OpenSSF
OPEN SOURCE SECURITY FOUNDATION

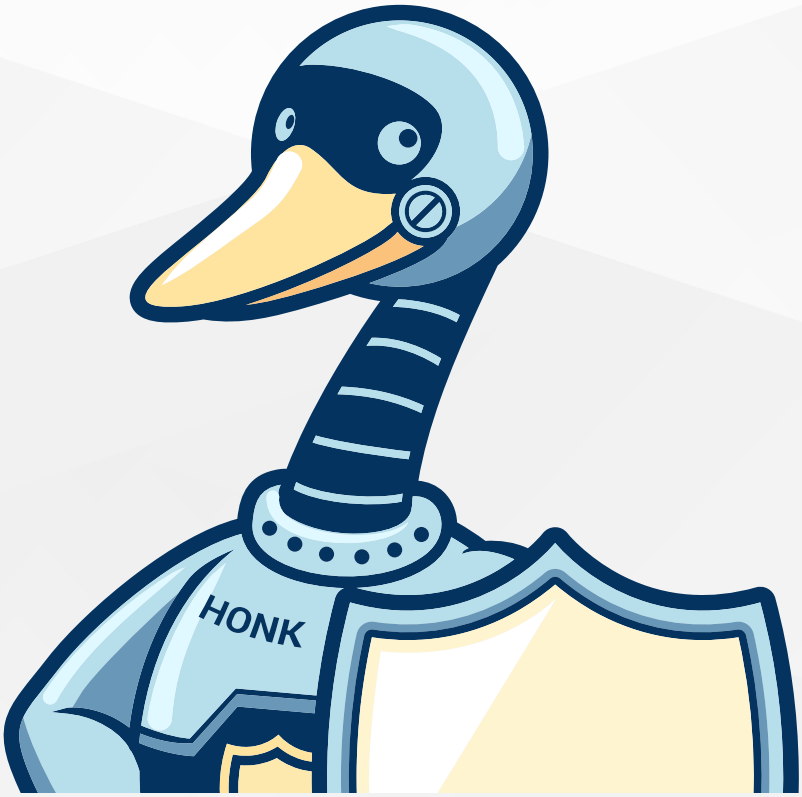
ソフトウェア開発者のセキュリティ教育改善プラン

openssf.org

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

概要

本稿は、セキュリティ教育のトレーニング教材とインセンティブを拡大することによって、世界中のソフトウェア開発者のセキュリティ教育を改善する方法についてアドバイスを提供します。このレポートでは、なぜセキュアなソフトウェア開発の教育が必要なのかを簡単に説明し、教材の現状をまとめます。次に、2022 年から 2023 年にかけての OpenSSF の教育への取り組み、特に「コンテンツの収集と管理」の必要性について議論し、重要な要件を明らかにしていきます。そして、OpenSSF の教育への取り組みに対する 2024 年に向けた提案と現在進行中の取り組みについてまとめます。付録 A では、利用可能な多くの関連教材について詳細を示します。付録 B では、主要なセキュアなソフトウェア開発のライフサイクルモデルについて述べます。



OpenSSF
OPEN SOURCE SECURITY FOUNDATION

もくじ

セキュアなソフトウェア開発教育の必要性

セキュアなソフトウェア開発教育が求められています。

CSO Online の調査によると、一般的にサイバー セキュリティのスキルは不足してると示しています。2016 年には「46% の組織がサイバー セキュリティ スキルの不足が問題であると回答」しており、2021 年にはこの状況が拡大し、「57% の組織が世界的なサイバー セキュリティ スキル不足の影響を受けている」となりました。2021 年には、埋まらない 400 万ものサイバー セキュリティ ポジションがあると推定されています（Dataconomy 2021）。この増加はもはや持続不可能です。

多くの政府は共通認識として重要な要因と判断しています。ソフトウェアは一般的にセキュア バイ デザインではなく、セキュア バイ デフォルトでもないということなのです。（2023 年 10 月の論文 *Secure by Design* を参照）。「本質的に安全でないものを安全にしよう」とするのは、成功する見込みがほとんどないでしょう。

ソフトウェアのほとんどが設計時点で安全でないのは当然のことです。なぜなら、ソフトウェア開発者の多くはセキュアなソフトウェアの開発方法を知らないからです。この知識不足のため、ほとんどのソフトウェアはセキュアではなく、ユーザーが期待する安全性を満たしていないことが多々あります。以下に、開発者は一般的にセキュアなソフトウェアの開発方法を知らないということを示す根拠を示します。

- 2019 年に米国の「コーディング」上位 40 校、米国以外の CS 上位 5 校でセキュア コーディングを必須とした学校はありませんでした（Forrester 2019）。
- U.S. News の CS ランキング上位 24 校のうち、学部生にセキュリティを義務付けているのは 1 校（カリフォルニア大学サンディエゴ校）のみ。
- ある記事では、「大学はコンピューター サイエンスの学生にセキュリティの教育をしていない」と指摘しています。
- ソフトウェア開発者の 53% が、所属する組織ではセキュアなコーディングに関するトレーニングが実施されていないと回答（Poneman 2020）。
- OSS のセキュリティを向上させる方法として 3 番目に多かった回答は、OSS コミュニティへのトレーニングの提供でした（Stephen Hendrick (VP Research, The Linux Foundation）& Martin McKeay（Senior Editorial Research Manager, Snyk）による「オープンソースソフトウェアにおけるサイバーセキュリティ上の課題への対応」2022 v2.0 調査、質問 q0050mrv による)。より上位のランクしている項目は、「セキュアなソフトウェア開発のためのベストプラクティスを定義する」と「上位 500 のオープンソース コンポーネントの脆弱性を分析し、修正するためのツールを提供する」であり、これらが唯一トレーニングと重ならない内容でした。
- ある調査では異なる結果が出ましたが、それは誤解を招くものでした。Secure Code Warrior による 2022 年の「開発者主導のセキュリティに関する調査」によると、89% の開発者がセキュア コーディングのスキルについて十分なトレーニングを受けたと報告しています。しかし、実際にはこの調査が示したのは、開発者が自分の知識を過大評価しているということです（ダニング＝クルーガー効果の残念な例ですした）。回答者の半数以上が一般的なソフトウェアの脆弱性やそれらの回避方法、どのように悪用されるかについて詳しく知りませんでした。92% の開発者はセキュリティ フレームワークについてもっとトレーニングが必要だと感じ、86% はセキュアなコーディングを実践するのが難しいと答えました。要するに、彼らは十分な知識を持っていると思っていたが、実際にはほとんど何も知らず、必要な知識が不足していました。

この知識の不足は、オープンソース ソフトウェア（OSS）でもクローズドソース ソフトウェアでも、すべてのソフトウェア開発者にとって障害となります。OSS の潜在的なセキュリティ上の利点は、大規模なピア レビューにより高品質を達成できることです。しかし、レビュアーが何を探すべきかを理解していなければ、ピア レビューは効果を発揮しません。セキュアなソフトウェア開発に必要な知識は、ライセンス形態に関わらず同じです。

すべてのソフトウェア開発の課題を解決する銀の弾丸はありませんが、ソフトウェアや製品開発のライフサイクル全体で最も重要で頻繁に使われるのは知識です。自動化やツールは特定の段階の特定のタスクに役立ちますが、ソフトウェア開発には常に人間の関与が必要です。ツールは有用ですが、その効果は使用する人の知識に依存します。防御的なコーディング スキル、脅威モデリング、セキュリティ コード レビューのスキルなど多くの

教育資料の現状に関する概要

技術は、常に知識豊富なエンジニアとともにあり、あらゆる活動において絶えず知見を提供します。サイバー セキュリティとセキュアなソフトウェア開発に関する質の高い教育とトレーニングは、実践的な知識とスキルを提供し、あらゆるタスクにおいて常に力を発揮する重要な要素となります。

サイバーセキュリティに関連する教育コースは数多く存在します。この文書では、ソフトウェア開発者向けのセキュリティ教育に特化したコースに焦点を当てています。これには、設計、実装、外部コンポーネントの選定、検証、そしてサプライチェーン攻撃への対策が含まれます。詳細は、付録 A「教育資料の現状に関する詳細(『収集とキュレーション』)」をご参照ください。

以下にその簡単な要約を示します。

- The OpenSSF Secure Software Fundamentals • OpenSSF のセキュアなソフトウェアの基礎コースは、セキュアなソフトウェア開発の基礎について最新の情報を提供する、人気があり高評価の無料コースです。自身のペースで進められる e ラーニング形式のため、簡単に多くの人が利用できます。このコースは効果的ですが、ラボ機能がありません。さらに高度で具体的な資料へのリンクを追加することでより良いものとなる可能性があります。
- SAFECode には、現在の開発の実践方法に反映できる、取り入れるべき興味深い資料があります。一般的なセキュアなソフトウェア開発のコースではありませんが、特定のトピックに特化しています。
- IBM の無料 Coursera コースは、おそらく OpenSSF の基礎コースに最も近い内容です。ラボ機能がありますが、Python を使用したウェブ アプリケーション開発に特化しているため、非常に限定的です。
- Snyk Learn - 開発者向けセキュリティは無料で、OWASP トップ 10 の脆弱性に対処するための短いレッスンに焦点を当てています。
- Securityjourney.com や Cydrill などのコースがあります。これらは、数千ドル程度の費用がかかり、商業を目的とした企業などの団体に適したものかもしれません。
- ISC2 や SANS など、セキュリティ教育に特化した多くの組織が、実務者向けに無料および有料のコンテンツを提供しています。しかし、これらは主に運用時のサイバーセキュリティに焦点を当てており、アプリケーションや開発のセキュリティにはあまり注力していません。ISC2 には CSSLP という資格がありますが、これはセキュアなソフトウェア開発の管理に重点を置いており、一般的に高レベルの内容を取り扱います。これらの資料はそれぞれの専門分野で価値がありますが、まだカバーされていない部分もあります。

私たちは、ソフトウェアの展開や運用に重点を置いたセキュリティ資料については調査しませんでした。代わりに、セキュア バイ デザインであり、セキュア バイ デフォルトを取り入れたソフトウェアの作成に注力したいと考えています。

背景：2022-2023 年の OpenSSF 教育プランに関する取り組み

この文書の背景を理解するためには、2022 年から 2023 年にかけて OpenSSF が行った教育計画の取り組みを把握することが重要です。

Log4Shell 脆弱性の発見を契機に、OpenSSF はオープンソースソフトウェア（OSS）のセキュリティ向上を目指す「動員プラン」を策定しました。この計画の第 1 ステージでは、ソフトウェア開発者の教育に重点を置きました。OpenSSF は教育に特化した OpenSSF 教育の Special Interest Group（SIG）を設立し、教育プランを策定しました。このプランでは、以下の 3 つのエリアに分けられています。

- **コンテンツの収集とキュレーション**：これは、既存の高品質なコンテンツを見つけ出し、必要な教育資料の不足を見極めることを目的としています。
- **トレーニングの拡大**：これは、1 つ目のエリアで収集したデータをもとに、「教育を支える三本柱」に基づいた新しい教材を作成し、すべての学習者が利用できる教材を提供することを目指しています。
- **開発者とメンテナへの報酬とインセンティブ**：これは、コースを受講し、そのスキルを証明した開発者や学習者の成果を見える化し、コミュニティや彼らを雇用したいと考える人々や企業との間での名声や地位を高める方法を促進することを目指しています。

この計画が作成されたとき、大量の資金が利用可能であると期待されていました。政府と産業界の双方が、Log4Shell が多大な影響を与えたことを認識しており、同様の問題の再発を防ぎたいと考えていました。そのため、この計画には多くの項目とそのリソースコストの見積もりが含まれていました。しかし、様々な理由から、この計画に対する資金提供は実現しませんでした。

しかし、すべてが失われたわけでは**ありません**。これら 3 つの分野は依然として有効な高レベルの戦略を示しており、当初の計画には興味深いアイデアが含まれています。必要なのは、低コストでスケーラブルで持続可能な最大の利益をもたらす努力に**絞り**、アイデアを実行に移すことです。結果として、元の大規模な計画ほど多様な機能は得られないかもしれませんが、それでも**実現可能**です。

これを達成するためには、まず特定の要件を明確にし、それに集中して成果を上げる必要があります。我々は特に、1 年以内に達成できることに焦点を当てることを提案します。その後、これらの重点分野に対するコンテンツの収集とキュレーションに注力し、上記の 3 つの分野の中で重要なものを実行します。

重要な要件

人によってニーズは異なります。その人の役割、その人が知っていること、その人が知るべきこと、そして利用可能なリソース（時間やお金）によるのです。これらの違いを明確にするために、この人物像を「ペルソナ」に分類します。

リソースは限られているため、まずは重点分野を特定することをお勧めします。これにより、他のステップが容易になります。例えば、具体的な要件がわからないと、関連する既存のリソースを特定するのは非常に困難ですが、焦点を絞ることでそれがより簡単になります。現時点では、以下の教育資料を優先的にご活用いただくことをお勧めします。

優先度	学習対象者	コンテンツ	備考
1	すべてのソフトウェア開発者	「セキュアなソフトウェア開発の基礎」コースの改良版	OpenSSF にはすでに人気があり、無料で、評価の高い基礎コースがを提供しています。しかし、オプションのラボが不足しています。ラボは学習効果を高めますが、利用には時間がかかるため、時間のない開発者には不向きです。また、提供するメディアを増やすという要望もあります。 これは無料である必要があります。なぜなら、すべてのソフトウェア開発者に利用してもらいたいからです。費用が発生すると、参加者が減り、その影響力が低下します。一度作成すれば、メンテナンス費用は比較的低くなると予想されます。なぜなら、基礎的な内容はほとんど変更されないからです。対象者は基本的に「すべてのソフトウェア開発者」であり、オンライン教育が求められています。 計画では、教育の SIG チームが 2024 年 9 月 27 日までに、ラボ一式と、コース全体を少なくとも 15 のメディアへ拡大し、リリースすることになっています。
2	開発者を監督するマネージャー	マネージャーが開発者にセキュアなソフトウェアを開発するために期待すべき知識と行動	マネージャーが開発者に何を期待すべきかを説明し、セキュアなソフトウェアを開発するために、適切な人材の採用、トレーニングの提供、管理、解雇ができるようにします。現在、これを広く要求する明確な方法がないため、無料または少なくとも低コストで提供することを考えています。現在、これは（基礎コースよりはるかに）短いもので、Intel が提供する資料に基づくものと想定しています。 計画では、2024 年 6 月 21 日までに草案を作成し、レビューアーに共有する予定です。
3	ソフトウェア開発者（特定のエコシステムや専門分野、例えば脅威モデリング）	特定のエコシステムやトピックに関するより深いセキュリティ知識	これは、さまざまなエコシステム（プログラミング言語など）や専門分野に焦点を当てた一連の教材となるでしょう。メンテナーの負担が大きくなる可能性があり、教材の数も多いことから、少なくとも一部は有料にする必要があるかもしれません。どの分野に焦点を当てるかはまだ未定です。データに基づいて選択できるよう、LF Research と協力して、今年取り組むべき分野を特定するための調査を行っています。これは有料コースになる可能性が高く、収益は OpenSSF に還元される予定です。 この計画は 2024 年 12 月 20 日までに完了する予定です。タイムラインが短すぎるかもしれませんが、調査の結果によっては、2024 年 10 月 24 日までに完了できることを期待しています。

重要な要件

このリストには根拠があります。

1. ソフトウェア開発者のための基礎は基本的なものです。多くのことはすべての開発者が知っておくべきことです。

2. 経営陣の理解と支援がなければ、その取り組みは効果を生む可能性が低くなります。マネージャーが作業できる必要はありませんが、何をすべきかを理解している必要があります。

エコシステムに焦点を当て、より専門的な資料は有益です。しかし、それぞれを開発するにはリソースが必要です。

パート 1（基礎）では、現在のコースでカバーされている内容を出発点とします。ソフトウェア開発者が知っておくべきことを目標とします。

1. **基礎：**セキュリティ、プライバシー、リスク管理などとは何か。

2. **要件：**一般的なセキュリティ要件。

3. **設計：**セキュリティ設計の原則（Saltzer & Schroeder を含む）。

4. **再利用：**ソフトウェア（特にオープンソース ソフトウェア）を安全に再利用する方法。

5. **実装：**一般的な実装上の脆弱性（OWASP Top 10 や CWE top 25 で特定されたものを含む）と、それらを体系的に防ぐ方法。

6. **検証：**一般的なツール（SAST、ファジング、Web アプリケーションスキャナー、シークレットスキャニング）やテスト（カバレッジ、ネガティブテスト）など、セキュリティ検証の方法。

このリストを検証し、これが対処すべき最も重要な優先事項であるかどうか（あるいは他の分野がより重要かどうか）、また、好ましい特定の種類の情報やフォーマットがあるかどうかを確認する手順を踏むことができます。Linux Foundation は、OpenSSF に参加している人々の見解を得るために調査を行うことができます。

以下に簡単に説明します。

1. **簡単にできるところは一般化しましょう。**セキュアなソフトウェアの開発は、OSS であろうとなかろうと基本的には同じであり、多くの開発者は OSS やクローズド ソースのソフトウェアを開発しています。簡単にできるところは、ソフトウェアの開発に問わず対応すべきです。OSS の選択も OSS に限ったことではありません。その知識は誰にとっても有用です。

2. **まずは単独の教材から始めましょう。**ソフトウェア開発に関する教材に「セキュアなソフトウェアの開発方法」を組み込むべきだという意見もあります。

3. **英語から始めましょう。**教材を多くの言語に翻訳することは望ましいことですが、まずは英語から始めるのが一番手っ取り早い方法です。

4. **教育は娯楽ではありません。**教材は面白いものであってもかまいませんが、学習することが目的です。特に、中身のない派手な動画や、学習者にとって何の相互作用もない動画を作成することは可能ですが、そのような動画では「学習者」はほとんど何も学べません。目標は常に、学習者が学習することであるべきです。

5. **持続可能なものにすること。新たな攻撃や防御方法が発見された際に、教材をアップデートできる価格帯でなければなりません。また、メンテナンス費用をどのように調達するかについても決定する必要があります。**

限られたリソースを節約するため、OpenSSF では当面はこれらに重点を置くべきではないと考えます。

- 経営陣。他のことが整うまでは、このことはあまり役立ちそうにありません。また、彼らの時間は貴重です。
- セキュアなソフトウェア開発に関する研究者のためのコース。大学がその役割を果たす可能性が高く、また、利用できるポジションの数も限られています。
- 運用。運用に関する教育は、他にも多くの機関が行っています。また、多くの OSS プロジェクトは、従来の意味での「運用」を行っておらず、単にコードをリリースしているだけです。

重要な要件

- 公共政策や規制コンプライアンスに関する専門的なニーズ。基礎コースでは、広く適用される一般的な規制上の問題（GDPR など）についても触れています。ただし、特定の政府機関や、特殊な規制やコンプライアンス要件を持つ人々向けに、特別なガイダンスを作成しようとしているわけではありません。例えば、NIST サイバーセキュリティ フレームワークおよび NIST プライバシー フレームワーク（NIST SP 800-53 を含む）や米国国防総省サイバーセキュリティ成熟度モデル認証（CMMC）への準拠といったテーマについては、対象外としています。

認定の不正行為対策として、さらなる措置を講じるという選択肢もあります。後述の通り、「基礎」コースでは、現在、単純な「修了証」に重点を置いており、これを回避する方法がいくつか考えられます。例えば誰かが他人の代わりに試験を受けるなどです。より複雑な認定プロセスを確立する価値があるかどうかは明らかではありませんが、場合によっては価値があるかもしれません。OpenSSF のパートナーである ISC2 や LF Training & Certification は、この分野での経験を積んでいます。

この文書は、OpenSSF の教育に関する活動について最小限の提案を記載したものですので、他の提案も大歓迎です。本稿執筆時点では、OpenSSF の教育 SIG において、セキュアなソフトウェア開発に関する教育（大学など）の最低要件を定義することについても議論されています。また、OpenSSF のさまざまなメンバーと、各組織内で OpenSSF の教育資料を活用する方法についても話し合う予定です。こうした追加の取り組みも歓迎します。

今後取り組むべきこと・現在進行中のこと

以上のことから、今後取るべきステップ（すでに進行中のものも含む）は以下のとおりです。

1. 基礎コースを改善する。

a. a. 基礎コースに関する 2023 年のフィードバックを分析しました。こちらを参照してください。

b. マルチメディア（動画クリップ、アニメーションなど）を追加。

c. ハンズオン ラボの追加。

   i. 任意のラボの導入方法を検討します。ハンズオン ラボを参照してください。

   ii. いくつかのサンプル ラボを作成します。

   iii. ラボを充実させます。テンプレートとサンプルが完成すれば、ボランティアにヘルプを依頼します。これは非常に容易に並列処理ができる作業となります。

2. 基礎コースと教育の必要性を広く一般的に認知させる。

a. Google 広告の費用を調査する。

b. 基礎コースの名称を統一する。現在、異なる名称が多く存在し、混乱を招く可能性があります。「セキュアなソフトウェア開発の基礎」や「セキュアなソフトウェア開発 (LFD121)」など。「基礎」という用語を追加したり、何らかの変更を加えることが重要かもしれません。これにより、基礎コースと様々な専門的なコースを区別することができます。

c. 教育資料を紹介する OpenSSF や Linux Foundation のブログ記事を作成する。

d. OSS NA で教育資料や進捗についての講演を行う（講演が採択された場合）。

e. セキュアなソフトウェア開発の基本について、地元で開催されるカンファレンスで発表する「アンバサダー」のグループを育成することです（これにより、トレーニングコストを削減できます）。例えば、OpenSSF の ”A Brief Introduction to Developing Secure software” などのプレゼンテーション資料を提供することができます。これらは、フルセットのコースへのステップとなります。

f. OpenSSF のメンバーに、このコースを推奨コースおよび LMS に追加するよう説得してください（SCORM Connect を使用して LMS システムに追加できます）。

g. DEI WG と協力し、多額の費用をかけずに「情報を広める」方法を見つけましょう。

3. セキュリティ教育の改善点を見極める。

a. LF Research 調査。LF Research と協力し、人々が何を必要としていると考えているかを把握するための調査を開始しました（異なるグループによってニーズの認識が異なる可能性があるため、対象者を分類する意図があります）。

b. OpenSSF 運営委員会（GB）による非公式アンケート。特に、当社の基礎コースを利用していないのであれば、利用しない理由は何でしょうか？ソフトウェア開発に関するセキュリティ教育で、他に何が必要でしょうか？

4. 開発者（メンテナーを含む）にセキュアなソフトウェアの開発方法を学んでもらうための報酬/インセンティブを用意する。

a. 開発者の基礎知識のレベルを自動的に判定する仕組みを導入する - プロジェクトに、この知識を持つメンテナーが少なくとも 1 人いるかどうかを判定するための、シンプルなドキュメント化された API を作成する。

   i. Scorecard や Best Practices バッジが利用される可能性があります。Scorecard のイシュー #3534 を参照してください。この点は Scorecard では評価されない可能性があります。

   ii. 例えば、LFD121 は完了時に Credly バッジを付与します（例）。GitHub ID/ GitLab ID/ メールアドレスといった紐づけられた情報を見つける必要があります。

   iii. この情報を決定するには、該当する人たちの個人情報が必要になります。私たちは、プライバシーを保護する方法について、LF の法務およびプライバシーの専門家と協議しています。

今後取り組むべきこと・現在進行中のこと

5. 開発/リリース マネージャー向けのコースを作成する。

a. 現在の計画では、インテルの成果物（スライド資料）を基に構築する予定です。

b. （公開されたら）教育 SIG でレビューし、問題なければ OpenSSF テンプレートに変換し、調整を行います（その際にはクレジットを記載します）。

c. これをオンラインコースにするべきか？ビデオにするべきか？これらは未定です。現在の計画では、インテルがリリースした内容を確認してから決定することになっています。

6. フォローアップ コースの順序を特定し、それらを実行に移す。

a. まずは、既存および計画中の LF T&C コースを整理することから始めます。

b. ISC2 などのパートナーと協力して、その他のコースを特定します。

c. 基礎コースをアップデートし、次のステップとしてそれらのフォローアップ教材を参照できるようにします。

d. その後、これらのコースを新たに作成または拡張する可能性があります。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

ここでは、既存のコースや教材について詳しく説明します。このセクションは、当初の計画における「コンテンツの収集とキュレーション」のステップの簡易版に相当します。これらのコースや教材の 1 つでも活用できるものがあれば、素晴らしいことです。活用できなくても、インスピレーションの源となるでしょう。ここでは、書籍ではなくコースに焦点を当てています。廃止されたコースや利用されていないコース、ソフトウェア開発と関連性のないコースについては省略しています。

このような教材の一般的なリストをいくつか挙げた後、何らかの形で重要または参考になりそうなものについていくつか説明します。完璧で完全なリストを作成することが私たちの目標ではありません。そのようなリストを作成するには時間がかかりすぎます。私たちの目標は、十分な教材を特定し、近い将来に何をすべきかに集中できるようにすることです。また、可能な場合は教材を再利用したり、そこから教訓を学んだりします。

コース教材のリスト

OpenSSF 教育 SIG の「よりセキュアなソフトウェアを開発するための教育資料のマトリックス - スプレッドシート」には、多くの教育資料、特に無料で利用できるものが数多く記載されています。不完全ではありますが、非常に多くの選択肢が記載されているため、優先順位付けが必要です。多くのリソースが「Static Guide / Documentation」として記載されていますが、少なくともクイズや演習のような何らかのインタラクティブ性があるコースを求めているため、直接関連性があるものとしては除外します。もちろん、これらは有用な資料です。

以下はその他のリストです。

1. Coursera は、コースそのものではなく、コースを公開するためのプラットフォームです。 https://www.coursera.org/courses?query=software%20security
2. https://www.g2.com/categories/secure-code-training
3. LF トレーニングプラットフォームで利用できるセキュリティトレーニングについて、OpenSSF のものも含め、簡単な概要をまとめました。 https://docs.google.com/document/d/1GZqtm90nj14CrZQlblZIHTUbnaE3TUZETkC2IZknVr4/edit#heading=h.uy1j1kt9p485
4. Google のスポンサー リンク。すべてを検証することはできませんが、Google のスポンサーリンクは、自社を検索結果に表示させるために料金を支払っている組織を示しているため、調査する価値があると考えました。

Google で「セキュアなソフトウェア開発コース」を検索すると、特に関連性の高い以下のスポンサー リンクが表示されました。

1. Security Journey の開発者およびソフトウェア開発ライフサイクルに携わるすべての方を対象とした、セキュアなアプリケーション開発トレーニング。 https://www.securityjourney.com/secure-coding-training-pricing
2. Cydrill - https://cydrill.com/ - は、開発者やテスターを対象としたオンラインまたはインストラクターによるトレーニングを提供しています。
3. Coursera で IBM が提供する「開発者および DevOps プロフェッショナル向けのアプリケーションセキュリティ」 https://www.coursera.org/learn/application-security-and-monitoring

以下でそれぞれ簡単に説明します。.

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

OpenSSF 基礎コース

OpenSSF は、<https://openssf.org/training/courses/> から無料で利用できる自習コース「セキュアなソフトウェア開発の基礎」をリリースしました。これはソフトウェア開発者を対象とした基礎コースです。Linux Foundation のトレーニング&認定プラットフォームと edX で利用できます。

無料で、多くのトピックをカバーしています。現在のところ、オプションの演習がないという弱点があります。オプションの演習を追加すれば、大幅に改善されるでしょう。また、管理、運用、特定のエコシステムに深く踏み込んだ内容には重点を置いていない点にもご注意ください。

OpenSSF「セキュアなソフトウェア開発に関する簡単な紹介」スライド資料

OpenSSF の "A Brief Introduction to Developing Secure Software"（セキュアなソフトウェア開発に関する簡単な紹介）は、30 ～ 40 分程度の基礎コースを要約した短いプレゼンテーションです。このプレゼンテーションには講演者が必要であり、このプレゼンテーションだけでは、聞き手に必要な基礎知識を十分に提供することはできません。

これは、開発者にとって有益な情報を提供し、基礎コースの受講を促す「導入編」と考えるのが一番良いでしょう。私たちは、OpenSSF の「アンバサダー」がさまざまなカンファレンス（地元で開催されるカンファレンスなど）でこれを発表することを想定しています。これにより、ソフトウェア開発者は必要な知識を得ることができ、無料のフルセットのコースの存在に気づいてもらえるでしょう。

OpenSSF - その他のコース

OpenSSF では、<https://openssf.org/training/> に記載されているような専門的トピックに関するコースを提供しています。

- OpenSSF スコアカードを用いてプロジェクトをセキュアにするコース：「OpenSSF スコアカードを用いてプロジェクトをセキュアにする（LFEL1006）は、Linux Foundation Training & Certification プラットフォームで提供されており、主にスコアカードのツールを使用するエンドユーザー向けに設計されています。このコースでは、OpenSSF スコアカードをソフトウェア開発ライフサイクルに統合する方法を学習します。」
- Sigstore によるソフトウェア サプライチェーンをセキュアにするコース：Sigstore によるソフトウェア サプライチェーンをセキュアにする（LFS182x）コースは、Linux Foundation Training & Certification プラットフォームで提供されており、Sigstore ツールのエンドユーザー向けに設計されています。「ライフサイクル全体を通じてセキュアなソフトウェアを構築し、配布することは容易ではなく、多くのプロジェクトがセキュリティ バイ デフォルトで構築するための準備ができていないのが現状です。攻撃や脆弱性は、ソフトウェアの記述からパッケージング、エンドユーザーへの配布に至るまでのあらゆる段階で発生します。Sigstore は、ソフトウェア サプライチェーンの安全性を向上させ、開発者が日々の業務の中でセキュリティ対策を実施する際に直面する課題を緩和する、数ある革新的なテクノロジーの 1 つです。」

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

SAFECode コース

SAFECode はこちらのサイト <https://safecode.org/> です。SAFECode は 2007 年に設立され、動画を含む多くの役立つ無料の教育資料を提供しています。いくつかの教材では、動画の後にオプションでクイズ問題が出題されます。これらの教材は、多くが以前から利用可能であり、質が高いことから、大きな影響力を持っています。

公開された動画は高品質であることが多いですが、メンテナンスが行き届いていない場合や、完全に放置されている場合もあります。例えば、「セキュアな Java プログラミング入門」の動画では、「Java のモバイルコード」は異なるものであると強調されていますが、実際にはモバイルコードは今日ほとんど使用されていません（その役割はすでに JavaScript に取って代わられています）。このケース（他のケースにも言えることですが）では、単に視聴するだけの動画です。クイズや演習、学習の成果を確認するためのテストなど、学習を促すための双方向的な要素は一切ありません。もちろん、SAFECode の教材の中にはクイズがあるものもあります。

根本的な問題として、動画の編集やグループでのピアレビュー、更新作業が非常に難しいことが挙げられます。このような、長期間にわたってメンテナンスが必要な教材の場合、更新が容易なテキストの作成を重視する方が理にかなっている場合が多いのです。変更される可能性が低い資料のビデオセクションを用意することは有益ですが、ビデオのみの資料は更新するのが困難です。音声やビデオを希望する人がいる場合は、最新の自動読み上げ機能（AI 機能付き）の使用や統合についてさらに調査することが有益かもしれません。なお、スクリーンリーダーを使用することもできます。

また、これは、単に受け身な動画だけでなく、クイズやその他の対話型の教材を用意することが重要であることを示しています。SAFECode にクイズやその他の対話型教材がない場合、動画は簡単に再生できますが、そこから学ぶことはできません。SAFECode には確かな情報があり、一部の開発者に情報を提供するための重要な仕組みとなっています。彼らの努力に感謝するとともに、彼らの例を参考にしたいと考えています。

全カタログは <https://safecode.org/training/> でご覧いただけます。おそらく最も一般的なのは、「クラウドアプリケーションのセキュアな開発のための基本的な実践方法 - 第 1 部および第 2 部」ですが、奇妙なことに、一般的なセキュアなソフトウェア開発のためのコースはありません。「セキュリティ開発ライフサイクル入門」は、ライフサイクルを作成することに重点を置いており、何をすべきか（実装など）には重点を置いていません。「システム強化入門」は有用ですが、セキュアなソフトウェアを開発するには十分ではありません。SAFECode のカタログには、特定のトピックに関する、有益ではありながらも非常に専門的な学習コースも多数含まれています。前述の通り、多くの場合、クイズ付きのバージョンを選ぶことができます（クイズ付きのバージョンがある場合はそちらを選ぶことをお勧めします）。

SAFECode のコースを順番に確認し、OpenSSF の基礎コースからリンクすべきものを確認すると良いでしょう（そして、リンクしてください）。

ISC2

ISC2 は、Certified Information Systems Security Professional（CISSP）で有名です。CISSP は、ここで述べられているものとは異なる範囲を対象としています。CISSP の目的は、個人が「最高水準のサイバーセキュリティプログラムを効果的に設計、実装、管理できるかどうか」を判断することです。

実際には、主に管理と運用に重点を置いています。セキュアなソフトウェア開発の基本をいくつか知っていることは、その一部に過ぎませんが、焦点を当てているわけではありません。ソフトウェア開発者に特化した資格ではありません。https://www.isc2.org/certifications/cissp

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

ISC2 Certified Secure Software Lifecycle Professional (CSSLP) <https://www.isc2.org/certifications/csslp> は、セキュアなソフトウェアの開発方法に重点を置いています。米国での受験料は 599 ドルです。実際には、より費用のかかる教育コースを先に受講する人が多いでしょう <https://www.isc2.org/register-for-exam/isc2-exam-pricing>。これらは 8 つの分野に分かれています。

- **ドメイン 1. セキュアなソフトウェアの概念**
- **ドメイン 2. セキュアなソフトウェア ライフサイクル管理**
- **ドメイン 3. セキュアなソフトウェアの要件**
- **ドメイン 4. セキュアなソフトウェア アーキテクチャと設計**
- **ドメイン 5. セキュアなソフトウェアの実装**
- **ドメイン 6. セキュアなソフトウェア テスト**
- **ドメイン 7. セキュアなソフトウェアの展開、運用、保守**
- **ドメイン 8. セキュアなソフトウェア サプライチェーン**

CSSLP は、セキュアなソフトウェア開発の管理に重点を置いている傾向があり、全般的にハイレベルな内容の質問が多いという意見もあります。例えば、Jayton Birch は、CSSLP がセキュアなソフトウェアを実装するためにソフトウェア開発者が必要とする技術的知識に重点を置いているわけではないことを強調しています。また、Alexandr Fadeev は、開発者ではなくマネージャーの視点で考える必要があることを強調しています。これは、CSSLP がより深いプロセスに関する知識（およびそれを証明する証明書）を必要とするソフトウェア開発者のマネージャー向けの深い情報として広く認識されていることを示唆しており、基本的にソフトウェア開発者向けというわけではありません。

OpenSSF/ISC2 は、2023 年にコラボレーションを計画していることを発表しました。
https://openssf.org/press-release/2023/11/02/linux-foundation-isc2-and-openssf-collaborate-to-target-secure-code-development/

とても楽しみです！

Securityjourney.com

Securityjourney.com では、開発者およびソフトウェア開発ライフサイクル（SDLC）に携わるすべての方を対象としたセキュアなアプリケーション開発トレーニングを提供しています。 https://info.securityjourney.com/secure-code-training-1 をご覧ください。価格は変動しますが、例えば、5 ユーザー、3 年契約の場合、年間 2,750 ドル（この方法で購入すると、ユーザー 1 人あたり 900 ドル以上）です。

彼らは、自分たちのアプローチを次のように説明しています。

- **攻撃的および防御的アプローチ** - ハンズオン形式のトレーニングでは、開発者がアプリケーションを破壊し、攻撃者の行動をシミュレートし、破壊したものを修正します。一連の作業が同じレッスンに含まれます。
- **コード修正に対する説明責任** - 既存のアプリケーション セキュリティ テストツールと統合し、自身のコードの脆弱性を特定し対処する、迅速な対応を行う開発者のトレーニングプラン。
- **Web ベースのサンドボックスでの実践課題** - ハンズオン用に用意された環境を使って、開発者が実際のシナリオに従って脆弱性を見つけ、修正し、競い合うことができます。
- **独自の、組み立て可能なアプローチ** - 改善に向けた自社独自の取り組みに基づいたっカスタマイズ可能なラーニングパス。

彼らのアプローチは、「ラーニング パス」ごとに分類された 800 以上のレッスンで構成されています。彼らは多くの「ラーニング パス」を定義し、ユーザーは使用するラーニング パスを選択します。ラーニング パスには 2 種類あり、それぞれ多くの具体的なパスがあります。

- **ロールベースのラーニングパス：**
  - ビジネス学習者
  - Web 開発者（バックエンド）

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

- Web 開発者（フロントエンド）
- ネイティブ開発者
- モバイル開発者（iOS）
- モバイル開発者（Android）
- データ サイエンティスト
- テスター
- DevSecOps
- クラウド エンジニア
- プライバシー エンジニア

- **コンプライアンスベースのラーニングパス：**
  - OWASP ラーニングパス
  - PCI ラーニングパス
  - エグゼクティブ オーダー ラーニングパス

彼らの教材のライブラリは、特定のプログラミング言語やフレームワーク、さまざまなアプリやテクノロジーに焦点を当てています。

ある参加者（David Russo）は、「無料ではないが、良い」と報告しています。

Cydrill

Cydrill は、自社のトレーニング プログラムについて、「ハッカーに負けないセキュア コーディング スキルをあなたの開発者に身につけさせます」と述べています。ビジネス リーダー、タレント マネージャー、ソフトウェア開発者の 3 つの役割向けのコースを用意しています。

2024 年 1 月 9 日時点で、37 のコースが用意されています。検索システムにより、自分の好みに合わせてコースを絞り込むことができます。

提供形態は次の 2 つがあります。

- E ラーニング（オンライン）
- インストラクター主導（オンサイトまたはオンライン）

また、コースを次のように分類しています。

- 対象者（開発者、テスター）
- テーマ（C、C#、C++、Java、Node、Python）
- プラットフォーム（ARM、クラウド、デスクトップ、ウェブ）
- 専門トピック（自動車、銀行・金融、ヘルスケア、機械学習、医療機器、ネットワーク セキュリティ、PCI DSS）

例えば、“Web Application Security”は 12 人の参加者を対象とした 3 日間のクラスで、Java に焦点を当てています。インストラクターが指導するクラスで、開発者向けです（ハンズオン形式）。26 のラボと 13 のケース スタディがあります。価格は 1 人あたり 2250 ユーロです。概要は以下のとおりです。

- サイバーセキュリティの基礎
- OWASP トップ 10 2021 年版
- 総括

コースの例としては、次のようなものがあります。

- Java によるデスクトップ アプリケーション開発
- Python の Web アプリケーションを対象としたセキュリティ テスト
- 医療機器向け、C および C++ のセキュア コーディング
- AWS での Python によるクラウド アプリケーション セキュリティ
- Azure での C# によるクラウド アプリケーション セキュリティ

多くの場合、1 つの技術を別の技術に置き換えた同様のコースが用意されています。これは、特定の情報を得ることができますが、1 つの技術にのみ焦点を当て、技術に依存しない文脈をあまり含んでいないため、比較的短期間で終わってしまいます。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

Cydril は、短い電子書籍“People over tools: the key to real software security（ツールよりも人：真のソフトウェア セキュリティの鍵）”を出版しており、そこには「セキュリティ上の問題を修正するのは素晴らしいことだろう。しかし、それを未然に防ぐことの方がさらに優れている」という素晴らしいキャッチフレーズが付けられています。電子書籍には、「DevSecOps は正しい方向への動きを表しているが、テストや運用にセキュリティ対策を追加するということは、ソフトウェア開発ライフサイクルの最終段階のみを対象としていることを意味する。それより前の段階はどうなるのか？」悲しいことに、多くの「DevSecOps」の実装は、テストと運用にいくつかのステップを追加するだけで、それ以外にすべてのプロセスにセキュリティを組み込むことはしていない、という彼らの意見は正しいと思います。また、「理想的なセキュア コーディングは、あらゆるプログラミング コースのカリキュラムに含まれるべきですが、現状ではまだそこまでのレベルには達していません（おそらく、当分は到達しないでしょう）。この重大な盲点を克服する唯一の方法は、開発者にセキュア コーディングのベストプラクティスを教育することです。最も一般的な脆弱性タイプへの対処方法と、適切な入力検証による強固なコードを記述するための防御的プログラミング技術の適用方法の両方について教育を行う必要があります。」と述べています。

Linux Foundation セキュリティ ワークショップ（SKFベース コース）

Linux Foundation セキュリティ ワークショップは、実際には次の 3 つのコースで構成されています。

- Understanding Vulnerabilities and Security Threats (WSKF603).
  - 1 日間のワークショップで、「OWASP® トップ 10 セキュリティ脅威」を理解することを目的としています。
- Securing Coding Fundamentals (WSKF601)
  - これは 3 日間のワークショップです。
- Advanced Secure Coding (WSKF602)
  - 一連の追加の演習です。

これらはすべてインストラクターが指導するコースであり、各自のペースで進めるコースではありません。通常は対面式で行われます。これらのワークショップは、もともと Glenn van Tate が指導していたコースに基づいて、コミュニティ カレッジなどの教育機関をサポートするために考案されました。SKF フレームワークを使用した多くのラボに重点を置いています。インストラクターには、Glenn van Tate（ヨーロッパ）と Randall T. Vasquez（米国）がいます。

価格と利用可能状況は、現時点では公表されていません。

IBM「開発者と DevOps 担当者のためのアプリケーション・セキュリティ」（Coursera）

The IBM のコース「開発者と DevOps 担当者のためのアプリケーション・セキュリティ」は Coursera を通じて受講でき、John Rofrano によって開発されました。約 17 時間で受講料無料のコースであり、OpenSSF の基礎コースとほぼ同じ時間です。受講者のレビューは総じて高い評価でした。

実践的な演習やコースの最後に実施するプロジェクトが含まれている点は、このコースの利点です。Python のプログラミング知識が必要であり、Python に特化した演習問題を使用しているため、Python を使用しない人にとっては難しいかもしれません。OWASP トップ 10 に重点を置いており、CWE トップ 25 は扱っていません（そのため、多くのトップ脆弱性と軽減策がこの教材では取り上げられていません）。（概要を見る限り）セキュリティを考慮した設計にそれほど重点を置いていないようです。1 つの潜在的なリスクとして、特定の技術（OpenSSL や Python など）に重点を置きすぎているため、他の技術を使用しているユーザーは、必要な基礎知識を学べない可能性があることが挙げられます。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

このコースには、以下の 4 つのモジュールがあります。

- **アプリケーション開発のためのセキュリティ入門** - 「このモジュールでは、セキュリティがワークフローにどのように組み込まれるかを理解し、セキュリティの概念や用語に関する実務的な知識を得ることができます。ソフトウェア開発ライフサイクル（SDLC）におけるセキュリティ設計の方法や、DevSecOps として知られる一連の実践方法についても学ぶことができます。また、OSI モデルについて学び、開発者にとって必要な OSI レイヤーを明らかにし、アプリケーション開発の 4 つのレイヤーにセキュリティ対策を実施します。セキュリティ パターンに関する知見を深め、それらを整理する方法を学びます。TLS（Transport Layer Security）と SSL（Secure Sockets Layer）について説明し、SDLC において TLS をセキュアに保つ方法を理解し、OpenSSL とその利用目的について学びます。また、脅威や脆弱性からアプリケーションを保護するために、セキュリティをコードの初期段階から組み込むための戦略、ベストプラクティス、評価方法についても学びます。さらに、脆弱性スキャナーや脅威モデルなどのツールをどのように使用してセキュリティ上の脆弱性を最小限に抑える方法についても学びます。また、認証、暗号化、完全性といった重要な用語を自分のセキュリティ分野の語彙に追加するいい機会にもなります。最後に、OpenSSL を使用してファイルの暗号化と復号化を行う実習や、Nmap を使用してネットワーク環境をスキャンする実習も行います。」
- **セキュリティテストと緩和策** - 「このモジュールでは、開発から運用においてアプリケーションをセキュアに保つための重要なリスク軽減戦略を学びます。また、静的解析、動的解析、脆弱性解析、ソフトウェア コンポーネント解析、継続的なセキュリティ解析など、さまざまなセキュリティ テスト手法についても学びます。コードレビューの実施方法と、アプリケーション開発におけるランタイム保護の確保についても学びます。また、静的解析、動的解析、脆弱性スキャン、および脆弱性検出に基づくハンズオンラボも実施します。」
- **OWASP アプリケーション・セキュリティ・リスク** - 「このモジュールでは、Open Web Application Security Project（OWASP）とそのトップ 10 のセキュリティ上の懸念事項について学びます。アプリケーションの脆弱性について学び、セキュリティのエキスパートや専門家が懸念するトップの脆弱性を知ることができます。SQL インジェクション、クロスサイト スクリプティング、機密情報の安全な保管について学びます。また、ソフトウェアとデータの整合性に関する障害を調査し、この種の脆弱性を検出する方法を発見し、その影響を軽減する方法を検討します。また、Snyk を使用してコードリポジトリを分析する方法や、Vault Python API（hvac）を使用して Vault 内のキーバリュー型シークレットの読み取り、書き込み、削除を行う方法についても実習します。」
- **セキュリティのベストプラクティス、最終プロジェクト、アセスメント** - 「このモジュールでは、コーディングのベストプラクティスとソフトウェアの依存関係について学びます。また、集中管理されたリポジトリに保存すべきものと GitHub に保存すべきでないものを決定することで、開発環境をどのようにセキュアにするかを学びます。さらに、flask-talisman を使用して HTTP セキュリティヘッダーを作成し、pass CLI（コマンドラインインターフェース）を使用してシークレットを安全に保存および取得するためのハンズオンラボを行います。最終プロジェクトとして、GitHub 上のコードを脆弱性の影響度に応じて確認し、脆弱性を修正します。また、脆弱性リスクを軽減するためのベストプラクティスを習得します。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

IBM: Application Security for Developers (edX)

IBM のコース“Application Security for Developers”は edX を通じて受講でき、John Rofrano によって開発されました。受講料は無料です。所要時間は約 45 時間（1 週間あたり 8 ～ 10 時間の 5 週間）と記載されており、同様の名称の Coursera コースの所要時間の約 3 倍です。Coursera コースのアウトラインと同じであり、同じような内容をカバーしていることから、外部の資料を見る限りでは、Coursera コースの長いバージョンであると思われます（同じ内容であれば、Coursera の所要時間が edX の 3 分の 1 である理由は不明です）。

長所と短所は似ています。長所は実践的な演習です。短所は、OWASP トップ 10 についてのみ説明しているため、広く知られている多くのトップ脆弱性がまったく取り上げられていないことです。また、（概要を見る限り）セキュリティを考慮した設計に重点が置かれているようには見えません。

以下はそのシラバスを日本語表記したものです。

- **モジュール 1** - アプリケーション開発のためのセキュリティ入門
  - セキュリティ バイ デザイン
  - DevSecOps とは何か
  - 脆弱性スキャンと脅威モデリング
  - 脅威の監視
  - アクティビティ：セキュリティの概念と用語
- **モジュール 2** - セキュリティテストと緩和策
  - セキュリティテストと緩和策の概要
  - 静的解析
  - ハンズオン ラボ：静的解析の使用
  - 動的解析
  - ハンズオン ラボ：動的解析の使用
  - コード レビュー
  - 脆弱性分析
  - ハンズオン ラボ：脆弱性分析の評価
  - ランタイム保護
  - ソフトウェア コンポーネント分析
  - ハンズオン ラボ：ソフトウェア コンポーネントの分析を評価する
  - 継続的なセキュリティ分析
- **モジュール 3** - OWASP アプリケーション・セキュリティ・リスク
  - OWASP（トップ 10）のセキュリティ脆弱性入門
  - OWASP トップ 1-3
  - OWASP トップ 4-6
  - OWASP トップ 7-10
  - SQL インジェクション
  - その他の SQL インジェクション攻撃の種類
  - ハンズオン ラボ：SQL インジェクションを理解する
  - クロスサイト スクリプティング
  - ハンズオン ラボ：クロスサイト スクリプティング
  - シークレットの安全な保管
  - ハンズオン ラボ：シークレットの安全な保管
- **モジュール 4** - セキュリティのベストプラクティス
  - コード プラクティス
  - ハンズオン ラボ：コード プラクティス
  - 依存関係
  - ハンズオン ラボ：依存関係
  - セキュアな開発環境
  - ハンズオン ラボ：セキュアな開発環境
- **モジュール 5** - 最終試験

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

Implementing DevSecOps (LFS262)

"Implementing DevSecOps" (LFS262) は、299 ドルの LF コースです。このコースでは、「オープンソース ソフトウェアを使用して、ソフトウェア デリバリー パイプラインに DevSecOps のプラクティスを組み込む方法」に重点を置いています。「このコースのハンズオン ラボの演習を行うには、受講者はインターネット アクセス、Web ブラウザ、Git、クラウド プロバイダーのアカウント（Google Cloud Platform や AWS など）が必要です。」

このコースでは、セキュアなソフトウェア開発の基本的な知識があることを前提としています。このコースでは、セキュアなソフトウェアを開発する方法ではなく、セキュリティ関連のツールをパイプラインに統合する方法に焦点を当てています。ツールはソフトウェアをセキュアに設計するものではありません。また、ツールには多くの偽陽性や偽陰性があります。これは、基本を理解している開発者にとっては問題ないことですが、開発者は依然としてソフトウェアをセキュアに設計する必要があり、ツールの結果を理解して使用する必要があります。

概要 (コースは英語です) :

- **第 1 章 コース紹介**
- **第 2 章 DevSecOps とは?**
- **第 3 章 ラボ環境のセットアップ**
- **第 4 章 DevOps パイプラインの構築**
- **第 5 章 SCA によるサプライ チェーンの保護**
- **第 6 章 静的セキュリティ テスト（SAST）**
- **第 7 章 コンテナ イメージの監査**
- **第 8 章 安全なデプロイと動的アプリケーション セキュリティ テスト（DAST）**
- **第 9 章 IAC によるシステム セキュリティ監査**
- **第 10 章 Kubernetes デプロイメントのセキュリティ確保**
- **第 11 章 Vault によるシークレット管理**
- **第 12 章 ランタイム セキュリティの監視と改善**

基礎コースでは、パイプラインの実装方法に関する追加教材として LFS262 を紹介していることにご留意してください。

Roadmaps

「ロードマップ」は、インタラクティブな閲覧が可能なシステムで、大変興味深いものです。一連の学習教材の構造（「ロードマップ」）を示します。ログインしてさまざまな項目をクリックすると、さらに詳しく読むことができます。サイバー セキュリティのロードマップを見るには、https://roadmap.sh/cyber-security を参照してください。セキュアなソフトウェアの開発についても、同様のビューを作成することができるでしょう。

ソースの素材は GitHub にあります。
https://github.com/kamranahmedse/developer-roadmap

It これは、時折「オープンソース」と誤って呼ばれますが、実際には「オープンビュー」であるようです。ライセンスについてはこちらをご覧ください。
https://github.com/kamranahmedse/developer-roadmap/blob/master/license

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

小規模なコース

iOpenSSF 教育 SIG は、非常に狭い範囲に焦点を絞った小規模なコース（express learning コースなど）を作成するというアイデアについて議論をしました。それは、非常に狭い範囲に焦点を当てたもの（例えば、特定のプロセスや特定のエコシステムにおける特定の脆弱性など）です。このような小規模なコースの作成には、それほど多くのリソースを必要としません。その一例が「Express Learning」コースです。状況によっては、このコースが効果的な場合もあります。

しかし、Timothy Serewicz（Director、Training Program、LF Training & Certification）は、「小規模なコース」には重要な問題があるとしています。コースをこのように個別に開発し、細分化してしまうと、学習者がコースを見つけにくくなるだけでなく、コース間の統合も不十分になりがちです。どのコースも、多くの小さなレッスンに分割する必要があります。しかし、学習者が選択できる大きなユニットを用意することが重要です。さもないと、コースを見つけ、選択するまでに時間がかかりすぎ、コースの順序が意味をなさなくなります。

これは、レッスンを学習しやすいユニットに分割すべきである一方、それらを単に個別に作成するのではなく、より大きな構造として考えるべきであることを示唆しています

OWASP WebGoat

OWASP WebGoat <https://owasp.org/www-project-webgoat/> は、「あなたのような熱心な開発者が、一般的なオープンソースコンポーネントを使用する Java ベースのアプリケーションに一般的に見られる脆弱性をテストできるようにするために、意図的にセキュリティ上の脆弱性を残したアプリケーションです。」OWASP トップ 10 に基づく短いレッスンが用意されています。これは学習の助けにはなりますが、それ自体が完全なコースであるとは言い難いでしょう。

GMU Design & Implementation of Secure Software (SWE/ISA 681)

David A. Wheeler は、ジョージ メイソン大学（GMU）で“Design & Implementation of Secure Software”（SWE/ISA 681）というコースを対面式の大学院コースとして教えています。大学院コースとして、より詳細な内容（例えば、バッファ オーバー フローが発生した場合に基盤となるプラットフォームで何が起こるのか、オリジナル資料の多読など）を扱っています。このトピックについてより深く学びたいと考えている大学院生にとっては、大学院コースを受講することは有益でしょう。ただし、一部の詳細については、学部生には必要ありません。最も重要な部分の簡略版が、基礎コースのスタート地点として用いられました。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

Introduction to Cyber Security Specialization

Edward G. Amoroso 博士による Coursera コース “Introduction to Cyber Security Specialization” は、サイバー セキュリティの専門分野に関するハイレベルな概要説明です。ソフトウェア開発に関する具体的な内容ではなく、サイバー セキュリティを専門とする可能性のある人向けの基礎分野について学習します。

概要：「サイバー セキュリティの専門分野の紹介は、受講者が現代の情報およびシステムを保護するテクノロジーと方法についてより深い理解を身につけることを目的としています。学習成果はシンプルです。受講者がサイバー セキュリティに対する生涯にわたる情熱と敬意を育むことを期待しています。それが将来の取り組みに役立つことは間違いありません。 学生、開発者、管理者、エンジニア、そして一般市民も、この学習体験から恩恵を受けるでしょう。サイバー セキュリティの概念を実際のビジネス経験に結びつけるために、業界関係者との個別の特別インタビューも収録されています。」

基本的に以下の 4 コースで構成されています。

- サイバー攻撃入門
- サイバー攻撃対策
- リアルタイムのサイバー脅威検知と軽減策
- 企業およびインフラのセキュリティ

IT Security: Defense against the digital dark arts

Coursera のコース “IT Security: Defense against the digital dark arts” は、一般的なセキュリティ概念とその運用に焦点を当てています。対象者は IT サポート担当者です。ソフトウェア開発についてはあまり詳しく説明されていないため、私たちが検討している教育資料の要件には該当しません。

概要：「このコースでは、IT セキュリティに関するさまざまな概念、ツール、ベストプラクティスを取り上げます。脅威や攻撃、それらのさまざまな形を紹介します。暗号化アルゴリズムの背景と、データの保護にどのように使用されるかを説明します。その後、情報セキュリティの 3 つの A、すなわち認証、認可、アカウンティングについて詳しく説明します。」

- セキュリティ上の脅威を理解する
- （暗号学）
- サイバー セキュリティの 3 つの A：認証、認可、アカウンティング
- ネットワーク セキュリティ
- 多層防御
- セキュリティを重視する企業文化の確立
- IT サポートの仕事に備える

SANS SEC275: Foundations: Computers, Technology, & Security

SANS’ SEC275: Foundations: Computers, Technology, & Security コースは、受講者がコンピューターについてほとんど知らないことを前提に、コンピューター技術について抽象的な説明から、Linux、Python、C などの具体的な例題まで幅広く取り上げています。

このコースでは、既存の開発者が必要としない多くの内容（コンピューター技術やプログラミング方法など）をカバーしていますが、安全なソフトウェア開発に不可欠な内容はカバーしていません（OWASP トップ 10 や CWE トップ 25 の多くをカバーしていないなど）。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

まず、システム アーキテクチャ、オペレーティング システムとは何か、コンテナとは何か、そして Linux オペレーティング システムのコマンド ラインの使用方法について具体的に学びます。さらに、プログラミングの基礎についても学びます。最後のセクションでは、バッファ オーバー フローやフォーマット文字列、暗号技術など、セキュリティについてさらに詳しく説明します。

このコースでは、コンピューターの概念について「ゼロから」入門することができます。しかし、セキュアなソフトウェア開発におけるほとんどの概念はカバーされていません。シラバスでは、セキュアデザインの原則についても触れられていません。バッファ オーバー フローについては触れられていますが、OWASP トップ 10 や CWE トップ 25 で特定されている最も一般的な脆弱性の種類をすべて特定し、網羅する試みはなされていません。

全体的なシラバスは次のとおりです。

- SEC275.1：システム アーキテクチャ、オペレーティング システム、Linux
- SEC275.2：検索、ウェブ、ネットワーク
- SEC275.3：サーバーとプログラミング入門
- SEC275.4：セキュリティ概念と高度なセキュリティ概念

SANS は「コース内容は 50 ～ 60 時間で修了できますが、ほとんどの学生はコース内容を何度も復習し、ラボや小テストを繰り返したり、追加の演習を行ったりするため、平均修了時間は 120 ～ 140 時間になります」と報告しています。

Linux コマンドラインと IDE を備えたオンラインラボシステムを使用しており、90 以上のラボが用意されています。すべてウェブブラウザからアクセスできます。

ウェブベースのコースは 3,020 ドル、別途認定試験を受ける場合は 380 ドルです。

SANS の「サイバー エース」教材 廃止されたため、ここでは取り上げません。

SANS Security Awareness Developer Training Program

SANS の “SANS Security Awareness Developer Training Program / Web Application Security Awareness Training” - 2 つの異なる名称がありますが、同じ教材を 2 つの異なる名称で提供しているようです。タイトルが示すように、これは意識向上のためのトレーニングであり、開発者がセキュアなソフトウェアを作成できるようになるためのトレーニングではありません。しかし、「意識向上」トレーニングとしては、期待されるよりも少し深い内容となっています。OWASP トップ 10 2021 に重点を置いていますが、バッファ オーバー フローやモバイル アプリケーション セキュリティについても説明しています。

データシートに、カバーする内容について説明しています（時間は分単位に丸められています）。

- OWASP トップ 10 2021。49 分バージョンと 62 分バージョンがあります。
- モバイル アプリケーション セキュリティ（38 分）
- Web アプリケーションセキュリティのためのインタラクティブ モジュールの適用（各言語 7 分。対応言語は Node.js（JavaScript）、C#、Java、Python、PHP）
- セキュアコーディングの原則
- よく見られる設計の欠陥（24 分）
- モダンなアプローチ（フルスタックの利用、API の利用、クラウド開発者）（17 分）
- 脅威の認識（脅威モデリングを含む）（29 分）
- 古典的な問題（バッファ オーバー フローを含む）（32 分）
- ソフトウェア開発ライフサイクル（SDLC）（ウォーター フォール、アジャイル、DevOps）（34 分）

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

「SANS セキュリティ意識向上の開発者向けトレーニングには、最も一般的な 5 つのプログラミング言語をカバーする 40 以上のモジュールが含まれており、従来のトレーニングとインタラクティブなトレーニングの両方を利用できます」と記載されています。

価格は公開されていません。しかし、Center for Internet Security は、「SANS 開発者向けセキュリティ意識向上トレーニングプログラム」を会員向けに「大幅割引」で提供しています。

2024 年 1 月 16 日時点で、「SANS 開発者トレーニングの次回の大幅割引購入期間は、2022 年 12 月 1 日から 2023 年 1 月 31 日までです。10 ユーザーまで 1 年間のトレーニングの最低注文額は 2,890 ドルです。それ以降は 1 ユーザーあたり 289 ドルとなります。10 ユーザーまで、2 年間のトレーニングの最低注文額は 5,780 ドルです。それ以降はユーザー 1 人あたり 578 ドルとなります」と書かれています。おそらく個人であれば、はるかに高い料金を支払うことになるでしょうが、他の組織であれば同様の割引を交渉できるかもしれません。

SANS：その他の資料

SANS には多くの資料があります。その例としては、以下のものがあります。

- Eric Johnson の 2015 年のブログ記事“Securing the Software Development Lifecycle”では、セキュアな開発についてまとめ、「... でさらに詳しく学び、無料のデモ版に登録してください」と書かれていますが、ドメイン securingthehuman.org は現在使用されていません。
- Thien La による SANS ペーパー“Secure Software Development and Code Analysis Tools”(2002 年)。2 つのパート、「セキュアなプログラミング ガイドライン」（Perl、Java、C/C++ に重点を置いています）と「ソースコード分析ツール」で構成されています。これは論文なので、弊社のコースリストの対象外です。また、20 年以上前のものです。

Synk Learn - Security for Developers

“Synk Learn”には、さまざまなレッスンとラーニングパスが用意されていますが、ここでは「開発者のためのセキュリティ」というラーニングパスに焦点を当てます。これは、OWASP トップ 10 で指摘されている実装上の脆弱性対策に焦点を当てた短い一連のレッスンです。そのほか、セキュアな設計に関するレッスンも 1 つ用意されています。脅威モデリングについては（安全ではない設計のレッスンで）言及されていますが、説明はされていません。より広範な背景（要件とリスク管理）や検証については説明されていません。JavaScript コードの例が示されています。

このラーニングパスは、以下の特定のレッスンで構成されています。

- アクセス制御の不具合
- 安全でないハッシュ
- 安全でないランダム性
- クロス サイト スクリプティング（XSS）
- コード インジェクション
- クロス サイト リクエスト フォージェリ（CSRF）
- プロト タイプ汚染攻撃
- NoSQL インジェクション攻撃
- SQL インジェクション（SQLi）
- XML 外部エンティティ参照（XXE）
- XPath インジェクション
- 安全でない設計
- 脆弱で時代遅れのコンポーネント
- ディレクトリ トラバーサル
- ログの脆弱性
- サーバー サイド リクエスト フォージェリ（SSRF）

各レッスンは簡単なクイズで終わります。サンプル内の JavaScript の例は、非常に役立つようです。

付録 A：教育資料の現状に関する詳細（『収集とキュレーション』）

“Security for Developers”は、Snyk とニューヨーク大学タンドン工学部との提携により提供されています。誰でもコースを修了し、修了証をダウンロードできますが、業界バッジを受け取れるのはニューヨーク大学タンドン校の学生のみです。

メンバーから寄与された資料

インテルは、ソフトウェア開発者のマネージャーを教育することを目的とした新しいコースを OpenSSF に提供しました。スライド資料であるこの教材の目的は、マネージャーがセキュアなソフトウェアを開発するためのソフトウェア開発の管理方法を理解できるようにすることです。

現在の計画では、この資料を更新し、ソフトウェア マネージャー向けの短期コースの教材として活用することになっています。将来的には、クイズや最終試験問題付きのビデオ教材になる予定です。

この分野におけるメンバーのさらなる貢献を、当グループは常に歓迎しています。メンバーからの寄与は、教材の作成や拡充のプロセスを迅速に進めるのに役立ちます。また、教育活動全体に対するコントリビューターや協力者の数を増やすことにもつながります。多くの、全てではないにしても、OpenSSF のメンバーは、コミュニティへの貢献にふさわしいセキュリティ教育教材を持っています。そうすることにより、これらの組織は、作業に対する支援が得られるコミュニティのコントリビューターの集まりに参加し、それまで内部で維持していた資料の継続的なメンテナンスコストを削減することができるのです。

提供の可能性がある潜在的な資料

私たちは、作成した資料を OpenSSF や、あるいは単に世界全体に対して、共有できる可能性のある組織と協議中です。私たちは彼らに会ったことはありませんが、彼らと長時間にわたる協議を重ねており、うまくいくことを期待しています。

特に、米国海軍は安全なソフトウェア開発に関する資料をいくつか所有しており、その中には学ぶべき攻撃の例も含まれていると聞いています。これらは、ラボでの実習用サンプルとして役立つかもしれません。しかし、現時点ではまだそれらを目にしたこともなければ、公開されたこともありません。

最終的には、それらの組織が資料を配布するかどうかを（また、その確認プロセスについても）決定することになります。私たちは、資料の提供をぜひお勧めします。

付録 B：セキュアなソフトウェア開発ライフ サイクル モデル

セキュアなソフトウェア開発をうまく機能させるには、ソフトウェア開発ライフ サイクル（SDLC）全体に組み込む必要があります。ISO/IEC/IEEE 12207:2017（「システムおよびソフトウェア工学：ソフトウェア ライ フサイクル プロセス」）などの文書では、ソフトウェアの開発とデプロイの際に使用される一般的なプロセスが説明されています。また、セキュアなソフトウェア開発を SDLC に組み込む方法について具体的に説明している文書もあり、重要なトピックがきちんとカバーされているかどうかを確認するために参照する価値があります。ここでは、セキュアなソフトウェア開発を SDLC に組み込む方法について述べている重要な文書をいくつかご紹介します。

NIST Special Publication 800-218

NIST Special Publication 800-218 は Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities（2022 年 2 月）です。この資料では、「組織は、（セキュアソ フトウェア開発フレーム ワーク（SSDF））を既存のソフトウェア開発プロセス全体に統合し、SSDF の規約を使用して第三者サプライヤーにセキュアなソフトウェア開発要件を明示し、SSDF で説明されている実践方法に適合したソフトウェアを取得すべきである」と主張しています。また、「SDLC の早い段階でセキュリティ対策が講じられるほど、最終的に同じレベルのセキュリティを達成するために必要な労力とコストが少なくて済む」という重要な指摘もしています。この原則は「シフト レフト」として知られており、技術的負債を最小限に抑えることができるため、より強固なセキュリティと回復力を備えたソフトウェアを実現できるという点において、非常に重要です。

新しい実践方法を導入するのではなく、「確立された基準、指針、セキュアなソフトウェア開発の実践に関する文書に基づいて、一連のハイレベルな実践方法を説明」します。これにより、さまざまな資料を幅広い枠組みに抽象化できるという利点があります。この文書では、これらは 4 つのカテゴリーに分類されています。

- 組織の準備（PO）：組織は、組織レベルに応じたセキュアなソフトウェア開発を実行するために、人材、プロセス、テクノロジーの準備を整える必要があります。
- ソフトウェアを保護する（PS）：組織は、ソフトウェアのすべてのコンポーネントを改ざんや不正アクセスから保護する必要があります。
- 十分に安全なソフトウェアを生産する（PW）：組織は、リリースされるソフトウェアにセキュリティ上の脆弱性が最小限に抑えられた、十分に安全なソフトウェアを生産すべきです。
- 脆弱性への対応（RV）：組織は、ソフトウェア リリースにおける残存する脆弱性を特定し、それらの脆弱性に対処し、将来的に同様の脆弱性が発生することを防ぐために適切に対応する必要があります。

理論的には、この文書はあらゆる SDLC に適用されるはずですが、実際には、この文書の一部は残念ながらリスクの高いウォーターフォール型アプローチを助長しています。例えば、最初の項目である「ソフトウェア開発のためのセキュリティ要件の定義（PO.1）」では、「ソフトウェア開発のためのセキュリティ要件が常に把握され、SDLC 全体を通じて考慮され、要件に関する情報が一度に収集し共有されるため、作業の重複が最小限に抑えられるようにする」と書かれています。実際には、システム廃棄まで要件（セキュリティ要件を含む）が完全に把握されていないことがよくあります。要件を正確に把握しようとするのは賢明ですが、要件は往々にして十分に理解されず、時間とともに変化します。要件を「常に」把握しておくことは、いわゆる「ウォーターフォール」プロセスの特徴です。1970 年に Winston W. Royce noted in 1970 博士が指摘したように、ウォーター フォール プロセス（他の作業を行う前にすべての要件を特定しようとするなど）は、実際には「リスクが高く、失敗を招きやすい」ものです。したがって、これらの実践方法の一部を適用する際には、慎重さが必要です。さらに、これは他の多くの古い文書をまとめたものであるため、新しいアプローチや懸念事項がこの資料から省略されている可能性があります。

付録 B：セキュアなソフトウェア開発ライフ サイクル モデル

とはいえ、こうした制限はあるにせよ、この NIST の文書は、セキュリティを考慮したソフトウェア開発プロセスに関連する重要な分野を明らかにしています。

BSIMM

Building Security In Maturity Model (BSIMM) は、NIST Special Publication（SP）800-218 で使用されている資料のひとつですが、BSIMM はそれ自体も重要な資料であるため、別途記載しています。BSIMM は、130 以上の組織のソフトウェア セキュリティの取り組み（アプリケーション セキュリティ、プロダクト セキュリティ、DevSecOps プログラムとも呼ばれる）を分析したものです。これにより、多くの企業が安全なソフトウェアを開発するためにどのような取り組みを行っているかについての知見が得られます。特定の組織が選択する方法は必ずしも効果的ではないかもしれませんし、別の組織にとっては適切ではないかもしれません。しかし、その結果は、組織が何をやっているのか、何が特に一般的であるのかについて有益な知見を提供します。

BSIMM ソフトウェア セキュリティ フレームワークは 4 つのドメインに分けられ、各ドメインはさらに 12 の実践方法に細分化されています。これらの実践方法はさらにアクティビティに細分化されています。ドメインと実践方法は以下のとおりです。

1. **ガバナンス**：戦略と評価基準、コンプライアンスとポリシー、トレーニング
2. **インテリジェンス**：攻撃モデル、セキュリティ機能と設計、基準と要件
3. **セキュアなソフトウェア開発ライフ サイクル（SSDL）のタッチ ポイント**：アーキテクチャ分析、コード レビュー、セキュリティ テスト
4. **デプロイメント**：侵入テスト、ソフトウェア環境、構成管理および脆弱性管理（CMVM）

BSIMM 2023 によると、これらのアクティビティが最も頻繁に行われていることがわかりました（1 番が最も一般的）。

1. セキュリティ チェックポイントとそれに関連するガバナンスを導入する。
2. インシデントレスポンスを作成または連携させる。
3. 個人情報保護に関する義務を明確にする。
4. 外部からの侵入テストを実施して問題を発見する。
5. ホストおよびネットワークセキュリティの基本が確実に実施されるようにする。
6. 自動コード レビュー ツールを使用する。
7. QA 中に境界値分析テストを実施する。
8. セキュリティ機能のレビューを実施する。
9. 規制上の要件を統一的に管理する。
10. セキュリティ ポータルを作成する。

最も大きな増加が見込まれる（したがって、頻繁に見られるようになる可能性が高い）活動は次のとおりです。

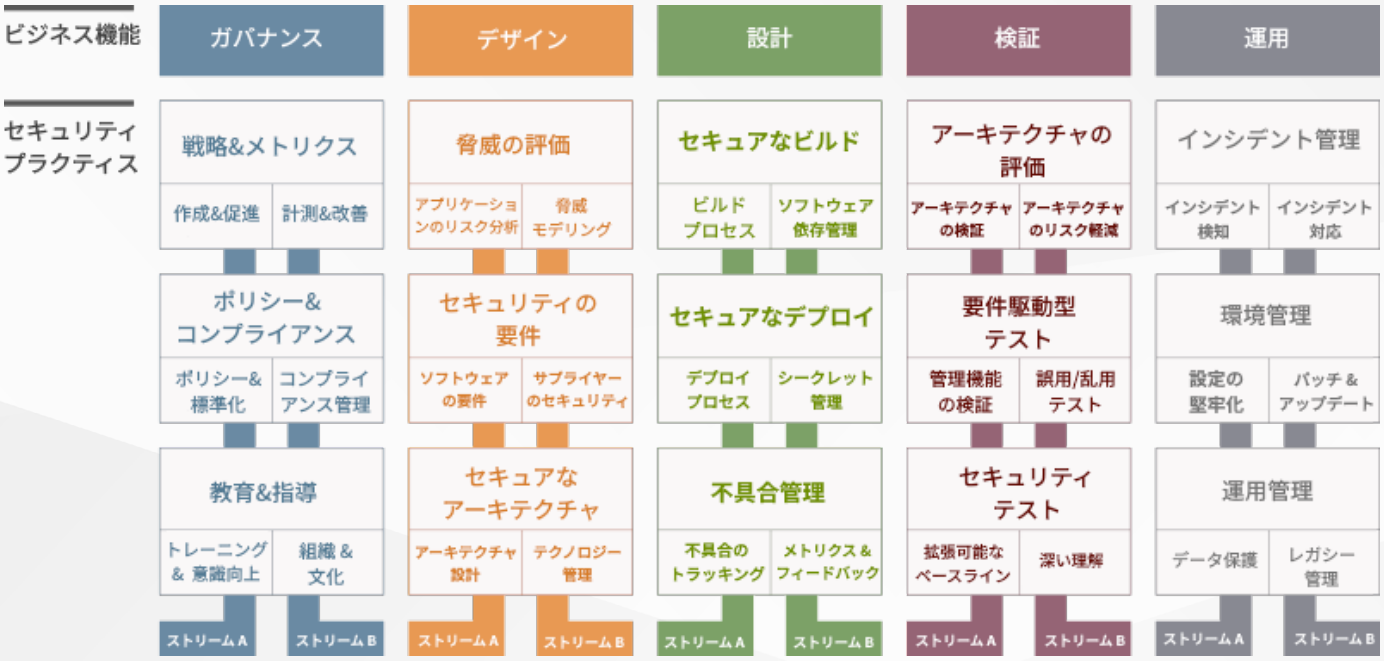
1. 外部からの申告に対する脆弱性対応の合理化。
2. クラウド セキュリティ管理を導入する。
3. すべてのプロジェクトでコード レビューを義務付ける。
4. 新しい攻撃手法を研究するグループを設置する。
5. 安全なデプロイメントのパラメータと構成を定義する。
6. セキュリティ上の目標を達成するためにアプリケーション コンテナを利用する。
7. アプリケーションを網羅する定期的な侵入テストを計画する。
8. オープンソースを特定する。
9. ソフトウェアコンプライアンスの履歴を記録する。
10. セキュリティ チェックポイントの実施と例外の追跡。

詳細については、BSIMM レポートをご覧ください。

付録 B：セキュアなソフトウェア開発ライフ サイクル モデル

OWASP SAMM

OWASP ソフトウェアセキュリティ保証成熟度モデル (SAMM) の目的は、「セキュアな開発ライフサイクルを分析し、改善するための効果的かつ測定可能な方法を提供すること」です。これは、セキュアなソフトウェアの開発と運用に関する重要なタスクを特定することを重視しています。OWASP SAMM の次の図は、視覚的に理解しやすい表現となっています。



OWASP SAMM の少し変わった点として、他のほとんどのモデル（ISO/IEC/IEEE 15288、ISO/IEC/IEEE 12207、その他多数）では別個に扱われているセキュリティ要件が、このモデルでは「設計」の一部として扱われていることが挙げられます。しかし、これは些細な問題です。なぜなら、「設計」というビジネス機能は、実際には「要件と設計」に関するものだからです（おそらく、この長い名称が図にうまく収まらなかっただけでしょう）。SAMM のより大きな特徴は、「実装」がビジネス機能として定義されているにもかかわらず、SAMM にはソフトウェアのセキュアな実装（つまり、セキュアなソースコードを記述し、一般的な脆弱性のタイプを回避すること）をカバーするセキュリティのプラクティスが見当たらないことです。アーキテクチャ、構築、テストはセキュリティのプラクティスとして取り上げられていますが、ソースコードの実装は取り上げられていません。この記事を書いている時点では、OWASP SAMM をいくつかの調査質問の指針として使用し、この問題に対処するために、SAMM にセキュアな実装に特化したカテゴリーを追加する予定です。

ライセンス

著者について

David A. Wheeler
Director of Open Source Supply Chain Security
Open Source Security Foundation (OpenSSF)

Dr. David A. Wheeler は、オープンソース ソフトウェア（OSS）とセキュアなソフトウェア開発に関する専門家です。セキュアなソフトウェア開発に関する彼の業績には、”セキュア プログラミングの HOWTO”コースである Open Source Security Foundation(OpenSSF) Secure Software Development Fundamentals コース、「多様な二重コンパイルによる信頼を信頼するための完全な対策」などがあります。彼は Linux Foundation のオープンソース サプライ チェーン セキュリティ担当ディレクターであり、ジョージ メイソン大学（GMU）でセキュアなソフトウェア開発に関する大学院コースの講師を務めています。Wheeler 博士は、ジョージ メイソン大学（GMU）で情報テクノロジーの博士号、コンピューター サイエンスの修士号、情報セキュリティの修了証、ソフトウェアエンジニアリングの修了証、電子工学の学士号を取得しています。彼は、Certified Information Systems Security Professional（CISSP）であり、米国電気電子学会（IEEE）の上級会員でもあります。彼はバージニア州北部在住です。

本訳文について

この日本語文書は、Plan for Improving Software Developer Security Education の参考訳として、The Linux Foundation Japanが便宜上提供するものです。英語版と翻訳版の間で齟齬または矛盾がある場合（翻訳版の提供の遅滞による場合を含むがこれに限らない）、英語版が優先されます。

この日本語文書を引用する際には、下記の一文を記載してください。
引用：Plan for Improving Software Developer Security Education 参考訳（The Linux Foundation Japan提供）

翻訳協力：松本央

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Thank you!

openssf.org