DevSecOpsの6つの柱：

セキュリティ、開発、運用の統合による再帰的セキュリティの実現

SAFECode
Software Assurance Forum for Excellence in Code
Driving Security and Integrity

CSAJC cloud security alliance SM
Japan Chapter

CSA cloud security alliance®

クラウドセキュリティアライアンスのDevSecOpsの恒久的かつ公式な場所は、下記のURLを参照してください。
https://cloudsecurityalliance.org/group/DevSecOps/

謝辞

Lead Authors:

John Martin
Setumadhav Kulkarni
Ronald Tse
Michael Roza
Sean Heide

Contributors:

David Lewis
Eric Gauthier
Lee Szilagyi

CSA Staff:

Sean Heide

日本語版提供に際しての告知及び注意事項

本書「DevSecOpsの6つの柱：セキュリティ、開発、運用の統合による再帰的セキュリティの実現」は、Cloud Security Alliance (CSA)が公開している「The Six Pillars of DevSecOps: Achieving Reflexive Security Through Integration of Security, Development, and Operations」の日本語訳です。本書は、CSAジャパンが、CSAの許可を得て翻訳し、公開するものです。原文と日本語版の内容に相違があった場合には、原文が優先されます。
翻訳に際しては、原文の意味および意図するところを、極力正確に日本語で表すことを心がけていますが、翻訳の正確性および原文への忠実性について、CSAジャパンは何らの保証をするものではありません。
この翻訳版は予告なく変更される場合があります。以下の変更履歴（日付、バージョン、変更内容）をご確認ください。

変更履歴

日付	バージョン	変更内容
2023年11月14日	日本語版1.0	初版発行

CSAジャパン成果物の提供に際しての制限事項

日本クラウドセキュリティアライアンス（CSAジャパン）は、本書の提供に際し、以下のことをお断りし、またお願いします。以下の内容に同意いただけない場合、本書の閲覧および利用をお断りします。

１．　責任の限定

CSAジャパンおよび本書の執筆・作成・講義その他による提供に関わった主体は、本書に関して、以下のことに対する責任を負いません。また、以下のことに起因するいかなる直接・間接の損害に対しても、一切の対応、是正、支払、賠償の責めを負いません。

(1)　本書の内容の真正性、正確性、無誤謬性
(2)　本書の内容が第三者の権利に抵触しもしくは権利を侵害していないこと
(3)　本書の内容に基づいて行われた判断や行為がもたらす結果
(4)　本書で引用、参照、紹介された第三者の文献等の適切性、真正性、正確性、無誤謬性および他者権利の侵害の可能性

２．　二次譲渡の制限

本書は、利用者がもっぱら自らの用のために利用するものとし、第三者へのいかなる方法による提供も、行わないものとします。他者との共有が可能な場所に本書やそのコピーを置くこと、利用者以外のものに送付・送信・提供を行うことは禁止されます。また本書を、営利・非営利を問わず、事業活動の材料または資料として、そのまま直接利用することはお断りします。
ただし、以下の場合は本項の例外とします。

(1)　本書の一部を、著作物の利用における「引用」の形で引用すること。この場合、出典を明記してください。
(2)　本書を、企業、団体その他の組織が利用する場合は、その利用に必要な範囲内で、自組織内に限定して利用すること。
(3)　CSAジャパンの書面による許可を得て、事業活動に使用すること。この許可は、文書単位で得るものとします。
(4)　転載、再掲、複製の作成と配布等について、CSAジャパンの書面による許可・承認を得た場合。この許可・承認は、原則として文書単位で得るものとします。

３．　本書の適切な管理

(1)　本書を入手した者は、それを適切に管理し、第三者による不正アクセス、不正利用から保護するために必要かつ適切な措置を講じるものとします。
(2)　本書を入手し利用する企業、団体その他の組織は、本書の管理責任者を定め、この確認事項を順守させるものとします。また、当該責任者は、本書の電子ファイルを適切に管理し、その複製の散逸を防ぎ、指定された利用条件を遵守する（組織内の利用者に順守させることを含む）ようにしなければなりません。
(3)　本書をダウンロードした者は、CSAジャパンからの文書（電子メールを含む）による要求があった場合には、そのダウンロードしまたは複製した本書のファイルのすべてを消去し、削除し、再生や復元ができない状態にするものとします。この要求は理由によりまたは理由なく行われることがあり、この要求を受けた者は、それを拒否できないものとします。
(4)　本書を印刷した者は、CSAジャパンからの文書（電子メールを含む）による要求があった場合には、その印刷物のすべてについて、シュレッダーその他の方法により、再利用不可能な形で処分するものとします。

４．　原典がある場合の制限事項等

本書がCloud Security Alliance, Inc.の著作物等の翻訳である場合には、原典に明記された制限事項、免責事項は、英語その他の言語で表記されている場合も含め、すべてここに記載の制限事項に優先して適用されます。

５．　その他

その他、本書の利用等について本書の他の場所に記載された条件、制限事項および免責事項は、すべてここに記載の制限事項と並行して順守されるべきものとします。本書およびこの制限事項に記載のないことで、本書の利用に関して疑義が生じた場合は、CSAジャパンと利用者は誠意をもって話し合いの上、解決を図るものとします。

その他本件に関するお問合せは、info@cloudsecurityalliance.jp までお願いします。

日本語版作成に際しての謝辞

「DevSecOpsの6つの柱：セキュリティ、開発、運用の統合による再帰的セキュリティの実現」は、CSAジャパン会員の有志により行われました。作業は全て、個人の無償の貢献としての私的労力提供により行われました。なお、企業会員からの参加者の貢献には、会員企業としての貢献も与っていることを付記いたします。
以下に、翻訳に参加された方々の氏名を記します。（氏名あいうえお順・敬称略）

松浦 一郎, CISSP, CISM, CDPSE
諸角 昌宏

内容

はじめに

背景

クラウドコンピューティングの広範な導入は、前例のないセキュリティ上の課題を提起しています。組織は日々、侵害、漏洩、機密データ盗難といったニュース見出しに直面しています。これらの課題は、セキュアでないアプリケーション、設定ミス、問題のあるプロトコル、貧弱なインフラストラクチャー設計、および教育やトレーニングの不足などの影響によって発生します。

デジタルトランスフォーメーションが確実に進行する中、ソフトウェアはビジネスリスクとエクスプロイトの原因の上位に急浮上しています。アプリケーションの開発と配信の量とペースが急速に増加した結果、アプリケーションに対する攻撃の数と複雑さも増加しています。適切かつ十分なセキュリティスキルと資質を備えた人材の不足は、かつてないほど深刻になっています。

別の言い方をすれば、コンピューティングのユビキタス化とその複雑化によって、ソフトウェアシステムのアタックサーフェスは指数関数的に拡大しています。この状況を「ムーアの復讐」と表現する人もいます[1]。

複数の技術レイヤー間の複雑な相互作用は、今日のアプリケーションのグローバルな相互接続と常時オンの性質と結びつき、システム、ソフトウェア、およびハードウェアに潜んでいた潜在的な脆弱性によって、さらに悪化します。このような状況は、世界中の悪意をもつ組織が果実を収穫するためのうってつけの場を作り出しています。

脆弱性は、基盤となるインフラストラクチャー、サードパーティのサービス、ソフトウェア製品に組み込まれているライブラリなど、あらゆる場所で発生する可能性があり、ソフトウェア開発会社は現在のところ、このような課題を迅速に発見し解決する体制を整えていません。例えば、2017年の調査では、80%以上の組織が本番前の脆弱性スキャンを実施していないと回答しています[2]。

DevOpsのインパクト

DevOpsは、集団的コラボレーションなどの開発のベストプラクティスをインフラストラクチャー運用に適用するプラクティスであり、特にクラウド環境において、今日の開発および運用チームの効率にプラスな影響を与えることが示されています。

DevOpsの採用が強化されていることから、情報セキュリティ自体への影響を検討し、情報セキュリティ管理の分野に当該の慣行の適用を検討する必要があります。

DevOps、マイクロサービス、およびオープンソースのような、デプロイサイクルを加速する最新のトレンドとともに、コンピューティングデバイスとコンピューティングパワーの遍在化が進んでいるため、エクスプロイト可能な欠陥をタイムリーに検出・特定する能力に負担がかかり続け、全体的なセキュリティリスクの大幅な増加につながっています。

サーバーレスコンピューティング、APIファーストアプリケーション、マイクロサービスベースのアーキテクチャといったクラウドネイティブな開発パラダイムは、これらのパラダイムを実現するDevOpsの台頭とともに、主流の選択肢となっています。

このようなソフトウェア開発への新しいアプローチは、必ずしもセキュリティが熟考されたものではない設計原則のまま発展してきました。このようなパラダイムに持続可能な方法でセキュリティを導入するために利用できる唯一の現実的な選択肢は、このようなパラダイムを現実のものとしたコンセプトそのものであるDevOpsに依拠することです。

セキュリティをDevOpsに直接組み込むことで、DevSecOpsのような最新の統合セキュリティパラダイムを通じて既存のセキュリティを開発や運用プロセスに統合し、成果を大幅に改善できます。例えば、DevSecOpsを導入している組織は、最新のマイクロサービスベースのアプリケーションのセキュリティに関する結果が、はるかに優れていることに気づいています。さらに、セキュリティがソフトウェア開発ライフサイクルに組み込まれていれば、組織は、開発の初日であっても、セキュリティに関連するパフォーマンスをコントロールできることに気づいています。

本目標は、ソフトウェアの開発と公開における複雑さを軽減し、既知の信頼できるコンポーネントとサービスのみを使用するようにし、開発手法に直接統合されたセキュリティリソース（自動化されたものと人の手によるものの両方）をソフトウェア開発チームに提供し、セキュリティが十分に確保され、監視された開発環境を使用し、最終的に、設計どおりに、設計された機能のみを備えた最終製品を提供することです。

スコープ

この文書は、DevSecOps を実装し組織に統合するために不可欠な、DevSecOps の6つの重点分野を定義しています。

この文書で提供されるDevSecOpsの柱は、従来はサイロ化されていた開発、インフラストラクチャー運用、および情報セキュリティの各業務を、セキュアなソフトウェアの作成を促進する結束力のあるグループへと融合させる、全体的なフレームワークの提供を意図しています。

引用文書

Information Security Management through Reflexive Security, CSA

ISO/IEC 27000:2018, Information technology -- Security techniques -- Information security management systems -- Overview and vocabulary

用語と定義

この文書の目的のため、情報セキュリティ管理に関する用語と定義は、再帰的セキュリティ、ISO27000、およびCSAで示される用語と定義が適用されます。

クラウドファーストの課題解決のためのDevSecOps

DevSecOpsは、インフラストラクチャーと製品のライフサイクルがますます短くなっている、今日における相互接続され急速に変化するセキュリティ環境の課題に対処するアプローチのひとつです。
DevSecOpsのプラクティスは、クラウドファーストのソフトウェアから生じた課題を解決するための対応策として作成されました。これは、DevOps の文化、プラクティスおよびワークフローに、継続的なセキュリティの原則、プロセスおよび技術を統合することと、簡潔に定義できます。

DevSecOpsはまた、従来はサイロ化されていた開発、インフラストラクチャー運用、情報セキュリティの各業務を統合し、コントロールされたプロセスの下でセキュアなソフトウェアの開発を促進する、全体的なフレームワークを提供することも意図しています。

本CSA DevSecOps ワーキンググループは、再帰的セキュリティフレームワークで説明されている6つの柱に従って、DevSecOps を組織に統合するために重要な以下の6つの重点領域を定義しました。

柱1：集団的責任

DevOpsにセキュリティを組み込むための最大の課題の1つは、組織のマインドセット、ソフトウェアセキュリティに関する考え方、習慣および行動を変えることです。

DevOps組織にセキュリティが導入されることで、セキュリティはもはや他人事ではなくなります。それは、他の人たちの仕事が一段落してから取り組むような、後回しにされるものであってはなりません。さらに、セキュリティをビジネス目標と切り離して考えることはできません。最後に、セキュリティは進歩や貢献を計測できない刹那的なものではありません。

誰もが、組織のセキュリティ姿勢に責任を負います。CSO （クラウドセキュリティオフィサー） は、組織における情報セキュリティの指導的・監督的役割を果たしますが、各人がセキュリティの責任を負い、組織のセキュリティ姿勢対する自らの貢献を意識付けする必要があります。エッジのユーザーと開発者は、単に「セキュリティを意識している」というだけでなく、防御の第一線にいます。

これは、「再帰的セキュリティ」のフレームワークの柱である「責任ある集団」に相当します。

柱2：コラボレーションと統合

開発、運用、セキュリティの各分野において、ソフトウェア業界には膨大なスキル （知識） と人材 （リソース） のギャップがあります。セキュリティの導入に関して組織全体が協力しなければ、成功は限られたものになります。セキュリティは、対立ではなく協力によってのみ達成できます。すべての機能チームのメンバーが潜在的な異常を報告できるようにするためには、セキュリティを意識した協力的な文化が必要です。人的要因はしばしば最も弱いリンクであり、実際に、セキュリティインシデントのほとんどは、単純なヒューマンエラーによって引き起こされることを覚える必要があります。

これは、「再帰的セキュリティ」のフレームワークの柱である「協力と統合する」に相当します。

柱3：実践的な実施

DevSecOpsでは、ポイントソリューションが数多く提供されています。組織は、ソフトウェアライフサイクルの中でアプリケー

ションセキュリティを実装するために、さまざまなツールやソリューションを選択できます。ソフトウェアライフサイクルは、構造、プロセス、全体的な成熟度においてそれぞれ異なるため、DevSecOpsを実装するための万能なツールは存在しません。組織はしばしば、導入が難しく、運用が困難で、結局は真のセキュリティリスクの軽減に役立つ実用的な洞察を提供しないツールやポイントソリューションの調達に終始します。

組織は、ソフトウェアのライフサイクル、組織自身のセキュリティのニーズ、そして求める将来の状態を総合的に捉え、高度な統合性を提供するプラットフォームソリューションを選択する必要があります。

デジタル社会における安全、プライバシー、および信頼を確保するために、アプリケーション開発に焦点を当てたフレームワークにとらわれない「デジタルセキュリティとプライバシーモデル」を使用することで、組織はDevOpsにおけるセキュリティに実用的な方法でアプローチできます。このモデルは、すべての利害関係者（開発、運用、セキュリティ）を、セキュリティが、アプリケーションとアプリケーションを生み出すソフトウェアライフサイクルに組み込まれる形で組み込まれるという、現状では満たされていないニーズを満たすものです。

これは、「再帰的セキュリティ」の枠組みにおける「実践的」な柱に相当します。

柱4：コンプライアンスと開発の橋渡し

リスクに関連した要求事項を、時間の経過に従って容易に測定できるセキュリティ要求事項に変換することは難しいものです。セキュリティチームがリスクベースの手法をサポートするために要件を作成するとしても、コンプライアンス要件はDevOpsや製品要件にうまく変換されていません。逆に、技術的なコントロールが実装されていても、セキュリティ要件が満たされているという証拠を得ることは容易ではありません。

ソフトウェア開発のパラダイムとプラクティスの急速な進化を考えると、コンプライアンスとアジャイルソフトウェア開発はもはや同じ立場に置けません。規制・コンプライアンス部門は、その各プロセスの実行を実際に監査することよりも、プロセスが存在することを証明することに関心があります。一方で、ほとんどのDevOpsチームは、証明はコードの中にあり、プロセスの文書化の中にはないと考えています。

コンプライアンスと開発の間のこのギャップに対処する鍵は、適用可能なコントロールを特定し、それを適切なソフトウェア対策に変換し、ソフトウェアライフサイクル内の変曲点を特定し、そこでこれらのコントロールを自動化して測定することにより、リスク軽減の質を向上させ、その結果、コンプライアンスを改善することです。

これは、「再帰的セキュリティ」のフレームワークの「整列と橋渡し」の柱に相当します。

柱5：自動化

最小限のコストで迅速にセキュアな配備を実現するためのソフトウェア開発手法を妨げている課題のいくつかは、手作業による行き当たりばったりのコーディング、テスト、デプロイ、およびパッチの適用です。

自動化された品質チェックがなければ、手作業によるコーディングは容易に、手直しが必要なパフォーマンスの低い、セキュアでないソフトウェアになります。さらに、手作業やタイミングの悪いテストでは、デプロイ前に脆弱性が特定される可能性を低くします。手作業によるデプロイとパッチの適用は、セキュアでないソフトウェアを本番環境にリリースする可能性があります。

自動化されたセキュリティ対策は、手作業によるプロセスを減らし、効率を高め、手戻りを減らすことができるため、プロセス効率の中核となります。ソフトウェアの品質は、テスト／フィードバックの徹底性、適時性、および頻度を改善することによって向上できます。自動化できるプロセスは自動化し、自動化できないプロセスは可能な限り自動化するか、廃止を検討すべきです。自動化されたセキュリティチェックは、ビルドの遅延や失敗といった新たな課題を引き起こす可能性がありますが、これらは通常、ワークフローの改善や半自動化アプローチで対処できます。ソフトウェア開発には、「同じことを3回やったら、そろそろプログラミングの時期だ」という格言があり、これは、再帰的セキュリティに正面から当てはまります。

これは、「再帰的セキュリティ」の枠組みにおける「自動化」の柱に相当します。

柱6：測定、監視、報告および行動

「測定できない（あるいは測定しない）ものは管理できない」という格言は、DevSecOpsの実装と保守ほど当てはまるものはありません。一般的なDevSecOpsイニシアチブは、スコープと複雑さにもよりますが、実装に数ヶ月から数年を要します。実行可能なメトリクスがなければ、進捗を測定することはできず、失敗をタイムリーに発見することもできません。

DevSecOps環境で監視すべき最も重要なメトリクスには、デプロイメントの頻度、脆弱性パッチの適用時間、自動的にテストされるコードの割合、アプリケーションごとの自動テストがあります。DevSecOpsを成功させるためには、ソフトウェア開発中だけでなく、デリバリー後の結果も、適切な人々によって適切なタイミングで （継続的に） 測定、監視、報告、および対処される必要があります。

これは、再帰的セキュリティのフレームワークの「測定と改善」の柱に相当します。

概要

CSA DevSecOps ワーキンググループは、本文書に記載された重点領域は、DevSecOps の文脈におけるセキュアソフトウェア開発の弱点に対処することが可能であり、適切に実装された DevSecOps 環境を将来的にダイナミックに構築するためのビルディングブロックとして機能すると結論付けています。

それぞれの柱については、今後、別のホワイトペーパーで詳しく取り上げる予定です。

参考文献

[1] ‘Moore’s Revenge’ is upon us and will make the world weird, Mark Pesce, The Register. https://www.theregister.co.uk/2018/06/04/moores_revenge/

[2] 2017 State of Application Security: Balancing Speed and Risk, Jim Bird, SANS. Available at: https://www.tenable.com/whitepapers/research-report-sans-2017-state-of-application-security-balancing-speed-and-risk