CyberRes

ホワイトペーパー

2022 年アプリケーション セキュリティトレンドレポート

デジタルイノベーションの加速というビジネス要件とソフトウェアセキュリティリスクとのバランスを取る必要があることを組織が認識するにつれ、アプリケーションセキュリティ業界は着実な進化を続けています。

目次

アプリケーションセキュリティの主なトレンド

全コードベースの 98% が、オープンソースコンポーネントに依存しています。

アプリケーションセキュリティは、クラウド主導型の時代に進むにつれて、シフトレフトから「シフトエブリウェア」へと進化し続けています。2022 年のリストは、大きなトレンドと興味深い新しいイノベーションを組み合わせたものになっています。あなたならこのリストに何を追加するでしょうか。

はじめに

年次トレンドリストに予想外の項目が登場することはほとんどありません。最終的に、ほとんどのトレンドは、1 か月前あるいは 1 年前から重要だったことの続きです。Fortify では、卓越した基本要素に基づく総合的な AppSec ビジョンを掲げています。たとえば、幅広い正確な言語対応、AppSec をお客様の既存のツールに円滑に統合するためのエコシステム、SaaS やオンプレミス間のハイブリッドに対応したエンドツーエンドのアプリケーションセキュリティなどです。

デジタルイノベーションの加速というビジネス要件とソフトウェアセキュリティリスクとのバランスを取る必要があることを組織が認識するにつれ、アプリケーションセキュリティ業界は着実な進化を続けています。これは目新しいことではありませんが、テクノロジーの変革 (API、マイクロサービス、IAC イノベーション、クラウド技術の爆発的な普及を含み、市場投入期間の短縮に対する需要がますます高まっている) のペースは加速しています。どの組織でも、AST 配信の速度とコードセキュリティ解析の深さと品質を引き換えにすることはできないことを認識しながら、絶えず境界線を押し進めています。

このリストには、お客様から取り組みや提供を求められているものが数多く含まれています。こうしたトピックの多くは、決して数段落で説明しきれるものではありません。リストに追加したいトレンドや項目をぜひお聞かせください。

それでは、2022 年のアプリケーションセキュリティの主なトレンドをご紹介します。

ソフトウェアサプライチェーンの保護

近年、ソフトウェアサプライチェーンへの攻撃の深刻さと頻度が著しく増加しています。オープンソースコンポーネントを利用して開発プロセスを加速することに大きな利点があることが証明されてきています。そのため、全コードベースの 98% がオープンソースに依存しています。しかし、サプライチェーンには、攻撃者が利用できる盲点や亀裂が数多くあります。

Log4J や SolarWinds といった最近のソフトウェアサプライチェーン攻撃のいくつかは、幅広く報道され、政府や企業に、サプライチェーンを精査し、リスクから保護するために必要なプロセスを導入することを促しました。しかしソフトウェアコンポジション分析以外にも、内部ユーザーによる脅威分析 (悪意のあるコードインジェクション)、安全でないコンパイル (トロイのソース)、ハッカーレベルインサイト (ブラウザーで実行時にダウンロードおよび実行されるサードパーティのクライアント側 JavaScript) など、多くの新しい脅威が発生しています。

ガートナー社は、2023 年までに B2B 取引の 50% 以上が従来のアプローチと異なりリアルタイム API を介して行われるようになると述べています。

たとえば、脆弱なコンポーネントは、そのソフトウェアを本番稼動させると同時に、暴露を生じさせる可能性があります。従来のコンポジション解析は、開発と導入の間で行われるため、脆弱なコンポーネント対策には有効な防御策となります。ただし、マルウェアコンポーネントの問題は、開発者のワークステーション自体に大きな損害を与える可能性もあります。ソフトウェアコンポジション分析は、このような攻撃に対する効率的な防御にはなりません。一般的なアンチマルウェアソフトウェアが必要になります。

今後のサプライチェーンセキュリティは、ユーザーが使うソフトウェアの CVE スキャンを超えるものになると見込まれます。それには、悪意のあるコードインジェクション、SDLC を通じてのコードの整合性、導入と運用を推進するインフラストラクチャ、そしてソフトウェアが実行時に相互作用するサードパーティのコード、コンポーネント、およびインターフェイスなどの攻撃ベクトルが含まれます。同様に重要なのは、どのようなアプリケーションを構築する場合でも、プロアクティブに、最適かつ最も安全なオープンソースコードを発見 / 選択する能力です。

ますます高まる API セキュリティのニーズ

API を対象とした攻撃が急増していますが、開発者やアプリケーションのセキュリティ管理者には認識されておらず、見過ごされがちです。

最新のクラウドネイティブアプリケーションは一般的に、分散型アーキテクチャ、サービス / マイクロサービス、サーバーレス機能を採用しています。これらのコンポーネントは、相互通信およびエンドユーザーや API との通信を行うためコンポーネントレベルでもシステムレベルでもセキュリティを評価する必要があります。コンポーネントレベルでは、サービス間の通信に HTTP、SOAP、gRPC などさまざまなプロトコルが使用されます。システムレベルでは、通常は API ゲートウェイを使用して、個々のサービス API を HTTP (通常は REST) ベースで統合型ビジネスアプリケーション API に統合します。しかし近年では、Facebook で作成され 2015 年にコミュニティにリリースされた言語である GraphQL の人気が高まっています。

API のテストと検出は、複数の手順から成るプロセスです。API を保護するための最初のステップは、独立したコンポーネントそれぞれの DevSecOps パイプラインに SAST を組み込むことです。次に、HTTP が使用されるコンポーネントレベルとシステムレベルの API において、API のセキュリティに DAST スキャンを組み込みます。次のステップは攻撃対象領域の把握です。つまり、API 攻撃対象領域となるエンドポイントとパラメーター (「攻撃対象」) を特定します。攻撃対象領域を適切に把握するためには、「攻撃対象」に加えて、API の使用方法 (「攻撃方法」) も特定する必要があります。これは、ビジネスロジックワークフローにおいて、そして場合によってはシステムレベルの API ゲートウェイでの複雑な認証において重要となります。

シフトレフトから「シフトエブリウェア」に進化する AppSec

「スキャンの速度と頻度を上げて SCA チケットの優先順位を上げると、CI/CD パイプラインにセキュリティテストを緊密に統合している企業では、新しい問題の 91.4% が修正されていることがわかりました」

シフトレフトは、SDLC の中でアプリケーションのテストとセキュリティの実装が**どこで行われるか**だけでなく、セキュリティテストに**だれが責任を持つか**にも大きな影響を与えています。AppSec テストの購入と実装に関しては、開発者が主要な推進主体になりつつあります。

現実には、ビジネスがセキュリティよりも優先されがちです。開発者は、バグをできるだけ少なくし、できる限り迅速に機能を提供するように奨励されています。そのため、開発を遅らせることなく開発者が脆弱性を修正できるように、開発者パイプラインにセキュリティを入れ込む方法を見つけ出すことが重要です。

SDLC のあらゆる段階へのシームレスな統合が、AppSec ツールの標準になりつつあります。DevOps ツールチェーンにおけるツールに関しては、AppSec チームの影響力は依然として大きくありません。開発組織の影響力が大きくなると、多くのベンダーが非常に簡便なスキャンを提供し始めました。初期のサービスでは、利便性とコスト削減のために、より強固な AppSec ツールで発見できる脆弱性のほんの一部しか検出できないようなツールが提供されていましたが、組織はこれによってセキュリティ対応済みとしていました。

利便性と堅牢性のせめぎ合いにより、AppSec 業界全体がソフトウェア開発ライフサイクル全体を通じた緊密な統合を目指すようになったのです。トップレベルの AppSec ツールが CI/CD パイプラインに「シームレスに」統合されるようになると、成熟した AppSec プログラムを使用している組織では「シフトレフト」という考え方が現れます。実際には、「シフトレフト」の振り子は「シフトエブリウェア」にまで振れています。それはまさに、より確実な防御のため、適切な職務に適切なツールを見つけることです。

セキュリティは、明らかに DevSecOps の重要な要素になっています。ベンダーやツールが成熟するにつれて、統合や活用の実績が重要なものになっています。結果の品質と、リスクを効率的に修正、削減できるようにすることこそ重要であり、簡便なスキャン機能でセキュリティ対応済みとすることは重要ではありません。

クラウドネイティブな AppSec

IT 業界では広くクラウド化が進んでおり、各企業や組織はさまざまな段階にあります。最新のソフトウェアスタックには、クラウドネイティブなアーキテクチャ要素が含まれています。CNFC は、クラウドネイティブをこのように定義しています。「パブリッククラウド、プライベートクラウド、ハイブリッドクラウドなどの最新の動的環境でスケーラブルなアプリケーションを構築して実行できるよう組織を導く技術。このアプローチの例として、コンテナ、サービスメッシュ、マイクロサービス、イミュータブルインフラストラクチャ、宣言型 API があります」

組織の半数以上 (57%) が 3 つ以上のクラウドプラットフォームを使用しています。

AWS、Azure、GCP は、市場で大きなシェアを占めています。各社は、クラウドインフラストラクチャのコンポーネントと連携する、似たような機能と SDK を提供しており、サーバーレス機能を通じてさらに高いレベルの抽象化を実現しています。クラウドコンピューティングにより、多くのタスクをクラウドプロバイダーにオフロードできますが、クラウドにオフロードしたデータを保護する責任はエンドユーザーの組織が負います。これは、AppSec にさまざまな点で影響します。その 1 つは、クラウドで実行されるアプリケーションに特定の AppSec 要件があることです。

その内容は、アプリケーションのアーキテクチャに依存します。アプリケーションをクラウドに配置する場合、最も簡単なアプローチは、クラウドインフラで実行する「リフトアンドシフト」です。この場合、変更点はほとんどありません。もう 1 つのアプローチは、クラウドテクノロジーを最大限に活用して「クラウドネイティブ」にすることです。こちらの場合には、多くの変更点があります。二者択一ではなく、これは連続したものです。大規模な組織には、そうした連続した範囲の中で分散したアプリケーションの大規模なポートフォリオが存在します。

クラウドネイティブのケースでは、AppSec の具体的なポイントは以下のようなものです。

- **クラウド SDK の使用。**たとえば標準の Java JDBC を使用して MySQL に接続するのではなく、AWS Java SDK を使用して S3 に接続することもできます。これには、実践とリスクに関して良い面と悪い面がそれぞれあります。{クラウドのプロバイダー} x {サービス} x {SDK 言語} の組み合わせを考えると、この領域は非常に広範です。AppSec ツール (特に SAST) がこれに対応するのは困難です。
- **Infrastructure as Code。**この分野では以下のようなことが考えられます。
  - 本当の意味での「AppSec」ではないこと。静的解析の場合、AppSec はコード解析でもあり、同じリポジトリにあることが多いので、自然に IaC に拡張されるというのが私たちの見解です。動的テストの側面もあります。クラウドの実際の構成 (構成ドリフト / 本番の監視) はどうでしょうか。これはさらに AppSec から離れた分野です。
  - この分野はまだ急速に発展中です。クラウドベンダー固有のソリューション (CloudFormation、ARM テンプレート)、構成上の抽象化レイヤー (Ansible、Terraform)、コンテナ管理レイヤーによる抽象化 (Kubernetes) があります。
- **コンテナ**
  - セキュリティを評価するためのさまざまなアプローチがあります。
    . AppSec に近い作成プロセスファイルのチェック (Dockerfile)。
    . AppSec とは離れたイメージのスキャン。
- **サーバーレス /Function-as-a-Service**
  - クラウドに特化したデプロイメントモデル。(すべてクラウドプロバイダーが処理するため) アプリケーションサーバー / コンテナ /VM に煩わされることなく、単純にコードをデプロイする。どんな AppSec ツールでも自動的に認識されません。
- **クラウドの秘密情報**
  - クラウドと IaC のトレンドにより、ハードコードされた秘密情報の保存による新たなリスクを生じさせます。クラウドの秘密情報が漏えいした場合の影響は甚大となります。
  - 可能性：多くのクラウド /API の秘密情報はフォーマットが固定されている傾向があり、比較的単純な解析で正規表現を見つけることができます。多くのベンダーがこの分野に参入しています。

- **クラウド CI/CD**
  - ホスト型 DevOps システム (ADO、GitLab、GitHub、AWS CodeStar など)。
  - 独自の (サプライチェーン) リスク、特にワークフローでのインジェクションの可能性があります。

これらの AppSec の新しい側面による結果を見てみましょう。

- これらすべてが AppSec ツールへの新たな要求です。
- AppSec と InfraSec との境界があいまいになりつつあります。
- 1 つのことに特化して優れたものを持った多くのニッチなベンダーが現れています。
- 同時に、組織はツールの乱立を懸念し、統合ソリューションを好むようになっています。

AppSec のオーケストレーションと相関関係

アプリケーションセキュリティのオーケストレーションと相関関係は、業界でますます注目されるようになってきています。これらは同時に語られることが多いのですが、実際には 2 つの別々のトピックの組み合わせです。ここではこれらを別々のセクションに分けて論じます。

オーケストレーション
現代の開発においてスピードと複雑さが増す中、AppSec チームへの要求は増大し続けています。多くの組織では、さまざまなベンダーのさまざまなツールを使用して、SCA、SAST、DAST などの AppSec のニーズに対応しています。これらのツールを個別に管理しようとすると、複雑さによる問題や帯域幅の問題が発生します。より広い観点から見ると、1 人のセキュリティ専門家は、担当するアプリケーションで使用されているセキュリティツールにしかアクセスできないかもしれません。AppSec オーケストレーションは、そうした AppSec プロフェッショナルの小規模なチームが、増大する要求に対応し、組織全体の大規模な開発者チームにスケーラブルで動的および静的スキャンソリューションを提供できるようにする上で重要な役割を果たします。これは、単一のソースを使用し、組織全体で使用されている多数のツールにわたって、自動化されたスケーラブルなスキャンをスケジュールすることによって実現されます。

相関関係
開発組織のリーダーや経営陣は、主に環境のリスクに関心を示しています。リスク管理では、アプリケーションおよびそのサポートインフラストラクチャのリスクを包括的に把握します。このアプローチに重点を置くことで、経営陣は資産、ビジネスコンテキスト、ROI をより明確に把握できます。Saltworks Security の SaltMiner などのソリューションは、単なる AppSec 以上のコンテキストデータを取得することでこれを実現します。

脆弱性管理では、結果を集約、分析して 1 つの画面にレポートし、組織内のすべてのアプリケーションセキュリティイニシアチブを可視化するようなソリューションが進化し続けています。これにより、組織では AppSec データを経営幹部レベルで全体像を把握できるようになります。

AppSec 相関の基本的な概念をさらに拡大して拡張脆弱性管理とするときに、起こり得るシステム的な問題やパターンがあります。静的解析の結果に動的解析の結果を重ねることにより、価値のあるリスク評価指標が追加で得られるため、実際のリスク状況の全体像をより良く把握することが可能になります。静的解析などのテクノロジーにより SDLC の初期段階で脆弱性を特定することは重要ですが、DAST スキャンによりこれらの解析結果が実行環境でいつ顕在化するかを特定できるフィードバックループを作成することも極めて重要です。SDLC の初期段階で XSS などの問題を特定し、運用環境でそれらの問題を検知することができれば、トレーニングと開発のリソースをシステム的な問題への対処に集中させることができます。

統合されたアプリケーションセキュリティ脆弱性管理のプラットフォームの重要性は、簡素化された優先度設定とそれによるトリアージのワークフローを実現できることだけでなく、データからパターンを収集できることにあります。よりインテリジェントなスキャンとは、SAST の結果を DAST で検証し、SAST の結果で DAST を調整できるようにすることです。

次世代の DAST

DAST のパイプラインの前段階での実施はさらに進みつつあります。従来、DAST スキャンには時間がかかるため、厳重な DevSecOps ワークフローへの統合は進みませんでした。しかし、開発者主導の DAST テストが拡大し始めています。DAST の使用は AppSec/QA から、完全に Dev CI/CD 自動化パイプラインに組み込まれつつあります。

これにより、DAST をより迅速なテストサイクルに組み込むことができます。パイプラインに自動化されたセキュリティスキャンが組み込まれると、多くの利点が得られ、検出と修正を迅速化できます。

- なにか新しい脆弱性があれば本番導入前に開発者に警告が上がり、必要に応じてビルドを中断してリリース前にレビューが行われるようにします。
- テストは、お客様向けアプリケーションに限定されるのではなく、基盤となるサービスや API に対して実行できるため、バグが見つかった場合に根本的な問題を迅速に特定できます。

機能テストスクリプトに沿った DAST スキャンを行うと、作業中のアプリケーションの一部分だけが、作業中のコードのコンテキストに残ります。自動的に実行され、既存のプロセスやツールと統合できるスキャンの導入により、セキュリティチームや開発チームの作業を迅速に進められるようになります。スキャンのスケジュール設定よりも、重要な問題の解決が重点であることには変わりません。このアプローチは通常、最近増加している IAST メソッドよりも優れた結果をもたらします。パッシブ IAST はアプリケーションをクロールせず、その効果は、機能テストスクリプトを作成しアプリケーションを手動で実行するユーザーに依存します。DAST はこれらの機能を備えているだけでなく、アプリケーションの攻撃対象領域を独自に発見することもできます。これは、QA がスクリプトを作成する際に、すべてのシナリオとコードパスをカバーできるという信頼がどの程度あるかという問題になります。シナリオを 100% カバーできない限り、すべての攻撃対象領域を見つけるには DAST が必要です。

より早い段階でのテストを行えば、以前のように開発プロセス全体を工程後半のセキュリティチェックに合わせて調整する必要はありません。これにより、セキュリティチームにとって大きなハードルとなることが多い DAST のスケーラビリティが向上します。スキャンを一元化するソリューションは、DevSecOps パイプラインで DAST を広範に機能させるための重要な要素です。

迅速なフィードバックと包括的なスキャンをともに実現できるように DAST を設定するには、次のような優れた方法があります。

- チェックインのたびに、あらゆる機能テストを DAST で実行する。これにより開発者は、IAST と同様に、変更に関するフィードバックをすばやく得ることができます。
- 夜間のビルドでは、アプリケーション全体をクロールするさらに包括的なスキャンを実行し、完全なカバレージを得る。

セキュリティテストの大部分 (75% 以上) を自動化している組織は、わずか 29% です。コーディングワークフローの一部として、セキュリティテストとレビューが含まれている組織は半数未満 (44%) です。

機械学習と AI が今後の自動化の進化における重要な鍵

自動化は、シフトレフトのセキュリティを強化する最大推進要因の 1 つです。このことは、いくつかの調査で、自動化を使用する企業ではセキュリティテストを実装する可能性が 2 倍にもなると示されたことで裏付けられています。多くの組織では自動化の必要性があることを認識し、一部の自動化は行われていますが、改善の余地はまだ残っています。ガートナー社によると、回答者の 95% が自動化を使用しているもののデプロイメントパイプラインを完全に自動化しているのはわずか 33% だとされています。さらにガートナー社は、企業の 32% がセキュリティツールを手動で統合していると示しています。

開発プロセス全体で使用されている実装とツールの自動化を進めるという課題があり、その推進も続いていますが、既存のデータと機械学習を活用した自動修復という形で、さらに多くのメリットが得られます。たとえば、自動化されたプルリクエストを使用したソフトウェアコンポジション分析で、このような効果が見られます。Fortify は Audit Assistant ツールを使用して、この分野でも革新を続けています。Fortify のサービス型アプリケーションセキュリティ製品 (Fortify on Demand) を使えば、週に何千もの静的スキャン、動的スキャン、モバイルスキャンを実行し、数十億行ものコードをスキャンできます。Fortify on Demand では、お客様のアプリケーションのソースコードを取得してスキャンを実行し、(付加価値サービスとして) これらの生のスキャン結果を、対象分野の専門家である監査チームに渡します。そのチームが、重要な事項を特定して優先順位を付けつつ、結果からノイズを除去します。

その結果、Fortify on Demand のお客様は、実用的な結果を得ることができ、最も重要な問題の解決に集中することができます。Fortify Audit Assistant サービスは、機械学習アルゴリズムを使用して、Fortify on Demand の専門家による数億件の匿名の監査判定をフィードバックします。これらの判定モデルは、Fortify on Demand のために積極的に使用および開発されたものですが、Audit Assistant を使用してオンプレミスの Fortify Static Code Analyzer の結果に自動的に適用することもできます。この革新的な特許出願中の技術を、Fortify のお客様には過去 5 年間にわたってご提供しています。

今後は、他の AppSec 脆弱性タイプの AI 支援による監査が、SAST 解析結果のサブセット (設定 /IAC スタイルなど) から始まって、もっと多く出てくると思われます。また、機械学習の進歩には、多くのユースケースがあります。当社のソフトウェアコンポジション分析の製品である Debriced は、これを Open Source Select で行います。GitHub 上のすべてのオープンソースの健全度を分析比較し、ライブラリやフレームワークを調査する際により適切な判断を下すことができます。

結論

デジタルイノベーションの加速というビジネス要件とソフトウェアセキュリティリスクとのバランスを取る必要があることを組織が認識するにつれ、アプリケーションセキュリティ業界は着実な進化を続けています。Fortify の Software Security Research チームによると、Web アプリケーションの大部分は、少なくとも 1 つの重大または重要の問題を抱えています (最新の AppSec リスクレポートでは 79%)。これに、オープンソースコンポーネント内の既知の脆弱性、クラウドへの移行に伴う新たな攻撃対象などを合わせて考えると、信頼できるアプリケーションセキュリティパートナーを確保することが成功に不可欠であることは明らかです。

本文書で紹介したトレンドは、開発者主導のセキュリティとデジタルイノベーションを実現する実用的な成果に焦点をあてた、最新の開発フレームワークに関するものです。

お客様の声もぜひお聞かせください。最も関心を持たれたトレンドは、どれですか。注目している AppSec のイノベーションは、他にありますか。

Fortify について

Fortify は、業界をリードする信頼できる AppSec パートナーとして、お客様のソフトウェアレジリエンスの構築をサポートします。Fortify の静的、動的、インタラクティブ、およびランタイムセキュリティテストテクノロジーは、オンプレミス、SaaS、マネージドサービスとして使用でき、必要に応じて柔軟にエンドツーエンドのソフトウェアセキュリティ保証プログラムを構築できます。

詳細情報はこちら：
www.microfocus.com/appsecurity

お問い合わせ先：CyberRes.com
この記事はいかがでしたか？シェアはこちら

マイクロフォーカスエンタープライズ株式会社
jp-info-enterprise@microfocus.com
www.microfocus-enterprise.co.jp

CyberRes
A Micro Focus Line of Business