、なお人間の専門家の領域として残されています。

最もクリティカルな OSS ソフトウェアに対する、サードパーティのコードレビューと、関連する修復作業のマネージメントおよび促進のために、業界全体の協調的な取り組みを提案します。評価の高いサードパーティ セキュリティ企業に、最もクリティカルなオープンソース プロジェクトのコード レビューとセキュリティ監査を依頼し、OSS メンテナーと協力して、監査の結果に対する「公開レポート」を発行することにより、次のことを行います。1）まだ検出されていない影響の大きい脆弱性を見つけて修正します。2）重要なソフトウェア プロジェクトに対する開発者、業界、および公共部門の信頼を向上させます。3）監査品質を高い基準に設定します。4）見つかった脆弱性を修正し、関連するすべての調査結果を公開して、エコシステム全体に利益をもたらします。これは人手のかかる困難な作業ですが、監査が成功するたびにそれによって得られる利益は増大していきます。

初年度は最もクリティカルな 50 のプロジェクトをカバーし、2 年目以降は上位 200 をカバーすることを目指しています。

費用：初年度は 1,100 万ドル、それ以降は年間 4,200 万ドル。

ストリーム 8：最もクリティカルな OSS コンポーネントを決定する調査の改善のために、業界全体で広くデータを共有

どの OSS が実際に「クリティカル」であるかを客観的に決定する際の大きな課題は、使用状況、ダウンロード回数、および依存関係のデータが、その情報を収集することのできるソフトウェア ディストリビューション チャネルによって、かれらの所有する優位性だと見なされることが多いことです。「Harvard Census II」は、データ共有を促進することで、一定の進歩を遂げました。しかし、グローバルな OSS の使用状況とリスクを理解するためには、よりメトリック主導（metrics-driven）のアプローチを推進することにより、さらに多くのことを達成することができるでしょう。より多くのソースからのデータをセキュアにかつ安全に管理するためのリソースを提供し、さらにデータの解釈、および関連する研究に資金を提供することで、現場の実情に対する理解が大幅に向上し、「予期しない結果を得ること」のリスクが低減します。

費用：初年度は 185 万ドル、それ以降は年間 205 万ドル。

目標3：エコシステムでのパッチ適用応答時間の短縮

オープンソース プロジェクトの脆弱性を見つけて修正することは、問題に対処するための最初の重要なステップです。しかし、そのような取り組みの主たる目標は、これらのソフトウェア コンポーネントの修正バージョンが、当該コンポーネントを使用しているすべてのソフトウェアにあまねく、適用されることでなければなりません。これを実現させるには、エコシステムのすべての関係者、すなわちソフトウェア ベンダー、仲介業者、サービスプロバイダー、そして最終的にエンドユーザーのエンジニアリング チームとインフラストラクチャ チームが、当該 OSS コンポーネントがインフラストラクチャや彼らの提供している製品のどこにあるかを理解している必要があります。このような状況の中で、ソフトウェア部品表（SBOM: Software Bills of Material）が果たす役割と、公共政策レベルで SBOM が注目を浴びつつあることを十分に認識しています。ただし、エコシステムのすべての参加者が、使用するすべてのコンポーネントを最新の安全なバージョンに更新し、そのような更新されたバージョンのコンポーネントを下流のエコシステム参加者にタイムリーにリリースするためには、それを補完する取り組みが必要です。

ストリーム9：あらゆるところにSBOMを — SBOMツールとトレーニングを改善し、採用を促進

深刻な新しい脆弱性が検出されると、企業は多くの場合、それがあるかどうか、もしあるなら、どこに、どのような脆弱性があるのかを判断するために混乱に陥ります。多くの場合、使用しているソフトウェア資産のインベントリもなく、採用しているソフトウェア内のコンポーネントに関するデータも持っていません。さらに、組織はソフトウェアの入手を検討していても、多くの場合、そのソフトウェアのコンポーネントに既知の脆弱性が含まれているリスクを知る手段も持っていません。多くの人が、この問題を解決する基礎となる要素として「ソフトウェア部品表」（SBOM）を特定しています。しかし、この課題に適切に対処するには、SBOM の採用を広め、標準化し、可能な限り正確な内容にしなければなりません。

ツールとアドボカシ活動に焦点を当てることで、SBOM の作成、利用、およびあらゆる局面で SBOM を採用する上で、障壁となっているものを取り除き、オープンソース エコシステム全体（作成者、利用者、メンテナー）のセキュリティ体制（security posture）を改善することができます。このストリームは、開発者のチームにツール改善のためのリソースを提供し、すべての主要プログラミング言語にわたって多く利用されるソフトウェア ビルド ツールとインフラストラクチャにそれを組み込むことで、この問題に取り組みます。また、SBOM の作成と利用を常態化するための教材、トレーニング ビデオ、既定値のテンプレート（default template）、事例、およびその他のコミュニティによる推奨活動にもリソースを提供します。最も重要なことは、クリティカルなプロジェクトと直接連携するチームを用意して、SBOM サポートを追加するための改善を行い、惰性によるラストマイルでの障壁や抵抗を取り除くことです。このストリームは 3 つのポイントすべてに対応し、SPDX チームと直接連携して、それらを新しい SPDX セキュリティ プロファイルとして実装します。

費用：初年度は 320 万ドル、それ以降は未確定。

ストリーム 10：より優れたサプライチェーン セキュリティ ツールとベスト プラクティスを使用し、最もクリティカルな 10 の OSS ビルド システム、パッケージ マネージャー、およびディストリビューション システムを強化

オープンソース ソフトウェアは、パッケージ マネージャーを介してエンドユーザーに配布される前に、さまざまな言語固有のビルドシステムを使用してビルドされます。これにより、さまざまなエコシステムに由来するコンポーネントが本番環境に集ることになり、非常に異なる品質レベル、リスクレベルのソフトウェアがサプライチェーンを通して混在してしまい、リスク管理やリスク軽減に関して、統一されたポリシーを設定することが非常に困難になっています。

このストリームは、これらのソフトウェア成果物の配布に対して、私たちができる最も強力なセキュリティ強化を探究することを意図しており、コンポーネント レベルのセキュリティとエコシステム リスクに焦点を当て、他のストリームを補完し、パッケージ マネージャー レベルでの改善を推進します。

期待される成果は、オープンソース エコシステムに大きな影響を与えるレバレッジ ポイントを改善して、パッケージ管理システムにより高いレベルの可視性とセキュリティを持たせ、オープンソースの利用者が 彼らのオープンソース ソフトウェアの構成や来歴について、より大きな信頼を得られるようにすることです。

長期的には、構成データと来歴データに関するパッケージングや配布方法の改善により、欠陥検出と修正の時間を短縮し、ダウンストリーム ユーザーに対して脆弱性とパッチの透明性を向上させ、すべての開発者により良いセキュリティ ツールが提供されるようになります。

費用：初年度は 810 万ドル、それ以降は年間 810 万ドル。

総費用

各ストリームを提案した起案者には、実用的、かつ意欲的であること、既存の取り組みを拡大するか、または、OSS エコシステムの一部が実証済みの技術を残りの部分に適用すること、そして十分なリソースがあれば、開始して 2 年以内に達成可能な有意義な目標を設定することを条件としてそれに対するアプローチの策定をお願いしました。先に述べたように、これら費用を増額させたり、または減額させたりする影響要因は多くありますが、これらの目標を達成できるという確信を持つのに必要な大まかな投資規模を見積もるために、無駄のない現実的なアプローチで予算を作成するように依頼しました。以下に示す見積もりは、必要な投資の上下限の範囲に対する適切な期待値を設定していると感じていますが、それでも、増額、減額、どちらの方向にもおそらく50%の誤差があるでしょう。すべてのこのような作業には、時間、お金、スコープの 3 つの変数が存在します。資金調達、リーダーシップ、アプローチが確定し、「初年度」の時計が動き始めることを目指しており、その計画の作業が進行するにつれて、それら 3 つのすべての変数が進化していくことを期待しています。

その精神で、以下は、重要なこととして、攻撃や混乱の減少という形で、大きな利益をもたらす可能性が非常に高い投資ポートフォリオを提示していると考えています。私たちは、データ侵害（および罰金）の減少、アップグレードの急増によるシステムのダウンタイム、サイバーセキュリティ保険の支払いと保険料の引き下げなど、他の側面からも、投資が報われていることを示し、その価値を定量化してくれる学者や研究者と共同で作業することも切望しています。

ストリーム	初年度	次年度
1：基本基準を満たすセキュアなソフトウェア開発教育	450万ドル	350万ドル
2：OSSリスク評価用パブリック ダッシュボード	350万ドル	390万ドル
3：デジタル署名による強化された信頼性の提供	130万ドル	400万ドル
4：メモリセーフでない言語の置き換え	550万ドル	200万ドル
5：オープンソース セキュリティ インシデント対応チーム	275万ドル	300万ドル
6：新規脆弱性の検出、修正の加速	1,500万ドル	1,100万ドル
7：サードパーティによる監査、コード レビューと修正作成	1,100万ドル	4,200万ドル
8：クリティカル プロジェクト決定のためのデータ共用	185万ドル	205万ドル
9：SBOMの普及:セキュリティ ユースケース、ツール	320万ドル	未確定
10：ビルド システム、パッケージ マネージャー、およびディストリビューション システム	810万ドル	810万ドル
総額	**6,840万ドル**	**7,950万ドル**

結論

オープンソース ソフトウェアは、継続的に更新される形態で利用されるデジタル公共財として、社会全体に富と優れた機能を生み出しています。OSS の作成や、OSS をセキュアにするための民間部門の多額の投資は、公共の価値を生み出し、OSS は、高速道路や橋と同様に、クリティカルなインフラストラクチャとして位置付けられるようになりました。セキュリティを改善するための場当たり的な取り組みはもはや限界です。セキュリティを全面的に改善するための、これらの新しい、組織化されたアプローチに対する投資は、非常に大きな効果を生み出すと信じています。

私たちは、グローバルな OSS エコシステムに関与、または依存しているソフトウェアの専門家や組織と、このドキュメントを継続して改善することで、協力しあうことを切望しています。この計画を参照基準品質バージョン 1.0 以降のドキュメントとして完成させ、計画を実行するための必要なリソースを明確にすることをご支援ください。ぜひ operations@openssf.org まで連絡いただくか、OpenSSF コミュニティへの参加をご検討ください。

謝辞

このドキュメントは、以下の方々の支援と協力を得て作成されました。
Josh Aas (ISRG)、Andrew Aitken (Wipro)、Neil Allen (Morgan Stanley)、Pete Allor (Red Hat)、Gaja Anand (Morgan Stanley)、Abhishek Arya (Google)、Stephen Augustus (Cisco)、Mikael Barbero (Eclipse Foundation)、William Bartholomew (Microsoft)、Wayne Beaton (Eclipse Foundation)、Brian Behlendorf (Linux Foundation)、Aeva Black (Microsoft)、Stewart Blacklock (Intel)、Jeff Borek (IBM)、VM Brasseur (Wipro)、Josh Bressers (Anchore)、Eric Brewer (Google)、Bob Callaway (Google)、Hilary Carter (Linux Foundation)、Adam Cazzolla (Sonatype)、Mark Curphey (OWASP)、Vincent Danen (Red Hat)、Emmy Eide (Red Hat)、Jennifer Fernick (NCC Group)、Brian Fox (Sonatype)、Robbie Gallagher (Atlassian)、Kathleen Goeschel (Red Hat)、Sarah Gran (ISRG)、Mike Hanley (GitHub)、Jordan Harband (Coinbase)、John Heimann (Oracle)、Stephen Hendrick (Linux Foundation)、Trey Herr (Atlantic Council)、Justin Hutchings (GitHub)、Dustin Ingram (Google)、Shubhra Kar (Linux Foundation)、Amelie Koran (Atlantic Council)、Georg Kunz (Ericsson)、Rao Lakkakula (JP Morgan Chase)、Arnaud Le Hors (IBM)、Adrian Ludwig (Atlassian)、Brandon Lum (Google)、Frank Nagle (Harvard Business School)、Gunnar Nilsson (Ericsson)、Declan O'Donovan (Morgan Stanley)、Bruce Mayhew (Sonatype)、Jonathan Meadows (Citi)、Amir Montazery (OSTIF)、Tim Pepper (VMWare)、Alex Quesada (Wipro)、Tracy Ragan (DeployHub)、Sumod Rajan George (Wipro)、Xavier René-Corail (GitHub)、Christopher Robinson (Intel)、Chris Rohlf (Meta)、Trevor Rosen (GitHub)、Matt Rutkowski (IBM)、Stewart Scott (Atlantic Council)、Michael Scovetta (Microsoft)、Clyde Seepersad (Linux Foundation)、Azeem Shaikh (Google)、Nell Shamrell-Harrington (Microsoft)、Ax Sharma (Sonatype)、Laurent Simon (Google)、Jim St. Clair (LF Public Health)、Zach Steindler (GitHub)、Jason Swank (Sonatype)、Steve Taylor (DeployHub)、David Stewart (Intel)、Kate Stewart (Linux Foundation)、Andrew van der Stock (OWASP)、Eric Tice (Wipro)、Santiago Torres-Arias (Purdue University)、John Viega (NYU School of Engineering)、David A. Wheeler (Linux Foundation)、Michael Winser (Google)、Chris Wright (Red Hat)、Justin Young (Sonatype)、Alexios Zavras (Intel)。

付録 0

ワークストリーム プランにおける人員配置費用

以下の付録は、OpenSSF コミュニティの多くのコア参加者と、コミュニティ外の個人や組織が共同で実施する活動を説明しており、OpenSSF、およびより広範なオープンソース エコシステム内の既存のワーキング グループやその他の活動を踏まえて作成しています。ただし、現在活動しているものだけに限定されるものではありません。それらの計画は可能な限り、既存のプログラム、システム、プラクティス、その他の活動を参照し、「すぐに開始」できるようにしています。

これらの計画の多くは、サービスやツールの形でソリューションを開発することを求めており、これらのサービスやツールに関連するコストの大部分は（ホスト型または共有型のソフトウェアサービスとして提供される場合でも）、提案された方式を詳細化し、設計し、構築し、配備し、サポートし、さらに普及させるために必要な人員の費用が占めます。有能な人材の採用は好景気の時期には特に困難であり、これらの技術分野における報酬は高くなり続けています。効率的な理解のために、これらの計画では、活動の成功を目指して、高い確度で機能と時間の目標を達成するのに必要な事項を十分に理解した人員が担当することを想定しています。しかし、ほとんどの場合、各ワークストリームの提案されたアプローチにチーム要員を採用するには、3 ～ 6 か月、場合によっては 9 か月かかることが予想されます。そのため、最初の「年」のコストは、実際には、プロジェクトの初日から平均 12 ～ 18 か月以上かかることもあることを理解すべきです。人員の採用は、それぞれのワークストリームが提示する時間枠内で目標を達成することに対する唯一最大のリスクである可能性があります。

また、同業に引けをとらない福利を提供することから休暇の扱いまで、どんな組織にも存在する人事、会計、その他の業務間接費など、予算として考慮しなければならない給与外のコストが発生することもあります。これらについては、効率性のために本書では個別に説明しませんでした。

このため、これらの計画の各コンポーネントの「初年度」を含むすべてのテクニカル業務に対して、**プロフェッショナル スタッフ総費用**（間接費込み）として年間 $300K* の簡易値を採用しています。これは、採用されるすべての人員に同じ給与が提供されること、あるいはまた、職種に関係なく給与が $300K であることを意味するものではありません。場合によっては、実際に $300K の給与を上回る職種もあります。しかし、テクニカル ポジションとリーダーシップ ポジションが混在する場合の平均としては妥当であると感じられました。（* 訳注：本文書の付録部分では、原文に合わせて、費用の表現を $300K や $4.5M のように表記しています。それぞれ、30 万ドル、450 万ドルを意味しています。）

これらの計画で想定されているさまざまなチームに対する企業からのボランティア「出向」も利用して、これらの人件費を削減する方法を検討したいと考えています。しかし、非常に機密性の高い情報が含まれる活動では、これは不可能です。また、どんな活動においても、そのようなボランティアは、採用または直接契約した個人と同じ要件、同じパフォーマンスを満たす必要があり、一定の月数、週当たり何時間かの労働が必要になるでしょう。しかし、企業によっては、長期休暇の代替として、または 1 年間のトレーニングとして、これらの活動にスタッフをボランティアで参加させることには非常に価値がある場合もあります。

付録 1

ストリーム 1：基本基準を満たすセキュアなソフトウェア開発の教育と認定をすべての人に提供

問題：

歴史的に、従来のソフトウェア エンジニアリングのコースワークでは、正しいサイバーセキュリティ手法とセキュアなコーディング技術の重要性を強調したり、また教育したりすることにはほとんど注意が払われていません。訓練を受けた開発者がいないという問題を悪化させているのは、開発者がコードを作成 / テスト / リリースするのを支援することのできる訓練を受けたサイバーセキュリティ専門家の不足が厳しさを増しているからです。OpenSSF は、この問題に対処するために、コンテンツの収集と管理、トレーニングの拡大、開発者への報酬とインセンティブの提供という多面的なアプローチを提案します。

提案アプローチ：

1. コンテンツの収集と管理

私たちは、セキュアな開発プラクティス、最新のコード管理、デプロイメント方法論、（開発者用の）ソフトウェア コンポーネントの選択などに関する教育資料、さらには、開発言語、固有技術のドキュメントとトレーニング資料でレビューや手直しが必要なものを収集し、管理し、改善し、必要なら開発します。OpenSSF が提供する Secure Software Development Fundamentals Courses（セキュア ソフトウェア開発基礎コース）は、この作業の重要な基礎となります。可能な限り、これらの優れたプラクティスは既存のプロジェクト、またはツールのドキュメントに追加されます。ギャップ分析を行い、収集または作成を要する新たな資料が特定されます。以下に例を示します。

- OpenSSF、OWASP、業界団体、および主要なオープンソース プロジェクトから既存資料が収集され、必要とされるコンテンツとギャップがないかの分析
- セキュアな開発手法に関する OER (Open Educational Resource) ライブラリの作成
- 開発スキルやテクニックだけでなく、アクセス制御、テストと検証、DevSecOps (Development, Security, Operation) シナリオでのソフトウェアの構築、デプロイ、保守方法など、現代の開発者が必要とする関連スキルにも焦点

2. トレーニングの拡大

OSS コミュニティとソフトウェア業界全体のソフトウェア開発者のスキルを向上させるためには、業界全体にトレーニングを提供することが不可欠です。トレーニングの提供においては、積極的なパートナーシップ戦略を提案し、さまざまな組織や開発者イベントの主催者と協力します。また、個人のセキュアなソフトウェア開発スキルのコンピテンシーを実証するためには、一貫したアプローチが重要であると考え、世界経済が必要とする人材を創出するために、相互運用性を保証した、統一認証制度とバッジ取得プログラムを推奨します。

そこで、以下の具体的な行動を提案します。

- 1 年間に少なくとも 4 回、FOSDEM、SCALE、Open Source Summit などの主要なオープンソース開発者会議、またはセキュリティ会議において、上記トレーニングを提供する。

- このコース教材を提供するために、資格のあるトレーナーのグループを立ち上げる。
- 教育コンテンツの配信のために、既存の配信ストリームをスポンサーするか、代替配信ストリームを立ち上げる（たとえば、OpenSSF Secure Development Podcast、毎月開催のウェビナー、特定の開発トピックに関するより詳細なトレーニング ビデオ、ソートリーダーのブログなどのポッドキャスト）。
- 既存の Linux Foundation 認定インフラストラクチャをベースにしてトレーニングがどこで行われたかにかかわらず、認定やバッジを提供するための統一された仕組みを立ち上げる。
- SKF やその他の開発者教育ツールのラーニング ラボを拡大する。.
- Historically Black Colleges and Universities (HBCUs)、コミュニティ カレッジ、専門学校、高等学校、「Girls Who Code」、Codebar、Code2040、その他の類似組織と 5 ～ 10 件の有意義なパートナーシップを構築し、オルタナティブな非大学環境でコンテンツを提供する。
- 大学のカリキュラムにセキュアなソフトウェア開発コースを追加するために、主要な大学と 5 ～ 10 件の有意義な関係を構築する。
- ACM や IEEE と協力して、セキュアなソフトウェア開発資料を一般的なソフトウェア開発カリキュラムの推奨事項および認定要件に組み込む。

3. 開発者の報酬とインセンティブ

プロフェッショナルなソフトウェア開発者のかなりの数、おそらく大多数は、自己学習をしており、彼らの専門分野の教育修了の認定の機会を逃しています。さらに、多くの大学は、ソフトウェア開発関連の学位のカリキュラムの一部として、セキュアなソフトウェアの開発方法を学ぶことを要求していません。キャリアの浅い開発者は、他の報酬に直結した事項と比較して、教育コースを受講し、認定を得るために充てる時間を正当化できないのかもしれません。
開発者がこれらの教育コース（単にセキュアなコードを作成できるようになるレベルを超えたもの）を受講するための積極的なインセンティブが求められています。これらのインセンティブの状況は、トレーニングへの投資の提供範囲と影響を定量化するためにも使用できます。

- OSS 開発の主要な「ハブ」（GitHub や GitLab など）と協力して、それらの「ハブ」で活動するオープンソース プロジェクトのコントリビューターやメンテナーの認定状況や修了実績を表示するようにし、主要な OSS プロジェクトのメンテナーがこれらのコースを修了するように動機づけする。
- 認定取得が最も必要な OSS のコア メンテナーに金銭的インセンティブ（資産調査に基づく）を提供。年間平均 $1,000 のインセンティブと 200 人の開発者へのリーチを提案する。
- 主要な求人掲示板や求人サイト（Indeed や LinkedIn など）と協力して、求職候補者の履歴書を閲覧する際にこれらの認定が示され、かつ正確で最新のものであるとの信頼を獲得する。
- 階層的なバッジ取得プログラムを立ち上げ、バッジに対応した報酬を奨励する。
- 新しいバッジ取得、あるいは認証プログラムを既存の Linux Foundation や OpenSSF のバッジ プログラムに統合する。
- 開発者を採用しようとする組織に対して、バッジを取得した、あるいは認証を受けた開発者を強調して推奨する。

初年度（6 ～ 18 か月）の目標とコスト

- 既存のコンテンツとトレーニング資料（トレーナー養成を含む）の拡大および改善（2 人 x $300K）：$600K
- 教育する側の主導による、個人を対象としたトレーニング資材（ラボでの指導を伴う）を開発、およびトレーニング促進のための会議への参加（4 人 × $300K）：$1.2M
- 5 つの重要な開発言語とそれらのエコシステムをより深くカバーするための教材を拡張（例 :JavaScript/Node/Typescript、Python、Java、Rust、Go など）（0.5 人 × 5 × $300K）：$750K
- 講義形式ではない教材に基づいてラボを拡張（(2 人 × $300K）：$600K
- ブログ、ウェビナー、ポッドキャストを通じてセキュアな開発のソート リーダーシップを拡大（1 人 x $300K）：$300K
- 既存の認証システムやバッジ取得プログラムを拡張し、ソースコード リポジトリ、OpenSSF Tooling（Best Practices Badges、Scorecard）、および LinkedIn のようなシステムに表示して、デジタル社会における認証所得者やバッジ取得者への報酬拡大を推進：$200K
- コンテンツ配信に向けた関係性構築するためにパートナーシップオフィスを開設し、スタッフを配置（2 人 × $300K)：$600K
- 教育コースを受講しパスするための（資産調査に基づく）奨学金 $200K* と管理間接費 $50K の合算：$250K
  （訳注：ここは**原文**では $300k ですが、この原文の文末の値 $250k やトータルの値から正しくは $200k と思われるため、訳文では $200K にしました。）
- **合計：$4.5M**

初年度以降の年間目標とコスト

- 最も重要な教材を多言語化する（5 言語 x $200K/ 言語）：$1M
- セキュアなソフトウェア コンテンツの採用拡大のために、組織や大学との関係をマネージメントする（2 人 × $300K）：$600K
- Open Educational Resource プラットフォームにセキュア開発プラクティス ライブラリを開設し、資料を収集する（1 人 x $300K + インフラストラクチャ $100K）：$400K
- さまざまな認証資格を取得した人の数と、認証資格を取得したメンテナーがいる OSS プロジェクトの数を経時的に追跡するシステムを立ち上げ、その情報を容易に利用できるように他の組織と協力する：$100K
- 認定バッジと報奨プログラムを拡大する：$500K
- パートナーシップ オフィスを継続する：$600K
- 開発者向け奨学金を準備する：$250K
- **合計：$3.45M**

付録 2

ストリーム 2：上位 10,000（またはそれ以上）の OSS コンポーネントを対象とした、ベンダーに中立で客観的指標に基づくリスク評価用パブリック ダッシュボードの開設

問題：

オープンソース ソフトウェアのサプライチェーンにおいては、標準化されたリスク評価とモニタリングを行うためのソリューションの欠如が、さまざまなステークホルダーに循環的な問題を引き起こしています。

OSS 創設者とメンテナーは、脆弱性が悪用された場合、開発成果の評判やダウンストリームの導入に悪影響を与える可能性があるので、オープンソース プロジェクトやオープンソース パッケージのセキュリティ体制を改善するために苦労しています。

エンドユーザーの開発者は、評判のよいオープンソース パッケージをアップストリーム依存ソフトウェアとして組み込んでいるため、自身の製品やプロジェクトをリスクにさらしています。どのアップストリーム パッケージが安全であるか、または脆弱であるかが判断できないからです。

ビジネスやミッションクリティカルなソフトウェアの一部としてこれらのオープンソース パッケージを使用している**組織**は、攻撃者からの多数の攻撃ベクターにさらされ、さらにビジネスが中断する危険性にさらされています。

提案アプローチ：

OSS の開発者と利用者にリスクに関する「状況認識」を提供するために、主要な開発言語とエコシステムの上位 10,000 件のオープンソース プロジェクト、およびパッケージを継続的に分析する、ベンダー中立な、リスク評価と健全性監視のためのプラットフォームを構築することを提案します。私たちは、可能な範囲として上限を設定するのではなく、初期コストと期待値の基礎データを獲得するために 10,000 件のプロジェクトを目標として提案します。時間と資金が許す限り、より多くのプロジェクトに拡大していきます。

このプラットフォームでは、以下のように各プロジェクト / パッケージの Key Risk Indicators（主要リスク指標）を提供しますが、それらに限定するわけではありません。

- **アクティビティ評価：**採点エンジンは、コード アクティビティ、採用状況、コントリビューターの拡大と維持、企業・組織による関与、ダウンストリームにおける依存状況などを指標として評価する。
- **脆弱性とベストプラクティスの評価：**採点エンジンは、プロジェクト コードにおける公に宣言された脆弱性と未発表の脆弱性、依存するアップストリーム パッケージの脆弱性、コード シークレットの暴露、コミュニティのリスク評価、ベストプラクティス バッジ レベルなどを指標として評価する。

- **コンプライアンス評価：**ポリシー エンジンが、コードベース（および依存関係）で使用されるライセンスを検出し、プロジェクトのライセンス ポリシーで定義されたコンプライアンス リスクのレベルに応じて、また可能な場合はソフトウェア属性やその他のコンプライアンス基準に基づいて評価を行う。プロジェクトの SBOM アーティファクトも自動生成される。

技術概要

信頼できる目標とコスト見積もりを作成するために、この計画は Linux Foundation の既存の運用中プラットフォーム LFX Security and Insights platform 上に構築することを予定しています。資金を得ることができれば、この取り組みの初期段階で、最も費用対効果の高い、適切なスターティング ポイントが選択されるよう、多くのアプローチが公に議論され、検討されます。
LFX は、Linux Foundation がホストするオープンソース プロジェクトの管理、分析、およびリスク管理を行うためのプラットフォームです。LFX のセキュリティ機能は、現在はリスク関連の限られたデータを追跡していますが、LFX のハブ & スポーク アーキテクチャを通じてより多くのデータソースをサポートするように拡張できます。
評価データ プロデューサーは、コネクター フレームワークを介して公開あるいはクエリされます。すべての評価データは LFX プラットフォームの Data Lake House (DLH) に格納されます。複数の「プロセッサー」と「ビュー」が DLH 上に記述され、さまざまな利用者のためのダッシュボードのようなユーザー インターフェイスと、サードパーティ システムへのアウトバウンド統合を可能にする API を提供します。

OSS パッケージのスキャンは、最小 1 週間 (通常スキャン)、最大 1 日 (本システムへの組み込み当初) の頻度で行われます。一度組み込まれると、1 時間以内にダッシュボードで評価データを使用できるようになります。

データセットとプロデューサー（既存のコネクター、および今後予定されるもの）

- LFX Insights、および OpenSSF の ScoreCards と Criticality Scores（マージ予定）
  - コード開発アクティビティ（コミット、PR、イシュー、リリースの件数）
  - 採用（コンテナ レジストリ、パッケージ マネージャーからのプルの件数）
  - 新規コントリビューター、離脱したコントリビューター、コードチャーンの状況
  - Scorecards 情報
  - 可能であればファイアウォール設置（stackshare.io とともに LFX コネクターを使用）
- 依存関係分析 — Snyk/OSV/Deps.dev/libraries.io
  - 一時的なものを含む依存関係とアプリケーション スタック マップ
  - 依存関係における公開された脆弱性（CVE、CWE）
  - 依存関係のライセンス
- 静的分析 — Blubracket/Snyk/LGTM(CodeQL)/Sonatype Lift/ その他の OSS
  - パッケージのベースコードで公開された脆弱性（CVE、CWE）
  - 未公開の潜在的な脆弱性（コード品質）
  - コード シークレットの暴露

- ベースコードのライセンス
- OpenSSF ベストプラクティス バッジ
- 自動 SBOM ジェネレーター（参照：https://github.com/opensbom-generator/spdx-sbom-generator ）

OSS パッケージのソースコードは、ソース管理システムからプラットフォーム内のローカル コンテナ ワークスペースに毎日クローンされます。データ プロデューサー（例 :Snyk CLI、Blubracket CLI、libraries.io）のスキャン エンジンは、一体化されたスキャン エージェントにパッケージされ、これらのコードベースに対して実行されます。

本システムに組み込まれたパッケージには Admin ユーティリティが提供され、変更制御の管理、およびスキャン インフラストラクチャの保守を行います。LFX は、このニーズに迅速に対応するために、プロジェクト コントロール センター（PCC）ツールを再利用できる可能性があります。Admin ユーティリティの使用はロールベースのアクセス制御（RBAC）が適用され、コミュニティ管理者およびプロジェクト メンテナーが使用できます。

このアプローチは、LFX プラットフォームがスキャン コンテナ インフラストラクチャを保守し、また規模拡大する必要があるため、データ プロデューサーが自律的にスキャンして評価データを DLH で公開するよりもコストがかかります。しかし、SCM システムのパッケージに対してメンテナーがボットやベンダーエージェント / アプリをインストールする必要がないため、より実現可能なアプローチだと考えられます。

モデリング

重要性、リスク、コンプライアンスの領域を横断的に採点するデータのモデリングは、各プロジェクトに特化した、メンテナーの設定で変更可能なポリシー ベースの重み付け機構を使用して実行されるようになります。

ユーザー エクスペリエンス

複数の利用者（メンテナー、ガバナンス グループ、スポンサー組織）向けのダッシュボードが作成されます。ロールベースのアクセス制御（RBAC) は、高いレベルのアクセス権限とアクセス許可を実現します (たとえば、特定の人だけが潜在的な未知の脆弱性に関する情報を見ることができる)。LFX IAM（現在 LFX Security で使用されている）は、RBAC の基盤として使用される見込みです。広くコミュニティで使用するためのパブリック アクセス ダッシュボード (集合化され、匿名化されたデータ) も公開される可能性があります。必要に応じて、出力 API が構築され、組織が評価データと採点状況をクエリすることをサポートします。

初年度（6 ～ 18 か月）の目標とコスト

- LFX プラットフォームに対して提案された拡張機能の提供
- Criticality Score やその他の選択基準を使用し、最も重要なオープンソース プロジェクト1,000件の本システムへの組み込み。（注：プロジェクトとしては、ただ 1 つのリポジトリのものから数百のリポジトリのものまで対応可能。たとえば、Kubernetes には 5 つの github オーガナイゼーションにわたる 196 のリポジトリがある。1,000 件のプロジェクトは 10,000 以上のリポジトリに相当することもありうる。この予算では、1,000 件のプロジェクトと 10,000 のリポジトリを想定している。）
- グローバル レベルおよびプロジェクト レベルのダッシュボード。

- プロジェクト / パッケージのスキャンを迅速に実行・保守するための Admin ユーティリティ。
- **開発：**
  - バックエンドに重点を置いたフルスタック エンジニア 5 名 — コントラクター
  - フロントエンドに重点を置いたフルスタック エンジニア 3 名 — コントラクター
  - プロダクト マネージャー1名 — スタッフ
  - UI/UX デザイナー1名 — スタッフ
  - 技術リーダー1名 — スタッフ
  - QA エンジニア1名 — コントラクター
  - コスト :$2.5M（12 人 × $200K/ 年）
  - 注：ここでの見積もりは、このドキュメントで定義されている最小範囲に基づいている。範囲は変わることがあり、追加要件が人員配置のニーズに影響を与える可能性がある。
- クラウド運用費用 : $100K（下の表を参照、バッファーを加算）
- 本システムへの誘導とアドボカシ活動：(3 名 x300 K/ 年) = $900K
- **合計：$3.5M**

初年度以降の目標と年間コスト

- プロジェクトプラットフォームの保守と機能の進化
- 最も重要なオープンソース プロジェクト 10,000 件と 10 万以上のリポジトリの組み込み
- サードパーティとの統合を可能にする出力 API の開発
- 成功した機能は常に新しい要件を生み出すので、開発と保守にはほとんど同じスタッフが必要。10 人のスタッフと $2M
- クラウド運用費用：$1M/ 年（追跡対象プロジェクトとリポジトリの数を 10 倍拡張することに基づく）
- 前述と同じ注釈：ここでの見積もりは、このドキュメントで定義されている最小範囲に基づいている。範囲は変わることもあり、追加要件が人員配置のニーズに影響を与える可能性がある。
- 継続的な本システムへの誘導、プロジェクト支援、アドボカシ活動：3 人、$300K/ 年 = $900K
- **合計：$3.9M**

インフラストラクチャに関する注

前述したように、オープンソース プロジェクトにはパッケージ当たりのリポジトリ数が 1 ～ 100 範囲で存在する可能性があります。1 プロジェクト当たりリポジトリ数の平均を”10”として外挿した場合、1 万件のプロジェクト（10 万リポジトリ）のインフラストラクチャ コストは年間 $820k になります。できれば、2 年目以降は年間 $1M 以上の予算を立てるべきです。

リポジトリ	インジェスチョン/月	ストレージ/月	コンピュート/月	可視化/月	月間合計	年間
10K (初年度)	$1,780	$1,250	$3,600	$206	$6,836	$82,032
100K (2年目以降)	$17,800	$12,500	$36,000	$2,060	$68,360	$820,320

このデータは、6つのコネクターからのデータに基づいて生成される。追加のデータ プロデューサーまたはメトリック ストリームが必要な場合、コストはこれよりも高くなる可能性がある。

付録 3

ストリーム 3：デジタル署名を使用してソフトウェア サプライチェーンに強化された信頼性を提供

問題：

Web ブラウザーとサーバー間のプロトコルへの 20 年以上にわたる投資のおかげで、今日では、表示されるコンテンツが Web サイトから送信されたコンテンツと同じであることが保証されるため、Web サイト上のコンテンツはあまねくセキュリティが確保されています。このような保証は、ソフトウェア サプライチェーン全体ではあまり普遍的ではありません。デジタル署名が使用され、確認されても、「最後の 1 マイル」しかカバーしていないか、または、自動化および監査が困難な、異なったさまざまなアプローチを使用しています。ソフトウェア ディストリビューション用のデジタル署名は、増加しつつある広範な攻撃ベクターに対応するために、もともとの開発者や開発チームから、エンドユーザーやエンドデバイスに至るまで、エンドツーエンドでカバーし、さらに開発者が簡単に適用でき、ユーザーが簡単に確認できる必要があります。

デジタル署名を使用すると、人間または自動化されたシステム（コンピューティング「ワークロード」と呼ばれる）が、ソフトウェア ライフサイクルにおいて実行されたアクションの整合性を判断し、エンドユーザーが受信したすべてのビットが作成者の意図したビットであることの強力な保証を可能にします。ソフトウェア開発ライフサイクル（SDLC）全体で実施されるイベントにデジタル署名の作成と確認を要求することで、オープンソース ソフトウェアのアーティファクト（生産物など含む行為の証跡）を、そのソースコードの由来、それをビルドしたプロセス、およびそこに組み込まれた構成物にバインドすることができ、現代のシステムに比類のない透明性とセキュリティを実現します。

提案アプローチ：

過去のデジタル署名システムでは、採用コストが急騰するか、正しく使用することが非常に困難でした。 sigstore プロジェクト（意図的に小文字を使用）は、2020 年に設立され、シンプルで人間工学的な経験を通じて、あらゆる規模のプロジェクトが採用できる広範なソフトウェア署名を可能にすることを目指しています。同プロジェクトは、業界標準の暗号アルゴリズム、署名フォーマット、およびアイデンティティ プロトコルをサポートするモジュラーな設計に基づいています。

sigstore は、署名とそれに付随するインフラストラクチャを「摩擦のない」、「見えない」ものにしようとしています。活気に満ちた多様なコミュニティによって構築、保守されている多種多様なシステムに組み込まれようとしており、署名ソフトウェアの実装、RFC、あるいは Kubernetes、Maven Central、RubyGems、PyPI、npm からの採用意図表明により大きな牽引力を持っています。

sigstore のコンポーネント

sigstore には 3 つの主要なコンポーネントがあります。

1. **短寿命の署名証明書を発行するための認証局（CA）。**開発者、またはワークロードのアイデンティティを一時的、または長寿命の暗号鍵にバインドする。認証局は多くの信頼モデルと共存して動作する：スタンドアロン、OSS 基盤から派生したもの、sigstore の trust root を使ったものなど。
2. 有効な署名 / 証明の正確・不変で検証可能な履歴を保存する**透明性ログ。**これらは、現在、主要なブラウザーの SSL/TLS 証明書に使用されている既存の IETF 標準の Certificate Transparency の概念から派生したもの。
3. 既存のパッケージリポジトリやツール（Python PyPI、Ruby Gems、コンテナーレジストリなど）と「自然な」統合を提供する**エコシステム固有のライブラリ**とユーティリティ。

sigstore 証明書の基礎となるアイデンティティは、さまざまな供給源や技術実装から生成され、人間に限定される必要はありません。一例として、sigstore による GitHub Actions の直接サポートでは、検証された GitHub エンティティ アイデンティティに基づいて（オーガナイゼーション リポジトリ）、マシン間のアーティファクト署名とプロブナンス（来歴）の生成がすでに実証されています。

SLSA と sigstore

sigstore のツールが優れている分野の 1 つは、OSS プロジェクトが、GitHub Actions や CircleCI といったクラウドベースの CI（Continuous Integration）システムを介してオープンに構築されるユースケースです。そのようなシステムの多くは OpenID Connect(OIDC) アイデンティティ プロバイダーを選択しているため、アーティファクト作成時にマシン ワークロード アイデンティティにバインドされたプロブナンス（来歴）ステートメントを比較的簡単に作ることができます。これにより、人間にバインドされた暗号鍵や、問題のある長期間有効な証明書がソフトウェア信頼性の概念に不可分に結びつくシナリオを回避することができます。クラウドベースの CI システムで sigstore を使用することで、OSS メンテナーは、アーティファクトを作成する際に使用されたプロセスと基礎となるソースコードを証明し、比較的少ない労力で SLSA (Supply-chain Levels for Software Artifacts) レベル 3 を達成することができます。これにより、NIST SSDF (Secure Software Development Framework) の指針採用への迅速な道が可能になると考えています。

「公共財」sigstore：信頼と透明性のネットワーク

公共の場での摩擦のない経験をサポートし、OSS がより良いものへと根本的に変化する可能性を具現化するために、sigstore は、情報の一貫性と同一性を保証する透明性のある分散ネットワーク システムであることを明確に示す必要があります。第三者による独立した監視と監査が重要であり、そのために「公共財」としての sigstore インフラストラクチャは、複数のログと証人のネットワークとして構成され、そのことがそのネットワークが共有された状態であることを証明します。TLS 証明書の透明性インフラストラクチャと同様に、公共財としての sigstore ネットワーク内のノードは、緊急の脅威やその他の運用上の懸念に対する保証を提供するために、明示的なセキュリティと中立性の要件（たとえば、NIST などの政府標準への準拠、定期的なセキュリティ監査、さらには学術機関、政府、企業が混合したオーナーシップ状態など）を遵守しなければなりません。

2022 年、sigstore コミュニティは、1.0 版の公開に向けて CA と透過性ログプロジェクトの安定化に取り組み、一般公開の準備が完了したことを示しました。一方、公共財としての sigstore ネットワークは、初期の形はすでに存在しており、単一モニターではあるものの、自由に利用できる CA/ledger のインスタンスとして Purdue 大学によって運営されています。コミュニティはさまざまな産業や公共セクターのパートナーとネットワーク拡大のために活動しています。

T 拡張への次のステップは、各ノードの運営者が守ることのできる一連の強力な Service Level Objectives（SLO：サービスレベル目標）を確立することです。今日に至るまでの sigstore の力強い採用曲線に基づくと、私たちは、ネットワーク全体で記録され、独立して確認される必要のある署名事象が 1 日に数十万件（オープンソースソフトウェアの全体的な傾向とともに増加し続ける見込み）になると予測しています。これをサポートするために、sigstore コミュニティは、現在のネットワークノードの運用責任を引き受けることに関して、Internet Security Research Group (ISRG) と初期の議論を行っています。ISRG には、2 億 6000 万を超える Web サイトの TLS インフラストラクチャを強力に支える Let' s Encrypt などの同様の共有サービスを運営する、大きな信頼性と経験があります。

相互運用性による採用の促進

広範なシステム（ソフトウェアのデプロイメントやコンピュート デバイス）の署名をサポートすることが sigstore の主要な目標です。さまざまなアイデンティティ システムと署名フォーマットはそれぞれが異なる利点を提供しており、sigstore として勝者を選ぶようなことは行いません。高パフォーマンスのクラウド環境であれ、低電力の IoT デバイスであれ、sigstore は、個々の開発者やエンタープライズ ソフトウェア企業を含む OSS ソフトウェアの消費者と生産者の両方のニーズに対して、最大限の有効性と好影響を与えることを目指しています。

sigstore ネットワークを運用するにあたり、私たちは、**まさに今の状況で OSS メンテナーに会い**、影響力のあるプロジェクトに優先度をつけて、エコシステム全体にツールと接点部分に投資したいと考えています。 私たちは、これらのエコシステムへの対応のために、限られた（しかし広く適用可能な）一連のプロトコルと標準をサポートするために資金を投じようしています。一般的に、私たちは汎用的な相互運用標準の出現を支持しています。これにより、アーティファクト パブリッシャーはさまざまな構築システムを簡単に使用して「公開下で構築」し、公共財ネットワーク上で公開することができます。たとえば、現在または計画されている標準サポートの具体的な例としては、以下のものがあります。

- **署名フォーマット：**DSSE、COSE
- **アイデンティティ：**DiD、OIDC
- **証明書フォーマット：**X509、SSH

このアプローチでは、大企業に導入されている他の署名、および関連するアイデンティティ システムとの相互運用性も実現可能になります。これらは、大企業の内部システム、または大企業が構築および配布する製品全体にわたって同様のニーズに対応するために使われています。これらのシステムは消え去ることはなく、これらとシームレスな体験を可能にするための技術的なブリッジが推奨され、実装されることになるでしょう。

まとめ

オープンソース コミュニティは、OSS アーティファクトに署名するための実行可能なソリューションとして sigstore を選定し、署名システムを構築しました。sigstore は過去 18 年間で大きな牽引力とマインドシェアを獲得しました。sigstore ではまだ多くの作業が残されていますが、必要なツールキットとグローバルな採用へのクリティカル パスは明らかです。業界の境界を越えて力を合わせ、ソフトウェア署名の標準と相互運用性を促進するという確固とした哲学のもと、コミュニティは、世界中のすべてのソフトウェア コンシューマーの利益のために、ソフトウェア サプライチェーン インテグリティの深刻な問題の解決に取り組むことができます。

初年度（6 ～ 18 か月）の目標とコスト

- 公共財としての sigstore ネットワークの確立ー業界標準の運用（おそらく ISRG を通じて）による初期コアノード（ログとモニター）：立ち上げの固定費 $0.5M と運用費用 $1.5M/ 年
- 最も重要な OSS エコシステムのトップ 5（OpenSSF リーダーシップ チームによって決定）で sigstore サポートをネイティブに実装、および保守するために必要な開発作業 ：(エコシステムあたり $2M × 5) = $10M
- “DevRel (Developer Relations)” およびオープンソース プロジェクト、開発者、エンドユーザー向けの sigstore 利用拡大のためのアドボカシ活動：3 人 $1M
- **合計 :$13M**

初年度以降の目標と年間コスト

- sigstore ツールの採用を OSS エコシステムのトップ 5 からトップ 10 に拡大するための投資 :(エコシステム 1 つにつき $2M × 5) = $10M（1 回限り）
- サポートするプロトコル、アイデンティティ システムの拡張、およびネットワークのユーザビリティの全体的な改善 : 年間 $1M（毎年継続）
- ネットワーク内のノード数の増加、およびデジタル証明のトランザクション負荷とストレージの増加による継続的な運用コスト：年間 $2.5M（毎年継続）
- **合計 :$10M +$4M/ 年（毎年継続）**

付録 4

ストリーム 4：メモリセーフでない開発言語を置き換えることで多くの脆弱性の根本原因を排除

問題：

プログラムがメモリを誤って管理することが原因となって脆弱性が発生することはよくあります。この種の脆弱性は、「メモリ安全性」の脆弱性と呼ばれます。このような脆弱性が存在するのは、安全でない特定の開発言語（主に C と C++）において、プログラマーがメモリ管理上の誤りを容易に犯すことがあるためです。Rust、Go、Java などのメモリ安全性の高い言語では、プログラマーがメモリ安全性の脆弱性を引き起こすような誤りを犯す余地はありません。

Microsoft は、過去 10 年間の自社製品の脆弱性の 70% がメモリ安全性の脆弱性であると評価しています。Google は、Android の脆弱性の 90% がメモリ安全性の脆弱性であるとしています。その割合はオープンソース ソフトウェアでも同様であり、メモリ安全性の脆弱性が発表され、パッチが適用される流れが絶えません。

2021 年版の Google Project Zero によると、検出され、悪用されていることが明らかになった脆弱性を分析したところ、67% がメモリ安全性の欠如によるものであることがわかりました。「メモリ破壊の脆弱性は、過去数十年にわたってソフトウェアを攻撃する際の標準となってきたが、今もなお、攻撃者が成功している方法だ」と彼らは述べています。

これらの脆弱性やハッキングの結果は、技術的な影響だけではありません。それらの結果、多額の財政的損失（$B）、病院や送電網などの重要なサービスの中断、数百万人の個人生活や個人データへの侵入（たとえば、悪意を持ってホーム セキュリティ システム カメラを制御）が発生しました。

提案アプローチ：

他の多くの種類の脆弱性とは異なり、私たちはメモリ安全性の脆弱性を軽減するだけでなく、完全に取り除く方法を知っています。ソフトウェアを C や C++ から、より安全な言語に移行することで、すべての脆弱性の大きな割合を占めるメモリ安全性の脆弱性を排除できます。

メモリ安全性の脆弱性の数を減らすために私たちが計画している作業は、2 つの部分から構成されています。

1. 最もセキュリティに影響されやすいコンポーネントを最初にアップグレードすることに重点を置いた効率的な戦略を採り、インターネットで最もクリティカルなソフトウェアを C や C++ などの安全でない言語から遠ざける。
2. システム エンジニアがローレベルのシステムレベル コードをより安全な言語に移行できるようにするツールへの投資。これにより、最も重要なプロジェクト以外にも効果を及ぼすことができる。このようなシステムレベル コードはエコシステムの中で最も遍在的で脆弱なソフトウェアであり、C などの安全でないシステム言語で書かれている可能性が最も高いため、システムレベル コード用のツールに焦点を当てる。

重要なソフトウェア インフラストラクチャを安全な言語に移行

私たちの戦略のこの部分は、非営利の Internet Internet Security Research Group (ISRG) が運営する Prossimo プロジェクトに依拠しています。同プロジェクトは、より安全な言語に移行する必要がある最もクリティカルなソフトウェア インフラストラクチャを特定し、ステークホルダーとともに効果的で効率的な移行を計画し、計画の実行を監督します。

投資を判断するために Prossimo が使用する高レベル リスク基準は以下のとおりです。

1. 非常に広範囲で使用（ほぼすべてのサーバーやクライアント）
2. ネットワーク境界上で動作
3. クリティカルな機能の実行
4. メモリセーフではない言語（C、C++ など）で記述

これらの基準に適合するソフトウェアの中から、以下の基準で取り組み優先順位が評価されます。

1. これは多くの異なるプロジェクトで使用できるライブラリまたはコンポーネントか？
2. キー コンポーネントを既存のメモリセーフ ライブラリに効率的に置き換えることができるか？
3. 資金提供者はその仕事に資金を提供する意思があるか？
4. メンテナーが参加し、協力的か？

上記のすべての基準に基づいて、Transport Layer Security (TLS)、Domain Name System (DNS)、Network Time Protocol (NTP) に関連する Linux カーネルとアプリケーションおよびライブラリに対して、Prossimo での取り組みが進行中です。

より安全なシステム開発ツールへの投資

世界で最も遍在的でクリティカルなソフトウェアの多くは、世界中のほとんどすべてのコンピューティング デバイス（カーネル、基本的なネットワーク機能、時間管理など）の基礎となるローレベル システム ソフトウェアであり、銀行、病院、政府のシステムの中で動作しています。これらは通常、安全でない C 言語で書かれています。コンピューターの歴史の大部分において、C はシステム ソフトウェア開発のための主要言語だったからです。

ローレベル ソフトウェアは、ハイレベル ソフトウェアよりも運用上の制約が多いため（たとえば、ランタイム ライブラリやガベージ コレクションによるメモリ管理を許容できないなど）、システム ソフトウェアに適したメモリ安全性の高い言語の開発は特に困難です。しかし Rust 言語はその課題に対処しており、多くのシステム アプリケーションで C に代わる優れた候補となっています。

私たちは、システム レベルの開発者がソフトウェアを Rust に移行できるようにするツールへの投資を計画しています。これは、パッケージ管理、コンパイラー、Foreign Function Interface (FFI) ジェネレーターの改善に投資することを意味します。多くの場合、これには、移行を可能にするために、広く使用されている既存のコンポーネントと互換性のあるインターフェイスを提供することが含まれます。これらのツールを使用することで、作業を繰り返すことなく、はるかに迅速に、メモリ安全性の高い代替手段の採用が拡大します。

また、Go コミュニティや Java コミュニティにおける同様の取り組みに投資することも提案します。どちらの言語も、セキュリティが重視される環境を含め、システム レベルおよびネットワーク アプ

リケーションで頻繁に使用されています。私たちは、上記と同じ優先順位と基準を参考としつつ、一連の助成活動を通じてコミュニティに働きかけることを提案しようとしています。その際、プロダクション リリースや採用に向けてさらなるプッシュを必要とする既存のプロジェクトや、本計画の他のワークストリームを支援する可能性のあるプロジェクトに焦点を当てようと考えています。

これらの投資は、おもに ISRG、Rust 関連の活動については Rust Foundation、Java 関連の活動については Eclipse Foundation、そして Go 関連の活動については独立した助成金を通じて行われます。

初年度（6 ～ 18 か月）の目標とコスト

- RustTLS の改善と、Rust での DNS リゾルバーと NTPd の実装：$2.5M
- Rust やその他のメモリ安全性の高い言語への移行と開発言語の混在した開発を簡素化するための Foreign Function Interface (FFI) ジェネレーターの改善：$1M
- 上記の基準に基づいて、他の重要なソフトウェアを調査し、活動を支援し、Go と Java に移植するための助成金 :$2M
- **合計 :$5.5M**

初年度以降の目標と年間コスト

- RustTLS、DNS リゾルバー、NTPd に関する活動とアドボカシ活動の維持：年間 $1M
- Go と Java セキュリティ活動に関する作業とアドボカシ活動の維持：年間 $1M
- **合計 : 年間 $2M**

付録 5

ストリーム 5：OpenSSF に Open Source Security Incident Response Team を設立

問題：

Heartbleed や Log4Shell に続くような、世界を激変させるクリティカルなオープンソース ソフトウェアの脆弱性は、現在私たちのコードベースのどこかにすでに存在しており、発見されていないことはほぼ確実です。これらの脆弱性が、最終的に、友好的な研究者、または悪意のある脅威アクターによって発見された場合、オープンソース メンテナーは、業界全体のサイバーセキュリティ体制に劇的な影響を与える可能性のある迅速な決定を行う必要があります。このことを念頭に置いて、リソース不足のプロジェクトでサイバーセキュリティの緊急事態が発生した場合、オープンソース ソフトウェア メンテナーに、十分に吟味された、可用性の高い、専門的なセキュリティ脆弱性修復への支援を得られる場所がどこにもないことが多いということを知ると、非常に心細くなります。その結果、開発者は、業界全体のセキュリティ リスクに対して突然（時には知らないうちに）間接的に責任を負うことになり、一人で修復を試みる以外に選択の余地がありません。多くの場合、それを安全に実施するためのセキュリティの専門知識や、協調的情報開示のためのコネクションを具えた専門家も持ち合わせていません。

提案アプローチ：

OpenSSF Open Source Security Incident Response Team (OSSSIRT) の創設を提案します。同チームは、業界全体の協力によって作られる専門家グループであり、影響の大きいセキュリティ脆弱性や関連するセキュリティ緊急事態の修復におけるあらゆる側面でオープンソースメンテナーを支援します。研究者との間の情報開示スケジュールの交渉から、ソフトウェア パッチの作成、影響の大きいダウンストリーム プロジェクトへのパッチ適用の調整まで、オープンソース メンテナーは、信頼できる、ベンダー中立な、吟味された、広く調整された、経験豊富なセキュリティ専門家グループを無償で利用できるようになります。メンテナーの雇用主や地理的な場所にこだわることはなく、また OpenSSF や Linux Foundation の信頼性に裏打ちされています。

検討範囲外：

- クローズドソース / プロプライエタリ ソフトウェアの脆弱性に関連するものすべて
- オープンソース ソフトウェアに新しく報告された重大な影響を及ぼす脆弱性にパッチを当てることに対し、喫緊ではないとみなされるオープンソース ソフトウェアのセキュリティ改善
- 別のオープンソース プロジェクトのセキュリティ脆弱性に起因するセキュリティ露出（security exposure）を修正するためにプロジェクトや個々の企業を支援すること

初年度（6 ～ 18 か月）の目標とコスト

目標：

- オープンソース開発者やセキュリティ インシデント対応者、特にオープンソース ソフトウェアの影響の大きい脆弱性の開示と修復を管理した経験を持つ人々と、私たちの提案について議論することで、**問題の全体像を理解し、文書化する**。
- 上記の調査に基づいて、これらの問題を解決するためにメンテナーに提供する**サービスのコアセットを特定する（1 年目）**。これには、次の支援の最小限のサブセットが含まれる可能性が高い。
  - 脆弱性レポートおよび / または概念実証用攻撃実行コードを受け取る際に、メンテナーとセキュリティ研究者の間の安全なコミュニケーションの確保
  - セキュリティ研究者とのコミュニケーションを実施、さらに情報開示スケジュールの交渉
  - メンテナーが脆弱性とその影響を理解することを助けるために、精査されたセキュリティ専門家から専門知識を提供
  - 特定された脆弱性を修正するためのソフトウェア パッチ案の作成（メンテナーから特に要求された場合のみ）
  - 提案された修正を公開前にレビューして、報告された脆弱性を完全に解決しているかどうかの確認
  - 広範な脆弱性悪用の可能性を最小限に抑えるよう、メンテナーがソフトウェア パッチやセキュリティ アドバイザリを作成し、公開するのを支援
  - セキュリティ パッチの導入に対するリスク軽減アプローチを支援するために、ダウンストリームの影響を受けるプロジェクトと機密の情報交換を適宜調整
  - オープンソース プロジェクトにおける新たな脆弱性が悪用されているという報告をメンテナーが受けた場合に、取るべき措置を評価・提案し、メンテナーを支援
- これらのサービスを利用するための**適格基準**を定義する。これは、次のようなものに基づく可能性が高い。
  - オープンソース プロジェクトがどれほどクリティカルであるか
  - 脆弱性の重要度（CVSS スコアまたはその他の指標に基づく）
  - 脆弱性が実際に悪用されている証拠の有無
  - メンテナーがパッチを書く準備ができていると感じているかどうか
  - プロジェクトがすでに同様のリソースを利用できているかどうか
  - 関連する協調的脆弱性開示プロセスの複雑さ
- これらのサービスを提供するために必要な **IT および通信インフラストラクチャ**を選択し、その展開、運用上の可用性、およびセキュリティ保証のための計画を作成する。
- **オープンソース メンテナーを対象としたプレイブック / ガイダンス文書**を増補する。この文書には、サイバーセキュリティ緊急事態（例：重大な脆弱性が報告された）が発生した場合に何をすべきかについて一般的に有用なガイダンスを提供し、どのようにして、またいつサポートを受けるべきかについて明確な説明を提供する。
- **契約への記述が予定されるもの（審査プロセス、および倫理協約を含む）、および各「消防士」/ インシデント対応者に必要とされるスキル / 経験**を定義する。

- 「消防士」/ インシデント対応者の**関与モデル**を設計し、以下のような事柄に対応する。
  - 報酬 / 資金モデル
  - 法務 / 契約の詳細
  - オンコール（呼び出し待機）ローテーションの設定 / 適切なサービス可用性を確保する方法
- **インシデント対応者の最初のグループ**を、最低 2 年間のコミットメントにて募集する。
- 以下のような、本サービスの**運用モデル**を公開する。
  - 時期尚早な情報開示の防止を含む、各「消防士」/ インシデント対応者によって必ず守られるべき約束
  - 本サービスと関わろうとするメンテナー / 開発者向けの「ハウツー」ガイド
  - SLA (Service Level Agreement)
- 専用の情報 Web ページの作成、カンファレンス プレゼンテーション、ウェビナー、「営業時間」中の会合、メディア インタビュー、OSS 開発に関わるリーダーや業界グループとの対話などを通じて、この新しいサービスについて教育するために、**関連する開発者コミュニティに積極的に働きかける**。
- 最大 30 件の緊急事態に対してサービスを開始および提供する。
- 1 年間のこの活動の成功と影響を理解するための主要な**指標を定め、報告する**。

コスト：

- 年中無休（24 時間・週 7 日・年間 365 日）の第一線 / トリアージ / プログラム管理サポート：
  - プロジェクト マネージャー　1 名
  - 呼び出し待機トリアージ / セキュリティ インシデント ハンドラー　3 名
  - スタッフ　(4 人× $300K/ 年) = $1.2 M/ 年
- 法務、IT インフラストラクチャ、Web デザイン、アウトリーチなどのサービス開始コスト：
  - プログラム リード：1 人　$300K/ 年
  - 法務、IT、Web 設計（アウトソーシング / 契約）: $250K
  - アウトリーチ / マーケティング：$250K
- 10 ～ 20 人のセキュリティ専門家を維持するための資金（必要に応じて）
  - 関与モデルはパートナー組織と共同で開発する。初年度は、費用なしのインセンティブを探るが、初年度の目標を達成するためには、さらなる資金が必要と判断される部分が出てくる可能性がある。ここでは、控えめに見積もって、そのようなインセンティブのための準備金として $750K を予算計上する。
- **合計：$2.75M**

初年度以降の年間目標とコスト

目標：

- サービスの利用状況と成果に関して **1 年目を評価**し、2 年目以降に有意義な変更と、意図したサービスレベルを維持するために必要となる可能性のある資金がどれほどなのかを決定する。
- **サービスのコアセット(2 年目以降) を拡大し**、1 年目よりも幅広いセキュリティ緊急対応サポートを提供する。
- **こ**れらの追加サービスに対応する**適切なインシデント対応者を採用**し、これらの目標を達成するための適切な資金を確保する。
- 2 年目以降の成功と影響を理解するための主要な**指標を定め、報告する**。
- 100 件の有効なセキュリティ インシデント処理を提供する。

コスト：

- 1 年目と同様に、年中無休の第一線トリアージ、プロジェクト管理、法務、アドボカシ活動の提供に関連する基本コスト：$2M
- 1 年目と同様に、控えめに算定した参加へのインセンティブのための準備金を予算に計上：$750K
- より多くのセキュリティ インシデント処理を提供するためには、第 2 線のトリアージとして、追加のプログラム管理と技術的な専門知識が必要となる場合がある。このために 1 人の追加技術スタッフを提供 :$300K/ 年
- **合計 :$3.05M**

付録 6

ストリーム 6：メンテナーと専門家により、新しい脆弱性の検出、修復、および協調的開示の加速

問題：

脆弱性の数は劇的に増加しており、ソフトウェア開発速度の増加も一因になっています。NIST によると、2021 年には 22,000 件以上の新規の脆弱性が発見され、CVE (Common Vulnerabilities and Exposures) として報告されています。これは、2016 年に「わずか」約 6,000 件の新規脆弱性が特定されていたころと比較して大きく増加しています。2022 年はすでにその数に匹敵するか、それを上回っています。ソフトウェアが従来型の開発と配布の手法から、より機敏で現代的な手法へと進化してきたように、システムのセキュリティを保ち、脆弱性の修復にコントリビューションする手法も同様に進化できるはずです。平均的な組織では、クリティカルと評価された脆弱性にパッチを当てるのに約 60 日を要するとされる中で、（サイバー防護訓練の防御側である）ブルーチームのディフェンダー（社内要員、および請負の脆弱性評価者）は、開発者に権限を与え、この膨大な脆弱性の波にさらされる時間を減らすために、入手できる限りのあらゆるツールや強みとなるものを必要としています。

提案アプローチ：

OpenSSF は、オープンソース開発者の手になるソフトウェア スキャン、および分析のためのツール（SAST、DAST、ファジング テストなど）の使用を増やすこと、OSS メンテナーと直接的に関わりを持っているセキュリティ専門家を伴う一元化されたマネージドサービスを通じてこれらのツールの使用を提供すること、新たな脅威の原因として出現した脆弱性の重要度分類に照らして手持ちの OSS コードの状態を体系的にスキャンすること、また OSS メンテナーと協力して協調的脆弱性開示の状況を改善することなどによって、この深刻な問題に対処するためのオープンで多面的なアプローチを提案しています。それぞれの詳細は以下のとおりです。

この活動の多くは、OpenSSF の Alpha-Omega プロジェクト、特にこれらの問題に対処するために 2021 年に開始された「Omega」の部分を基盤として、これらの作業を実施することができます。OpenSSF 内、および業界全体の他の関連する取り組みと連携することで、これらの活動はさらに拡大してゆくでしょう。

メンテナーによる高品質なセキュリティ ツールの利用を増やす

メンテナーによるセキュリティ ツールとサービスの利用を促進するため、以下の分野に焦点を当てたキャンペーンを実施することを提案します。

- どんなツールがあるのか、またどうすれば簡単に使用できるかをメンテナーが理解できるようにする。必要に応じて「OpenSSF 推奨」ルールセットを含め、セキュリティ ツール メンテナー自身との調整を行う。
- これらのツールの実行結果をメンテナーが理解するための（内部構造にまで及ぶ）詳細なガイダンスの場を確保し、必要性に応じて開示のプロセスを進める。

プロプライエタリなセキュリティ ツール プロバイダーと協力して、オープンソース プロジェクトに無償の「ワンクリック インストール」を提供するよう説得します。これらのキャンペーンは、まず、Omega を通じて行われる分析作業と並行して、最も重要なオープンソース プロジェクトのトップ 10,000 件をターゲットにし、メンテナー向けに用意されるバッジ取得プログラムや、可能ならツールの有効性を実証するための評価データの収集を含むようにしていきます。

Omega による一元的な脆弱性検出

前述の Omega プロジェクトですでに開始されている作業を基に、最先端のセキュリティ ツール（静的分析やファジング テストなど）を使用して、最もクリティカルな OSS プロジェクト 10,000 件を分析し、その結果をトリアージ（分類）し、エラー率を最小限にするような検出ロジックを作成 / 改良し、OSS メンテナーと協力し、協調的脆弱性開示を通じて修正適用を進めることを提案します。

脆弱性の重要度分類削減キャンペーン

脆弱性が発見された場合は、それらが多くのプロジェクトに存在するクリティカルな脆弱性として分類される例であるかどうかも確認する必要があります。Omega の一部としてすでに行われているセキュリティ分析と組み合わせて、OpenSSF は、すべてのオープンソース プロジェクトにわたって、特定の脆弱性重要度分類を有意義に削減または削除するためのキャンペーンを実施します。この作業には、ツールの検証、アドボカシ活動（問題とその簡単な検出方法を記述したブログ / 記事）、Omega による一元化された自動検出などが含まれます。このようなセキュリティ キャンペーンの例には、次のようなものがあります。

- Log4Shell インシデントにあるような JNDI インジェクション（検出可能であり、エコシステム内の稀少性のため対応可能））
- コマンド /SQL インジェクション（その重要性と広播性のため）
- デシリアライゼーション（その重要性と限られた認知度のため））

セキュリティ ツールの品質向上

ツール メンテナーに実践的なフィードバックを定期的に提供したり、オープンソース セキュリティ ツールの改善にコントリビューション（コアエンジン、ルールの提供）したり、Omega プロジェクトの結果を分析して、ツールの品質とギャップを洞察したりすることで、セキュリティ ツールの精度を向上させることを提案します。また、研究をマネージし、ツールを構築し、ルールを作成し、さらには有効性と利用に関してより大きなエコシステムと協力しながら、最新技術とのギャップを特定し、それらを目標にして進めます。

Coordinated Vulnerability Disclosure (CVD：協調的脆弱性開示) の成熟

脆弱性の発見と修正は、問題に対処するための重要な最初のステップです。いかにダウンストリームのコンシューマーや、依存プロジェクトのエコシステムに通知し、迅速かつ効率的にそれらのグループと修正のリリースを調整するかが最後の 1 マイルの鍵です。OpenSSF は、脆弱性ライフサイクルのこの段階を改善するために、以下のアクションのいくつかに積極的に取り組み、推進しています。

- OpenSSF Best Practices WG と協力して、オープンソース開発者に優れた CVD プラクティスについて教育。たとえば、「Guide to coordinated vulnerability disclosure for open source software projects」など。
- セキュリティ研究者を教育。特に、オープンソース メンテナーに報告し、オープンソース メンテナーと関わる効果的な方法について情報提供。「OSSF Guide to coordinated vulnerability disclosure for security researchers engaging with open source software projects」は近日公開予定。
- 悪意のあるハイジャック、インジェクション、サボタージュなどの脆弱性を、CVE などのツールを使用して適切に文書化、影響を受けるコンポーネントの自動識別などを含めて、アドバイザリ上での明確な伝達を確実に実施。
- CERT-CC の VINCE ツールなどの CVD ツールの使用を提唱。このツールは、研究者が脆弱性を報告し、ソフトウェア メンテナーが脆弱性情報を交換するための中立で安全な場を提供。

初年度（6 ～ 18 か月）の目標とコスト

- メンテナーによる高品質なセキュリティ ツール使用の増加
  - 「growth campaign（利用促進キャンペーン）」を実施して、より多くのメンテナーが高品質のセキュリティツールを活用できるようにし、ツールの出力結果を処理するための直接的かつ詳細なガイダンスを提供：8 人のスタッフ（技術 / アドボカシ活動）: $2M
- Omega による一元的な脆弱性検出
  - Omega への投資を拡大し、上位 10,000 の OSS プロジェクトを完全にカバー
  - クラウドスケール分析など（10 人× 300K）+ インフラストラクチャに $2M = $5M
- 脆弱性重要度分類削減キャンペーン
  - 3 つの脆弱性重要度分類削減キャンペーン（6 か月に 1 回）を実施し、OpenSSF Developer Best Practices WG に学習事項をフィードバックする：$3M
- セキュリティ ツールの品質向上
  - 一般的に使用されているツールの品質を改善するための投資（助成および契約による）：$2M
- 協調的脆弱性情報開示（CVD）の成熟
  - 技術業務とコンテンツ開発を担当するスタッフ 8 名：$3M
- **合計：$15M**

初年度以降の年間目標とコスト

- メンテナーによる高品質なセキュリティ ツールの使用拡大
  - メンテナーが高品質のセキュリティ ツールを使用することを引き続き推奨：年間 $1M
- Omega による一元的な脆弱性検出
  - トップ 30,000 件の OSS プロジェクトを継続的にカバーするよう Omega を拡張：年間 $6M
- 脆弱性重要度分類削減キャンペーン
  - 脆弱性重要度削減キャンペーンを年 4 回実施：年間 $2M
- セキュリティ ツールの品質向上
  - 一般的に使用されるツールの品質を理解 / 改善するための投資：年間 $1M
- 協調的脆弱性開示（CVD）の成熟：年間 $1M
- **合計：$11M/ 年**

付録 7

ストリーム 7：1 年に 1 度、最大 200 の最もクリティカルな OSS コンポーネントのサードパーティ コード レビュー（および必要な修復作業）の実施

問題：

セキュアなソフトウェアを作成するのは難しい仕事です。実際に、世界中で最も高く評価されているテクノロジー企業やオープンソース プロジェクトでさえも、作成したコードにセキュリティ上の欠陥があることが後で判明し、修復のためにソフトウェアのアップデートを必要とすることがあるほどです。これらの欠陥の数は、コードが実世界（「本番環境」）に投入される前に劇的に減らすことができますが、これらの脆弱性を検出するために使用される多くの技術ツールでも、最もクリティカルで複雑なバグの、いくつかは検出することができません。これらのバグは、今でも人間の専門家の領域に在ります。専門家のセキュリティ コード レビューによって慎重にレビューされていないコードには、一般的にセキュリティ上の欠陥が含まれており、脅威アクターによって発見され悪用されると、企業や国家のセキュリティに重大な損害を与える可能性があります。

オープンソース セキュリティとサードパーティ コード レビューに関する先行調査では、次の 2 つのキーポイントが示唆されています。(1) 多くのオープンソース ソフトウェア プロジェクトは、サードパーティ コード レビューをほとんど、あるいはまったく受けていない（「Threats, Risks, and Mitigations in the Open Source Ecosystem」を参照）、(2) いくつものレイヤーを掘り下げて脆弱性を発見するには、より詳細な監査、ロジック レビュー、ソースコード分析が必要になることが多い（「Zero Days: Thousands of Nights」を参照）。これには、専用の資金、プロジェクト管理、および計画と実行のためのノウハウが必要です。

提案アプローチ：

クリティカルな OSS ソフトウェアのサードパーティ コード レビューおよび関連する修復作業の管理および実行のために、OpenSSF が主導する業界全体の協調的な取り組みを提案します。信頼できるサードパーティセキュリティ企業に重要なオープンソース プロジェクトのコード レビュー / セキュリティ監査を実施してもらい、監査結果の公開レポートを発行することで、次のことが可能になります。

1. クリティカルなオープンソース ソフトウェア コンポーネントの、まだ検出されていない、影響の大きい脆弱性を、脅威アクターによって発見され悪用される前に検出し、修正。
2. 監査の範囲と関連するセキュリティ調査結果の透明性ある報告を通じて、クリティカルなソフトウェア プロジェクトに対する開発者、業界、公共部門の信頼を向上。
3. エコシステム全体に利益をもたらすために、最もクリティカルなオープンソース プロジェクトに優先順位を付け、高い監査基準を設定し、検出された脆弱性を修正し、関連するすべての調査結果を公開するといった協調的アプローチを通じて、リソースを集中させ、費やされたすべての資金を効果化。

タイムライン：協調的活動により、最初の 1 年以内に、最も重要なコンポーネント 50 件のサードパーティ コード レビューに取り組み、その結果を提供し、3 年目の終わりまでに 100 ～ 200 件の年次監査をカバーするように拡張します。

コストの議論：セキュリティ監査と関連する修復作業のコストは、プロジェクトの規模や複雑さによって異なります。筆者たちの経験を総合すると、ある程度の大きさのオープンソース プロジェクトの監査と修復作業は、1 回あたり $50K から $250K の範囲になります。監査 1 回あたりの平均コストを $200K と想定し、経験を積みながら時間をかけて補正していきます。現在のキャパシティに基づくと、最初の 1 年間に 50 件の監査を実施でき、その後、監査環境への投資により年間 100 ～ 200 件の監査を実施できると予測しています。プログラムの管理、プロジェクトの公正な選択、レビュー担当者のパフォーマンス管理、結果の整理、影響の測定などのためのオーバーヘッドは、50 件の監査を処理する最初の 1 年間にさらに $1M 、その後は規模拡大により年間 $2M になります。

初年度（6 ～ 18 か月）の目標とコスト

- 公開するセキュリティ監査の目標（および目標としないもの）、および OSS のセキュリティ監査のベストプラクティスについて説明した**論説文書、ブログ投稿、または同様のものを公表。**
- 本プログラムの趣旨を説明する**ブログ投稿**を公開し、プログラムの範囲、プロジェクトの選択基準 / プロセス、ベンダーの選択基準 / プロセス、スケジュール、プログラム参加者の期待、その他の詳細を説明。
- サードパーティの**セキュリティ監査公開報告書**の要件とフォーマットを定め、監査 / ベンダー間の一貫性を確保し、すべての報告書に、私たちが求める価値があり、実行可能で、公開可能な情報が含まれることを確実にする。
- このワーク ストリームに対する**プログラム マネージメント**のニーズを見極めて、監査プログラム全般を管理するためにスタッフを雇用し、またはベンダーと契約する。
- **オープンソース プロジェクトの選択基準とプロセス**を定義し、資金提供を受けて実施されるサードパーティ セキュリティ監査の受益者となるプロジェクトの優先順位付けと選択方法を定める。これには、このプログラム全体を通じてプロジェクト メンテナーやコア開発者とどのように協力するか、プロジェクトが自己推薦できるかどうか（および、どのように）を含める。
  - これを広範なコミュニティに発表
  - 1 年目に監査するプロジェクトとして、最大 50 プロジェクトを選択し、発表
- **セキュリティ ベンダーの選択基準とプロセス**を定義し、セキュリティベンダーが適格であるための要件、選択の条件、およびベンダーがプログラムの 1 年目の活動に入札するためのプロセスとスケジュールを明確に規定。
  - これをより広範なコミュニティに発表し、ベンダーに RFP/ 入札プロセスへの参加を呼びかけ
  - このプログラムのすべてのプロジェクト監査用に MSA (Master Service Agreement) テンプレートを作成
  - 1 年目に最大 50 件のプロジェクト監査を実施するベンダーを選定し、発表

- 1 年目として、**最大 50 件のクリティカルなオープンソース プロジェクトのサードパーティ セキュリティ監査 / コード レビュー**を実施し、公開報告書を提供する。
  - これらの報告書の最初のバージョンは、特にプロジェクトのメンテナー、および検出された脆弱性の修復に関与する人に向けて提供が必要。
  - これらの報告書の最終バージョンは、検出された脆弱性の修正後、openssf.org または openSSF の github リポジトリで公開。
- 監査プログラム中に**検出された脆弱性**を、事前に指定された時間枠内（たとえば、最初のレポートを受け取ってから 90 日以内）に修正し、関連する技術アドバイザリを発行。
- 1 年目の終わりに、このプログラムに関連する**主要指標**に関する報告書を発行。
- 1 年目に学んだ教訓に基づいて、OSS のサードパーティ セキュリティ監査プログラムの **2 年目以降の計画**を決定。
- 50 件の監査を実施、平均 $200K = $10M
- 管理スタッフ、プログラム管理、法務、IT、アドボカシ活動：$1M
- **合計：$11M**

初年度以降の目標と年間コスト

- 2 年目以降：年間最大 200 件のオープンソース プロジェクトの監査を実施。
  - リスクの深刻さに基づき 1 年、3 年、または 5 年の間隔で、最もクリティカルなコンポーネントを再度監査。
  - メジャー リリースや新規コードを含むプロジェクトは、変更監査を要求できる。
- サードパーティ セキュリティ監査プログラムを通じて発見されたすべての脆弱性を修正する。
- このプログラムに関連する主要指標に関する年次報告。
- 毎年 200 件の監査を実施。これも年間平均コスト $200K = $40M
- 管理スタッフ、プログラム管理、法務、IT、アドボカシ活動：$1M
- **合計 : 年間 $42M**

付録 8

ストリーム8：最もクリティカルなOSSコンポーネントを決定する調査の改善のために、業界全体で広くデータを共有

問題：

オープンソース ソフトウェアのセキュリティを向上させるための私たちの取り組みの根底として、広く使用されているにもかかわらず、十分に保守されていないプロジェクトを知ることに対する大きなニーズがあります。どのオープンソース ソフトウェアが実際に「クリティカル（非常に重要）」であるかを客観的に判断する上で、大きな課題は、使用状況、ダウンロード数、依存関係などのデータが、そのデータを収集することのできるソフトウェア ディストリビューション チャネルにより、彼らの独占的な利益だと見なされることが多いことです。さらに、この生データの一部は誤解を招く可能性もあります。たとえあるモジュールが頻繁にダウンロードされたからといって、必ずしもその重要度と相関しているわけではありません。（ディストリビューション チャネルが保持する）機密情報を共有することに対する彼らの懸念に対処できれば、クリティカルなオープンソース ソフトウェアの、より良いリストを得ることができるかもしれません。さらに、（libraries.io や deps.dev と同様に）依存関係、ライセンスなどの調査研究を目的に、クロスエコシステムの現況について公開データセットを取得するためのより簡単なメカニズムを提供する方法を開発することには多くの利点もあります。

提案アプローチ：

この課題は、単独の組織で取り組むことはできません。この提案では、学術研究者や企業研究者がアクセスするために、中立的な場所に、匿名化データをプールする意思のある少数の組織・ベンダーと多面的な枠組みと合意を作ります。ただし、この情報は、非常に広範に利用されているが保守が不十分なオープンソース ソフトウェアを特定する問題に対処するためにのみ使用できるという厳格な条件の下で使用します。このデータにアクセスできる個人は、一連の使用条件と処理要件に同意する必要があります。

参加組織の数は、ガバナンス モデルを単純に保つために控えめ（メンバー数 10 ～ 15 以下）にすべきですが、さらに拡大する方法も定めます。想定メンバーには、以下が含まれるでしょう。

- The Linux Foundation
- Linux オペレーティング システム ベンダー上位 3 ～ 4 社
- クラウド サービス プロバイダー大手 5 ～ 7 社
- ソフトウェア構成分析ベンダー、最大 3 ～ 4 社

このグループの作業成果には、これらのリーダーが提供する最良の情報が含まれます。ただし、情報源およびデータに含まれる個人情報（PII: Personally Identifiable Information）は、匿名性を維持するために（差分プライバシー手法を通じて）削除、またはあいまいにされます。

法的枠組みが確立されれば、最も保守されていないが、最も使用されているプロジェクトを多数の情報源から総合的に理解するために、このグループは Harvard Census II データベースなどの情報源から開始し、さらに独自の情報源を追加していきます。

技術概要

詳細な技術アーキテクチャは、各業界パートナー / ベンダーが彼らのプラットフォームにデータを公開する際に使用されているデータセットと作成者（ソフトウェア）の形式の分析が行われた後に作成されます。

The Linux Foundation の技術チームがホストする Data Lake House (DLH) は、さまざまな関係者のためにデータを保存、処理、および照会する記録保存用のシステムとして使用されています。最初、データはその作成者のサポートする既存のモデルに取り込まれます。 データ フィードの統一モデルへの変換は、DLH の受信向けに構築されたコネクター フレームワークを介して行われます。

データ作成者には、まずサポートされているフォーマット（JSON や CSV など）でデータ セットをアップロードするためのシングル ページ アプリケーション（SPA）が提供されます。さらに、アップロードのために、抽象化されたクラウド ストレージの HTTP エンドポイントも公開します。

インフラストラクチャとしては、非構造化データ、または半構造化データの取り込みと検索をサポートするのに、月に 1 ～ 5 TB が必要であると予想しています。分析とデータ処理はクラウド リソース外で行われる可能性が高いと想定していますが、レート制限つきクエリと BI ツールのサポートが役に立つでしょう。100% のアップタイムは必要ありませんが、米国のビジネス時間内では迅速なターンアラウンド オペレーションのサポートが望ましいでしょう。

最初のリリース（MVP: Minimum Visual Product）では、ビューは軽量な BI インターフェイス（非プログラマー用）、および SQL クエリ文字列（プログラミング バックグラウンド用）のようなコンシューマーを介して提供されます。ダッシュボード、ポータル、UI などは MVP の範囲に含まれません。MVP ソリューションの主要なコンシューマーは研究者になる予定です。

将来的には、業界パートナーがデータへのアクセスを希望する場合、パートナー間でプライバシーデータが共有されないよう、厳格なロール ベースのアクセス制御（RBAC）と適切なデータ匿名化およびパーティショニングを使用してマルチ テナント ダッシュボードやポータルを開発する可能性があります。将来のリリースには、データ保護とプライバシーについての同様の原則に従いながらデータをクエリするための送信 API が含まれる可能性があります。

初年度（6 ～ 18 か月）の目標とコスト

- コラボレーションの開始
- すべての当事者が合意できる法的枠組みの明確化
- クリティカルさの程度と使用状況のデータはパートナーによって共有
- 情報源をまとめた一覧表の最初のドラフトの公開
- データは研究者が利用できるようにし、情報提供者を含むコラボレーション参加者によって決定されたサブセットまたは集約データを一般公開。

- コスト、スタッフ：5 人 x 300K = $1.5M　(*FTE: Full Time Equivalent：フルタイム当量)
  - クラウド運用：1 人 (FTE)
  - データ収集（インジェスチョン）、コネクター、モデリング、データサイエンスのニーズに対応する開発者：3 人 (FTE)
  - 研究者とデータ情報源との関係構築：1 人 (FTE)
- 運用およびサードパーティ ソフトウェア：$250K
  - 管理および法務：$220K
  - BI ツール アクセス（研究者は必要に応じて）：$30K
- Linux Foundation の試算による Compute & Storage コスト：$100K/ 年

データサイズ /月	インジェスチョン &ストレージ /月	コンピュート /月	可視化 /月	月間	年間
5 TB	**$2,827**	**$1,560**	**$2,800**	**$7,187**	**$86,000**

- **合計：$1.85M**

初年度以降の年間目標とコスト

- スタッフ要件 / 役割：
  - 会議の開催、議事ノートの作成と配布、メンバー候補へのアウトリーチなどを含む運営支援。
  - プログラミングおよび技術コンサルティング：コラボレーション参加グループから取り込むデータの明確化、データの匿名化、および統合レポートの作成。
  - コラボレーション体制の構成、および課題解決に関する法務担当者からのコンサルテーション。
- スタッフのコスト : 上記と同じ 5 人、$1.5M
- 管理、法務、およびサードパーティのコスト：$250K
- 運用コスト、データサイズ、利用率、需要の増加を想定：$300K
- **合計：$2.05M/ 年**

付録 9

ストリーム 9:あらゆるところに SBOM をー SBOM のツールとトレーニングを改善し、採用を促進

問題:

新たに重大な脆弱性が発見された時、企業は多くの場合、脆弱性の影響があるのかどうか、どのように危険なのか、どこにあるのかを判断するために奔走します。導入したソフトウェア資産のインベントリがないことや、入手したソフトウェア内に組み込まれたコンポーネントに関するデータがないことがあまりにも多くあります。また、組織はソフトウェアの入手を検討しますが、ソフトウェア コンポーネントに既知の脆弱性が含まれているリスクを判断する手段がないことがよくあります。多くの企業は、SBOM (Software Bill of Materials) がこれを解決するための基本的な構成要素であると認識しています。しかし、この課題への対応に適切に用いるためには、その採用が広く行われ、標準化され、可能な限り正確である必要があります。

SBOM は、ソフトウェア システム内のソフトウェア コンポーネントのリストです。通常は、開発者や組織がリスク マネジメントのユースケース（活用形態、たとえば、脆弱性分析など）を効果的かつ効率的に評価するためのインベントリ リストとして維持・管理されます。ソフトウェアの保守・サポート、インシデント調査、（アプリケーションの）実行中保護など、新たな SBOM ユースケースが増えています。

しかし、SBOM はまだソフトウェア業界で広く作成も利用もされていません。

提案アプローチ:

あらゆる場所で SBOM を使用可能にすることで、オープンソース エコシステム全体（プロデューサー、コンシューマー、メンテナー）のセキュリティ体制を向上させることができます。SBOM を作成するだけでは不十分であり、積極的に活用する必要があります。

さらなる SBOM の利用に対する頑強な障壁を取り除くには、以下を実行する必要があります。

1. SBOM のユースケースを構築するための要件が明確に理解され、現行の SBOM 仕様に記述され、実装されること
2. これらの要件を満たした SBOM を生成する「摩擦のない」オープンソースツールが存在すること
3. だれでも容易に利用可能な教育、啓発・実装のための指導書、および第三者による支援があること

このストリームは、上記の 3 つのポイントすべてに対応し、SPDX チームと直接協力しながら新しい SPDX セキュリティ プロファイルとして実装します。

セキュリティ ユースケースを可能にするための SBOM 要件

私たちは、SPDX や CycloneDX など、いくつかの SBOM 仕様（配布フォーマット）があることを認識しています。単一仕様への統一は有益であり、目標でもありますが、現時点では私たちの焦点ではありません。

すべての SBOM 仕様に実装さる共通要件に関する合意は、ツール間の相互運用性と統合性を向上させ、その結果、既存のソリューションが改善され、新しいソリューションの開発、実装および保守が容易になります。ここで説明する活動には、可能なかぎり仕様間の一貫性を確保するために、他の SBOM 仕様コミュニティの代表者が参加を予定しています。目標は、重要なケースについて、異なるフォーマット間の**シームレスな相互運用性**を可能にすることです。このイニシアチブには、規制当局、企業・組織の最高情報セキュリティ責任者（CISO）、セキュリティ エンジニア、セキュリティ ツール開発者、およびその他の関連するステークホルダーを含む一連のアドバイザーとのコラボレーションが含まれます。

セキュリティ ユースケースを可能にする SBOM ツール

SBOM が遍在的であればあるほど、業界にとって価値が高くなります。一般的に、SBOM は、ソフトウェアがビルドまたは配布されるたびに、十分に検証されたツールを通じて自動的に生成されるべきだと考えています。たとえば、Java 用の Apache Maven や Javascript 用の npm などの一般的なパッケージ管理プラットフォームは、そのプラットフォームによって配布されるすべてのパッケージに対して SBOM が利用可能であることを保証します。同様に、ディストリビューション パッケージやコード リポジトリなどの上位レベルのパッケージが作成される時にも、SBOM はそのプロセスの一部としてデフォルトで生成されるべきです。

SBOM がソフトウェア ディストリビューションで遍く存在するようにするためには、採用のためのあらゆる摩擦を取り除く必要があります。この問題に対処するには、誰でも簡単に SBOM を作成して活用できるオープン ツールを提供する必要があります。次に、プロデューサー、コンシューマー、メンテナーに対して、そのようなツールを採用するようインセンティブを与える必要があります。これらのインセンティブには、手順やツール動作のデフォルトの変更やバッジ付与などが含まれます。

次のことを行う必要があります（皮切りとして、すべて、セキュリティ ユースケースのサポートに重点を置いています）。

1. 既存のオープンソース ツールにコントリビュートする
2. 必要に応じて新しいオープンソース ツールを構築する
3. 開発者ツールのエコシステムと協力して、SBOM を普及推進する

私たちは SBOM 生成ツールに関して実用的なアプローチを採り、開発者への摩擦と技術的な制約があることが認識されているものの、リリースが最も早く、経済的な有効性が確かなツール セットから始めます。これらの取り組みの後、私たちは、リリースに時間がかかり、より多くの開発努力を必要とするが、大多数の開発者の摩擦を可能な限り少なくするツールに焦点を当てます。

最初の実装作業は次のとおりです。

- インベントリと脆弱性分析のユースケースに焦点を当て、一連の文書化された要件とマシンリーダブルなデータ スキーマを提供する。そのような仕様に準拠した SBOM 生成ツール（下記参照）が実装された場合、インターネット全体のリスクを軽減し、オープンソースのセキュリティ体制を改善することが期待される。
- SBOM の生成と検証だけでなく、複数の SBOM フォーマット間のロスレス変換、さらには SBOM と SBOM のマージやバージョン管理などの関連操作を可能にするツールを作る。
- SPDX チームと協力して、これらの要件を今後公開予定の SPDX 3.0 仕様にセキュリティ プロファイルとして実装する。

レベル 1 ―クライアントと SDK：オペレーティング システムとビルド システムに依存しないコマンドライン インタープリター (CLI) で、ソースを処理して出力アーティファクトを作成し、またオペレーティング システムやその他の依存関係を処理することができる。すべてのユースケースに対応する必要データを含め、仕様に準拠した SBOM を出力する。これらのツールは、エンドツーエンドの CI/CD ワークフローの一部として、手動または自動（スクリプトなど）で実行される必要がある。これらのツールには、たとえば追加のパッケージ マネージャーをサポートするなど、ベース ツールをカスタマイズし、また拡張するために開発者が使用できる SDK が含まれる。

レベル 2―パッケージ マネージャー プラグイン：Maven、npm、PyPI などの主要なパッケージ マネージャーおよびリポジトリとネイティブで連動するプラグインやモジュールのセット。これらのツールは、通常、後続のビルドごとに実行するために 1 行の設定変更を追加する必要があり、それによって仕様に準拠した SBOM を出力する。この作業により、既存の最高のオープンソース プラグインが、使用中の環境で強化されます。

レベル 3 ―ネイティブ パッケージ マネージャーの統合：主要なパッケージ マネージャーにネイティブで SBOM 生成機能を追加することにより、すべての開発者とすべてのビルド システムは、通常のワーク フローの一部としてデフォルトで自動的に SBOM を生成するようになる。SBOM の生成は、ツールが舞台裏でソフトウェア ビルドのログ エントリをログ ファイルに生成するのと同じくらい一般的でシームレスなものになる。

レベル 4 ―コンテナリゼーションの統合：コンテナリゼーション ビルド プロセスにネイティブな SBOM 生成機能を追加することで、システムは、組み込まれたパッケージによって提供される SBOM コンテンツと、コンテナ ビルド中に追加されたアーティファクトを使用して、コンテナを構成するすべてのコンポーネントを指定した SBOM を出力する。

レベル 5 ―アプリケーション、ソリューションの統合 / デプロイメント：複数の異なるコンポーネント（コンテナ、マシン イメージ、イベント駆動型サービス）で構成されるアプリケーションをデプロイする場合、コーディネーション マネージャは、デプロイされるすべてのアーティファクトを反映するために、構成要素の SBOM を集約する必要がある。

バッジ システム

SPDX セキュリティ プロファイルに準拠した SBOM が、脆弱性ステータスなどのユースケースをすばやく識別できるようにバッジを表示するバッジ システムです。

関連ツールは、おもに既存のツールを開発・保守するチームに助成金を提供することによって、および専任のコントラクターを採用して新しいツールを開発・保守することによって開発されます。

アドボカシ活動

SBOM はソフトウェア業界のほとんどの人にとって、新しいものです。私たちは、管理職層、ソリューション（ユースケース）開発者、セキュリティ エンジニア、監査とコンプライアンスの専門家、開発者ツール チーム、オープンソース プロジェクトなどのオーディエンスを対象とした幅広い教育資料と啓発資料を作成します。

資料には、ガイダンス、チュートリアル、ビデオ、ウェビナーなどが含まれます。これらはすべて、ユーザー エクスペリエンスに重点を置いた Web サイトでホストされます。

初年度（12 ～ 18 か月）の目標とコスト

1. 広く合意され、公開された一連の要件と、インベントリおよび脆弱性分析のユースケースをサポートするデータ スキーム。この第一版は、遅くとも 8 月までに公表されるべきである。
2. 要件とデータ スキームは、SPDX 3.0 セキュリティ プロファイルとして実装される。 ここでのスケジュールは SPDX コミュニティに依存する。
3. . 以下のような SPDX セキュリティ プロファイル SBOM を生成する、自由に広く利用でき、高品質で、保守およびサポートの得られるオープンソース ツール。
    a. . コマンドライン インタープリターおよび SDK（cli）
    b. Maven (Java)、npm (JavaScript)、PyPi (Python)、GoModules (GoLang)、Ruby (rubygems)、NuGet (.NET)、Composer (PHP)、Rust (Cargo)、RPM、APT、dpkg、DEB (C/C++) など、広く採用されているプラットフォーム用のパッケージ マネージャー プラグイン
4. セキュリティ プロフェッショナルのための SBOM ポータル。教育および啓発資料をホストし、ツール ダウンロードのハブとして機能し、「SBOM のすべて」に関する中核ニュース サイトとして機能させる。教育、および啓発コンテンツは Linux Foundation 認定プログラムに統合する。

コスト：

セキュリティ ユースケースを可能にするための SBOM 要件 — この活動は、おもに Linux Foundation の従業員とコントラクターの支援を受けたボランティアによって行われます。

▸ セキュリティ ユースケースを実現するためのツール

- SBOM 生成ツール：(10 エコシステム×各半年間× $300K/ 年) = $1.5M
    - CLI および SDK
    - パッケージ マネージャー プラグイン
- Developer Relations (DevRel) による普及促進 (4 人× $300 K/ 年) = $1.2M
- OSV.dev と共同で実施する脆弱性分析のリファレンス実装

▸ 教育と啓発：～ $500K

- SBOM ポータル
- コンテンツ開発

- アドボカシ活動

▸ **合計：$3.2M**

初年度以降の年間目標とコスト

1. 世界で最も一般的なビルドおよびパッケージ管理のソリューション（Maven、npm、PyPi、GoLang、.NET、Rubygems、Composer、および C/C++/Assembler）におけるネイティブ SPDX セキュリティ プロファイル生成。
2. 1 年目に開発されたツールと、これらのユースケースの標準規格化をサポートする活動との連携。
3. 作成されたツールのサポートと保守。
4. SPDX セキュリティ プロファイルのバッジ システム。

活動の初期段階を超えて必要とされるリソースの正確な見積もりは、初期段階で発生する多くの要因に左右されるため、そのようなコストのガイドを提供することは（たとえ大まかなものでも）、現時点では適切ではありません。

付録 10

ストリーム 10：より優れたサプライチェーン セキュリティ ツールとベストプラクティスを 使用して、最もクリティカルな 10 の OSS ビルド システム、パッケージ マネージャー、 およびディストリビューション システムを強化

問題：

オープンソース ソフトウェアは、さまざまな開発言語固有のビルド システムを使用して構築された後、パッケージ マネージャーを介してエンドユーザーに配布されます。このために、異なったエコシステムのコンポーネントがユーザー本番環境に集まることになります。これは、ソフトウェア サプライチェーンに、大きく異なったレベルの品質とリスクのものが一緒に混在することを意味し、リスクの管理・削減に関する統一されたポリシーを適用する取り組みを非常に困難にします。

提案アプローチ：

このストリームの目的は、これらのソフトウェア アーティファクトの配布に対して、最も影響力を持つセキュリティ強化策を調査し、パッケージ マネージャー レベルで改善を推進して、コンポーネント レベルのセキュリティとエコシステムのリスクに焦点を当てた他のストリームの活動を補完することです。

期待される成果は、パッケージ マネージャー システムの、より高いレベルの可視性とセキュリティであり、オープンソース エコシステムの重要なレバレッジ ポイントで改善を見出し、オープンソースのコンシューマーがオープンソース ソフトウェアの構成物とその来歴に対してより高い信頼を得ることができるようになることです。

長期的には、構成物と来歴のデータに関するパッケージングと配布の改善が、より迅速な脆弱性の検出と修復によるパッチ時間の短縮、ダウンストリーム ユーザーへの脆弱性とパッチの情報の透明性の向上、さらにはすべての開発者のためのセキュリティ ツールの改善を助けるはずです。

次の 3 つの活動が提案されています。

- **ライン 1：エンドツーエンドのパッケージ管理：** パッケージ マネージャーに焦点を当てる。どのような基本的セキュリティ機能が必要か、たとえばビル ドシステムからの取り込み、ダウンストリームにおける検証に利用可能なデータ、開発者向けのパッチ通知とリコールのためのツール、悪意のあるパッケージの確実な検出と削除など。パッケージ マネージャーを列挙し、必要なセキュリティ機能に照らして評価する。その後、それらを所有する組織に能力ギャップを埋めるよう要請し、非営利またはコミュニティ所有の場合には資金を提供する。
- **ライン 2：パッケージの構成物：** ソフトウェア構成の可視性の向上に焦点を当てる。 オープンソースの複雑な依存関係の連鎖のために、1 つのパッケージは、多くの開発言語やパッケージ形式にまたがって依存していることが多くあり、既知の脆弱性を迅速に検出し、修正するには、深い部分を含め完全に可視化することが重要であると認識したもの。

- **ライン 3：パッケージ マネージャーにおける共通的セキュリティ インテグリティ レベルの実現を容易にする：**SLSA (Supply chain Levels for Software Artifacts) の進化を推進し、共通のセキュリティ コントロール、一貫性のある属性、標準化された用語とツールについてのコンセンサスを構築する。クリティカルなオープンソース プロジェクト（ストリーム 8 で特定されている）の、ビルド システム、パッケージ マネージャー、ディストリビューション システムについて、SLSA の採用をサポートする。これらの重要なプロジェクトを改善した経験に基づいて、パッケージ マネージャーとリポジトリが SLSA をより簡単にサポートできるように改善を行い、それによってより広範な採用を推進する。

初年度（12 ～ 18 か月）の目標とコスト

- 最も影響の大きいビルド システムとパッケージ マネージャーを特定する：エコシステムのギャップを特定するために現存するセキュリティ ケイパビリティの目録作りを行う。
- ビルド システムに必要な改善、ビルド システムからパッケージ マネージャーへ引き渡されるもの、およびパッケージ マネージャーが UI/API を介して提供する情報を見極めるために、セキュリティ ケイパビリティの綿密な計画を作成し、コメントと実装のために公開する。
- 相互運用性とより広範な採用を確かなものとするために、特定されたパッケージ マネージャー / リポジトリ / レジストリに共通のセキュリティ ケイパビリティと標準を明らかにし、それらを実装する。
- 10 件のパッケージ マネージャー / リポジトリ / レジストリについて、この共通のセキュリティ ケイパビリティ セットの実装を開始する。

コスト：

- 全体管理（コアチーム）の人員数の見積もり：(合計 7 人 × 年間 $300K) = $2.1M
  - プログラム マネージャー 1 名
  - セキュリティ アーキテクト 1 名
  - プロジェクト テクニカル リード（3 つの作業ラインごとに 1 人ずつ）3 名
  - ライン 3 の開発者 2 名
- エコシステムあたりの人員数推定（エコシステム チーム）：
  (10 リポジトリ × 2 人 / リポジトリ × $300K/ 年)：$6M
  - 2 人の開発者が、ライン 1 とライン 2 を合わせて対処：
    - 各環境におけるこれらの標準 / 機能の設計と実装テスト
    - マルウェア検出および修復の実装
    - より多くの情報を伝達するためのビルド システム移行 / 再設定
    - 上記のすべてを検証
    - 注：直接 / 間接込みのコスト (20 人の開発者、年間 $300K) = $6M ですが、ここでは、献身的で資格のあるボランティアのフルタイム要員が支援組織から提供されれば、大幅なコスト削減の可能性がある。
- **合計：$8.1M**

初年度以降の年間目標とコスト

- 1 年目の目標のほとんどは、完了するのに 2 年目を必要とする活動や目標となっている。人員 7 名のコアチームとエコシステム チーム（それぞれ 2 件、10 件のエコシステム）の両方を継続すべきであり、2 年目のコストは $8.1M になる。
- しかし、そのようなコストは、2 年後には、目標が達成されたり、支援が広く行われたり、既存のステークホルダーの投資によって活動が引き受けられたりすることで、大幅に減少する可能性がある。
- 2 年目に検討される可能性のある追加目標：
    - 企業が所有するビルド システムおよびパッケージ マネージャーと協力して、標準化された情報共有とセキュリティ機能を実装する。
    - 標準化された情報共有を実装するために、非営利およびコミュニティ所有のビルド システムやパッケージ マネージャーに資金を提供する。
- **総額 :$8.1M**（2 年目は未定、3 年目以降は少なくなる可能性が高い）

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSFは、業界の壁を越えたコラボレーションです。世界中のリーダーを結集し、より広範なコミュニティ、的を絞ったイニシアチブ、およびベストプラクティスを構築することにより、オープンソース ソフトウェア（OSS）のセキュリティを強化しています。OpenSSFは、オープンソース セキュリティ活動を1つのファウンデーションのもとに結集します。詳細については、https://openssf.org をご覧ください。

THE LINUX FOUNDATION

The Linux Foundationは、オープンソースがクローズド プラットフォームと優位に競うために必要な統一されたリソースやサービスを提供することにより、Linuxの普及・保護・標準化を行っています。The Linux Foundationや他のイニシアチブの詳細については、www.linuxfoundation.orgを参照してください。