サイバーセキュリティ戦略には「シフトレフト」と「シールドライト」の両アプローチを含める必要がある

クラウドコンピューティング、コンテナ、Infrastructure as Code（IaC）などの最新テクノロジーを採用している企業は、競争力の強化と設備投資の削減を実現しています。しかし、デジタルフットプリントを拡大しようとするあまり、セキュリティ上のギャップが生じることもあります。アプリケーションセキュリティは、もはや1部門だけの責任ではありません。

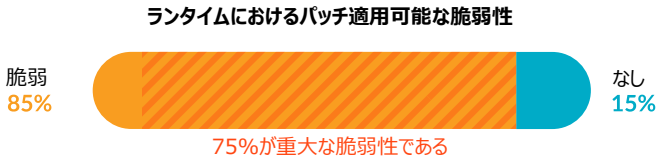
もちろん、多くの場合、これは「言うは易く行うは難し」の典型例です。クラウドネイティブアプローチは、開発者のスピードとアジリティを向上させますが、ある調査によれば、クラウドエンジニアとセキュリティの専門家の41%が、アジャイル手法はより多くの複雑さをもたらすため、クラウドセキュリティの取り組みに大きな影響を及ぼすと回答しています。また、この調査では、クラウドネイティブアプリケーションを構築している企業の45%が、既知の脆弱性に起因するインシデントに見舞われていることがわかりました。

DevOpsチームが使用するアジャイル手法は、従来のセキュリティアプローチを維持できない状態にします。開発のペースが速いため、セキュリティテストをソフトウェア開発サイクルの最後に配置する従来のウォーターフォール型アプローチに割く時間はほとんどなくなります。多くの組織は、本番環境におけるセキュリティ脆弱性を探すランタイムソリューションから始めています。これは、このようなソリューションが、本質的には飛行中にエンジンを構築するようなものであるにもかかわらず、実装と運用がより容易であると認識されているためです。

リリース前にセキュリティ問題を解消するため、チームは、クラウドネイティブセキュリティに関する専門知識が必要です。追加のトレーニングや教育を設定し、セキュリティのレビュープロセスやツールを設計や開発の初期段階に移行して、セキュリティにおける「シフトレフト」を実現する必要もあります。問題は、開発者がセキュリティの専門家ではなく、ビジネス機能の提供に集中しなければならないことです。オープンソースソフトウェアが開発現場であたり前の存在になったことも背景となり、開発者は自動化されたソフトウェアコンポジション解析（SCA）を求めています。Linux FoundationとSnykによれば、今日、モダンなソフトウェアアプリケーションの70%から90%にオープンソースソフトウェアが含まれているとのことです。

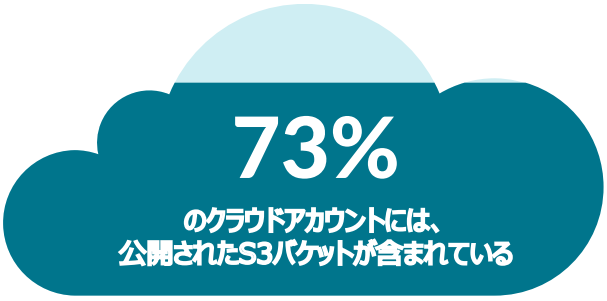
しかし、すべての問題をリリース前に解決できるわけではありません。全くの新しい問題や未知の問題についてはテストが行えないからです。問題を発見してから、それが最終的に修正されるまでに時間がかかることもよくあります。また、コードは第三者が所有している場合があり、これは発見された問題を修正する責任が誰にあるのかという懸念を生じさせます。ソフトウェアの実行中に攻撃を阻止または軽減するような「シールドライト」型のセキュリティアプローチが、デジタルフォレンジックとインシデントレスポンス（DFIR）を可能にするためにも不可欠となります。ランタイムセキュリティは、すべての情報セキュリティおよびサイバーセキュリティプログラムの基礎となるものです。

DevOpsチームは、人材、プロセス、テクノロジーを結集することで、セキュリティの永続的なサイクルを作り出し、デジタル資産の将来を保証する必要があります。サイバーセキュリティ戦略を再検討する上で、今ほど重要な時期はありません。DevOpsチームは、より迅速なイノベーションに向けて準備中ですが、これは、サプライチェーン、経済、世界の混乱が続いている間でも同じことです。また、イノベーションのスピードが上がれば、脆弱性のバックログも増えることになります。ランサムウェア攻撃のリスクは、セキュリティリーダーにとって高いランクに位置しており、これらの攻撃は既知の脆弱性を悪用することで永続化します。Gartner社では、2025年までに少なくとも75%のIT組織がランサムウェア攻撃に直面することになると予測しています。



出典： Sysdig 2022 Cloud-Native Security and Usage Report

約75%のコンテナは、パッチを適用可能な重大な脆弱性が存在する状態で稼働しています。また、73%のクラウドアカウントがS3バケットを公開しており、機密データを不要なリスクにさらしています。



出典： Sysdig 2022 Cloud-Native Security and Usage Report

ライフサイクル全体のセキュリティを確保するためには、シフトレフトとシールドライトの両アプローチをバランスよく取り入れることを、すべての**セキュリティプログラム**の目標とすべきです。その理由を下記に示します。

シフトレフトアプローチとは何か、また、それがないと何が起きるのか？

たった1行のコードの不備が、プロジェクト全体に影響を及ぼすことがあります。セキュリティの脆弱性が1つでもあれば、それと同じことが言えます。シフトレフトアプローチは、開発プロセスの早い段階でこれらのセキュリティ問題に取り組み、リリース前に問題の発生源でそれらを特定、管理、除去できるようにします。伝統的に、シフトレフトアプローチとは次のようなものです。

アプリケーションセキュリティテストは、コードから始まります。ソフトウェア構成分析（SCA）と静的アプリケーションセキュリティテスト（SAST）のためのコードセキュリティツールは、コードと依存関係を分析して、開発の初期段階で問題を発見するのに役立ちます。Linux FoundationとSnykの最近の調査によると、これらの2つのツールは、セキュリティの懸念に対処するために使用されるトップ2のツールとなっています。動的コード解析も同様に重要な役割を果たします。これは、コードを実行し、その結果を検証するものであり、可能性のあるコードの実行パスのテストを含みます。

コードが本番環境に移された場合、実行時に発見された問題点から基礎となるコードへのフィードバックループが必要となります。ランタイムセキュリティから情報を得た対話型分析ツールにより、チームは脆弱性にリアルタイムで対応し、変更を加え、指示を与えることが可能となります。当然ながら、DevOpsチームは、継続的統合／継続的開発（CI/CD）ビルドの一部として、完全な自動化が可能なスキャニングを好みます。

開発者フレンドリーなエクスペリエンスと実用的な修正ガイダンスを提供するツールがいくつか登場しています。これらのソリューションは、DevSecOpsプラクティスにとって基本的なものであり、早期に自動テストを実行し、開発者のワークフローや開発ペースに合わせた形で開発者に結果を提示します。開発段階におけるこうした包括的なテストプロセスは、スムーズな本番リリースに向けた強固な基盤を提供しますが、クラウドネイティブ環境では、セキュリティの複雑性が何層にもわたって追加されます。

まず、セキュリティテストの自動化は、テストデータ管理に関するプロセスや、Seleniumスクリプトのようなオープンソースのテスト自動化ツールの運用に関して、困難が伴います。また、テスト環境が実際の本番環境を十分に反映していることはほとんどなく、意図しない副作用によりテスト結果が無効になる可能性もあります。

また、セキュリティチームは、使用するコードソースの数が膨大であること、導入するバージョン追跡および管理ツール、戦略的に選択しなければならない統合ポイントなどが理由で、すべてのコードと潜在的な脆弱性に関する可視性を確保できないことがよくあります。

DevOpsチームは、サードパーティやオープンソースのソフトウェアを使用することがよくありますが、コードそのものを管理していないため、修正が彼らの責任範囲外になってしまうことがあります。さらに、多くのコードは互いに依存しており、これらの依存関係の連鎖は多くの場合入れ子状の複雑な構造になっています。このように多くの推移的な依存関係があるため、脆弱性を持つコードのコンポーネントが少なくとも1つ存在すると思われる場合でも、それがランタイムに実行されるか、悪用可能かどうかは別の課題となります。

また、アプリケーションコード、IaC（Infrastructure-as-Code）、PaC（Policy-as-Code）のように、処理すべきコードのアーティファクトタイプが複数存在しているため、それぞれに脆弱性やコンプライアンスを効果的に監査するためのルールが必要となります。さらに、各コードタイプは、複数のCI/CDパイプラインにつながります。開発チームが、これらのコードすべてを効果的にレビューするためのスキルセットや、正式なソフトウェア開発ライフサイクルプロセスを備えているとは限りません。

すべてのセキュリティ問題がコードレベルに起因しているわけでもありません。設計全体、アプリケーションソース、インフラストラクチャーの構成、セキュリティ対策の緩和などに起因する場合もあり、これらはスキャンでは容易に特定できません。スキャナの有効性は、ソース言語とアーティファクトタイプによって異なります。ほとんどのセキュリティスキャンツールは、セキュリティペルソナを対象としています（後述するように、開発者はセキュリティ問題の専門家ではありません）。また、スキャンツールの中には、実用的な詳細情報の提供や、修正の自動化を行わないものもあります。

コード解析の課題

アプリケーションを実行しないコードの静的解析は、より多くの潜在的な発見をもたらす可能性があり、それらは時に誤検知としてチームにより無視されるか、または抑制されます。また、静的解析によっては、完全に統合されたシステムの全体像を描くことができません。

コードの動的解析（ソフトウェアの実行中に行われる）は、API中心のアーキテクチャーや、モバイルを含む複数のフロントエンドを持つアーキテクチャーには適していないことがよくあります。なぜなら、そのようなアーキテクチャーでは、良好なテストカバレッジを達成することが困難であるためです。コードのすべての側面に到達するように機能を完全にテストすることは、理論的には簡単ですが、実際には困難です。また、動的解析は、ロジックの欠陥を検知することが苦手です。この種の欠陥が発生するコードの条件を正確に特定することは困難だからです。動的解析を使いこなすのは、静的解析と比較してより複雑です。なぜなら、動的解析では、より良い結果を取得し、できるだけ多くのコードカバレッジを達成するために、ユーザー認証、権利認証、およびアプリケーションへの十分なデータ供給を行う必要があるからです。

開発者は、単に脆弱性を並べた長いリストを提供するだけではないSASTおよび SCAツールを必要としています。それらのツールは、問題を修正する方法に関する実用的なアドバイスを提供する必要があります。

インタラクティブな解析を行うには、ある種のインスツルメンテーションエージェントを適切に機能させることが必要となります。これには、アプリケーションランタイムエージェント、コンテナランタイムエージェント、Webプロキシなどが含まれます。

大量の調査結果から相対的なセキュリティリスクを判断することは、どの分野の専門家にとっても難しいことですが、機能を提供することを主な業務としている開発者にとってはなおさらです。開発者は、調査結果のトリアージや修正の優先順位付けの専門家ではありません。また、問題をいかにして修正すればよいのかさえ分からないという課題もあります。多くのツールは、実用的な修正アドバイスを提供することはなく、脆弱性を並べた長いリストを提供するだけです。セキュリティテストが効率的でなく、スキャナの出力にしきい値が設定されていない場合、リリース速度が低下し、その結果、組織のビジネス目標達成能力に直接影響を与える可能性があります。開発者は、単に脆弱性を並べた長いリストを提供するだけではないSASTおよびSCAツールを必要としています。それらのツールは、問題を修正する方法に関する実用的なアドバイスを提供する必要があります。

完璧に設計、開発、配備されたランタイムシステムであっても、攻撃を受ける可能性はあります。組織は、あらゆる種類の初期段階テストの範囲には入らない、その他の多くの脅威（ランサムウェア、悪意あるクリプトマイニング、ランタイムのハッキングなど）に直面しています。これが、シフトレフト型のセキュリティとシールドライト型のアプローチのバランスをとることが不可欠である理由です。

シールドライトアプローチとは何か、また、それがないと何が起きるのか？

シールドライトアプローチとは、実行中のサービスを保護・監視するためのセキュリティの仕組みに重点を置いたアプローチです。セキュリティ従事者は、このアプローチを、注力する分野に応じて、ランタイムセキュリティ、ランタイムプロテクション、またはランタイム脅威の検出と対応と表現することがよくあります。このようなランタイム能力は、NISTのサイバーセキュリティフレームワーク（CSF）などのガイダンスに見られるように、現代のサイバーセキュリティプログラムの基礎となるものです。組織は、自らのリスクプロファイルを理解するために、すべてのシステムとコードの問題を完全に特定する必要がありますが、セキュリティテスト機能はその助けとなります。また、CSFの他の分野である保護、検知、対応、回復にも焦点を当てる必要があります。

ランタイムセキュリティのアプローチには、さまざまな形態があります。アプリケーションとそれを支えるインフラに関して、セキュリティチームは伝統的に、侵入防止システム（IPS）、ファイアウォール、次世代ファイアウォール（NGFW）、Webアプリケーションファイアウォール（WAF）などに依存しています。これらのツールは、ホスト、ネットワーク、アプリケーションの保護に重点を置くものであり、ワークロード（またはコンテナ）のコンテキストにはあまり重きを置いていません。

コンテナやサーバーレス技術を含むクラウドネイティブ設計を活用している組織は、これらの抽象化された「エフェメラル」なコンピューティングパターンをサポートする最新のセキュリティツールや、PaaS（Platform-as-a-Service）などの新しいクラウドホスティングモデルをサポートするツールを必要としています。セキュリティツールは、クラウドセキュリティポスチャー管理（CSPM）やクラウドインフラストラクチャーエンタイト

米国国立標準技術研究所（NIST）は、企業や組織がサイバーセキュリティのリスクをよりよく理解し、管理し、低減できるようにするために、サイバーセキュリティに関するフレームワークを提供しています。

このフレームワークは、下記に示す5つの同時かつ連続的な機能により構成されています。

- **特定（Identify）：**クリティカルなビジネスリソースと関連するセキュリティリスクをマッピングすることで、取り組みの焦点と優先順位を決定します。
- **保護（Protect）：**サイバーセキュリティイベントがクリティカルなビジネスサービスに及ぼす影響を抑制するための保護手段を実装します。
- **検知（Detect）：**継続的な監視と検知を実現し、異常やイベントのタイムリーな検知を促進します。
- **対応（Respond）：**サイバーセキュリティインシデントの影響を抑制するためのアクションを即座に実施できることを確実にします。
- **回復（Recover）：**セキュリティイベントの影響を軽減するためのサービス復旧計画を策定し維持します。

これらの5つの機能により、さまざまな分野の専門家がセキュリティライフサイクルに参加できるようになります。

sysdig snyk

ルメント管理（CIEM）に分類されるエージェントレス機能により、クラウド制御プレーンの監査と監視を提供する必要があります。これらの機能は、それぞれクラウドの設定ミスの検証、およびクラウド環境について誤って権限を設定されたリソースの検証を支援するものです。セキュリティツールは、ワークロード、コンテナランタイム、およびオーケストレーションエンジンのインストルメンテーションも提供する必要があり、これらはしばしば、クラウドワークロード保護プラットフォーム（CWPP）またはクラウドネイティブアプリケーション保護プラットフォーム（CNAPP）と呼ばれます。

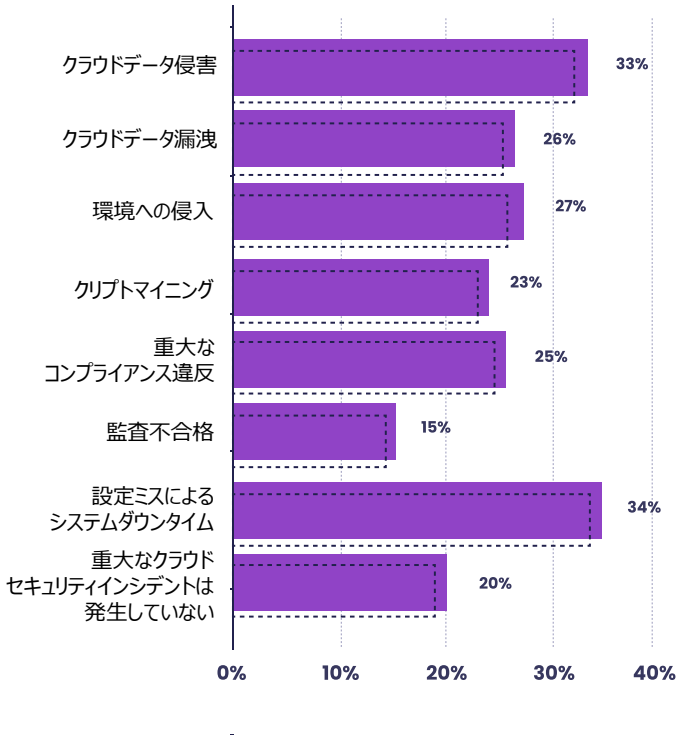
理想的には、これらの機能は、クラウドおよびクラウドネイティブ環境における最新の脅威の検知と対応をサポートするために、環境間でテレメトリを収集し相互に関連付けるような統一されたエンジンも提供します。このようなランタイム脅威の検知と対応は、多くの場合、クラウドログの取り込みとリアルタイム分析によって行われます。クラウドログ内の情報は、その他のサービスやワークロードのアクティビティと相互に関連付けられます。その結果、組織のアタックサーフェスをより正確に把握し、脅威が運用環境にどのような影響を及ぼしているかを把握できるようになります。これは、クラウドの検知と対応（CDR）として最もよく表現される、より新しいタイプの機能です。

しかし、現実には、本番環境では、最良のセキュリティ対策が通用しないような問題が発生します。だからこそ、実行時のセキュリティと開発段階のセキュリティプロセスのバランスを取ることが不可欠となるのです。広大な混在環境、クラウドホスティングモデル、および多数のワークロードタイプはいずれも、セキュリティ管理の実装と運用を複雑にしています。

コンテナ化されたアプリケーションでは、ワークロードに関する可視性が悪化します。コンテナでは、セキュリティ管理の失敗につながる死角がしばしば発生します。特に、コンテナのコンテキストに合わせて構築されていないツールでは、セキュリティインシデントや違反が発生した際にアラートが発報されないことがあります。また、コンプライアンス上の問題やサービスの中断が発生する可能性もあり、これらのいずれもがパフォーマンスの低下やダウンタイムを引き起こす可能性があります。

最近の設計では、エフェメラルなワークロードや環境が一般的になっています。これらのリソースは短時間しか稼働しないため、イベントの保持とログ解析が問題となります。このような現実は、フォレンジック調査やインシデント対応も複雑にします。診断のための重要なデータは、残すためのプランを策定していなければ消えてしまう可能性があり、実行中に何が起こったかの痕跡を残せません。

経験したことのある重大なクラウドセキュリティインシデント



出典：The 2022 State of Cloud Security Report, Snyk

また、ランタイムの問題を元のインフラストラクチャーの構成へと遡ることも問題となります。IaCの解析は、企業や組織がインフラストラクチャーの自動化手法を採用しており、その結果としてIaCアーティファクトを生成している場合には役立ちます。しかし、アプリケーション全体とそれをサポートするインフラストラクチャーをインスタンス化するために使用されるIaCアーティファクトが複数存在している場合、それらのすべてが組織レベルのクラウド構成と設定により上書きされる可能性があります。

テクノロジースタックの各層（アプリケーション層、ワークロード層、ランタイム層、ネットワーク層）において、適切なランタイムセキュリティ機能を利用する必要があります。オープンシステム相互接続（OSI）は、よく理解されているメンタルモデルですが、それは最新のアーキテクチャーとセキュリティ制御に完全に適合するわけではありません。

DevOpsチームと同様に、SecOpsチームも大量のアラートを受け取ります。既知の脆弱なライブラリの利用、不適切なコーディング手法、設定ミスなどのような容易に解決できる問題は、防止できたはずの問題を追いかけているセキュリティチームにとって無駄な仕事を増やすことになります。大量のデータが組織のセキュリティ情報およびイベント管理（SIEM）に押し寄せると、脅威の効果的な検知と対応が阻害されます。また、SecOpsの作業をマネージドセキュリティサービスプロバイダー（MSSP）やマネージドディテクション＆レスポンス（MDR）ベンダーに委託した場合、費用が増加したり、イベントを迅速に検知できなかったりすることが予想されます。

ギャップに注意すること

また、情報のギャップも問題となります。ログデータは十分な期間保持されない可能性があるため、その結果、問題を把握できなくなることがあります。クラウドプロバイダーは、特定のテレメトリや計測のAPIを公開していない場合があるため、さらに情報のギャップが拡大します。

また、多くの組織がスキルギャップに直面しています。最新のアーキテクチャー向けにDFIRを実現することは複雑であるため、SecOpsアナリストの専門知識はアプリケーション、コンテナ、サーバーレス機能に限定される場合があります。さらにセキュリティオペレーションセンター（SOC）が存在しないか、分散しているか、またはMSSPにアウトソーシングされている可能性もあります。

セキュリティツールは、組織内のさまざまなペルソナや問題への対応方法を考慮して構築することが可能であり、またそうする必要があります。

シールドライト型のセキュリティでは、実行時に問題が検出された場合、コードにおける根本的な問題に対処するため、または適切なセキュリティ緩和策を設定するために、特定分野の専門家が正当な問題を検証する必要があります。従来は、何が起きたのか、どのように修正するのか、誰が責任を負うのかを決定するために、セキュリティ以外の役割とセキュリティの役割を担う多数の人たちが協力する必要がありました。セキュリティツールは、組織内のさまざまなペルソナや問題への対応方法を考慮して構築することが可能であり、またそうする必要があります。これには、個人の役割とアプリケーション環境に合わせた修正、実際のリスクに基づいた優先順位付け、セキュリティの世界とセキュリティ以外の世界をつなぐ役割を果たすワークフロー統合などの機能が含まれます。

効果的なサイバーセキュリティプログラムには、両方のアプローチが必要となる

プロセスやツールにシフトレフトおよびシールドライトの両アプローチを採用することで、DevSecOpsとも呼ばれる、セキュリティとエンパワーメントに関する永続的なサイクルがもたらされます。

シールドライトアプローチ（ランタイムセキュリティ）は、企業や組織がセキュリティギャップの存在を認識している場合、「出血を止める」のに役立ちます。このようなギャップは、多くの場合、急速に変化する、複雑で分散型のエフェメラルな環境の結果として生じます。ランタイムセキュリティには、防御を回避したり、テストを行っても環境に忍び込んだりする問題に対する検知、防御、および対応を行う機能が含まれます。

静的なセキュリティテストを行う場合、リスクの優先順位付けと、何が本当に実行され、何が悪用可能となるかを理解するために、ランタイムのインテリジェンスを通じて情報を提供する必要があります。セキュリティツールは、クラウドの脆弱性がどのリージョン、クラスター、ネームスペースに存在するかというメタデータを含む適切なコンテキストを提供する必要があるほか、クラウド環境とオンプレミス環境に存在するすべてのワークロードタイプにおいて動作する必要があります。

また、ランタイムに発見された問題をエンジニアリングワークフローに組み込むことで、対応と修正を迅速化する必要もあります。企業や組織は、セキュリティ監視ツールを通じてアラートを発行することで、SecOpsチームに通知し、セキュリティリスクを追跡する必要があります。また、欠陥の追跡を開始し、DevOpsへのフィードバックループを提供する必要もあります。セキュリティが理由で、価値を提供するためのエンジニアリングワークフローを中断することや妨げることがあってはなりません。

結論：セキュリティの永続的なサイクル

サイバーセキュリティプログラムには、セキュリティに対するシフトレフトとシールドライトの両アプローチが必要です。これは、完全なライフサイクルセキュリティを実現するためのDevSecOpsが必要であることを意味します。

アプリケーション開発、インフラストラクチャーエンジニアリング、および運用は、DevOpsプラクティスの副産物として、密接に結び付くようになりました。これと同様に、セキュリティを、コーディング時の初期段階から、DevOpsのプラクティスとツールチェーンに組み込む必要があります。

これらの2つのアプローチをともに採用することで、企業や組織は、クラウドやクラウドネイティブアーキテクチャーにおけるセキュリティインシデントを迅速に検知して対応できるようになります。これらは、最新のサイバーセキュリティプログラムの基盤となるものです。

sysdig snyk