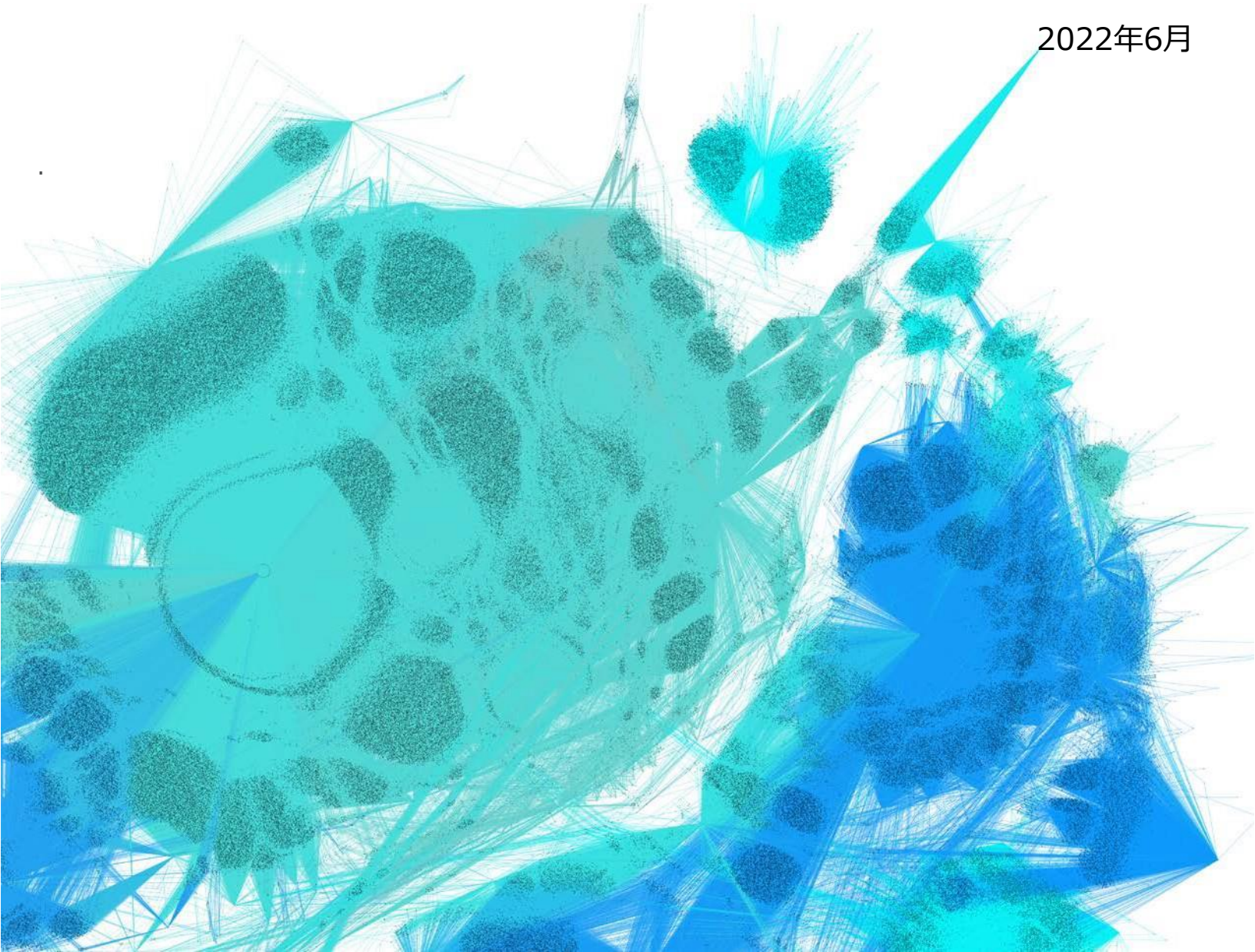
A Trend Micro White Paper | February 2020

トレンドマイクロのソリューションによる NIST SP800-190(Application Container Security Guide) 準拠の支援

トレンドマイクロ株式会社

2022年6月



目次

エグゼクティブサマリ

このホワイトペーパーでは、Trend Micro Cloud One™（以下、Cloud One）の各種セキュリティソリューションを利用して、米国国立標準技術研究所（NIST）が発行するSP 800-190（Application Container Security Guide）が提示する要件を満たしながら、コンテナ環境に対するセキュリティの適用範囲、保護内容について考察します。

Cloud Oneは、サーバはもちろん、コンテナやサーバレス環境など多様なサービスで構成されるクラウド環境をまとめて保護するセキュリティサービス群の総称です。Cloud Oneでは、コンテナエンジンが稼働するコンテナホストを総合的に保護するTrend Micro Cloud One – Workload Security™（以下、Workload Security）や、コンテナイメージに対して脆弱性やウイルススキャンを実施するTrend Micro Cloud One – Container Security™（Container Security）をはじめ、全7種のセキュリティサービスを提供しています。

Cloud Oneの各サービスを利用することでコンテナ環境のライフサイクル全体、パイプラインにおけるセキュリティを包括的に実装し、NIST SP 800-190（Application Container Security Guide）が定める各要件への準拠を支援できます。

NIST SP 800-190とは

米国国立標準技術研究所（NIST）が2017年9月に公開した、コンテナ特有のセキュリティリスクとその対策方法についてまとめた文章がNIST SP 800-190（Application Container Security Guide）です。

このガイドの中ではコンテナ環境におけるリスクと対策は大きく5つに、さらに細分化された分類では23項目に分けられています。

これらが示すセキュリティリスクに対応するようにコンテナ環境のセキュリティを設計、運用することがコンテナを安全に使うこと、そしてコンテナを用いたシステムの可用性や開発速度の向上につながります。

NIST SP 800-190が定める要件

大項目	中項目
1 イメージリスク	1.1 コンテナイメージの脆弱性
	1.2 コンテナイメージの設定の不備
	1.3 マルウェアの埋め込み
	1.4 平文パスワードの埋め込み
	1.5 信頼できないイメージの利用
2 レジストリリスク	2.1 レジストリへの安全でない接続
	2.2 古いコンテナイメージの残存
	2.3 レジストリへのアクセス時の不十分な認証・権限
3 オーケストレーターリスク	3.1 管理者権限の不適切な割り当て
	3.2 認証のないアクセス
	3.3 コンテナ間のネットワークトラフィックの不十分な分離
	3.4 さまざまな重要度のワークロードが混在する環境
	3.5 ノードの信頼性
4 コンテナリスク	4.1 ランタイムソフトウェアの脆弱性
	4.2 コンテナからの制約のないネットワークアクセス
	4.3 安全ではないコンテナ実行設定
	4.4 アプリケーションの脆弱性
	4.5 管理されていないコンテナの存在
5 ホスト OS リスク	5.1 広い攻撃範囲
	5.2 カーネルの共有
	5.3 ホスト OS コンポーネントの脆弱性
	5.4 適切ではないユーザのアクセス権
	5.5 Host OS ファイルシステムの改ざん

コンテナ環境を適切に保護するには、コンテナイメージの作成（Build）・レジストリへの保存（Push）・レジストリからコンテナエンジンへのコンテナイメージのダウンロード（Pull）・コンテナイメージを基にコンテナ作成（Deploy）といった、一連のコンテナライフサイクルを意識して、セキュリティの抜け漏れの無いようにすることが重要です。 NIST SP 800-190はコンテナライフサイクルにおいて登場するコンポーネント別にリスクがまとめられており、これを参考にしてセキュリティを実装することで、コンテナのビルドパイプライン全体の保護を実施することができます。

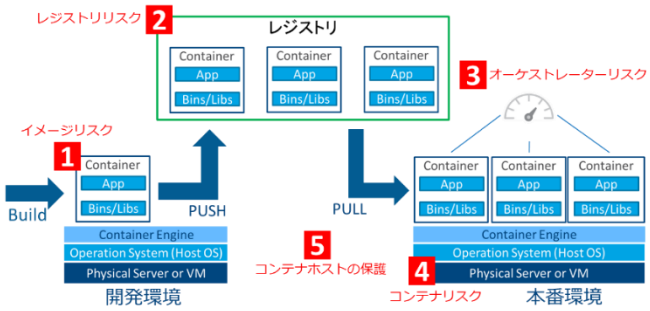


図1：コンテナライフサイクルおよびセキュリティリスク

トレンドマイクロのコンテナ環境保護ソリューション

① Trend Micro Cloud One – Workload Security

Workload Securityはサーバを多層防御するソフトウェアです。物理・仮想・クラウドなど、多様なサーバ環境に対応しており、Workload Security Agent（別名、Deep Security Agent）をインストールすることで、それらすべてに統一した、多層のセキュリティを提供することが可能です。ウイルスを用いた攻撃や脆弱性を利用して外部から不正アクセスされることによるデータ侵害やサーバリソースの不正利用による業務の混乱を防ぎ、コンプライアンス準拠を支援・実現、運用コストを低減するためにも役立ちます。

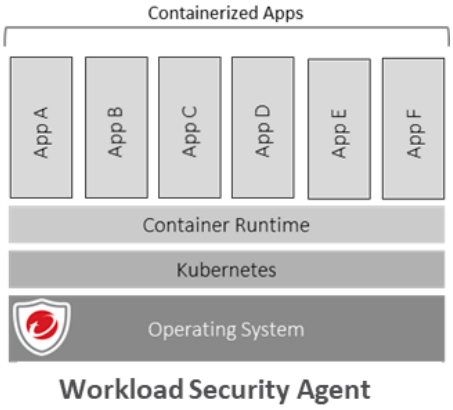


図2： Workload Security Agentのコンテナ環境に対する保護内容

【実装方法】

Workload Security Agentをコンテナエンジンが動作しているノードのホストOSにインストールします。ホストOS単位でセキュリティを実装することで、コンテナアプリケーションそのものにセキュリティ機能を実装することなく、ホスト上で動作するコンテナ環境を保護することが可能です。

※ 2022年5月時点で、Workload Securityがサポートしているコンテナエンジンは Dockerエンジンのみです。

【保護機能】

- **不正プログラム対策**

  サーバに不正プログラムが感染することを防止します。不正プログラムがサーバに侵入しようとした際に検出するリアルタイム検索や、毎週/毎日など事前に設定した時間に検索を行うスケジュール検索によりサーバを不正プログラム感染から保護します。また、Dockerコンテナ内で不正プログラムを検出した場合、どのコンテナで検出したかを把握できます。

- **IPS/IDS（侵入防御）**

  脆弱性に対応するIPS/IDSルール「仮想パッチ」によって、脆弱性を突いた攻撃からサーバを保護します。対応している脆弱性は、LinuxやWindowsなど主要なサーバOSや、Apache、WordPress、Oracle等100以上のアプリケーションやミドルウェアです。

  全てのルールは、防御モード（パケットを破棄するモード）と、検出モード（イベントのみをログに記録しトラフィックは通過させるモード）を選択することが可能です。Workload Securityの導入に伴い、正常パケットを止めてしまう等の誤検知が心配な場合、初めは検知モードで導入し、アプリケーションの動作が確認されてから防御モードにするなどの方法で、安全に導入することが可能です。

- **Webレピュテーション**

  サーバがWebサイトにアクセスするなどの通信が発生する際、Trend Micro Smart Protection Networkに問い合わせを行い、接続先ドメイン、Webサイトが不正な場合にはアクセス自体をブロックすることによって不正プログラムのサーバへの感染、情報漏えいなどを防ぐことができます。

- **ファイアウォール**

  適正なサーバ運用に必要なポートおよびプロトコルを介した通信を可能にし、それ以外のポートおよびプロトコルはすべてブロックすることで、サーバへの不正アクセスのリスクを低減します。IPアドレス、MACアドレス、ポートなどに基づいてファイアウォールルールを設定し、トラフィックをフィルタリングするだけでなく、ポートスキャンなどの不審なアクティビティを検出することも可能です。ARPトラフィックなど、IP 以外のトラフィックを制限することもできます。

- **アプリケーションコントロール**

  サーバにインストールされたアプリケーションをホワイトリスト化し、許可されていないプログラムが実行された時に検知・ブロックする機能です。未知の不正プログラムの実行を防止したり、サーバの用途を限定したりしたい時などに有効です。また、サーバに新しいアプリケーションを追加するときや、既に実行されているアプリケーションのバージョンアップをするときなどは、管理コンソール上でアプリケーションコントロールを「メンテナンスモード」に変更して、その間に追加・変更されたアプリケーションを許可するルールセットに追加することが可能です。定期的に動作するアプリケーションにアップデートが発生する環境でも、できるだけホワイトリスト運用時の負荷をかけないように設計された機能です。

- **変更監視**

  ディレクトリ、レジストリキーおよび値など、オペレーティングシステムとアプリケーションの重要なファイルを監視して、その変更を検出します。ファイルとディレクトリの内容、属性（所有者、アクセス権、サイズなど）、日付と時刻のタイムスタンプの変更を監視できます。アクセス制御リスト、ログファイルの追加、変更、削除も監視して警告できます。

- **セキュリティログ監視**

  オペレーティングシステムおよびアプリケーションのログからセキュリティイベントを収集して分析する機能を提供します。セキュリティログ監視ルールを設定することで、複数のログエントリに埋もれた重要なセキュリティイベントの識別を実施します。これらのイベントは SIEMまたは一元化されたログサーバに転送して、関連付け、レポートを作成して、アーカイブすることが可能です。Linuxの各プラットフォーム全体のイベントと、Web サーバ、メールサーバ、sshd、Samba などからのアプリケーションイベント、またカスタムアプリケーションログイベントを収集して関連付けることができるので、使用するサーバ上で発生した可能性のある疑わしい挙動を確認できます。

② Trend Micro Cloud One – Container Security

Container Securityはレジストリ内に保存されているコンテナイメージの脆弱性や不正プログラムなどを検知するコンテナイメージスキャン製品です。システムの開発時にコンテナイメージのリスクを可視化することで、開発のタイミングで修正プログラムを適用する、コンテナ運用時にセキュリティ製品を用いて脆弱性を悪用する攻撃を防ぐなどの対策を講じることができます。

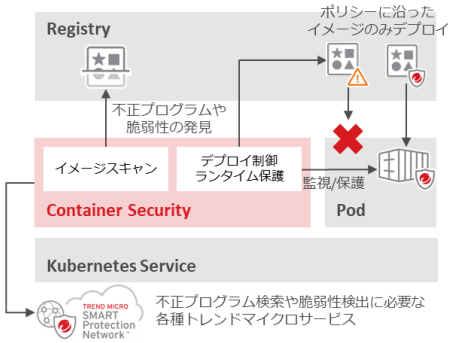


図3：Container Securityの保護・構成イメージ

【実装方法】

Container Securityは、helmを使用してKubernetes環境にPodとしてデプロイします。 コンテナで動作しているDeep Security Smart Check（トレンドマイクロのコンテナイメージスキャナー、Container Securityのコンポーネントの1つ）にスキャン先のコンテナレジストリを登録して、各種検索を実施します。

【保護機能】

- **不正プログラム対策**

  コンテナイメージ内の不正プログラムをSmart Protection Networkに基づいたスマートスキャンや、機械学習型検索 などの技術を用いてスキャンします。

  ※ 機械学習型検索はすべての未知の脅威に対応するものではありません。

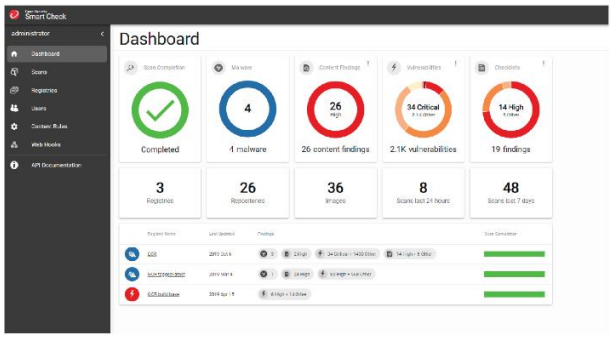


図4：Deep Security Smart Checkの管理コンソール

- **脆弱性検索**

  コンテナイメージ内に存在する脆弱性を検出します。脆弱性を緊急度に応じて「Critical」「High」「Medium」「Low」「Negligible」「Unknown」の６つのレベルで表示し、コンテナイメージ内の脆弱性を可視化します。

- **クラウドサービスのシークレットキーや秘密鍵の検索**

  コンテナイメージ内にAmazon Web Service・Google Cloud Platformのシークレットキーや SSHで利用される秘密鍵が存在する場合、これを検出します。

- **ポリシーベースのデプロイ制御**

  Kubernetesオブジェクトのプロパティと Container Securityによる検索の結果に基づくルールをベースに、イメージのデプロイを許可またはブロックするポリシーを作成できます。

- **継続的なコンプライアンス準拠**

  クラスタに割り当てられたポリシーを定期的にチェックし、実行中のコンテナが定義したポリシーに引き続き準拠していることを確認します。

- **ランタイムセキュリティ**

  ルールに違反する、実行中のコンテナのアクティビティを可視化します。ルールには、コンテナのMITRE ATT&CKフレームワークの戦術に対する可視性を提供するルールも含みます。

③ Trend Micro Cloud One – Network Security™

Trend Micro Cloud One – Network Security（以下、Network Security）はハードウェアIPS製品TippingPointのテクノロジーをクラウド環境で実装した、クラウド環境上のIPS製品です。AWS環境では、ネットワークを外部とAmazon Virtual Private Cloud（以下、Amazon VPC）の通信経路上でNetwork Securityにルーティングし、ここで仮想パッチを提供して、Amazon VPCへの脆弱性を利用する通信を検知・ブロックをすることができます。Amazon VPCへのインバウンド・アウトバウンド通信ともに対応可能です。Microsoft Azure環境の場合はAzure Virtual Network（以下、VNet）と外部ネットワークとの通信経路上で同様の機能を提供します。

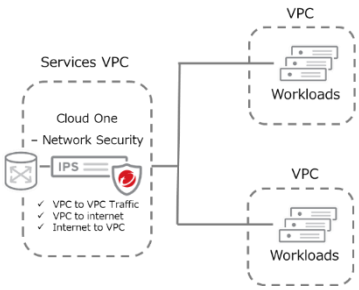


図5： Network Securityの保護･構成イメージ

【実装方法】

Network Securityは、インターネットとAmazon VPCやVNetの間にインラインで設置します。既存のAmazon VPC/VNet内のネットワークは変更せずに利用可能です。

【保護機能】

- **Digital Vaccine（デジタルワクチン・仮想パッチ）**

  Digital Vaccineフィルタ（仮想パッチ）により、脆弱性を利用する攻撃通信を検知・ブロックします。新しい脆弱性の発見から正式なパッチがベンダーからリリースされるまでの空白期間に、Network Security配下のAmazon VPC/VNetにおいて脆弱性対策を施すことができます。

- **Malware Filters**

  Network Security配下の環境にある端末からネットワーク外部に出ていく通信を監視することで、Amazon VPC/VNet内で不正プログラムに感染した後の通信を補足して活動をブロックすることが可能です。Domain Generation Algorithm（DGA）によって生成されたドメイン通信もブロックできます。

- **Reputation Feed**

  IPアドレス・DNSドメイン・URLのブラックリストで不正な通信をブロックします。C&Cサーバやフィッシングサイトへのアクセスを阻止したり、ボットネット感染端末からのDDoS攻撃をブロックしたりすることも可能です。

NIST SP 800-190が要求する項目とトレンドマイクロソリューションの適用範囲

次の表はトレンドマイクロが支援できる、NIST SP 800-190準拠要件の概要を示しています。

大項目	中項目	Cloud One – Workload Security	Cloud One – Container Security	Cloud One – Network Security
1 イメージリスク	1.1 コンテナイメージの脆弱性	—	●	—
	1.2 コンテナイメージの設定の不備	—	◖	—
	1.3 マルウェアの埋め込み	—	●	—
	1.4 平文パスワードの埋め込み	—	●	—
	1.5 信頼できないイメージの利用	—	●	—
2 レジストリリスク	2.1 レジストリへの安全でない接続	—	◕	—
	2.2 古いコンテナイメージの残存	—	●	—
	2.3 レジストリへのアクセス時の不十分な認証・権限	—	◖	—
3 オーケストレーターリスク	3.1 管理者権限の不適切な割り当て	◖	◕	—
	3.2 認証のないアクセス	◖	◕	—
	3.3 コンテナ間のネットワークトラフィックの不十分な分離	—	—	—
	3.4 さまざまな重要度のワークロードが混在する環境	—	—	—
	3.5 ノードの信頼性	◖	—	—
4 コンテナリスク	4.1 ランタイムソフトウェアの脆弱性	●	●	●
	4.2 コンテナからの制約のないネットワークアクセス	◖	—	◖
	4.3 安全ではないコンテナ実行設定	◖	◕	—
	4.4 アプリケーションの脆弱性	●	●	●
	4.5 管理されていないコンテナの存在	—	●	—
5 ホスト OS リスク	5.1 広い攻撃範囲	●	—	●
	5.2 カーネルの共有	—	—	—
	5.3 ホスト OS コンポーネントの脆弱性	●	—	●
	5.4 適切ではないユーザのアクセス権	◕	—	—
	5.5 Host OS ファイルシステムの改ざん	●	—	—

次に、各要求事項と対応方法、トレンドマイクロのソリューションについて説明します。

1. イメージリスク

1.1 コンテナイメージの脆弱性	
リスク概要	イメージは実質的に静的なアーカイブファイルで、特定のアプリを実行するために使用されるすべてのコンポーネントが含まれていいます。このため、イメージ内のコンポーネントには重要なセキュリティアップデートが漏れている、あるいは古くなっている可能性があります。 完全に最新のコンポーネントで作成されたイメージは、作成後数日から数週間は既知の脆弱性がないかもしれませんが、ある時に1つ、または複数のイメージのコンポーネントに脆弱性が発見された時点でイメージは最新のものではなくなります。 デプロイされたソフトウェアが実行されるホスト上の「現場」で更新される従来の運用パターンとは異なり、コンテナの場合は、イメージ自体の更新を上流で行い、それを再デプロイする必要があります。 このように、コンテナ化された環境での一般的なリスクは、コンテナを生成するために使用されたイメージに脆弱性があるため、デプロイされたコンテナが脆弱性を持つことである。
対応方法	コンテナ技術特有の脆弱性管理ツールとプロセスが必要です。従来の脆弱性管理ツールは多くの場合、コンテナ内の脆弱性を検出できず、安全性を確保できないケースがあります。組織がより実用的で信頼性の高いシステムを提供するために、これらを実現するためのビルドパイプラインの構築とコンテナイメージの安全性を確認するためのツールを使用する必要があります。 効果的な対応方法は次のとおりです： 1. イメージのビルド開始から組織が使用しているレジストリ、ランタイムまで、イメージのライフサイクル全体をパイプラインとして構築。 2. イメージのベースレイヤーだけでなく、組織が使用しているアプリケーションフレームワークやカスタムソフトウェアも含め、イメージのすべてのレイヤーにおける脆弱性の可視化。可視化は組織全体で一元化され、組織の要求に合わせた柔軟なレポートおよび監視ビューを提供できる必要があります。 3. ポリシーベースの管理。組織はコンテナの構築および展開プロセスの各段階で「品質ゲート」を作成して、組織の脆弱性基準およびセキュリティポリシーを満たすイメージのみが次のフェーズに進行できるようにする必要があります。たとえば、設定したしきい値を超えるCommon Vulnerability Scoring System（CVSS）レーティングの脆弱性を含むイメージの進行を防ぐために、ビルドプロセスでルールを構成できるようにする、などです。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、コンテナイメージをスキャンし、コンテナOSおよびアプリケーションにパッチ未適用の脆弱性が含まれていないか検索します。検索には、最新の脆弱性データベースを活用します。 検索で脆弱性が見つかった場合、対応するCVE番号と深刻度がスキャン結果の確認画面に表示されます。

1.2 コンテナイメージの設定の不備	
リスク概要	ソフトウェアの不備に加えて、イメージには設定の不備がある場合もあります。 たとえば、特定のユーザーアカウントでのみ実行するようにイメージが構成されていないため、必要以上の特権で実行される場合があります。 別の例として、イメージにSSHデーモンが含まれている場合が挙げられます。この場合、コンテナが不要なネットワークリスクにさらされる可能性があります。従来のサーバや仮想マシンと同様、不十分な構成は完全に最新システムであっても攻撃にさらす可能性があります。このように設定に不備があるイメージは、含まれているすべてのコンポーネントが最新の状態であっても、リスクが高まる可能性があります。
対応方法	組織は安全な構成のベストプラクティスへの準拠を検証および実施する必要があります。たとえば、イメージは非特権ユーザとして実行するように設定する必要があります。 採用すべき対応方法は次のとおりです。 1. ベンダーの推奨事項やサードパーティのベストプラクティスをふまえて、イメージ構成設定の検証 2. 組織レベルで弱点とリスクを特定するための、イメージのコンプライナス状況のレポート・監視 3. コンプライアンスに準拠していないイメージのデプロイをオプションで防止することによるコンプライアンス要件の強制 4. 信頼できるソースからのベースレイヤーのみの使用、ベースレイヤーの頻繁な更新、Alpine LinuxやWindows Nano Server などの選択による攻撃対象領域の削減 イメージ構成の最後の推奨事項は、ホストにリモートシェルを提供するように設計されたSSHおよびその他のリモート管理ツールをコンテナ内で有効にしないことです。これらのツールを介してリモートアクセスを有効にすると、ネットワークベースの攻撃のリスクが高まります。代わりに、コンテナのすべてのリモート管理はコンテナランタイムAPI を介することが推奨されます。コンテナランタイムAPIはオーケストレーションツールを介してアクセスするか、コンテナが実行されているホストへのリモートシェルセッションを作成することでアクセスできます。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、NIST、HIPAA、PCIなどに準拠しているかイメージを検証し、特定の要件に対する合否結果を提供します。検索結果はwebhook APIを介して認証されたユーザが利用でき、開発パイプライン上で合否タスクを自動化するために使用できます。 ※Container Securityで対処できるOSは、CentOSとRHELに限ります。

1.3 マルウェアの埋め込み	
リスク概要	イメージはパッケージ化されたファイルの集まりであるため、悪意のあるファイルが意図的または不注意に含まれている可能性があります。 このようなマルウェアは、イメージ内の他のコンポーネントと同じ機能を持つため、環境内の他のコンテナまたはホストを攻撃するために使用される可能性があります。 埋め込まれたマルウェアの原因として考えられるのは、信頼されないサードパーティが提供するベースレイヤーやイメージの使用です。
対応方法	組織は、すべてのイメージに対して埋め込まれたマルウェアを継続的に監視する必要があります。 監視プロセスには、マルウェアパターンファイルの使用と、実際の攻撃の挙動に基づいた行動検出ヒューリスティックを含める必要があります。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、Trend Micro Smart Protection Networkからマルウェア情報を取得し、Windowsイメージの場合、機械学習型検索を使用して脅威を検出します。 イメージは継続的に監視され、最新のマルウェアパターンを使用してスキャンされます。

1.4 平文パスワードの埋め込み	
リスク概要	多くのアプリケーションでは、コンポーネント間の安全な通信を可能にするために秘密情報が求められます。 たとえば、Webアプリケーションはバックエンドにあるデータベースに接続するためにユーザ名とパスワードを必要とする場合があります。 埋め込まれた秘密情報の他の例には、接続文字列、SSH秘密鍵、およびX.509秘密鍵が含まれます。 アプリケーションをイメージにパッケージ化すると、これらの秘密情報がイメージに直接埋め込まれてしまいます。ただし、この方法ではイメージにアクセスできる人は誰でも簡単に解析してこれらの秘密情報を知ることができるため、セキュリティ上のリスクが生じます。
対応方法	暗号鍵などの秘密情報はイメージの外部に保存し、実行時に必要に応じて動的に提供する必要があります。 Docker SwarmやKubernetesなどのほとんどのオーケストレーターには、秘密情報の管理が含まれています。組織はこれらのツールを使用して、データベース接続文字列をWebアプリケーションコンテナに安全にプロビジョニングできます。オーケストレーターは、Webアプリケーションコンテナのみがこの秘密情報にアクセスできること、それがディスクに保持されないこと、およびWebアプリがデプロイされるたびに秘密情報が提供されることを確認できます。 組織は、すでに非コンテナ環境で秘密情報を保持するために使用されている既存の企業秘密情報管理システムと、コンテナのデプロイを統合することもできます。これらのツールは通常、コンテナがデプロイされたときに秘密情報を安全に取得するAPIを提供します。これにより、秘密情報をイメージ内に保持する必要がなくなります。選択したツールに関係なく、組織は事前に定義され管理者が制御する設定に基づいて、秘密情報が必要な特定のコンテナにのみ提供されるようにする必要があります。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは秘密情報、およびSSHやX509秘密鍵を含む鍵のカスタムスキャンを提供します。 ユーザはカスタムルールを作成して、他の機密性の高いクリアテキスト文字列を検出することもできます。

1.5 信頼できないイメージの利用	
リスク概要	あらゆる環境において最も一般的なハイリスクな攻撃シナリオのひとつに、信頼できないソフトウェアの実行があります。コンテナの移植性の高さと再利用の容易さにより、十分に検証されていない、または信頼できない外部ソースが作成したイメージを実行する可能性があります。 たとえば、Webアプリケーションの問題のトラブルシューティングを行う場合、ユーザはサードパーティが提供するイメージでそのアプリケーションの別のバージョンを利用できる場合があります。 この外部提供のイメージを使用すると、マルウェアの侵入、データの漏洩、脆弱性のあるコンポーネントの組み込みなど、外部ソフトウェアが従来持っていたのと同じタイプのリスクが生じます。
対応方法	組織は、信頼できるイメージとレジストリを用いて、信頼できるイメージのみをコンテナ環境で実行できるようにして、信頼されていないコンポーネントや悪意のあるコンポーネントが展開されるリスクを軽減する必要があります。これらのリスクを軽減するには、以下を含む多層アプローチを取る必要があります。 1. 環境内で信頼できるイメージとレジストリを厳密に一元管理 2. NISTで認証済みの実装を使用した、暗号署名による各イメージの識別 3. 環境内のすべてのホストが、承認済みリストのイメージのみ実行することを保証 4. イメージが信頼できるものであることを確認するためのイメージ実行前の署名の検証 5. ソースが改ざんされていないことの確認 6. リポジトリの継続的な監視とメンテナンス
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、Trend Micro Smart Protection Networkからマルウェア情報を取得し、Windowsイメージの場合、機械学習型検索を使用して脅威を検出します。 イメージは継続的に監視され、最新のマルウェアパターンを使用してスキャンされます。 また、コンテナイメージ内のコンテナOSおよびアプリケーションにパッチ未適用の脆弱性が含まれているかをチェックすることもできます。 ※スキャン対象のサポート対応OSは下記をご参照ください。 https://cloudone.trendmicro.com/docs/jp/container-security/sc-about/

2. レジストリリスク

2.1 レジストリへの安全でない接続	
リスク概要	イメージには多くの場合、組織独自のアプリケーションや埋め込まれた秘密情報などの機微なコンポーネントが含まれています。 レジストリへの接続が安全でない通信を介して実行される場合、イメージの内容は、平文で送信される他のデータと同じ機密性に対するリスクにさらされます。 また、レジストリを対象としたネットワークトラフィックを傍受し、そのトラフィック内の開発者または管理者の認証情報を盗み、オーケストレーターに不正、または古くなったイメージを提供する可能性がある中間者攻撃のリスクが高くなります。
対応方法	組織は暗号化された通信を介してのみレジストリに接続するように、開発ツール、オーケストレーター、およびコンテナランタイムを構成する必要があります。 具体的な手順はツールによって異なりますが、重要な目標は、レジストリにプッシュおよびプルされるすべてのデータが信頼できるエンドポイント間で発生し、転送中に確実に暗号化されることです。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Security自身は安全な暗号化された通信を介してレジストリに接続します。 ※Container Securityでは、開発ツールやオーケストレーター、コンテナランタイムとレジストリの通信を暗号化する機能は提供していません。

2.2 古いコンテナイメージの残存	
リスク概要	レジストリは通常、組織が利用するすべてのイメージのソースであるため、時間の経過とともに、保存するイメージには多くの脆弱性を含む古いバージョンが存在することがあります。 これらの脆弱なイメージは、単にレジストリに保存されるだけなら直接脅威をもたらすことはありませんが、既知の脆弱性のあるバージョンが偶発的に展開される可能性を高めてしまいます。
対応方法	古いイメージを使用するリスクは、2つの主要な方法で軽減できます。 1. 組織は、使用すべきではない脆弱性を含む安全でないイメージの登録を削除する必要があります。このプロセスは、タイムトリガーとイメージに関連付けられたラベルに基づいて自動化できます。 2. 運用上、使用するイメージの中で特定バージョンを指定するための一意の名前を利用してイメージにアクセスする必要があります。たとえば、my-appというイメージを使用するようにデプロイメントジョブを構成するのではなく、特定のバージョンのイメージ（my-app：2.3やmy-app：2.4など）をデプロイするように設定し、各ジョブの一部として脆弱性汚含まないイメージのインスタンスがデプロイされるようにします。 別の方法として、イメージに[Latest]タグ（最新タグ）を使用し、デプロイメント自動化でこのタグを参照することも可能です。ただし、このタグはイメージに添付されたラベルにすぎず、いわゆる「鮮度」を保証するものではないため、組織は過度に信頼しないように注意する必要があります。組織が個別の名前を使用するか[Latest]タグ（最新タグ）を使用するかを問わず、自動化で最新の一意の名前を使用するか、[Latest]タグ（最新タグ）が付いたイメージが実際に使用されることを確認するプロセスを導入することが重要です。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、コンテナイメージをスキャンしてコンテナイメージにパッチ未適用の既知の脆弱性を検出します。Container Securityによるレジストリへ継続的なスキャンにより、古いイメージに対しても最新の脆弱性情報を用いてスキャンが実施できます。 スキャンの結果、イメージに脆弱性が含まれ、許容可能なセキュリティリスクのしきい値を下回っている古いイメージは実行できなくなります。

2.3 レジストリへのアクセス時の不十分な認証・権限	
リスク概要	レジストリには、機微または専用アプリの実行や機密データへのアクセスに使用されるイメージが含まれている可能性があるため、アクセス認証が不十分であると知的財産が失われ、アプリに関する重要な技術的詳細が攻撃者にさらされる可能性があります。 さらにレジストリが有効で承認されたソフトウェアのソースとして信頼されている場合、レジストリが侵害されることで、コンテナやホストの侵害につながる可能性があります。
対応方法	独自のイメージまたは機密イメージを含むレジストリへのすべてのアクセスには認証が必要です。 信頼できるエンティティからのイメージのみをレジストリに追加できるよう、レジストリへの書き込みアクセスはすべて認証を必要とする必要があります。たとえば、開発者には自分が担当する特定のリポジトリにのみイメージをプッシュできるようにします。 組織は、既存のアカウント（自社またはクラウドプロバイダのディレクトリサービスなど）との連携を考慮して、それらのアカウントに対して既に設定されているセキュリティ制御を活用する必要があります。レジストリへのすべての書き込みアクセスを監査し、機密イメージの読み取りアクションも同様にログに記録するべきです。 また、継続的な統合プロセスを設定することで、イメージは脆弱性スキャンとコンプライアンス評価に合格した後にのみ許可された担当者によって署名され、レジストリにプッシュできるようにすることができます。組織は、これらの自動スキャンをプロセスに統合して、脆弱なイメージや誤って構成されたイメージの昇格とデプロイを防止する必要があります。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、Kubernetesと連携してセキュリティポリシーに準拠していないコンテナのデプロイをブロックできます。 また脆弱性およびコンプライアンススキャン用のAPIを使用してCI/CDパイプラインと統合できます。ウイルススキャンや脆弱性スキャンなどの検索結果に基づいて、不正アクセスによって設置された、悪意のあるイメージの展開をブロックできます。 ※Container Securityでは、レジストリへのアクセス認証に関する機能は提供しておりません。

3. オーケストレーターリスク

3.1 管理者権限の不適切な割り当て	
リスク概要	これまで多くのオーケストレーターは、それらと対話するすべてのユーザが管理者であることを想定し、これらの管理者が環境全体を制御する必要があるという前提で設計されていました。 ただし多くの場合、1つのオーケストレーターが様々なアプリケーションを実行し、それぞれが異なるチームによって管理され、異なる環境で実行される場合があります。 ユーザ、およびグループに提供されるアクセスの範囲が最低限に制限されていない場合、悪意のある、または不注意なユーザによりオーケストレーターが管理する他のコンテナの操作に影響を及ぼす、操作を妨害するなどの可能性があります。
対応方法	特に制御範囲が広いツールであるオーケストレーターは、ユーザに特定のホスト、コンテナ、およびジョブロールに必要なイメージで特定のアクションを実行する機能のみを付与する最小特権アクセスモデルを使用する必要があります。たとえば、 テストチームのメンバーには、テストで使用されるイメージとそれらの実行に使用されるホストへのアクセスのみを許可し、そのために作成したコンテナのみを操作できるようにする必要があります。テストチームのメンバーは、本番環境で使用されるコンテナへのアクセスが制限されているか、アクセスできないことが望ましい状態です。
トレンドマイクロのソリューション	
Cloud One – Workload Security	オーケストレーターツールとして多く採用されているKubernetesは、攻撃者に侵害されることで許可されていないユーザにアクセスを許可してしまう可能性があります。 Workload Securityは、重要なシステムファイルと構成の変更についてKubernetesを監視して、この攻撃シナリオの発生を防ぎます。 ※Workload Securityでは、管理者権限の適切な割り当てをはじめとする機能は提供しておりません。
Cloud One – Contaner Security	Container Securityは、コンテナランタイム内部のシステムファイルと構成の変更についてKubernetesのファイルを監視して、この攻撃シナリオの発生を防ぎます。

3.2 認証のないアクセス	
リスク概要	オーケストレーターには多くの場合、独自の認証ディレクトリサービスが含まれています。これは、組織内で既に使用されている一般的なディレクトリとは別のものである場合があります。 これらのシステムは厳密に管理されていないため、これにより、オーケストレーターのアカウント管理が脆くなり、「孤立した」アカウントになる可能性があります。 これらのアカウントの多くはオーケストレーター内で高度な特権を持っているため、これらのアカウントの侵害はシステム全体の侵害につながる可能性があります。 コンテナは通常、オーケストレーションツールによって管理され、ホスト固有ではないデータストレージボリュームを使用します。 コンテナはクラスタ内の任意のノードで実行される可能性があるため、実行中のホストに関係なく、コンテナ内のアプリに必要なデータがコンテナで利用可能である必要があります。 同時に、多くの組織は、不正アクセスを防ぐために保管時に暗号化する必要があるデータを管理します。
対応方法	クラスタ全体の管理アカウントアカウントは環境内のすべてのリソースに影響を与える・変更を加えることができるため、これらへのアクセスは厳しく制御する必要があります。組織は、パスワードだけでなく多要素認証を要求するなど、強力な認証方法を使用する必要があります。既存のディレクトリシステムへのシングルサインオンなども有効です。 シングルサインオンにより、オーケストレーター認証が簡素化され、ユーザが強力な認証資格情報を使用しやすくなり、アクセスの監査が一元化され、異常検出がより効果的になります。 保存されているデータの暗号化は多くの場合、コンテナと互換性のないホストベースの機能を使用します。したがって、組織はコンテナで使用されるデータを暗号化するツールを使用して、実行中のノードに関係なくコンテナからデータに適切にアクセスできるようにする必要があります。このような暗号化ツールは、NIST SP 800-111で定義されているものと同じ暗号化アプローチを使用して、不正アクセスと改ざんに対する同じ障壁を提供する必要があります。
トレンドマイクロのソリューション	
Cloud One – Workload Security	オーケストレーターツールとして多く採用されているKubernetesは、攻撃者に侵害されることで許可されていないユーザにアクセスを許可してしまう可能性があります。 Workload Securityは、重要なシステムファイルと構成の変更についてKubernetesを監視して、この攻撃シナリオの発生を防ぎます。 ※Workload Securityでは、認証のないアクセスを止める機能は提供しておりません。
Cloud One – Container Security	Container Securityは、コンテナランタイム内部のシステムファイルと構成の変更についてKubernetesのファイルを監視して、この攻撃シナリオの発生を防ぎます。 ※Container Securityでは、管理者権限の適切な割り当てをはじめとする機能は提供しておりません。

3.3 コンテナ間のネットワークトラフィックの不十分な分離	
リスク概要	ほとんどのコンテナ化された環境では、個々のノード間のトラフィックは仮想オーバーレイネットワーク経由でルーティングされます。通常、このオーバーレイネットワークはオーケストレーターによって管理され、多くの場合、既存のネットワークセキュリティでは対応できません。たとえば、従来のネットワークフィルタでは、Webサーバコンテナから別のホスト上のデータベースコンテナに送信されるデータベースクエリを表示する代わりに、2つのホスト間を流れる暗号化されたパケットのみを表示し、実際のコンテナエンドポイントも送信されているトラフィックも見えません。暗号化されたオーバーレイネットワークは運用上およびセキュリティ上の多くの利点を提供しますが、組織が自身のネットワーク内のトラフィックを効果的に監視できないセキュリティ上の「盲目」のシナリオを生み出す可能性があります。 さらに重要なのは、同じ仮想ネットワークを共有する異なるアプリケーションからのトラフィックのリスクです。社外公開用のWebサイトや社内の財務管理アプリなど、機密レベルの異なるアプリが同じ仮想ネットワークを使用している場合、機密性の高い社内アプリはネットワーク攻撃のより大きなリスクにさらされる可能性があります。たとえば、公開しているWebサイトが侵害された場合、攻撃者は共有ネットワークを使用して財務アプリケーションなどを攻撃できる可能性があります.
対応方法	オーケストレーターは、ネットワークトラフィックを機密性のレベルごとに個別の仮想ネットワークに分離するように構成する必要があります。アプリケーションごとのセグメンテーションも可能ですが、ほとんどの組織およびユースケースでは、ネットワークを機密性のレベルで定義するだけで、管理可能な複雑さでリスクを十分に軽減できます。たとえば、公開アプリケーションは仮想ネットワークを共有でき、社内のアプリケーションは別のアプリケーションを使用でき、2つの間の通信は少数の明確に定義されたインターフェイスを介して行われる必要があります。

3.4 さまざまな重要度のワークロードが混在する環境	
リスク概要	オーケストレーターは通常、主にワークロードの規模と密度を高めることに重点を置いています。 これはデフォルトで、同じホストに異なる機密性のワークロードを配置できることを意味します。 たとえば、デフォルトの構成では、オーケストレーターは、公開しているWebサーバを実行するコンテナを、機密性の高い財務データを処理するコンテナと同じホストに配置できます。これは、そのホストが新規のコンテナ展開時に最も利用可能なリソースを持っているからです。 Webサーバに重大な脆弱性がある場合、これにより、機密性の高い財務データを処理するコンテナが侵害されるリスクが大幅に高くなります。
対応方法	オーケストレーターは、重要度・機密性に応じて特定のホストセットへの展開を分離するように構成する必要があります。これを実装するためには、使用中のオーケストレーターによって異なるものの、一般的なモデルとして、高機密なワークロードが、それほど機密性が高くないワークロードを実行しているホストと同じホストに置かれないようにルールを定義することが有効です。これは、オーケストレーター内でホストを「固定」することや、機密性のレベルに応じて管理されたクラスタを個別に持つことによっても実現できます。 ほとんどのコンテナランタイム環境は、コンテナを相互に、またホストOSから隔離されていますが、場合によっては、同じホストOSで異なる機密レベルのアプリケーションを一緒に実行することは不必要なリスクになる場合があります。目的、機密性、および脅威の状態ごとにコンテナをセグメント化することにより、さらに深い防御を実現できます。アプリケーションの展開を計画するときは、アプリケーションの階層化やネットワークとホストのセグメンテーションなどの概念を考慮する必要があります。たとえば、ホストが財務データベースと公開ブログの両方のコンテナを実行しているとします。通常、コンテナランタイムはこれらの環境を互いに効果的に分離しますが、各アプリケーションのDevOpsチーム間での安全な運用と不必要なリスクの排除も求められます。 したがって、コンテナを関連する機密性でグループ化し、特定のホストカーネルが単一の重要度・機密性のコンテナのみを実行するようにすることがベストプラクティスです。このセグメンテーションは、複数の物理サーバを使用することで実現できますが、最新のハイパーバイザは、これらのリスクを効果的に軽減するのに十分に強力な分離も提供します。前の例は、組織がコンテナに対して2つの機密レベルを持っていることを意味する場合があります。 1つは財務アプリケーション用で、データベースはそのグループに含まれています。もう1つはWebアプリケーション用で、ブログはこのグループに含まれています。組織には、それぞれが単一の機密レベルのコンテナをホストする2つのＶMのプールがあります。たとえば、vm-financialというホストは、財務データベースと税務申告ソフトウェアを実行するコンテナをホストし、vm-webというホストは公開Webサイトとブログをホストします。 この方法でコンテナをセグメント化することにより、セグメントの1つを侵害した攻撃者が他のセグメントにその侵害を拡大することを困難にさせます。攻撃者は、同様の機密レベルの他のコンテナに対して偵察や攻撃を実行できる限られた能力を発揮する可能性がありますが、それを超える更なるアクセスは難しいでしょう。このアプローチにより、キャッシュや一時ファイル用にマウントされたローカルボリュームなどの残留データが、データのセキュリティゾーン内にとどまることが保証されます。前の例から、このゾーニングは、コンテナの終了後にローカルにキャッシュされた財務データが、より低い重要度レベルでアプリケーションを実行しているホストで使用できないことを保証します。 数百のホストと数千のコンテナがある大規模環境では、このセグメンテーションを自動化して実用化する必要があります。一般的なオーケストレーションツールには通常、アプリケーションをグループ化できるという機能が含まれており、コンテナセキュリティツールはコンテナ名やラベルなどの属性を使用してセキュリティポリシーを適用できます。これらの環境では、単純なホスト分離を超えた追加の多層防御もこのセグメンテーションを活用できます。たとえば、組織は個別のホスティングゾーンまたはネットワークを実装して、ハイパーバイザ内でこれらのコンテナを分離するだけでなく、ネットワークトラフィックをより個別に分離して、ある重要度レベルのアプリケーションのトラフィックが他の重要度レベルのトラフィックから分離されるようにすることもできます。

3.5 ノードの信頼性	
リスク概要	環境内に存在するノードの信頼を維持するには、特別な注意が必要です。 オーケストレーターは最も基本的なノードです。脆弱なオーケストレーター構成では、オーケストレーターとその他のコンテナ技術のコンポーネントのリスクが高まります。考えられる結果の例は次のとおりです。 • 不正なホストがクラスタに参加し、コンテナを実行している。 • ひとつのクラスタのホストの侵害によって、クラスタ全体が侵害される。 たとえば、認証に使用される同じキーペアがすべてのノードで共有される場合。 • オーケストレーターとDevOps担当者、管理者、およびホスト間の通信が暗号化されず、認証もされていない。
対応方法	オーケストレーションプラットフォームは、実行するすべてのアプリケーションに対して安全な環境を作成する機能を提供するように構成する必要があります。オーケストレーターは、ノードがクラスタに安全に導入され、ライフサイクル全体を通じて永続的なIDを持ち、ノードとその接続状態の正確なインベントリを提供できるようにする必要があります。組織はオーケストレーションプラットフォームが、クラスタの全体的なセキュリティを損なうことなく、個々のノードの侵害に対して回復力があるように特別に設計されていることを確認する必要があります。侵害されたノードは、クラスタ全体の操作を中断または低下させることなく、クラスタから分離および削除できる必要があります。また、クラスタメンバー間の相互認証されたネットワーク接続とクラスタ内トラフィックのエンド間暗号化を提供するオーケストレーターを選択する必要があります。コンテナイメージは高い移植性があるため、組織が直接制御しないネットワーク間で多くのコンテナ展開が発生する可能性があるため、このシナリオではデフォルトでセキュアな配置が特に重要です。
トレンドマイクロのソリューション	
Cloud One – Workload Security	重要なシステムファイルと構成の変更についてKubernetesのファイルを監視して、この攻撃シナリオの発生を防ぎます。 またノードのホストOSにインストールすることで、ノードの多層防御や脆弱性対策を行うことも可能です。 ※Workload Securityでは、ノードの侵害から自動的に回復できる機能は提供しておりません。

4. コンテナリスク

4.1 ランタイムソフトウェアの脆弱性	
リスク概要	悪意のあるソフトウェアが他のコンテナやホストOS自体のリソースを攻撃できる「コンテナエスケープ」シナリオを、ランタイムアプリケーション内の脆弱性が可能にする場合、特に危険です。 攻撃者は脆弱性を悪用してランタイムソフトウェア自体を侵害し、そのソフトウェアを変更して、攻撃者が他のコンテナにアクセスしたり、コンテナ間の通信を監視したりできるようにすることもできます。
対応方法	コンテナランタイムは脆弱性を注意深く監視する必要があり、問題が検出された場合は、迅速に修正する必要があります。 脆弱なランタイムは、ホスト自体だけでなく、サポートするすべてのコンテナを潜在的に重大なリスクにさらします。 組織はツールを使用して、デプロイされたランタイムでCommon Vulnerabilities and Exposures（CVE）の脆弱性を探し、リスクのあるインスタンスをアップグレードし、オーケストレーターが適切にメンテナンスされたランタイムへのデプロイのみを許可するようにします。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、ノードに対する侵入防止を提供し、コンテナ内に存在する可能性のある脆弱性の悪用から保護します。 ただし、推奨スキャンはコンテナでは機能しないため、事前に構成したポリシーを介してIPSルールを割り当てる必要があります。
Cloud One – Container Security	Container Securityは、レジストリ内のイメージを継続的にスキャンして最新の脆弱性を検出します。コンテナランタイムの脆弱性を利用するウイルスが混入している場合、不正プログラム対策機能が有効です。 さらにコンテナ内のファイル読み書き、コマンド実行を監視することで攻撃の検知が可能です。
Cloud One – Network Security	Network Securityは、ネットワークの前段でコンテナ環境に対する侵入防止を提供し、コンテナ内に存在する可能性のある脆弱性の悪用から保護します。 ただし、コンテナ間の通信は検出できません。

4.2 コンテナからの制約のないネットワークアクセス	
リスク概要	デフォルトでは、ほとんどのコンテナランタイムで、個々のコンテナはネットワーク経由で相互およびホストOSにアクセスできます。コンテナが攻撃を受け、悪意のある動作をする場合、このネットワークトラフィックを許可すると環境内の他のリソースがリスクにさらされる可能性があります。たとえば、侵害されたコンテナは、攻撃者が悪用する他の弱点を見つけるために接続先のネットワークをスキャンするために使用される可能性があります。このリスクは[3.3]で説明したように、分離が不十分な仮想ネットワークのリスクに関連していますが、アプリケーションの「クロストーク」シナリオではなく、コンテナから任意のアウトバウンド先へのフローに重点を置いています。 接続の大部分はコンテナ間で仮想化されるため、コンテナ化された環境での出力ネットワークアクセスの管理はより複雑です。したがって、あるコンテナから別のコンテナへのトラフィックは、最終的な送信元、宛先、またはペイロードを直接示すことなく、ネットワーク上でカプセル化されたパケットのように見える場合があります。コンテナを認識しないツールと運用プロセスは、このトラフィックを検査したり、脅威を表しているかどうかを判断したりすることはできません。
対応方法	コンテナが送信する出力ネットワークトラフィックを制御する必要があります。少なくとも、これらの制御はネットワーク境界で実施し、コンテナが安全なデータをホストする環境からなど、異なる機密レベルのネットワークを越えてトラフィックを送信できないようにする必要があります。インターネットは、従来のアーキテクチャで使用されているパターンに似ていますが、コンテナ間トラフィックの仮想化されたネットワークモデルには追加の課題があります。 複数のホストに展開されたコンテナは通常、暗号化された仮想ネットワークを介して通信するため、従来のネットワークデバイスは多くの場合このトラフィックを検査しません。さらに、コンテナには通常、オーケストレーターによって展開されるときに動的IPアドレスが自動的に割り当てられ、これらのアドレスはアプリのスケーリングと負荷分散に応じて継続的に変化します。 理想的には、既存のネットワークレベルのデバイスとアプリケーションに対応したネットワークフィルタリングの組み合わせを使用する必要があります。アプリケーション対応ツールは、コンテナ間のトラフィックを確認できるだけでなく、コンテナで実行されているアプリケーションに基づいてこのトラフィックをフィルタリングするために使用されるルールを動的に生成できる必要があります。具体的には、アプリ対応ツールは次の機能を提供する必要があります。 • インバウンドポートとプロセスポートバインディングの両方を含む、適切なコンテナネットワーキングサーフェスの自動決定 • コンテナと他のネットワークエンティティ間の「フロー」トラフィックとカプセル化されたトラフィックの両方を介したトラフィックフローの検出 • 組織のネットワーク内の予期しないトラフィックフロー、ポートスキャン、 または潜在的に危険な宛先へのアウトバウンドアクセスなどのネットワーク異常の検出
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、各コンテナでマルウェアと悪意のあるトラフィックを検出できます。
Cloud One – Network Security	Network Securityは、Amazon VPC/VNetからの悪意のある接続をスキャンしてブロックできます。 ただし、コンテナ間の通信は検出できません。

4.3 安全ではないコンテナ実行設定	
リスク概要	コンテナランタイムは通常、多くの構成可能なオプションがあり、それらを不適切に設定するとシステムのセキュリティが低下する可能性があります。たとえば、Linuxコンテナホストでは、許可されるシステムコールのセットは、デフォルトでコンテナの安全な操作に必要なものだけに制限されることがよくあります。このリストを広げると、コンテナとホストOSがコンテナの侵害によるリスクの増加にさらされる可能性があります。同様に、コンテナが特権モードで実行されている場合、コンテナはホスト上のすべてのデバイスにアクセスできるため、ホストOSの一部として本質的に機能し、実行中の他のすべてのコンテナに影響を与えることができます。 安全でないランタイム構成の別の例は、コンテナがホスト上に機密ディレクトリをマウントできるようにすることです。コンテナは、ホストOSファイルシステムをほとんど変更せず、ホストOSの基本機能を制御する場所（Linuxコンテナの場合は/ bootまたは/ etc、Windowsコンテナの場合はC：¥ Windows）を変更しないことを推奨します。侵害されたコンテナがこれらのパスを変更することを許可されている場合、それを使用して特権を昇格し、ホスト自体およびホスト上で実行されている他のコンテナを攻撃できます。
対応方法	組織は、コンテナランタイム構成標準への準拠を自動化する必要があります。Center for Internet Security Docker Benchmark などの文書化された技術的な実装ガイダンスには、オプションと推奨設定の詳細が記載されていますが、このガイダンスの運用は自動化に依存しています。組織はさまざまなツールを使用して、ある時点でコンプライアンスをスキャンおよび評価できますが、そのようなアプローチは拡張できません。代わりに、組織は環境全体の構成設定を継続的に評価し、それらを積極的に実施するツールまたはプロセスを使用する必要があります。さらに、SELinuxやAppArmorなどの必須アクセス制御（MAC）テクノロジーは、Linux OSを実行しているコンテナの制御と分離を強化します。たとえば、これらの技術を使用して、コンテナが特定のファイルパス、プロセス、およびネットワークソケットのみにアクセスできるように追加のセグメンテーションと保証を提供し、侵害されたコンテナでさえホストまたは他のコンテナに影響を与える能力をさらに制限できます。MACテクノロジーは、ホストOSレイヤーでの保護を提供し、特定のファイル、パス、およびプロセスのみがコンテナ化されたアプリケーションにアクセスできるようにします。組織は、すべてのコンテナ展開でホストOSが提供するMACテクノロジーを使用することをお勧めします。 セキュアコンピューティング（seccomp）7プロファイルは、コンテナが実行時に割り当てられるシステムレベルの機能を制限するために使用できるメカニズムです。 Dockerのような一般的なコンテナランタイムには、安全ではなく、通常はコンテナ操作に不要なシステムコールをドロップするデフォルトのseccompプロファイルが含まれています。さらに、カスタムプロファイルを作成してコンテナランタイムに渡し、その機能をさらに制限することができます。少なくとも、組織は、ランタイムによって提供されるデフォルトのプロファイルを使用してコンテナが実行されるようにし、リスクの高いアプリに追加のプロファイルを使用することを検討する必要があります。
トレンドマイクロのソリューション	
Cloud One – Workload Securit	Workload Securityは変更監視機能を使用して強化された構成が変更されないようにします。 ※Workload Securityでは、コンテナの設定を最適化する機能は提供しておりません。
Cloud One – Container Security	コンテナ内のファイル読み書き、コマンド実行を監視することで機密ディレクトリのアクセス検知が可能になります。 また、ホストのファイルシステムをマウントしているイメージのデプロイを制御することができます。

4.4 アプリケーションの脆弱性	
リスク概要	組織がこのガイドで推奨する予防措置を講じている場合でも、実行するアプリの欠陥のためにコンテナが危険にさらされる可能性があります。 これは、コンテナ自体の問題ではなく、コンテナ環境内のソフトウェアの欠陥によるものです。 たとえば、コンテナ化されたWebアプリケーションはクロスサイトスクリプティングの脆弱性に対して脆弱であり、データベースフロントエンドコンテナは（SQL）インジェクションの対象となる場合があります。 コンテナが危険にさらされると、機密情報への不正アクセスを許可する、他のコンテナやホストOSに対する攻撃を可能にするなど、様々な方法で悪用される可能性があります。
対応方法	既存のホストベースの侵入検知プロセスとツールは、コンテナホスト全体で見たときには有効な手段ですが、コンテナ内の攻撃を検知して防ぐことができないことがあります。組織はコンテナを認識し、コンテナで動作するように設計された追加のツールを実装する必要があります。これらのツールは、振る舞いの学習を使用してコンテナ化されたアプリケーションを自動的にプロファイリングし、それらのセキュリティプロファイルを構築して、人間の操作を最小限に抑える必要があります。これらのプロファイルは、実行時に次のようなイベントを含む異常を防止および検出できる必要があります。 • 無効または予期しないプロセス実行 • 無効または予期しないシステムコール • 保護された構成ファイルとバイナリの変更 • 予期しない場所とファイルタイプへの書き込み • 予期しないネットワークリスナーの作成 • 予期しないネットワーク宛先に送信されたトラフィック • マルウェアの保管または実行 コンテナは、ルートファイルシステムを読み取り専用モードで実行すると、具体的に定義されたディレクトリへの書き込みを分離し、前述のツールでより簡単にモニタリングできます。さらに、読み取り専用のファイルシステムを使用すると、改ざんは特定の場所に分離され、アプリケーションの残りの部分から簡単に分離できるため、より侵害に強いコンテナを作成できます。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、ノードに対する侵入防止を提供し、コンテナ内に存在する可能性のある脆弱性の悪用から保護します。 ただし、推奨スキャンはコンテナでは機能しないため、事前に構成したポリシーを介してIPSルールを割り当てる必要があります。
Cloud One – Container Security	Container Securityは、レジストリ内のイメージを継続的にスキャンして最新の脆弱性を検出し、検出された場合、実行中のアプリケーションを保護するために適切な対策を講じることができます。
Cloud One – Network Security	Network Securityは、SQLインジェクションおよび悪意のあるファイルの検出に対する保護を提供します。

4.5 管理されていないコンテナの存在	
リスク概要	不正コンテナとは、環境内に存在する計画外または認可されていないコンテナを指します。 これは、特にアプリケーション開発者がコードをテストする手段としてコンテナを起動する可能性のある開発環境でよく発生します。 これらのコンテナが、脆弱性スキャンと適切な構成の確認を経ていない場合、悪用されやすくなる可能性があります。 したがって、不正なコンテナは、特に開発チームやセキュリティ管理者が気付かないうちに環境内に存在する場合、組織にリスクをもたらします。
対応方法	組織は、開発、テスト、本番環境、およびその他プロジェクトごとに個別の環境を構築し、それぞれがコンテナの展開および管理アクティビティの役割ベースのアクセス制御を提供する特定の制御を備えている必要があります。 コンテナの作成はすべて、個々のユーザIDに関連付けて記録し、アクティビティの明確な監査証跡を提供する必要があります。 さらに組織は、イメージの実行を許可する前に、脆弱性管理とコンプライアンスのベースライン要件を実施できるセキュリティツールを使用することをお勧めします。
トレンドマイクロのソリューション	
Cloud One – Container Security	Container Securityは、レジストリ内のイメージを継続的にスキャンして最新の脆弱性を検出します。脆弱性が検出された場合、実行中のコンテナを保護するために適切な対策を講じることができます。

5. ホストOSリスク

5.1 広い攻撃範囲	
リスク概要	OS上で動作している、ネットワークにアクセス可能なサービスは、攻撃者に潜在的なエントリポイントを提供します。 攻撃対象となりうる領域が大きいほど、攻撃者が脆弱性を見つけてアクセスできる可能性が高くなり、ホストOSとその上で実行されるコンテナのリスクにつながります。
対応方法	コンテナ向けOSを使用している組織の場合、OSはコンテナをホストし、他のサービスと機能を無効にするように特別に設計されているため、通常、脅威は最初から最小限に抑えられます。さらに、これらの最適化されたOSはホスティングコンテナ専用に設計されているため、通常は読み取り専用のファイルシステムを備えており、その他のセキュリティ強化にもつながります。 組織は可能な限り、これらの最小限の機能を携えたOSを使用して攻撃リスクを軽減する必要があります。 コンテナ向けOSを使用できない組織は、NIST SP 800-123、Guide to General Server Securityのガイダンスに従って、ホストの攻撃対象領域を可能な限り減らす必要があります。たとえば、コンテナを実行するホストは、コンテナのみを実行し、Webサーバやデータベースなど、コンテナの外部にある他のアプリケーションは実行しないようにすることが推奨されます。コンテナランタイムだけでなく、コンテナが安全でコンパートメント化された操作に依存しているカーネルなどの低レベルコンポーネントにも、脆弱性と更新を迅速に適用するためにホストを継続的にスキャンする必要があります。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityには、OSへの攻撃を最小限に抑えるため、ファイアウォール、およびアプリケーションのホワイトリスト化するアプリケーションコントロール機能などセキュリティ制御の完全なスタックがあります。また、コンテナおよびホストOSへの脆弱性を利用した攻撃通信を検出・防御できます。
Cloud One – Network Security	Network Securityは、ネットワーク経由で攻撃が行われている場合、攻撃を検出してブロックできます。

5.2 カーネルの共有	
リスク概要	コンテナ向けOSの攻撃対象領域は、汎用OSの攻撃対象領域よりもはるかに小さくなります。 たとえば、汎用OSがデータベースおよびWebサーバアプリケーションを直接実行できるようにするライブラリとパッケージマネージャは含まれていません。ただし、コンテナはソフトウェアレベルでリソースの強力な分離を提供しますが、共有カーネルの使用により、コンテナ向けOSであっても、ハイパーバイザで見られるよりも大きなオブジェクト間攻撃の対象となります。つまり、コンテナランタイムによって提供される分離レベルは、ハイパーバイザによって提供される分離レベルほど高くありません。
対応方法	コンテナのワークロードを機密レベルごとにホストにグループ化することに加えて、組織は同じホストインスタンスでコンテナ化されたワークロードとコンテナ化されていないワークロードを混在させないことを推奨します。 たとえば、ホストがWebサーバコンテナを実行している場合、ホストOS内に直接インストールされるコンポーネントとしてWebサーバ（またはその他のアプリケーション）を実行しないでください。 コンテナ化されたワークロードをコンテナ専用のホストに分離しておくと、コンテナを保護するために最適化された対策と防御をより簡単かつ安全に適用できます。

5.3 ホスト OS コンポーネントの脆弱性	
リスク概要	コンテナ向けのものも含め、すべてのホストOSは、リモート接続の認証に使用される暗号化ライブラリや、一般的なプロセスの呼び出しと管理に使用されるカーネルプリミティブなど、基本的なシステムコンポーネントを提供します。他のソフトウェアと同様に、これらのコンポーネントには脆弱性があり、コンテナテクノロジアーキテクチャの下位にあるため、これらのホストで実行されるすべてのコンテナとアプリに影響を与える可能性があります。
対応方法	組織は、基本的なOSの管理と機能のために提供されるコンポーネントのバージョンを検証するための管理手法とツールを実装することが求められます。コンテナ向けOSは汎用OSよりも最小限のコンポーネントセットですが、脆弱性は存在しえるもので、発見される度に修正が必要です。組織は、OSベンダーまたは他の信頼できる組織が提供するツールを使用して、OS内で使用されるすべてのソフトウェアコンポーネントを定期的にチェックし、更新を適用する必要があります。 OSは、セキュリティアップデートだけでなく、ベンダーが推奨する最新のコンポーネントアップデートでも最新の状態に保つ必要があります。これらのコンポーネントの新しいリリースは、多くの場合、単に脆弱性を修正するだけでなく、追加のセキュリティ保護と機能を追加するため、これはカーネルおよびコンテナランタイムコンポーネントにとって特に重要です。組織によっては、既存のシステムを更新するのではなく、必要な更新で新しいOSインスタンスを単純に再展開することを選択する場合があります。このアプローチも有効ですが、より洗練された運用方法が必要になります。 ホストOSは、ホスト上にデータや状態が一意かつ永続的に保存されておらず、ホストによってアプリケーションレベルの依存関係が提供されていない不変の方法で運用する必要があります。代わりに、すべてのアプリケーションと依存関係をパッケージ化し、コンテナに展開する必要があります。これにより、ホストをほとんどステートレスな方法で運用し、攻撃対象領域を大幅に削減できます。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、脆弱性を突いた不正侵入の防止とマルウェア対策機能をホストOSに提供します。さらに、疑わしい変更とアクティビティを検出できるセキュリティログ監視と変更監視機能でホストOSを保護できます。
Cloud One – Network Security	ネットワーク経由で脆弱性を突いた攻撃が行われている場合、攻撃を検出してブロックできます。

5.4 適切ではないユーザのアクセス権	
リスク概要	通常、コンテナ向けOSはマルチユーザシナリオをサポートするように最適化されていません。組織は、オーケストレーションレイヤを通過するのではなく、ホストに直接ログオンしてコンテナを管理することになるため、さらなるリスクにさらされます。直接管理することにより、 システムとその上のすべてのコンテナにさまざまな変更を加え、特定のアプリケーションコンテナを管理するだけでよいユーザが他の多くの人に影響を与える可能性が生まれます。
対応方法	ほとんどのコンテナ展開はオーケストレーターに依存してホストにジョブを分散しますが、組織はOSへのすべての認証が監査され、ログイン異常が監視され、特権操作を実行するためのエスカレーションがログに記録されることを確認する必要があります。 これにより、ホストに直接ログオンし、特権コマンドを実行してコンテナを操作するなど、異常なアクセスパターンを識別できます。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、ホストOS上の機密ディレクトリへの変更を検出するためのログ監視と変更監視を提供します。

5.5 Host OS ファイルシステムの改ざん	
リスク概要	安全でないコンテナ構成では、ホストボリュームがファイル改ざんのリスクにさらされる可能性があります。 たとえば、コンテナがホストOS上で機密ディレクトリをマウントできる場合、そのコンテナはそれらのディレクトリ内のファイルを変更できます。これらの変更は、ホストおよびホストで実行されている他のすべてのコンテナの安定性とセキュリティに影響を与える可能性があります。
対応方法	まずは必要な最小限のファイルシステム権限でコンテナが実行されていることを確認してください。 コンテナがローカルファイルシステムをホストにマウントすることはほとんどありません。 代わりに、コンテナがディスクに保持する必要があるファイルの変更は、この目的のために特別に割り当てられたストレージボリューム内で行う必要があります。 コンテナは、ホストのファイルシステム、特にオペレーティングシステムの構成設定を含むファイルシステムに機密ディレクトリをマウントすることはできません。
トレンドマイクロのソリューション	
Cloud One – Workload Security	Workload Securityは、ホストOS上の機密ディレクトリへの変更を検出するためのログ監視機能と変更監視を提供します。

トレンドマイクロ株式会社
www.trendmicro.com

東京本社
〒151-0053 東京都渋谷区代々木2-1-1 新宿マインズタワー
TEL.03-5334-3601(法人お問い合わせ窓口) FAX.03-5334-3639

名古屋営業所
〒460-0002 愛知県名古屋市中区丸の内3-22-24 名古屋桜通ビル7F
TEL.052-955-1221 FAX.052-963-6332

大阪営業所
〒532-0003 大阪市淀川区宮原3-4-30 ニッセイ新大阪ビル
13F TEL.06-6350-0330(代表) FAX.06-6350-0591

福岡営業所
〒812-0011 福岡県福岡市博多区博多駅前2-3-7 シティ21ビル7F
TEL.092-471-0562 FAX.092-471-05