SOLUTION BRIEF

フォーティネットのコンテナセキュリティ

概要

コンテナテクノロジーの活用によって、モジュール化され弾性に富んだ新たな方法でアプリケーションを設計し、開発できるようになります。これは、コードのライフサイクル管理やスケーラビリティを考慮して、アプリケーションの異なる論理機能を異なるコンテナに分割することで実現します。さらには、コンテナテクノロジーによってアプリケーションを異なる論理機能、すなわちマイクロサービスに分割することで、パブリッククラウドやプライベートクラウドの異なる環境間でのマイクロサービスやアプリケーションのポータビリティが向上します。

アプリケーションアーキテクチャのこのような根本的な変化によって、多くの組織は現在のセキュリティソリューションではコンテナテクノロジーに関連するリスクを十分に解決できないと考えるようになりました。このようなギャップを解消するには、セキュリティのポイント製品を購入してコンテナ環境を保護し、コンテナテクノロジーに関連する新しいアプリケーションライフサイクルへと強制的に適合させる必要があります。フォーティネットのコンテナセキュリティ戦略では、コンテナベースのアプリケーションのライフサイクル全体に対応する複数のソリューションを提供することで、コンテナベースのアプリケーションの多様な脅威ベクトルに関連する脅威からの包括的な保護を可能にします。

コンテナセキュリティの主な属性：

- コンテナ識別型セキュリティ
- コンテナ対応型セキュリティ
- コンテナ統合型セキュリティ
- コンテナレジストリセキュリティ

企業の開発部門は、オンプレミスあるいはクラウドのいずれにおいてもコンテナで速やかにモジュラーアプリケーションを開発しなければなりません。そして、セキュリティにもアプリケーションと同様のポータビリティと一貫性が必要です。

コンテナテクノロジーのメリット

コンテナを活用することで、仮想マシンよりもさらに小さなアプリケーション実行環境のピースに分割できるため、基本的には自律的な機能が実現します。必要とされるパフォーマンスがピース毎に異なるため、多くの組織はそれぞれのピースを個別に維持管理し（異なるバージョンやバグ修正を含む）、各々を個別に拡張することを望んでいます。これらのピースは一般的にサービスと呼ばれ、最近ではマイクロサービスという呼び方がよく使われるようになりました。

サーバー仮想環境では、イメージのリポジトリを管理するため、ハイパーバイザーまたは仮想インフラストラクチャのレベルで認識される「タグ」と呼ばれるメタデータ属性のセットが仮想マシン（VM）に存在します。コンテナの場合も、同様に「ラベル」と呼ばれるメタデータ属性が関連付けられます。

現在、最も一般的なコンテナ形式の標準であるのは Docker で、これにはオープンソースと商用の両方の実装があります。コンテナテクノロジーを使用してアプリケーションを構築するには、相互にやり取りし相互に依存する複数のサービスが必要で、これは一般的に POD と呼ばれます。アプリケーションの構築には複数のコンテナが必要であり、それらのコンテナを POD にグループ化すること、あるいはしないこともできますが、アプリケーションを正しく動作させるにはそれらすべてを相互接続する必要があります。

コンテナは、通常コンテナ化されたアプリケーションを起動するオーケストレーションプロセスの一部として実行されるサービス / アプリケーション構造によって接続されます。このオーケストレーションプロセスによって異なるサービスのアドレスが動的に割り当てられ、異なるサービスのサービス / 名前の解決機能が提供されて相互解決に使用されます。Kubernetes は、この全体像において重要な役割を果たします。

Kubernetes は、アプリケーション構造、サービスの依存関係、サービス規模の要件、サービス可用性の要件などの記述に必要な機能を備えた、最も広く利用されているコンテナオーケストレーションシステムです。Kubernetes には、アプリケーション全体の可用性を中断することなく、異なるサービスのライフサイクル、スケーラビリティ、可用性、およびパフォーマンスを個別に管理するためのツールも提供されています。

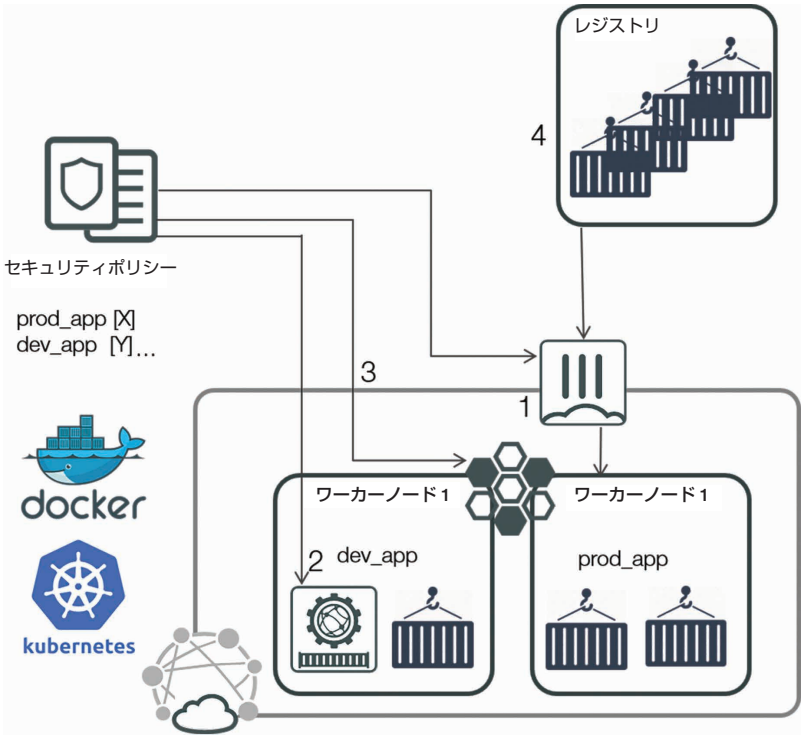

図 1. アプリケーションライフサイクル全体でアプリケーションコンテナを保護

アプリケーションコンテナライフサイクル全体を保護するフォーティネットのセキュリティ

コンテナは今なお新しいテクノロジー分野であるという認識から、さまざまなテクノロジーや標準がコンテナベースのアプリケーションの異なるサービスの相互接続に使用されているため、コンテナセキュリティの主要な属性に対応する包括的なセキュリティソリューションが必要です。

フォーティネットのコンテナセキュリティソリューションは、以下のようにこれらの要件に対応します。

1. **コンテナ識別型セキュリティ：**FortiGate 次世代ファイアウォール（NGFW）は、コンテナ管理レイヤーへと効率的に接続し、異なるコンテナのラベルを学習します。セキュリティポリシーはラベル識別型であるため、これらのラベルを使用してセキュリティポリシーにオブジェクトを記述できます。このソリューションは、主としてコンテナインフラストラクチャへと出入りするトラフィックのセキュリティ、すなわち垂直方向のセキュリティに対応します。FortiGate NGFW は、Kubernetes、AWS EKS、GCP GKE、Azure AKS、OCI OKE などの主要なコンテナオーケストレーションシステムとのネイティブインタフェースとなるファブリック コネクタを提供し、セキュリティポリシーオブジェクトとしてメタデータの活用を可能にします。コンテナ化された環境の境界を離れるトラフィックが FortiGate NGFW を通過する際に、コンテナのロールに基づいてポリシーが適用されます。また FortiGate は、FortiSandbox との統合によって提供される侵入防止システムおよび高度なマルウェア保護機能を活用し、コンテナの送受信トラフィックをスキャンして脆弱性やファイルベースの脅威に対する保護を実現します。
2. **コンテナ対応型セキュリティ：**FortiWeb Web アプリケーションファイアウォールをコンテナイメージとして利用し、アプリケーションチェーンにバンドルすることもできます。Web サービスベースのアプリケーション用にマイクロサービスを作成することは極めて一般的であるため、Web アプリケーションと API 保護をマイクロサービスベースのアプリケーションと組み合わせる機能は、これらのアプリケーションを構築する組織にとって大きなメリットとなります。開発者は、アプリケーション開発のライフサイクルに合わせてセキュリティ制御を展開し、アプリケーションライフサイクル全体を通じ、他のアプリケーションサービスと並行してアプリケーションセキュリティを異なる環境に展開することができます。FortiWeb は

効率化が進んだ DevOps 組織をそうでない DevOps 組織と比べると、コードのコミットは 46 倍、平均復旧時間は 96 倍、確定から配備までのリードタイムは 440 倍、障害による変更率は 5 分の 1 であると報告されています[1]。

現在、ネイティブの Docker コンテナとしてだけでなく AWS EKS マーケットプレイスでも提供されています。また FortiWeb は、FortiSandbox と統合することで送受信トラフィックに対するゼロデイ脅威保護のレイヤーを追加します。

3. **コンテナ統合型セキュリティ：**コンテナベースの内部アプリケーショントラフィックの多くはコンテナホストの内部で発生し、ネットワークインフラストラクチャでは認識されません。それぞれのトラフィックフローのインスペクションにおいては、アプリケーションまたはサービス挿入のメカニズム内部でサービス間のトラフィックフローを変更する必要があります。セキュリティサービスをアプリケーション構造に付加することで、すべてのトラフィックフローがセキュリティ処理サービスに認識されるようになります。これらのサービスは、コンテナベースまたはネットワークベース（コンテナインフラストラクチャの外部に存在する）のどちらでも構いません。サービスとコンテナベースのアプリケーションの並列での挿入に使用する手法やテクノロジーは発展途上であるため、どのテクノロジーが主流になるかの予測は極めて困難です。このような状況に対応するため、フォーティネットはコンテナインフラストラクチャのセキュリティソリューションをリードするサードパーティプロバイダーと協力し、コンテナセキュリティのすべての属性を単一ソリューションに統合するセキュリティソリューションを提供しています。
4. **コンテナレジストリセキュリティ：**コンテナイメージは、一般的にレジストリと呼ばれるパブリックリポジトリに保存され、新しいコンテナイメージはほぼ制限なくレジストリに公開されます。そのため、意図的あるいは不注意によってコンテナイメージに不正なコードが埋め込まれ、アプリケーション開発者によってレジストリから簡単に「取り出されて」しまう危険性があります。このような状況は、アプリケーション開発プロセスに不要なリスクをもたらすことになります。FortiSandbox では、コンテナベースのアプリケーション開発者のニーズに対応する専用の API（アプリケーションプログラミングインタフェース）や統合の機能が提供されているため、アジャイル開発の方法論によってもたらされる潜在的リスクが減災されます。

最上クラスの DevOps セキュリティ組織を最下層のセキュリティ組織と比べると、セキュリティ監査追跡を実施する傾向が 187%、依存関係の分析を実施する傾向では 96%、構成ミスを見つけるためにクラウドインスタンスをスキャンする傾向で 6%、コードコミットの監視と管理を実行する傾向は 45% 強いと報告されています [2]。

安全で包括的なコンテナ戦略の実現

コンテナテクノロジーは、新たなアプリケーションインフラストラクチャと開発用のテクノロジーとして瞬く間に大きな注目を集めるようになりました。しかしながら、コンテナテクノロジーによってもたらされるリスクは、従来型セキュリティツールでは適切に対処することができません。パブリックとプライベートの両方のクラウド環境にアプリケーションやコンテナインフラストラクチャを導入する組織にとっては、広範なコンテナオーケストレーションシステムと互換性がある包括的コンテナセキュリティソリューションを導入する手段が不可欠です。

フォーティネットのコンテナセキュリティソリューションは、このように拡大した攻撃対象領域に完全対応し、コンテナアプリケーションのライフサイクルへのセキュリティの統合、そしてさらに安全なアプリケーションの提供を可能にします。

[1] 「State of DevOps: Market Segmentation Report」、Puppet、2018 年 5 月（英語）：
https://puppet.com/resources/report/state-of-devops-market-segmentation-report/

[2] 「2019 State of DevOps Security Report」、フォーティネット、2019 年 5 月 10 日（英語）：
https://www.fortinet.com/content/dam/maindam/PUBLIC/02_MARKETING/08_Report/report-devops.pdf

FORTINET
フォーティネットジャパン株式会社

〒106-0032
東京都港区六本木 7-7-7
Tri-Seven Roppongi 9 階
www.fortinet.com/jp/contact

お問い合わせ