BLACKDUCK®

堅実なソフトウェア・セキュリティ・イニシアティブ（SSI）を立ち上げる方法

（さらにそれを必要とする 10 の理由）

目次

ソフトウェア・セキュリティに関する戦略はお持ちですか？

今年、あなたは 46 個の Web アプリケーションと 19 個のモバイル・アプリ、20 個のクライアント・サーバー・アプリをテストしました。新しいアプリケーション・セキュリティのテスト・ツールも購入し、それで 112 個の脆弱性を発見しました。気分は上々です。

しかし、浮かれる前に次のことを自問してみてください。リスクは大幅に減りましたか？そもそも減っていますか？重要な脆弱性を放置していませんか？役員会は、あなたの業務の重要性や実績の成果を理解していますか？

これらの質問に対する答えが分からないとすれば、それはソフトウェア・セキュリティのテスト計画はあっても、ソフトウェア・セキュリティに関する戦略が欠けているということです。

すでに、アプリケーション・セキュリティのテストに投資をしているのであれば、リスク低減の方向としては間違っていません。ただし、今こそ次の段階に進むべき時です。すなわち、アプリケーション・セキュリティ対策をコストセンターから組織の競争優位性へと転換するために、ソフトウェア・セキュリティ・イニシアティブ (SSI) を立ち上げるということです。

このガイドの対象者

このガイドは、以下のような経験がある人に向けたものです。

- 直感だけに頼ってセキュリティ予算の投資先を決めてきた
- セキュリティ問題の優先順位と修復を巡って開発チームともめたことがある
- セキュリティの要件と成果を経営陣や他部門に伝えるうえで課題を抱えていた
- 同じセキュリティ上の不具合が何度も同じチームによって発見された
- 開発スケジュールの変更や規制の見直しによるキャパシティ問題に対処するためのリソースの確保に奔走した
- アプリケーションの脆弱性のテストを土壇場で要求されたことで製品の発売が遅れた
- 情報漏洩のニュース（Twitter、Uber、Twilio、DoorDash など）を耳にして、「自分の会社で起きる可能性はあるか？」と考えたことがある
- 過去にセキュリティ計画の不備で損失を被ったことがある
- セキュアなソフトウェア開発ライフサイクル（SDLC）が欠如していたこと、またはどのベンダーがセキュアで優れたソフトウェア・プラクティスを採用しているかを知らなかったことで顧客に譲歩を求められ、取引が遅延したことがある
- 連邦取引委員会などの規制機関の監視下に置かれたことがある

正直にお願いします。あなたの答えを記録することはありません。いずれかについて身に覚えがあれば、さらに読み進めて、SSI を立ち上げて展開するための確立された手順を学習してください。それによって現在のセキュリティに関する取り組みが構造的かつ戦略的で堅実なプログラムに生まれ変わるはずです。

でも、アプリケーション・セキュリティ・テストは既に実施しているので、それで十分では？

一言で言えば、それでは不十分です。

ケーススタディやホワイトペーパーで、アプリケーション・セキュリティ・テストが事実上のソフトウェア・セキュリティ技術として提示されているのをよく目にしますが、これは企業がセキュリティを真剣にとらえていることを示すのに使われる特効薬のようなものです。

アプリケーション・セキュリティ・テストはすべてのセキュリティ・プログラムにとって重要で不可欠な要素に違いありません。とはいえ、単にペンテストで脆弱性を発見してパッチを当てるというだけでは、到底セキュリティ戦略とは言えません。アプリケーション・セキュリティ・テストはスタートであって、ゴールラインではないのです。

プロアクティブなセキュリティによって
時間とコストを節約できますが、
それでも十分ではないでしょう。
セキュリティ・プログラムは、
リスクにさらされる度合いを全般的に
減らすために実行すべきものです。

タイラー・シールズ
Forrester Research シニア・アナリスト

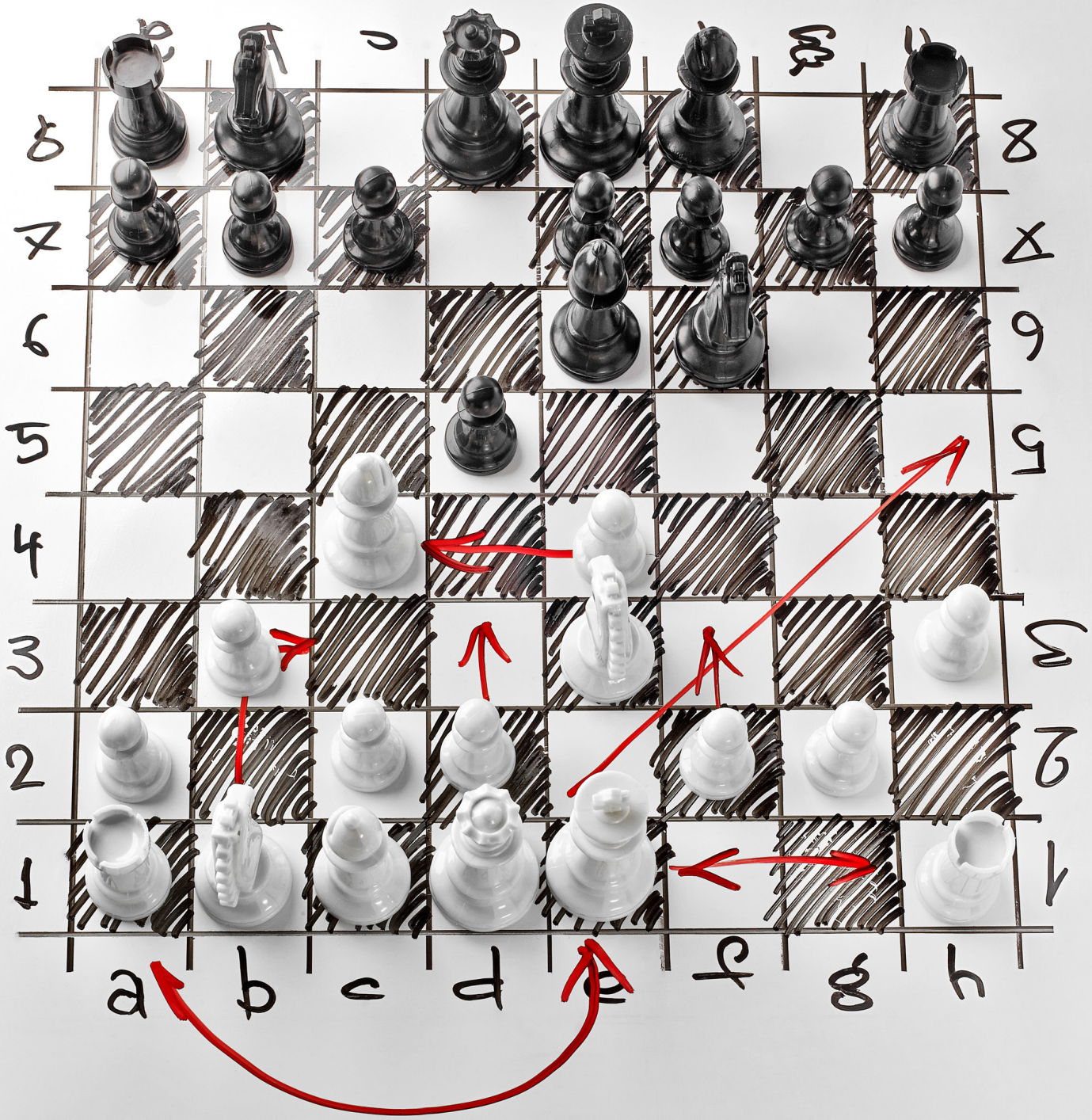
ソフトウェア・セキュリティ・イニシアティブ（SSI）が必要な理由のトップ 10

SSI を立ち上げることには、以下のように多くのメリットがあります。

1. 優先的に、受容できないリスクに確実に対処する。
2. 作業の中断を最小限に抑えてセキュアなソフトウェアを作成する方法を開発者に提供することで生産性が向上します。
3. 特定の担当者やグループにソフトウェア・セキュリティのリスク軽減の責務を与えることで、困難な仕事もやり遂げられます。
4. セキュリティ・チームと開発チームの間を正式に橋渡しして、優先順位や責務、動機付けを共有することで混乱を抑え、全員がより効率的に協力できるようにします。
5. 製品マネージャー、アーキテクト、開発者、テスター、およびその他すべての利害関係者のソフトウェア・セキュリティ要件を文書化して意見を擦り合わせ、組織的な調整を行います。
6. 社内チームや外部ベンダーを含むソフトウェア・サプライチェーン内の全員に一貫性のある期待値を提供することで、ソフトウェアがどこから来たものであろうと、ソフトウェアがセキュアに構築されていることを信頼できます。
7. すべてのソフトウェア・セキュリティのニーズ（ポリシー、標準、ツール、専門家）に関するセンター・オブ・エクセレンスを確立し、人々が答を得てスキルを向上させる場を提供します。
8. 成果を評価し、顧客、パートナー、および役員会に報告できるようになります。
9. ソフトウェア開発と関連する全ステーク・ホルダーに対して一貫した支援や訓練を確実に提供し、セキュリティを優先する文化を根付かせます。
10. 開発チームのニーズの変化に対応しながらリスクを管理します。

堅実なソフトウェア・セキュリティ・イニシアティブ（SSI）の設計図

最も効果的な SSI は、組織に適合するよう微調整され、スタッフ、プロセス、およびソフトウェアのポートフォリオに基づいて拡張できるように構成されています。SSI によって、リスクを低減するための明確に理解できる方法論を導入し、どのように投資判断を行なったかを説明することで、自分の成果を示すことができるようになります。



SSI の強固な基盤を確立する（あるいは時代遅れの SSI をよみがえらせる）最善の方法は、次の 5 つのアプローチだと考えられます。

構築　　評価　　検証　　改善　　管理

構築

SSI の適切な基礎を固めるためには、次のように必要となる要素がいくつかあります。すなわち、優先度を設定するための一部の重要な情報、管理構造、セキュリティを開発サイクルに組み込むための訓練やツールなどです。

それでは、それぞれについてさらに詳しく見ていきましょう。

すべてのセキュリティ・リーダーが知っておくべき 5 つのこと

アプリケーションのセキュリティ対策の優先度の設定を始める前に、課題の全範囲を把握しておく必要があります。以下について自問してください。

- どんな開発プロジェクトが進行中か、その納期は？
- どのチームがどのアプリケーションに関わっているか？
- どのコードが社内開発で、どのソフトウェアが市販品やサードパーティー製か？
- 技術的な最大のリスクはどこにあるか？
- アプリケーションの資産目録にはどんなアプリケーションがあって、事業への影響が最も大きいのはどのアプリケーションか？

現場で調査してみましょう。これらの質問に今すぐすべて答えられなくても構いません。SANS Institute の最近の調査によると、回答者の 4 分の 1 以上が自社で使用または管理しているアプリケーションの数を知らなかったということです。

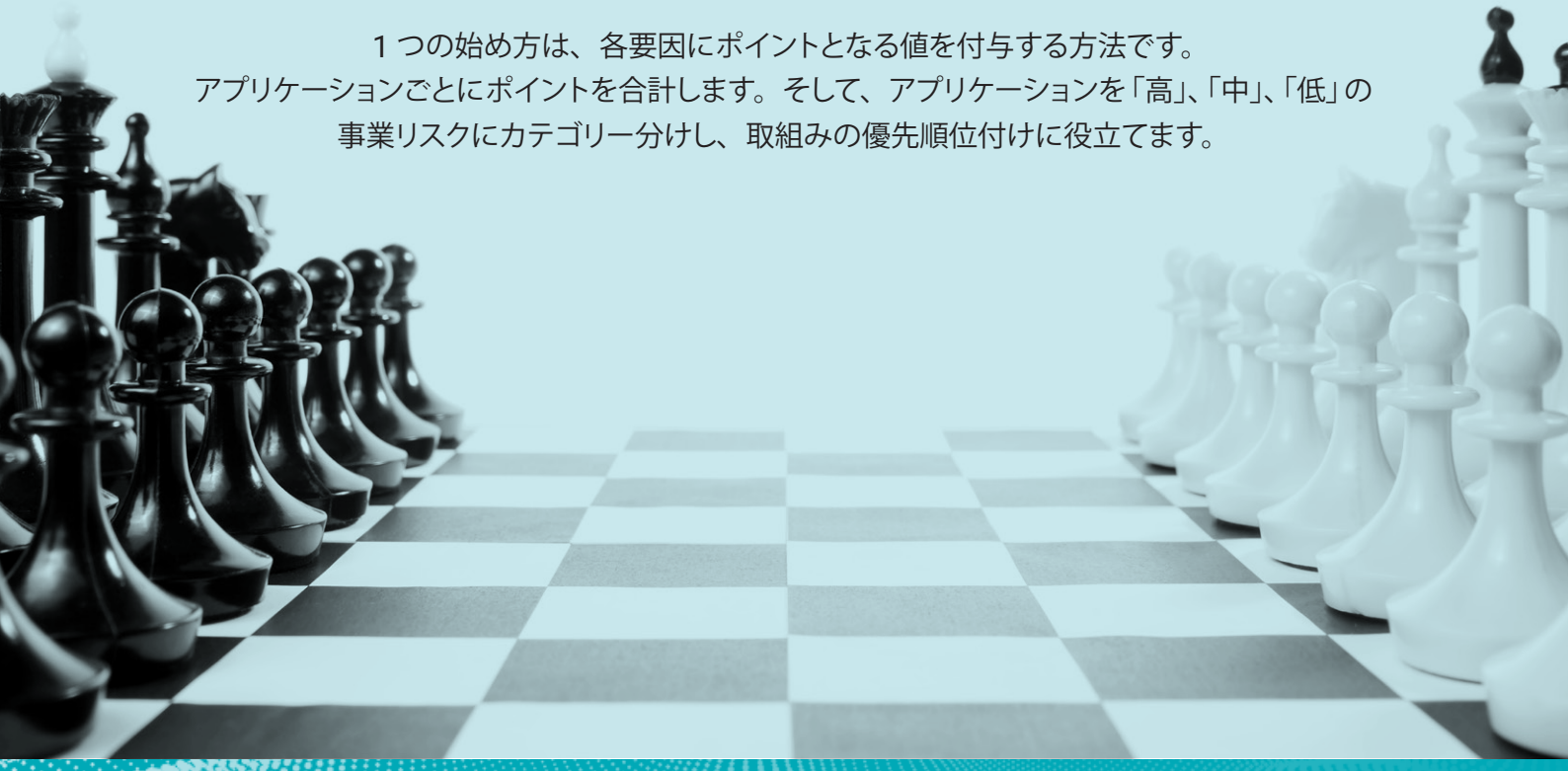
現在把握していることから始めて、継続的に知識の棚卸しを進めましょう。

リスク = 可能性 x 影響

アプリケーションの事業に影響を与える要因には以下が含まれます。

- 収益との関係
- 事業の継続性に対する影響
- コンプライアンス要件や規制要件
- サービスを提供する相手
- 保存またはアクセスされる機密データの量
- アクセス方法
- 他のシステムとの接続／統合
- 人の安全
- 国の安全保障

1 つの始め方は、各要因にポイントとなる値を付与する方法です。
アプリケーションごとにポイントを合計します。そして、アプリケーションを「高」、「中」、「低」の事業リスクにカテゴリー分けし、取組みの優先順位付けに役立てます。



構築

堅実な SSI の秘訣

SSI を成功させる秘訣はガバナンスです。ガバナンスによってセキュリティ対策の責務と対処に対する期待が定められ、持続可能な SSI の基礎を築き、規模を適正化するために必要な施策です。

ガバナンスを構築するのが、正式なポリシーや明確な基準、体系的なプロセスを用いる集中型セキュリティ・グループであろうと、アプリケーション開発チーム向けの技術標準やコーディング標準を用いるスクラム・マスターであろうと違いはありません。実際には、誰かが責務を負い、ソフトウェア・セキュリティに対して求められた期待を背負わなければ、「セキュアな SDLC」を確立することはできないからです。

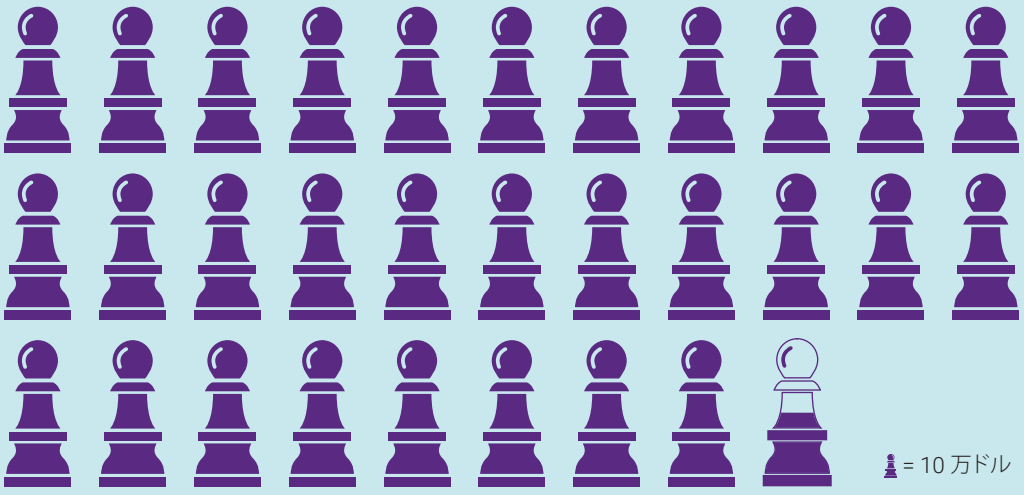
注意!セキュリティ・ポリシーは全ての利害関係者と共に作成しましょう。最終的な責任はセキュリティ・リーダーに委ねられますが、アプリケーション・セキュリティについては、他のリーダーたちと広く議論して組織全体に浸透させる必要があります。最も重要なのは、早期に開発チームを参加させ、ポリシーの作成と実行に対する当事者意識を徹底することが重要です。

ほとんどの企業がセキュアな SDLC を備えていません。

というのも、セキュアな SDLC を備えているという主張が多いにもかかわらず、大半の企業がソフトウェア・セキュリティのガバナンスもなければ、アプリケーション・ポートフォリオのセキュリティ態勢に対する体系的な対策をも持ち合わせていないからです。

このような組織になってはなりません。顧客は良しとしないでしょう、保険会社や規制団体や、会社の幹部や役員会もそれを受け入れるとは思えません。例として、セキュリティ・ポリシーが与える影響に注目してみましょう。

セキュリティ・オートメーションを導入したことで侵害のライフサイクルを 74 日短縮



セキュリティ・オートメーションを完全に導入したことで平均 305 万ドルのコストを節約

5 つの重要なセキュリティ・ポリシー

1. ソフトウェア・セキュリティ：ソフトウェア・セキュリティを達成することの高い期待を組織内で伝え、セキュリティを製品の要件や実装、調達、デプロイ、運用に組み込むこと。

- **セキュアな SDLC：**適用は必須である。
- **アプリケーションのリスク・ランキング：**事業にとって最も重要なアプリケーションがどれかを判断する明確な指標を示す。
- **アプリケーション設計：**システム設計にセキュリティ対策の組み込みを義務づける。
- **アプリケーション開発：**特定の技術スタックと遵守すべきコーディング規約を義務づけ、開発者に明確な指標と事前に構築済みのセキュア・バイ・デザインのモジュールを提供する。
- **アプリケーション・テスト：**どのアプリケーションをテストすべきか、どのゲートを合格すべきかを判断し、テスト間隔のスケジュールを設定する。
- **ソフトウェア・プロジェクトの影響度ランキング：**ソフトウェア・プロジェクトの影響度ランキングを定義し、ランキングによって関連するセキュリティ保証の取組みがどのように推進されるか概説する。
- **不具合の重大度と修正：**バグや欠陥の重大度を設定する規則と、コーディング上のバグや設計上の欠陥を修正するスケジュールを定める。

2. ネットワーク・セキュリティ：アプリケーション・セキュリティに有効なプロトコルと権限レベルを決定する。

3. データ・セキュリティ：重要な知的財産や顧客の機密データを特定して分類し、開発者が正しいセキュリティ機能を適用できるようにする。

4. 物理セキュリティ：アクセス制御を管理して、物理インフラを保護する。

5. 災害復旧：攻撃が発生した際に取るべき措置を決めること。例えば、アプリケーションに対する攻撃の報告、記録、分析など。

ソフトウェア・セキュリティ・ポリシーと他のポリシーとの間の整合性を確保すること。

定着する訓練を開発する秘訣

ソフトウェア開発ライフサイクルに関わる全員（経営陣、中間管理職、製品担当者、テスター、システム・アーキテクト、開発者、およびその他全員）が、自分の役割と関連付けられたソフトウェア・セキュリティに関する責務を、どのように果たすかを知っている必要がある。

なぜ開発者なのか？

3 年ごとに Open Web Application Security Project (OWASP) から、セキュリティ意識を向上させるために、Web アプリケーション・セキュリティの脆弱性トップ 10 のリストが発表されます。SQL インジェクションやクロス・サイト・スクリプティング（XSS）などの良く知られた脆弱性は、毎年のようにリストに載っているにもかかわらず、ソフトウェア開発者は、未だに、これらの脆弱性を繰り返しアプリケーションにコーディングし続けています。

効果的な SSI では、その核心部分、つまりコードが書かれた場所でアプリケーション・セキュリティに対処する必要があります。アプリケーションのビルドからバグや欠陥を早く除去できれば、その分、QA ステージでの時間とコストがかかる修正の必要が少なくなります。その結果、セキュアなアプリケーションをより早く市場に投入でき、同時に競争優位にもなるはずです。

SSI の計画にインセンティブを盛り込み、機能を実現するだけでなく、セキュアなコードの作成能力を向上するように開発者を促すことが不可欠です。また、対面やオンラインのトレーニングの機会を提供することで開発者をサポートできます。とは言え、インセンティブを真剣にとらえてキャリアパスの一部として価値を認めることを開発者に求めるのであれば、インセンティブを業績評価や報酬に組み込む必要があります。

セキュリティを開発工程に組み込むツール

動的解析やペンテストなどのセキュリティ・テストの技術は、セキュリティ・チームがさまざまな脆弱性を一貫して特定するのに有効です。とは言うものの、こうした技術が本番または本番前の状態のアプリケーションで使用される場合、問題の修正には時間もコストも嵩んでしまいます。

セキュア SDLC でのシフトレフトに役立つツールを探しましょう。セキュリティ・テストと修正に早く対処できれば費用対効果と生産性が向上します。

> 「不具合に早期に対処することで、生産性が 15% 向上します」
>
> —ジム・ルース
> Aetnaの最高情報セキュリティ責任者

既存のワークフローと使用している技術（例えば、統合開発環境など）にセキュリティ・ツールを直接組み込むことで、開発者は最初からセキュアなコードを作成できるようになります。しかし、セキュリティ・ツールを利用するために、開発者はプロセスの変更を求められ、好みのシステムを取り上げられたりするようだと、セキュリティ・ツールを使ってもらうのは困難でしょう。

評価

ソフトウェア・セキュリティは測定できないと考えている人は少なくありません。ソフトウェア・セキュリティの目標は、攻撃の成功を阻止することです。何も発生していないことをどうやって測定しますか?

セキュリティ侵害が発生していたことを知らないというだけで、発生していなかったということにはなりません。また、セキュリティ侵害が発生していなくても、将来発生しないということにはなりません。

自分の会社にハッカーが侵入するのを防いだことを証明できないとしても、他の方法で SSI の成果を示すことができます。

内部評価基準は、事業目標に向けた継続的な改善に役立ちます。SSI の目標を定め、それを基礎となる事業目標に結びつけます。そして結果を共有することで、SSI が組織の運営方法をどのようにして根本的に変えたかを示すことができるはずです。

運用プロセスの改善だけでなく、ソフトウェアの出荷を早め、費用を節約していることを経営陣に示すことで、セキュリティ・プログラムは一連の「確認」作業から重要な事業機能へと変わるはずです。

証拠が無いことは、無いことの証拠にはなりません。

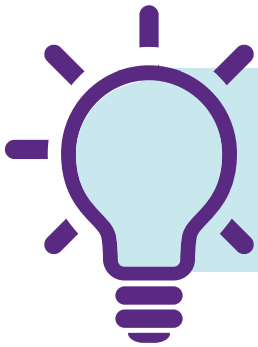
経営陣の言葉で話す方法

経営陣が理解し、評価する測定指標に注力すれば、SSI に対する継続的なサポートの獲得やリソースの増強の主張ができる可能性が高まります。

外部評価基準によって、さまざまな分野の SSI との比較が可能になります。内部の改善を示すことに加えて、自社の SSI を他社と比較することで、進捗状況に関する経営陣の視点を広げることもできます。課題を直視すること。つまり、他社の活動を知ることは、企業のトップがセキュリティを真剣にとらえる強い動機付けになる可能性があります。

SSI の評価と計画に関して業界全体をリードするモデルは、セキュア開発成熟度モデル（BSIMM）です。BSIMM プロジェクトは、ソフトウェア・セキュリティを強化することが実証済みのソフトウェア・セキュリティ・プラクティスを評価し、あらゆる種類の企業で採用されています。これによって、データに裏付けられた業界の標準的なセキュリティ対策と自社のプログラムを比較することができます。

BSIMM による評価の実施を検討しましょう。自社の現在の姿を確認し、他社のエクスペリエンスに学びつつ、自社プロジェクトを進化させることができます。



BSIMMについての詳細情報

ソフトウェア・セキュリティの 10 大測定指標

ソフトウェア・セキュリティの継続的な改善を実証できる 10 大測定指標（および追加の測定指標）は以下のとおりです。

1. 各エントリーの堅牢な特性とリスク・データを含む、ソフトウェア・インベントリが最新であること
2. 必要に応じてまたは定期的にすべてのアプリケーションをテストした割合
3. それぞれの種類とレベルのリスクベース・テストを、ゼロから軽量、そして詳細までの各アプローチで実施したアプリケーションの数
4. ソフトウェア・セキュリティ・ポリシーやコンプライアンス要件を満たすために必要になった変更の数
5. セキュリティ上の各種不具合を修正する時間
6. 製品版（あるいは本番システム）での、セキュリティ上のバグや設計上の欠陥の残存数
7. 開発か調達かを問わず、すべてのセキュアな SDLC ゲートを通過したソフトウェア・プロジェクトの割合
8. 開発者が脆弱性の修正以外の作業に回すことができたはずの時間
9. 要件から本番までの各ステージでのソフトウェアのセキュリティ問題に起因する遅延の頻度
10. コンプライアンス要件を満たすか超えているアプリケーションの数
11. 職務に対する適切なスキルレベルを保有するソフトウェア・セキュリティのステーク・ホルダーの数

検証

ポリシーと測定計画が準備できると、チームが SSI の活動を実行しているかどうか、SSI の要件を満たしているかどうか、および期待した効果を発揮しているかどうかを検証するためのチェック・ポイントを設定できます。

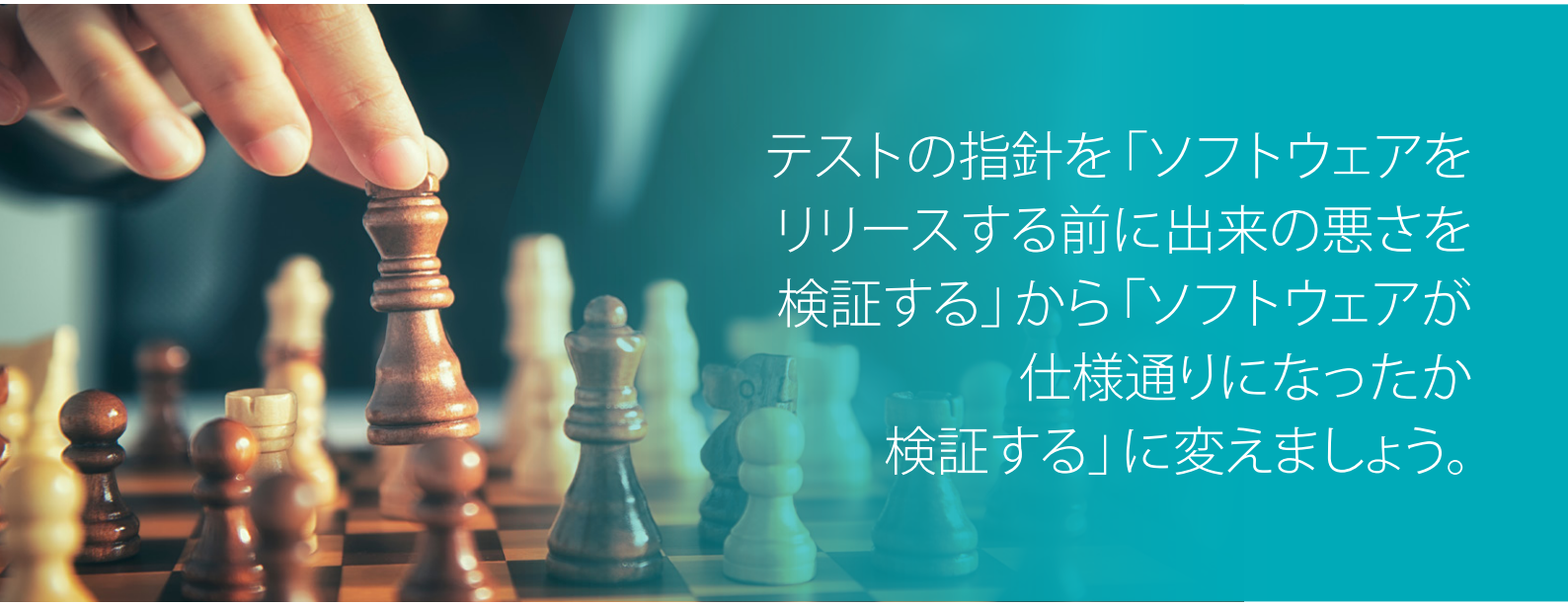
検証手順はこのように考えてください。自家用車は定期点検や車検に合格する必要がありますが、車のダッシュボードに「エンジン警告灯」が点灯したら、車を速やかに販売店に持ち込み、問題を発見するためのテストを実施します。

不具合の発見は、ちょうど自動車の「エンジン警告灯」の点灯と同じようなものだと言えます。ここで警告を発するのは、システムの問題にできる限り早く対処する必要がある場合です。

また、開発サイクルの終わりになってから複雑なセキュリティ・テスト計画を実施するのではなく、開発の早い段階で幾つかの小規模なテストを実行してください。つまり、テストの指針を「ソフトウェアをリリースする前に出来の悪さを検証する」から「ソフトウェアが仕様通りになったか検証する」に変えるということです。

これは、ウォーターフォール型開発を採用している企業の場合は、いくつかの段階（要件定義、アーキテクチャ設計、コーディング、QA）にテストを追加することを意味します。また、アジャイル手法を採用している企業の場合は、セキュリティをユーザー・ストーリーに組み込んで、開発者が問題の発見や修正を円滑にできるようにすることを意味します。

SSI が機能しているかどうかは、リリース前のセキュリティのステージがどんどん短くなることで把握できます。キャパシティを使い果たし、リリースを遅らせ、全員が心を痛める原因となる、セキュリティ問題の大量放出は無くなります。



外部の視点の重要性

評価と修正の作業を社内で実施している場合、時々外部の視点を取り入れることは良いやり方です。外部のテスト・パートナーから専門的な意見を得て、テスト結果が正確かどうか、基盤となるシステムが攻撃を防御できるかどうかを調べることができます。

外部のアプリケーション・テスト・ベンダーは、社内のツールが見逃す恐れのある脆弱性を発見するのに必要なツールとマニュアル・テストの戦略を備えています。そして、テスト結果をまとめて不審な点を確認し誤検知を排除できます。そしてなにより、外部ベンダーはテスト結果を解釈し、チームが発見するどのような問題でも修正できるよう支援します。

注意：これで終わりではありません。SSI の役目を不具合を発見することだと言うなら、決して改善は望めないでしょう。

改善

簡単にできる SSI の立ち上げ

SSI の立ち上げは一度で終わる活動ではありません。日常的な困難を伴う現場の環境で最初の SSI の枠組みの立ち上げに取り組む場合、改善の余地のある領域が見つかります。忘れてはならないのは、自分が利用可能なツールや技術が日々進歩するということは、攻撃側も同様に進歩しているということです。

1. **パターンに注意する。**検証プロセスで同じセキュリティ問題が絶えずあらわれる場合は、基準を見直すか、トレーニングを強化するか、より効果的なツールを開発者に提供するかのいずれかが必要になる場合があります。あるいは、脆弱性が基本的な設計上の欠陥に起因することが判明した場合は、設計チームにアーキテクチャの変更を求める必要があるでしょう。
2. **弾力的なキャパシティを準備する。**SSI は、ポートフォリオに含まれるアプリケーションの数や種類の変化、組織の変更、コンプライアンス要件の改訂、新しい攻撃経路などに対応できる、生きて呼吸をする様なプログラムでなければなりません。時折、アプリケーション・テストの需要が社内のキャパシティを必然的に上回る場合がありますが、適切なテスト・パートナーを見つけられれば、社内チームの負荷を軽減し、すべてのアプリケーションにわたって一貫したテストとセキュリティ対策を維持することができます。
3. **将来のロードマップを作成する。**SSI が完璧に機能しているとしましょう。つまり、すべてのソフトウェア・セキュリティ機能において専門技術を有しているとしましょう。その一部は次のとおりです。
    - リスクとコンプライアンス
    - オープンソース管理
    - ベンダー管理
    - セキュア・アーキテクチャ設計レビュー
    - アプリケーション・セキュリティ・テスト
    - ソースコード・レビュー
    - 不具合管理

すべての能力において同じ割合で専門性を身につける必要はありませんが、着実に進歩していることを確認し、SSI がビジネス上の優先事項として継続的に進化し続けるという期待を持つべきです。

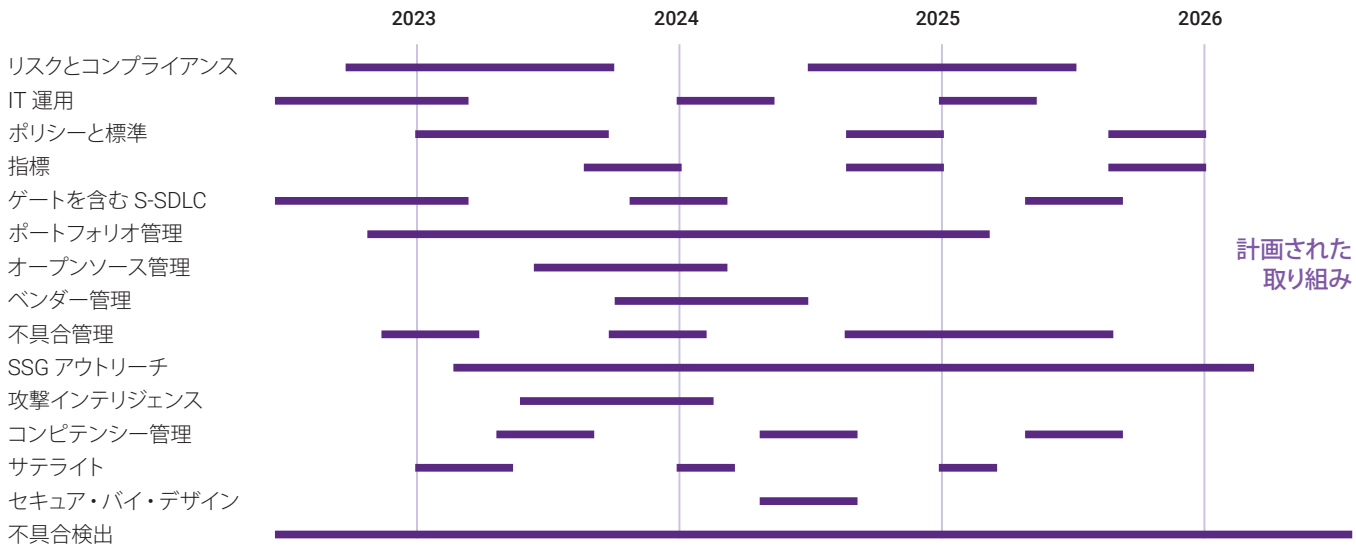
SSI の立ち上げは、一度で終わる活動ではありません。

パターンの探索とその対応の調整を継続する必要があります。

- トレーニングへの集中力を高める必要があるか？
- 新しいコーディング規約が必要か？
- より効果的な実施計画か？
- 実行中のテストを調整すべきか？

ロードマップのサンプル

SSI を成功させる次の、そして最後の手順は「管理」です。



管理

ここまで読んでお分かりのように、効果的な SSI には変動要素が数多くあり、多くの人や部門が関与します。舵取り役としてのあなたの仕事は、メンバー全員がコースを外れないようにすることです。

そして、組織を維持し、各セキュリティ対策に関する知見を得るためには、目標に応じた堅牢なプロジェクト管理システムを導入することです。

セキュリティ対策を、開発ライフサイクルのセキュリティ・ゲートや、ソフトウェアのリリースおよびアップグレードのスケジュールに合わせやすくしてくれるシステムを模索しましょう。一般的なプロジェクト管理ツールと比べて、セキュリティに特化したツールは時間を無駄にすることなく、SSI が管理対象とする重要な要素を漏らすことは決してありません。

適切なシステムを利用すれば、時間、アプリケーションの種類、事業単位、特定のプロジェクトを越えて比較がしやすくなります。一目で進捗を把握し、目標に対してどの程度遅れているかが分かり、注意すべき領域を特定できるようになります。

加えて、実用的なデータをタイムリーに会社の経営陣に報告できます。

セキュリティに特化したツールは時間を節約でき、
SSI が管理対象とする重要な要素を漏らすことは決してありません。

総括

すべての SSI には、親組織の構造と文化が反映されます。一元管理の企業もあれば、フェデレート（連合）管理の企業もあります。アウトソースしたリソースを利用する企業もあれば、新たにスタッフを雇用する企業もあります。マネージド・サービスを利用する企業もあれば、自前の技術チームを育てる企業もあります。

以下の 5 段階のプロセスによって成功への道が開かれます。つまり、開発サイクルにおけるすべてのステーク・ホルダー間の連携が強化され、事業目標に対する効果が実証され、長期的に構築した堅実なプログラムを手にすることができるのです。

SSI 早見表

1. 構築

- アプリケーション・ポートフォリオ、コンプライアンス要件、技術上および事業上のリスク分野に関する情報を収集。
- リーダーシップに裏打ちされたオーナーシップの責務とポリシーを含むガバナンス・ストラクチャを立ち上げる。
- 社内チームとサードパーティー・ベンダーの枠を越えて広くコミュニケーションする。
- 社内外の適切なリソースを動員し、SSI で定義された評価作業や修正作業を実行。
- スタッフがセキュリティのスキルを向上させる機会を設けて奨励。

2. 評価

- 基礎的なビジネス目標に紐付けた、継続的な進捗状況を示す測定値を決定。
- 自社のセキュリティ・プラクティスをセキュア開発成熟度モデル（BSIMM）と比較。

3. 検証

- 開発の最後だけでなく、開発プロセス全体を通して不具合を発見するためのチェック・ポイントを設定。
- 社内での結果を外部の分析と比較することで正確性を担保、誤検知を低減。

4. 改善

- パターンを割り出し、リソースの追加、トレーニング、継続的投資の対象領域を特定。
- ロードマップを作成し、ソフトウェア・セキュリティ能力における専門性を身に付ける。

5. 管理

- セキュリティに特化したプロジェクト管理ツールを準備し、SSI の管理と運営に役立てる。
- チームのパフォーマンスとアプリケーションの種類を分析して比較。
- 経営陣と組織全体のすべてのステーク・ホルダーと進捗を共有。

ブラック・ダックについて

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

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